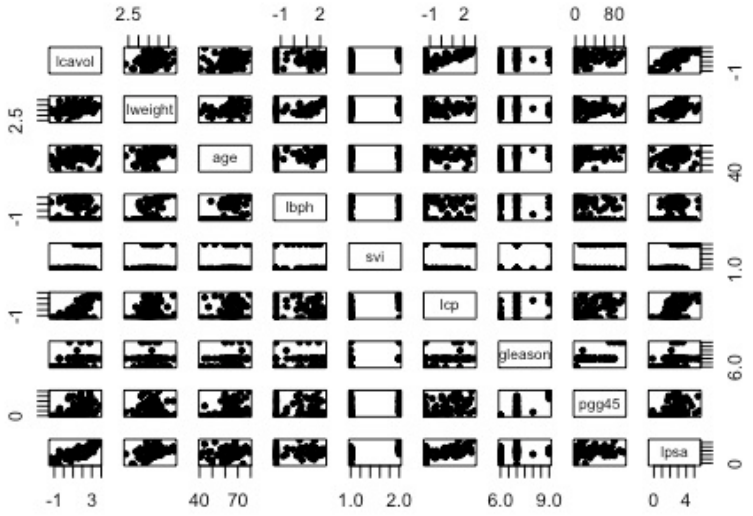
Mini Project #4

Names of group members: Junmei Fan, Xi Cui
Contribution of each group member: 100% for Junmei Fan, 100% for Xi Cui

Question1

(a). Perform an exploratory analysis of the data.



From scatter plot, we can observe the response variable lpsa has strong linear relationship with variable lcavol and lweight. The categorical variable svi clearly separates lpsa into two groups. So lcavol, lweight and svi might be important predictors for response variable lpsa.

(b)

Fit the multiple linear regression to predict lpsa by all variables (ie,lcavol, lweight, svi, age, lbph, lcp, gleason, pgg45)

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_{p-1} x_{i,p-1} + \varepsilon_i$$

Testing the multiple linear model significance:

H_0: $\beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0$ VS H_a: at least one β_i not equal to 0

Test statistic: F = $\frac{MSreg}{MSerr} = 21.68 \sim F_{8,88}$, P-value =2.2e-16.

Conclusion: As the P-value is small, we reject the H_0 and conclude that there is linear relation ship between lpsa and predictors of lcavol, lweight etc. The multiple linear model is reasonable. The linear mode for the least square is:

However from below the output we find that only coefficients for lcavol, leight and svi1 are significant. So we might need to find the important predictors. The linear model for the least square is:

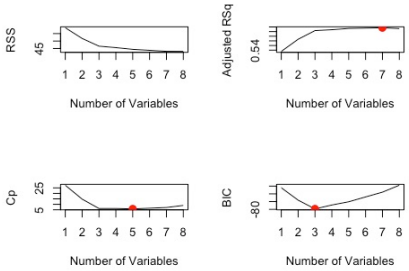
lpsa=0.181561+0.564341lcavol + 0.622020lweight -0.021248 age +0.096713 lbph + 0.096713svi - 0.106051lcp + 0.049228gleason + 0.004458pgg45

R output:

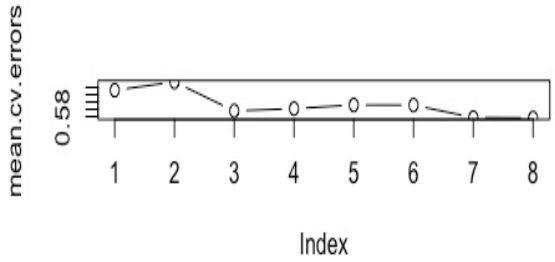
```
> summary.lm(fit.linear)
Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason + pgg45, data = prostate)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.181561   1.320568   0.137  0.89096
lcavol       0.564341   0.087833   6.425 6.55e-09 ***
lweight      0.622020   0.200897   3.096  0.00263 **
age         -0.021248   0.011084  -1.917  0.05848 .
lbph         0.096713   0.057913   1.670  0.09848 .
```

```
svi1         0.761673   0.241176   3.158  0.00218 **
lcp         -0.106051   0.089868  -1.180  0.24115
gleason      0.049228   0.155341   0.317  0.75207
pgg45        0.004458   0.004365   1.021  0.31000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6995 on 88 degrees of freedom
Multiple R-squared:  0.6634,      Adjusted R-squared:  0.6328
F-statistic: 21.68 on 8 and 88 DF,  p-value: < 2.2e-16

Coefficients:
(Intercept)      lcavol     lweight         age        lbph         svi1         lcp
   0.181561    0.564341    0.622020   -0.021248    0.096713    0.761673   -0.106051
    gleason       pgg45
   0.049228    0.004458

> glm.fit <- glm(lpsa ~ lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45, data = prostate)
> cv.err <- cv.glm(prostate, glm.fit, K=10)$delta[1]
> cv.err
[1] 0.5440029
```

(c). Best subset selection:
Conclusion: From below output picture, we can conclude that “elbow” is at 3 variables and the lowest BIC is also at 3. Using 10-fold cross validation, we plot the test error rate versus size for each best model and find that the best model with 3 variables. So we choose the model with 3 variables as the best model. The three variables are lcavol, lweight and svi1. So the best model is: lpsa=-0.7771566+0.5258519lcavol+0.6617699lweight+0.6656666svi1




```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         lcavol lweight age lbph svi1 lcp gleason pgg45
1  ( 1 ) "*"    " "     " "  " "  " "  " "  " "     " "
2  ( 1 ) "*"    "*"     " "  " "  " "  " "  " "     " "
3  ( 1 ) "*"    "*"     " "  " "  "*"  " "  " "     " "
4  ( 1 ) "*"    "*"     " "  "*"  "*"  " "  " "     " "
5  ( 1 ) "*"    "*"     "*"  "*"  "*"  " "  " "     " "
6  ( 1 ) "*"    "*"     "*"  "*"  "*"  " "  " "     "*"
7  ( 1 ) "*"    "*"     "*"  "*"  "*"  "*"  " "     "*"
8  ( 1 ) "*"    "*"     "*"  "*"  "*"  "*"  "*"     "*"
```



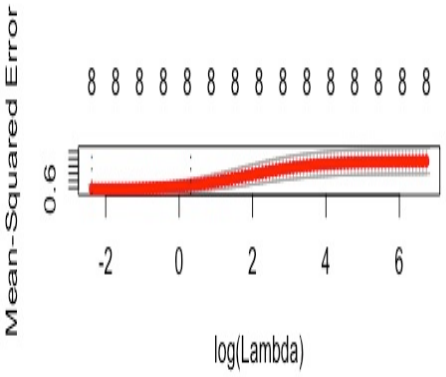

```
> coef(fit.full, 3)
(Intercept)      lcavol     lweight        svi1
 -0.7771566   0.5258519   0.6617699   0.6656666

> err.bestmodel<-min(mean.cv.errors)
> err.bestmodel
[1] 0.5263124
```

(d). Ridge Regression

Using 10-fold cross validation for all the training data, we get the best lambda which corresponds to the minimum mean-squared test error is 0.4501261.

Using this best lambda to fit the ridge regression model, we get the ridge regression model:

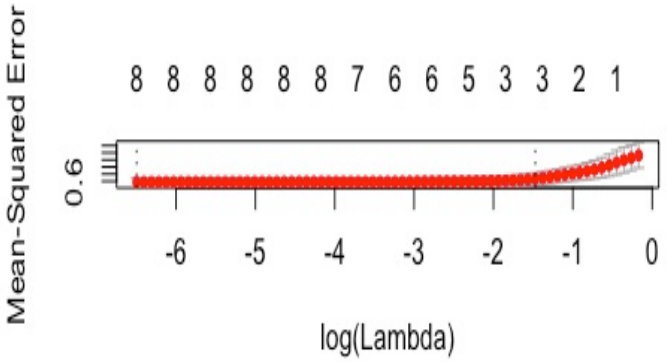
lpsa=0.001186317+0.486637989lcavol+0.602180412lweight-0.016375895age+0.084978159lbph+0.680024170svi1-0.035177998lcp+0.064723335gleason+0.003351327pgg45




```
> bestlam
[1] 0.09256606
> ridge.coef
9 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept)  0.001186317
lcavol       0.486637989
lweight      0.602180412
age         -0.016375895
lbph         0.084978159
svi1         0.680024170
lcp         -0.035177998
gleason      0.064723335
pgg45        0.003351327

> err.ridge<-min(cv.out$cvm)
> err.ridge
[1] 0.574035
```

(e). Lasso Regression
Using 10-fold cross validation for all the training data, we get the best lambda which corresponds to the minimum mean-squared test error is: 0.001508596.
Using this best lambda to fit the lasso regression model, we get the lasso regression model:
lpsa=0.180169065+0.560646399lcavol+0.6189780841lweight-0.020656789age+0.095156007lbph+0.751227602svi1-0.098783143lcp+0.047264205gleason+0.004319271pgg45

> bestlam
[1] 0.001508596



> lasso.coef
9 x 1 sparse Matrix of class "dgCMatrix"
1
(Intercept) 0.180169065
lcavol 0.560646399
lweight 0.618978084
age -0.020656789
lbph 0.095156007
svi1 0.751227602
lcp -0.098783143
gleason 0.047264205
pgg45 0.004319271

> mErr.lasso<-min(cv.out$cvm)
> mErr.lasso
[1] 0.5753727

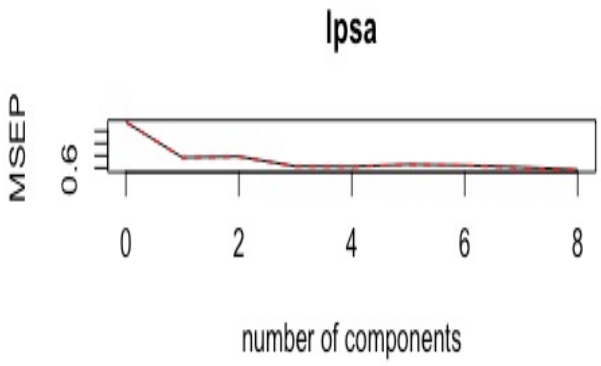
(f) PCR
Using 10-fold cross validation for PCR analysis, from RMSEP, we know that the model with 8 components has smallest error rate 0.5600426 which is slightly smaller than ridge and lasso regression but higher than best subset model and linear model with least square. Even though all 8 components containing all the predictor information, it could only explain 66.34% variance of the lpsa response, which is equal to the variance explained by linear model using least square.

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
CV	1.16	0.8788	0.8873	0.7936	0.7902	0.8115	0.8030	0.7841	0.7566
adjCV	1.16	0.8760	0.8843	0.7905	0.7865	0.8075	0.7983	0.7783	0.7517

TRAINING: % variance explained

	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps	7 comps	8 comps
X	42.01	62.61	74.81	82.71	88.75	94.28	97.56	100.00
lpsa	47.04	47.67	58.61	59.45	60.73	61.66	64.21	66.34



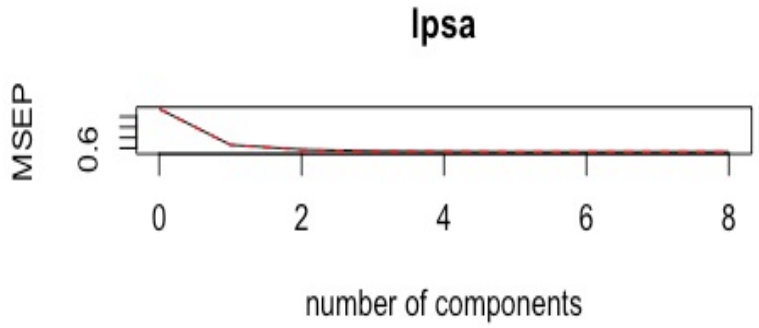

```
> pcr.fit <- pcr(lpsa ~ ., data = prostate, scale = TRUE, ncomp = 8)
> summary(pcr.fit)
Data:   X dimension: 97 8
        Y dimension: 97 1
Fit method: svdpc
Number of components considered: 8
TRAINING: % variance explained
     1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X      42.01    62.61    74.81    82.71    88.75    94.28    97.56   100.00
lpsa   47.04    47.67    58.61    59.45    60.73    61.66    64.21    66.34

> coef(pcr.fit, intercept=T)
, , 8 comps
                lpsa
(Intercept)  0.18156086
lcavol       0.66514667
lweight      0.26648026
age         -0.15819522
lbph         0.14031117
svi1         0.31532888
lcp         -0.14828568
gleason      0.03554917
pgg45        0.12571982

> err.pcr<-min(MSEP(pcr.fit)$val[1,1,])
> err.pcr
0.5600426
```

(g). PLS

Using 10-fold cross validation for PLS analysis which considering response variable, we know that the model with 5 components has smallest error rate 0.552916, which is slightly smaller than ridge and lasso regression and and pcr, but slightly higher than best subset model and linear model with least square. Comparing to pcr, the first component of pls could explain 54.62% of the response variable, while the 1st component of pcr could only explain 47.04% of the response variables. The first a few components are much more efficient in explaining the response variables. Even though 5 components containing 84.35% of the predictor information, it could explain 66.32% variance of the lpsa response, which is very close to the variance explained by linear model using least square or by all 8 components of pcr.

```
> set.seed(1)
> pls.fit <- plsr(lpsa ~ ., data = prostate, scale = TRUE, validation = "CV", segments = 10)
> summary(pls.fit)
Data:   X dimension: 97 8
```

```
          Y dimension: 97 1
Fit method: kernelpls
Number of components considered: 8

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
CV           1.16   0.8321   0.7542   0.7467   0.7472   0.7436   0.7438   0.7440   0.7441
adjCV        1.16   0.8289   0.7508   0.7432   0.7428   0.7394   0.7396   0.7398   0.7399

TRAINING: % variance explained
     1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X      41.30    53.96    66.37    78.64    84.35    90.34    94.36   100.00
lpsa   54.62    63.55    65.31    66.12    66.32    66.34    66.34    66.34
```




```
> summary(pls.fit)
> err.pls<-min(MSEP(pls.fit)$val[1,1,])
> err.pls
[1] 0.552916
> pls.fit <- plsr(lpsa ~ ., data = prostate, scale = TRUE, ncomp = 5)
> summary(pls.fit)
Data:     X dimension: 97 8
          Y dimension: 97 1
Fit method: kernelpls
Number of components considered: 5
TRAINING: % variance explained
     1 comps  2 comps  3 comps  4 comps  5 comps
X      41.30    53.96    66.37    78.64    84.35
lpsa   54.62    63.55    65.31    66.12    66.32

> coef(pls.fit, intercept=T)
, , 5 comps
                lpsa
(Intercept)  0.00180846
lcavol       0.66656056
lweight      0.26406838
age         -0.14987480
lbph         0.13647612
svi1         0.31916031
lcp         -0.14663135
gleason      0.05112294
pgg45        0.10262395
```

(h). Summary Table: Estimated coefficients and test error results, for different subset and shrinkage methods applied to the prostate data. The blank entries correspond to variables omitted.

Term	LS	Best Subset	Ridge	Lasso	PCR	PLS
Intercept	0.181561	-0.7771566	0.001186317	0.180169065	0.18156086	0.00180846
lcavol	0.564341	0.5258519	0.486637989	0.560646399	0.66514667	0.66656056
lweight	0.622020		0.602180412	0.618978084	0.26648026	0.26406838
age	-0.021248		-0.016375895	-0.020656789	-0.15819522	-0.14987480
lbph	0.096713		0.084978159	0.095156007	0.14031117	0.13647612
svi	0.761673	0.6656666	0.680024170	0.751227602	0.31532888	0.31916031
lcp	-0.106051		-0.035177998	-0.098783143	-0.14828568	-0.14663135
gleason	0.049228		0.064723335	0.047264205	0.03554917	0.05112294
Pgg45	0.004458		0.003351327	0.004319271	0.12571982	0.10262395
Test Error	0.5440029	0.5263124	0.574035	0.5753727	0.5600426	0.552916

The above table shows the coefficients from a number of different selection and shrinkage methods. They are best-subset selection using an all-subsets search, ridge regression, the lasso, principal components regression and partial least squares. Each method has a complexity parameter, and this was chosen to minimize an estimate of prediction error based on ten fold cross-validation. Considering the lowest test error rate across all different models, best subset model is the best model. And the best subset model has the least number of predictors. Therefore we chose the best subset model as the best model. So lpsa=-0.7771566+0.5258519lcavol+0.6617699lweight+0.6656666svi1 is the best model.

R code:

```
install.packages("ElemStatLearn")
library(ElemStatLearn)
install.packages("car")
library(car)
attach(prostate)
?prostate
summary(prostate)
str(prostate)
prostate$svi<-factor(prostate$svi,level=c('0','1'))
str(prostate)
#take all observation as the training data
prostate$train<-NULL
head(prostate)
plot(prostate,pch=20)
#Linear Model
fit.linear<-lm(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45,data=prostate)
fit.linear
summary.lm(fit.linear)
summary.aov(fit.linear)
fit.linear$coefficients
#cross validation to get error rate
library(boot)
glm.fit <- glm(lpsa ~ lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45, data = prostate)
cv.err <- cv.glm(prostate, glm.fit, K=10)$delta[1]
cv.err
#Best-subset selection
totpred <- ncol(prostate) - 1
library(leaps)
fit.full <- regsubsets(lpsa ~ ., prostate, nvmax = totpred)
fit.summary <- summary(fit.full)
fit.summary
names(fit.summary)
# Cross-validation approach (using best subset selection)
k <- 10 # No. of folds
n<-nrow(prostate)
set.seed(1)
# Split the observations into 10 folds
```

```
f<-ceiling(n/k)
#folds <- sample(rep(1:k, length.out=nrow(prostate)), nrow(prostate))
folds<-sample(rep(1:k,f), n)
# Create a k x totpred matrix to store test errors for
# each fold number and model size combination
cv.errors <- matrix(NA, k, totpred, dimnames = list(NULL, paste(1:totpred)))

# Write a function to easily get predictions for a model
# from a regsubsets object
predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars] %*% coefi
}

for (j in 1:k) {
  # Best subset selection on the training folds
  best.fit <- regsubsets(lpsa ~ ., data = prostate[folds != j,
                                        ], nvmax = totpred)
  # Prediction on the test fold
  for (i in 1:totpred) {
    # Using the predict.regsubsets function written above
    pred <- predict(best.fit, prostate[folds == j, ], id = i)
    cv.errors[j, i] = mean((prostate$lpsa[folds == j] - pred)^2)
  }
}

# Get the mean for each column (model size)

mean.cv.errors <- apply(cv.errors, 2, mean)

# Plot cv errors against model size
par(mfrow = c(1, 1))
plot(mean.cv.errors,type="b")
which.min(mean.cv.errors)
#nbest=3
fit.best<-regsubsets(lpsa ~ ., prostate, nbest = 3)
?plot.regsubsets
plot(fit.full, scale = "r2")
plot(fit.full, scale = "adjr2")
plot(fit.full, scale = "Cp")
plot(fit.full, scale = "bic")

# Get coefficients of best model for a given size
coef(fit.full, 3)
#error rate
err.bestmodel<-min(mean.cv.errors)
err.bestmodel
# Ridge Regression #
y <- prostate$lpsa
x <- model.matrix(lpsa ~ ., prostate)[, -1]
install.packages("glmnet")
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
# Fit ridge regression for each lambda on the grid
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
plot(ridge.mod, xvar = "lambda")
# Use 10 fold cross-validation to estimate test MSE from training data
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha = 0)
```

```
?cv.glmnet
#cv.out
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam
# Refit the model on the full dataset
out <- glmnet(x, y, alpha = 0, lambda=bestlam)
ridge.coef <- predict(out, type = "coefficients", s = bestlam)
ridge.coef
# Test MSE for the best value of lambda
err.ridge<-min(cv.out$cvm)
err.ridge
#ridge.pred <- predict(out, s = bestlam, newx = x)
#mean((ridge.pred - y)^2)

# Lasso #
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid)
plot(lasso.mod, xvar = "lambda")
# Use cross-validation to estimate test MSE using training data
set.seed(1)
?cv.glmnet
cv.out <- cv.glmnet(x, y, alpha = 1)
plot(cv.out)
bestlam <- cv.out$lambda.min
#cv.out$cvm
bestlam
out <- glmnet(x, y, alpha = 1, lambda = bestlam)
lasso.coef <- predict(out, type = "coefficients", s = bestlam)
lasso.coef
#error rate
mErr.lasso<-min(cv.out$cvm)
mErr.lasso

#PCR
install.packages('pls')
library(pls)
set.seed(2)
pcr.fit <- pcr(lpsa ~ ., data = prostate, scale = TRUE, validation = "CV", segments = 10)
#results
summary(pcr.fit)
#get MSE
MSEP(pcr.fit)
sqrt(MSEP(pcr.fit)$val[1, 1,])
which.min(MSEP(pcr.fit)$val[1, 1,])
#plot the cross-validation test MSE estimates
validationplot(pcr.fit, val.type = "MSEP")
# We see that lowest cross-validation error occurs when M = 8 -->
# Compute test MSE for M = 8
err.pcr<-min(MSEP(pcr.fit)$val[1,1,])
err.pcr
#fit the model with all the data
pcr.fit <- pcr(lpsa ~ ., data = prostate, scale = TRUE, ncomp = 8)
summary(pcr.fit)
pcr.fit$coefficients
coef(pcr.fit, intercept=T)

# PLS #
# Fit PLS on training data
set.seed(1)
pls.fit <- plsr(lpsa ~ ., data = prostate, scale = TRUE, validation = "CV", segments = 10)
summary(pls.fit)
# We see that lowest cross-validation error occurs when M = 5 -->
```

```
# Compute test MSE for M = 5 -->
#test error rate for pls
err.pls<-min(MSEP(pls.fit)$val[1,1,])
err.pls
# This result is slightly higher than those for PCR and lasso -->
# Refit the model on the full dataset  -->
pls.fit <- plsr(lpsa ~ ., data = prostate, scale = TRUE, ncomp = 5)
summary(pls.fit)
#Look at results
validationplot(pls.fit, val.type = "MSEP")
coef(pls.fit, intercept=T)
```