SDS 383C: Homework 2

October 6, 2016

Professor Sarkar

Spencer Woody

Problem 1

(a) A multiple linear regression is performed to predict MPG (fuel efficieny in mi / gallon) of cars from the dataset. the covariates VOL (cab space in ft^3), HP (engine horsepower), SP (top speed in miles per hour) and WT (vehicle weight in hundreds of pounds). Figure 1 reports a summary of this multiple linear regression. A general R function for fitting multiple linear regression is included in the script at the end of this report.

Covariate	Estimate	S.E.	t-statistic	p-value
(intercept)	192.48	23.53161	8.178	4.62×10^{-12}
VOL	-0.01565	0.02283	-0.685	0.495
HP	0.39221	0.08141	4.818	7.131×10^{-6}
SP	-1.29482	0.24477	-5.290	1.11×10^{-6}
WT	-1.85980	0.21336	-8.717	4.22×10^{-13}

Figure 1: Summary of multiple linear regression for predicting MPG

(b) Here we use Mallows's C_p to find a best sub-model under two methods: forward stepwise selection and backward stepwise selection. Mallows's C_p is used as a predictor of test error and is defined as

$$C_p = RSS(p) + 2\hat{\sigma}^2 p \tag{1}$$

where p is the number of non-intercept covariates included in the model. Traceplots for Mallows's C_p for these two methods are shown in Figure 2.

(i) **Forward Stepwise Selection**
Here we start with a null model including only the intercept term. Then we add the most significant covariate, determined by evaluating a model including only one covariate at a time and then choosing the covariate with minimal C_p out of all of these options. This process is repeated iteratively until we reach a point where adding another covariate cannot reduce C_p. We include three variables before reaching this point, which are, in order, WT, SP and HP.

(ii) **Backward Stepwise Selection**
Now we start with a full model and remove the least significant covariate, determined by evaluating a p models, each one calculated by removing one covariate at a time, and choosing the covariate at which the model achieves minimal C_p. This process is repeated iteratively until we reach a point where removing an additional convariate cannot reduce C_p. We only remove one covariate here, which is VOL.

(c) Now we use the Zheng-Loh model selection method to select a sub-model. We order the non-intercept covariates from most to least significant according to their calculated t-statistic from the full model, and then we find the optimal number of covariates

$$j^* = \arg\min_j \left\{ RSS(j) + j\hat{\sigma}^2 \log N \right\} \tag{2}$$

where $RSS(j)$ is the residual sum of squares of the linear model containing the first j most significant non-intercept covariates covariates, $\hat{\sigma}^2$ is the estimated variance under the full model, and N is the number of data points.

In this particular case, the most significant covariates in decreasing order, as seen in Figure 1, are WT, SP, HP, and VOL. Figure 3 demonstrates that we achieve optimality at $j = 3$, meaning that we include WT, SP, and HP. *Forward stepwise selection, backward stepwise selection, and the Zheng-Loh method all give the same best sub-model.*

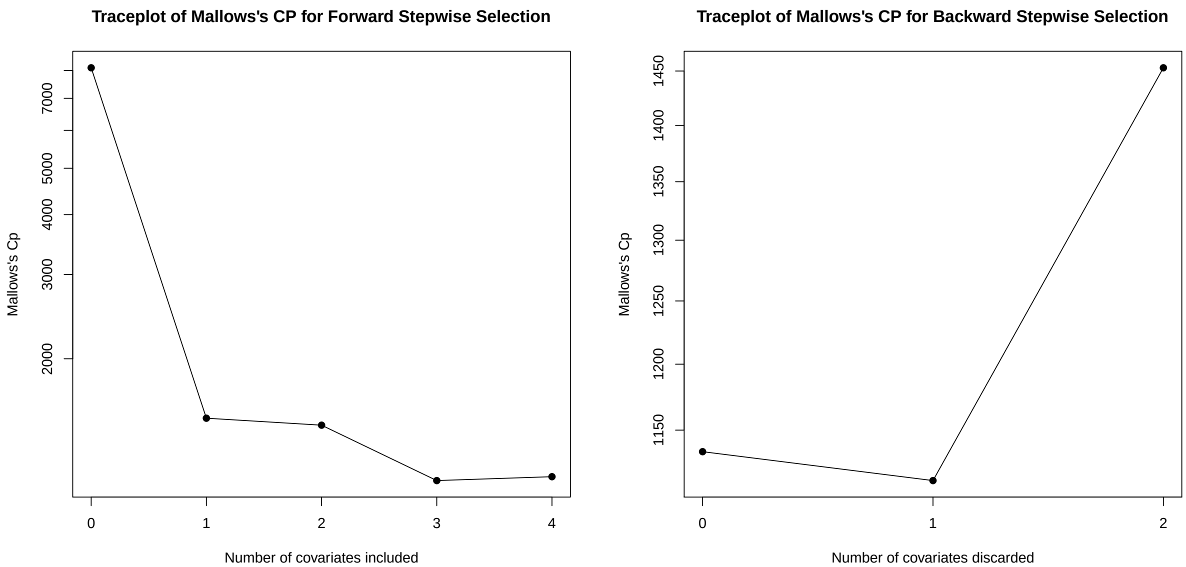

Figure 2: Forward stepwise selection and backward stepwise selection

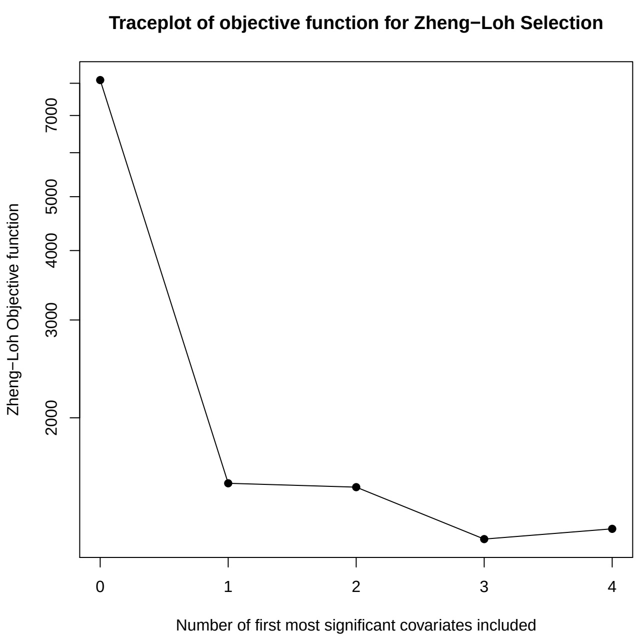

Figure 3: Zheng-Loh Model Selection

Problem 2

(a) We use the `leaps` package within R to perform best-subset selection. Results are shown in Figure (4), along with AIC, BIC, estimated RSS for five-fold cross validation, and estimated RSS for ten-fold cross validation. For cross validation, we use the one-standard error rule to select the number of predictors to include in our model. For five-fold, we include one predictor, and for ten-fold we include 2 predictors. Best subset selection tells us that the predictors to include are lcaval and lweight.

p	(Int)	lcaval	lweight	age	lbph	svi	lcp	gleason	pgg45	AIC	BIC	CV_5	CV_{10}
1	•	•								32.82	-43.26	**0.6966**	0.6929
2	•	•	•							19.26	**-51.30**	0.5792	**0.5998**
3	•	•	•			•				16.57	-51.16	0.6474	0.6897
4	•	•	•		•	•				**13.81**	-51.09	0.6269	0.5816
5	•	•	•		•	•			•	15.10	-48.43	0.6496	0.6369
6	•	•	•		•	•	•		•	13.90	-47.50	0.5985	0.5700
7	•	•	•	•	•	•	•		•	14.07	-45.76	0.5619	0.5436
8	•	•	•	•	•	•	•	•	•	18.05	-41.58	0.5705	0.5560

Figure 4: Best-subset linear regression analysis

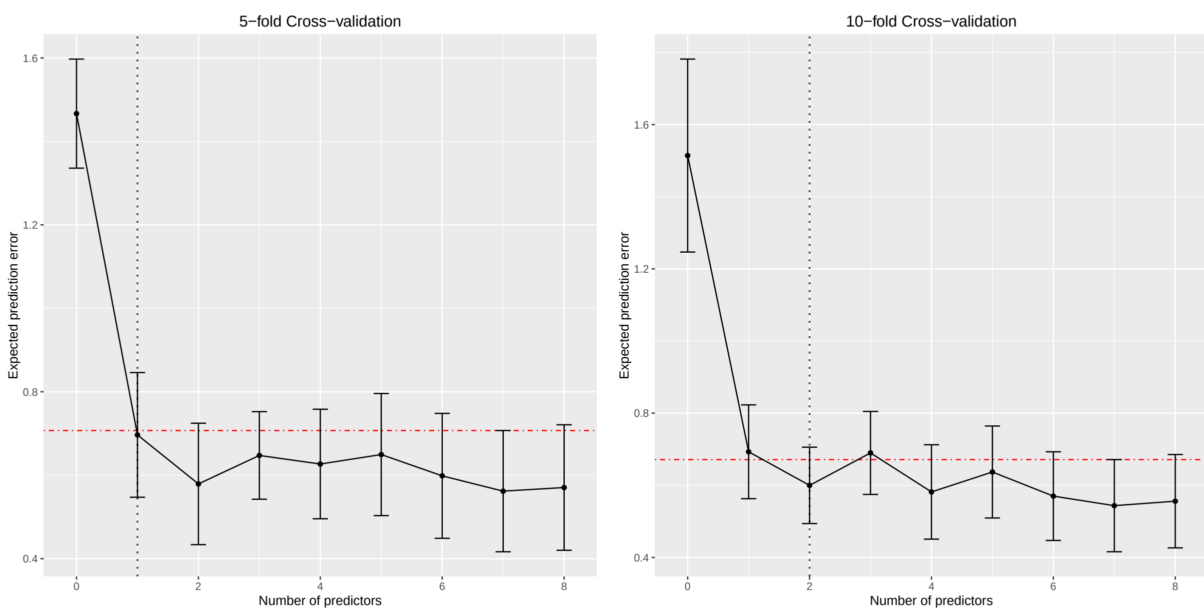

Figure 5: Cross-Validation for $b = 5$ and $b = 10$ bins

(b) The book's model selection, whereby the two variables lcaval and lweight are chosen, is in line what is given in the table above from ten-fold cross validation and BIC.

Problem 3

We define

$$R_{\text{tr}}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2 \tag{3}$$
$$= \frac{1}{N}(y - X\beta)^T(y - X\beta) \tag{4}$$

and

$$R_{\text{te}}(\beta) = \frac{1}{M}\sum_{j=1}^{M}(\tilde{y}_j - \beta^T \tilde{x}_j)^2 \tag{5}$$
$$= \frac{1}{M}(\tilde{y} - \tilde{X}\beta)^T(\tilde{y} - \tilde{X}\beta). \tag{6}$$

Because all pairs (x, y) are i.i.d., the expected value of $R_{\text{tr}}(\beta)$ is calculated as

$$E\left[R_{\text{tr}}(\beta)\right] = E\left[\frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2\right] \tag{7}$$
$$= \frac{1}{N}E\left[\sum_{i=1}^{N}(y_i - \beta^T x_i)^2\right] \tag{8}$$
$$= \frac{1}{N}\sum_{i=1}^{N}E\left[(y_i - \beta^T x_i)^2\right] \tag{9}$$
$$= \frac{1}{N} \times N \times E\left[(y_k - \beta^T x_k)^2\right], k \in \{1, 2, \ldots N\} \tag{10}$$
$$= E\left[(y_k - \beta^T x_k)^2\right], k \in \{1, 2, \ldots N\}. \tag{11}$$

Similarly,

$$E\left[R_{\text{te}}(\beta)\right] = E\left[(\tilde{y}_l - \beta^T \tilde{x}_l)^2\right], l \in \{1, 2, \ldots M\} \tag{12}$$

Note that $\hat{\beta}$ is defined as

$$\hat{\beta} = \arg\min_{\beta}(y - X\beta)^T(y - X\beta). \tag{13}$$

By definition, $\hat{\beta}$ minimizes Eqn. (3). Suppose we also computed $\tilde{\beta}$ in an analogous way where

$$\tilde{\beta} = \arg\min_{\beta}(\tilde{y} - \tilde{X}\beta)^T(\tilde{y} - \tilde{X}\beta). \tag{14}$$

Because all pairs (x_i, y_i), $i \in 1, 2, \ldots, N$ and $(\tilde{x}_j, \tilde{y}_j)$, $j \in 1, 2, \ldots, M$ are drawn from the same distribution, the respective expected values of $R_{\text{tr}}(\beta)$ and $R_{\text{te}}(\beta)$ evaluated at their respective minimized values are equal, which can be notated as

$$E\left[R_{\text{tr}}(\hat{\beta})\right] = E\left[R_{\text{te}}(\tilde{\beta})\right]. \tag{15}$$

However, when R_{te} is evaluated at $\hat{\beta}$ instead, it will be *at least* as large as when it is evaluated at $\tilde{\beta}$. That is to say,

$$E\left[R_{\text{te}}(\hat{\beta})\right] \geq E\left[R_{\text{te}}(\tilde{\beta})\right]. \tag{16}$$

Combining Eqns. (15) and (16) yields

$$E\left[R_{\text{tr}}(\hat{\beta})\right] \leq E\left[R_{\text{te}}(\hat{\beta})\right]. \tag{17}$$

Problem 4

(a) We have it that $X^T X = I_{p\times p}$. The objective function for the least-squares estimate becomes

$$L(\beta) = \frac{1}{2}(X\beta - y)^T(X\beta - y) \tag{18}$$
$$= \frac{1}{2}(\beta^T X^T - y^T)(X\beta - y) \tag{19}$$
$$= \frac{1}{2}(\beta^T X^T X\beta - \beta^T X^T y - y^T X\beta + y^T y) \tag{20}$$
$$= \frac{1}{2}(\beta^T \beta - 2\beta^T X^T y + y^T y) \tag{21}$$

Taking the gradient with respect to β,

$$\nabla_\beta L(\beta) = \frac{1}{2}(2\hat{\beta} - 2X^T y) = 0 \tag{22}$$

Thus we see that $\hat{\beta} = X^T y$. Hereafter we denote this as β^{OLS}. The ith element of β^{OLS} is $v_i^T y$ where v_i is the ith column of X.

(b) First, let us rearrange our old objective function in Eqn. (18) as

$$L(\beta) = \frac{1}{2}(X\beta - y)^T(X\beta - y) \tag{23}$$
$$= \frac{1}{2}(y^T y - y^T X X^T y + (\beta - X^T y)^T(\beta - X^T y)) \tag{24}$$
$$= \frac{1}{2}(y^T y - y^T y + (\beta - X^T y)^T(\beta - X^T y)) \tag{25}$$
$$= \frac{1}{2}(\beta - X^T y)^T(\beta - X^T y) \tag{26}$$
$$= \frac{1}{2}(\beta - \beta^{\text{OLS}})^T(\beta - \beta^{\text{OLS}}). \tag{27}$$

Now our regularization stated in the problem becomes

$$\tilde{\beta} = \arg\min_{\beta} \frac{1}{2}(\beta - \beta^{\text{OLS}})^T(\beta - \beta^{\text{OLS}}) + \lambda \left\|\beta\right\|_0 . \tag{28}$$

Element-wise, this can be written as

$$\tilde{\beta}_i = \arg\min_{\beta_i} \frac{1}{2}(\beta_i - \beta_i^{\text{OLS}})^2 + \lambda \times \mathbf{1}(\beta_i \neq 0) \tag{29}$$
$$= \arg\min_{\beta_i} \frac{1}{2}(\beta_i - v_i^T y)^2 + \lambda \times \mathbf{1}(\beta_i \neq 0). \tag{30}$$

We can see that the solution to our objective function will take one of two forms. Either $\tilde{\beta}_i$ will be 0, in which case the loss function takes on the value $\frac{1}{2}(v_i^T y)^2$, or $\tilde{\beta}_i$ will be $v_i^T y$, in which case the loss function takes on a value of λ. Any value of $\tilde{\beta}_i$ between these two values will give quadratic part *and* λ in the loss function. The threshold at which $\tilde{\beta}_i$ changes from 0 to $v_i^T y$ is when

$$\frac{1}{2}(v_i^T y)^2 > \lambda. \tag{31}$$

By solving this inequality, we have an element-wise solution to Eqn. (28)

$$\tilde{\beta}_i = \begin{cases} v_i^T y & \text{if } |v_i^T y| > \sqrt{2\lambda} \\ 0 & \text{if } |v_i^T y| \leq \sqrt{2\lambda}. \end{cases} \tag{32}$$

Problem 5

(a) The coefficient vector for ridge regression $\hat{\beta}^{\mathrm{r}}$ is found as follows:

$$\hat{\beta}^{\mathrm{r}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i,1} - \sum_{j=2}^{p} \beta_j x_{i,j} \right)^2 + \lambda \left(\beta_1^2 + \sum_{j=2}^{p} \beta_j^2 \right) \right\}. \tag{33}$$

Once we add $m-1$ copies of variable X_1, our ridge solution becomes

$$\hat{\beta}^{*\mathrm{r}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{k=1}^{m} \beta_{1,k} x_{i,1} - \sum_{j=2}^{p} \beta_j x_{i,j} \right)^2 + \lambda \left(\sum_{k=1}^{m} (\beta_{1,k})^2 + \sum_{j=2}^{p} \beta_j^2 \right) \right\}. \tag{34}$$

This expression is similar to our original objective function because, but now each $x_{i,1}$ is multiplied by a summation of "new" coefficients. If we assume that the number of data points n is large, then the RSS term of the loss function overwhelms the penalty term, and therefore the minimization is largely determined by minimizing the RSS. Since the minimum of the RSS term in Eqn. (33) is achieved at $\hat{\beta}^{\mathrm{r}}$, the RSS term of Eqn. (34) is minimized at the same "fitted" coefficients for each covariate. This gives us

$$\sum_{k=1}^{m} \beta_{1,k}^{\mathrm{r}} x_{i,1} = \beta_1^{\mathrm{r}} = a. \tag{35}$$

We can consider this a constraint under which we must minimize the sum $\sum_{k=1}^{m} \left(\beta_{1,k}^{\mathrm{r}} \right)^2$, the penalty term of Eqn. (34). A sum of squares of elements given some constaint is minimized when all those elements are equal to one another, as a result of the Cauchy-Schwartz inequality. Thus,

$$\beta_{1,k}^{\mathrm{r}} = \frac{a}{m}, \forall \ k \in \{1, 2, \ldots, m\}. \tag{36}$$

(b) The coefficient vector for lasso regression $\hat{\beta}^{\mathrm{lasso}}$ is found as follows:

$$\begin{aligned} \hat{\beta}^{\mathrm{lasso}} &= \arg\min_{\beta} \left\{ (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} && (37) \\ &= \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i,1} - \sum_{j=2}^{p} \beta_j x_{i,j} \right)^2 + \lambda \left(|\beta_1| + \sum_{j=2}^{p} |\beta_j| \right) \right\}. && (38) \end{aligned}$$

Similarly as in (a), we add $m-1$ copies of variable X_1, and our lasso solution becomes

$$\hat{\beta}^{*\mathrm{lasso}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{k=1}^{m} \beta_{1,k} x_{i,1} - \sum_{j=2}^{p} \beta_j x_{i,j} \right)^2 + \lambda \left(\sum_{k=1}^{m} |\beta_{1,k}| + \sum_{j=2}^{p} |\beta_j| \right) \right\}. \tag{39}$$

So now we want to minimize $\sum_{k=1}^{m} |\beta_{1,k}|$ subject to the constraint

$$\sum_{k=1}^{m} \beta_{1,k}^{\mathrm{lasso}} x_{i,1} = \beta_1^{\mathrm{lasso}} = a. \tag{40}$$

However, there is no unique solution to this problem. We can pick any vector $\beta_{1,\bullet}^{\mathrm{lasso}}$ such that its components sum to a and all have the same sign.

```
###################################################
##### Created by Spencer Woody on 26 Sep 2016 #####
###################################################

library(microbenchmark)
library(leaps)
library(ggplot2)
library(xtable)

#
##
# Problem 1
##
#

car <- read.csv("car.csv", header = T)
attach(car)

X <- as.matrix(car[, c(2, 3, 5, 6)])
y <- as.matrix(car[, 4])

N <- nrow(X)
int <- rep(1, N)
X <- cbind(int, X)

mymodel1 <- lm(MPG ~ VOL + HP + SP + WT)
summary(mymodel1)


my.lm <- function(X, y) {
    # Note: this function assumes that X already has an intercept term
    # (or doesn't, if we want to force OLS through the origin)
    N <- nrow(X)
    p <- ncol(X)

    XtX <- crossprod(X)

    # Calculate beta.hat
    beta.hat <- solve(XtX, crossprod(X, y))

    # Calculate predicted values and residuals
    y.hat <- crossprod(t(X), beta.hat)
    res <- y - y.hat

    rss <- sum(res^2)

    # Calculate \hat{sigma^2}
    var.hat <- rss / (N - p)

    # Calculate covariance matrix of beta and SE's of beta
    var.beta <- var.hat * solve(crossprod(X))
    beta.SE <- diag(var.beta) ^ 0.5
```

```
    # Calculate t-score of each beta
    beta.t <- beta.hat / beta.SE

    # Calculate p-values for coefficients
    beta.p <- 2 * (1 - pt(abs(beta.t), N - p))

    # Calculate r-squared and adjusted r-squared
    r.sq <- 1 - rss / sum((y - mean(y))^2)
    r.sqadj <- r.sq - (1 - r.sq) * (p - 1) / (N - p - 2)

    # Create a list of calculated values, return it back
    mylist <- list(Beta.hat = beta.hat, Beta.SE = beta.SE,
                   Beta.t = beta.t, Beta.p = beta.p, RSS = rss, Var.hat = var.hat,
                   R.sq = r.sq, R.sqadj = r.sqadj)
    return(mylist)
}

mymodel2 <- my.lm(X, y)

mymodel2$Beta.hat
mymodel2$Beta.SE
mymodel2$Beta.t
mymodel2$Beta.p

fw.stepwise <- function(X, y) {
    N <- nrow(X)
    p <- ncol(X)
    CP.null <- sum((y - mean(y))^2)
    CP.trace <- CP.null
    keep   <- 1
    remain <- 2:p
    for (i in 2:p) {
        CP.list <- NULL
        for (j in remain) {
            X.j <- X[, c(keep, j)]
            model.j <- my.lm(X.j, y)
            CP.j <- model.j$RSS + 2 * model.j$Var.hat * (ncol(X.j) - 1)
            CP.list[length(CP.list) + 1] <- CP.j
        }
        CP.trace[length(CP.trace) + 1] <- min(CP.list)
        if (min(CP.list) < CP.null) {
            newkeep <- remain[ which(CP.list == min(CP.list)) ]
            keep[length(keep) + 1] <- newkeep
            remain <- remain[- which(remain == newkeep) ]
            CP.null <- min(CP.list)
        }
        else {
            finalvars <- keep
            finalmod  <- my.lm(X[, finalvars], y)
            break
        }
        finalvars <- keep
        finalmod  <- my.lm(X[, finalvars], y)
```

```
    }
    fw.list <- list(finalvars, finalmod, CP.trace)
    return(fw.list)
}


fw <- fw.stepwise(X, y)

CP.tracefw <- fw[[3]]

pdf("forward.pdf")
plot(1:length(CP.tracefw) - 1, CP.tracefw,
log = "y",
type = "l",
ylab = "Mallows's Cp",
xlab = "Number of covariates included",
main = "Traceplot of Mallows's CP for Forward Stepwise Selection")
points(1:length(CP.tracefw) - 1, CP.tracefw, pch = 19)
dev.off()

bw.stepwise <- function(X, y) {
    N <- nrow(X)
    p <- ncol(X)
    fullmodel <- my.lm(X, y)
    CP.null <- fullmodel$RSS + 2 * fullmodel$Var.hat * (ncol(X) - 1)
    CP.trace <- CP.null
    keep <- 2:p
    remove <- NULL
    for (i in 2:p) {
        CP.list <- NULL
        for (j in keep) {
            remove.j <- keep[-which(keep == j)] # Delete one at a time
            X.j <- X[, c(1, remove.j)]
            model.j <- my.lm(X.j, y)
            CP.j <- model.j$RSS + 2 * model.j$Var.hat * (ncol(X.j) - 1)
            CP.list[length(CP.list) + 1] <- CP.j
        }
        CP.trace[length(CP.trace) + 1] <- min(CP.list)
        if (min(CP.list) < CP.null) {
            newremove <- keep[which(CP.list == min(CP.list))]
            remove[length(remove) + 1] <- newremove
            keep <- keep[-which(keep == newremove)]
        }
        else {
            finalvars <- keep
            finalmod  <- my.lm(X[, c(1, finalvars)], y)
            break
        }
    }
    bw.list <- list(finalvars, finalmod, CP.trace)
    return(bw.list)
}
```

```r
bw <- bw.stepwise(X, y)

CP.tracebw <- bw[[3]]

pdf("backward.pdf")
plot(1:length(CP.tracebw) - 1, CP.tracebw,
log = "y",
type = "l",
ylab = "Mallows's Cp",
xlab = "Number of covariates discarded",
xaxt = "n",
main = "Traceplot of Mallows's CP for Backward Stepwise Selection")
axis(1, at = 0:length(CP.tracebw))
points(1:length(CP.tracebw) - 1, CP.tracebw, pch = 19)
dev.off()


zhengloh <- function(X, y) {
    N <- nrow(X)
    p <- ncol(X)
    fullmodel <- my.lm(X, y)
    tvec <- fullmodel$Beta.t[2:p]
    indices.tvec <- cbind(2:p, tvec)
    sorted <- indices.tvec[order(-abs(indices.tvec[, 1])), ]

    zl.list <- sum((y - mean(y))^2)

    for (i in 1:nrow(sorted)) {
        X.i <- X[, c(1, sorted[1:i, 1])]
        model.i <- my.lm(X.i, y)
        zl.i <- model.i$RSS + i * fullmodel$Var.hat * log(N)
        zl.list[i + 1] <- zl.i
    }
    opt <- which(zl.list == min(zl.list))
    vars <- sorted[1:opt, 1]
    return(zl.list)
}

zl <- zhengloh(X, y)

pdf("zl.pdf")
plot(1:length(zl) - 1, zl,
log = "y",
type = "l",
ylab = "Zheng-Loh Objective function",
xlab = "Number of first most significant covariates included",
xaxt = "n",
main = "Traceplot of objective function for Zheng-Loh Selection")
axis(1, at = 0:length(zl))
points(1:length(zl) - 1, zl, pch = 19)
dev.off()

#
```

```
##
# Problem 2
##
#

# START HERE

# Read in data

prostate <- read.table("prostate.txt", header = T)

# Create matrices of covariates

X2 <- as.matrix(prostate[which(prostate$train == TRUE), 1:8])
y2 <- as.matrix(prostate[which(prostate$train == TRUE), 9])
colnames(y2)[1] <- "lpsa"

# Make data frame X2 and y2

mydata <- as.data.frame(cbind(X2, y2))

N2 <- nrow(mydata)
P2 <- ncol(mydata) - 1

# Add intercept column to X2

X2 <- cbind(rep(1, N2), X2)
colnames(X2)[1] <- "(Intercept)"

# Perform best-subset selection

toLatex(xtable(obj, digits = 2))

regsubsets.out <- regsubsets(lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason + pgg45,
    data = mydata, method = "exhaustive")

mysummary <- summary(regsubsets.out)
mysummary$which

# Perform cross validation

numbins <- 10
jumble <- sample(1:N2, N2, replace = F)
bin.indices <- split(jumble, cut(1:N2, numbins))

est.rss <- rep(NA, P2 + 1)
se.rss <- rep(NA, P2 + 1)

for (i in 0:P2) {
    res.vec <- rep(NA, numbins)
    for (j in 1:numbins) {
        # Deifine indices
        indices.j <- bin.indices[[j]]
```

```
        # Create testing data
        mydata.j <- mydata[-indices.j, ]

        y.te <- y2[indices.j]

        if (i == 0) {
            y.hat <- mean(y2[-indices.j])
        }
        else {

        # Perform best subset for i variables
        regsubsetsout.j <- regsubsets(lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason + pg
            data = mydata.j, nvmax = i)
        summary.j <- summary(regsubsetsout.j)

        # Grab coefficients from
        coefs.j <- coef(regsubsetsout.j, i)

        # Create subset of X2 and Y2 which only contain testing data and best subset vars
        colnumbers <- which(colnames(X2) %in% names(coefs.j))
        X.te <- X2[indices.j, colnumbers]

        # Predict
        y.hat <- X.te %*% coefs.j
        }
        # Calculate average RSS, add it to rss vector
        rss <- sum((y.te - y.hat)^2) / length(indices.j)
        res.vec[j] <- rss

    }
    est.rss[i+1] <- mean(res.vec)
    se.rss[i+1] <- sqrt(var(res.vec) / numbins)
}

whichsmallest    <- which(est.rss == min(est.rss))
redline <- est.rss[whichsmallest] + se.rss[whichsmallest]

bools <- est.rss < redline

if (sum(bools) > 0) {
    numvarchoice <- min(which(bools == T)) - 1
} else {
    numvarchoice <- whichsmallest - 1
}


pdf(sprintf("cv%i.pdf", numbins))
qplot(0:P2, est.rss) +
geom_vline(xintercept = numvarchoice, linetype = 3, col = "gray30", size = 0.75) +
geom_hline(aes(yintercept=redline), linetype = "dotdash", col = "red") +
geom_line() +
geom_errorbar(aes(x=0:P2, ymin = est.rss - se.rss, ymax = est.rss + se.rss), width=0.25) +
```

```r
xlab("Number of predictors") +
ylab("Expected prediction error") +
labs(title = sprintf("%i-fold Cross-validation", numbins))
dev.off()

if (numbins == 5) {
    cv.5 <- est.rss[-1]
} else if (numbins == 10) {
    cv.10 <- est.rss[-1]
}

AIC <- mysummary$cp / (mysummary$rss / (N2 - P2))
BIC <- mysummary$bic

# Make sure to run twice, once with numbins = 5, once with numbins = 10

toLatex(xtable(cbind(mysummary$which, AIC, BIC, cv.5, cv.10), digits = 4))
```