STAT 151A HW4 Solutions (excluding 2c)

Billy Fang

October 25, 2017

1

The notation below follows those lecture notes. We use a few facts that were proved in the "Regression Diagnostics 4" notes.

- $b^{(p)} = \widehat{\beta}_p$
- $Y^{(p)} = (I - H(-p))Y$
- $X^{(p)} = (I - H(-p))X(p)$
- $H(-p)Y + (I - H(-p))X(p)\widehat{\beta}_p = HY$

Then,

$$\begin{aligned} e^{(p)} &= Y^{(p)} - b^{(p)}X^{(p)} \\ &= (I - H(-p))Y - (I - H(-p))X(p)\widehat{\beta}_p \\ &= Y - HY \\ &= \widehat{e}. \end{aligned}$$

2

```
dat <- read.csv("bodyfat.csv")
dat <- dat[, -1] # remove density
n <- dim(dat)[1]
p <- dim(dat)[2] - 1
library(leaps)
library(SignifReg)
```

a)

```
alpha <- 0.15
```

i) - ii)

I will use $\alpha = 0.15$ here. Of course, answers may vary with different choices of α.

Both backward and forward selection via p-values yielded the same set of variables (see code below): **Age, Weight, Neck, Abdomen, Thigh, Forearm, Wrist.**

If you use a Bonferroni correction, both methods selected **Weight, Abdomen, Forearm, Wrist.**

```
# using SignifReg
mod.forward <- SignifReg(bodyfat~., data=dat, alpha=alpha,direction="forward",
                         criterion="p-value", correction="None")
mod.backward <- SignifReg(bodyfat~., data=dat, alpha=alpha, direction="backward",
                          criterion="p-value", correction="None")
# Replace "None" with "Bonf" for Bonferroni
mod.backward
```

```
##
## Call:
## lm(formula = reg, data = data)
##
## Coefficients:
## (Intercept)          Age       Weight         Neck      Abdomen
##   -33.25799      0.06817     -0.11944     -0.40380      0.91788
##       Thigh      Forearm        Wrist
##     0.22196      0.55314     -1.53240
```

```
mod.forward
```

```
##
## Call:
## lm(formula = reg, data = data)
##
## Coefficients:
## (Intercept)      Abdomen       Weight        Wrist      Forearm
##   -33.25799      0.91788     -0.11944     -1.53240      0.55314
##        Neck          Age        Thigh
##    -0.40380      0.06817      0.22196
```

```
# sort(names(coef(mod.backward)[-1]))
# sort(names(coef(mod.forward)[-1]))
```

Note the need for `correction="None"` if you do not want a correction. **The default option for `SignifReg` is `correction = "FDR"`.**

A way to do this manually is as follows.

```
# Manual backward
f <- "bodyfat ~ ."
mod.backward.manual <- lm(f, data=dat)
for (j in 1:p) {
  pval <- coef(summary(mod.backward.manual))[-1,4] # take the last column of the summary.lm table
  idx <- which.max(pval)
  varname <- names(pval)[idx]
  if (pval[idx] < alpha) {break} # replace with alpha / p for Bonferroni
  message("Removing ", varname)
  f <- paste0(f, " - ", varname)
  mod.backward.manual <- lm(f, data=dat)
}
```

```
## Removing Knee
```

```
## Removing Chest
```

```
## Removing Height
```

```
## Removing Ankle
```

```
## Removing Biceps
```

```
## Removing Hip
```

```
mod.backward.manual
```

```
##
## Call:
## lm(formula = f, data = dat)
##
## Coefficients:
## (Intercept)          Age       Weight         Neck      Abdomen
##   -33.25799      0.06817     -0.11944     -0.40380      0.91788
##       Thigh      Forearm        Wrist
##     0.22196      0.55314     -1.53240
```

```
# sort(names(coef(mod.backward.manual))[-1])
```

```
# Manual forward
f <- "bodyfat ~ 1"
inactive <- colnames(dat[,-1])
for (j in 1:p) {
  min.pval <- Inf
  min.idx <- -1 # index in "inactive"
  for (k in 1:length(inactive)) {
    mod.tmp <- lm(paste0(f, " + ", inactive[k]), data=dat)
    pval.tmp <- tail(coef(summary(mod.tmp))[,4], 1) # get last p-value
    if (pval.tmp < min.pval) {
      min.pval <- pval.tmp
      min.idx <- k
    }
  }
  if (min.pval > alpha) {break} # replace with alpha / p for Bonferroni
  message("Adding ", inactive[min.idx])
  f <- paste0(f, " + ", inactive[min.idx])
  inactive <- inactive[-min.idx]
}
```

```
## Adding Abdomen
```

```
## Adding Weight
```

```
## Adding Wrist
```

```
## Adding Forearm
```

```
## Adding Neck
```

```
## Adding Age
```

```
## Adding Thigh
```

```
mod.forward.manual <- lm(f, data=dat)
# sort(names(coef(mod.forward.manual)[-1]))
```

iii)

9 variables: **Age, Weight, Neck, Abdomen, Hip, Thigh, Biceps, Forearm, Wrist** (see code below).

iv)

8 variables: **Age, Weight, Neck, Abdomen, Hip, Thigh, Forearm, Wrist** (see code below).

v)

4 variables: **Weight, Abdomen, Forearm, Wrist** (see code below)

vi)

7 variables: **Age, Weight, Neck, Abdomen, Thigh, Forearm, Wrist** (see code below).

Note the need for `nvmax=p` in order to consider all subsets of the variables.

```
par(mfrow=c(2,2),
    oma = rep(1, 4) + 0.1,
    mar = rep(2, 4) + 0.1
)

rs <- regsubsets(bodyfat~., data=dat, nvmax=p)
models <- summary(rs)$which
# models
varnames <- colnames(models[,-1])

##### Adj. R^2

y <- dat$bodyfat
TSS <- sum((y - mean(y))^2)

myadjr2 <- 1 - summary(rs)$rss / TSS / (n - 1:p - 1) * (n - 1)
plot(1:p, myadjr2, type='l',  main="Adj. R^2")
idx.adjr2 <- which.max(myadjr2)
idx.adjr2
```

```
## [1] 9
```

```
# which.max(summary(rs)$adjr2) # check
points(idx.adjr2, myadjr2[idx.adjr2], pch=8)
varnames[models[idx.adjr2,-1]]
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Biceps"
## [8] "Forearm" "Wrist"
```

```
##### AIC and BIC

myaic <- n * log(summary(rs)$rss / n) + n * log(2 * 3.14159 * exp(1)) + 2 * (1 + (1:p))
mybic <- n * log(summary(rs)$rss / n) + n * log(2 * 3.14159 * exp(1)) + (1 + (1:p)) * log(n)


plot(1:p, myaic, type='l',  main="AIC")
idx.aic <- which.min(myaic)
idx.aic
```

```
## [1] 8
```

```
points(idx.aic, myaic[idx.aic], pch=8)
varnames[models[idx.aic,-1]]
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Forearm"
## [8] "Wrist"
```

```
plot(1:p, mybic, type='l',  main="BIC")
idx.bic <- which.min(mybic)
# which.min(summary(rs)$bic) # check
idx.bic
```

```
## [1] 4
```

```
points(idx.bic, mybic[idx.bic], pch=8)
varnames[models[idx.bic,-1]]
```

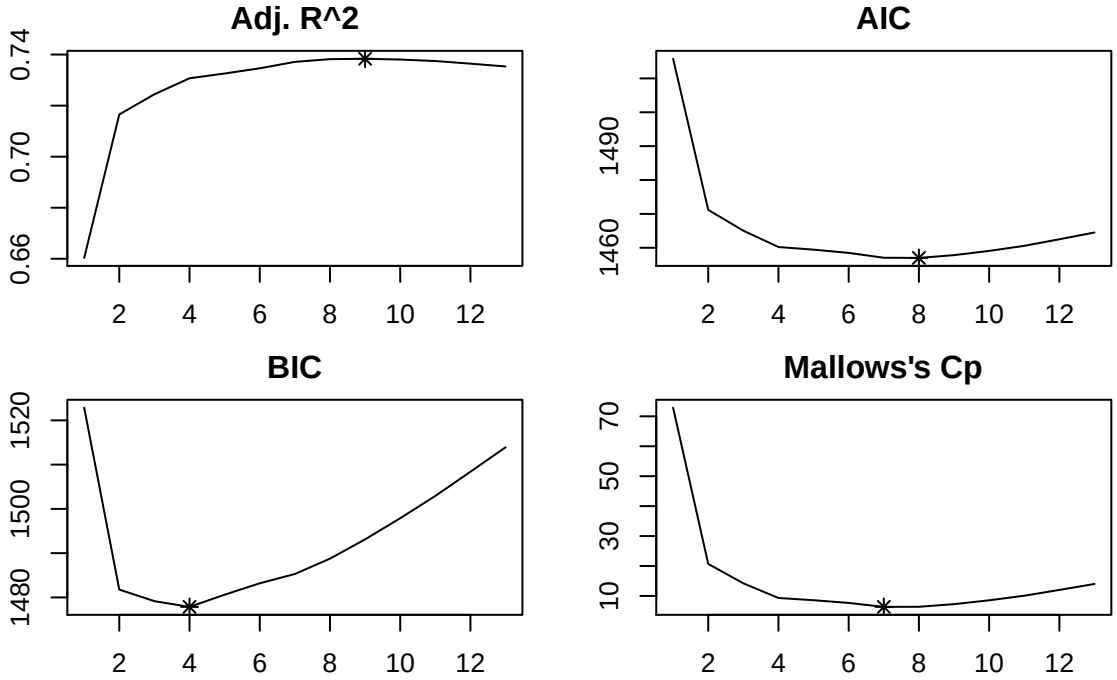
```
## [1] "Weight"  "Abdomen" "Forearm" "Wrist"
```

```
##### Mallows's Cp

sigma.hat <- sigma(lm(bodyfat~., data=dat))
mycp <- summary(rs)$rss / sigma.hat^2 - (n - 2 * ((1:p) + 1))
# summary(rs)$cp # check
plot(1:p, mycp, type='l',  main="Mallows's Cp")
idx.cp <- which.min(mycp)
idx.cp
```

```
## [1] 7
```

```
points(idx.cp, mycp[idx.cp], pch=8)
```




```
varnames[models[idx.cp,-1]]
```

```
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Thigh"   "Forearm" "Wrist"
```

(b)

Based on the above work, we have four models to consider (some methods selected the same model).

1. **Weight, Abdomen, Forearm, Wrist**
2. **Age, Weight, Neck, Abdomen, Thigh, Forearm, Wrist**
3. **Age, Weight, Neck, Abdomen, Hip, Thigh, Forearm, Wrist**
4. **Age, Weight, Neck, Abdomen, Hip, Thigh, Biceps, Forearm, Wrist**

```
vars1 <- varnames[models[idx.bic, -1]]
vars2 <- varnames[models[idx.cp, -1]]
vars3 <- varnames[models[idx.aic, -1]]
vars4 <- varnames[models[idx.adjr2, -1]]
vars.list <- list(vars1, vars2, vars3, vars4)
vars.list
```

```
## [[1]]
## [1] "Weight"  "Abdomen" "Forearm" "Wrist"
##
## [[2]]
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Thigh"   "Forearm" "Wrist"
##
## [[3]]
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Forearm"
## [8] "Wrist"
##
## [[4]]
## [1] "Age"     "Weight"  "Neck"    "Abdomen" "Hip"     "Thigh"   "Biceps"
## [8] "Forearm" "Wrist"
```

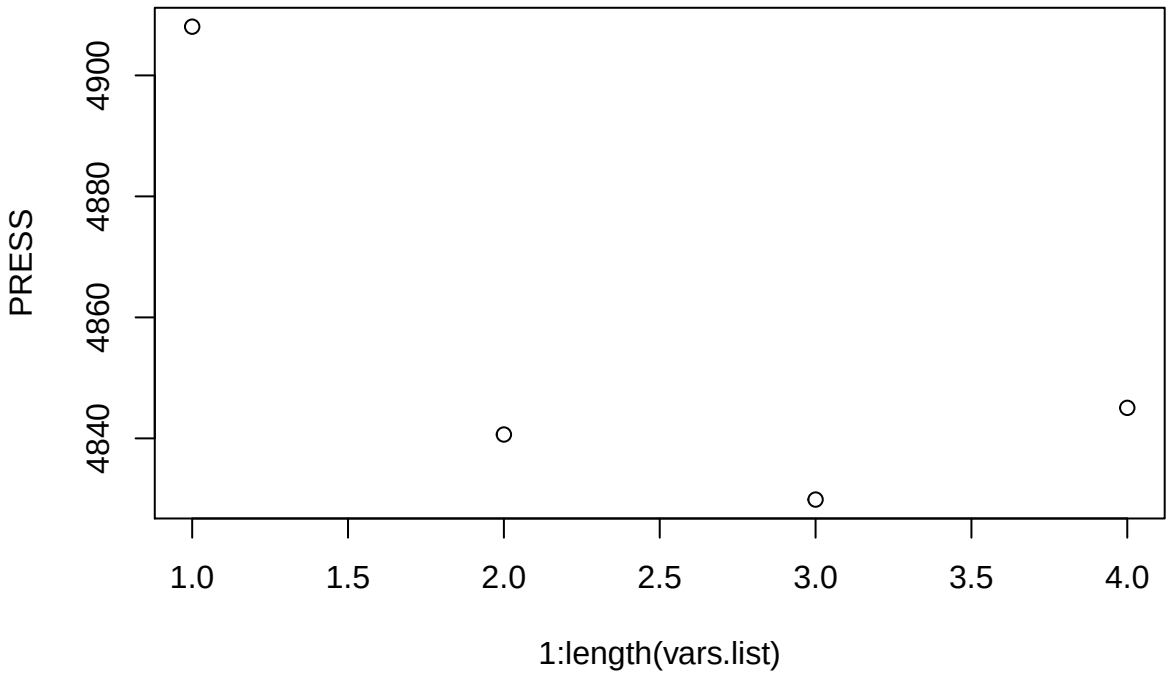
```
# m1 <- lm(bodyfat ~ ., data=dat[, c("bodyfat", vars1)])
# m2 <- lm(bodyfat ~ ., data=dat[, c("bodyfat", vars2)])
# m3 <- lm(bodyfat ~ ., data=dat[, c("bodyfat", vars3)])
# m4 <- lm(bodyfat ~ ., data=dat[, c("bodyfat", vars4)])
```

If you did leave-one-out cross-validation, then the computation will look like the following.

```
PRESS <- rep(0, length(vars.list))
for (i in 1:length(vars.list)) {
  vars <- vars.list[[i]]
  m <- lm(bodyfat~., data=dat[, c("bodyfat", vars)])
  h <- hatvalues(m)
  res <- resid(m)
  PRESS[i] <- sum(res^2 / (1 - h)^2)
}
PRESS
```

```
## [1] 4908.053 4840.639 4829.885 4845.045
```

```
plot(1:length(vars.list), PRESS)
```




```
vars.LOO <- vars.list[[which.min(PRESS)]]
mod.LOO <- lm(bodyfat~., data=dat[, c("bodyfat", vars.LOO)])
mod.LOO
```

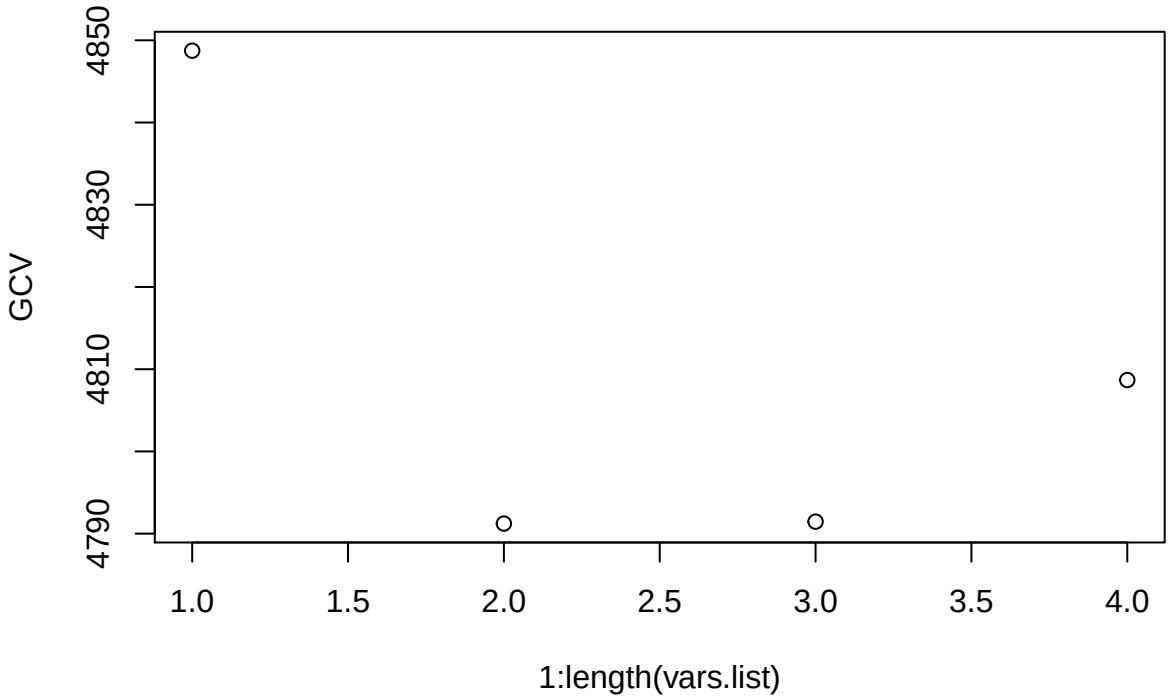
```
##
## Call:
## lm(formula = bodyfat ~ ., data = dat[, c("bodyfat", vars.LOO)])
##
## Coefficients:
## (Intercept)          Age       Weight         Neck      Abdomen
##   -22.65637      0.06578     -0.08985     -0.46656      0.94482
##         Hip        Thigh      Forearm        Wrist
##    -0.19543      0.30239      0.51572     -1.53665
```

If you used generalized cross-validation, the computation will look like the following.

```
GCV <- rep(0, length(vars.list))
for (i in 1:length(vars.list)) {
  vars <- vars.list[[i]]
  m <- lm(bodyfat~., data=dat[, c("bodyfat", vars)])
  RSS <- sum(resid(m)^2)
  GCV[i] <- RSS / (1 - (1 + length(vars)) / n)^2
}
GCV
```

```
## [1] 4848.737 4791.225 4791.460 4808.682
```

```
plot(1:length(vars.list), GCV)
```




```
vars.GCV <- vars.list[[which.min(GCV)]]
mod.GCV <- lm(bodyfat~., data=dat[, c("bodyfat", vars.GCV)])
mod.GCV
```

```
##
## Call:
## lm(formula = bodyfat ~ ., data = dat[, c("bodyfat", vars.GCV)])
##
## Coefficients:
## (Intercept)          Age       Weight         Neck      Abdomen
##   -33.25799      0.06817     -0.11944     -0.40380      0.91788
##       Thigh      Forearm        Wrist
##     0.22196      0.55314     -1.53240
```

If you did k-fold cross-validation, the computation will look like the following. (I chose $k = 10$.)

```
library(caret)
```

```
## Loading required package: lattice
```

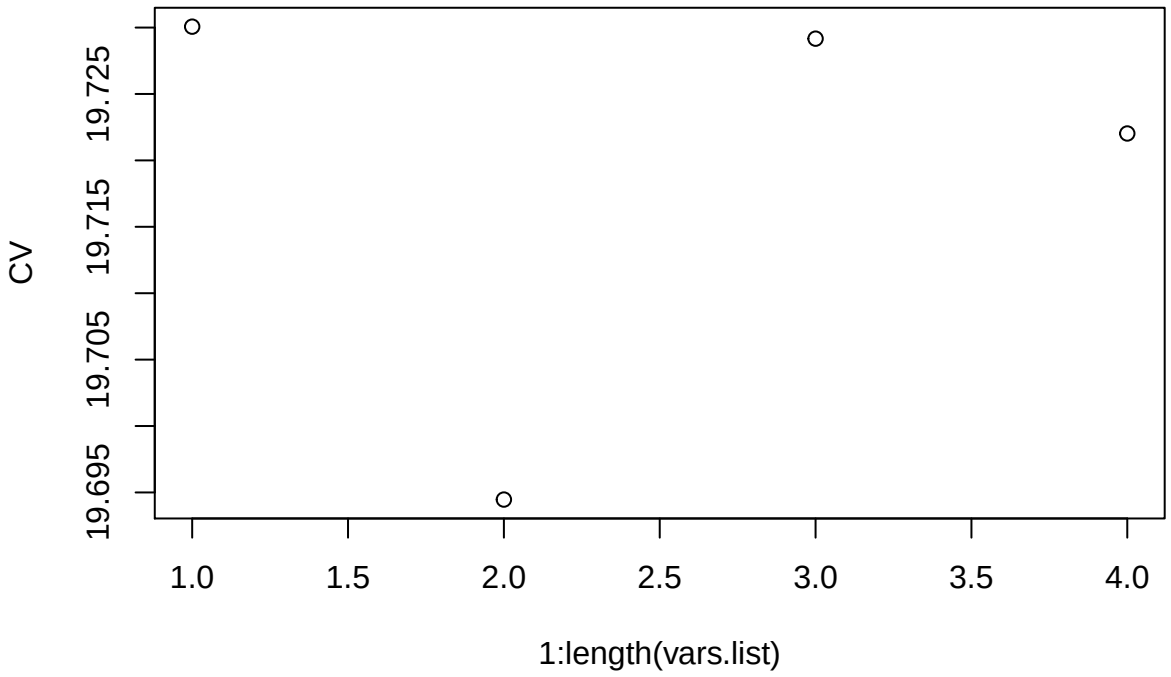
```
## Loading required package: ggplot2
```

```
k <- 10
folds <- createFolds(dat$bodyfat, k=k)
MSE <- matrix(0, length(vars.list), k)
for (i in 1:length(vars.list)) {
  vars <- vars.list[[i]]
  for (j in 1:k) {
    m <- lm(bodyfat~., data=dat[-folds[[j]], c("bodyfat", vars)])
    preds <- predict(m, dat[folds[[j]], vars])
    MSE[i, j] <- 1 / length(folds[[j]]) * sum((preds - dat$bodyfat[folds[[j]]])^2)
  }
}
CV <- apply(MSE, MARGIN=1, FUN=mean)
```

```
CV
```

```
## [1] 19.73007 19.69446 19.72917 19.72203
```

```
plot(1:length(vars.list), CV)
```




```
vars.kfold <- vars.list[[which.min(CV)]]
mod.kfold <- lm(bodyfat~., data=dat[, c("bodyfat", vars.kfold)])
mod.kfold
```

```
##
## Call:
## lm(formula = bodyfat ~ ., data = dat[, c("bodyfat", vars.kfold)])
##
## Coefficients:
## (Intercept)          Age       Weight         Neck      Abdomen
##   -33.25799      0.06817     -0.11944     -0.40380      0.91788
##       Thigh      Forearm        Wrist
##     0.22196      0.55314     -1.53240
```

- Leave-one-out chooses the model with eight variables: **Age, Weight, Neck, Abdomen, Hip, Thigh, Forearm, Wrist**.
- GCV chooses the model with seven variables: **Age, Weight, Neck, Abdomen, Thigh, Forearm, Wrist**
- The result of cross-validation will depend on the randomness in the fold selection.

(c)

Omitted.