hw8_yw3204

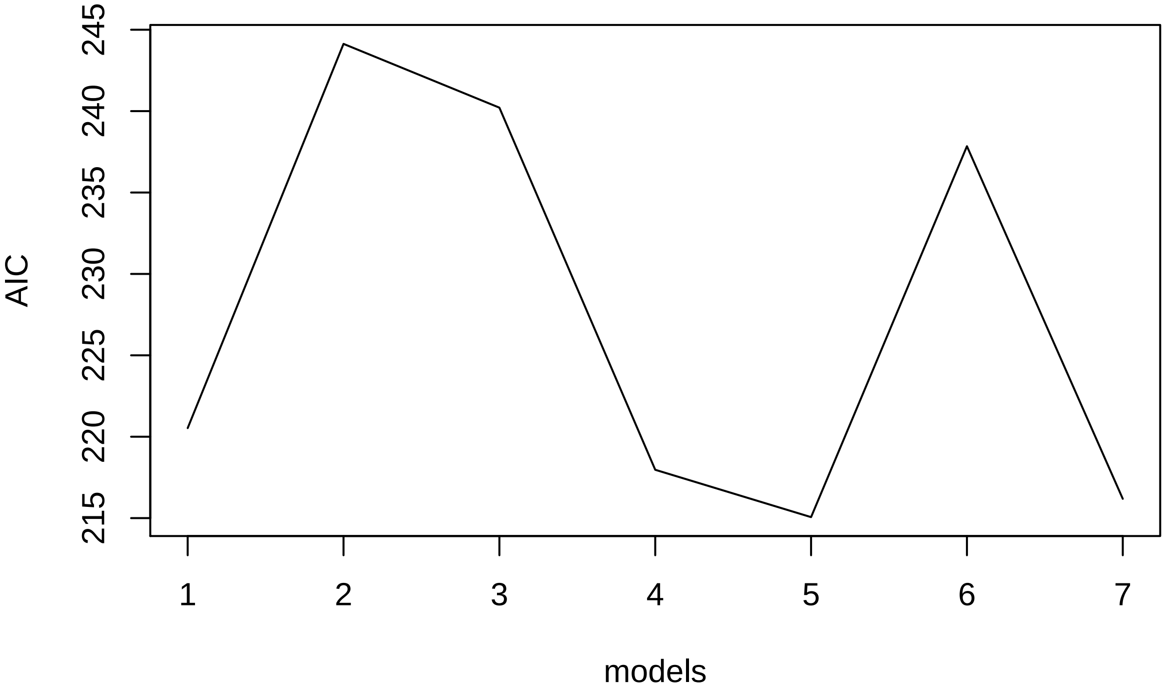
Yuhao Wang, yw3204

11/18/2018

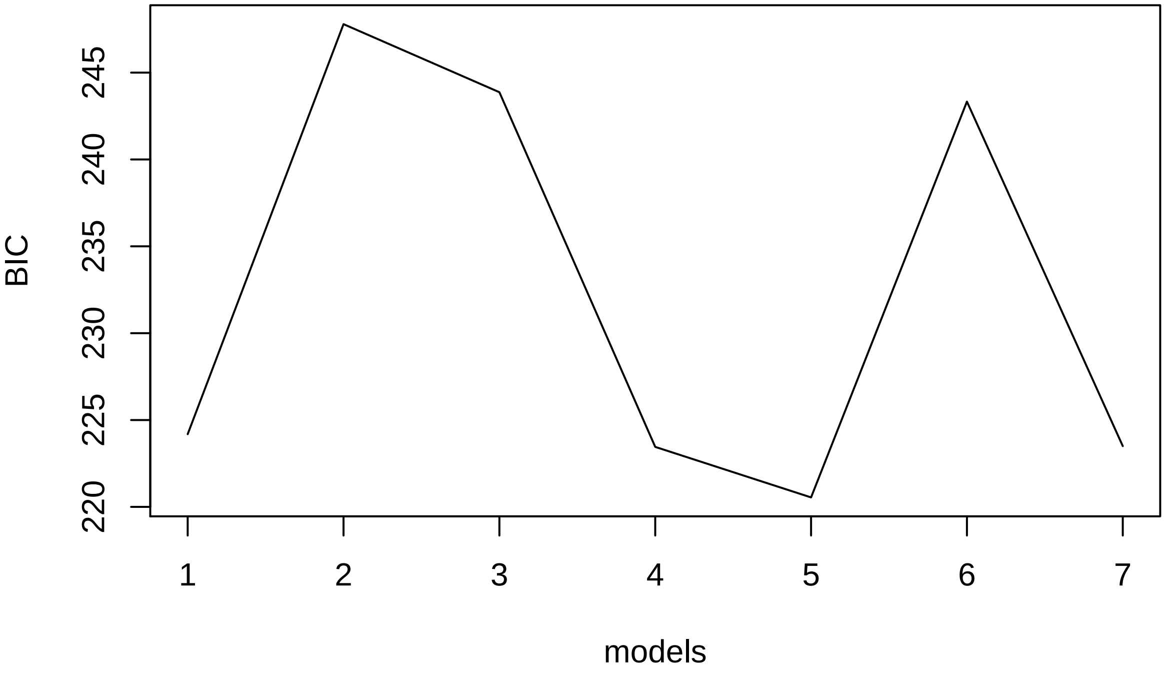
9.9

a.

```
patient <- read.table("CH06PR15.txt")
names(patient) <- c("Y", "X1", "X2", "X3")
n_patient <- nrow(patient)

models <- list()
models[[1]] <- lm(Y~X1, patient)
models[[2]] <- lm(Y~X2, patient)
models[[3]] <- lm(Y~X3, patient)
models[[4]] <- lm(Y~X1+X2, patient)
models[[5]] <- lm(Y~X1+X3, patient)
models[[6]] <- lm(Y~X2+X3, patient)
models[[7]] <- lm(Y~X1+X2+X3, patient)

# AIC
AICs <- c()
for(i in c(1:7)) {
  err_mle <- sum(models[[i]]$residuals^2) / n_patient
  AICs[i] <- n_patient * log(err_mle) + 2 * (floor((i-1) / 3) + 2)
}

plot(c(1:7), AICs, xlab = "models", ylab = "AIC", type = "l")
```

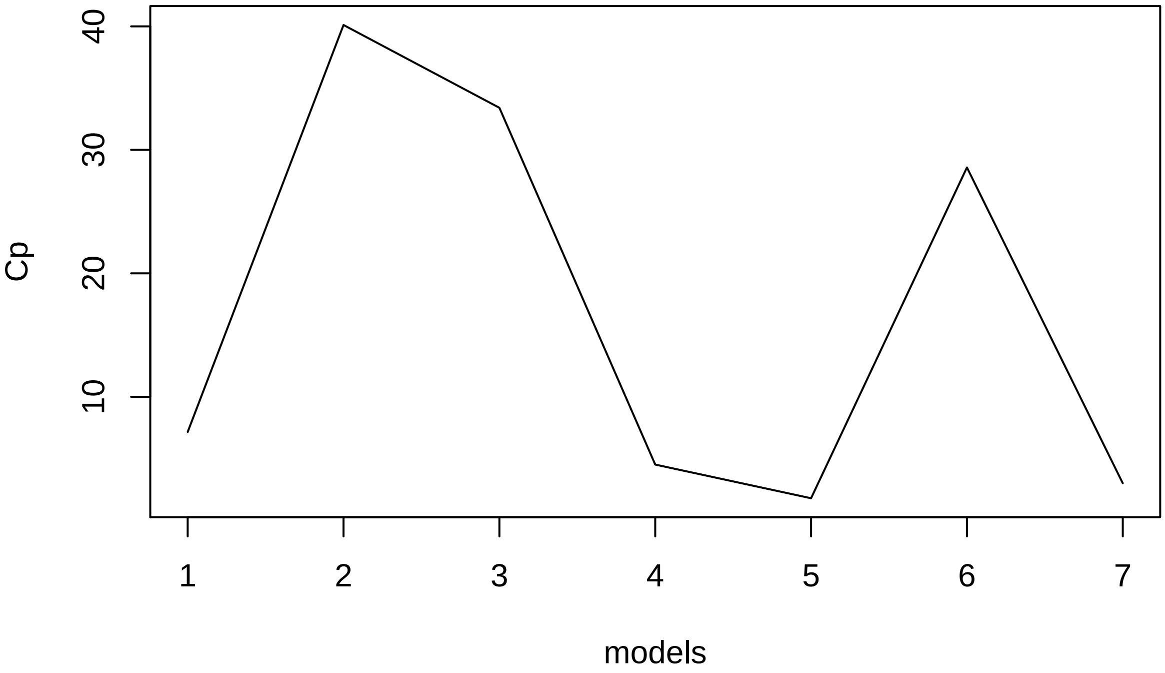


```r
# BIC
BICs <- c()
for(i in c(1:7)) {
  err_mle <- sum(models[[i]]$residuals^2) / n_patient
  BICs[i] <- n_patient * log(err_mle) + log(n_patient) * (floor((i-1) / 3) + 2)
}

plot(c(1:7), BICs, xlab = "models", ylab = "BIC", type = "l")
```



```r
# Cp
Cps <- c()
```

```
err_full <- sum(models[[7]]$residuals^2) / (n_patient-5)
for(i in c(1:7)) {
  sse <- sum(models[[i]]$residuals^2)
  Cps[i] <- sse / err_full - n_patient + 2 * (floor((i-1) / 3) + 2)
}

plot(c(1:7), Cps, xlab = "models", ylab = "Cp", type = "l")
```



According to the three plots above, all three critiria suggest model 5 as the best, which corresponds to the set of X1 and X3 variables.

b.

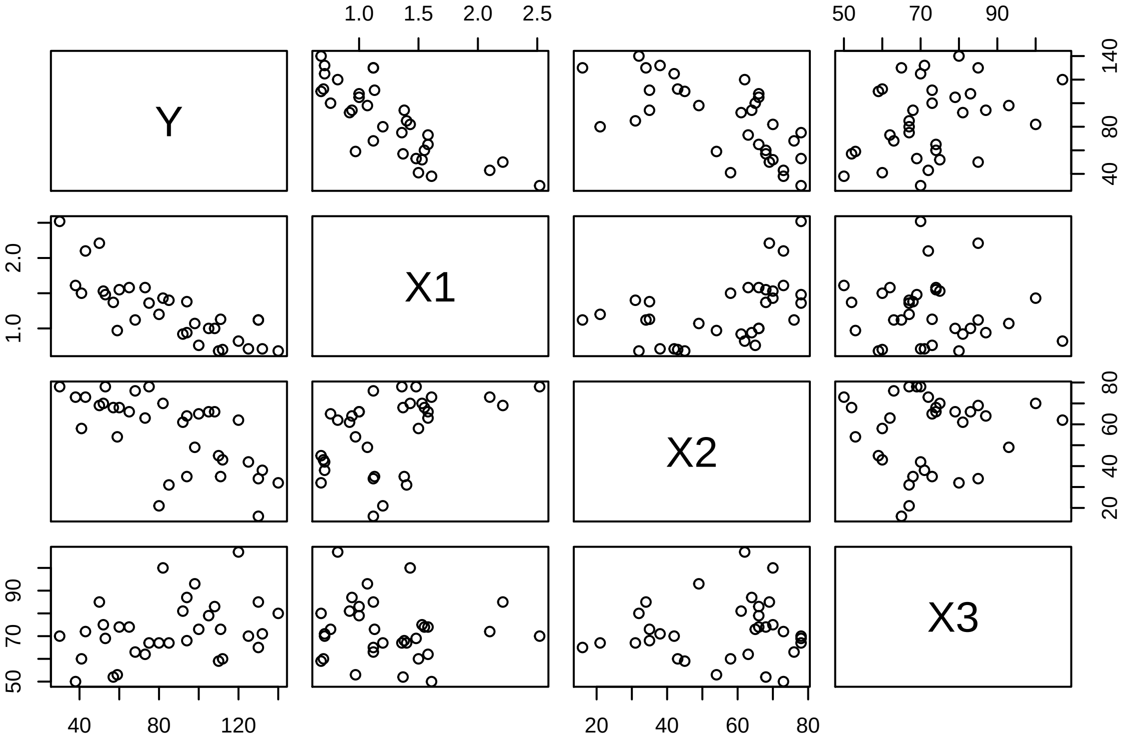
Luckily, the three criteria in part (a) suggest the same best set which doesn's always holds. But for AIC and Cp, we can show that they are basically equivalent. That's to say, under linear regression, AIC and Cp always suggest the same best set.

c.

No. Due to the limiting number of all variables here, forward stepwise regression and all-possible-regressions take almost the same steps to reach the optimal set.

9.15

```
kidney <- read.table("CH09PR15.txt")
names(kidney) <- c("Y", "X1", "X2", "X3")

pairs(kidney[, c(1:4)])
```



```
cor(kidney[, c(2:4)])
```

```
##              X1          X2          X3
## X1  1.00000000 0.46773179 -0.08898262
## X2  0.46773179 1.00000000  0.06848147
## X3 -0.08898262 0.06848147  1.00000000
```

The scatter plot matrix suggests that the response Y has negative linear relationship with X1 and x2 and a seemlingy positive linear relationship with X3. According to the correlation matrix, the correlation between X1 and X2 is 0.47 and thus they have a severe multicollinearity.

c.

```
lm_first <- lm(Y~X1+X2+X3, data = kidney)
summary(lm_first)
```

```
## 
## Call:
## lm(formula = Y ~ X1 + X2 + X3, data = kidney)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -28.668  -7.002   1.518   9.905  16.006 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 120.0473    14.7737   8.126 5.84e-09 ***
## X1          -39.9393     5.6000  -7.132 7.55e-08 ***
## X2           -0.7368     0.1414  -5.211 1.41e-05 ***
## X3            0.7764     0.1719   4.517 9.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.46 on 29 degrees of freedom
## Multiple R-squared:  0.8548, Adjusted R-squared:  0.8398
## F-statistic: 56.92 on 3 and 29 DF,  p-value: 2.885e-12
```

According to the result, the P-values of all three coefficients are quite close to 0 and thus are all significant. We then believe they should all be retained.

9.16

a.

```
kidney_centered <- apply(kidney[, c(2:4)], 2, function(x) x-mean(x))
sqre <- as.data.frame(kidney_centered^2)
names(sqre) <- c("X1_sqre", "X2_sqre", "X3_sqre")
kidney_centered <- cbind(kidney_centered, sqre)
kidney_centered <- cbind(kidney_centered, X12 = kidney_centered$X1 * kidney_centered$X2)
kidney_centered <- cbind(kidney_centered, X13 = kidney_centered$X1 * kidney_centered$X3)
kidney_centered <- cbind(kidney_centered, X23 = kidney_centered$X2 * kidney_centered$X3)
kidney_centered <- as.data.frame(cbind(Y = kidney$Y, kidney_centered))

library("leaps")
# using forward selection
fwd_slt <- regsubsets(Y~., data = kidney_centered, nvmax = 9, method = "forward")
fwd_slt_res <- summary(fwd_slt)
fwd_slt_res
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = kidney_centered, nvmax = 9,
##     method = "forward")
## 9 Variables  (and intercept)
##         Forced in Forced out
## X1          FALSE      FALSE
## X2          FALSE      FALSE
## X3          FALSE      FALSE
## X1_sqre     FALSE      FALSE
## X2_sqre     FALSE      FALSE
## X3_sqre     FALSE      FALSE
## X12         FALSE      FALSE
## X13         FALSE      FALSE
## X23         FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: forward
##          X1  X2  X3  X1_sqre X2_sqre X3_sqre X12 X13 X23
## 1  ( 1 ) "*" " " " " " "     " "     " "     " " " " " "
## 2  ( 1 ) "*" "*" " " " "     " "     " "     " " " " " "
## 3  ( 1 ) "*" "*" "*" " "     " "     " "     " " " " " "
## 4  ( 1 ) "*" "*" "*" " "     " "     " "     "*" " " " "
## 5  ( 1 ) "*" "*" "*" " "     " "     "*"     "*" " " " "
## 6  ( 1 ) "*" "*" "*" " "     "*"     "*"     "*" " " " "
## 7  ( 1 ) "*" "*" "*" " "     "*"     "*"     "*" " " "*"
## 8  ( 1 ) "*" "*" "*" "*"     "*"     "*"     "*" " " "*"
## 9  ( 1 ) "*" "*" "*" "*"     "*"     "*"     "*" "*" "*"
```

```
fwd_slt_res$cp
```

```
## [1] 48.509656 26.683707  6.512138  3.302215  3.384990  4.766392  6.356750
## [8]  8.002543 10.000000
```

Based on the above result, the best three models are model 4, 5 and 6, which correspond to predictors set {X1, X2, X3, X12}, {X1, X2, X3, X3_sqre, X12} and {X1, X2, X3, X2_sqre, X3_sqre, X12}. And their C_p are 3.30, 3.38 and 4.77.

b.

Their C_p differ slightly.

9.19

a.

```
# select step: check whether p-value is less than 0.1, if yes, chooes the smallest
# select function
# cur_var: index vector which indicates the current added variables
# tba: index vector which indicates the candidate variables
# return a list containing a t-statistics vector and a corresponding p-value vector
select <- function(cur_var, tba) {
  n <- nrow(kidney_centered)
  t <- c()
  p <- c()

  # this correspond to the first select step
  if(length(cur_var) == 0) {
    for(i in tba) {
      lm <- lm(Y~kidney_centered[, i], kidney_centered)
      nume <- sum(lm$residuals^2) / (n-2)
      deno <- sum((kidney_centered[, i] - mean(kidney_centered[, i]))^2)
      s_b <- sqrt(nume/deno)
      b <- as.numeric(lm$coefficients[2])
      t <- c(t, b/s_b)
      p <- c(p, 2*(1-pt(abs(b/s_b), n-2)))
    }
  }

  else {
    lm_cur <- lm(Y~., kidney_centered[, c(1, cur_var)])
    for(i in c(tba)) {
      lm_added <- lm(Y~., kidney_centered[, c(1, cur_var, i)])
      d_sse <- sum(lm_cur$residuals^2) - sum(lm_added$residuals^2)
      sse <- sum(lm_added$residuals^2)
      t <- c(t, sqrt(d_sse / sse * (n-length(cur_var)-2)))
      p <- c(p, 2*(1-pt(abs(sqrt(d_sse/sse*(n-length(cur_var)-2))), n-length(cur_var)-2)))
    }
  }
```

```r
  return(list(t = t, p = p))
}


# drop step: check whether p-value is greater than 0.15 or not, if yes, drop
# drop function
# before: index vector which indicates the variables before select
# added: index indicates the added variable, a scalar
# return a list containing a t-statistics vector and a corresponding p-value vector
drop <- function(before, added) {
  n <- nrow(kidney_centered)
  t <- c()
  p <- c()

  lm_added <- lm(Y~., kidney_centered[, c(1, before, added)])

  for(i in before) {
    lm_drop <- lm(Y~., kidney_centered[, c(1, before[before != i], added)])
    d_sse <- sum(lm_drop$residuals^2) - sum(lm_added$residuals^2)
    sse <- sum(lm_added$residuals^2)

    t <- c(t, sqrt(d_sse / sse * (n-length(before)-1-1)))
    p <- c(p, 2*(1-pt(abs(sqrt(d_sse / sse * (n-length(before)-1-1))), n-length(before)-1-1)))
  }

  return(list(t = t, p = p))
}

# select step
# according to the result select X1
select(c(), c(2:10))
```

```
## $t
## [1] -7.4706477 -4.9962072  2.0526937 -1.7723278  1.1349730  0.1953459
## [7] -1.2784248 -0.3540848  1.4141439
##
## $p
## [1] 2.041142e-08 2.171255e-05 4.862554e-02 8.616912e-02 2.650864e-01
## [6] 8.463969e-01 2.105887e-01 7.256725e-01 1.672849e-01
```

```r
# select step
# selelct X2
select(c(2), c(3:10))
```

```
## $t
## [1] 3.64892123 2.84772337 1.59024816 2.51235424 0.62610927 1.96975363
## [7] 0.07162157 1.38934292
##
## $p
## [1] 0.000992071 0.007874614 0.122263227 0.017602180 0.535978385 0.058160425
## [7] 0.943378427 0.174955248
```

```r
# drop step
# retaine X1
drop(c(2), 3)
```

```
## $t
## [1] 6.098471
##
## $p
## [1] 1.059128e-06
```

```
# select step
# select X3
select(c(2, 3), c(4:10))
```

```
## $t
## [1] 4.5170851 1.9001410 0.3390887 0.1950709 1.5661367 0.6811171 0.7653587
##
## $p
## [1] 9.685152e-05 6.739860e-02 7.369862e-01 8.466977e-01 1.281648e-01
## [6] 5.012005e-01 4.502423e-01
```

```
# drop step
# retain X1 and X2
drop(c(2, 3), 4)
```

```
## $t
## [1] 7.132081 5.210738
##
## $p
## [1] 7.546924e-08 1.412091e-05
```

```
# select step
# stop adding
select(c(2, 3, 4), c(5:10))
```

```
## $t
## [1] 1.9806806 0.9831879 1.6020640 2.3550413 0.1170398 0.8452339
##
## $p
## [1] 0.05752798 0.33393484 0.12036460 0.02576128 0.90766402 0.40514528
```

Thus, the final model contains only X_1, X_2 and X_3.