```r
## Coffee example  (Coffee Quality Institute, 2018) continued
coffee <- read.csv("coffee_arabica.csv")

mfull <- lm(Flavor~ factor(Processing.Method) + Aroma + Aftertaste +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture, dat=coffee)
summary(mfull)$adj.r.squared
AIC(mfull)
BIC(mfull)

library(leaps)
all_regs <- regsubsets(Flavor ~ ., data = coffee, nvmax = 10, nbest = 2^10,
                       really.big = TRUE)
all_regs_summ <- summary(all_regs)
# all_regs_summ$which
# all_regs_summ$adjr2
# all_regs_summ$bic

# Organize results according to number of variables in model
p <- 10
k <- c(rep(1, choose(p,1)),
           rep(2, choose(p,2)),
           rep(3, choose(p,3)),
           rep(4, choose(p,4)),
           rep(5, choose(p,5)),
           rep(6, choose(p,6)),
           rep(7, choose(p,7)),
           rep(8, choose(p,8)),
           rep(9, choose(p,9)),
           rep(10, choose(p,10)))
boxplot(all_regs_summ$adjr2 ~ k, xlab = "Number of predictors", ylab =
          expression(R[adj]^2), ylim = c(0,1))
abline(h = c(0,1), lty = 2, col = "red")
```

```r
boxplot(all_regs_summ$bic ~ k, xlab = "Number of predictors", ylab = "BIC")
```

```r
max(all_regs_summ$adjr2)
bestR2adj <- which.max(all_regs_summ$adjr2)
min(all_regs_summ$bic)
bestBIC <- which.min(all_regs_summ$bic)

# Find out which predictors in those models
all_regs_summ$which[bestR2adj,]
all_regs_summ$which[bestBIC,]


coffee$wet <- ifelse(coffee$Processing.Method == 'Washed / Wet', 1,
                     0) # 1 = wet, 0 otherwise
coffee$semi <- ifelse(coffee$Processing.Method == 'Semi-washed / Semi-pulped',
                      1, 0) # 1 = semi/dry, 0 otherwise
coffee$Processing.Method <- NULL

m_bestr2adj <- lm(Flavor~ wet + Aroma + Aftertaste +
            Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
            dat=coffee)
```

```r
summary(m_bestr2adj)
AIC(m_bestr2adj)
BIC(m_bestr2adj)

m_bestBIC <- lm(Flavor~ wet + Aroma + Aftertaste +
                  Body + Acidity + Sweetness , dat=coffee)
summary(m_bestBIC)
AIC(m_bestBIC)
BIC(m_bestBIC)

# Let's also try stepwise methods
library(MASS)

# Full model and empty model with just intercept
full <- lm(Flavor ~ ., data = coffee)
empty <- lm(Flavor ~ 1, data = coffee)

# default stepAIC uses AIC criterion
stepAIC(object = empty, scope = list(upper = full, lower = empty), direction
        = "forward")

# Let's get stepAIC to use BIC by specifying the penalty k = log(n)
# Forward
stepAIC(object = empty, scope = list(upper = full, lower = empty), direction
        = "forward", k = log(nrow(coffee)))
m_f <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
               direction = "forward", trace = 0, k = log(nrow(coffee)))
summary(m_f)

# Backward
stepAIC(object = full, scope = list(upper = full, lower = empty),
        direction = "backward", k = log(nrow(coffee)))
m_b <- stepAIC(object = full, scope = list(upper = full, lower = empty),
               direction = "backward", trace = 0, k = log(nrow(coffee)))
summary(m_b)

# Forward-backward
stepAIC(object = empty, scope = list(upper = full, lower = empty),
        direction = "both", k = log(nrow(coffee)))
m_h <- stepAIC(object = empty, scope = list(upper = full, lower = empty),
               direction = "both", trace = 0, k = log(nrow(coffee)))
summary(m_h)

# 10 variables is still a fairly small problem:  in this example
# all 3 approaches identify the same BIC-based model as the exhaustive search.
```