Stat 495 - Chapter 16 - PreLasso Examples

A.S. Wagaman

Techniques for Choosing Predictors

Chapter 16 begins with a review of forward stepwise selection. Forward selection, backward elimination, and stepwise regression were covered in Stat 230, but we can review how to run these procedures in R.

The HELPrct data set contains 453 observations on 27 variables. The data is the data set on Health Evaluation and Linkage to Primary Care study results. The HELP study was a clinical trial for adult inpatients recruited from a detoxification unit. Patients with no primary care physician were randomized to receive a multidisciplinary assessment and a brief motivational intervention or usual care, with the goal of linking them to primary medical care.

```
data(HELPrct)
names(HELPrct)
```

```
##  [1] "age"                "anysubstatus"     "anysub"           "cesd"
##  [5] "d1"                 "daysanysub"       "dayslink"         "drugrisk"
##  [9] "e2b"                "female"           "sex"              "g1b"
## [13] "homeless"           "i1"               "i2"               "id"
## [17] "indtot"             "linkstatus"       "link"             "mcs"
## [21] "pcs"                "pss_fr"           "racegrp"          "satreat"
## [25] "sexrisk"            "substance"        "treat"            "avg_drinks"
## [29] "max_drinks"         "hospitalizations"
```

```
#help(HELPrct)
str(HELPrct)
```

```
## 'data.frame':    453 obs. of  30 variables:
##  $ age             : int  37 37 26 39 32 47 49 28 50 39 ...
##   ..- attr(*, "label")= chr "age (years)"
##  $ anysubstatus    : int  1 1 1 1 1 1 NA 1 1 1 ...
##  $ anysub          : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 NA 2 2 2 ...
##   ..- attr(*, "label")= chr "post-detox substance use"
##  $ cesd            : int  49 30 39 15 39 6 52 32 50 46 ...
##   ..- attr(*, "label")= chr "CESD at baseline"
##  $ d1              : int  3 22 0 2 12 1 14 1 14 4 ...
##   ..- attr(*, "label")= chr "hospitalizations"
##  $ daysanysub      : int  177 2 3 189 2 31 NA 47 31 115 ...
##   ..- attr(*, "label")= chr "days to 1st post-detox substance use"
##  $ dayslink        : int  225 NA 365 343 57 365 334 365 365 382 ...
##   ..- attr(*, "label")= chr "time to linkage to primary care (days)"
##  $ drugrisk        : int  0 0 20 0 0 0 0 7 18 20 ...
##   ..- attr(*, "label")= chr "Risk Assessment Battery at baseline"
##  $ e2b             : int  NA NA NA 1 1 NA 1 8 7 3 ...
##   ..- attr(*, "label")= chr "previous detox stays (last 6 months)"
##  $ female          : int  0 0 0 1 0 1 1 0 1 0 ...
##   ..- attr(*, "label")= chr "female"
##  $ sex             : Factor w/ 2 levels "female","male": 2 2 2 1 2 1 1 2 1 2 ...
##   ..- attr(*, "label")= chr "sex"
##  $ g1b             : Factor w/ 2 levels "no","yes": 2 2 1 1 1 1 2 2 1 1 ...
##   ..- attr(*, "label")= chr "serious suicidal thoughts (last 30 days)"
##  $ homeless        : Factor w/ 2 levels "homeless","housed": 2 1 2 2 1 2 2 1 1 1 ...
##   ..- attr(*, "label")= chr "housing status"
##  $ i1              : int  13 56 0 5 10 4 13 12 71 20 ...
##   ..- attr(*, "label")= chr "avg. drinks per day"
##  $ i2              : int  26 62 0 5 13 4 20 24 129 27 ...
##   ..- attr(*, "label")= chr "max. drinks per day"
##  $ id              : int  1 2 3 4 5 6 7 8 9 10 ...
##   ..- attr(*, "label")= chr "subject ID"
##  $ indtot          : int  39 43 41 28 38 29 38 44 44 44 ...
##   ..- attr(*, "label")= chr "Inventory of Drug Use Consequences"
##  $ linkstatus      : int  1 NA 0 0 1 0 0 0 0 0 ...
##   ..- attr(*, "label")= chr "post-detox linkage to primary care"
##  $ link            : Factor w/ 2 levels "no","yes": 2 NA 1 1 2 1 1 1 1 1 ...
##   ..- attr(*, "label")= chr "post-detox linkage to primary care"
##  $ mcs             : num  25.11 26.67 6.76 43.97 21.68 ...
##   ..- attr(*, "label")= chr "SF-36 Mental Component Score"
##  $ pcs             : num  58.4 36 74.8 61.9 37.3 ...
##   ..- attr(*, "label")= chr "SF-36 Physical Component Score"
##  $ pss_fr          : int  0 1 13 11 10 5 1 4 5 0 ...
##   ..- attr(*, "label")= chr "perceived social support by friends"
##  $ racegrp         : Factor w/ 4 levels "black","hispanic",..: 1 4 1 4 1 1 1 4 4 4 ...
##   ..- attr(*, "label")= chr "race/ethnicity"
##  $ satreat         : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 2 2 1 2 ...
##   ..- attr(*, "label")= chr "previous BSAS substance abuse treatment"
##  $ sexrisk         : int  4 7 2 4 6 5 8 6 8 0 ...
##   ..- attr(*, "label")= chr "Risk Assessment Battery sex risk score"
##  $ substance       : Factor w/ 3 levels "alcohol","cocaine",..: 2 1 3 3 2 2 2 1 1 3 ...
##   ..- attr(*, "label")= chr "primary substance of abuse"
##  $ treat           : Factor w/ 2 levels "no","yes": 2 2 1 1 1 2 1 2 1 2 ...
##   ..- attr(*, "label")= chr "randomized to HELP clinic"
##  $ avg_drinks      : int  13 56 0 5 10 4 13 12 71 20 ...
##   ..- attr(*, "label")= chr "avg. drinks per day"
##  $ max_drinks      : int  26 62 0 5 13 4 20 24 129 27 ...
##   ..- attr(*, "label")= chr "max. drinks per day"
##  $ hospitalizations: int  3 22 0 2 12 1 14 1 14 4 ...
```

```
HELPrct <- with(HELPrct, HELPrct[ !is.na(drugrisk), ])
dim(HELPrct)
```

```
## [1] 452  30
```

We eliminate the ONE data point with missing values.

Now, suppose we want to predict a baseline measure of depression (cesd) using other baseline variables including age, number of previous hospitalizations (d1), drug risk, average and maximum number of drinks in a day (last 30 days)(i1 and i2), inventory of drug use score (indtot), mental and physical component scores (mcs and pcs), perceived social support (pss_fr), and sex risk score.

Let's fit a model with all 10 predictors to start.

```
modall <- lm(cesd ~ age + d1 + drugrisk + i1 + i2 + indtot +
               mcs + pcs + pss_fr + sexrisk, data = HELPrct)
msummary(modall)
```

```
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  64.9575     4.3436   14.95   <2e-16 ***
## age          -0.0411     0.0565   -0.73    0.467    
## d1           -0.0820     0.0702   -1.17    0.244    
## drugrisk      0.0467     0.0993    0.47    0.638    
## i1            0.0408     0.0397    1.03    0.304    
## i2            0.0128     0.0282    0.45    0.649    
## indtot        0.0697     0.0648    1.08    0.283    
## mcs          -0.6171     0.0352  -17.55   <2e-16 ***
## pcs          -0.2396     0.0410   -5.84    1e-08 ***
## pss_fr       -0.2275     0.1057   -2.15    0.032 *  
## sexrisk      -0.3085     0.1492   -2.07    0.039 *  
## 
## Residual standard error: 8.66 on 441 degrees of freedom
## Multiple R-squared:  0.532,  Adjusted R-squared:  0.521 
## F-statistic: 50.1 on 10 and 441 DF,  p-value: <2e-16
```

```
car::vif(modall)
```

```
##      age       d1 drugrisk       i1       i2   indtot      mcs      pcs 
##  1.13368  1.13800  1.11399  3.79328  3.76596  1.28994  1.22316  1.17986 
##   pss_fr  sexrisk 
##  1.06699  1.05145 
```

What variables appear most significant in predicting cesd? How well does the model fit? Do we observe any issues with multicollinearity that might make interpreting coefficients difficult?

Now, we want to try to use variable selection techniques to come up with a good set of predictors for predicting cesd.

At the top of this file, note we loaded a new library: *leaps*, which is needed for these functions. You should check that it is installed/loaded if trying to work with these commands.

1. Best Subsets and Mallow's Cp Code

First provide a full model (response ~ all possible explanatory predictors). Then we tell it to run using the best subsets method and plot the results.

```
best <- regsubsets(cesd ~ age + d1 + drugrisk + i1 + i2 + indtot +
               mcs + pcs + pss_fr + sexrisk, data = HELPrct, nbest = 1)
with(summary(best), data.frame(rsq, adjr2, cp, rss, outmat))
```

```
##                rsq    adjr2       cp     rss age d1 drugrisk i1 i2 indtot mcs
## 1  ( 1 ) 0.464077 0.462886 56.79474 37870.4                                *
## 2  ( 1 ) 0.512002 0.509828 13.65368 34483.8                                *
## 3  ( 1 ) 0.518003 0.514775 10.00083 34059.8                  *             *
## 4  ( 1 ) 0.523356 0.519091  6.95863 33681.5                  *             *
## 5  ( 1 ) 0.527413 0.522115  5.13689 33394.8                  *             *
## 6  ( 1 ) 0.529268 0.522921  5.38979 33263.7      *           *             *
## 7  ( 1 ) 0.530761 0.523363  5.98401 33158.2      *           *         *   *
## 8  ( 1 ) 0.531377 0.522914  7.40393 33114.7   *  *           *         *   *
##          pcs pss_fr sexrisk
## 1  ( 1 )                   
## 2  ( 1 )   *               
## 3  ( 1 )   *               
## 4  ( 1 )   *      *        
## 5  ( 1 )   *      *       *
## 6  ( 1 )   *      *       *
## 7  ( 1 )   *      *       *
## 8  ( 1 )   *      *       *
```

We can examine lots of properties about the best subsets solutions in the output above. Here, we want to examine the model with the minimum Cp value, which we can see is size 5.

We can then fit the best subset model, using only those variables marked with an asterisk (i.e. "True").

```
Cpmod <- lm(cesd ~ i1 + mcs + pcs + pss_fr + sexrisk, data = HELPrct)
msummary(Cpmod)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.0192     2.3428   28.18  < 2e-16 ***
## i1            0.0505     0.0209    2.42    0.016 *
## mcs          -0.6324     0.0324  -19.51  < 2e-16 ***
## pcs          -0.2291     0.0388   -5.91  6.9e-09 ***
## pss_fr       -0.2526     0.1040   -2.43    0.016 *
## sexrisk      -0.2887     0.1476   -1.96    0.051 .
##
## Residual standard error: 8.65 on 446 degrees of freedom
## Multiple R-squared:  0.527,  Adjusted R-squared:  0.522
## F-statistic: 99.5 on 5 and 446 DF,  p-value: <2e-16
```

How does this solution compare to the model with all 10 predictors?

If you wanted the AIC for the model, you could attain it with:

```
AIC(Cpmod)
```

```
## [1] 3241.44
```

Remember that AIC is another criterion that could be used, like Cp, to pick models. What are other criteria to consider?

2. Backward elimination

Recall that backward elimination starts from the full model. The process is achieved in R as follows, with a simple summary table provided. Here, you can set the maximum size of subsets to examine with the nvmax option. The default is 8, which is fine for this example. If you have a very large set of predictors, you may need to adjust that value.

```
backward <- regsubsets(cesd ~ age + d1 + drugrisk + i1 + i2 + indtot +
               mcs + pcs + pss_fr + sexrisk, data = HELPrct, method = "backward", nbest = 1)
with(summary(backward), data.frame(cp, outmat))
```

```
##                 cp age d1 drugrisk i1 i2 indtot mcs pcs pss_fr sexrisk
## 1  ( 1 ) 56.79474                                *
## 2  ( 1 ) 13.65368                                *   *
## 3  ( 1 ) 10.00083                  *             *   *
## 4  ( 1 )  6.95863                  *             *   *      *
## 5  ( 1 )  5.13689                  *             *   *      *       *
## 6  ( 1 )  5.38979      *           *             *   *      *       *
## 7  ( 1 )  5.98401      *           *         *   *   *      *       *
## 8  ( 1 )  7.40393   *  *           *         *   *   *      *       *
```

In this particular case, the final model is the same one achieved via minimizing the Cp from the best subset model. This does NOT always occur.

3. Forward selection

For forward selection, we start from a model with just an intercept and build up a model one predictor at a time.

```
forward <- regsubsets(cesd ~ age + d1 + drugrisk + i1 + i2 + indtot +
               mcs + pcs + pss_fr + sexrisk, data = HELPrct, method = "forward", nbest = 1)
with(summary(forward), data.frame(cp, outmat))
```

```
##                 cp age d1 drugrisk i1 i2 indtot mcs pcs pss_fr sexrisk
## 1  ( 1 ) 56.79474                                *                    
## 2  ( 1 ) 13.65368                                *   *                
## 3  ( 1 ) 10.00083                  *             *   *                
## 4  ( 1 )  6.95863                  *             *   *      *         
## 5  ( 1 )  5.13689                  *             *   *      *       * 
## 6  ( 1 )  5.38979      *           *             *   *      *       * 
## 7  ( 1 )  5.98401      *           *         *   *   *      *       * 
## 8  ( 1 )  7.40393   *  *           *         *   *   *      *       *
```

This method also obtains the same model, and the same ones for other sizes. Again, the methods do not always agree on final models!

4. Stepwise Regression

Starts off looking like forward selection but allows for predictors to be kicked out as the process goes.

```
stepwise <- regsubsets(cesd ~ age + d1 + drugrisk + i1 + i2 + indtot +
               mcs + pcs + pss_fr + sexrisk, data = HELPrct, method = "seqrep", nbest = 1)
with(summary(stepwise), data.frame(cp, outmat))
```

```
##                 cp age d1 drugrisk i1 i2 indtot mcs pcs pss_fr sexrisk
## 1  ( 1 ) 56.79474                                *                    
## 2  ( 1 ) 13.65368                                *   *                
## 3  ( 1 ) 10.00083                  *             *   *                
## 4  ( 1 )  6.95863                  *             *   *      *         
## 5  ( 1 )  5.13689                  *             *   *      *       * 
## 6  ( 1 )  5.38979      *           *             *   *      *       * 
## 7  ( 1 )  5.98401      *           *         *   *   *      *       * 
## 8  ( 1 ) 15.28611   *  *        *  *  *      *   *   *
```

The final model based on Cp remains the same, but you can see the size 8 model was different.

Other Functions

There are functions in other libraries that can run these as well. stepAIC is in the MASS library(but suggest calling it as MASS::stepAIC due to issues with dplyr). Here are the same examples with other code for each method.

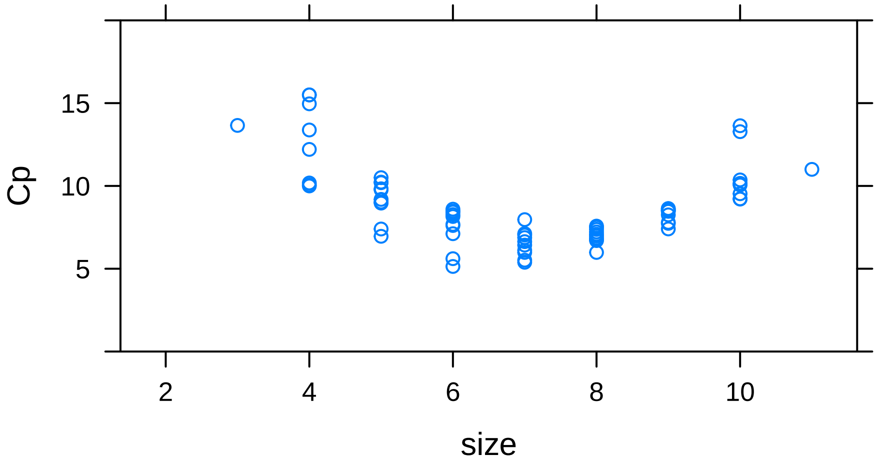
1. Best Subsets and Mallow's Cp Code

First we set the list of explanatory variables for the leaps function to work on. Then we tell it to run using the Cp method and plot the results.

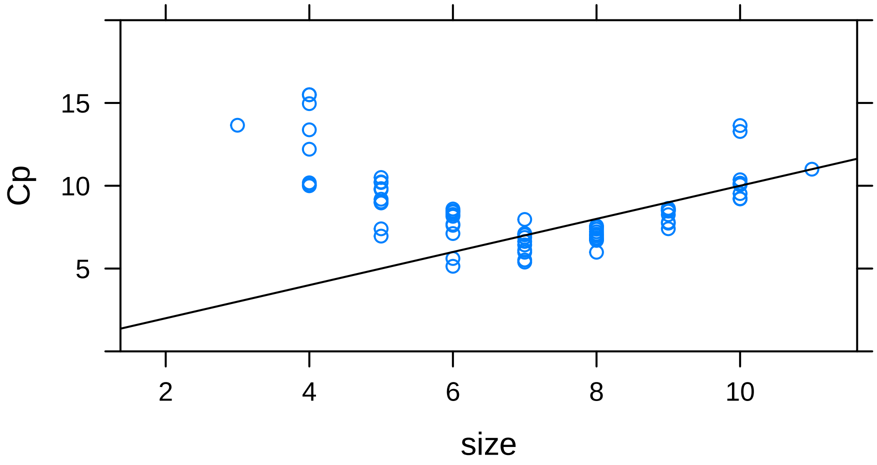
```
explanatory <- with(HELPrct, cbind(age, d1, drugrisk, i1, i2, indtot, mcs, pcs, pss_fr, sexrisk))
head(explanatory, 1) #take a look at the predictors, be sure correct set used
```

```
##      age d1 drugrisk i1 i2 indtot    mcs     pcs pss_fr sexrisk
## [1,]  37  3        0 13 26     39 25.112 58.4137      0       4
```

```
results <- with(HELPrct, leaps(explanatory, cesd, method = "Cp")) #obtain best subsets results with Cp
xyplot(Cp ~ size, ylim = c(0,20), data = results)
```



```
ladd(panel.abline(a = 0,b = 1))
```



Note that reported size includes the intercept term, so since Cp for a full set of k terms would be k+1, size=max Cp. We add the line size=Cp to consider where good models are (below the line). We can examine lots of properties about the best subsets solutions as follows:

```
favstats(results$Cp ~ results$size)
```

```
##   results$size      min        Q1    median        Q3       max      mean
## 1            2 56.79474 425.56890 463.89212 482.04233 493.78231 419.32801
## 2            3 13.65368  48.19208  54.07016  57.35077 331.83470  77.04619
## 3            4 10.00083  10.68444  14.16933  15.49684  43.03201  18.60632
## 4            5  6.95863   8.97890   9.47434  10.10326  10.49997   9.20512
## 5            6  5.13689   7.23806   7.91788   8.38488   8.59630   7.50656
## 6            7  5.38979   6.01655   6.53127   6.99522   7.96410   6.50609
## 7            8  5.98401   6.78643   7.02933   7.33634   7.57403   7.00342
## 8            9  7.40393   7.91274   8.48131   8.54091   8.63522   8.24900
## 9            10  9.20697   9.66147  10.26019  13.54672 317.07469  44.56185
## 10           11 11.00000  11.00000  11.00000  11.00000  11.00000  11.00000
##              sd  n missing
## 1  131.827484 10       0
## 2   90.476552 10       0
## 3   12.599104 10       0
## 4    1.192551 10       0
## 5    1.219571 10       0
## 6    0.788630 10       0
## 7    0.468878 10       0
## 8    0.439407 10       0
## 9   96.304165 10       0
## 10         NA  1       0
```

```
minimum <- which.min(results$Cp); minimum #determine which model has lowest Cp value
```

```
## [1] 41
```

```
results$Cp[minimum] #determine value of lowest Cp
```

```
## [1] 5.13689
```

```
results$which[minimum, ] #pull out model that has minimum Cp
```

```
##     1     2     3     4     5     6     7     8     9     A
## FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

In order to determine the model, note that you need to match up the predictors in order with the list. I found it helpful to reprint the head command above to match up the predictors. We can then fit the best subset model, using only those variables marked "True".

```
Cpmod <- lm(cesd ~ i1 + mcs + pcs + pss_fr + sexrisk, data = HELPrct); summary(Cpmod)
```

```
##
## Call:
## lm(formula = cesd ~ i1 + mcs + pcs + pss_fr + sexrisk, data = HELPrct)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -26.75  -5.94   0.04   5.35  25.54
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.0192     2.3428   28.18  < 2e-16 ***
## i1            0.0505     0.0209    2.42    0.016 *
## mcs          -0.6324     0.0324  -19.51  < 2e-16 ***
## pcs          -0.2291     0.0388   -5.91  6.9e-09 ***
## pss_fr       -0.2526     0.1040   -2.43    0.016 *
## sexrisk      -0.2887     0.1476   -1.96    0.051 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.65 on 446 degrees of freedom
## Multiple R-squared:  0.527,  Adjusted R-squared:  0.522
## F-statistic: 99.5 on 5 and 446 DF,  p-value: <2e-16
```

How does this solution compare to the model with all 10 predictors?

2. Backward elimination

Recall that backward elimination starts from the full model. The process is achieved in R as follows, with a simple summary table provided.

```
MASS::stepAIC(modall, direction="backward",trace=FALSE)$anova
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## cesd ~ age + d1 + drugrisk + i1 + i2 + indtot + mcs + pcs + pss_fr + 
##     sexrisk
## 
## Final Model:
## cesd ~ i1 + mcs + pcs + pss_fr + sexrisk
## 
## 
##         Step Df Deviance Resid. Df Resid. Dev     AIC
## 1                              441    33084.4 1962.50
## 2       - i2  1  15.5274       442    33100.0 1960.71
## 3 - drugrisk  1  14.7760       443    33114.7 1958.91
## 4      - age  1  43.5185       444    33158.2 1957.50
## 5   - indtot  1 105.4637       445    33263.7 1956.94
## 6       - d1  1 131.0699       446    33394.8 1956.72
```

3. Forward selection

For forward selection, we start from a model with just an intercept and build up a model one predictor at a time. We have to tell it a model that has the maximal list of predictors, as well as what our minimum is (just intercept).

```
modsmall <- lm(cesd ~ 1, data = HELPrct) #fits a model with just an intercept
MASS::stepAIC(modsmall, scope = list(upper = modall, lower = ~1), direction="forward",trace=FALSE)$anova
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## cesd ~ 1
## 
## Final Model:
## cesd ~ mcs + pcs + i1 + pss_fr + sexrisk
## 
## 
##        Step Df  Deviance Resid. Df Resid. Dev     AIC
## 1                              451    70663.8 2285.51
## 2     + mcs  1 32793.449       450    37870.4 2005.57
## 3     + pcs  1  3386.544       449    34483.8 1965.22
## 4      + i1  1   424.085       448    34059.8 1961.63
## 5  + pss_fr  1   378.273       447    33681.5 1958.58
## 6 + sexrisk  1   286.712       446    33394.8 1956.72
```

4. Stepwise Regression

Starts off looking like forward selection but allows for predictors to be kicked out as the process goes.

```
MASS::stepAIC(modsmall, scope = list(upper = modall, lower = ~1), direction = "both",trace = FALSE)$ano
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## cesd ~ 1
##
## Final Model:
## cesd ~ mcs + pcs + i1 + pss_fr + sexrisk
##
##
##         Step Df  Deviance Resid. Df Resid. Dev     AIC
## 1                               451    70663.8 2285.51
## 2      + mcs  1 32793.449       450    37870.4 2005.57
## 3      + pcs  1  3386.544       449    34483.8 1965.22
## 4       + i1  1   424.085       448    34059.8 1961.63
## 5   + pss_fr  1   378.273       447    33681.5 1958.58
## 6  + sexrisk  1   286.712       446    33394.8 1956.72
```

Note that stepAIC can give you individual models based on its internal criteria OR you can print out the steps to find other size models. The regsubsets command lets you pick descriptive statistics about the fit that you want to use for assessment.