# Regression Variable Selection in R

Mike Levine

June 29, 2019

**Background:** Variable Selection in regression is a tool that allows us to, simply put, pick better predictors for a model. To understand how this works, let's review a couple concepts surrounding regression.

When picking variables for a particular model, there's two ways we can go about this. First, there's 'Hierarchical.' This means we select predictors based on past research (or we add a small number of variables and slowly add others we suspect are correlated with our dependent variable. Forced entry, on the other hand, is when we add all known variables in our dataset to a model to start, and remove variables as necessary. The order we add variables to our model matters for determining how effective the model will be. Thus, this is where variable selection comes into play.

1. 'Forward' is the first type of variable selection. Here, we start with no predictors and add variables in the order that they're correlated with our dependent variable.
2. 'Backward' is the reverse. Instead of starting with no predictors, we add all of the variables in our dataset to the model. We remove variables that hurt our model, and we keep removing them until the improvement to our model is minimal.
3. 'Both' is the third variable selection method. Here, we essentially blend both 'Forward' and 'Backward' variable selection. In other words, we may add highly correlated variables to start but remove the least helpful variables at the same time.
4. Lastly, there's 'All-Subsets' Regression. This method tries every possible combination of variables and chooses the best model. The issue with this method is that it's computationally heavy.

Luckily, each one of these methods is fairly automated in R. It's just a matter of a couple lines of code to run a regression model using each of these.

***Note:*** *The assignment below was originally completed through the Fox School of Business at Temple University's Master of Science in Business Analytics program. It was completed for STAT 5607-308 (Advanced Business Statistics) in spring 2019.*

**Start:** In this tutorial, our goal is to fit a model that predicts 'salary' for Major League Baseball (MLB) players. First, let's start by setting our working directory and pulling in our dataset, 'MLB1.csv.'

```
setwd("C:/Users/mdlev/OneDrive/Documents/Education/Graduate - Temple
University/2nd Semester/Advanced Business Statistics/R Datasets")
getwd()
```

```
## [1] "C:/Users/mdlev/OneDrive/Documents/Education/Graduate - Temple
University/2nd Semester/Advanced Business Statistics/R Datasets"
dat <- read.csv("MLB1.csv", header = T)

head(dat)
##    salary  teamsal nl years games atbats runs hits doubles triples hruns
## 1 6329213 38407380  1    12  1705   6705 1076 1939     320      67   231
## 2 3375000 38407380  1     8   918   3333  407  863     156      38    73
## 3 3100000 38407380  1     5   751   2807  370  840     148      18    46
## 4 2900000 38407380  1     8  1056   3337  405  816     143      18   107
## 5 1650000 38407380  1    12  1196   3603  437  928      19      16   124
## 6  700000 38407380  1    17  2032   7489 1136 2145     270     142    40
##   rbis bavg  bb   so sbases fldperc frstbase scndbase shrtstop thrdbase
## 1  836  289 619  948    314     989        0        1        0        0
## 2  342  259 137  582    133     968        0        0        1        0
## 3  355  299 341  228     41     994        1        0        0        0
## 4  421  245 306  653     15     971        0        0        0        1
## 5  541  258 316  725     32     977        0        0        0        0
## 6  574  286 416 1098    660     987        0        0        0        0
##   outfield catcher yrsallst hispan black whitepop blackpop hisppop pcinc
## 1        0       0        9      0     0  5772110  1547725  893422 18840
## 2        0       0        2      0     1  5772110  1547725  893422 18840
## 3        0       0        0      0     0  5772110  1547725  893422 18840
## 4        0       0        0      0     0  5772110  1547725  893422 18840
## 5        1       0        0      0     1  5772110  1547725  893422 18840
## 6        1       0        2      0     1  5772110  1547725  893422 18840
##     gamesyr   hrunsyr atbatsyr  allstar  slugavg   rbisyr  sbasesyr
## 1 142.08330 19.250000 558.7500 75.00000 46.02535 69.66666 26.166670
## 2 114.75000  9.125000 416.6250 25.00000 39.42394 42.75000 16.625000
## 3 150.20000  9.200000 561.4000  0.00000 41.39651 71.00000  8.200000
## 4 132.00000 13.375000 417.1250  0.00000 39.43662 52.62500  1.875000
## 5  99.66666 10.333330 300.2500  0.00000 37.49653 45.08333  2.666667
## 6 119.52940  2.352941 440.5294 11.76471 37.64188 33.76471 38.823530
##     runsyr percwhte percblck perchisp
## 1 89.66666 70.27797 18.84423  10.8778
## 2 50.87500 70.27797 18.84423  10.8778
## 3 74.00000 70.27797 18.84423  10.8778
## 4 50.62500 70.27797 18.84423  10.8778
## 5 36.41667 70.27797 18.84423  10.8778
## 6 66.82353 70.27797 18.84423  10.8778
```

Above we can see each of the variables in our dataset. Beyond our dependent variable, we have a couple inepdendent variables that we'll be using to try and predict salary.

First, let's name our 'null' and 'full variables. This will help us later on when we run each type of variable selection.

```
null <- lm(salary ~ 1, data=dat)
full <- lm(salary ~ ., data=dat)
```

Now let's use forward stepwise regression to run our model.

```
forward <- step(null, scope=list(lower=null, upper=full),
direction="forward")
summary(forward)
##
## Call:
## lm(formula = salary ~ rbisyr + allstar + runs + yrsallst + teamsal +
##     hrunsyr + so + nl + frstbase, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2567245  -455377   -63570   307666  2515742
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.948e+05  2.003e+05  -1.971   0.0495 *
## rbisyr       1.327e+04  5.480e+03   2.422   0.0160 *
## allstar      4.649e+04  6.513e+03   7.139 5.64e-12 ***
## runs         2.170e+03  3.594e+02   6.039 4.04e-09 ***
## yrsallst    -3.057e+05  6.611e+04  -4.624 5.33e-06 ***
## teamsal      1.285e-02  5.482e-03   2.344   0.0197 *
## hrunsyr      3.732e+04  1.633e+04   2.285   0.0230 *
## so          -6.327e+02  2.695e+02  -2.348   0.0195 *
## nl           1.451e+05  9.390e+04   1.545   0.1231
## frstbase    -2.105e+05  1.432e+05  -1.470   0.1424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849800 on 343 degrees of freedom
## Multiple R-squared:  0.6447, Adjusted R-squared:  0.6354
## F-statistic: 69.16 on 9 and 343 DF,  p-value: < 2.2e-16
forward$anova
##          Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1             NA           NA       352 6.971850e+14 9995.996
## 2    + rbisyr -1 3.489791e+14       351 3.482059e+14 9752.924
## 3   + allstar -1 6.027341e+13       350 2.879325e+14 9687.830
## 4      + runs -1 1.141696e+13       349 2.765155e+14 9675.547
## 5  + yrsallst -1 1.745797e+13       348 2.590576e+14 9654.526
## 6   + teamsal -1 3.553647e+12       347 2.555039e+14 9651.650
## 7   + hrunsyr -1 1.634313e+12       346 2.538696e+14 9651.385
## 8        + so -1 2.830185e+12       345 2.510394e+14 9649.428
## 9        + nl -1 1.784343e+12       344 2.492551e+14 9648.910
## 10 + frstbase -1 1.561246e+12       343 2.476938e+14 9648.692
```

Looking at our output, we can see that our model performs fairly well. Our adjusted R2 value is 0.6354 which is fairly good. Our p-values, with the exception of 'nl' and 'frstbase,' are below 0.05 and therefore significant.

Now let's use backward stepwise regression to run our model.

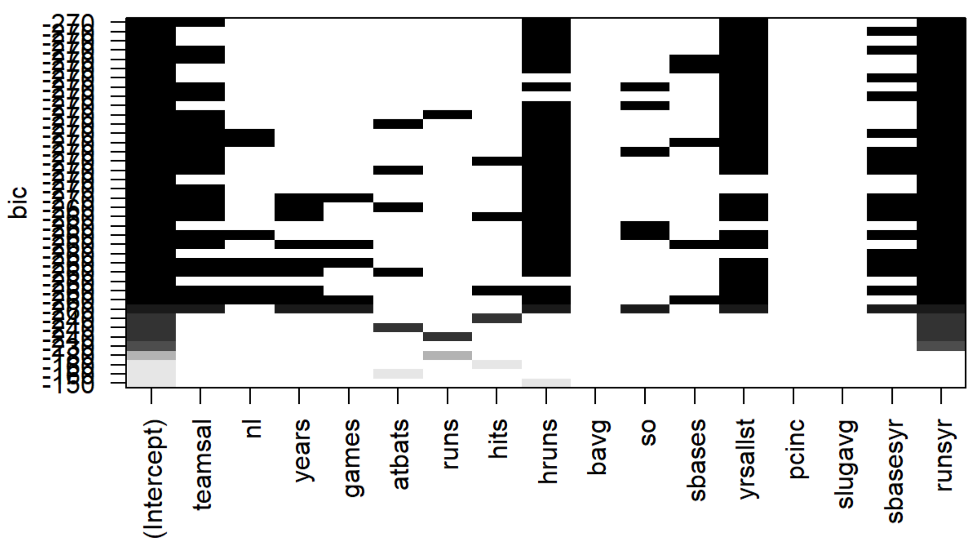
```
backward <- step(full, direction="backward")
summary(both)
##
## Call:
## lm(formula = salary ~ rbisyr + allstar + runs + yrsallst + teamsal +
##     hrunsyr + so + nl + frstbase, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2567245  -455377   -63570   307666  2515742
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.948e+05  2.003e+05  -1.971   0.0495 *
## rbisyr       1.327e+04  5.480e+03   2.422   0.0160 *
## allstar      4.649e+04  6.513e+03   7.139 5.64e-12 ***
## runs         2.170e+03  3.594e+02   6.039 4.04e-09 ***
## yrsallst    -3.057e+05  6.611e+04  -4.624 5.33e-06 ***
## teamsal      1.285e-02  5.482e-03   2.344   0.0197 *
## hrunsyr      3.732e+04  1.633e+04   2.285   0.0230 *
## so          -6.327e+02  2.695e+02  -2.348   0.0195 *
## nl           1.451e+05  9.390e+04   1.545   0.1231
## frstbase    -2.105e+05  1.432e+05  -1.470   0.1424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849800 on 343 degrees of freedom
## Multiple R-squared:  0.6447, Adjusted R-squared:  0.6354
## F-statistic: 69.16 on 9 and 343 DF,  p-value: < 2.2e-16
both$anova
##          Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1             NA           NA       352 6.971850e+14 9995.996
## 2    + rbisyr -1 3.489791e+14       351 3.482059e+14 9752.924
## 3   + allstar -1 6.027341e+13       350 2.879325e+14 9687.830
## 4      + runs -1 1.141696e+13       349 2.765155e+14 9675.547
## 5  + yrsallst -1 1.745797e+13       348 2.590576e+14 9654.526
## 6   + teamsal -1 3.553647e+12       347 2.555039e+14 9651.650
## 7   + hrunsyr -1 1.634313e+12       346 2.538696e+14 9651.385
## 8        + so -1 2.830185e+12       345 2.510394e+14 9649.428
## 9        + nl -1 1.784343e+12       344 2.492551e+14 9648.910
## 10 + frstbase -1 1.561246e+12       343 2.476938e+14 9648.692
```

The 'Both' method yielded an equally comparable model. It's adjusted R2 value is 0.6354 and its p-values are mostly below 0.05.

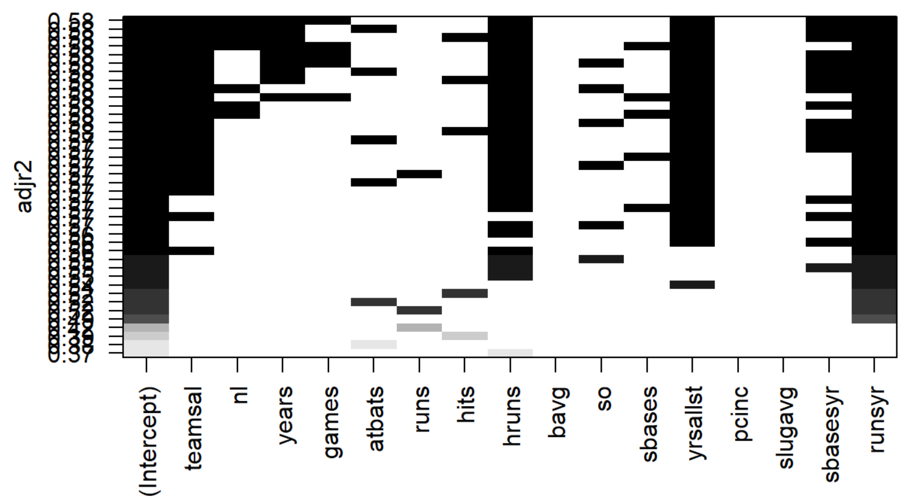
Each of the three stepwise regression methods produced comparable models. The only real difference, aside from the obvious differences in variables selected, is that the backward method produced a slightly higher adjusted R-squared.

Now let's use the 'All-Subsets' variable selection method.

```
library(leaps)
allSubset <-regsubsets(salary ~ teamsal + nl + years + games + atbats + runs
+ hits + hruns + bavg + so + sbases + yrsallst + pcinc + slugavg + sbasesyr +
runsyr,data=dat, nbest=5)
plot(allSubset, scale = "bic")
```



```
plot(allSubset, scale = "adjr2")
```




```
allSubsetMod <- lm(dat$salary ~ dat$teamsal + dat$nl + dat$years + dat$games
+ dat$hruns + dat$yrsallst + dat$sbasesyr + dat$runsyr)
summary(allSubsetMod)
##
## Call:
## lm(formula = dat$salary ~ dat$teamsal + dat$nl + dat$years +
##     dat$games + dat$hruns + dat$yrsallst + dat$sbasesyr + dat$runsyr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2453093  -517701   -40846   368501  2744557
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.412e+05  2.346e+05  -3.586 0.000384 ***
## dat$teamsal   1.397e-02  5.857e-03   2.385 0.017634 *
## dat$nl        1.617e+05  1.007e+05   1.606 0.109284
## dat$years     1.061e+05  4.807e+04   2.207 0.028009 *
## dat$games    -1.049e+03  4.476e+02  -2.343 0.019682 *
## dat$hruns     3.702e+03  1.190e+03   3.110 0.002030 **
## dat$yrsallst  1.574e+05  3.624e+04   4.343 1.85e-05 ***
## dat$sbasesyr -1.580e+04  6.203e+03  -2.547 0.011297 *
## dat$runsyr    3.833e+04  4.172e+03   9.189  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 911500 on 344 degrees of freedom
## Multiple R-squared:   0.59,  Adjusted R-squared:  0.5805
## F-statistic: 61.89 on 8 and 344 DF,  p-value: < 2.2e-16
```

Using all-subsets regression, we found the best model to be salary (predicted) = eamsal + nl + years + games + hruns + yrsallst + sbasesyr + runsyr. Our adjusted R2 value is 0.5805, which is less than our other models. Our p-values are mostly significant.

Now let’s use AIC to score each of our models. To review, we’ve selected variables using four types of variable selection: forward, backward, both, and all-subsets regression. Using AIC to judge each of our models will generate a standardized scale. Essentially, the lower the number the better the model. Looking at our AIC values below, we can determine that the backward model performs (minimally) better than the others since it has the lowest AIC score of 10,652.

```
AIC(forward)
## [1] 10652.46
AIC(backward)
## [1] 10652
AIC(both)
## [1] 10652.46
AIC(allSubsetMod)
## [1] 10700.99
```

Variable selection is a powerful tool in R to automate the selecting of variables in a model. While each type of variable selection comes with its pros and cons, the ability to judge each model by the AIC is immensely helpful.