Math 158, Spring 2018
Jo Hardin
Multiple Regression IV – R code
Model Building

Consider the multiple regression model:

$$
\begin{aligned}
E[Y] &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 \\
Y &= \text{state ave SAT score} \\
X_1 &= \text{\% of eligible seniors who took the exam, } \textit{takers} \\
X_2 &= \text{median income of families of test takers, } \textit{income} \\
X_3 &= \text{ave number of years of formal eduction, } \textit{years} \\
X_4 &= \text{\% of test takers who attend public school, } \textit{public} \\
X_5 &= \text{total state expenditure on public secondary schools (\$100 /student), } \textit{expend} \\
X_6 &= \text{median percentile rank of test takers within their secondary school class, } \textit{rank}
\end{aligned}
$$

```
sat.data <- read.csv("http://pages.pomona.edu/~jsh04747/courses/math158/sat.csv",
                     header=TRUE)
sat.n <- nrow(sat.data)                # be careful with missing values!!
sat.data$ltakers <- log(sat.data$takers)                # variable is quite right skewed
```

AIC and BIC in R

1.
```
sat.lm0 <- lm(sat ~ 1, data=sat.data)
summary(sat.lm0)$coef

##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)  948.449   10.21404 92.85739 7.887706e-56

sat.sse0 <- sum(resid(sat.lm0) ^2)
sat.n + sat.n*log(2*pi) + sat.n * log(sat.sse0 / sat.n) + 2 * (1+1)

## [1] 560.4736

AIC(sat.lm0, k=2)

## [1] 560.4736

sat.n + sat.n * log(2*pi) + sat.n*log(sat.sse0/sat.n) + log(sat.n)*(1+1)

## [1] 564.2573

AIC(sat.lm0, k=log(sat.n))

## [1] 564.2573
```

2.
```r
sat.lm1 <- lm(sat ~ ltakers, data=sat.data)
summary(sat.lm1)$coef

##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 1112.40781  12.385667  89.81412 3.108860e-54
## ltakers      -59.17547   4.166524 -14.20260 1.265819e-18

sat.sse1 <- sum(resid(sat.lm1) ^2)
sat.n + sat.n*log(2*pi) + sat.n * log(sat.sse1 / sat.n) + 2 * (2+1)

## [1] 480.832

AIC(sat.lm1, k=2)

## [1] 480.832

sat.n + sat.n * log(2*pi) + sat.n*log(sat.sse1/sat.n) + log(sat.n) * (2+1)

## [1] 486.5075

AIC(sat.lm1, k=log(sat.n))

## [1] 486.5075
```

These notes belong at the end... I'm putting them here to save paper and keep the formatting reasonably clear.

- Notice that C_p and F-tests use a "full" model MSE. Typically, the MSE will only be an unbiased predictor of σ^2 in backwards variable selection.
- SBC usually results in fewer parameters in the model than AIC.
- Using different selection criteria may lead to different models (there is no one best model).
- The order in which variables are entered does not necessarily represent their importance. As a variable entered early on can be dropped at a later stage because it is predicted well from the other explanatory variables that have been subsequently added to the model.

Forward Variable Selection: F-tests

```
add1(lm(sat~1, data=sat.data), sat~ ltakers + income + years + public + expend + rank, test="F")

## Single term additions
##
## Model:
## sat ~ 1
##         Df Sum of Sq    RSS    AIC  F value    Pr(>F)
## <none>               245376 419.42
## ltakers  1    199007  46369 339.78 201.7138 < 2.2e-16 ***
## income   1    102026 143350 395.08  33.4513 5.711e-07 ***
## years    1     26338 219038 415.85   5.6515   0.02156 *
## public   1      1232 244144 421.17   0.2371   0.62856
## expend   1       386 244991 421.34   0.0740   0.78683
## rank     1    190297  55079 348.21 162.3828 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

add1(lm(sat~ ltakers, data=sat.data), sat~ ltakers + income + years + public + expend + rank, test="F")

## Single term additions
##
## Model:
## sat ~ ltakers
##        Df Sum of Sq   RSS    AIC F value    Pr(>F)
## <none>              46369 339.78
## income  1     785.1 45584 340.94  0.7922  0.378064
## years   1    6363.5 40006 334.54  7.3170  0.009546 **
## public  1     448.9 45920 341.30  0.4497  0.505838
## expend  1   20523.5 25846 313.14 36.5274 2.489e-07 ***
## rank    1     871.1 45498 340.85  0.8807  0.352900
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

add1(lm(sat~ ltakers + expend, data=sat.data), sat~ ltakers + income + years + public +
       expend + rank, test="F")

## Single term additions
##
## Model:
## sat ~ ltakers + expend
##        Df Sum of Sq   RSS    AIC F value Pr(>F)
## <none>              25846 313.14
## income  1     53.33 25792 315.04  0.0930 0.7617
## years   1   1248.18 24598 312.71  2.2835 0.1377
## public  1      1.29 25844 315.13  0.0023 0.9624
## rank    1   1053.60 24792 313.10  1.9124 0.1735
```

Note: `Sum of Sq` refers to the SSR(new variable | current model) (additional reduction in SSE). `RSS` is the SSE for the model that contains the current variables and the new variable.

Backward Variable Selection: F-tests

```
drop1(lm(sat ~ ltakers + income + years + public + expend + rank, data=sat.data), test="F")

## Single term deletions
##
## Model:
## sat ~ ltakers + income + years + public + expend + rank
##         Df Sum of Sq   RSS    AIC F value    Pr(>F)
## <none>               21397 311.88
## ltakers  1    2150.0 23547 314.57  4.2203   0.04620 *
## income   1     340.3 21737 310.65  0.6681   0.41834
## years    1    2531.6 23928 315.36  4.9693   0.03121 *
## public   1      20.0 21417 309.93  0.0393   0.84390
## expend   1   10964.4 32361 330.15 21.5221 3.404e-05 ***
## rank     1    2679.0 24076 315.66  5.2587   0.02691 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

drop1(lm(sat ~ ltakers + income + years + expend + rank, data=sat.data), test="F")

## Single term deletions
##
## Model:
## sat ~ ltakers + income + years + expend + rank
##         Df Sum of Sq   RSS    AIC F value    Pr(>F)
## <none>               21417 309.93
## ltakers  1    2551.7 23968 313.44  5.1232   0.02871 *
## income   1     505.4 21922 309.07  1.0147   0.31942
## years    1    3010.8 24428 314.37  6.0451   0.01805 *
## expend   1   12465.4 33882 330.40 25.0277 1.003e-05 ***
## rank     1    3161.7 24578 314.67  6.3480   0.01555 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you `add1` here (to see if it makes sense to add either public or income back), neither is significant.

```
drop1(lm(sat ~ ltakers + years + expend + rank, data=sat.data), test="F")

## Single term deletions
##
## Model:
## sat ~ ltakers + years + expend + rank
##         Df Sum of Sq   RSS    AIC F value   Pr(>F)
## <none>               21922 309.07
## ltakers  1    5094.3 27016 317.31 10.2249 0.002568 **
## years    1    2870.1 24792 313.10  5.7606 0.020687 *
## expend   1   13619.6 35542 330.75 27.3360 4.52e-06 ***
## rank     1    2675.5 24598 312.71  5.3700 0.025200 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Forward Stepwise: AIC

```r
step(lm(sat~1, data=sat.data), sat ~ ltakers + income + years + public + expend +  rank,
     direction = "forward")
```

```
## Start:  AIC=419.42
## sat ~ 1
##
##           Df Sum of Sq    RSS    AIC
## + ltakers  1    199007  46369 339.78
## + rank     1    190297  55079 348.21
## + income   1    102026 143350 395.08
## + years    1     26338 219038 415.85
## <none>                 245376 419.42
## + public   1      1232 244144 421.17
## + expend   1       386 244991 421.34
##
## Step:  AIC=339.78
## sat ~ ltakers
##
##          Df Sum of Sq   RSS    AIC
## + expend  1   20523.5 25846 313.14
## + years   1    6363.5 40006 334.54
## <none>                46369 339.78
## + rank    1     871.1 45498 340.85
## + income  1     785.1 45584 340.94
## + public  1     448.9 45920 341.30
##
## Step:  AIC=313.14
## sat ~ ltakers + expend
##
##          Df Sum of Sq   RSS    AIC
## + years   1   1248.18 24598 312.71
## + rank    1   1053.60 24792 313.10
## <none>                25846 313.14
## + income  1     53.33 25792 315.04
## + public  1      1.29 25844 315.13
##
## Step:  AIC=312.71
## sat ~ ltakers + expend + years
##
##          Df Sum of Sq   RSS    AIC
## + rank    1   2675.51 21922 309.07
## <none>                24598 312.71
## + public  1    287.82 24310 314.13
## + income  1     19.19 24578 314.67
##
## Step:  AIC=309.07
## sat ~ ltakers + expend + years + rank
##
##          Df Sum of Sq   RSS    AIC
## <none>                21922 309.07
## + income  1    505.37 21417 309.93
## + public  1    185.03 21737 310.65
##
## Call:
## lm(formula = sat ~ ltakers + expend + years + rank, data = sat.data)
##
## Coefficients:
## (Intercept)      ltakers       expend        years         rank
##     399.115      -38.100        3.996       13.147        4.400
```

Backward Stepwise: SBC/BIC

```
step(lm(sat ~ (ltakers + income + years + public + expend + rank), data=sat.data),
     direction = "backward", k=log(sat.n))

## Start:  AIC=325.12
## sat ~ (ltakers + income + years + public + expend + rank)
##
##           Df Sum of Sq   RSS    AIC
## - public   1      20.0 21417 321.28
## - income   1     340.3 21737 322.00
## <none>                 21397 325.12
## - ltakers  1    2150.0 23547 325.92
## - years    1    2531.6 23928 326.71
## - rank     1    2679.0 24076 327.01
## - expend   1   10964.4 32361 341.50
##
## Step:  AIC=321.28
## sat ~ ltakers + income + years + expend + rank
##
##           Df Sum of Sq   RSS    AIC
## - income   1     505.4 21922 318.53
## <none>                 21417 321.28
## - ltakers  1    2551.7 23968 322.90
## - years    1    3010.8 24428 323.83
## - rank     1    3161.7 24578 324.13
## - expend   1   12465.4 33882 339.86
##
## Step:  AIC=318.53
## sat ~ ltakers + years + expend + rank
##
##           Df Sum of Sq   RSS    AIC
## <none>                 21922 318.53
## - rank     1    2675.5 24598 320.28
## - years    1    2870.1 24792 320.66
## - ltakers  1    5094.3 27016 324.87
## - expend   1   13619.6 35542 338.31
##
## Call:
## lm(formula = sat ~ ltakers + years + expend + rank, data = sat.data)
##
## Coefficients:
## (Intercept)      ltakers        years       expend         rank
##     399.115      -38.100       13.147        3.996        4.400
```

To get an idea of how complicated your models can get, try this:

```
> step(lm(sat ~ (ltakers + income + years + public + expend + rank)^2),
             direction = "backward")
```

Cross Validation

The `caret` package estimates the CV prediction error. Note that RMSE (square root of MSE) from `print(model)` gives the CV RMSE which is *higher* than the RMSE (`residual standard error`) that comes from the entire dataset. The full model RMSE will be smaller than the CV RMSE because when all the points are used to find the coefficients, those coefficients produce the smallest possible RMSE.

```
library(caret)
model <- train( sat ~ ltakers + income + years + public + expend + rank, data=sat.data,
  method = "lm",
  trControl = trainControl(
    method = "cv", number = 10,
  )
)
print(model)

## Linear Regression
##
## 49 samples
##  6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 44, 44, 44, 44, 44, 44, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   24.59338  0.894614  21.08809
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

summary(lm(sat ~ ltakers + income + years + public + expend + rank, data=sat.data))

##
## Call:
## lm(formula = sat ~ ltakers + income + years + public + expend +
##     rank, data = sat.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.447 -10.361  -2.626  11.101  59.001
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 287.5242   259.4170   1.108   0.2740
## ltakers     -30.2149    14.7079  -2.054   0.0462 *
## income        0.1029     0.1259   0.817   0.4183
## years        13.1073     5.8798   2.229   0.0312 *
## public       -0.1011     0.5105  -0.198   0.8439
## expend        3.9367     0.8486   4.639  3.4e-05 ***
```

```
## rank          5.2738     2.2997   2.293   0.0269 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.57 on 42 degrees of freedom
## Multiple R-squared:  0.9128,Adjusted R-squared:  0.9003
## F-statistic: 73.28 on 6 and 42 DF,  p-value: < 2.2e-16
```