Linear Regression Models

STATS 762 – Lecture Slides 7

01.04.2019

Variable Selection

So far we have mainly looked at applications where we have a relatively small number of explanatory variables.

- Start by fitting a suitable model (ordinary, logistic or Poisson regression).
- Do diagnostic checks and fix up the model as necessary.

Variable Selection (cont.)

Sometimes we encounter a situation where we have a pool "candidate" explanatory variables and deciding which of these potential regressors should be included in the model is a non-trivial problem.

- We could, of course, use them all. However, sometimes this turns out to be not such a good idea.
- As the number of regressors increases it becomes less and less feasible for us to try different models one at a time.
- It becomes necessary to automate the process – fortunately computers are good at this sort of thing.
- The more rubbish in the variable list, the stronger the real relationships have to be before you can see them (Thomas Lumley).

Example

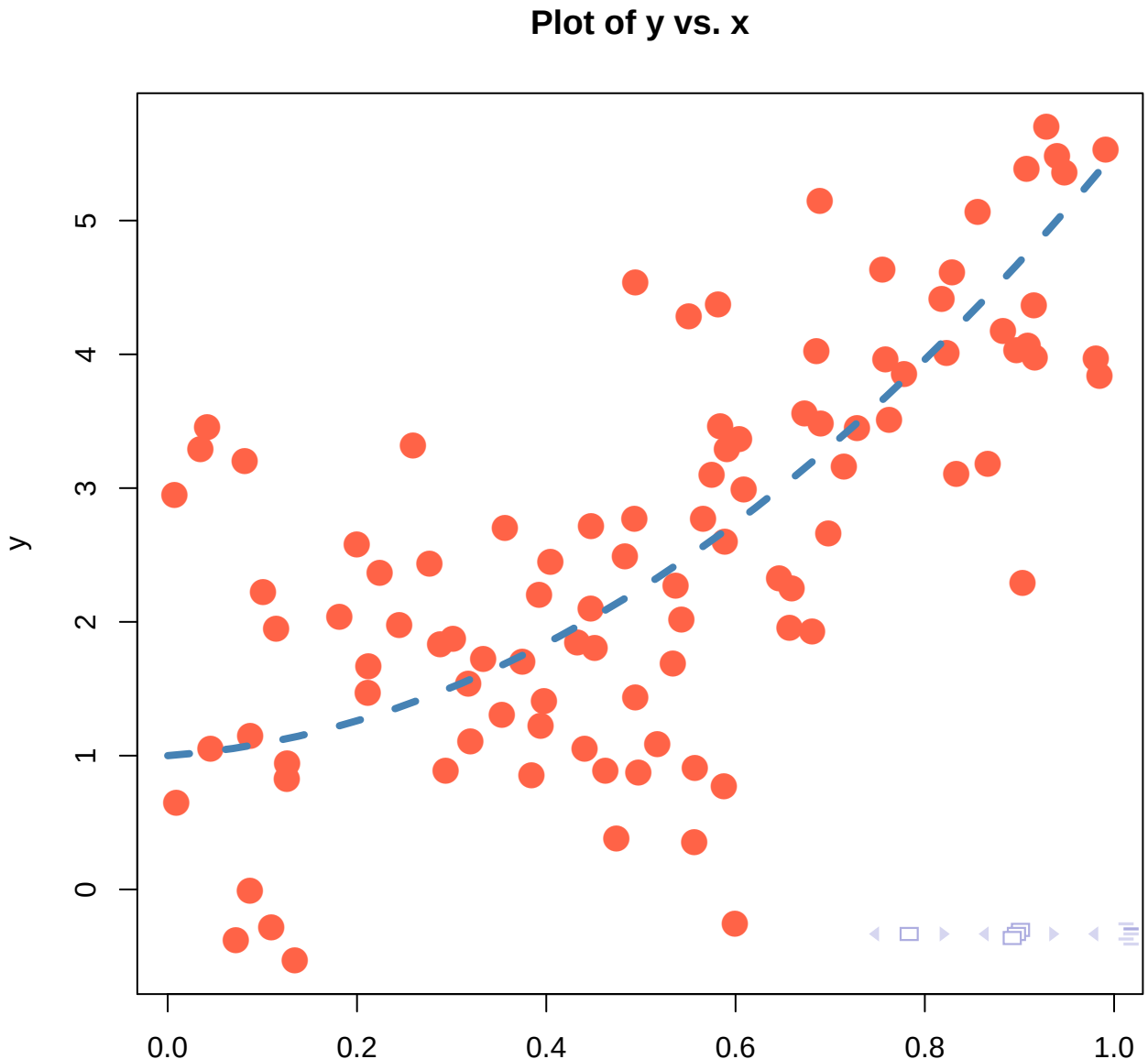
- Suppose we have some data which follow a quadratic model

$$Y = 1 + x/2 + 4x^2 + \varepsilon,$$

  where the x's are uniformly distributed on $[0, 1]$ and ε is standard normal distributed.

- The next slides shows the data, with the true data as a dashed line.

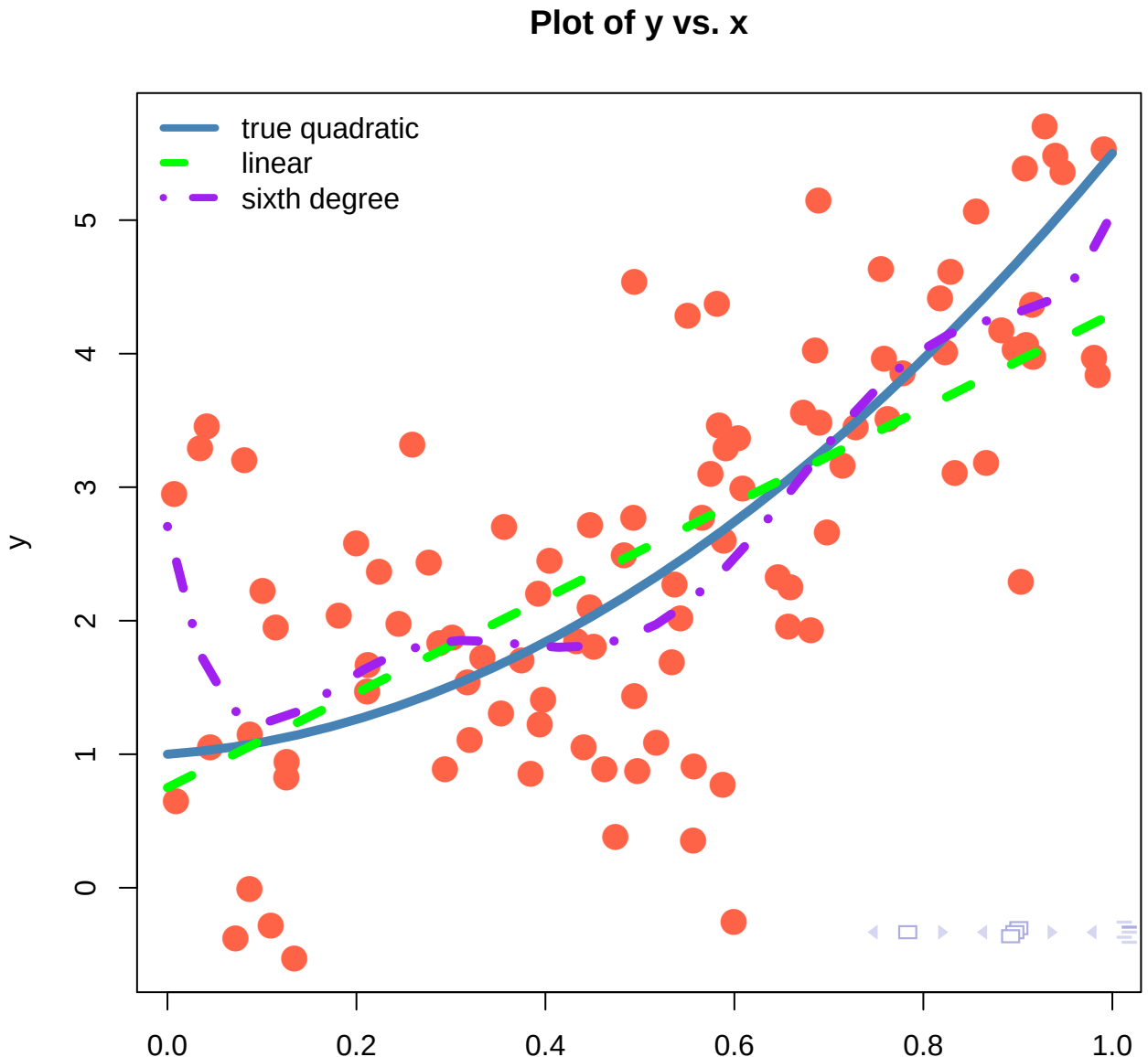
Data and true relationship



Under/Over fitting

- Suppose we fit a straight line. This is under-fitting, since we are not fitting the squared term. The fitted line (in green) is shown on the next slide.
- Alternatively, we could fit a 6-degree polynomial. This is over-fitting, since there are unnecessary terms in x^3, x^4, x^5 and x^6. The fitted polynomial is shown in magenta on the next slide. Fit using `lm(y~poly(x,6))`.

Data and true relationship



Points to note

- Straight line is biased: Cannot capture the curvature in the true regression.
- 6-degree line: too variable, attracted to the errors which would be different for a new set of data.
- Moral: For good models we need to choose the set of regressors wisely to avoid over-fitting and under-fitting. This is called variable selection.

Uses of regression

The two main uses are

1. To explain the role(s) of the explanatory variables in influencing the response.
   - Unless we can clearly identify the regressors that are affecting the response, it is risky to use a regression model for explanation.

2. To construct a prediction equation for predicting the response.
   - Model selection is more viable in this case as we aren't as concerned about finding the "true drivers".

Variable selection

If we have k variables, and assuming a constant term in each model, there are $2^k - 1$ possible subsets of variables, not counting the null model with no variables. How do we select a subset for our model?

Two main approaches:

- All possible regressions: Thorough but may be impractical for extensive data sets.
- Stepwise methods: Can handle much larger data sets but not necessarily optimal.

Evaporation Example

Data was recorded for factors affecting the evaporation water from a location in Texas.

- Ten variables were measured each day over a period of 46 days.
- The goal was to create a model that relates the daily amount of evaporation to the meteorological conditions that day.

Evaporation Example Variables

evap:	amount of evaporation from soil (response).
maxst:	maximum soil temperature
minst:	minimum soil temperature
avst:	average soil temperature
maxat:	maximum air temperature
minat:	minimum air temperature
avat:	average air temperature
maxh:	maximum humidity
minh:	minimum humidity
avh:	average humidity
wind:	average wind speed

The Data Frame

```
> head(evap.df)
  avst minst maxst avat minat maxat avh minh maxh wind evap
1   84    65   147   85    59   151  95   40  398  273   30
2   84    65   149   86    61   159  94   28  345  140   34
3   79    66   142   83    64   152  94   41  388  318   33
4   81    67   147   83    65   158  94   50  406  282   26
5   84    68   167   88    69   180  93   46  379  311   41
6   74    66   131   77    67   147  96   73  478  446    4
```

Variance Inflation Factors

The VIF's for the explanatory variables indicate extensive multicollinearity:

```
> round(diag(solve(cor(evap.df[,1:10]))),2)
 avst minst maxst  avat minat maxat   avh  minh  maxh  wind
39.29 14.08 52.34  8.83  8.89 22.22  1.98 25.38 24.12  1.98
```

The Variable Selection Problem

The goal is to find a good model for the amount of evaporation.

- Probably don't need all 10 regressors in the model.
- The total number of subsets for our 10 regressors is $2^{10} = 1024$.
- So which subset(s) are best?

The Full Model

```
> fit1.lm<-lm(evap~.,data=evap.df)
> summary(fit1.lm)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -54.074877 130.720826  -0.414  0.68164
avst          2.231782   1.003882   2.223  0.03276 *
minst         0.204854   1.104523   0.185  0.85393
maxst        -0.742580   0.349609  -2.124  0.04081 *
avat          0.501055   0.568964   0.881  0.38452
minat         0.304126   0.788877   0.386  0.70219
maxat         0.092187   0.218054   0.423  0.67505
avh           1.109858   1.133126   0.979  0.33407
minh          0.751405   0.487749   1.541  0.13242
maxh         -0.556292   0.161602  -3.442  0.00151 **
wind          0.008918   0.009167   0.973  0.33733
---
Residual standard error: 6.508 on 35 degrees of freedom
Multiple R-squared:  0.8463,Adjusted R-squared:  0.8023
F-statistic: 19.27 on 10 and 35 DF,  p-value: 2.073e-11
```

ANOVA Table

```
> anova(fit1.lm)
Response: evap
          Df Sum Sq Mean Sq  F value    Pr(>F)
avst       1 5705.6  5705.6 134.7227 1.495e-13 ***
minst      1  460.9   460.9  10.8838 0.0022355 **
maxst      1    7.0     7.0   0.1651 0.6870108
avat       1  171.4   171.4   4.0468 0.0520037 .
minat      1   97.5    97.5   2.3016 0.1382210
maxat      1  379.3   379.3   8.9568 0.0050424 **
avh        1  188.9   188.9   4.4605 0.0418956 *
minh       1  504.9   504.9  11.9217 0.0014685 **
maxh       1  604.2   604.2  14.2672 0.0005917 ***
wind       1   40.1    40.1   0.9463 0.3373290
Residuals 35 1482.3    42.4
```

Over Fitting

If we put too many variables in the model, including some unrelated to the response, we are over fitting. Consequences are:

- Fitted model is not good for prediction of new data - prediction error is underestimated.
- Model is too elaborate, models "noise" that will not be the same for new data.
- Variances of regression coefficients inflated (multicollinearity).

Under Fitting

If we put too few variables in the model, leaving out variables that could help explain the response, we are under fitting. Consequences are:

- Fitted model is not good for prediction - prediction is biased.
- Regression coefficients are biased (confounding).
- Estimate of error variance is inflated.

Choosing between Models

So far the only technique we have looked at for choosing between models are the added variable tests. There are difficulties in using these for model selection applications.

- These tests can only be used to compare a submodel to a full model.
- P-values are relevant for a single test but not a large number of tests. This raises the issue of what p-value should be used.

Choosing between Models (cont.)

A common approach is to identify a criterion for measuring "model goodness" which tries to balance over fitting (model too complex) with under fitting (model does not fit very well). and use this to select a model.

- A number of different criteria have been proposed based on different rationale.
- Our idea of a good criteria is affected by the purpose of the regression (explanation or prediction).
- In practice there is no such thing as the "true model" – the goal of model selection is to identify useful models.

Model Selection Criteria

We'll look at some of the more common (and more useful) model selection criteria.

- AIC and AIC_C.
- BIC.
- Mallow's C_p.
- Prediction error as measured by cross validation (Thomas will cover this).

AIC

Akaike's information criterion (AIC) was developed from an information theory perspective.

- Idea is to consider the information lost by using a model to approximate the true distribution of the response as measured by Kulback-Liebler divergence.
- The AIC for a given model is an estimate of the *relative* information loss incurred for that model.
- Useful for comparing models but it doesn't tell us whether an individual model is close to the true distribution or not.

AIC Formula

General formula for AIC is:

$$\text{AIC} = -2 \times \log(L) + 2p$$

- L is the likelihood and p is the total number of estimated parameters – for GLM's this includes the intercept and the scale parameter (if its estimated).

For ordinary regression this formula becomes:

$$\text{AIC} = n + n(\log 2\pi) + n \log(\text{RSS}/n) + 2(k + 2)$$

- k is the number of regressors (not counting the intercept).
- Terms that are identical for all models being considered can be omitted as only the relative values of AIC are important.

AIC Comments

AIC can be divided into two components: the log likelihood which evaluates how well the model fits the data and penalty for model complexity ($2p$).

- The penalty was derived based on asymptotic results – AIC is an asymptotically unbiased estimator of relative information loss.
- AIC has a tendency to over fit if the number of observations (n) is not sufficiently large relative to the number of regressors (k).

AICc

AICc is essentially AIC with a correction for small sample sizes:

$$\text{AICc} = \underbrace{-2 \times \log(L) + 2p}_{\text{AIC}} + \underbrace{\frac{2p(p+1)}{n-p-1}}_{\text{correction}}$$

- Increases the penalty as model size increases.
- The correction is exact for ordinary regression and approximate for other GLM's.
- As n gets large the correction goes to 0.

BIC

An alternative to AIC and AICc is the Bayesian information criterion (BIC).

- Idea is to identify the model that has the highest posterior probability.
- If BIC is used to compare models that have the same prior probability then the model with the lowest value of BIC will have the highest posterior probability.

Definition of BIC

General formula for BIC is:

$$\text{BIC} = -2 \times \log(L) + \log(n) \times p$$

- L is the likelihood and p is the total number of estimated parameters – for GLM's this includes the intercept and the scale parameter (if its estimated).

For ordinary regression this formula becomes:

$$\text{BIC} = n + n(\log 2\pi) + n \log(\text{RSS}/n) + \log(n)(k+2)$$

- BIC and AIC are the same except that for BIC the complexity penalty is $\log(n) \times p$ rather than $2p$.
- If $n \geq 8$ then $\log(n) > 2$ and BIC penalizes complexity more heavily than AIC – i.e. BIC is less apt to over fit.

Mallow's Cp

Mallow's Cp statistic is used for ordinary regression and defined as:

$$C_p = \frac{\text{RSS}_M}{\widehat{\sigma}_A^2} - n + 2(k+1)$$

where

- RSS_M is the residual SS for the model being considered,
- $\widehat{\sigma}_A^2$ is the estimate of σ^2 for the model containing all of the regressors.
- The model that minimizes Cp is the same as the model that minimizes AIC.

Mallow's Cp (cont.)

For any subset model that contains all the important regressors

$$\mathrm{E}(\mathrm{RSS}_M) = (n-k-1)\sigma^2 \quad \text{and} \quad \mathrm{E}(\widehat{\sigma}^2_A) = \sigma^2.$$

Substituting into the expression for Cp gives:

$$\mathrm{Cp} \approx \frac{(n-k-1)\sigma^2}{\sigma^2} - n + 2(k+1) = k+1$$

- So any subset model that contains all the important regressors should have Cp$\approx k+1$

Back to the Evaporation Example

For this example, we have a set of 10 candidate regressors. One way to sort through the possible models using R is to use the `regsubsets` function in the `leaps` package.

- The algorithm returns the best model(s) of each size. Note that this set is the same for all of our criteria.
- Options to do an exhaustive search or (if there are too many regressors) a stepwise procedure (forward selection, backward elimination or sequential replacement).

Output from regsubsets

```
> library(leaps)
> subsets.out<-regsubsets(evap~.,data=evap.df,nbest=1)
> sso<-summary(subsets.out)
> sso$outmat
         avst minst maxst avat minat maxat avh minh maxh wind
1  ( 1 ) " "  " "   " "   " "  " "   " "   " " " " "*"  " "
2  ( 1 ) " "  " "   " "   " "  " "   "*"   " " " " "*"  " "
3  ( 1 ) " "  " "   " "   " "  " "   "*"   " " " " "*"  "*"
4  ( 1 ) "*"  " "   "*"   " "  " "   "*"   " " " " "*"  " "
5  ( 1 ) "*"  " "   "*"   " "  " "   "*"   " " "*" "*"  " "
6  ( 1 ) "*"  " "   "*"   " "  " "   "*"   " " "*" "*"  "*"
7  ( 1 ) "*"  " "   "*"   "*"  " "   " "   "*" "*" "*"  "*"
8  ( 1 ) "*"  " "   "*"   "*"  " "   "*"   "*" "*" "*"  "*"
```

- The default is to do an exhaustive search – works for up to 40 or so regressors.

Output for Forward Selection

```
> library(leaps)
> subsetsFS.out<-regsubsets(evap~.,data=evap.df,method="forward")
> ssoFS<-summary(subsetsFS.out)
> ssoFS$outmat
         avst minst maxst avat minat maxat avh minh maxh wind
1  ( 1 ) " "  " "   " "   " "  " "   " "   " " " " "*"  " "
2  ( 1 ) " "  " "   " "   " "  " "   "*"   " " " " "*"  " "
3  ( 1 ) " "  " "   " "   " "  " "   "*"   " " " " "*"  "*"
4  ( 1 ) " "  " "   "*"   " "  " "   "*"   " " " " "*"  "*"
5  ( 1 ) "*"  " "   "*"   " "  " "   "*"   " " " " "*"  "*"
6  ( 1 ) "*"  " "   "*"   " "  " "   "*"   " " "*" "*"  "*"
7  ( 1 ) "*"  " "   "*"   "*"  " "   "*"   " " "*" "*"  "*"
8  ( 1 ) "*"  " "   "*"   "*"  " "   "*"   "*" "*" "*"  "*"
```

- Note that we start getting different models at 4 regressors.

Output for Backward Elimination

```
> library(leaps)
> subsetsBE.out<-regsubsets(evap~.,data=evap.df,method="backward")
> ssoBE<-summary(subsetsBE.out)
> ssoBE$outmat
         avst minst maxst avat minat maxat avh minh maxh wind
1  ( 1 ) " "  " "   " "   " "  " "   " "   " " " " "*"  " "
2  ( 1 ) "*"  " "   " "   " "  " "   " "   " " " " "*"  " "
3  ( 1 ) "*"  " "   " "   " "  " "   " "   " " "*" "*"  " "
4  ( 1 ) "*"  " "   "*"   " "  " "   " "   " " "*" "*"  " "
5  ( 1 ) "*"  " "   "*"   " "  " "   " "   "*" "*" "*"  " "
6  ( 1 ) "*"  " "   "*"   " "  " "   " "   "*" "*" "*"  "*"
7  ( 1 ) "*"  " "   "*"   "*"  " "   " "   "*" "*" "*"  "*"
8  ( 1 ) "*"  " "   "*"   "*"  " "   "*"   "*" "*" "*"  "*"
```

- Again we don't get the same set of models as for an exhaustive search.

Output from `regsubsets`

We can add the values of Cp and BIC to our table of best models for the exhaustive search:

```
> my.table<-cbind(sso$outmat,round(sso$cp,2),round(sso$bic,2))
> colnames(my.table)[11:12]<-c("Cp","BIC")
> print.table( my.table)
         avst minst maxst avat minat maxat avh minh maxh wind Cp    BIC
1  ( 1 )                                          *         30.52 -44.97
2  ( 1 )                          *               *         9.61  -58.6
3  ( 1 )                          *               *    *    6.39  -59.89
4  ( 1 ) *          *             *               *         4.07  -60.78
5  ( 1 ) *          *             *          *    *         3.76  -59.68
6  ( 1 ) *          *             *          *    *    *    4.65  -57.22
7  ( 1 ) *          *     *                *    *    *    *    5.92  -54.32
8  ( 1 ) *          *     *          *     *    *    *    *    7.2   -51.42
```

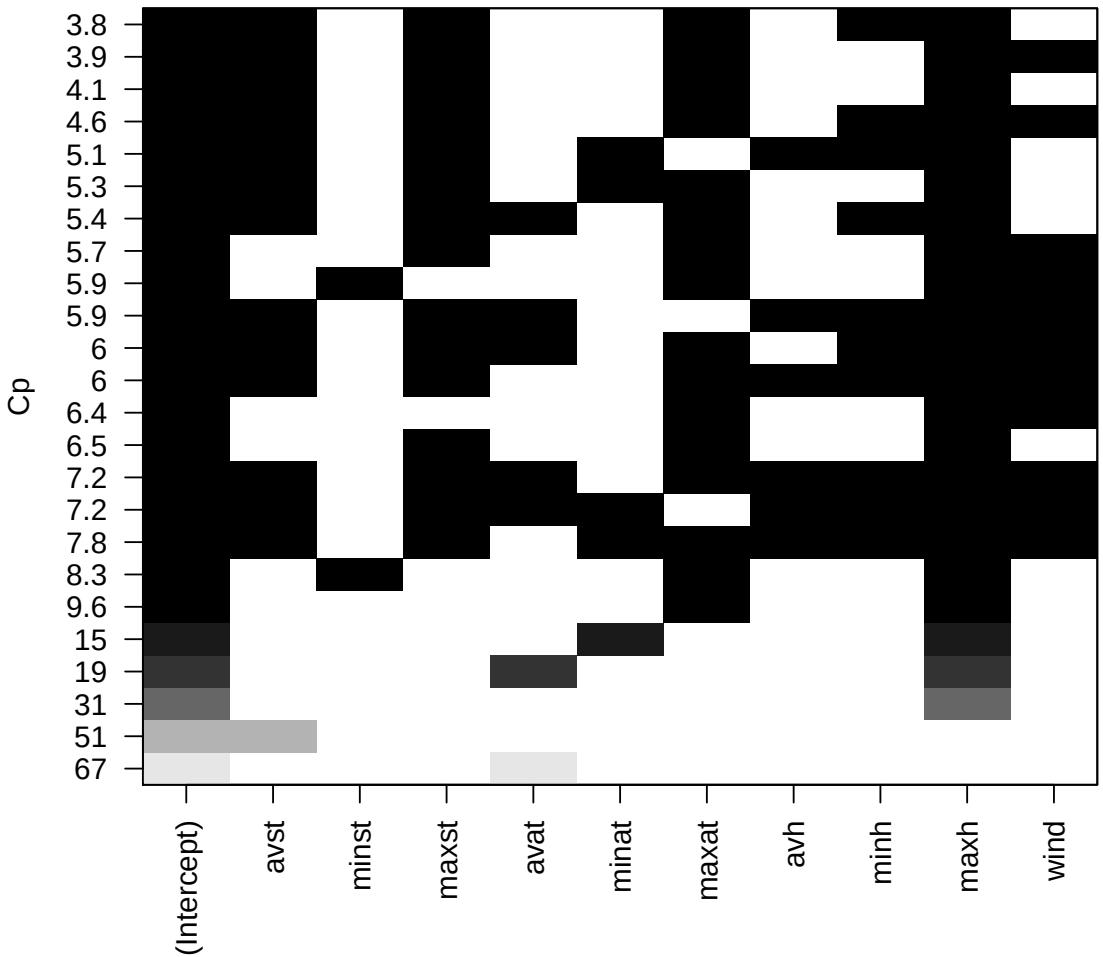
Plotting the Results

Usually, we want more than one model of each size, in which case it may be better to look at a plot of the results.

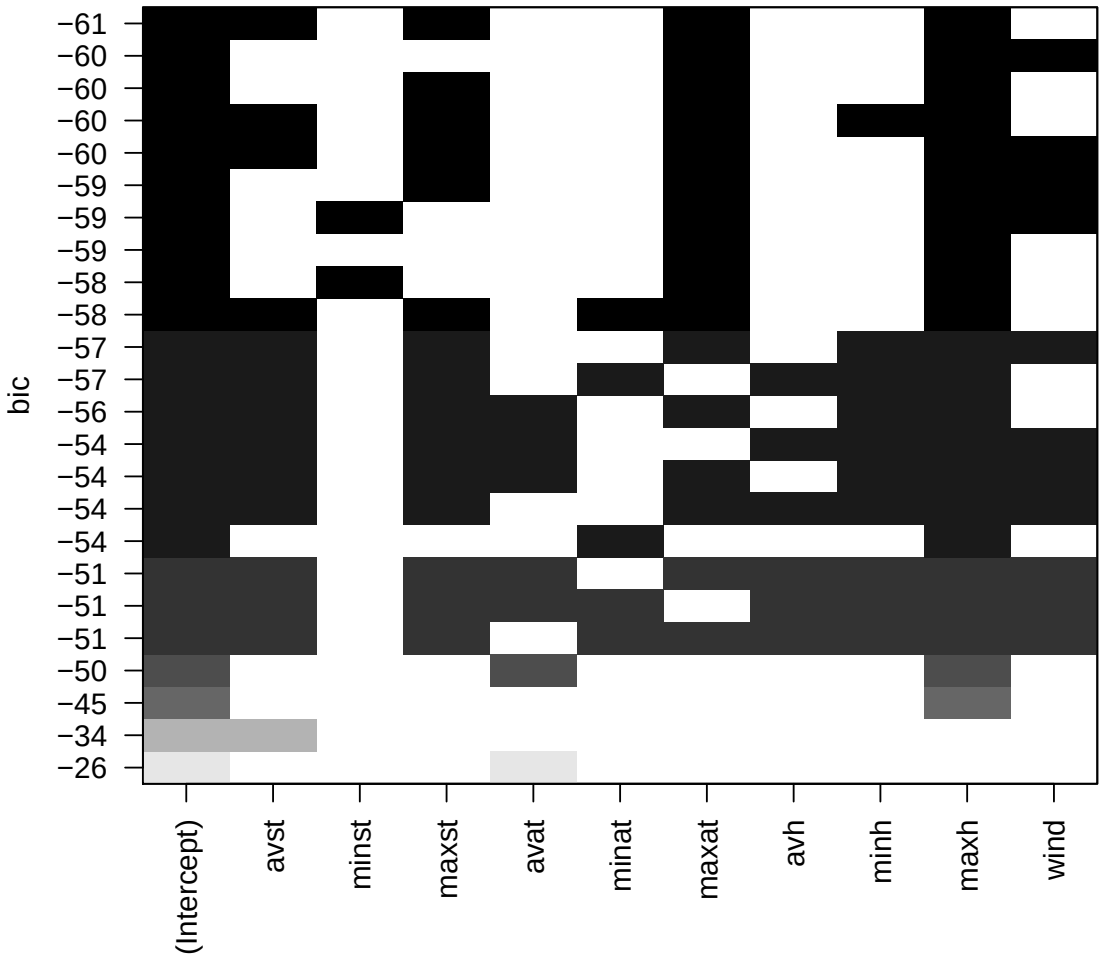
To get a plot of the Cp values for the best 3 models of each size:

```
subsets2.out<-regsubsets(evap~.,data=evap.df,nbest=3)
plot(subsets2.out,scale="Cp")
plot(subsets2.out,scale="bic")
```

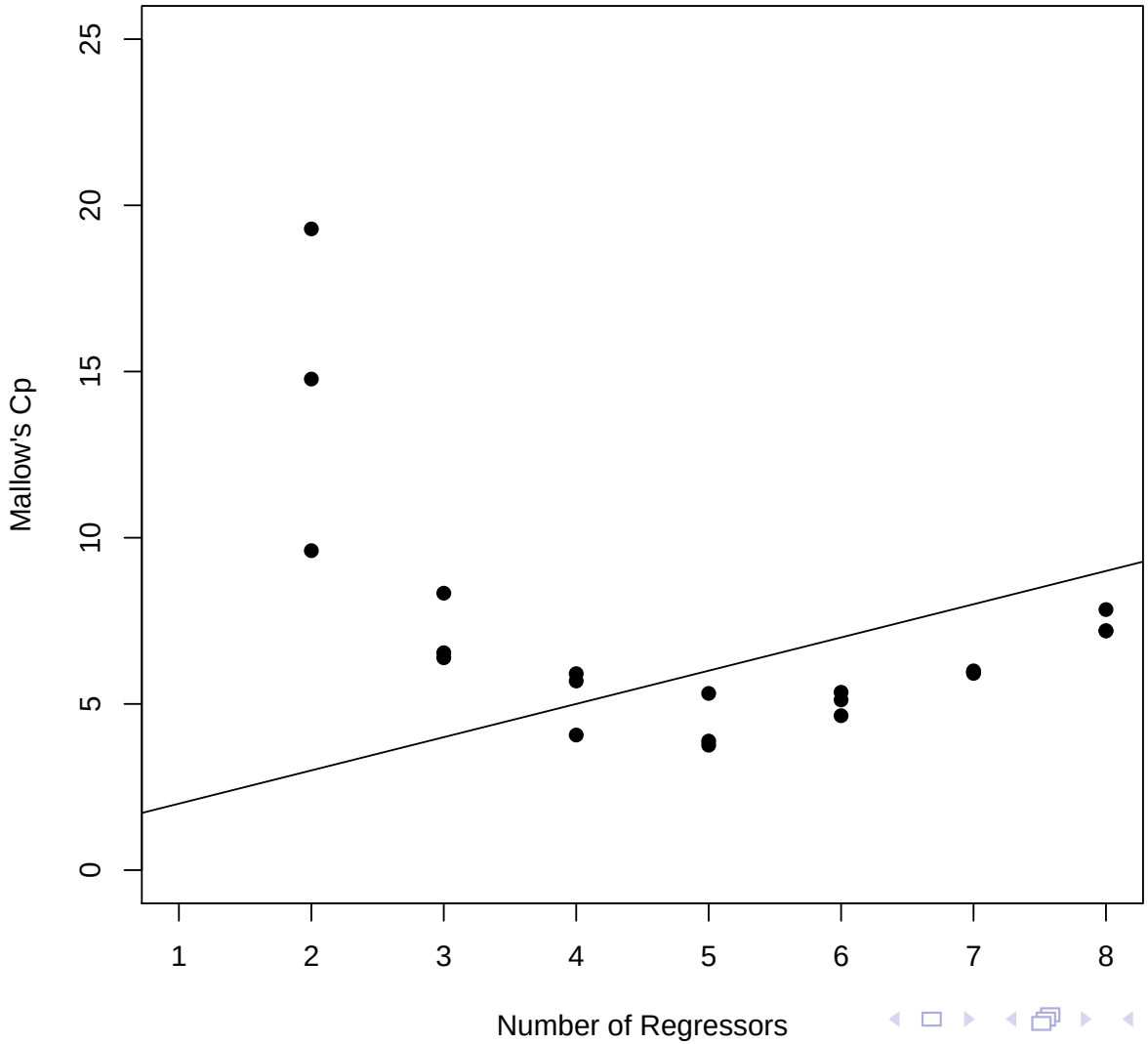
Cp Plot


Cp
3.8
3.9
4.1
4.6
5.1
5.3
5.4
5.7
5.9
5.9
6
6
6.4
6.5
7.2
7.2
7.8
8.3
9.6
15
19
31
51
67
(Intercept)
avst
minst
maxst
avat
minat
maxat
avh
minh
maxh
wind

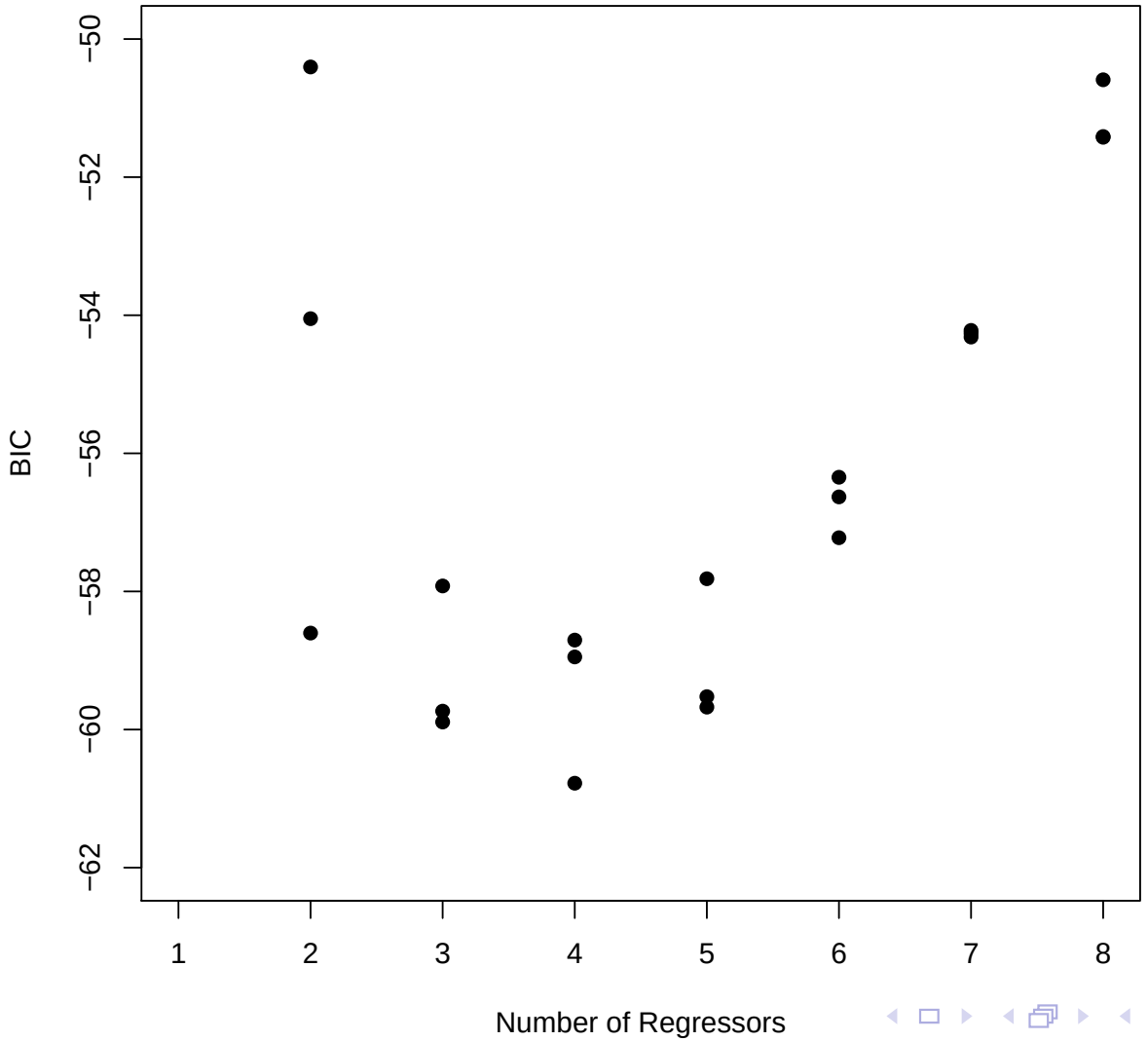
BIC Plot


bic
−61
−60
−60
−60
−60
−59
−59
−59
−58
−58
−57
−57
−56
−54
−54
−54
−54
−51
−51
−51
−50
−45
−34
−26
(Intercept)
avst
minst
maxst
avat
minat
maxat
avh
minh
maxh
wind

Cp versus Number of Regressors



BIC versus Number of Regressors



Best Models according to Cp and BIC

Cp and BIC have suggested a number of models that have similar performance.

		avst	minst	maxst	avat	minat	maxat	avh	minh	maxh	wind	Cp	BIC
3	(1)						*			*	*	6.39	-59.89
3	(2)			*			*			*		6.54	-59.73
3	(3)		*				*			*		8.33	-57.92
4	(1)	*		*			*			*		4.07	-60.78
4	(2)			*			*			*	*	5.69	-58.95
4	(3)		*				*			*	*	5.91	-58.7
5	(1)	*		*			*		*	*		3.76	-59.68
5	(2)	*		*			*			*	*	3.88	-59.52
5	(3)	*		*		*	*			*		5.32	-57.82

AIC and AICc

With a bit of work we can get the values for AIC and AICc

```
ic<-sso2$bic-log(n)*p + 2*p
> aicc<-sso2$bic-log(n)*p + 2*p + (2*p*(p+1))/(n-p-1)
> print.table(cbind(sso2$outmat,round(aic,2),round(aicc,2)))
         avst minst maxst avat minat maxat avh minh maxh wind   aic   aicc
3  ( 1 )                             *                *    *    -69.03 -67.53
3  ( 2 )            *                *                *         -68.88 -67.38
3  ( 3 )      *                      *                *         -67.06 -65.56
4  ( 1 ) *          *                *                *         -71.75 -69.6
4  ( 2 )            *                *                *    *    -69.92 -67.77
4  ( 3 )      *                      *                *    *    -69.68 -67.52
5  ( 1 ) *          *                *           *    *         -72.48 -69.53
5  ( 2 ) *          *                *                *    *    -72.32 -69.38
5  ( 3 ) *          *          *     *                *         -70.62 -67.67
6  ( 1 ) *          *                *           *    *    *    -71.85 -67.96
6  ( 2 ) *          *          *           *    *    *         -71.26 -67.37
6  ( 3 ) *          *     *            *           *    *         -70.97 -67.08
```

Summary

We have identified a number of models with similar performance.

- If the model is for prediction, then we could use cross validation to estimate the prediction error for these models (Thomas).
- It seems that `maxh` and `maxat` show up in all the best models and `maxst` shows up in most of them – suggests that taking the maximum daily readings for humidity, air temperature and soil temperature is the most effective way of accounting for their effects.

Summary (cont.)

The best model with $k = 4$ looks pretty good:

```
> fit2.lm<-lm(evap~avst+maxst+maxat+maxh,data=evap.df)
> summary(fit2.lm)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 60.30530   45.65538   1.321  0.19387
avst         1.29035    0.60287   2.140  0.03832 *
maxst       -0.56355    0.18237  -3.090  0.00359 **
maxat        0.42601    0.09389   4.538  4.9e-05 ***
maxh        -0.30734    0.05160  -5.956  5.0e-07 ***
---
Residual standard error: 6.433 on 41 degrees of freedom
Multiple R-squared:  0.824,Adjusted R-squared:  0.8069
F-statistic:    48 on 4 and 41 DF,  p-value: 6.089e-15
```

Mussels Example

The data for this example comes from: J.J. Sepkoski, Jr., M.A. Rex (1974). "Distribution of Freshwater Mussels: Coastal Rivers as Biogeographic Islands," Systematic Zoology, Vol. 23(2), pp. 165-188.

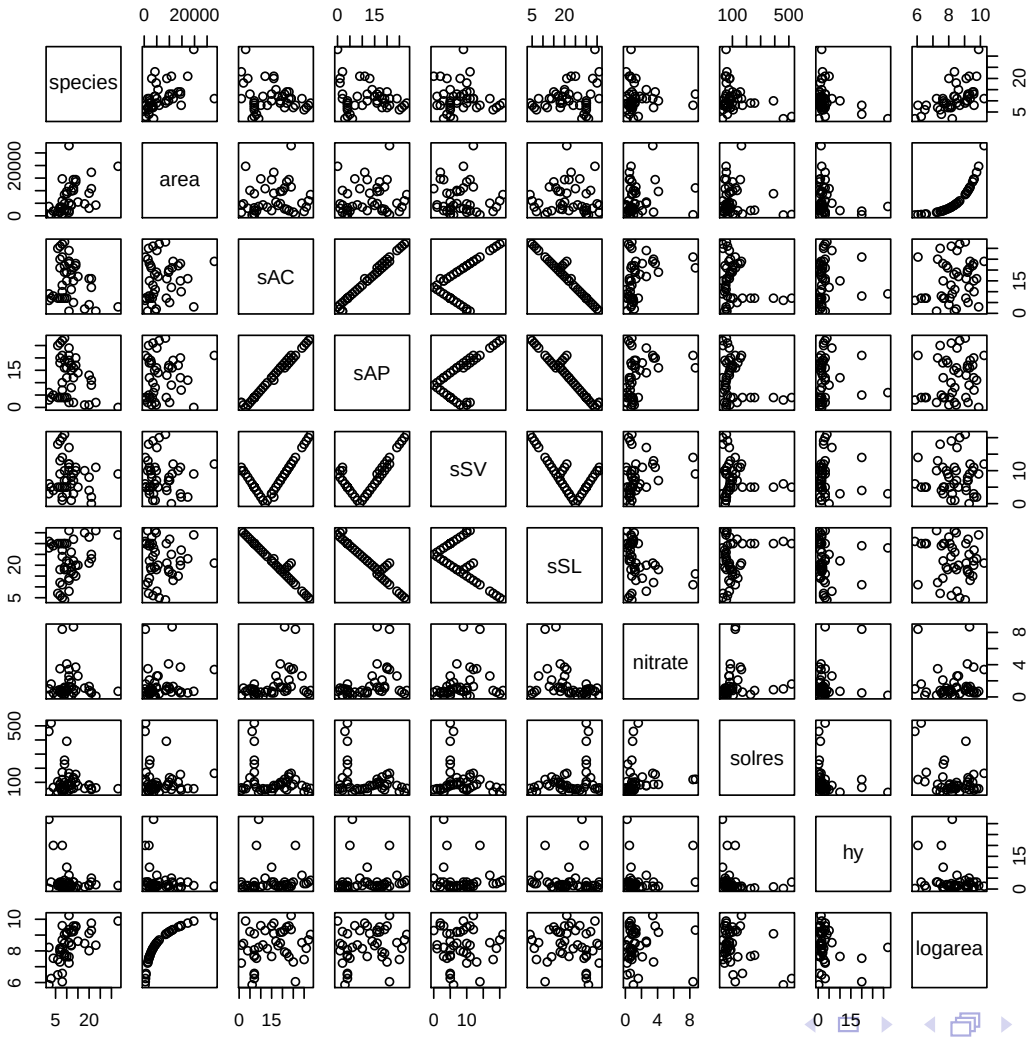
- The number of different mussel species was recorded in 41 rivers along the east coast of the USA.
- Nine explanatory variables were recorded and described as follows: Area, number of stepping stones (intermediate rivers) to 4 major species-source river systems (Alabama-Coosa (AC), Apalachicola (AP), St. Lawrence (SL), and Savannah (SV)), Nitrate Concentration, Hydronium concentration ($10^{(-pH)}$), solid residue.

Mussels Data

The data frame for the mussels data:

```
> head(mussels.df)
             species area sAC sAP sSV sSL nitrate solres   hy logarea
Penobscot          9 8440  33  28  21   4     0.8     57  4.0  9.0407
Kennebec           8 5960  32  27  20   5     0.4     31  3.2  8.6928
Androscoggin       7 3510  31  26  19   6     0.6     65  2.5  8.1634
Saco               6 1730  30  25  18   7     0.8     33  2.5  7.4559
Merrimac          11 5020  29  24  17   8     2.6     78  6.3  8.5212
Blackstone         8  425  26  21  14  11     8.4    120 20.0  6.0521
```

Pairs Plot



A Poisson Model

```
> fit1.glm<-glm(species~.,family="poisson",data=mussels.df)
> summary(fit1.glm)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.238e+00  2.124e+00   0.583  0.55998
area         2.931e-06  2.092e-05   0.140  0.88858
sAC         -2.263e-02  7.845e-02  -0.289  0.77296
sAP         -2.012e-02  5.726e-02  -0.351  0.72526
sSV          1.610e-02  1.155e-02   1.394  0.16335
sSL         -1.059e-02  4.137e-02  -0.256  0.79804
nitrate      5.002e-02  2.834e-02   1.765  0.07756 .
solres      -2.268e-03  7.049e-04  -3.218  0.00129 **
hy          -3.145e-02  1.152e-02  -2.729  0.00636 **
logarea      2.472e-01  1.273e-01   1.941  0.05225 .
---
    Null deviance: 127.527  on 43  degrees of freedom
Residual deviance:  44.218  on 34  degrees of freedom
```

The bestglm Package

The bestglm package allows us to search for best subset models for GLM's.

```
> library(bestglm)
> ?bestglm
bestglm                    package:bestglm                    R Documentation

Best Subset GLM using Information Criterion or Cross-Validation

Description:

     Best subset selection using 'leaps' algorithm (Furnival and
     Wilson, 1974) or complete enumeration (Morgan and Tatar, 1972).
     Complete enumeration is used for the non-Gaussian and for the case
     where the input matrix contains factor variables with more than 2
     levels. The best fit may be found using the information criterion
     IC: AIC, BIC, EBIC, or BICq. Alternatively, with IC='CV' various
     types of cross-validation may be used.
```

The bestglm Package (cont.)

The bestglm function requires the response variable to be in the last column of the data frame (don't ask me why).

```
> mussels2.df<-mussels.df[,c(2:10,1)]
> head(mussels2.df)
             area sAC sAP sSV sSL nitrate solres   hy logarea species
Penobscot    8440  33  28  21   4     0.8     57  4.0  9.0407       9
Kennebec     5960  32  27  20   5     0.4     31  3.2  8.6928       8
Androscoggin 3510  31  26  19   6     0.6     65  2.5  8.1634       7
Saco         1730  30  25  18   7     0.8     33  2.5  7.4559       6
Merrimac     5020  29  24  17   8     2.6     78  6.3  8.5212      11
Blackstone    425  26  21  14  11     8.4    120 20.0  6.0521       8
```

The bestglm Function

Now we can use bestglm to search for the best subset models (using AIC):

```
> bestglm.out<-bestglm(mussels2.df,family = poisson,IC = "AIC",
                                    TopModels = 5,method = "exhaustive")
Morgan-Tatar search since family is non-gaussian.
> bestglm.out$BestModels
   area   sAC   sAP   sSV   sSL nitrate solres   hy logarea Criterion
1 FALSE FALSE  TRUE  TRUE FALSE    TRUE   TRUE TRUE    TRUE  238.3312
2 FALSE  TRUE FALSE FALSE FALSE    TRUE   TRUE TRUE    TRUE  238.4542
3 FALSE FALSE  TRUE FALSE FALSE    TRUE   TRUE TRUE    TRUE  238.5282
4 FALSE  TRUE FALSE  TRUE FALSE    TRUE   TRUE TRUE    TRUE  238.6503
5 FALSE FALSE  TRUE  TRUE FALSE   FALSE   TRUE TRUE    TRUE  239.2076
```

The bestglm Function

Now we can use bestglm to search for the best subset models (using AIC):

```
> bestglm.out<-bestglm(mussels2.df,family = poisson,IC = "AIC",
                                    TopModels = 5,method = "exhaustive")
Morgan-Tatar search since family is non-gaussian.
> bestglm.out$BestModels
   area   sAC   sAP   sSV   sSL nitrate solres   hy logarea Criterion
1 FALSE FALSE  TRUE  TRUE FALSE    TRUE   TRUE TRUE    TRUE  238.3312
2 FALSE  TRUE FALSE FALSE FALSE    TRUE   TRUE TRUE    TRUE  238.4542
3 FALSE FALSE  TRUE FALSE FALSE    TRUE   TRUE TRUE    TRUE  238.5282
4 FALSE  TRUE FALSE  TRUE FALSE    TRUE   TRUE TRUE    TRUE  238.6503
5 FALSE FALSE  TRUE  TRUE FALSE   FALSE   TRUE TRUE    TRUE  239.2076
```

The bestglm Function (cont.)

Or we can try using BIC:

```
> bestglm.out2<-bestglm(mussels2.df,family = poisson,IC = "BIC",
                                  TopModels = 5,method = "exhaustive")
Morgan-Tatar search since family is non-gaussian.
> bestglm.out2$BestModels
   area   sAC   sAP   sSV   sSL nitrate solres   hy logarea Criterion
1 FALSE  TRUE FALSE FALSE FALSE   FALSE   TRUE TRUE    TRUE  246.6250
2 FALSE FALSE  TRUE FALSE FALSE   FALSE   TRUE TRUE    TRUE  246.6725
3 FALSE  TRUE FALSE FALSE FALSE    TRUE   TRUE TRUE    TRUE  247.3752
4 FALSE FALSE  TRUE FALSE FALSE    TRUE   TRUE TRUE    TRUE  247.4492
5 FALSE FALSE FALSE FALSE  TRUE   FALSE   TRUE TRUE    TRUE  247.5870
```

Summary

For this example BIC is selecting smaller models than AIC (it would have been nice to look at AICc).

- `solres`, `hy`, and `logarea` are showing up in all the best models.
- `nitrate` shows up in most models.
- `area` doesn't show up in any.
- seem to need at most 1 or 2 of the stepping stone variables.

A Good Submodel

```
> fit2.glm<-glm(species~nitrate+solres+hy+logarea+sAC,
                      family = poisson, data=mussels.df)
> summary(fit2.glm)
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.9483756  0.4701266   2.017 0.043667 *
nitrate      0.0514423  0.0285902   1.799 0.071971 .
solres      -0.0022974  0.0006491  -3.540 0.000401 ***
hy          -0.0329310  0.0112529  -2.926 0.003429 **
logarea      0.2495191  0.0513711   4.857 1.19e-06 ***
sAC         -0.0258976  0.0056666  -4.570 4.87e-06 ***
---
    Null deviance: 127.527  on 43  degrees of freedom
Residual deviance:  46.426  on 38  degrees of freedom
AIC: 240.45
```

GOOD LUCK!

Thanks for being a great class!

Good luck with the Test!