**Section 5.1 – Variable selection (continued)**

Stepwise search algorithms

These methods provide another way to find the “best” model from all possible models. The advantage these have over the previous methods is that they are simple and generally faster. The disadvantage is that they are not as good at finding the “best” model.

Why discuss these methods then?

* There may be instances when the previous methods cannot be implemented easily. This includes situations where models are much more complicated than those we discuss in our class.
* These methods are still prevalently used. You may need to read a paper which uses them, so it is good to understand how they are implemented.

There are three ways to implement these selection methods:

1. Forward selection
	1. Compute IC(k) for a model with no explanatory variables
	2. Compute IC(k) for all possible one explanatory variable models. Find the model that reduces the IC(k) value the most. If no model reduces IC(k) compared to the model in a), then use the model in a) as your “best” model.
	3. Compute IC(k) for all possible two explanatory variable models, where the model from step b) is used to start from. Find the model that reduces the IC(k) value the most. If no model reduces IC(k) compared to the model in b), then use the model in b) as your “best” model.
	4. Continue adding explanatory variables one at a time until no additional variable decreases the IC(K).
2. Backward selection – What do you think the steps are?
3. Alternating stepwise selection – What do you think the steps are?

Interactions are sometimes included in this process as well. However, one needs to make sure that the main effects corresponding to an interaction are still in the model.

Example: Placekicking (stepwisePlacekick.R, Placekick.csv)

Below is forward selection implemented by the step() function using BIC:

> empty.mod <- glm(formula = good ~ 1, family = binomial(link = logit), data = placekick)

> full.mod <- glm(formula = good ~ ., family = binomial(link = logit), data = placekick)

> forw.sel <- step(object = empty.mod, scope = list(upper = full.mod), direction = "forward", k = log(nrow(placekick)), trace = TRUE)

Start: AIC=1020.69

good ~ 1

 Df Deviance AIC

+ distance 1 775.75 790.27

+ PAT 1 834.41 848.93

+ change 1 989.15 1003.67

<none> 1013.43 1020.69

+ elap30 1 1007.71 1022.23

+ wind 1 1010.59 1025.11

+ week 1 1011.24 1025.76

+ type 1 1011.39 1025.92

+ field 1 1012.98 1027.50

Step: AIC=790.27

good ~ distance

 Df Deviance AIC

+ PAT 1 762.41 784.20

<none> 775.75 790.27

+ change 1 770.50 792.29

+ wind 1 772.53 794.32

+ week 1 773.86 795.64

+ type 1 775.67 797.45

+ elap30 1 775.68 797.47

+ field 1 775.74 797.53

Step: AIC=784.2

good ~ distance + PAT

 Df Deviance AIC

<none> 762.41 784.20

+ change 1 759.33 788.38

+ wind 1 759.66 788.71

+ week 1 760.57 789.62

+ type 1 762.25 791.30

+ elap30 1 762.31 791.36

+ field 1 762.41 791.46

> anova(forw.sel)

Analysis of Deviance Table

Model: binomial, link: logit

Response: good

Terms added sequentially (first to last)

 Df Deviance Resid. Df Resid. Dev

NULL 1424 1013.43

distance 1 237.681 1423 775.75

PAT 1 13.335 1422 762.41

While “AIC” is listed in the output, it is really BIC that is calculated because of what was specified in the k argument.

Below is forward selection implemented by the step() function using AIC:

> forw.sel2 <- step(object = empty.mod, scope = list(upper = full.mod), direction = "forward", k = 2, trace = TRUE)

Start: AIC=1015.43

good ~ 1

 Df Deviance AIC

+ distance 1 775.75 779.75

+ PAT 1 834.41 838.41

+ change 1 989.15 993.15

+ elap30 1 1007.71 1011.71

+ wind 1 1010.59 1014.59

+ week 1 1011.24 1015.24

+ type 1 1011.39 1015.39

<none> 1013.43 1015.43

+ field 1 1012.98 1016.98

Step: AIC=779.75

good ~ distance

 Df Deviance AIC

+ PAT 1 762.41 768.41

+ change 1 770.50 776.50

+ wind 1 772.53 778.53

<none> 775.75 779.75

+ week 1 773.86 779.86

+ type 1 775.67 781.67

+ elap30 1 775.68 781.68

+ field 1 775.74 781.74

Step: AIC=768.41

good ~ distance + PAT

 Df Deviance AIC

+ change 1 759.33 767.33

+ wind 1 759.66 767.66

<none> 762.41 768.41

+ week 1 760.57 768.57

+ type 1 762.25 770.25

+ elap30 1 762.31 770.31

+ field 1 762.41 770.41

Step: AIC=767.33

good ~ distance + PAT + change

 Df Deviance AIC

+ wind 1 756.69 766.69

+ week 1 757.26 767.26

<none> 759.33 767.33

+ elap30 1 759.11 769.11

+ type 1 759.13 769.13

+ field 1 759.33 769.33

Step: AIC=766.69

good ~ distance + PAT + change + wind

 Df Deviance AIC

<none> 756.69 766.69

+ week 1 755.07 767.07

+ type 1 756.06 768.06

+ elap30 1 756.43 768.43

+ field 1 756.66 768.66

> anova(forw.sel2)

Analysis of Deviance Table

Model: binomial, link: logit

Response: good

Terms added sequentially (first to last)

 Df Deviance Resid. Df Resid. Dev

NULL 1424 1013.43

distance 1 237.681 1423 775.75

PAT 1 13.335 1422 762.41

change 1 3.077 1421 759.33

wind 1 2.646 1420 756.69

We obtain a different set of variables! Remember that the BIC will generally favor smaller models than the AIC.

Comments:

* Investigate backward ("backward") and alternating stepwise ("both") on your own. Note that the three stepwise selection methods will not always obtain the same “best” models.
* While AICc is not implemented in step(), why would we generally expect the same model with this data set as with using AIC?
* Alternatively to using step(), one could use the AIC() function and estimate each model in the stepwise process.
* Interactions can be included in the process by specifying a full model that includes them.