Stepwise Logistic Regression and log-linear models with R

Akaike information criterion: AIC = 2k - 2 log L
= 2k + Deviance, where k = number of parameters

Small numbers are better
Penalizes models with lots of parameters
Penalizes models with poor fit

```
> fullmod = glm(low ~ age+lwt+racefac+smoke+ptl+ht+ui+ftv,family=binomial)
> summary(fullmod)

Call:
glm(formula = low ~ age + lwt + racefac + smoke + ptl + ht +
    ui + ftv, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8946  -0.8212  -0.5316   0.9818   2.2125

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.480623   1.196888   0.402  0.68801
age          -0.029549   0.037031  -0.798  0.42489
lwt          -0.015424   0.006919  -2.229  0.02580 *
racefacBlack  1.272260   0.527357   2.413  0.01584 *
racefacOther  0.880496   0.440778   1.998  0.04576 *
smoke         0.938846   0.402147   2.335  0.01957 *
ptl           0.543337   0.345403   1.573  0.11571
ht            1.863303   0.697533   2.671  0.00756 **
ui            0.767648   0.459318   1.671  0.09467 .
ftv           0.065302   0.172394   0.379  0.70484
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 201.28  on 179  degrees of freedom
AIC: 221.28

Number of Fisher Scoring iterations: 4

> nothing <- glm(low ~ 1,family=binomial)
> summary(nothing)

Call:
glm(formula = low ~ 1, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8651  -0.8651  -0.8651   1.5259   1.5259

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.790      0.157  -5.033 4.84e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 234.67  on 188  degrees of freedom
AIC: 236.67

Number of Fisher Scoring iterations: 4

> # Here was the chosen model from earlier
> redmod1 = glm(low ~ lwt+racefac+smoke+ptl+ht,family=binomial)
>
> backwards = step(fullmod) # Backwards selection is the default
Start:  AIC= 221.28
 low ~ age + lwt + racefac + smoke + ptl + ht + ui + ftv

          Df Deviance    AIC
- ftv      1   201.43 219.43
- age      1   201.93 219.93
<none>         201.28 221.28
- ptl      1   203.83 221.83
- ui       1   204.03 222.03
- racefac  2   208.75 224.75
- lwt      1   206.80 224.80
- smoke    1   206.91 224.91
- ht       1   208.81 226.81

Step:  AIC= 219.43
 low ~ age + lwt + racefac + smoke + ptl + ht + ui

          Df Deviance    AIC
- age      1   201.99 217.99
<none>         201.43 219.43
- ptl      1   203.95 219.95
- ui       1   204.11 220.11
- racefac  2   208.77 222.77
- lwt      1   206.81 222.81
- smoke    1   206.92 222.92
- ht       1   208.81 224.81

Step:  AIC= 217.99
 low ~ lwt + racefac + smoke + ptl + ht + ui

          Df Deviance    AIC
<none>         201.99 217.99
- ptl      1   204.22 218.22
- ui       1   204.90 218.90
- smoke    1   207.73 221.73
- lwt      1   208.11 222.11
- racefac  2   210.31 222.31
- ht       1   209.46 223.46

> 217.99-201.99
[1] 16

> #  backwards = step(fullmod,trace=0) would suppress step by step output.
> formula(backwards)
low ~ lwt + racefac + smoke + ptl + ht + ui
```

```
> summary(backwards)

Call:
glm(formula = low ~ lwt + racefac + smoke + ptl + ht + ui, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9049  -0.8124  -0.5241   0.9483   2.1812

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.086550   0.951760  -0.091  0.92754
lwt          -0.015905   0.006855  -2.320  0.02033 *
racefacBlack  1.325719   0.522243   2.539  0.01113 *
racefacOther  0.897078   0.433881   2.068  0.03868 *
smoke         0.938727   0.398717   2.354  0.01855 *
ptl           0.503215   0.341231   1.475  0.14029
ht            1.855042   0.695118   2.669  0.00762 **
ui            0.785698   0.456441   1.721  0.08519 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 201.99  on 181  degrees of freedom
AIC: 217.99

Number of Fisher Scoring iterations: 4

> # I would be inclined to drop ptl
> back2 = glm(low ~ lwt + racefac + smoke +  ht + ui,family=binomial)
> summary(back2)

Call:
glm(formula = low ~ lwt + racefac + smoke + ht + ui, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7396  -0.8322  -0.5359   0.9873   2.1692

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.056276   0.937853   0.060  0.95215
lwt          -0.016732   0.006803  -2.459  0.01392 *
racefacBlack  1.324562   0.521464   2.540  0.01108 *
racefacOther  0.926197   0.430386   2.152  0.03140 *
smoke         1.035831   0.392558   2.639  0.00832 **
ht            1.871416   0.690902   2.709  0.00676 **
ui            0.904974   0.447553   2.022  0.04317 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 204.22  on 182  degrees of freedom
AIC: 218.22

Number of Fisher Scoring iterations: 4
```

```
> redmod1$deviance; back2$deviance
[1] 204.8977
[1] 204.2166
> # back2 may be slightly "better," but I like redmod1 more.
> # Why? Because ptl is easier to assess than ui
>
> forwards = step(nothing,
scope=list(lower=formula(nothing),upper=formula(fullmod)), direction="forward")
Start:  AIC= 236.67
 low ~ 1

          Df Deviance    AIC
+ ptl      1   227.89 231.89
+ lwt      1   228.69 232.69
+ ui       1   229.60 233.60
+ smoke    1   229.81 233.81
+ ht       1   230.65 234.65
+ racefac  2   229.66 235.66
+ age      1   231.91 235.91
<none>         234.67 236.67
+ ftv      1   233.90 237.90

Step:  AIC= 231.89
 low ~ ptl

          Df Deviance    AIC
+ lwt      1   223.41 229.41
+ ht       1   223.58 229.58
+ age      1   224.27 230.27
+ racefac  2   222.53 230.53
+ smoke    1   224.78 230.78
+ ui       1   224.89 230.89
<none>         227.89 231.89
+ ftv      1   227.30 233.30

Step:  AIC= 229.41
 low ~ ptl + lwt

          Df Deviance    AIC
+ ht       1   215.96 223.96
+ racefac  2   217.68 227.68
+ smoke    1   220.54 228.54
+ age      1   221.05 229.05
+ ui       1   221.23 229.23
<none>         223.41 229.41
+ ftv      1   223.12 231.12

Step:  AIC= 223.96
 low ~ ptl + lwt + ht

          Df Deviance    AIC
+ racefac  2   210.85 222.85
+ ui       1   213.01 223.01
+ smoke    1   213.15 223.15
<none>         215.96 223.96
+ age      1   214.01 224.01
+ ftv      1   215.84 225.84

Step:  AIC= 222.85
```

```
 low ~ ptl + lwt + ht + racefac

        Df Deviance    AIC
+ smoke  1   204.90 218.90
+ ui     1   207.73 221.73
<none>       210.85 222.85
+ age    1   209.81 223.81
+ ftv    1   210.82 224.82

Step:  AIC= 218.9
 low ~ ptl + lwt + ht + racefac + smoke

       Df Deviance    AIC
+ ui    1   201.99 217.99
<none>      204.90 218.90
+ age   1   204.11 220.11
+ ftv   1   204.88 220.88

Step:  AIC= 217.99
 low ~ ptl + lwt + ht + racefac + smoke + ui

       Df Deviance    AIC
<none>      201.99 217.99
+ age   1   201.43 219.43
+ ftv   1   201.93 219.93

> formula(redmod1)
low ~ lwt + racefac + smoke + ptl + ht
> formula(backwards)
low ~ lwt + racefac + smoke + ptl + ht + ui
> formula(forwards)
low ~ ptl + lwt + ht + racefac + smoke + ui
> bothways =
+ step(nothing, list(lower=formula(nothing),upper=formula(fullmod)),
direction="both",trace=0)
> formula(forwards)
low ~ ptl + lwt + ht + racefac + smoke + ui
> formula(bothways)
low ~ ptl + lwt + ht + racefac + smoke + ui
```

Stepwise selection of log-linear Models

The R help says the step function will fork for any formula-based method for specifying models. Loglin is not formula based, but there is a package that puts a formula-based front end on loglin. In the Packages and Data menu, select MASS (Venable and Ripley's **M**ethods of **A**pplied **S**tatistics with **S**).

```
>
> # Remember the detergent data
> soapdata <- read.table("DetergentFrame.txt"); soapdata
   Softness Prev_Use   Temp Pref Freq
1    1=Soft    1=Yes 1=High  1=X   19
2    1=Soft    1=Yes 1=High  2=M   29
3    1=Soft    1=Yes  2=Low  1=X   57
4    1=Soft    1=Yes  2=Low  2=M   49
5    1=Soft     2=No 1=High  1=X   29
6    1=Soft     2=No 1=High  2=M   27
7    1=Soft     2=No  2=Low  1=X   63
8    1=Soft     2=No  2=Low  2=M   53
9    2=Medm    1=Yes 1=High  1=X   23
10   2=Medm    1=Yes 1=High  2=M   47
11   2=Medm    1=Yes  2=Low  1=X   47
12   2=Medm    1=Yes  2=Low  2=M   55
13   2=Medm     2=No 1=High  1=X   33
14   2=Medm     2=No 1=High  2=M   23
15   2=Medm     2=No  2=Low  1=X   66
16   2=Medm     2=No  2=Low  2=M   50
17   3=Hard    1=Yes 1=High  1=X   24
18   3=Hard    1=Yes 1=High  2=M   43
19   3=Hard    1=Yes  2=Low  1=X   37
20   3=Hard    1=Yes  2=Low  2=M   52
21   3=Hard     2=No 1=High  1=X   42
22   3=Hard     2=No 1=High  2=M   30
23   3=Hard     2=No  2=Low  1=X   68
24   3=Hard     2=No  2=Low  2=M   42
> soap <- xtabs(Freq ~ Softness+Prev_Use+Temp+Pref, data=soapdata)
> ind1 = loglin(soap,list(1,2,3,4));  ind1$lrt
2 iterations: deviation 1.136868e-13
[1] 42.92866
> ind2 = loglm(Freq ~ Softness+Prev_Use+Temp+Pref, data=soapdata)
> # Additive model: No interactions = complete independence
> ind2$lrt
[1] 42.92866
> ind2
Call:
loglm(formula = Freq ~ Softness + Prev_Use + Temp + Pref, data = soapdata)

Statistics:
                      X^2 df     P(> X^2)
Likelihood Ratio 42.92866 18 0.0008190181
Pearson          43.90225 18 0.0005957483

> # Exploration yielded this: [Softness Temp] [Prev_Use Pref] [Temp Pref]
> ModelC = loglin(soap,list(c(1,3),c(2,4),c(3,4))) ; ModelC$lrt
2 iterations: deviation 5.684342e-14
[1] 11.88649
> ModelC2 = loglm(Freq ~ Softness*Temp + Prev_Use*Pref + Temp*Pref, data=soapdata)
> ModelC2$lrt
[1] 11.88649
```

```
>
> # Try backwards selection. Model fullsoap is saturated.
> fullsoap = loglm(Freq ~ Softness*Prev_Use*Temp*Pref, data=soapdata)
> fullsoap
Call:
loglm(formula = Freq ~ Softness * Prev_Use * Temp * Pref, data = soapdata)

Statistics:
                 X^2 df P(> X^2)
Likelihood Ratio   0  0        1
Pearson            0  0        1


> backloglin = step(fullsoap)
Start:  AIC= 48
 Freq ~ Softness * Prev_Use * Temp * Pref

                              Df    AIC
- Softness:Prev_Use:Temp:Pref  2 44.737
<none>                           48.000

Step:  AIC= 44.74
 Freq ~ Softness + Prev_Use + Temp + Pref + Softness:Prev_Use +
    Softness:Temp + Prev_Use:Temp + Softness:Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Prev_Use:Temp + Softness:Prev_Use:Pref +
    Softness:Temp:Pref + Prev_Use:Temp:Pref

                         Df    AIC
- Softness:Temp:Pref      2 40.899
- Softness:Prev_Use:Temp  2 42.115
<none>                      44.737
- Prev_Use:Temp:Pref      1 44.959
- Softness:Prev_Use:Pref  2 45.309

Step:  AIC= 40.9
 Freq ~ Softness + Prev_Use + Temp + Pref + Softness:Prev_Use +
    Softness:Temp + Prev_Use:Temp + Softness:Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Prev_Use:Temp + Softness:Prev_Use:Pref +
    Prev_Use:Temp:Pref

                         Df    AIC
- Softness:Prev_Use:Temp  2 38.251
<none>                      40.899
- Prev_Use:Temp:Pref      1 41.115
- Softness:Prev_Use:Pref  2 41.495

Step:  AIC= 38.25
 Freq ~ Softness + Prev_Use + Temp + Pref + Softness:Prev_Use +
    Softness:Temp + Prev_Use:Temp + Softness:Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Prev_Use:Pref + Prev_Use:Temp:Pref

                         Df    AIC
<none>                      38.251
- Prev_Use:Temp:Pref      1 38.518
- Softness:Prev_Use:Pref  2 39.059
- Softness:Temp           2 39.816
>
> # Yields: [Softness Temp] [Softness Prev_Use Pref] [Prev_Use Temp Pref]
> # Besides rel betw softness and temp, rels between prev use & pref
> # depend on both softness and temp.
```

```
>
> # Forward selection
> forloglin = step(ind2, scope=list(lower=formula(ind2),upper=formula(fullsoap)),
direction="forward")
Start:  AIC= 54.93
 Freq ~ Softness + Prev_Use + Temp + Pref

                    Df    AIC
+ Prev_Use:Pref      1 36.347
+ Temp:Pref          1 52.567
+ Softness:Temp      2 52.830
<none>                 54.929
+ Prev_Use:Temp      1 55.676
+ Softness:Prev_Use  2 57.854
+ Softness:Pref      2 58.533

Step:  AIC= 36.35
 Freq ~ Softness + Prev_Use + Temp + Pref + Prev_Use:Pref

                    Df    AIC
+ Temp:Pref          1 33.986
+ Softness:Temp      2 34.248
<none>                 36.347
+ Prev_Use:Temp      1 37.094
+ Softness:Prev_Use  2 39.272
+ Softness:Pref      2 39.952

Step:  AIC= 33.99
 Freq ~ Softness + Prev_Use + Temp + Pref + Prev_Use:Pref + Temp:Pref

                    Df    AIC
+ Softness:Temp      2 31.886
<none>                 33.986
+ Prev_Use:Temp      1 35.294
+ Softness:Prev_Use  2 36.910
+ Softness:Pref      2 37.590

Step:  AIC= 31.89
 Freq ~ Softness + Prev_Use + Temp + Pref + Prev_Use:Pref + Temp:Pref +
    Softness:Temp

                    Df    AIC
<none>                 31.886
+ Prev_Use:Temp      1 33.195
+ Softness:Prev_Use  2 34.798
+ Softness:Pref      2 35.543
>
>
> # Same as redmod1, which was based on (manual) forward selection
> # Forloglin=redmod1 happens to be nested within backloglin: Testable
> anova(forloglin,backloglin)
LR tests for hierarchical log-linear models

Model 1:
 Freq ~ Prev_Use + Softness + Temp
Model 2:
 Freq ~ Prev_Use + Softness + Temp

           Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1   11.886487 14
Model 2    2.250554  6   9.635933         8        0.29151
Saturated  0.000000  0   2.250554         6        0.89527
> forloglin$lrt
[1] 11.88649
> forloglin$lrt-backloglin$lrt
[1] 9.635933
> # This says backloglin is not an improvement over forloglin, and
> # we already knew forloglin=redmod1 fits.
>
>
> bothloglin = step(ind2, scope=list(lower=formula(ind2),upper=formula(fullsoap)),
direction="both",trace=0)
> # formula(bothloglin) generates a pile of output (?!)
> bothloglin
Call:
loglm(formula = Freq ~ Softness + Prev_Use + Temp + Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Temp, data = soapdata, evaluate = FALSE)

Statistics:
                      X^2 df  P(> X^2)
Likelihood Ratio 11.88649 14 0.6154184
Pearson          11.91780 14 0.6129043
> forloglin
Call:
loglm(formula = Freq ~ Softness + Prev_Use + Temp + Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Temp, data = soapdata, evaluate = FALSE)

Statistics:
                      X^2 df  P(> X^2)
Likelihood Ratio 11.88649 14 0.6154184
Pearson          11.91780 14 0.6129043
>
>
```

```
> # Finally, for stepwise selection of a conditional model, start with
> # a model that has all interactions among explanatory variables, and
> # the explanatory variables are independent of the response variable(s).
> ind3 = loglm(Freq ~ Softness*Prev_Use*Temp + Pref, data=soapdata)
> forcond = step(ind3, scope=list(lower=formula(ind3),upper=formula(fullsoap)),
direction="both",trace=0); forcond
Call:
loglm(formula = Freq ~ Softness + Prev_Use + Temp + Pref + Softness:Prev_Use +
    Softness:Temp + Prev_Use:Temp + Prev_Use:Pref + Temp:Pref +
    Softness:Prev_Use:Temp + Prev_Use:Temp:Pref, data = soapdata,
    evaluate = FALSE)

Statistics:
                      X^2 df  P(> X^2)
Likelihood Ratio 5.656044  8 0.6856970
Pearson          5.649976  8 0.6863733
>
> backcond = step(fullsoap,
scope=list(lower=formula(ind3),upper=formula(fullsoap)),
direction="backward",trace=0); backcond
Call:
loglm(formula = Freq ~ Softness + Prev_Use + Temp + Pref + Softness:Prev_Use +
    Softness:Temp + Prev_Use:Temp + Softness:Pref + Prev_Use:Pref +
    Temp:Pref + Softness:Prev_Use:Temp + Softness:Prev_Use:Pref +
    Prev_Use:Temp:Pref, data = soapdata, evaluate = FALSE)

Statistics:
                       X^2 df  P(> X^2)
Likelihood Ratio 0.8991362  4 0.9246847
Pearson          0.9007636  4 0.9244512
```