Sharpening Occam's Razor: Using Bayesian Model Averaging in **R** to Separate the Wheat from the Chaff.

As published in Benchmarks RSS Matters, February 2011

http://web3.unt.edu/benchmarks/issues/2011/2/rss-matters

Jon Starkweather, PhD

Jon Starkweather, PhD
jonathan.starkweather@unt.edu
Consultant
Research and **S**tatistical **S**upport

http://www.unt.edu

http://www.unt.edu/rss

RSS hosts a number of "Short Courses".
A list of them is available at:
http://www.unt.edu/rss/Instructional.htm

The programming scripts contained in this article can also be found at:
http://www.unt.edu/rss/class/Jon/R_SC

Sharpening Occam's Razor: Using Bayesian Model Averaging in **R** to Separate the Wheat from the Chaff.

Bayesian Model Averaging (BMA) is a method of variable selection which quantifies the value of multiple models so that the analyst can select the most appropriate model for a given outcome variable. The metrics used for comparison of competing models are the Bayesian Information Criterion (BIC; Schwarz, 1978) and the posterior probability (of a particular model being the correct model). The best model, displays the lowest BIC (e.g. a BIC of -121.00 is preferred over a BIC of 21.00) and the highest posterior probability. In the simplest situation (linear regression), each model is characterized by a group of predictors for the outcome variable. When BMA is applied to all available predictors, and given an outcome variable of interest, it produces a posterior distribution of the outcome variable which is a weighted average of the posterior distributions of the outcome for each likely model (Raftery, Painter, & Volinsky, 2005[1]). Essentially, BMA is used to determine which predictors should be included in a regression model or general linear model (GLM), or extensions of the GLM (e.g. generalized linear models and survival or event history analysis). BMA is particularly useful when a large number of proposed predictors have been measured (e.g. 20, 30, or 40).

BMA is accomplished in the R programming language environment using the `BMA` package (Raftery, Hoeting, Volinsky, Painter, & Yeung, 2010[2]). The function `bicreg` is used in the regression situation while the `bic.glm` function is used in the GLM and generalized linear modeling situations. The `bic.surv` function is used for survival or event history analysis; which will not be covered in this article. These functions "do an exhaustive search over the model space using the fast leaps and bounds algorithm" (Raftery, et al., 2005, p. 2). The leaps and bounds algorithm (Furnival & Wilson, 1974) allows these functions to return a set of the best models rather than all possible models.

Regression Example

The first example involves a fictional data set which contains the outcome variable extroversion (extro) and 12 possible predictors; openness (open), agreeableness (agree), social engagement (social), cognitive engagement (cognitive), physical engagement (physical), cultural engagement (cultural), vocabulary (vocab), abstruse analogies (abstruse), block design (block), common analogies (common), letter sets (sets), and letter series (series). All 13 variables are assumed to be interval scaled. There are 750 cases in the data set, with no missing values.

First, import the data from the web using the `foreign` package, because the data file is in SPSS format. Then get a summary of the data, if desired, using the `summary` function.

[1] `http://journal.r-project.org/archive.html`

[2] `http://cran.r-project.org/web/packages/BMA/index.html`

```
R Console
File  Edit  Misc  Packages  Windows  Help

R version 2.12.1 (2010-12-16)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(foreign)
> data.1 <- read.spss("http://www.unt.edu/rss/class/Jon/R_SC/Module9/SAS_Ex/SEMData.sav",
+   use.value.labels=TRUE, max.value.labels=Inf, to.data.frame=TRUE)
Warning message:
In read.spss("http://www.unt.edu/rss/class/Jon/R_SC/Module9/SAS_Ex/SEMData.sav",  :
  C:\DOCUME~1\jds0282\LOCALS~1\Temp\Rtmp2aquBy\file68d713b9: Unrecognized record type 7, subtype 18 encountered ir
> summary(data.1)
     extro            open           agree           social          cognitive       physical        cultural
 Min.   :33.26   Min.   :19.28   Min.   :20.14   Min.   : 54.02   Min.   :29.33   Min.   :14.96   Min.   : 39.14
 1st Qu.:53.89   1st Qu.:35.78   1st Qu.:31.25   1st Qu.: 89.95   1st Qu.:44.93   1st Qu.:22.42   1st Qu.: 67.64
 Median :60.04   Median :39.90   Median :34.98   Median :100.62   Median :49.92   Median :24.90   Median : 74.62
 Mean   :60.00   Mean   :40.00   Mean   :35.00   Mean   :100.00   Mean   :50.00   Mean   :25.00   Mean   : 75.00
 3rd Qu.:66.11   3rd Qu.:44.20   3rd Qu.:38.56   3rd Qu.:109.96   3rd Qu.:54.85   3rd Qu.:27.61   3rd Qu.: 82.00
 Max.   :91.21   Max.   :60.70   Max.   :51.90   Max.   :145.90   Max.   :75.04   Max.   :36.64   Max.   :112.56
     vocab          abstruse         block           common           sets            series
 Min.   :31.01   Min.   :10.22   Min.   : 7.993   Min.   :24.02   Min.   : 42.60   Min.   :24.84
 1st Qu.:49.75   1st Qu.:18.01   1st Qu.:18.092   1st Qu.:40.48   1st Qu.: 71.93   1st Qu.:44.92
 Median :54.66   Median :19.88   Median :20.186   Median :45.15   Median : 80.31   Median :50.12
 Mean   :55.00   Mean   :20.00   Mean   :20.000   Mean   :45.00   Mean   : 80.00   Mean   :50.00
 3rd Qu.:60.30   3rd Qu.:22.10   3rd Qu.:21.958   3rd Qu.:49.80   3rd Qu.: 87.91   3rd Qu.:55.02
 Max.   :79.80   Max.   :29.59   Max.   :28.029   Max.   :71.00   Max.   :118.67   Max.   :69.89
>
```

Next, load the `BMA` package which contains the functions necessary for Bayesian Model Averaging. Note that there are three dependencies.

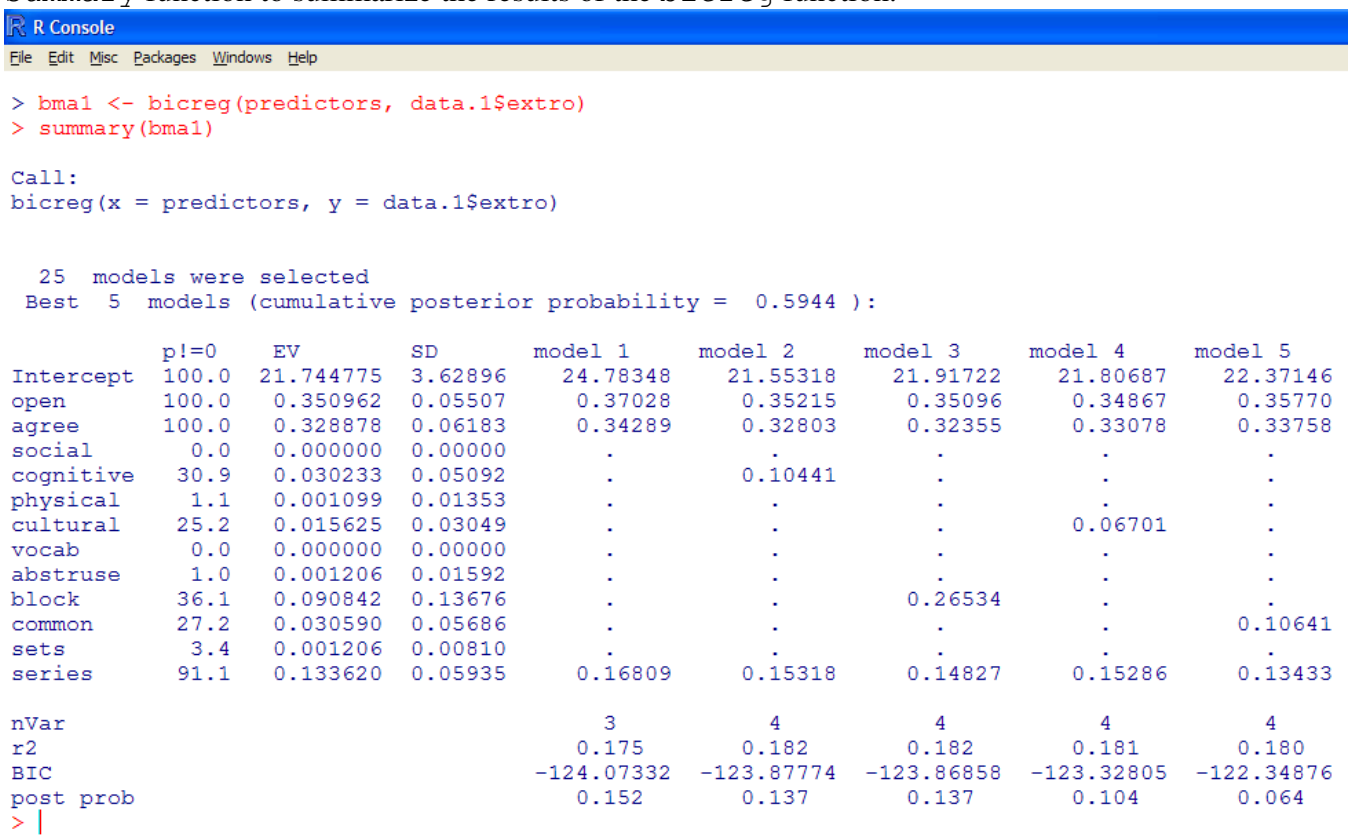
```
> library(BMA)
Loading required package: survival
Loading required package: splines
Loading required package: leaps
>
```

The `bicreg` function is used in the linear regression situation. However, the function requires a matrix of the possible predictor variables, so we must first create such a matrix. Using the attach function allows us to reference the variables by name directly (as opposed to using the tedious $ operator, e.g. data.1$open). The head function simply lists the first 6 elements of an object.

```
R Console
File  Edit  Misc  Packages  Windows  Help

> attach(data.1)
> predictors <- as.matrix(cbind(open, agree, social, cognitive, physical, cultural, vocab, abstruse,
+                               block, common, sets, series))
> detach(data.1)
> head(predictors)
       open  agree  social cognitive physical cultural  vocab abstruse  block common   sets series
[1,] 36.815 33.427 115.256    50.069   27.878   66.378 65.891   19.979 20.095 34.428 77.797 33.614
[2,] 37.888 41.627 131.831    37.026   22.555   70.691 50.655   20.416 22.117 42.485 82.092 48.746
[3,] 45.112 36.803 102.198    53.582   27.036   80.484 67.855   21.480 21.821 42.329 70.697 48.833
[4,] 44.371 32.699  86.584    51.327   18.473   74.482 58.050   22.196 16.520 55.954 66.662 59.838
[5,] 34.718 30.608  90.806    49.700   27.666   96.882 59.269   20.584 23.555 47.069 96.164 53.973
[6,] 39.189 32.762 102.514    58.546   28.249   73.264 58.794   19.540 17.459 42.871 73.331 43.602
>
```

Now we can submit the `bma` function by simply assigning it to a named object (here: `bma1`) and supply-

ing it with the matrix of predictors and the outcome variable (`data.1$extro`). We can use the common `summary` function to summarize the results of the `bicreg` function.



The column "p!=0" indicates the probability that the coefficient for a given predictor is NOT zero, among the 25 models returned. The column "EV" displays the BMA posterior distribution mean for each coefficient and the column "SD" displays the BMA posterior distribution standard deviation for each coefficient. Only the five best models are displayed. We can see that the first model "model 1" (which includes only open, agree, & series) is the best because it has the lowest BIC and the largest posterior probability (of being the *correct* model). Notice, at the bottom of each model column, the number of predictors and R^2 value is displayed. Generally, the first model (Model 1) is the best model; however, it may be the case that theory dictates the inclusion of some variables which were excluded by the first model. For each variable included in a given model, the coefficient (or parameter value) for that variable is given (e.g. Model 1, open coefficient = 0.37028). Remember that the substantive interpretation of each coefficient is, for instance: for a one unit change in open (predictor), there would be a corresponding change of 0.37028 in extro (outcome), based on Model 1.

The Ordinary Least Squares (OLS) part of the output (not printed by default) gives a matrix, with each model as a row and each predictor as a column; listing the estimated (OLS) coefficient for each variable in a given model (of all 25 models returned). The OLS output can be accessed using the $ naming convention (e.g. `bma1$ols`). The output below has been cut off at the right edge to save space in this article.

R Console

File Edit Misc Packages Windows Help

```
> bma1$ols
           Int       open      agree social  cognitive  physical   cultural vocab  abstruse     block      common        se
 [1,] 24.78348 0.3702778 0.3428869      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
 [2,] 21.55318 0.3521489 0.3280274      0 0.10441364 0.0000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
 [3,] 21.91722 0.3509556 0.3235459      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.2653369 0.00000000 0.000000
 [4,] 21.80687 0.3486714 0.3307764      0 0.00000000 0.0000000 0.06701130     0 0.0000000 0.0000000 0.00000000 0.000000
 [5,] 22.37146 0.3576982 0.3375796      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.0000000 0.10641364 0.000000
 [6,] 19.37852 0.3370756 0.3125669      0 0.09258001 0.0000000 0.00000000     0 0.0000000 0.2352047 0.00000000 0.000000
 [7,] 19.68302 0.3345595 0.3153809      0 0.00000000 0.0000000 0.05807000     0 0.0000000 0.2333771 0.00000000 0.000000
 [8,] 19.95026 0.3410159 0.3201676      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.2460047 0.09599173 0.000000
 [9,] 19.65668 0.3423598 0.3244702      0 0.09611869 0.0000000 0.00000000     0 0.0000000 0.0000000 0.09499189 0.000000
[10,] 19.80716 0.3384805 0.3212446      0 0.08489152 0.0000000 0.05290444     0 0.0000000 0.0000000 0.00000000 0.000000
[11,] 19.86933 0.3390535 0.3269584      0 0.00000000 0.0000000 0.06164403     0 0.0000000 0.0000000 0.09599900 0.000000
[12,] 22.36333 0.3497266 0.3312907      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.2881681 0.13975661 0.000000
[13,] 22.30417 0.3526712 0.3377302      0 0.10792544 0.0000000 0.00000000     0 0.0000000 0.0000000 0.14160294 0.000000
[14,] 23.00027 0.3653516 0.3407405      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.038280
[15,] 25.62717 0.3710178 0.3538204      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.0000000 0.15885299 0.000000
[16,] 22.47530 0.3485262 0.3402212      0 0.00000000 0.0000000 0.07035929     0 0.0000000 0.0000000 0.14219863 0.000000
[17,] 19.81275 0.3360036 0.3196786      0 0.09446924 0.0000000 0.00000000     0 0.0000000 0.2565509 0.12675253 0.000000
[18,] 17.78475 0.3290992 0.3103171      0 0.08576006 0.0000000 0.00000000     0 0.0000000 0.2199221 0.08690588 0.000000
[19,] 23.23667 0.3609978 0.3339856      0 0.00000000 0.0980681 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
[20,] 23.48868 0.3620512 0.3368082      0 0.00000000 0.0000000 0.00000000     0 0.1168054 0.0000000 0.00000000 0.000000
[21,] 20.55823 0.3477319 0.3226493      0 0.00000000 0.0000000 0.00000000     0 0.0000000 0.2529948 0.00000000 0.032035
[22,] 20.05711 0.3330312 0.3223470      0 0.00000000 0.0000000 0.06016601     0 0.0000000 0.2538211 0.12779117 0.000000
[23,] 18.04466 0.3266617 0.3128901      0 0.00000000 0.0000000 0.05373272     0 0.0000000 0.2180215 0.08809920 0.000000
[24,] 20.38230 0.3452994 0.3294206      0 0.00000000 0.0000000 0.06419038     0 0.0000000 0.0000000 0.00000000 0.033271
[25,] 18.04519 0.3264709 0.3079877      0 0.07660347 0.0000000 0.04604201     0 0.0000000 0.2150647 0.00000000 0.000000
```

Notice, both open and agree display fairly stable estimated coefficient values across all 25 models, this is why they both have a "p!=0" value of 100% (indicating that their coefficient is NOT zero 100% of the time among these models).

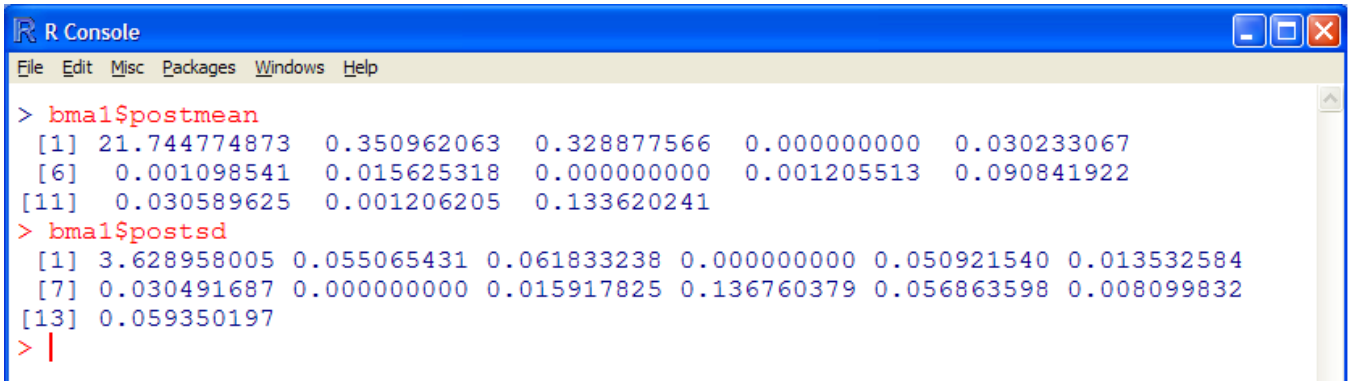
The standard errors for the above estimated coefficients can be retrieved using the `se` argument (e.g. `bma1$se`). Again, the output below has been cut off at the right edge to save space in this article.

R Console

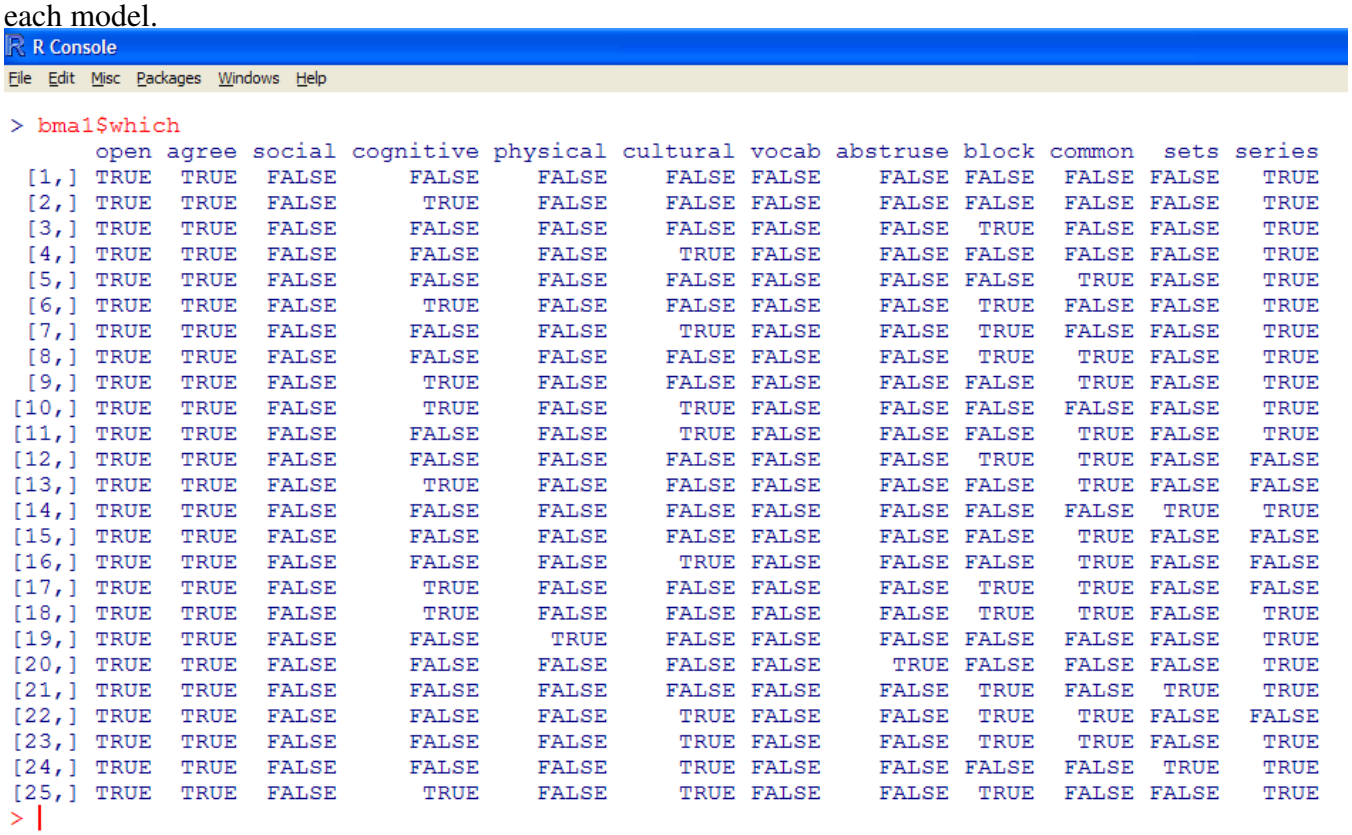
File Edit Misc Packages Windows Help

```
> bma1$se
           Int       open      agree social  cognitive   physical   cultural vocab  abstruse     block      common        se
 [1,] 2.899443 0.05345853 0.06095589      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
 [2,] 3.158245 0.05374503 0.06101909      0 0.04124540 0.00000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
 [3,] 3.103319 0.05381122 0.06121605      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.1049057 0.00000000 0.000000
 [4,] 3.140871 0.05402805 0.06096412      0 0.00000000 0.00000000 0.02768857     0 0.0000000 0.0000000 0.00000000 0.000000
 [5,] 3.091123 0.05362312 0.06084535      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.0000000 0.04816329 0.000000
 [6,] 3.297421 0.05402703 0.06125091      0 0.04147692 0.00000000 0.00000000     0 0.0000000 0.1054938 0.00000000 0.000000
 [7,] 3.277358 0.05426677 0.06120573      0 0.00000000 0.00000000 0.02791285     0 0.0000000 0.1057935 0.00000000 0.000000
 [8,] 3.250994 0.05393623 0.06111829      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.1051473 0.04822549 0.000000
 [9,] 3.296217 0.05387190 0.06092863      0 0.04138138 0.00000000 0.00000000     0 0.0000000 0.0000000 0.04827295 0.000000
[10,] 3.290592 0.05416083 0.06102904      0 0.04250172 0.00000000 0.02852148     0 0.0000000 0.0000000 0.00000000 0.000000
[11,] 3.282533 0.05413756 0.06087380      0 0.00000000 0.00000000 0.02776524     0 0.0000000 0.0000000 0.04826527 0.000000
[12,] 3.141575 0.05407323 0.06124440      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.1044508 0.04566934 0.000000
[13,] 3.178309 0.05400825 0.06104247      0 0.04137157 0.00000000 0.00000000     0 0.0000000 0.0000000 0.04564744 0.000000
[14,] 3.151776 0.05353018 0.06093067      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.0000000 0.00000000 0.026630
[15,] 2.923185 0.05375633 0.06096589      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.0000000 0.04534135 0.000000
[16,] 3.166339 0.05428971 0.06098072      0 0.00000000 0.00000000 0.02772517     0 0.0000000 0.0000000 0.04565110 0.000000
[17,] 3.328108 0.05426059 0.06128796      0 0.04160130 0.00000000 0.00000000     0 0.0000000 0.1050871 0.04590109 0.000000
[18,] 3.409652 0.05412808 0.06117193      0 0.04158801 0.00000000 0.00000000     0 0.0000000 0.1056779 0.04832172 0.000000
[19,] 3.178605 0.05401385 0.06139983      0 0.00000000 0.08269638 0.00000000     0 0.0000000 0.0000000 0.00000000 0.000000
[20,] 3.124363 0.05396032 0.06119127      0 0.00000000 0.00000000 0.00000000     0 0.1051081 0.0000000 0.00000000 0.000000
[21,] 3.302297 0.05386216 0.06120243      0 0.00000000 0.00000000 0.00000000     0 0.0000000 0.1053769 0.00000000 0.026671
[22,] 3.312118 0.05449700 0.06123689      0 0.00000000 0.00000000 0.02795867     0 0.0000000 0.1054125 0.04589650 0.000000
[23,] 3.393357 0.05435501 0.06112558      0 0.00000000 0.00000000 0.02797066     0 0.0000000 0.1059638 0.04831354 0.000000
[24,] 3.340491 0.05407522 0.06095090      0 0.00000000 0.00000000 0.02777019     0 0.0000000 0.0000000 0.00000000 0.026641
[25,] 3.396898 0.05437203 0.06125234      0 0.04260999 0.00000000 0.02866286     0 0.0000000 0.1061252 0.00000000 0.000000
```

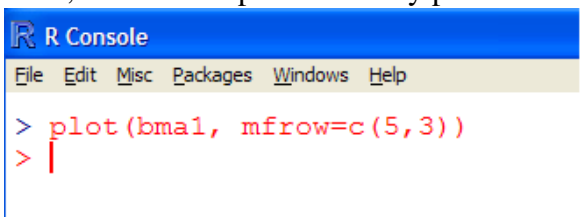
The `postmean` part of the output (printed with summary in the "EV" column) contains the average posterior coefficient for each predictor and the `postsd` provides the standard deviation of each average posterior coefficient.



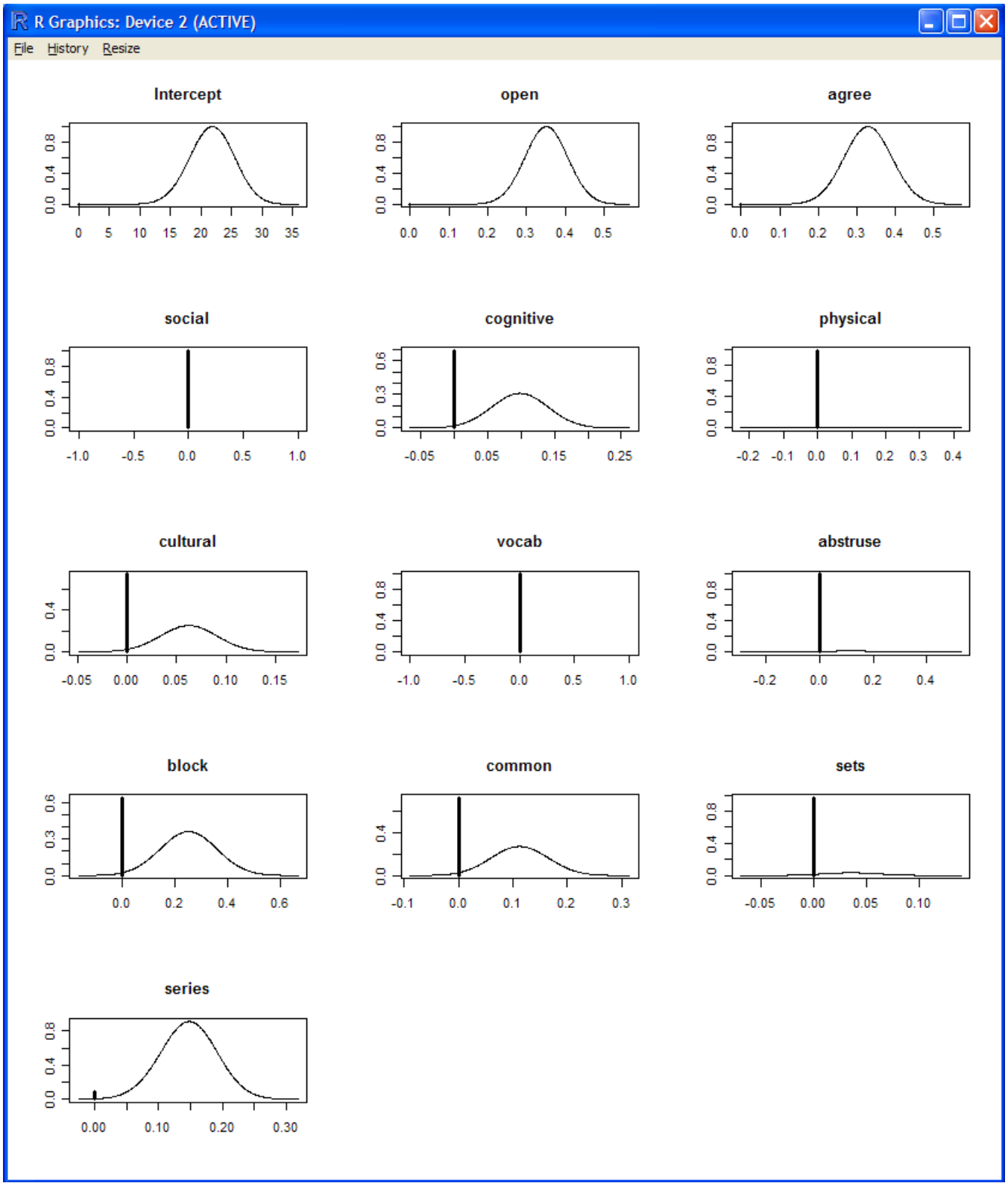
The `which` part of the output (not provided with the `summary`) contains a matrix, with each model as a row and each predictor variable as a column; listing (`TRUE` or `FALSE`) whether a variable was included in each model.



The `BMA` package also contains a `plot` function for displaying the posterior distributions of each coefficient; in this example the density plots are displayed in 5 rows and 3 columns.



Notice, among the density plots, each variable which is of little importance contains a spike at 0.0. These are the variables which are least influential to the outcome variable (e.g. social)); their coefficients are centered on, and most likely are, zero.



For a complete description of the `bicreg` function simply type `help(bicreg)` in the R console once the `BMA` package is loaded.

GLM Example

We can use the same data and predictors from above to illustrate the application of BMA to a GLM

situation using the `bic.glm` function. The `bic.glm` function can accept a matrix of predictors and the outcome variable (as above with the `bicreg` function), or the formula can be specified directly (e.g. `extro ~ open + agree + social + cognitive ... series`). The `bic.glm` function also accepts the `glm.family` argument to specify non-Gaussian models (e.g. Poisson, binomial, etc.).

R Console

File Edit Misc Packages Windows Help

```
> bma2 <- bic.glm(predictors, data.1$extro, glm.family = "gaussian")
> summary(bma2)

Call:
bic.glm.matrix(x = predictors, y = data.1$extro, glm.family = "gaussian")


  27  models were selected
 Best  5  models (cumulative posterior probability =  0.5857 ):

           p!=0    EV        SD        model 1     model 2     model 3     model 4     model 5
Intercept  100    21.722839  3.615784   2.478e+01   2.155e+01   2.192e+01   2.181e+01   2.237e+01
open       100.0   0.350899  0.054866   3.703e-01   3.521e-01   3.510e-01   3.487e-01   3.577e-01
agree      100.0   0.328811  0.061616   3.429e-01   3.280e-01   3.235e-01   3.308e-01   3.376e-01
social       0.0   0.000000  0.000000         .           .           .           .           .
cognitive   32.1   0.031378  0.051532         .       1.044e-01       .           .           .
physical     1.1   0.001082  0.013393         .           .           .           .           .
cultural    24.8   0.015386  0.030308         .           .           .       6.701e-02       .
vocab        0.0   0.000000  0.000000         .           .           .           .           .
abstruse     1.8   0.002019  0.020481         .           .           .           .           .
block       35.5   0.089368  0.136100         .           .       2.653e-01       .           .
common      26.8   0.030059  0.056465         .           .           .           .       1.064e-01
sets         4.2   0.001413  0.008717         .           .           .           .           .
series      91.3   0.133788  0.059011   1.681e-01   1.532e-01   1.483e-01   1.529e-01   1.343e-01

nVar                                        3           4           4           4           4
BIC                                    -4.183e+03  -4.182e+03  -4.182e+03  -4.182e+03  -4.181e+03
post prob                               0.148       0.136       0.135       0.103       0.063
>
```

Notice when using `bic.glm` and specifying "Gaussian" the estimation of the posterior means and standard deviations are slightly different from what was observed with the `bicreg` function. Below the means and standard deviations from the `bicreg` and `bic.glm` functions are displayed; `bma1` and `bma2` respectively.

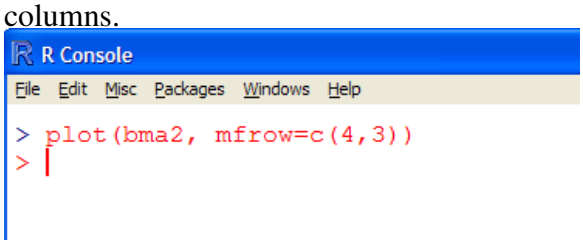
R Console

File Edit Misc Packages Windows Help

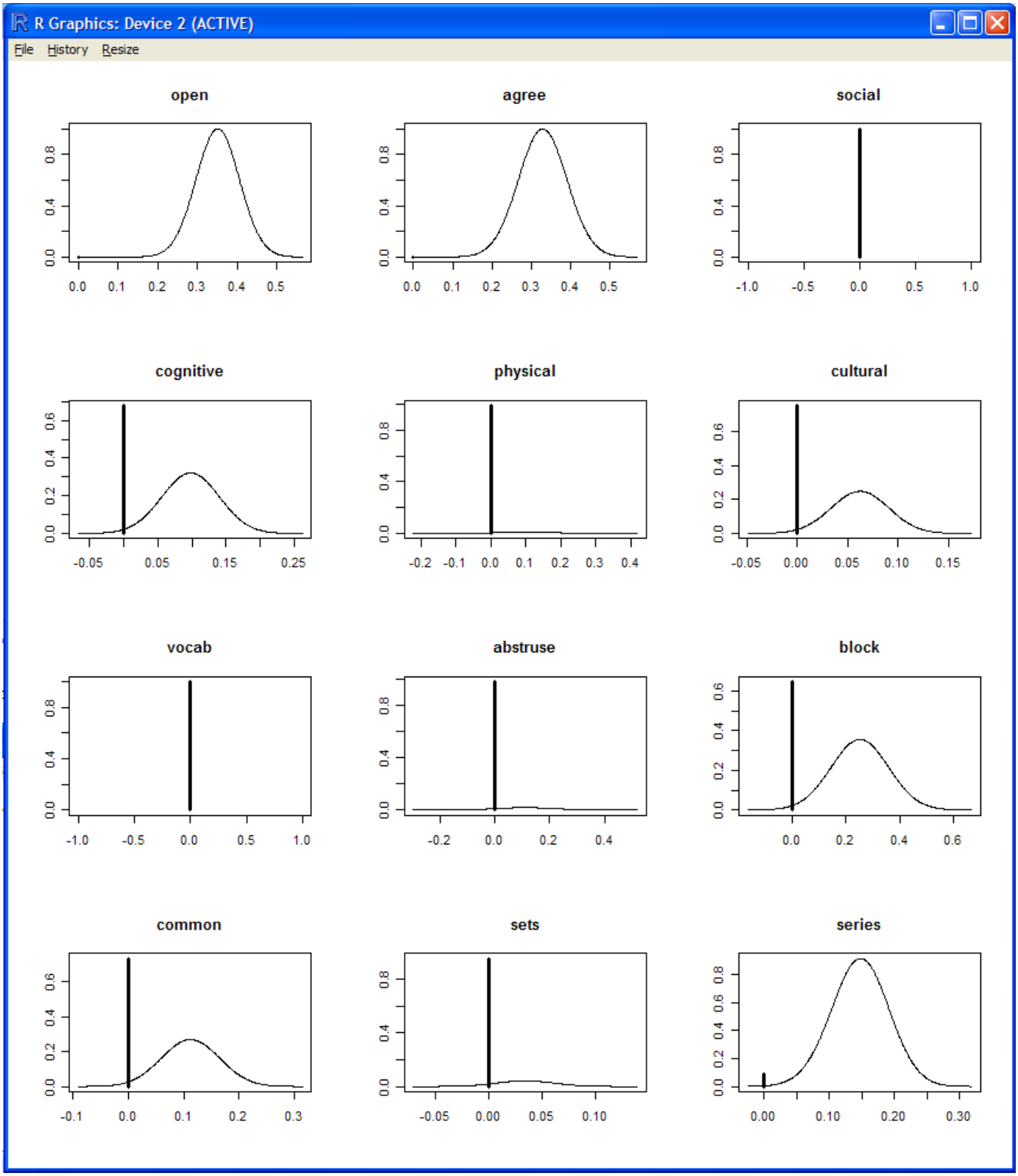
```
> bma1$postmean
 [1] 21.744774873  0.350962063  0.328877566  0.000000000  0.030233067
 [6]  0.001098541  0.015625318  0.000000000  0.001205513  0.090841922
[11]  0.030589625  0.001206205  0.133620241
> bma2$postmean
 [1] 21.722838937  0.350898681  0.328810972  0.000000000  0.031377816
 [6]  0.001081515  0.015386482  0.000000000  0.002019300  0.089368321
[11]  0.030059142  0.001413382  0.133788175
>
> bma1$postsd
 [1] 3.628958005 0.055065431 0.061833238 0.000000000 0.050921540 0.013532584
 [7] 0.030491687 0.000000000 0.015917825 0.136760379 0.056863598 0.008099832
[13] 0.059350197
> bma2$postsd
 [1] 3.615783733 0.054866450 0.061616092 0.000000000 0.051532184 0.013392648
 [7] 0.030308143 0.000000000 0.020481080 0.136099936 0.056465321 0.008716879
[13] 0.059011367
>
```

The `plot` function also works with `bic.glm` objects; here displaying the density plots in 4 rows and 3

columns.



Notice again, the variables which do not contribute to the outcome variable are shown with spikes at zero in their density plots; indicating that their coefficients are most likely zero.



For a complete description of the `bic.glm` function simply type `help(bic.glm)` in the R console once the `BMA` package is loaded.

Binomial Generalized Linear Model Example

The binomial generalized linear model is the logistic (logit) model. The `bic.glm` function is used, simply specifying the binomial `glm.family` argument as would be done with the standard `glm` function.

This example uses a simulated data set which contains one binary outcome variable (y = 0 or 1) and four interval predictor variables (x1, x2, x3, x4). There are 400 cases of data with no missing values. The data also contains a code variable which simply identifies each case. The data file is a space delimited text (.txt) file, so the `foreign` package is not necessary for importing it into R.

```
R Console
File  Edit  Misc  Packages  Windows  Help

> data.2 <- read.table("http://www.unt.edu/rss/class/Jon/R_SC/Module9/logreg1.txt",
+           header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
> summary(data.2)
      code             y               x1                x2               x3               x4
 Min.   :  1.0   Min.   :0.0   Min.   :-0.02921   Min.   :0.5000   Min.   :0.1590   Min.   :0.5000
 1st Qu.:100.8   1st Qu.:0.0   1st Qu.: 2.34900   1st Qu.:0.5523   1st Qu.:0.5000   1st Qu.:0.6746
 Median :200.5   Median :0.5   Median : 2.97260   Median :2.5402   Median :1.0000   Median :1.0000
 Mean   :200.5   Mean   :0.5   Mean   : 2.96276   Mean   :2.3514   Mean   :1.8551   Mean   :1.4831
 3rd Qu.:300.2   3rd Qu.:1.0   3rd Qu.: 3.61822   3rd Qu.:3.3644   3rd Qu.:2.9585   3rd Qu.:1.5000
 Max.   :400.0   Max.   :1.0   Max.   : 5.55478   Max.   :5.5479   Max.   :5.5584   Max.   :5.8059
> |
```

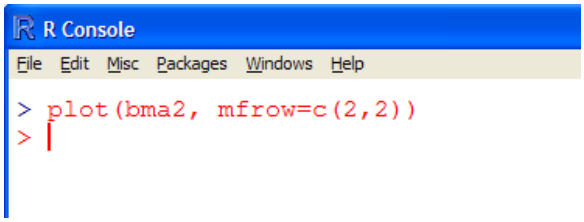
As mentioned above, when using the `bic.glm` function, one can either create a matrix of predictor variables or simply specify the formula directly. Above we used the matrix approach; here we will specify the formula directly. Of course, here we also specify the `glm.family` as binomial.

```
R Console
File  Edit  Misc  Packages  Windows  Help

> bma2 <- bic.glm(y ~ x1 + x2 + x3 + x4, data = data.2, glm.family = "binomial")
> summary(bma2)

Call:
bic.glm.formula(f = y ~ x1 + x2 + x3 + x4, data = data.2, glm.family = "binomial")


  3  models were selected
 Best  3  models (cumulative posterior probability =  1 ):

           p!=0    EV         SD       model 1     model 2     model 3
Intercept  100    -1.388e+01  3.32880  -1.599e+01  -1.156e+01  -1.600e+01
x1           2.5   2.176e-05  0.06404        .           .       8.727e-04
x2         100.0   1.380e+00  0.35031   1.553e+00   1.189e+00   1.553e+00
x3         100.0   8.288e+00  1.81053   8.922e+00   7.593e+00   8.922e+00
x4          52.4   1.026e+00  1.11520   1.959e+00         .     1.960e+00

nVar                                        3           2           4
BIC                                    -2.315e+03  -2.315e+03  -2.309e+03
post prob                               0.499       0.476       0.025
> |
```

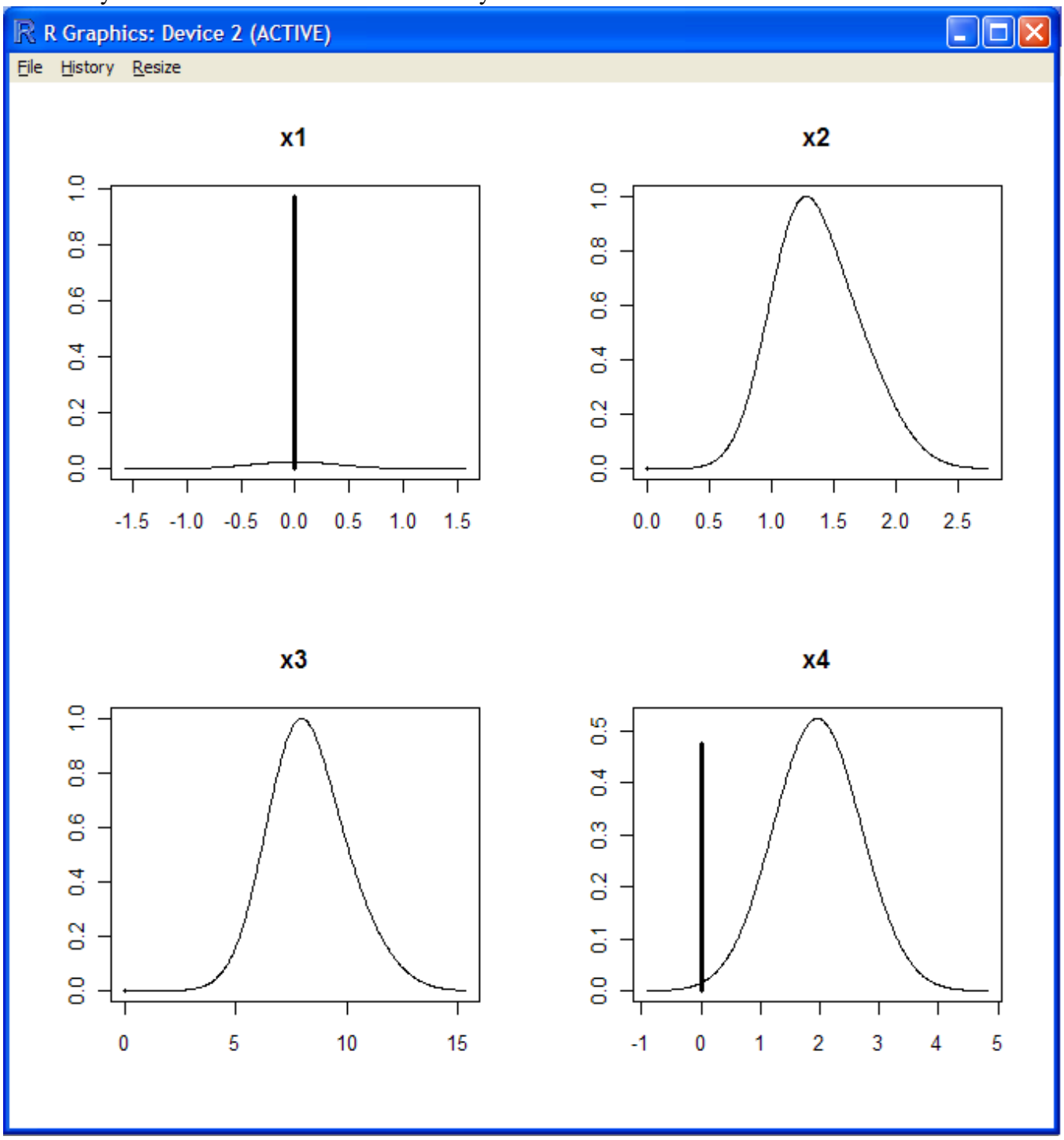
Although the summary of the `bic.glm` object here is interpreted the same way as the previous two examples (in terms of model/variable importance using BIC and posterior probability), it is important to remember that the coefficients for each predictor here (binomial setting) are NOT interpreted in the same way as they would be in the Gaussian setting(s).

Remember, when interpreting coefficients in a logistic (binomial) setting, the values are interpreted as changes in the logit. The logistic coefficient is the expected amount of change in the logit for each one unit change in the predictor. The logit is what is being predicted; it is the odds of membership in the category of the outcome variable with the numerically higher value (here a 1, rather than 0). The closer a logistic coefficient is to zero, the less influence it has in predicting the logit.

The `plot` function works the same with way with binomial models as it did with the above models.



The plots simply confirm what was expressed in the `summary` function, x1 has virtually nothing to contribute to y and x4 has a moderate influence on y.



For a complete description of the different families available to the `glm` function (and the `bic.glm` function), type `help("family")` in the R console.

Keep in mind, there are other packages available for conducting BMA in R. Perhaps most notable, is the `mlogitBMA`[3] package which offers extensions to the `bic.glm` function so that BMA can be applied in the multinomial logistic situation. Other packages which incorporate BMA include: `BAS`[4], `BMS`[5], and `ensembleBMA`[6].

An Adobe.pdf version of this article can be found here:
`http://www.unt.edu/rss/rssmattersindex.htm`.

[3] `http://cran.r-project.org/web/packages/mlogitBMA/index.html`
[4] `http://cran.r-project.org/web/packages/BAS/index.html`
[5] `http://cran.r-project.org/web/packages/BMS/index.html`
[6] `http://cran.r-project.org/web/packages/ensembleBMA/index.html`

References & Resources

Brown, P. J., & Vannucci, T. F. (2002). Bayes model averaging with selection of regressors. *Journal of the Royal Statistical Society (Series B: Statistical Methodology), 64*(3), 519 – 536.

Furnival, G. M., & Wilson, R. W. (1974). Regression by leaps and bounds. *Technometrics, 16*(4), 499 – 511.

Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science, 14*(4), 382 – 401.

Kass, R. E., & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association, 90*(430), 773 – 795.

Raftery, A. E. (1995). Bayesian model selection in social research. *Sociological Methodology, 25*, 111 – 163.

Raftery, A. E. (1996). Approximate Bayes factors and accounting for model uncertainty in generalized linear models. *Biometrika, 83*(2), 251 – 266.

Raftery, A. E., Painter, I. S., & Volinsky, C. T. (2005). BMA: An R package for Bayesian Model Averaging. *R News, 5*(2), 2 - 8. Available at:
`http://journal.r-project.org/archive.html`

Raftery, A. E., Hoeting, J., Volinsky, C. Painter, I., & Yeung, K. Y. (2010). Package 'BMA'. Available at:
`http://cran.r-project.org/web/packages/BMA/index.html`

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*, 461 – 464.

Wang, D., Zhang, W., & Bakhai, A. (2004). Comparison of Bayesian model averaging and stepwise methods for model selection in logistic regression. *Statistics in Medicine, 23*, 3451 – 3467.

Until next time, *It ain't me, it ain't me; I ain't no Senator's son...*

This article was last updated on February 14, 2011.

This document was created using LaTeX