Scaling Roll Call Votes with wnominate in R

Keith Poole
University of
Georgia

Jeffrey Lewis
University of California,
Los Angeles

James Lo
University of
Mannheim

Royce Carroll
Rice University

Abstract

This paper presents a software package designed to estimate Poole and Rosenthal W-NOMINATE scores in R. The package uses a logistic regression model to analyze political choice data, usually (though not exclusively) from a legislative setting. In contrast to other scaling methods, W-NOMINATE explicitly assumes probabilistic voting based on a spatial utility function, where the parameters of the utility function and the spatial coordinates of the legislators and the votes can all be estimated on the basis of observed voting behavior. Building on software written by Poole in Fortran, the new **wnominate** package in R facilitates easier data input and manipulation, generates bootstrapped standard errors, and includes a new suite of graphics functions to display the results. We demonstrate the functionality of this package by conducting a natural experiment using roll calls – an experiment which is greatly simplified by the data manipulation capabilities of the **wnominate** package in R.

Keywords: multivariate, R, roll call, scaling, **wnominate**.

1. Introduction

Legislative roll call analysis has enjoyed a long history in political science, dating back to the work of A. Lawrence Lowell in 1902 (Lowell 1902). In analyzing a roll call matrix containing hundreds of legislators and hundreds or even thousands of votes, procedures that are able to reduce the dimensionality of the data matrix can be helpful in uncovering patterns in voting. As MacRae pointed out as early as 1958 (MacRae 1958), "one way to try to account for political choices is to imagine that each chooser occupies a fixed position in a space of one or more dimensions, and to suppose that every choice presented to him is a choice between two or more points in that space."

The essence of MacRae's statement suggests that one can think of a roll call matrix as being the result of two different sets of variables – an ideal point for each legislator that repre-

sents their vote preference or ideology, and separate yea and nay locations for each roll call. Legislators are assumed to have an ideal point on each dimension, along with single-peaked and symmetric preferences. The utility of a yea or nay vote can then be thought of as a function of the distance between the legislator's ideal point and the yea/nay locations, along with an idiosyncratic utility shock or "error term." Legislators are then assumed to maximize their utility by voting for the bill outcome that minimizes the distance between the yea/nay location and their ideal point, subject to a stochastic random utility component (McFadden 1973). This multidimensional spatial model was formally proposed by Hinich and Ordeshook (Cahoon, Hinich, and Ordeshook 1976; Ordeshook 1976; Hinich and Pollard 1981) and developed in depth by Hinich and his colleagues (Enelow and Hinich 1984; Hinich and Munger 1994, 1997).

The Hinich-Ordeshook spatial model of voting embodies the key insight of Converse (1964). According to Converse, a belief system can be thought of as a set of beliefs that are often observed to be bundled together. Taking the current US Congress as an example, a legislator who favors raising the minimum wage and a federally funded universal health care system is also likely to support affirmative action, higher taxes, and lower defense spending. Empirically, a belief system containing high levels of 'constraint' means that given one or two issue positions from a legislator, we are able to predict many other seemingly unrelated issue positions. When applied to roll call data, the existence of constraint has a straightforward geometric interpretation, suggesting that the issues lie on a low-dimensional hyperplane. Stated another way, the vast majority of votes in a roll call matrix are classified correctly using cutlines derived from only one or two issue dimensions.

Although a number of different procedures (Poole 2000; Heckman and Snyder 1997; Clinton, Jackman, and Rivers 2004) have been developed to fit the Hinich-Ordeshook spatial model to Congressional roll call votes, the W-NOMINATE model remains among the most popular.[1] Fitting such a model allows us to recover estimates of the legislator ideal points and yea/nay locations of a roll call matrix, and these results can themselves be subjected to further analysis. W-NOMINATE has been widely applied to study polarization in American politics (McCarty, Poole, and Rosenthal 2006) and party discipline (McCarty, Poole, and Rosenthal 2001), among other topics. Outside of the setting of the US Congress, the model has been used to study voting in the United Nations (Voeten 2001), and the California state legislature (Masket 2007).

This package estimates Poole and Rosenthal W-NOMINATE scores from roll call votes supplied through a `rollcall` object from package **pscl** (Jackman 2010).[2] The R (R Development Core Team 2011) version of **wnominate** computes ideal points using the same Fortran code base as the previous **wnom9707** software, and is available from the Comprehensive R Archive Network at http://CRAN.R-project.org/package=wnominate. It improves upon the earlier software in three ways. First, it is now considerably easier to input new data for estimation, as the current software no longer relies exclusively on the old `.ord` file format for data input. Secondly, the software now allows users to generate standard errors for their ideal point estimates using a parametric bootstrap. Finally, **wnominate** includes a full suite of graphics functions to facilitate easier interpretation of the results.

W-NOMINATE scores are based on the spatial model of voting. Let s denote the number of

---

[1]See Poole and Rosenthal (1985) for a discussion comparing W-NOMINATE scores to the Guttman scaling procedure that historically preceeded it.

[2]Production of this package, **wnominate**, is supported by NSF Grant SES-0611974.

policy dimensions, which are indexed by $k = 1, \ldots, s$; let p denote the number of legislators ($i = 1, \ldots, p$; and q denote the number of roll call votes ($j = 1, \ldots, q$. Let x_i be the ideal point of legislator i, a vector of length s. Each roll call vote is represented by vectors of length s, z_{jy} and z_{jn}, where y and n stand for the policy outcomes associated with Yea and Nay, respectively.

Legislator i's utility for outcome y on roll call j is

$$U_{ijy} = u_{ijy} + \epsilon_{ijy} \tag{1}$$

$$u_{ijy} = \beta \exp\left[-\frac{\sum_{k=1}^{s} w_k^2 d_{ijyk}^2}{2}\right] \tag{2}$$

where u_{ijy} represents the deterministic part of the legislator's utility while ϵ_{ijy} represents the stochastic component. The d_{ijyk}^2 term in the exponent is the Euclidean distance between a legislator's ideal point x_i and the Yea bill location z_{jyk}; namely,

$$d_{ijy}^2 = \sum_{k=1}^{s} (x_{ik} - z_{jyk})^2 \tag{3}$$

Weight w and β are estimated but set with initial values of 0.5 and 15 respectively. β can be thought of as a signal-to-noise ratio, where as β increases in value, the deterministic portion of the utility function overwhelms the stochastic portion. In multiple dimensions, β is only estimated for the first dimension and is thereafter kept constant. For dimensions 2 to s, the corresponding w_k is estimated, with the starting value of w_k set at 0.5 each time.

The stochastic element in the utility, ϵ, is assumed to follow an extreme value distribution. This allows us to express the probability that legislator i votes for outcome y on roll call j as:

$$\Pr(\text{Yea}) = P_{ijy} = \frac{\exp(u_{ijy})}{\exp(u_{ijy}) + \exp(u_{ijn})} \tag{4}$$

Extending this even further, by replacing y with the index 1 and n with the index 2, and then allowing l to be the index for y and n, we can express the likelihood function to be maximized as:

$$L = \prod_{i=1}^{p} \prod_{j=1}^{q} \prod_{l=1}^{2} P_{ijl}^{C_{ijl}} \tag{5}$$

where $C_{ijl} = 1$ if choice l is the actual choice of legislator i on roll call j and is zero otherwise.

In estimating the outcome points for each bill, W-NOMINATE estimates and reports the outcome points in terms of their midpoint and the distance between them; namely,

$$z_{jy} = z_{mj} - d_j \quad \text{and} \quad z_{jn} = z_{mj} + d_j \tag{6}$$

where z_{mj} is the midpoint and $d_j = (z_{jy} - z_{jn})/2$.

To conduct this estimation, the W-NOMINATE algorithm has three basic steps:

1. Estimate z_{mjk} and d_{jk}, conditional on x_{ik}, β, and w_k.

2. Estimate x_{ik}, conditional on z_{mjk}, d_{jk}, β, and w_k.

3. Estimate β (if $k = 1$) and w_k (if $k = 2, \ldots, s$), conditional on z_{mjk}, d_{jk}, and x_{ik}.

These steps form a global iteration, which explains the acronym W-NOMINATE (<u>w</u>eighted <u>nomina</u>l <u>t</u>hree-step <u>e</u>stimation). The procedure repeats until the z, x, and d parameters all correlate at 0.99 or better with the set estimated on the previous global iteration.

2. Usage overview

In conducting the estimate, `wnominate` requires several inputs in addition to the roll call matrix, most of which are set by default. First, one must fix a cutoff level (the `lop` argument) in terms of the proportion of voters voting with the minority side that determines whether a given roll call in included. Levels that are set too high, such as 0.1, worsen legislator estimates by failing to allow for enough differentiation between extremists and moderates. In contrast, levels that are set too low, such as 0.01, lead to the inclusion of near-unanimous votes that contain little information about member locations. The default is set at 0.025. Secondly, `wnominate` requires the specification of an initial value of β, which by default is set at 15. This functions as a signal-to-noise ratio[3], and in general should not be modified.

As with any common scaling method, the polarities of the estimated coordinates in `wnominate` are arbitrary. `wnominate` requires the specification of a legislator to be positive in each dimension. The argument `polarity` is used to orient the estimated results in the desired direction. In general, users will likely wish to orient conservatives on the right and liberals on the left, so 'positive' in this case generally means conservative. The polarity can be specified in a number of ways, though the recommended procedure here is to specify polarities using the row numbers of the legislators designated to be positive. No default arguments are set for the polarity argument.

Finally, `wnominate` allows the user to specify the number of trials that they wish to use to bootstrap standard errors. By default, `wnominate` will not conduct a bootstrap, and hence will not calculate standard errors. In general, we recommend that at least 50 bootstrap trials be used to obtain reasonable standard errors. Users are cautioned that use of the bootstrap will lengthen the amount of time needed to conduct the estimation by a considerable amount. On a 1.67 HGz computer running Windows Vista with 2 GB of memory, 50 bootstrap trials on the 90th Senate in two dimensions (a roll call matrix with 102 legislators and 519 roll calls) took 61 minutes to complete. The estimation time required for all estimates is roughly proportional to $O(\text{legislators} \cdot \text{rollcalls})$.
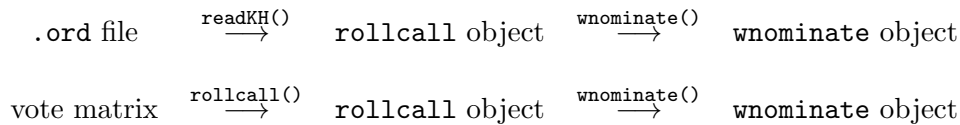
To input data into **wnominate**, we make use of the **pscl** package (Jackman 2010), which contains support for objects of class `rollcall`. The `rollcall` object simply acts as a container holding all the key components of a roll call data set in a standardized manner. Readers are advised to look at the documentation of **pscl** for further information.

`rollcall` objects can be generated in two different ways for use with **wnominate**. First, users can simply take an `.ord` file and format it as a `rollcall` object using the function `readKH`. Secondly, users can generate a vote matrix of their own in a spreadsheet and format it into a `rollcall` object using the function `rollcall`.[4] Each of these cases are supported by a

---

[3]The quantity β also controls the maximum choice probability, which is $\exp(\beta)/(\exp(\beta) + 1)$.

[4]A third option is to manually generate a `rollcall` object with known latent values using the `generateTestData` function for estimation. However, this is intended for Monte Carlo testing purposes only and is not discussed further in this paper.

similar sequence of function calls, as shown in the diagram below:

$$.\texttt{ord}\text{ file }\quad \overset{\texttt{readKH()}}{\longrightarrow} \quad \texttt{rollcall}\text{ object }\quad \overset{\texttt{wnominate()}}{\longrightarrow} \quad \texttt{wnominate}\text{ object}$$

$$\text{vote matrix }\quad \overset{\texttt{rollcall()}}{\longrightarrow} \quad \texttt{rollcall}\text{ object }\quad \overset{\texttt{wnominate()}}{\longrightarrow} \quad \texttt{wnominate}\text{ object}$$

The `rollcall` object is input to `wnominate` and the results of the estimation are output to a `wnominate` object. This object in turn can then be analyzed using the `plot` and `summary` methods built into the package, including:

- `plot.coords`: Plots ideal points in one or two dimensions.

- `plot.angles`: Plots a histogram of cut lines.

- `plot.cutlines`: Plots a specified percentage of cut lines (a Coombs mesh).

- `plot.skree`: Plots a skree plot with the first 20 eigenvalues.

- `plot.nomObject`: S3 method for a `wnominate` object that combines the four plots described above.

- `summary.nomObject`: S3 method for a `wnominate` object that summarizes the estimates.

Examples of the two cases described here, as well as an example of how the package can be used to conduct natural experiments using roll calls, are presented in the following sections. Before proceeding to the examples however, it should be noted that three types of roll call matrices are not estimated well (or at all) by **wnominate**. First, roll call matrices with perfect voting are poorly estimated in **wnominate** because metric information cannot be recovered in the absence of voting error. Secondly, roll call matrices that are too small (and hence have too little information to recover) are also poorly estimated. Finally, in roll call matrices where the legislators can be partitioned off into distinct groups with no linking legislators serving across partitions, **wnominate** is unable to estimate the problem because no information exists in the data to allow members across distinct groups to be placed in the same ideological space.

3. W-NOMINATE with `.ord` files

This is the use case that the majority of **wnominate** users are likely to fall into. Roll call votes in a fixed width `.ord` format for all US Congresses are stored online for download at:

- http://www.voteview.com/,

- http://adric.sscnet.ucla.edu/rollcall/ (latest Congress only, updates votes in real time).

wnominate takes `rollcall` objects from the **pscl** package as input. The **pscl** package includes a function, `readKH`, that takes an `.ord` file and automatically transforms it into a `rollcall` object as desired. Using the 90th Senate as an example, we access a local copy of the roll call file as follows:

```
R> library("wnominate")
R> sen90 <- data("sen90")
R> sen90


Source:                    ftp://voteview.com/sen90kh.ord
Number of Legislators:        102
Number of Votes:          596

Using the following codes to represent roll call votes:
Yea:               1 2 3
Nay:               4 5 6
Abstentions:       7 8 9
Not In Legislature:        0

Legislator-specific variables:
[1] "state"      "icpsrState" "cd"          "icpsrLegis" "party"
[6] "partyCode"
Detailed information is available via the summary function.
```

Although we will not do so in this example, we could easily subset the roll calls to apply
wnominate only to a subset of votes – a function not built into the earlier version of **wnomi-
nate**. Suppose for example that we were interested in applying **wnominate** only to bills that
pertained in some way to agriculture. Keith Poole and Howard Rosenthal's **Voteview** (Poole,
Rosenthal, and Shor 2000) software allows us to quickly determine which bills in the 90th
Senate pertain to agriculture.[5] Using this information, we could create a vector of roll calls
that we wish to select, then select for them in the `rollcall` object. In doing so, we also need
to take care to update the variable in the `rollcall` object that counts the total number of
bills. The code to do this is not executed, but is nevertheless included here as an example:

```
R> selector <- c(21, 22, 44, 45, 46, 47, 48, 49, 50, 53, 54, 55, 56, 58, 59,
+    60, 61, 62, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 77, 78, 80, 81,
+    82, 83, 84, 87, 99, 100, 101, 105, 118, 119, 120, 128, 129, 130, 131,
+    132, 133, 134, 135, 141, 142, 143, 144, 145, 147, 149, 151, 204, 209,
+    211, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 237,
+    238, 239, 252, 253, 257, 260, 261, 265, 266, 268, 269, 270, 276, 281,
+    290, 292, 293, 294, 295, 296, 302, 309, 319, 321, 322, 323, 324, 325,
+    327, 330, 331, 332, 333, 335, 336, 337, 339, 340, 346, 347, 357, 359,
+    367, 375, 377, 378, 379, 381, 384, 386, 392, 393, 394, 405, 406, 410,
+    418, 427, 437, 442, 443, 444, 448, 449, 450, 454, 455, 456, 459, 460,
+    461, 464, 465, 467, 481, 487, 489, 490, 491, 492, 493, 495, 497, 501,
+    502, 503, 504, 505, 506, 507, 514, 515, 522, 523, 529, 539, 540, 541,
+    542, 543, 544, 546, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557,
+    558, 559, 560, 561, 562, 565, 566, 567, 568, 569, 571, 584, 585, 586,
+    589, 590, 592, 593, 594, 595)
```

---

[5]**Voteview** for Windows can be downloaded at `http://www.voteview.com/`. We expect that at some point
in the future we will develop software that fully integrates **wnominate** in R with **Voteview**.

```
R> sen90$m <- length(selector)
R> sen90$votes <- sen90$votes[, selector]
```

wnominate takes a number of arguments described fully in the documentation. Most of the arguments can (and probably should) be left at their defaults, particularly when estimating ideal points from US Congresses. The default options estimate ideal points in two dimensions without standard errors, using the same beta and weight parameters as described in the introduction. Votes where the losing side has less than 2.5 per cent of the vote, and legislators who vote less than 20 times are excluded from analysis.

The most important argument that `wnominate` requires is a set of legislators who have positive ideal points in each dimension. This is the `polarity` argument to `wnominate`. In two dimensions, this might mean a fiscally conservative legislator on the first dimension, and a socially conservative legislator on the second dimension. Polarity can be set in a number of ways, such as a vector of row indices (the recommended method), a vector of names, or by any arbitrary column in the `legis.data` element of the `rollcall` object. Here, we use Senators Sparkman and Bartlett to set the polarity for the estimation. The names of the first 12 legislators are shown, and we can see that Sparkman and Bartlett are the second and fifth legislators respectively.

```
R> rownames(sen90$votes)[1:12]
```

```
[1] "JOHNSON (D USA)"  "SPARKMAN (D AL)"  "HILL (D AL)"     "GRUENING (D AK)"
[5] "BARTLETT (D AK)"  "HAYDEN (D AZ)"    "FANNIN (R AZ)"   "FULBRIGHT (D AR)"
[9] "MCCLELLAN (D AR)" "KUCHEL (R CA)"    "MURPHY (R CA)"   "DOMINICK (R CO)"
```

```
R> result <- wnominate(sen90, polarity = c(2, 5))
```

```
Preparing to run W-NOMINATE...

        Checking data...

                All members meet minimum vote requirements.

                Votes dropped:
                ... 77 of 596 total votes dropped.

        Running W-NOMINATE...

                Getting bill parameters...
                Getting legislator coordinates...
                Starting estimation of Beta...
                Getting bill parameters...
                Getting legislator coordinates...
                Starting estimation of Beta...
                Getting bill parameters...
                Getting legislator coordinates...
                Getting bill parameters...
```

```
                     Getting legislator coordinates...
                     Estimating weights...
                     Getting bill parameters...
                     Getting legislator coordinates...
                     Estimating weights...
                     Getting bill parameters...
                     Getting legislator coordinates...



W-NOMINATE estimation completed successfully.
W-NOMINATE took 10.625 seconds to execute.
```

`result` now contains all of the information from the `wnominate` estimation, the details of which are fully described in the documentation for `wnominate`. `result$legislators` contains all of the information from the `nom31.dat` file from the old **Fortran wnom9707** program, while `result$rollcalls` contains all of the information from the old `nom33.dat` file. The information can be browsed using the `fix` command as follows (not run):

```
R> legisdata <- result$legislators
R> fix(legisdata)
```

For those interested in just the ideal points, a much better way to do this is to use the `summary()` function:

```
R> summary(result)

SUMMARY OF W-NOMINATE OBJECT
----------------------------


Number of Legislators:          102 (0 legislators deleted)
Number of Votes:          519 (77 votes deleted)
Number of Dimensions:        2
Predicted Yeas:                20197 of 23899 (84.5%) predictions correct
Predicted Nays:                18942 of 22541 (84%) predictions correct
Correct Classifiction:        80.45% 84.28%
APRE:                     0.349 0.476
GMP:                      0.659 0.706


The first 10 legislator estimates are:
              coord1D coord2D
JOHNSON (D USA)   -0.452  -0.147
SPARKMAN (D AL)    0.231   0.706
HILL (D AL)        0.423   0.848
GRUENING (D AK)   -0.717   0.697
BARTLETT (D AK)   -0.440   0.502
HAYDEN (D AZ)      0.015   0.849
```
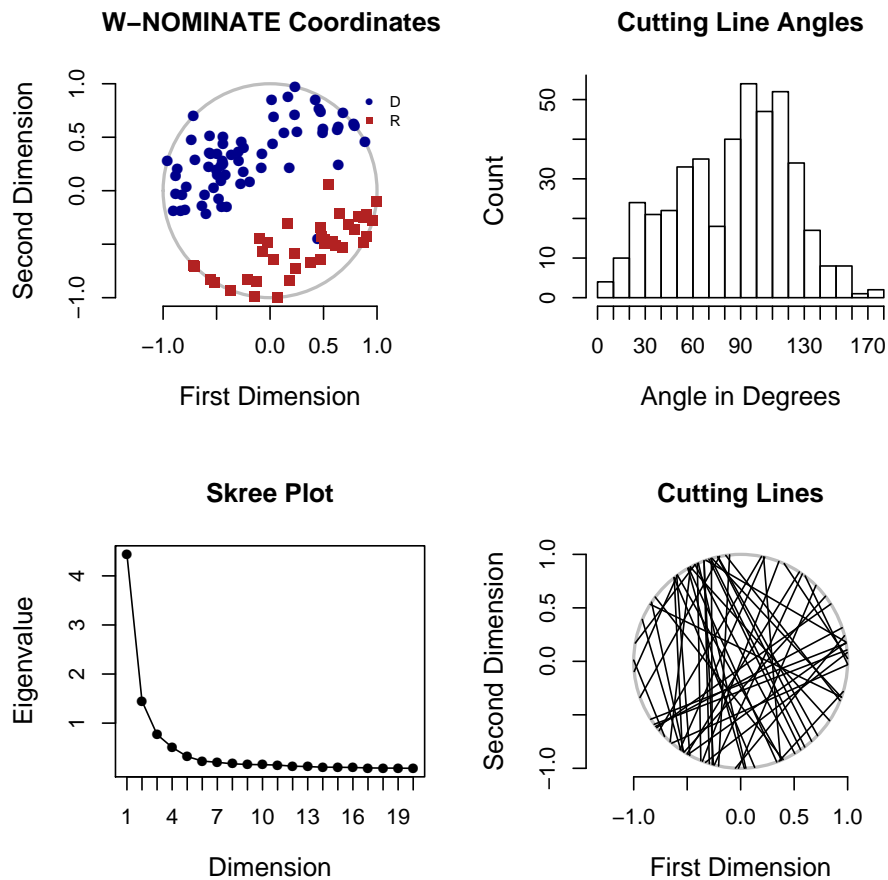
Figure 1: Summary plot of 90th Senate W-NOMINATE scores.

```
FANNIN (R AZ)        0.826   -0.238
FULBRIGHT (D AR)     0.035    0.688
MCCLELLAN (D AR)     0.629    0.569
KUCHEL (R CA)       -0.075   -0.572
```

`result` can also be plotted, with a basic summary plot achieved as follows as shown Figure 1. This basic plot splits the window into 4 parts and calls `plot.coords`, `plot.angles`, `plot.skree`, and `plot.cutlines` sequentially. Each of these four functions can also be called individually. In this example, the coordinate plot on the top left plots each legislator with their party affiliation. A unit circle is included to illustrate how W-NOMINATE scores are constrained to lie within a unit circle. The cutting angle histogram shows that most votes are well classified by a single dimension and are around 90°, although there are a number around 30° as well. The skree plot shows the first 20 eigenvalues, and the rapid decline after the second eigenvalue suggests that a two-dimensional model describes the voting behavior of the 90th Senate well. The final plot shows 50 random cutlines, and can be modified to show any desired number of cutlines as necessary.
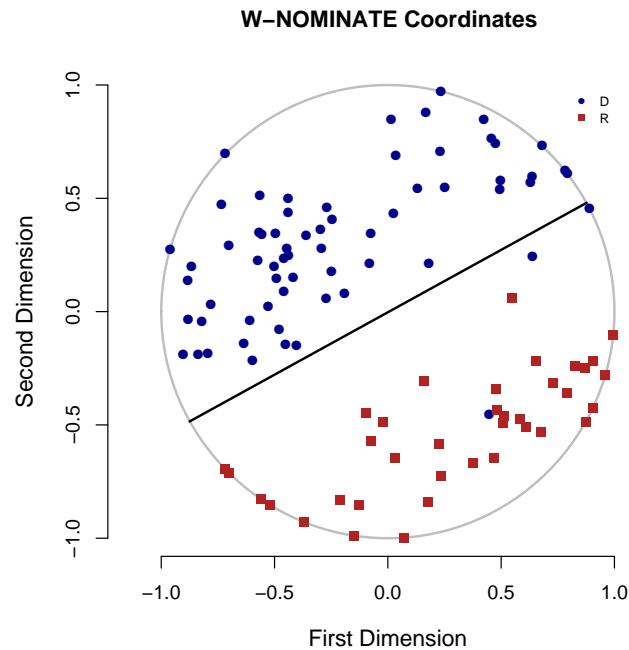
W–NOMINATE Coordinates

Figure 2: 90th Senate W-NOMINATE scores with cutline for the 58th vote.

```
R> plot(result)
```

Three things should be noted about the use of the `plot` functions. First, the functions always plot the results from the first two dimensions, but the dimensions used (as well as titles and subheadings) can all be changed by the user if, for example, they wish to plot dimensions 2 and 3 instead. Secondly, plots of one dimensional `wnominate` objects work somewhat differently than in two dimensions. Finally, `plot.coords` can be modified to include cutlines from whichever votes the user desires. The cutline of the 14th agricultural vote (corresponding to the 58th actual vote) from the 90th Senate with ideal points is plotted below in Figure 2, showing that the vote largely broke down along partisan lines.

```
R> par(mfrow = c(1, 1))
R> plot.coords(result, cutline = 58)
```

4. W-NOMINATE with an arbitrary vote matrix

This section describes an example of **wnominate** being used for roll call data not already in `.ord` format. The example here is drawn from the first three sessions of the United Nations, discussed further as Figure 5.8 from Poole (2005). To create a `rollcall` object for use with **wnominate**, one ideally should have at least three things:

- `data`: A matrix of votes from some source. The matrix should be arranged as a legislators × votes matrix. Previous versions of **wnominate** required that Yeas be coded

as 1, Nays as 6, and missing votes as 9. In this package however, Yeas, Nays and missing votes can be coded arbitrarily.

- `legis.names`: A vector of names for each member in the vote matrix.

- Optional Argument: A vector describing the party or party-like memberships for the legislator, to be stored in `legis.data` as the variable `party`.

The **wnominate** package includes all three of these items for the United Nations, which can be loaded and browsed with the code shown below. The data comes from Eric Voeten at George Washington University. In practice, one would prepare a roll call data set in a spreadsheet, like the one available at http://www.voteview.com/UN.csv, and read it into R using `read.csv`. Here, we simply access the local copy of the data on the package:

```
R> rm(list = ls(all = TRUE))
R> data("UN")
R> UN <- as.matrix(UN)
R> UN[1:5, 1:6]

  V1                 V2        V3  V4  V5  V6
1 "United States"  "Other"  "1"  "6"  "6"  "6"
2 "Canada"         "Other"  "6"  "6"  "6"  "6"
3 "Cuba"           "Other"  "1"  "6"  "1"  "1"
4 "Haiti"          "Other"  "1"  "6"  "6"  "9"
5 "Dominican Rep"  "Other"  "1"  "6"  "6"  "7"
```

Observe that the first column are the names of the legislators (in this case, countries), and the second column lists whether a country is a "Warsaw Pact" country or "Other", which in this case can be thought of as a 'party' variable. All other observations are votes. Our objective here is to use this data to create a `rollcall` object through the `rollcall` function in **pscl**. The object can then be used with **wnominate** and its `plot`/`summary` functions as in the previous `.ord` example.

To do this, we want to extract a vector of names (`UNnames`) and party memberships (`party`), then delete them from the original matrix so we have a matrix of nothing but votes. The `party` variable must be rolled into a matrix as well for inclusion in the `rollcall` object as follows:

```
R> UNnames <- UN[, 1]
R> legData <- matrix(UN[, 2], length(UN[, 2]), 1)
R> colnames(legData) <- "party"
R> UN <- UN[, -c(1, 2)]
```

In this particular vote matrix, Yeas are numbered 1, 2, and 3, Nays are 4, 5, and 6, abstentions are 7, 8, and 9, and 0s are missing. Other vote matrices are likely different so the call to `rollcall` will be slightly different depending on how votes are coded. Party identification is included in the function call through `legData`, and a `rollcall` object is generated and applied to **wnominate** as follows. The result is summarized below and plotted in Figure 3:

W−NOMINATE Coordinates

Cutting Line Angles
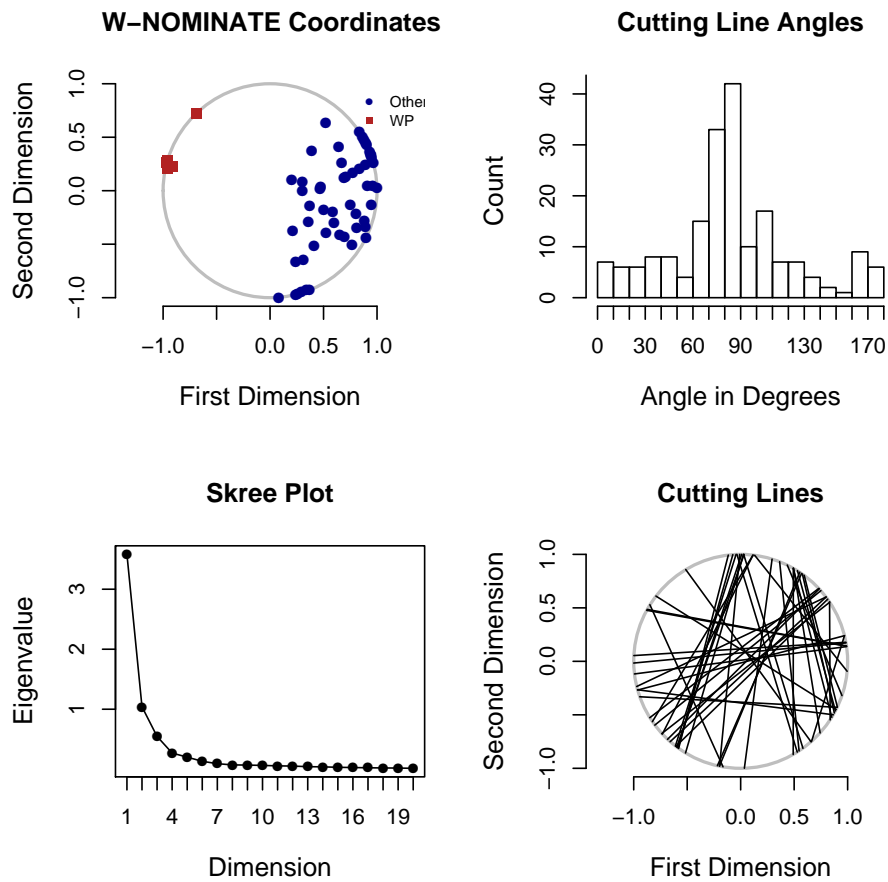
Skree Plot

Cutting Lines

Figure 3: Summary plot of UN data.

```
R> rc <- rollcall(UN, yea = c(1, 2, 3), nay = c(4, 5, 6),
+    missing = c(7, 8, 9), notInLegis = 0, legis.names = UNnames,
+    legis.data = legData, desc = "UN Votes",
+    source = "http://www.voteview.com")
R> result <- wnominate(rc, polarity = c(1, 1))

Preparing to run W-NOMINATE...

       Checking data...

              All members meet minimum vote requirements.

              Votes dropped:
              ... 18 of 237 total votes dropped.

       Running W-NOMINATE...
```

```
                    Getting bill parameters...
                    Getting legislator coordinates...
                    Starting estimation of Beta...
                    Getting bill parameters...
                    Getting legislator coordinates...
                    Starting estimation of Beta...
                    Getting bill parameters...
                    Getting legislator coordinates...
                    Getting bill parameters...
                    Getting legislator coordinates...
                    Estimating weights...
                    Getting bill parameters...
                    Getting legislator coordinates...
                    Estimating weights...
                    Getting bill parameters...
                    Getting legislator coordinates...


W-NOMINATE estimation completed successfully.
W-NOMINATE took 3.451 seconds to execute.


R> summary(result)


SUMMARY OF W-NOMINATE OBJECT
----------------------------

Number of Legislators:           59 (0 legislators deleted)
Number of Votes:             219 (18 votes deleted)
Number of Dimensions:         2
Predicted Yeas:                  4677 of 5039 (92.8%) predictions correct
Predicted Nays:                  4133 of 4488 (92.1%) predictions correct
Correct Classifiction:        89.57% 92.47%
APRE:                         0.576 0.695
GMP:                          0.783 0.84


The first 10 legislator estimates are:
              coord1D coord2D
United States   0.942   0.334
Canada          0.932   0.361
Cuba            0.525  -0.397
Haiti           0.370  -0.140
Dominican Rep   0.804  -0.219
Mexico          0.468   0.018
Guatemala       0.389   0.376
Honduras        0.597  -0.296
```

```
El Salvador      0.899  -0.438
Nicaragua        0.882  -0.285
```

```
R> plot(result)
```

5. Natural experiments with W-NOMINATE

In addition to analyzing pre-gathered roll call data, the **wnominate** package in R can also be used to conduct natural experiments using roll calls. These roll call experiments have been used extensively in the literature to test for effects on first dimension W-NOMINATE scores from events such as party switching, redistricting, and last-period effects (Nokken and Poole 2004; Goodman 2004). The **wnominate** package simplifies this procedure into two important ways. First, existing R functions in the base package can be leveraged to merge roll call data sets for experimentation more easily than before. Secondly, **wnominate** incorporates a bootstrapping function that allows uncertainty estimates from the experiments to be derived easily.

We demonstrate this new functionality with an example. In this roll call experiment, we are interested in determining the identities of the most erratic senators who served throughout the 105th-108th US Senate. By "erratic" in this context, we mean the legislator whose voting record has varied by the widest gap along their first dimension W-NOMINATE score. Following the logic described in Chapter 6 of Poole (2005), we conduct this experiment by treating the legislator as four separate individuals – one during the 105th Senate, another during the 106th Senate, and so on. By examining the shift in ideal points for each legislator between the four legislatures, we are able to gain some insight into this problem.

More specifically, we begin by generating a list of all senators who served throughout the entire 105th-108th Senate, with 70 senators meeting this criteria. Taking the 105th Senate as an example to begin the estimation, we then take one of the 70 senators and isolate their voting record for that Senate before merging the data sets of the four Senates together. Taking Senator John McCain as an example, we would estimate an ideal point for McCain's voting in the 105th Congress, an ideal point for his voting from the 106th to 108th Congresses, and an ideal point for every other senator who has ever served in the 105th-108th Senates. The idea behind this estimation is to allow McCain's ideal point in the 105th to shift independently, while all other senators are held fixed. The ideal point for McCain's voting in the 105th Senate is recorded, and the process repeated for the 106th-108th Senates, as well as for all legislators.

Once the entire legislature has been estimated, each of the 70 senators who served throughout the entire 105th-108th now have four separate ideal points corresponding to each separate Congress. The shift for each legislator is then computed simply as the difference between their maximum and minimum W-NOMINATE scores, and standard errors for that difference can be obtained simply as the standard error of a difference $\sigma_{a-b} = \sqrt{\sigma_a^2 + \sigma_b^2}$ (assuming independence).[6]

---

[6]The assumption of independence is clearly conservative, as the covariance between estimates is almost certainly positive (thus deflating the standard error). Even with a conservative confidence interval, however, we are able to reject the probability of 0 shift, as shown in the example.

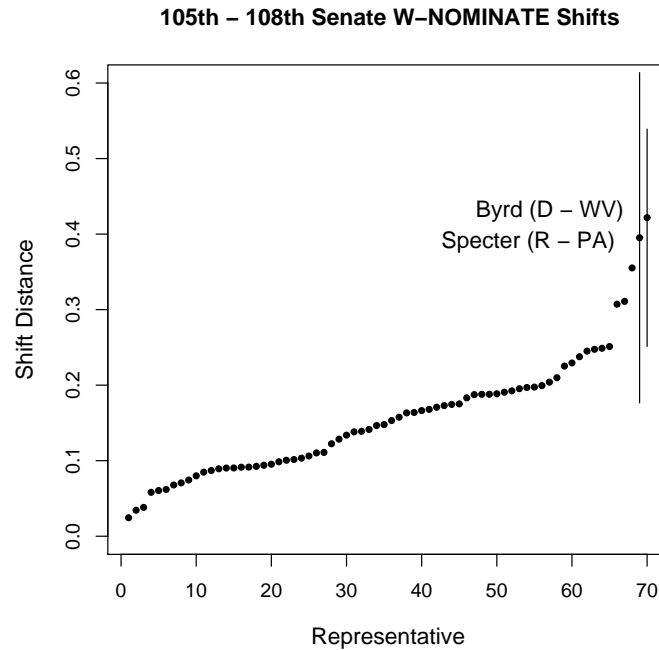**105th – 108th Senate W–NOMINATE Shifts**

Figure 4: Summary plot of 104th to 108th Senate shifts.

The example can be summarized in the pseudocode below:

```
for(i from 1 to 70 senators) {
      for(j from 105th to 108th Senate) {
            Make Senator_{ij} a separate individual
            Merge all the roll call data from j = 105:108
            Estimate the new roll call matrix,
            recording the ideal point of Senator_{ij}
      }
}

 Shift(Senator_i) = max(Senator_{ij}) - min(Senator_{ij})
```

Figure 4 represents the results of this roll call experiment in a graphical manner. For each of the N = 70 senators who served throughout the entire 105th-108th term, the figure plots the rank of that individual's shift against their actual shift. The results suggest that two senators in particular shifted their ideal points by a particularly large amount: Senators Robert Byrd (D–WV) and Arlen Specter (R–PA). Byrd shifted 0.422 units to the left (-0.478 to -0.900) between the 106th to the 108th Congress, while Specter shifted 0.395 units to the right (-0.085 to 0.310) between the 105th to the 108th Congress. This contrasts with a mean shift of 0.157 units for all senators in the sample, with a standard deviation for the shifts of 0.081. The 95% per cent confidence intervals shown with the Byrd and Specter ideal point shifts suggest that these shifts are statistically significant. After Byrd and Specter, the senators exhibiting the largest shift in their ideal points are (in descending order) Bob Graham (D–FL), John Breaux (D–LA), and Max Baucus (D–MT).

```
R> data <- read.csv("senators.csv")
R> plot(rank(data$shift), data$shift, ylim = c(0, 0.6),
+    main = "105th - 108th Senate W-NOMINATE Shifts",
+    xlab = "Representative", ylab = "Shift Distance", cex.main = 1.2,
+    cex.lab = 1.2, font.main = 2, pch = 20)
R> seByrd <- sqrt(0.06366407^2 + 0.03651869^2)
R> seSpecter <- sqrt(0.104247^2 + 0.03969758^2)
R> lines(c(69, 69), c(data[66,2] - 1.96 * seSpecter,
+    data[66,2] + 1.96 * seSpecter))
R> lines(c(70, 70), c(data[66,2] - 1.96 * seByrd,
+    data[66,2] + 1.96 * seByrd))
R> text(57, 0.43, "Byrd (D - WV)", cex = 1.2)
R> text(54, 0.39, "Specter (R - PA)", cex = 1.2)
```

6. Conclusion

W-NOMINATE has been a salient part of political science research for over three decades, contributing significantly to our understanding of how roll call votes evolve. However, it is our belief that more remains to be done, particularly in applying W-NOMINATE to the study of legislative voting outside the United States. This research agenda has been hindered in the past by the need to use .ord-formatted data, the lack of "canned" graphical functions, and difficulties in manipulating data sets. It is our hope that the release of this package into the public domain can eliminate some of these barriers and increase interest in the study of roll call voting.

Acknowledgments

References

Cahoon LS, Hinich MJ, Ordeshook PC (1976). "A Multidimensional Statistical Procedure for Spatial Analysis." Carnegie-Mellon University.

Clinton J, Jackman SD, Rivers D (2004). "The Statistical Analysis of Roll Call Data: A Unified Approach." *American Political Science Review*, **98**(2), 355–370.

Converse PE (1964). "The Nature of Belief Systems in Mass Publics." In *Ideology and Discontent*. Free Press, New York.

Enelow JM, Hinich M (1984). *The Spatial Theory of Voting.* Cambridge University Press, New York.

Goodman C (2004). *Ideological Constraint in Congress: Experiments in Roll Call Voting.* Ph.D. thesis, University of Houston.

Heckman JJ, Snyder JM (1997). "Linear Probability Models of the Demand for Attributes With an Empirical Application to Estimating the Preferences of Legislators." *Rand Journal of Economics*, **28**, 142–189.

Hinich MJ, Munger MJ (1994). *Ideology and the Theory of Political Choice.* University of Michigan Press, Ann Arbor.

Hinich MJ, Munger MJ (1997). *Analytical Politics.* Cambridge University Press, Ann Arbor.

Hinich MJ, Pollard W (1981). "A New Approach to the Spatial Theory of Electoral Competition." *American Journal of Political Science*, **25**(2), 323–341.

Jackman S (2010). ***pscl**: Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University.* Department of Political Science, Stanford University, Stanford, California. R package version 1.03.6, URL http://CRAN.R-project.org/package=pscl.

Lowell AL (1902). "The Influence of Party Upon Legislation in England and America." *Annual Report of the American Historical Association*, **1**, 321–542.

MacRae D (1958). *Dimensions of Congressional Voting: A Statistical Study of the House of Representatives in the Eighty-First Congress.* University of California Press, Berkeley.

Masket S (2007). "It Takes An Outsider: Extra-Legislative Organization And Partisanship In The California Assembly, 1849-2006." *American Journal of Political Science*, **51**(3), 482–497.

McCarty NM, Poole KT, Rosenthal H (2001). "The Hunt for Party Discipline in Congress." *American Political Science Review*, **95**(3), 673–687.

McCarty NM, Poole KT, Rosenthal H (2006). *Polarized America: The Dance of Ideology and Unequal Riches.* MIT Press, Cambridge.

McFadden D (1973). "Conditional Logit Analysis of Qualitatative Choice Behavior." In *Frontiers of Economics.* Academic Press, New York.

Nokken T, Poole KT (2004). "Congressional Party Defection in American History." *Legislative Studies Quarterly*, **29**(4), 545–68.

Ordeshook PC (1976). "The Spatial Theory of Elections: A Review and a Critique." In *Party Identification and Beyond.* John Wiley & Sons, New York.

Poole K, Rosenthal H, Shor B (2000). ***Voteview** for Windows: Software to Manipulate Roll Call Data.* Princeton, NJ. Version 3.01, URL http://www.princeton.edu/~voteview/.

Poole KT (2000). "Non-Parametric Unfolding of Binary Choice Data." *Political Analysis*, **8**(3), 211–237.

Poole KT (2005). *Spatial Models of Parliamentary Voting.* Cambridge University Press, Cambridge.

Poole KT, Rosenthal H (1985). "A Spatial Model for Legislative Roll Call Analysis." *American Journal of Political Science*, **29**(2), 357–84.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Voeten E (2001). "Outside Options and the Logic of Security Council Action." *American Political Science Review*, **95**(4), 845–858.

A. Code for natural experiment

Load the data and package:

```
R> library("wnominate")
R> data <- list()
R> for (k in 105:108)
+    data[[k]] <- readKH(sprintf("ftp://voteview.com/sen%ikh.ord", k))
```

The following function is used to merge roll call matrices for estimation: `member` is a string, the name of the member, and `legis` is the legislature of member to be isolated.

```
R> buildbigmatrix <- function(data, member, legis) {
+    index < -which(rownames(data[[legis]]$votes) == member)
+    rownames(data[[legis]]$votes)[index] <- paste(member, legis)
+    merged <- data[[105]]$votes
+    for (i in 106:108) {
+      merged <- merge(merged, data[[i]]$votes, by = 0, all = TRUE)
+      rownames(merged) <- merged$Row.names
+      merged$Row.names <- NULL
+      merged[is.na(merged)] <- 9
+    }
+    return(merged)
+  }
```

Here, we only calculate Byrd's coordinate for the 108th Congress. Add a list of all senators as "members" if all senators are desired (remove comments). Change `legis` to `legs <- 105:108` if all legislatures are desired. Warning: even this single calculation takes a very long time.

```
R> a <- intersect(rownames(data[[105]]$votes), rownames(data[[106]]$votes))
R> b <- intersect(rownames(data[[107]]$votes), rownames(data[[108]]$votes))
R> members <- intersect(a, b)
R> members <- "BYRD (D WV)"
R> legis <- 105
R> coords <- matrix(nrow = length(members), ncol = length(legis) + 1)
R> rownames(coords) <- members
R> colnames(coords) <- c(legis, "shift")
```

If standard errors are desired, include this code to create storage for SEs. Again, this increases computation time significantly. Also, be sure to uncomment the other SE-related messages below.

```
R> errors <- coords
```

Now conduct the estimation, setting `trials = 10` if standard errors are desired:

```
R> for(i in 1:length(members)) {
+    for(j in 1:length(legis)) {
```

```
+        votes<-buildbigmatrix(data, members[i], legis[j])
+        rm(partycodes, state.info)
+        pol <- which(rownames(votes) == "FRIST (R TN)")
+        dat <- rollcall(votes, yea = 1:3, nay = 4:6, missing = 7:9,
+          notInLegis = 0, legis.names = rownames(votes))
+        result <- wnominate(dat, dims = 1, polarity = pol, trials = 1,
+          lop = 0.025)
+        coords[i, j] <- result$legislators[paste(members[i],legis[j]),
+          "coord1D"]
+
+        errors[i, j] <- result$legislators[paste(members[i],legis[j]), "se1D"]
+        rm(votes, pol, result)
+      }
+    }
```

If all four legislatures are calculated, legislator shift can be obtained as follows:

```
R> coords[,"shift"] <- apply(coords, 1, max, na.rm = TRUE) -
+    apply(coords, 1, min, na.rm = TRUE)
```

Affiliation:

Keith T. Poole
Department of Political Science
Baldwin Hall
University of Georgia
Athens, GA 30602, United States of America
E-mail: kpoole@uga.edu
URL: http://www.voteview.com/

Jeffrey B. Lewis
Department of Political Science
University of California, Los Angeles
Box 951472
4289 Bunche Hall
Los Angeles, CA 90095, United States of America
E-mail: jblewis@polisci.ucla.edu
URL: http://www.sscnet.ucla.edu/polisci/faculty/lewis/

James Lo
University of Mannheim, SFB 884
L13, 15–17
68131 Mannheim, Germany
E-mail: lo@uni-mannheim.de

Royce Carroll
Department of Political Science, MS 24
Rice University
PO Box 1892
Houston, Texas 77251-1892, United States of America
E-mail: rcarroll@rice.edu
URL: http://rcarroll.web.rice.edu/