Using W-NOMINATE in R

James Lo

January 30, 2018

1 Introduction

This package estimates Poole and Rosenthal W-NOMINATE scores from roll call votes supplied though a `rollcall` object from package `pscl`.[1] The R version of W-NOMINATE computes ideal points using the same Fortran code base as the previous *wnom9707()* software. 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, the `wnominate` package includes a full suite of graphics functions to analyze the results.

W-NOMINATE scores are based on the spatial model of voting. Let s denote the number of policy dimensions, which are indexed by k=*1, . . . , s*; let p denote the number of legislators (i=*1,. . . ,p*); and q denote the number of roll call votes (j=*1,. . . ,q*). Let legislator i's ideal point be x_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:

$$U_{ijy} = \beta exp[-\frac{\sum_{k=1}^{s} w_k^2 d_{ijyk}^2}{2}] + \epsilon_{ijy}$$

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,

[1]Production of this package is supported by NSF Grant SES-0611974.

$$d^2_{ijy} = \sum_{k=1}^{s}(x_{ik} - z_{jyk})^2$$

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 all other dimensions, the corresponding w_k is estimated, with the starting value of w_k set at 0.5 each time.

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

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

where z_{mj} is the midpoint and $d_j = \frac{z_{jy} - z_{jn}}{2}$.

2 Usage Overview

The `wnominate` package was designed for use in one of three ways. First, users can estimate ideal points from a set of Congressional roll call votes stored in the traditional *ORD* file format. Secondly, users can generate a vote matrix of their own, and feed it directly into `wnominate` for analysis. Finally, users can also generate test data with ideal points and bill parameters arbitrarily specified as arguments by the user for analysis with `wnominate`. Each of these cases are supported by a similar sequence of function calls, as shown in the diagrams below:

$\texttt{\textit{ORD} file} \xrightarrow{\texttt{readKH()}} \texttt{\textit{rollcall} object} \xrightarrow{\texttt{wnominate()}} \texttt{\textit{wnominate} object}$
$\texttt{Vote matrix} \xrightarrow{\texttt{rollcall()}} \texttt{\textit{rollcall} object} \xrightarrow{\texttt{wnominate()}} \texttt{\textit{wnominate} object}$
$\texttt{Arguments} \xrightarrow{\texttt{generateTestData()}} \texttt{\textit{rollcall} object} \xrightarrow{\texttt{wnominate()}} \texttt{\textit{wnominate} object}$

Following generation of a `wnominate` object, the user then analyzes the results using the *plot* and *summary* methods, 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 cutlines (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 each of the three cases described here are presented in the following sections.

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 format *ORD* format for all U.S. Congresses are stored online for download at:

- *http://www.voteview.com/*
- *http://www.polisci.ucla.edu/faculty/lewis/rollcall/* (latest Congress only, updates votes in real time)

`wnominate` takes `rollcall` objects from Simon Jackman's `pscl` package as input. The package includes a function, *readKH()*, that takes an *ORD* file and automatically transforms it into a `rollcall` object as desired. Refer to the documentation in `pscl` for more detailed information on *readKH()* and *rollcall()*. Using the 90th Senate as an example, we can download the file *sen90kh.ord* and read the data in R as follows:

```
> library(wnominate)
> #sen90 <- readKH("ftp://voteview.com/sen90kh.ord")
> data(sen90)      #Does same thing as above
> sen90
```

```
Source:                 C:/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.
```

To make this example more interesting, suppose we were interested in applying *wnominate()* only to bills that pertained in some way to agriculture. Keith Poole and Howard Rosenthal's VOTEVIEW software allows us to quickly determine which bills in the 90th Senate pertain to agriculture.[2] Using this information, we create a vector of roll calls that we wish to select, then select for them in the `rollcall` object. In doing so, we should also take care to update the variable in the `rollcall` object that counts the total number of bills, as follows:

```
> selector <- c(21,22,44,45,46,47,48,49,50,53,54,55,56,58,59,60,61,62,65,66,67,6
> sen90$m <- length(selector)
> sen90$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 U.S. 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*

[2]VOTEVIEW for Windows can be downloaded at *www.voteview.com*.

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.

```
> 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)"

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

Preparing to run W-NOMINATE...

        Checking data...

                ... 1 of 102 total members dropped.

                Votes dropped:
                ... 36 of 208 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 17.412 seconds to execute.
```

`result` now contains all of the information from the W-NOMINATE estimation, the details of which are fully described in the documentation for *wnominate()*. `result$legislators` contains all of the information from the `nom31.dat` file from the old Fortran *wnom9707()*, while `result$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):

> *legisdata <- result$legislators*
> *fix(legisdata)*

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

```
> summary(result)

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

Number of Legislators:          101 (1 legislators deleted)
Number of Votes:          172 (36 votes deleted)
Number of Dimensions:         2
Predicted Yeas:                 6549 of 7732 (84.7%) predictions correct
Predicted Nays:                 6427 of 7570 (84.9%) predictions correct
Correct Classifiction:          79.79% 84.8%
APRE:                         0.366 0.523
GMP:                         0.652 0.708
```

```
The first 10 legislator estimates are:
                 coord1D coord2D
JOHNSON (D USA)   -0.486  -0.145
SPARKMAN (D AL)    0.374   0.759
HILL (D AL)        0.590   0.714
GRUENING (D AK)   -0.703   0.711
BARTLETT (D AK)   -0.533   0.846
HAYDEN (D AZ)      0.154   0.884
FANNIN (R AZ)      0.806  -0.244
FULBRIGHT (D AR)   0.175   0.437
MCCLELLAN (D AR)   0.810   0.295
KUCHEL (R CA)     -0.200  -0.291
```

`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 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. Observe that with agriculture votes, party affiliation does not appear to be a strong predictor on the first dimension, although the second dimension is largely divided by party line. The cutting angle histogram shows that most votes are well classified by a single dimension (i.e. 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.

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 and are covered in the example in the final section. 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

```
> plot(result)

NULL
```

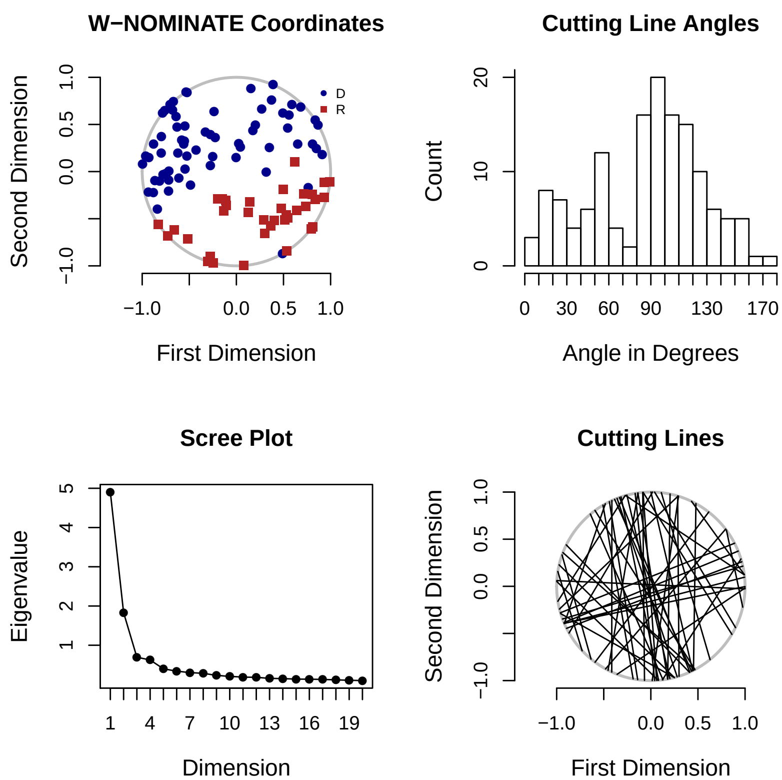

Figure 1: Summary Plot of 90th Senate Agriculture Bill W-NOMINATE Scores

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.

4 W-NOMINATE with arbitrary vote matrix

This section describes an example of W-NOMINATE 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 in Keith Poole's *Spatial Models of Parliamentary Voting*.

To create a `rollcall` object for use with *wnominate()*, one ideally should have three things:

- A matrix of votes from some source. The matrix should be arranged as a *legislators* x *votes* matrix. It need not be in 1/6/9 or 1/0/NA format, but users must be able to distinguish between Yea, Nay, and missing votes.
- A vector of names for each member in the vote matrix.
- OPTIONAL: A vector describing the party or party-like memberships for the legislator.

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 one *www.voteview.com/UN.csv*, and read it into R using *read.csv()*. The csv file is also stored in this package and can be read using:

UN<-read.csv("library/wnominate/data/UN.csv",header=FALSE,strip.white=TRUE)

The line above reads the exact same data as what is stored in this package as R data, which can be obtained using the following commands:

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

```
> par(mfrow=c(1,1))
> plot.coords(result,cutline=14)
```

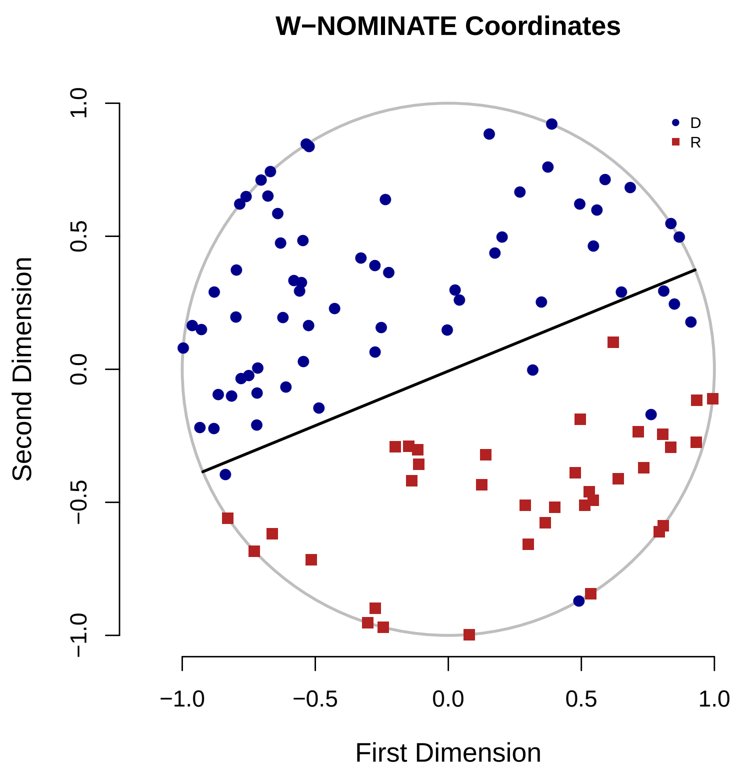

Figure 2: 90th Senate Agriculture Bill W-NOMINATE Scores with Cutline

```
  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:

```
> UNnames<-UN[,1]
> legData<-matrix(UN[,2],length(UN[,2]),1)
> colnames(legData)<-"party"
> 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 W-NOMINATE as follows. The result is summarized below and plotted in Figure 3:

```
> 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="www.voteview.com")
> 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 17.643 seconds to execute.

> 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:                  4692 of 5039 (93.1%) predictions correct
Predicted Nays:                  4126 of 4488 (91.9%) predictions correct
Correct Classifiction:          89.49% 92.56%
APRE:                          0.573 0.698
GMP:                          0.783 0.841


The first 10 legislator estimates are:
              coord1D coord2D
United States   0.939   0.344
Canada          0.932   0.362
Cuba            0.519  -0.387
Haiti           0.362  -0.131
Dominican Rep   0.796  -0.224
Mexico          0.459   0.026
Guatemala       0.381   0.363
Honduras        0.588  -0.267
El Salvador     0.887  -0.461
Nicaragua       0.876  -0.302
```

5 W-NOMINATE with Test Data

The W-NOMINATE package includes a test data generator that can be used to generate arbitrary vote matrices. These functions are typically only used for testing purposes, and full documentation of them is included with the package. The two functions are:

- *nomprob()*: A function that yields a matrix of probabilities of a Yea vote.
- *generateTestData()*: A function that calls nomprob and generates a `rollcall` object.

nomprob() takes a matrix of Yea and Nay locations, along with a matrix of ideal points, and generates the vote probability matrix. In this example,

```
> plot(result)
```

```
NULL
```

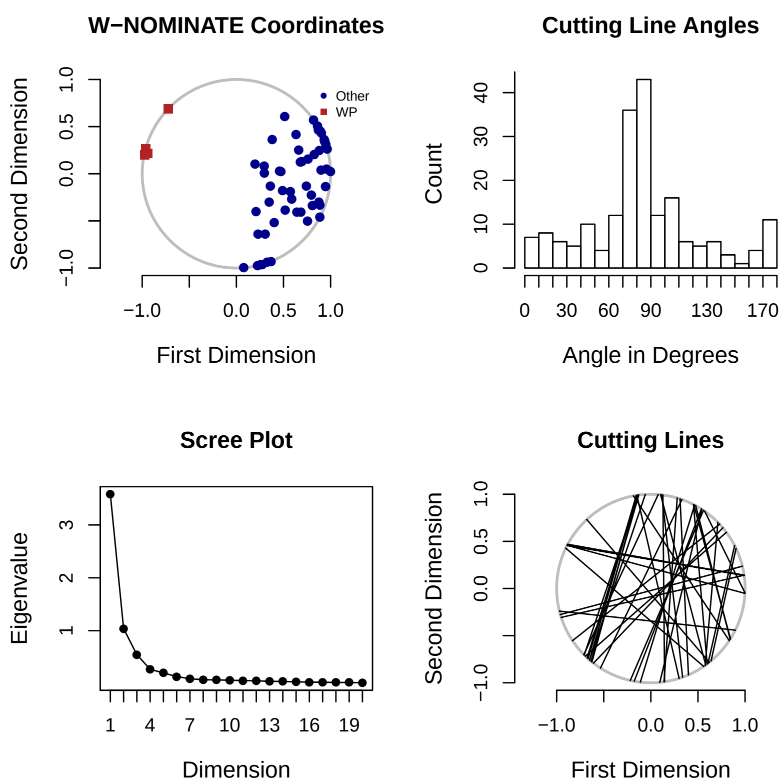

Figure 3: Summary Plot of UN Data

the Yea positions on 6 bills is set at 0.3, and the Nay position is set at -0.2. Bill parameters can be set in as many dimensions as desired, but since the input matrices in this example have only one column, the the bills are predominantly one-dimensional, as are the ideal points. For the ideal points, the first 5 legislators are set to have ideal points of -0.2 while the last 5 are set to have ideal points of 0.2. This setup leads to the last 5 legislators having a high probability of voting Yea on all of the bills, as confirmed in the example, and vice versa. The beta and weights can also be specified, and in this example we set them to be the *wnominate()* default values of 15 and 0.5. Both normal and logistic link functions can be used to generate the probabilities, although the default is to use normal probabilities.

```
> yp <- matrix(rep(0.3,6),nrow=6)
> np <- matrix(rep(-0.2,6),nrow=6)
> ideal <- matrix(c(rep(-0.2,5),rep(0.2,5)),nrow=10)
> nomprob(yp,np,ideal,15,0.5)
```

```
            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
 [1,] 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851
 [2,] 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851
 [3,] 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851
 [4,] 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851
 [5,] 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851 0.03898851
 [6,] 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172
 [7,] 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172
 [8,] 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172
 [9,] 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172
[10,] 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172 0.85958172
```

generateTestData() generates a full rollcall object using *nomprob()*, and in fact passes most of its arguments to it. A totally random set of parties and states are assigned to the legislators, so splits by party and region are not substantively meaningful. The function is set up so that users need only specify the number of legislators and roll call votes they wish to generate, and by default it generates a one dimensional model, as is shown here in Figure 4:

```
> dat <- generateTestData(legislators=100, rcVotes=1000)
> result <- wnominate(dat,polarity=1,dims=1)
```

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

        Checking data...

                All members meet minimum vote requirements.

                All votes meet minimum lopsidedness requirement.

        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...


W-NOMINATE estimation completed successfully.
W-NOMINATE took 74.926 seconds to execute.

> summary(result)

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

Number of Legislators:          100 (0 legislators deleted)
Number of Votes:          1000 (0 votes deleted)
Number of Dimensions:          1
Predicted Yeas:                  43747 of 48866 (89.5%) predictions correct
Predicted Nays:                  45728 of 51134 (89.4%) predictions correct
Correct Classifiction:          89.47%
APRE:                          0.757
GMP:                          0.793
```

```
The first 10 legislator estimates are:
             coord1D se1D
Legislator1    1.000    0
Legislator2    0.672    0
Legislator3    0.305    0
Legislator4    0.267    0
Legislator5   -1.000    0
Legislator6   -0.153    0
Legislator7   -0.252    0
Legislator8   -0.289    0
Legislator9    1.000    0
Legislator10  -0.453    0
```

Note that the one dimensional plot differs considerably from the previous two dimensional plots, since only a coordinate plot and a Skree plot are shown. This is because in one dimension, all cutlines are angled at 90°, so there is no need to plot either the cutlines or a histogram of cutline angles. Also, the plot appears to be compressed, so users need to expand the image manually by using their mouse and dragging along the corner of the plot to expand it. The Skree plot confirms that a single dimension captures the voting patterns accurately, as seen by the flatness of the curve from the second dimension on.

```
> plot(result)
```

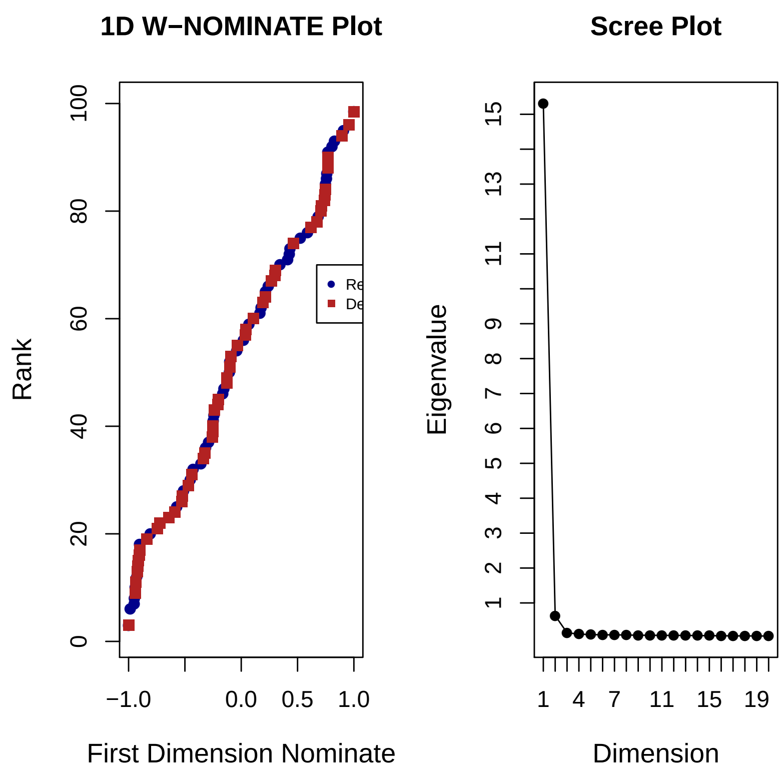

Figure 4: Summary Plot of Randomly Generated Data

References

[1] Lewis, Jeffrey B. and Poole, Keith (2004) "Measuring Bias and Uncertainty in Ideal Point Estimates via the Parametric Bootstrap." *Political Analysis* 12(2): 105-127.

[2] Poole, Keith (2005) *Spatial Models of Parliamentary Voting.* Cambridge: Cambridge University Press.

[3] Poole, Keith and Howard Rosenthal (1997) *Congress: A Political-Economic History of Roll Call Voting.* New York: Oxford University Press.