

DW-Nominate Parametric Bootstrapped SEs

Luke Sonnet

3/12/2018

This document walks through some of the results from a parametric bootstrap of DW-NOMINATE standard errors. These data are generated by generating 200 bootstrap samples, where votes on rollcalls for each member are regenerated using the estimated probability the legislator voted yea on the vote. On each bootstraped sample, only the ideal points were updated.

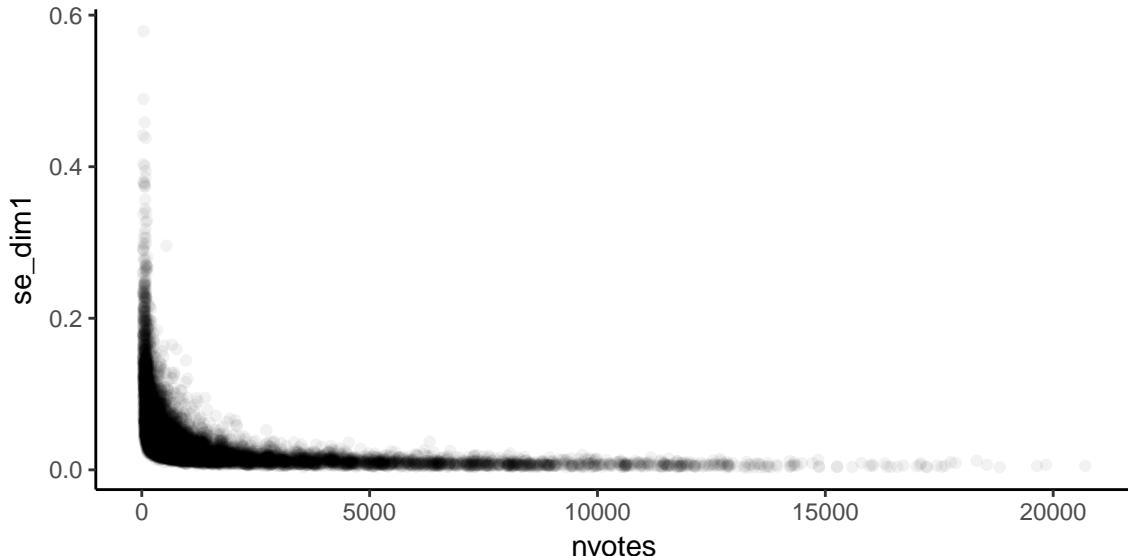
The dataset below includes the bootstrapped standard errors, the average of the bootstrap ideal points, the actual ideal points used in the estimation (pulled off of Voteview.com on March 10, 2018), the Pearson correlation across the two dimensions, and some other metadata such as the total number of valid votes and the last congress the member served in. I'll drop those who voted fewer than 25 times as we do not estimate ideal points for them.

```
library(dplyr)
library(ggplot2)
bt <- read.csv("boot.csv", stringsAsFactors = FALSE) %>%
  filter(nvotes >= 25)
```

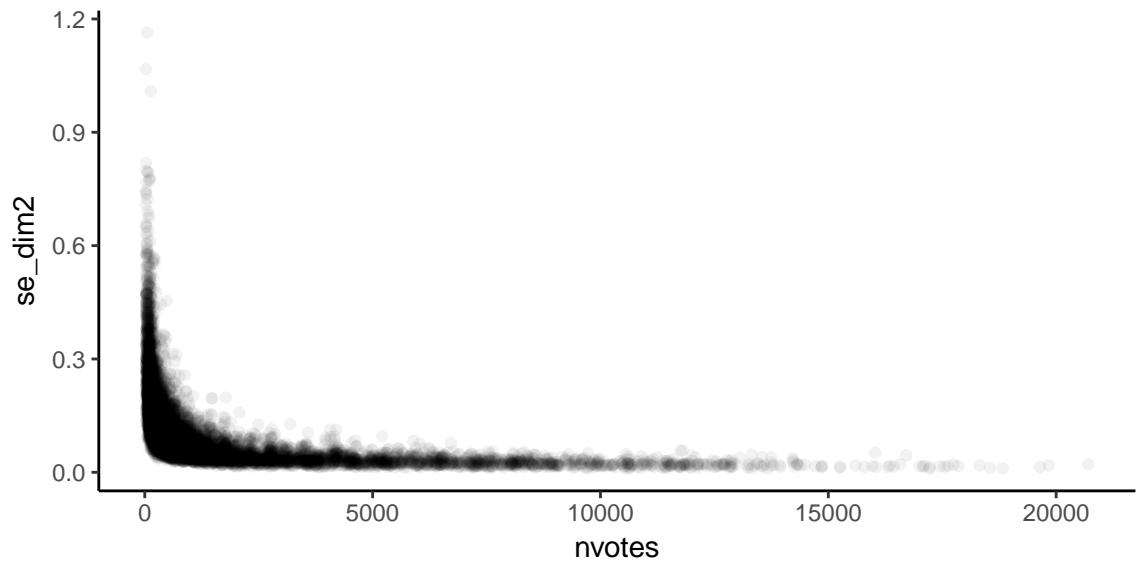
Sanity check

Now let's see how the standard error of each dimension tracks with the number of votes cast.

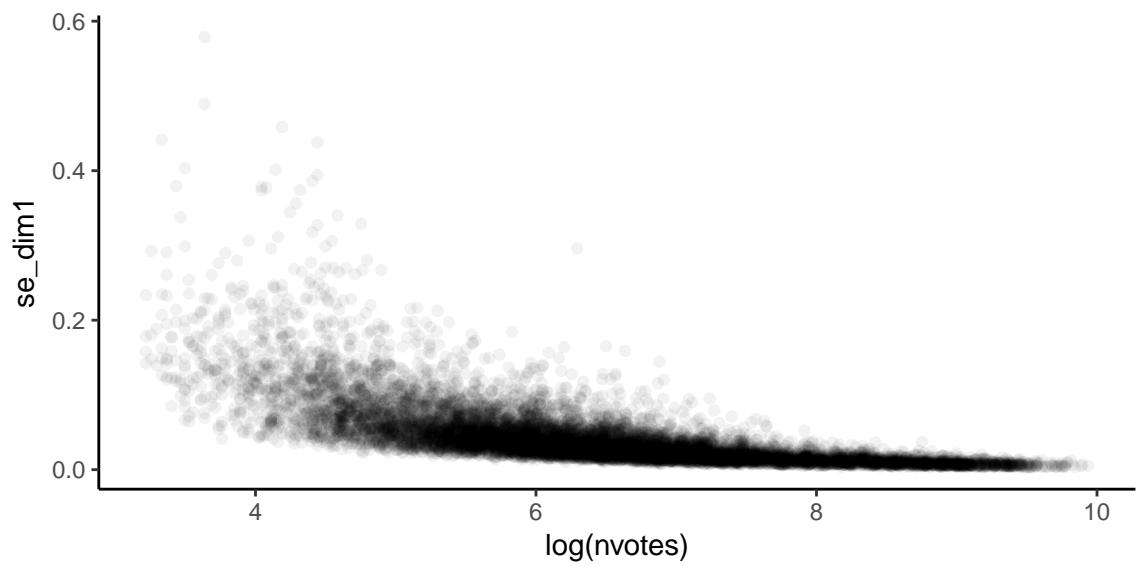
```
ggplot(bt, aes(x = nvotes, y = se_dim1)) + geom_point(alpha = 0.05) + theme_classic()
```



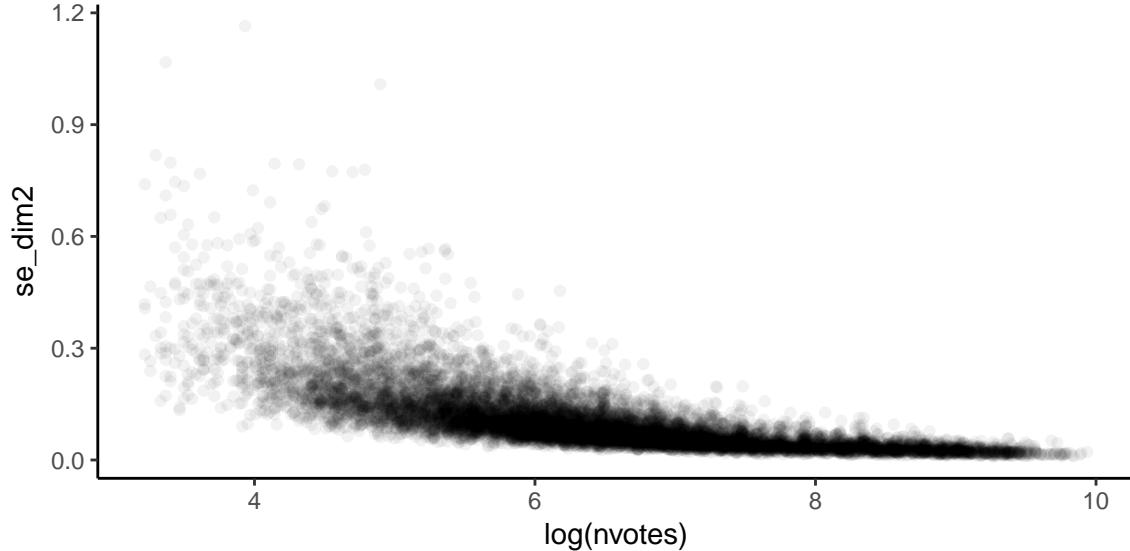
```
ggplot(bt, aes(x = nvotes, y = se_dim2)) + geom_point(alpha = 0.05) + theme_classic()
```



```
ggplot(bt, aes(x = log(nvotes), y = se_dim1)) + geom_point(alpha = 0.05) + theme_classic()
```



```
ggplot(bt, aes(x = log(nvotes), y = se_dim2)) + geom_point(alpha = 0.05) + theme_classic()
```



Makes sense, the votes track pretty cleanly with the number of votes. We can also use some regressions to get a peak.

```
summary(lm(cbind(se_dim1, se_dim2) ~ nvotes + last_congress, data = bt))
```

```
## Response se_dim1 :
##
## Call:
## lm(formula = se_dim1 ~ nvotes + last_congress, data = bt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04828 -0.02117 -0.00979  0.00860  0.51971
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.233e-02 7.493e-04 83.19 <2e-16 ***
## nvotes     -5.118e-06 1.720e-07 -29.76 <2e-16 ***
## last_congress -2.151e-04 1.309e-05 -16.43 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0368 on 12101 degrees of freedom
## Multiple R-squared:  0.1695, Adjusted R-squared:  0.1693
## F-statistic: 1234 on 2 and 12101 DF, p-value: < 2.2e-16
##
##
## Response se_dim2 :
##
## Call:
## lm(formula = se_dim2 ~ nvotes + last_congress, data = bt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04828 -0.02117 -0.00979  0.00860  0.51971
##
```

```

## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.233e-02  7.493e-04   83.19 <2e-16 ***
## nvotes        -5.118e-06  1.720e-07  -29.76 <2e-16 ***
## last_congress -2.151e-04  1.309e-05  -16.43 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0368 on 12101 degrees of freedom
## Multiple R-squared:  0.1695, Adjusted R-squared:  0.1693
## F-statistic:  1234 on 2 and 12101 DF,  p-value: < 2.2e-16

```

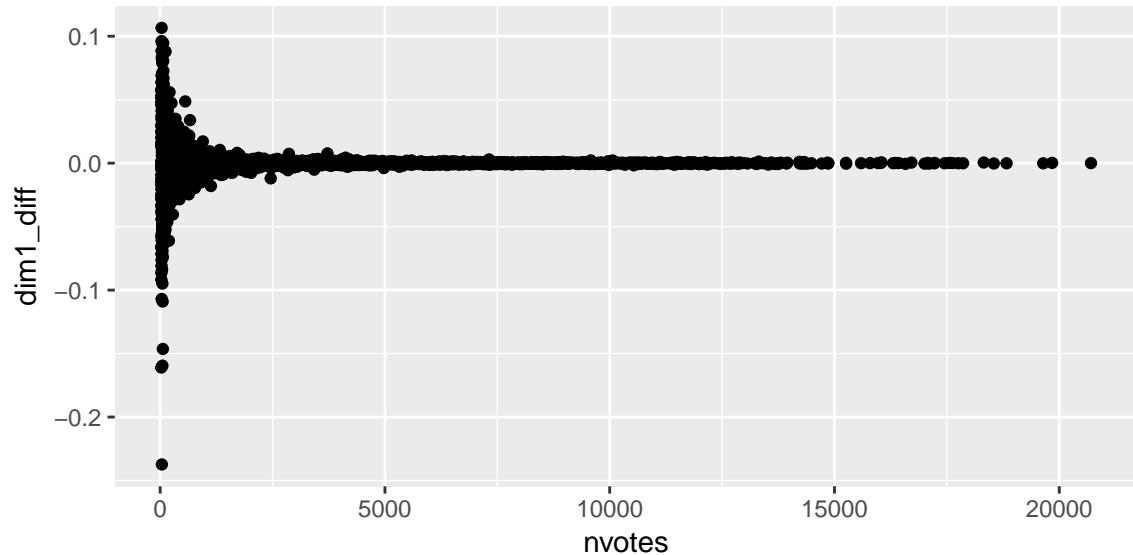
Bias

Let's see if there is any bias from fitting old points that could rotate:

```

# The diff here is our ML ideal point minus the mean of the bootstrapped idpt
bt %>%
  ggplot(aes(x = nvotes, y = dim1_diff)) + geom_point()

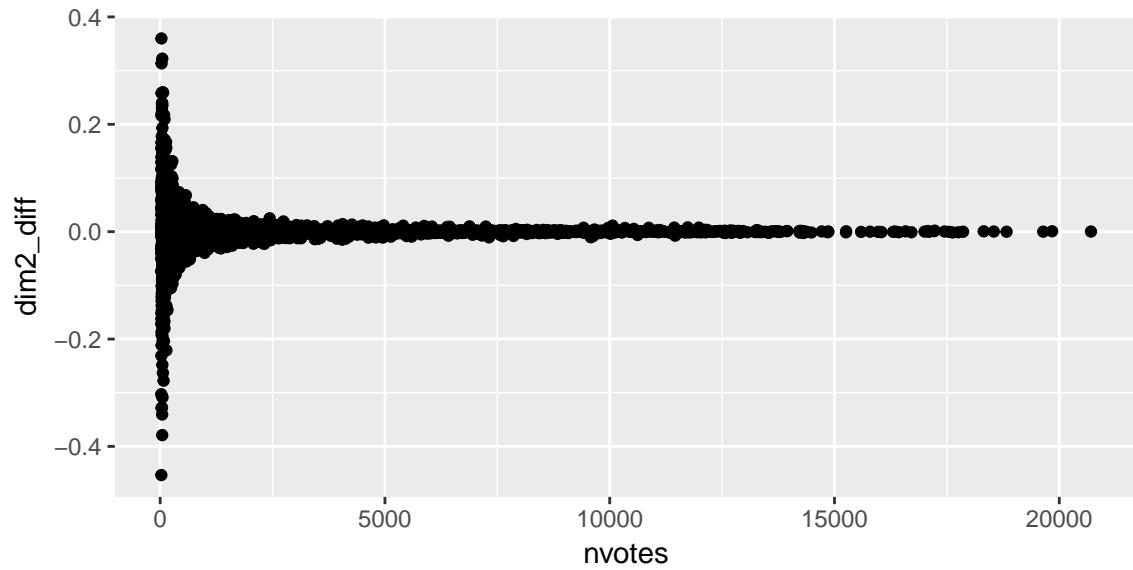
```



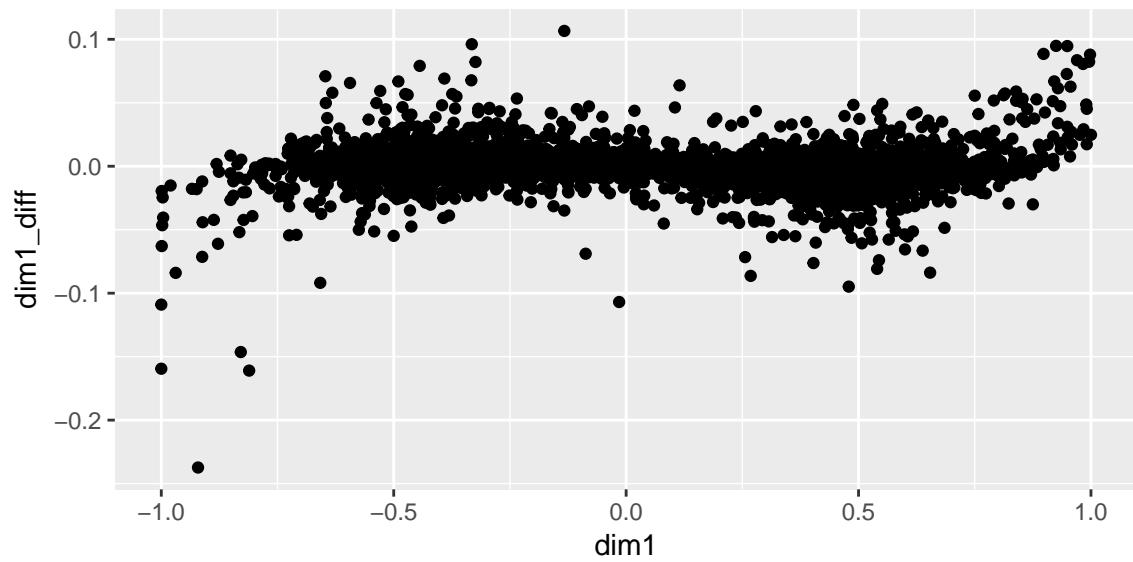
```

bt %>%
  ggplot(aes(x = nvotes, y = dim2_diff)) + geom_point()

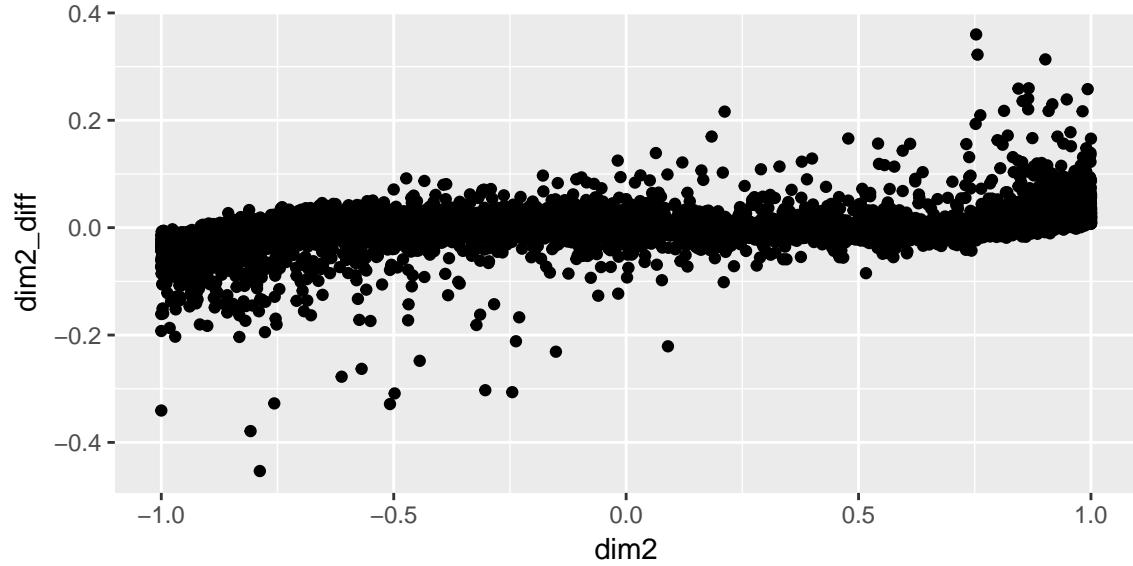
```



```
bt %>%  
  ggplot(aes(x = dim1, y = dim1_diff)) + geom_point()
```



```
bt %>%  
  ggplot(aes(x = dim2, y = dim2_diff)) + geom_point()
```



There is definitely some movement towards the middle on the average, away from the edges.

Bias does go down with the number of votes and if the votes were cast later.

```
lm(cbind(dim1_diff, dim2_diff) ~ nvotes + last_congress, data = bt)
```

```
##
## Call:
## lm(formula = cbind(dim1_diff, dim2_diff) ~ nvotes + last_congress,
##      data = bt)
##
## Coefficients:
##              dim1_diff    dim2_diff
## (Intercept) -7.643e-04 -5.023e-04
## nvotes       6.892e-08  1.256e-07
## last_congress 3.826e-06 -2.390e-06
lm(cbind(abs_dim1_diff, abs_dim2_diff) ~ nvotes + last_congress, data = bt)

##
## Call:
## lm(formula = cbind(abs_dim1_diff, abs_dim2_diff) ~ nvotes + last_congress,
##      data = bt)
##
## Coefficients:
##          abs_dim1_diff  abs_dim2_diff
## (Intercept) 7.252e-03   1.930e-02
## nvotes     -6.358e-07  -1.826e-06
## last_congress -3.201e-05 -7.394e-05
```