Brainless Calibration

Sometimes it is useful to compare set-analytic results with linear model results. When doing so it is often wise to use the means and standard deviation of the source variables to create the parallel fuzzy sets. Making them parallel helps to alleviate concerns that differences in findings are due to the fuzzy set calibration procedure.

The first step in brainless calibration is to convert the source variable to z scores. For example, Lijphart's data on democracies includes the variable "effective number of parliamentary parties." ("Parliamentary" is used generically to refer to representative bodies.)

```
summarize parties

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     parties |        36    3.191667    1.137876       1.38        5.2
```

To create z scores. it is necessary simply to subtract the mean from each raw score and divide this difference by the standard deviation.

There is a shortcut in Stata, using egen:

egen float partiesz = std(parties), mean(0) std(1)

Now you have a variable that ranges from -3 to +3, roughly, with most scores between -1.5 and +1.5.

It is possible to treat this metric as a quasi-log odds metric and convert it to a quasi-probability, which has the same metric as a fuzzy membership score. However, in its current form (with most scores ranging from -1.5 to +1.5), the resulting membership scores would be too tightly clustered around 0.5. So I recommend multiplying the z scores by a constant. In this exercise I use constant = 2.

In Stata you would issue the following command:

```
replace partiesz = 2*partiesz
```

Now the variable ranges from -6 to +6, with most scores in the -3 to +3 range. Recall that the calibration procedure in fsQCA uses +3 as the threshold for full membership and -3 as the threshold for full nonmembership.

Now it is in a form that warrants treating it as a quasi-log odds.

To convert it to a quasi-probability (i.e., the same metric as a fuzzy set), apply the standard formulae for converting logits to probabilities:

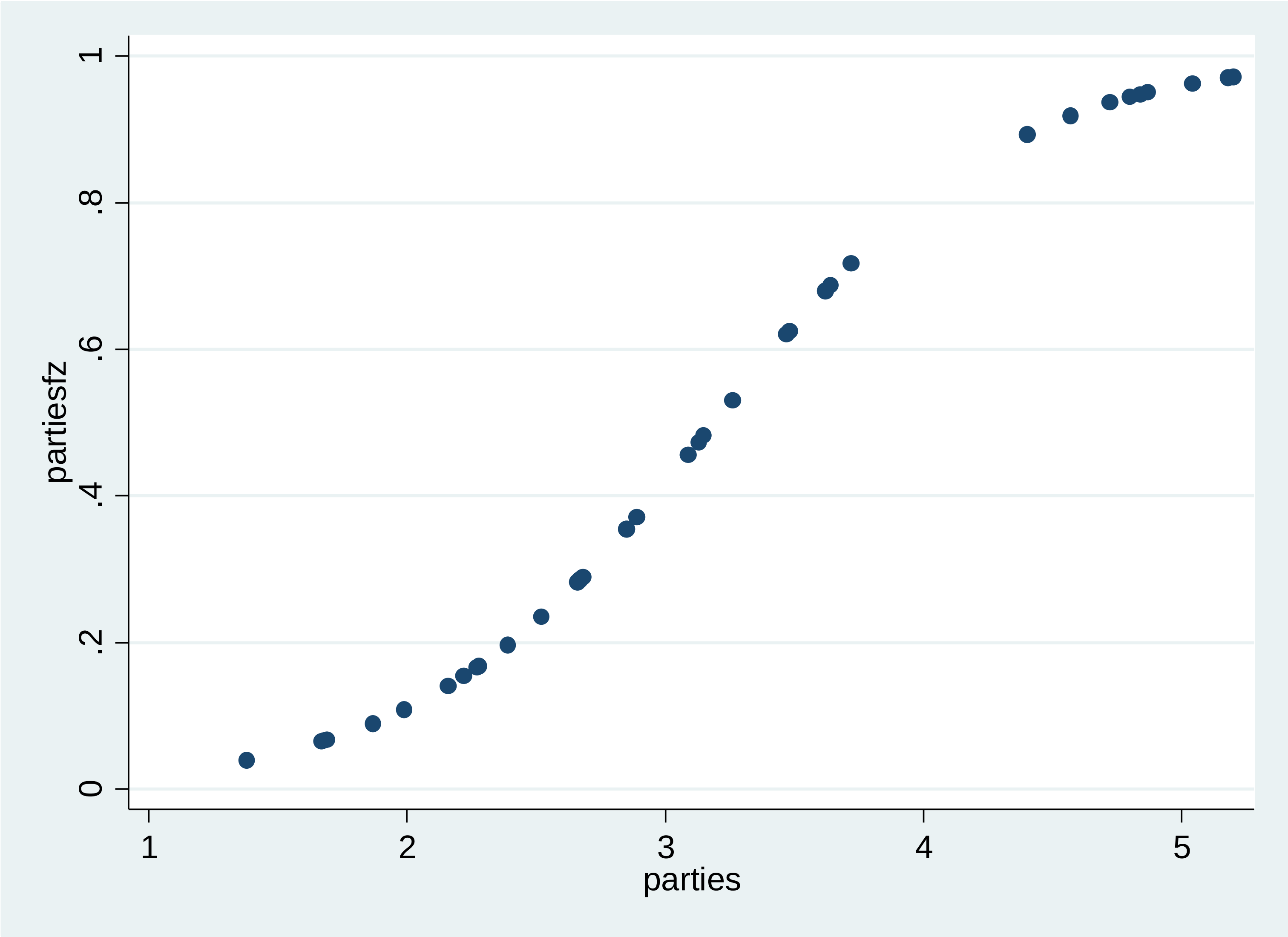
```
gen partiesfz = exp(partiesz)/(1+exp(partiesz))
```

The logit is exponentiated to odds and then divided by 1 + the odds.

Here's a spreadsheet showing the transformations. Notice that the mean of parties (3.2) is 0 in the z metric and 0.5 in the fuzzy metric. The fuzzy set could be interpreted as degree of membership in a multiparty political system.

Country	parties	2*partiesz	partiesfz
BOT	1.38	-3.18429	0.039761
JAM	1.67	-2.67457	0.064491
BAR	1.68	-2.657	0.065559
BAH	1.69	-2.63942	0.066644
TRI	1.87	-2.32304	0.089233
MAL	1.99	-2.11212	0.107924
UK	2.16	-1.81332	0.140237
AUL	2.22	-1.70786	0.153442
GRE	2.27	-1.61998	0.165208
NZ	2.28	-1.6024	0.167646
US	2.39	-1.40906	0.196383
CAN	2.52	-1.18056	0.234951
SPA	2.66	-0.93449	0.282015
CR	2.67	-0.91691	0.285587
AUT	2.68	-0.89934	0.289187
KOR	2.85	-0.60053	0.354222
MAU	2.85	-0.60053	0.354222
IRE	2.89	-0.53023	0.370464
GER	3.09	-0.1787	0.455445
POR	3.13	-0.10839	0.472929
ARG	3.15	-0.07324	0.481699
FRA	3.26	0.120107	0.529991
SWE	3.47	0.489215	0.619922
LUX	3.48	0.506792	0.624054
JPN	3.62	0.752864	0.679802
NOR	3.64	0.788018	0.687406
ICE	3.72	0.92863	0.716797
URU	4.4	2.123839	0.893199
DEN	4.57	2.422642	0.918538
BEL	4.72	2.68629	0.936213
IND	4.8	2.826904	0.944112
ITA	4.84	2.89721	0.947708
NET	4.87	2.949939	0.950261
FIN	5.04	3.248742	0.962628
ISR	5.18	3.494814	0.97054
SWI	5.2	3.529967	0.971528

And here's the scatterplot:



The correlation is 0.9893.