Online appendix

Simulating party shares

Denis Cohen Chris Hanretty

A1 Cases included in the data and distribution of N_{S0} and N_{V0}

The list of countries included in the data is reported as Table A1.1; the frequencies of different values of N_{S0} and N_{V0} are reported in Table A1.2. Note that the number of seat-winning parties can, on occasion, be larger than the number of vote-winning parties because of "independents" who win seats but who are not reported in the lists of vote-winning parties.

Table A1.1: Elections included in the analysis

Country	Elections	Earliest	Latest
Australia	46	1901-03-30	2019-05-18
Austria	28	1919-02-16	2019-09-29
Belgium	38	1900-05-27	2019-05-26
Bulgaria	9	1991-10-13	2017-03-26
Canada	35	1900-11-07	2019-10-21
Croatia	7	2000-01-03	2020-07-05
Cyprus	10	1976-09-05	2021-05-30
Czech Republic	9	1990-06-09	2017-10-21
Denmark	46	1901-04-03	2019-06-05
Estonia	8	1992-09-20	2019-03-03
Finland	30	1917-10-02	2019-04-14
France	29	1902-05-11	2017-06-18
Germany	28	1919-01-19	2017-09-24
Greece	18	1974-11-17	2019-07-07
Hungary	8	1990-04-08	2018-04-08
Iceland	32	1919-11-15	2017-10-28
Ireland	31	1922-06-22	2020-02-08
Israel	23	1949-01-25	2020-03-02
Italy	19	1946-06-02	2018-03-04
Japan	27	1946-04-10	2017-10-22
Latvia	10	1990-04-29	2018-10-06
Lithuania	9	1990-03-10	2020-10-11
Luxembourg	24	1919-10-26	2018-10-14
Malta	18	1947-10-27	2017-06-03
Netherlands	28	1918-07-03	2017-03-15
New Zealand	39	1902-11-25	2020-10-17
Norway	32	1900-07-01	2017-09-11
Poland	10	1989-06-18	2019-10-13
Portugal	16	1975-04-25	2019-10-06
Romania	9	1990-05-20	2020-12-06
Slovakia	10	1990-06-09	2020-02-29
Slovenia	9	1990-04-12	2018-06-03
Spain	15	1977-06-15	2019-11-10
Sweden	34	1911-09-24	2018-09-09
Switzerland	33	1902-10-26	2019-10-20
Turkey	11	1983-11-06	2018-06-24
United Kingdom	28	1918-12-14	2019-12-12

Table A1.2: Elections with this many ...

N_0	Seat-winning parties	Vote-winning parties
2	28	1
3	49	40
4	87	53
5	128	109
6	117	121
7	97	90
8	64	88
9	61	59
10	44	74
11	42	58
12	33	36
13	28	33
14	13	15
15	12	18
16	5	9
17	3	6
18	3	1
19		1
20	2	1

A2 R package

We provide an R package, `sharesimulatoR`, which can be used to simulate party systems of different sizes. For example: the following code simulates a system with five seat-winning parties:

```r
if (!require(sharesimulatoR)) {
    devtools::install_github("chrishanretty/sharesimulatoR")
}

sims <- simulate_shares(5,
                        what = "seat-winning parties",
                        basis = "Posterior draws of alpha",
                        n_sim = 1000,
                        seed = 25)
head(sims)
```

```
##           [,1]      [,2]      [,3]       [,4]        [,5]
## [1,] 0.3543476 0.3237846 0.2652073 0.04594701 0.010713508
## [2,] 0.5324663 0.2140941 0.1662906 0.07503400 0.012115017
## [3,] 0.5488410 0.2483659 0.1264265 0.07135274 0.005013934
## [4,] 0.3161309 0.2385102 0.2835152 0.12403142 0.037812270
## [5,] 0.4719239 0.3185325 0.1440487 0.04678374 0.018711092
## [6,] 0.3219415 0.2843948 0.1645433 0.14127280 0.087847635
```

Users can supply their own values of alpha if they wish to adjust the precision of the estimates. For example: the following code simulate a system with five seat-winning parties, but with greater variation in party shares (since the concentration parameter α is smaller):

```r
sims <- simulate_shares(5,
                        what = "seat-winning parties",
                        basis = "Freely chosen stochastic alpha",
                        n_sim = 1000,
                        alpha = 30,
                        sd_alpha = 5,
                        seed = 25)
head(sims)
```

```
##           [,1]      [,2]       [,3]       [,4]        [,5]
## [1,] 0.4929119 0.3157156 0.15278755 0.03330252 0.005282441
## [2,] 0.3246160 0.4078130 0.17708343 0.08953676 0.000950807
## [3,] 0.3874909 0.3986882 0.09334301 0.08324753 0.037230336
## [4,] 0.6107953 0.1830553 0.15057098 0.03977732 0.015801015
## [5,] 0.2435688 0.4328033 0.30379325 0.01441943 0.005415183
## [6,] 0.4269170 0.2872121 0.21915392 0.05003723 0.016679720
```

A3 Generative model for simulating shares from an ordered Dirichlet distribution

In this section, we explain how we simulate the marginal proportions (i.e., seat or vote shares) in a party system of a given size K using a vector of concentration parameters $\theta^{\star}$ estimated from our ordered Dirichlet models.

As explained in the main text, we estimate $\theta^{\star}$ as the product of a unit simplex of location parameters for the $K+1$ differences describing a party system of size K, $\mathbf{p}^{\star}$, and a general scale parameter α. Conversely, we can recover $\mathbf{p}^{\star}$ and α from $\theta^{\star}$ as follows:

$$\begin{aligned}\alpha &= \sum_{i=1}^{K+1} \theta_i^{\star} \\ \mathbf{p}^{\star} &= \frac{\theta^{\star}}{\alpha}\end{aligned}$$

Following Dorp and Mazzuchi (2004), we take the cumulative sum for each element i of $\mathbf{p}^{\star}$,

$$\mathbf{p_i}^{+} = \sum_{k=1}^{i} \mathbf{p_i}^{\star}$$

and use the scalar scale parameter α and the transformed location parameters $\mathbf{p}^{+}$ to simulate each marginal proportion y_i from a reparameterized Beta distribution (cf. Dorp and Mazzuchi (2004), Eqs. 1 and 5). Mapping this reparameterization back onto the conventional parameterizatzion of the Beta distribution with two vectors of positive shape parameters, Beta(a, b), yields

$$\begin{aligned}a_i &= \alpha \mathbf{p_i}^{+} \\ b_i &= \alpha(1 - \mathbf{p_i}^{+}).\end{aligned}$$

We then simulate the share y_i for party i, $i \in \{1, ..., K\}$ as $y_i \sim \text{Beta}(a_i, b_i)$. This procedure is implemented in the user-supplied function `orddir_rng()` in the Stan code below.

A4 Code

A4.1 Stan code used to fit models with deterministic $p/p^{\star}$

A4.1.1 Unordered Dirichlet

Note: Code used for null, logical, and political unordered Dirichlet models.

```
## data {
##   int<lower=1> nobs;
##   int<lower=1> max_K;
##   array[nobs] int<lower=1, upper = max_K> K;
##   matrix[nobs, max_K] y;
##   matrix[max_K, max_K] p;
## }
##
## parameters {
##    real<lower=0> theta;
## }
##
## model {
##    for (i in 1:nobs) {
##       segment(y[i, ], 1, K[i]) ~ dirichlet(theta * segment(p[K[i], ], 1, K[i]));
##    }
##    // Prior on theta
##    theta ~ lognormal(2.5, 0.75);
## }
##
## generated quantities {
##   // Structured this way to match output of dirichlet_rng
##   matrix[max_K, nobs] repy;
##   vector [nobs] log_lik;
##   vector [nobs] N_s;
##
##   // Pre-populate with zeros
##   repy = rep_matrix(0.0, max_K, nobs);
##
##   // Add on the predictions
##   for (i in 1:nobs) {
##     repy[1:K[i], i] = dirichlet_rng(theta * segment(p[K[i], ], 1, K[i])');
##   }
##
##   // Calculate N_s
##   for (i in 1:nobs) {
##      N_s[i] = 1.0 / sum(pow(repy[1:K[i], i], 2.0));
##   }
##   for (i in 1:nobs) {
##     log_lik[i] = dirichlet_lpdf(segment(y[i, ], 1, K[i]) | theta * segment(p[K[i], ], 1, K[i]));
##   }
## }
```

A4.1.2 Ordered Dirichlet

Note: Code used for logical and political ordered Dirichlet models.

```
## functions {
##   array[] real orddir_rng(vector theta) {
##      // See van Dorp and Mazzuchi 2003, section IV
##      int mp1 = rows(theta);
##      int m = mp1 - 1;
##      real beta = sum(theta);
##      vector[mp1] alpha = theta / beta;
##      vector[m] alpha_plus = cumulative_sum(alpha[1:m]);
##      array[m] real y = beta_rng(beta * alpha_plus,
##                                 beta * (1.0 - alpha_plus));
##      return(y);
##   }
## }
##
## data {
##   int<lower=1> nobs;
##   int<lower=1> max_K;
##   array[nobs] int<lower=1, upper = max_K> K;
##   matrix[nobs, max_K] y;
##   matrix[max_K, max_K + 1] p;
## }
##
## transformed data {
##   int<lower=1> max_Kp1 = max_K + 1;
##   matrix[nobs, max_Kp1] deltas;
##
##   deltas = rep_matrix(0.0, nobs, max_Kp1);
##   deltas[, 1] = y[, 1];
##   for (i in 1:nobs) {
##     for (j in 2:K[i]) {
##      deltas[i, j] = y[i, j] - y[i, j - 1];
##     }
##     deltas[i, K[i] + 1] = 1.0 - y[i, K[i]];
##   }
## }
##
## parameters {
##    real<lower=0> theta;
## }
##
## model {
##    for (i in 1:nobs) {
##       segment(deltas[i, ], 1, K[i] + 1) ~
##         dirichlet(theta * segment(p[K[i], ], 1, K[i] + 1));
##    }
##    // Prior on theta
##    theta ~ lognormal(2.5, 0.75);
## }
##
## generated quantities {
##   // Structured this way to match output of dirichlet_rng
##   matrix[max_K, nobs] repy;
##   vector [nobs] log_lik;
##   vector [nobs] N_s;
##
##   // Pre-populate with zeros
##   repy = rep_matrix(0.0, max_K, nobs);
##
##   // Add on the predictions
##   for (i in 1:nobs) {
##     array[K[i]] real tmp;
##     tmp = orddir_rng(theta * segment(p[K[i], ], 1, K[i] + 1)');
##     for (m in 1:K[i])
##       repy[m, i] = tmp[m];
##   }
##
##   // Calculate N_s
##   for (i in 1:nobs) {
##      N_s[i] = 1.0 / sum(pow(repy[1:K[i], i], 2.0));
##   }
##
##   for (i in 1:nobs) {
##     log_lik[i] = dirichlet_lpdf(segment(deltas[i, ], 1, K[i] + 1)' |
##        theta * segment(p[K[i], ], 1, K[i] + 1)');
##   }
## }
```

A4.2 Stan code used to fit models with estimated $p/p^\star$

A4.2.1 Unordered Dirichlet

Note: Code used for saturated unordered Dirichlet models.

```
## data {
##   int<lower=1> nobs;
##   int<lower=1> max_K;
##   array[nobs] int<lower=1, upper = max_K> K;
##   matrix[nobs, max_K] y;
## }
##
## parameters {
##    real<lower=0> theta;
##    vector[max_K] eta[max_K];
## }
##
## transformed parameters {
##   matrix[max_K, max_K] p = rep_matrix(0.0, max_K, max_K);
##   for (i in 1:max_K) {
##     p[i, 1:i] = softmax(segment(eta[i], 1, i))';
##   }
## }
##
## model {
##    for (i in 1:nobs) {
##       segment(y[i, ], 1, K[i]) ~ dirichlet(theta * segment(p[K[i], ], 1, K[i]));
##    }
##    // Prior on theta
##    theta ~ lognormal(2.5, 0.75);
## }
##
## generated quantities {
##   // Structured this way to match output of dirichlet_rng
##   matrix[max_K, nobs] repy;
##   vector [nobs] log_lik;
##   vector [nobs] N_s;
##
##   // Pre-populate with zeros
##   repy = rep_matrix(0.0, max_K, nobs);
##
##   // Add on the predictions
##   for (i in 1:nobs) {
##     repy[1:K[i], i] = dirichlet_rng(theta * segment(p[K[i], ], 1, K[i])');
##   }
##
##   // Calculate N_s
##   for (i in 1:nobs) {
##      N_s[i] = 1.0 / sum(pow(repy[1:K[i], i], 2.0));
##   }
##   for (i in 1:nobs) {
##     log_lik[i] = dirichlet_lpdf(segment(y[i, ], 1, K[i]) | theta * segment(p[K[i], ], 1, K[i]));
##   }
## }
```

A4.2.2 Ordered Dirichlet

Note: Code used for saturated ordered Dirichlet models.

```
## functions {
##   array[] real orddir_rng(vector theta) {
##      // See van Dorp and Mazzuchi 2003, section IV
##      int mp1 = rows(theta);
##      int m = mp1 - 1;
##      real beta = sum(theta);
##      vector[mp1] alpha = theta / beta;
##      vector[m] alpha_plus = cumulative_sum(alpha[1:m]);
##      array[m] real y = beta_rng(beta * alpha_plus,
##                                 beta * (1.0 - alpha_plus));
##      return(y);
##   }
## }
##
## data {
##   int<lower=1> nobs;
##   int<lower=1> max_K;
##   array[nobs] int<lower=1, upper = max_K> K;
##   matrix[nobs, max_K] y;
## }
##
## transformed data {
##   int<lower=1> max_Kp1 = max_K + 1;
##   matrix[nobs, max_Kp1] deltas;
##
##   deltas = rep_matrix(0.0, nobs, max_Kp1);
##   deltas[, 1] = y[, 1];
##   for (i in 1:nobs) {
##     for (j in 2:K[i]) {
##      deltas[i, j] = y[i, j] - y[i, j - 1];
##     }
##     deltas[i, K[i] + 1] = 1.0 - y[i, K[i]];
##   }
## }
##
## parameters {
##    real<lower=0> theta;
##    vector[max_Kp1] eta[max_K];
## }
##
## transformed parameters {
##   matrix[max_K, max_Kp1] p = rep_matrix(0.0, max_K, max_Kp1);
##   for (i in 1:max_K) {
##     int ip1 = i + 1;
##     p[i, 1:ip1] = softmax(segment(eta[i], 1, ip1))';
##   }
## }
##
## model {
##    for (i in 1:nobs) {
##       segment(deltas[i, ], 1, K[i] + 1) ~
##         dirichlet(theta * segment(p[K[i], ], 1, K[i] + 1));
##    }
##    // Prior on theta
##    theta ~ lognormal(2.5, 0.75);
##    // Prior on eta
##    for (i in 1:max_K)
##      eta[i] ~ normal(0, 1);
## }
##
## generated quantities {
##   // Structured this way to match output of dirichlet_rng
##   matrix[max_K, nobs] repy;
##   vector [nobs] log_lik;
##   vector [nobs] N_s;
##
##   // Pre-populate with zeros
##   repy = rep_matrix(0.0, max_K, nobs);
##
##   // Add on the predictions
##   for (i in 1:nobs) {
##     array[K[i]] real tmp;
##     tmp = orddir_rng(theta * segment(p[K[i], ], 1, K[i] + 1)');
##     for (m in 1:K[i])
##       repy[m, i] = tmp[m];
##   }
##
##   // Calculate N_s
##   for (i in 1:nobs) {
##      N_s[i] = 1.0 / sum(pow(repy[1:K[i], i], 2.0));
##   }
##
##   for (i in 1:nobs) {
##     log_lik[i] = dirichlet_lpdf(segment(deltas[i, ], 1, K[i] + 1)' |
##        theta * segment(p[K[i], ], 1, K[i] + 1)');
##   }
## }
```

A5 Observed versus predicted values

A5.1 Seat shares

A5.1.1 Null model, Dirichlet

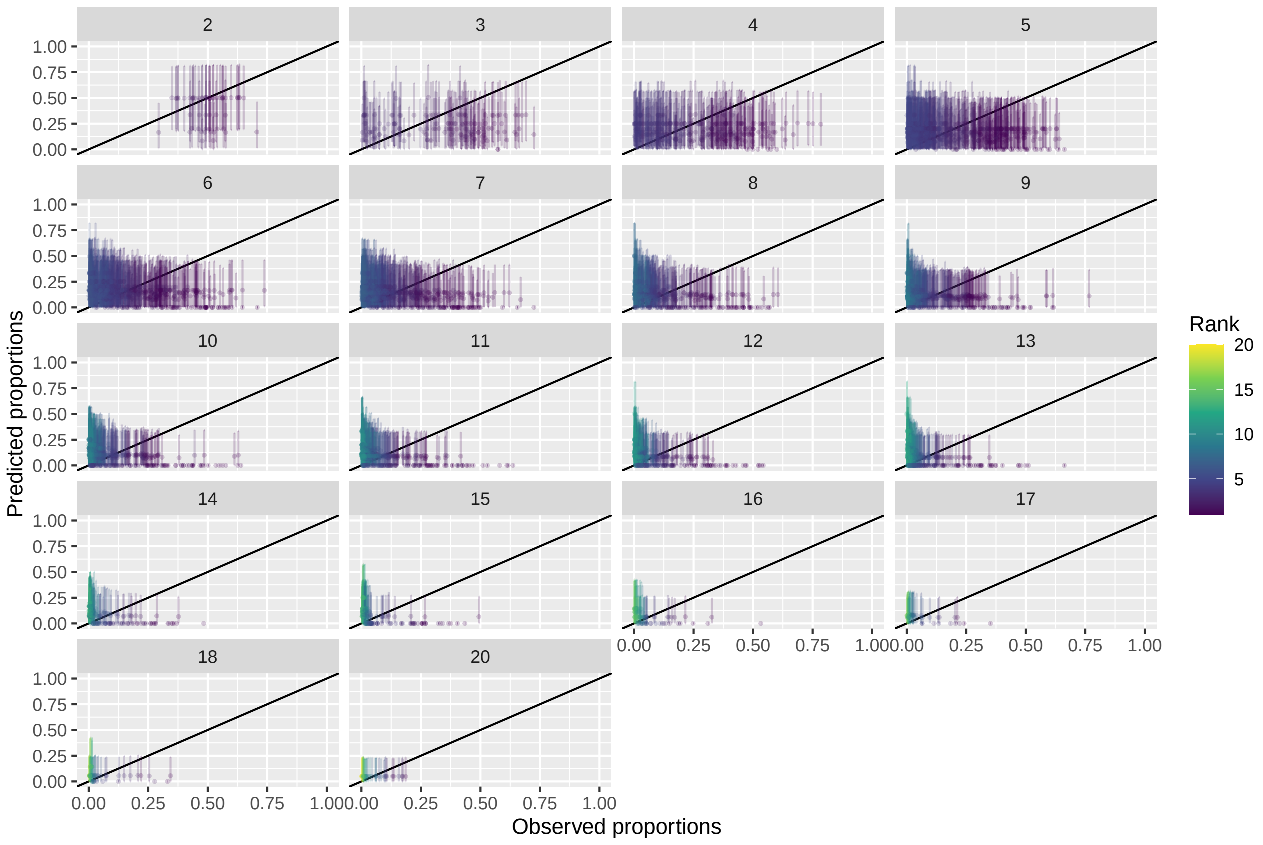

Figure A5.1: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.2 Logical model, Dirichlet

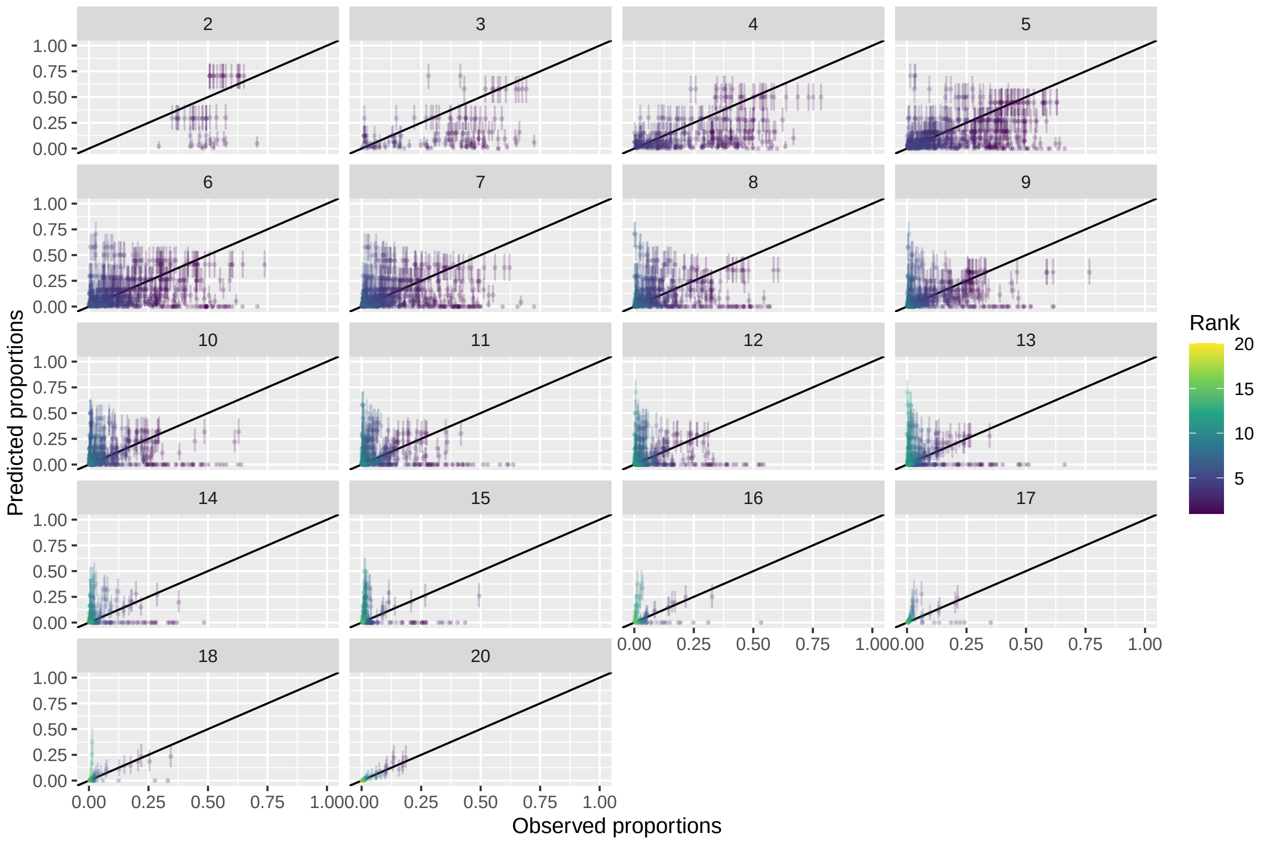

Figure A5.2: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.3 Logical model, Ordered-Dirichlet

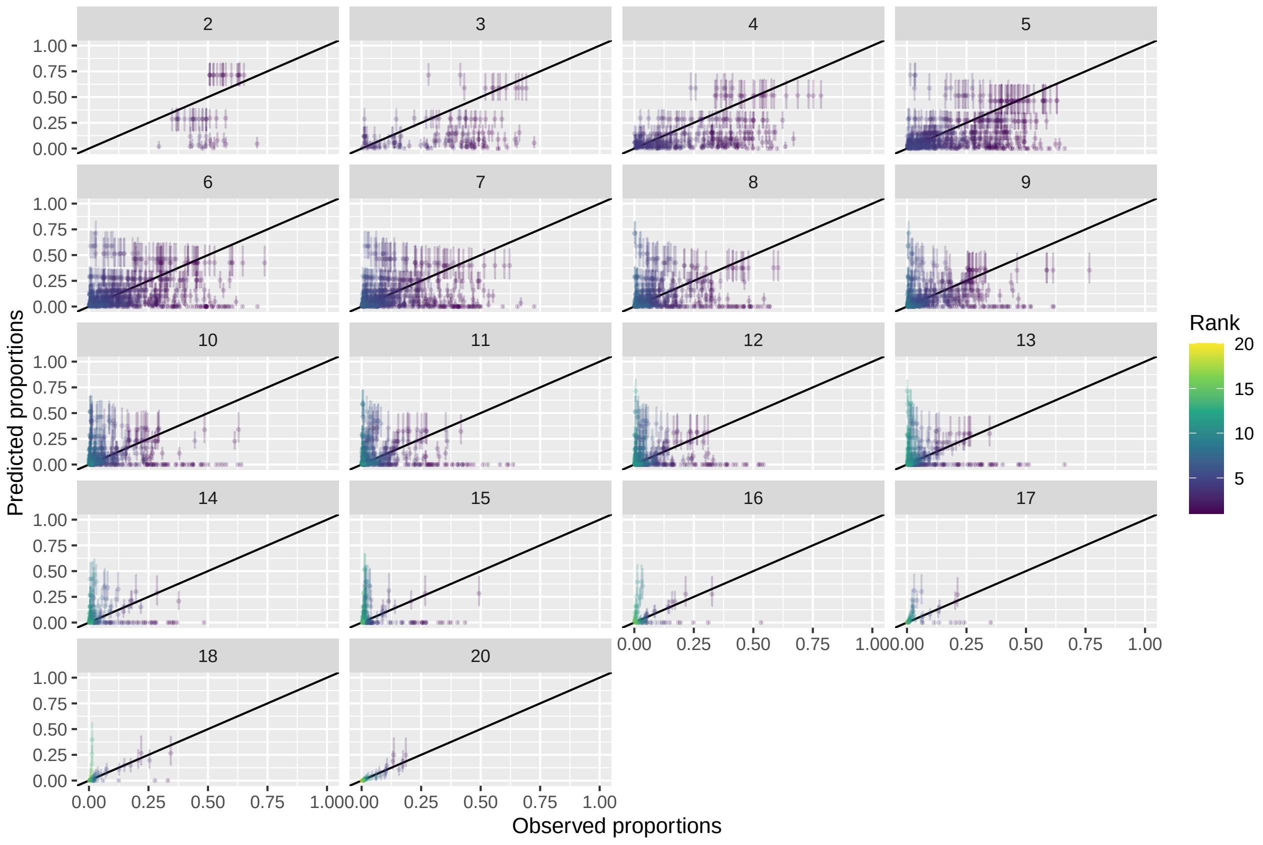

Figure A5.3: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.4 Political model, Dirichlet

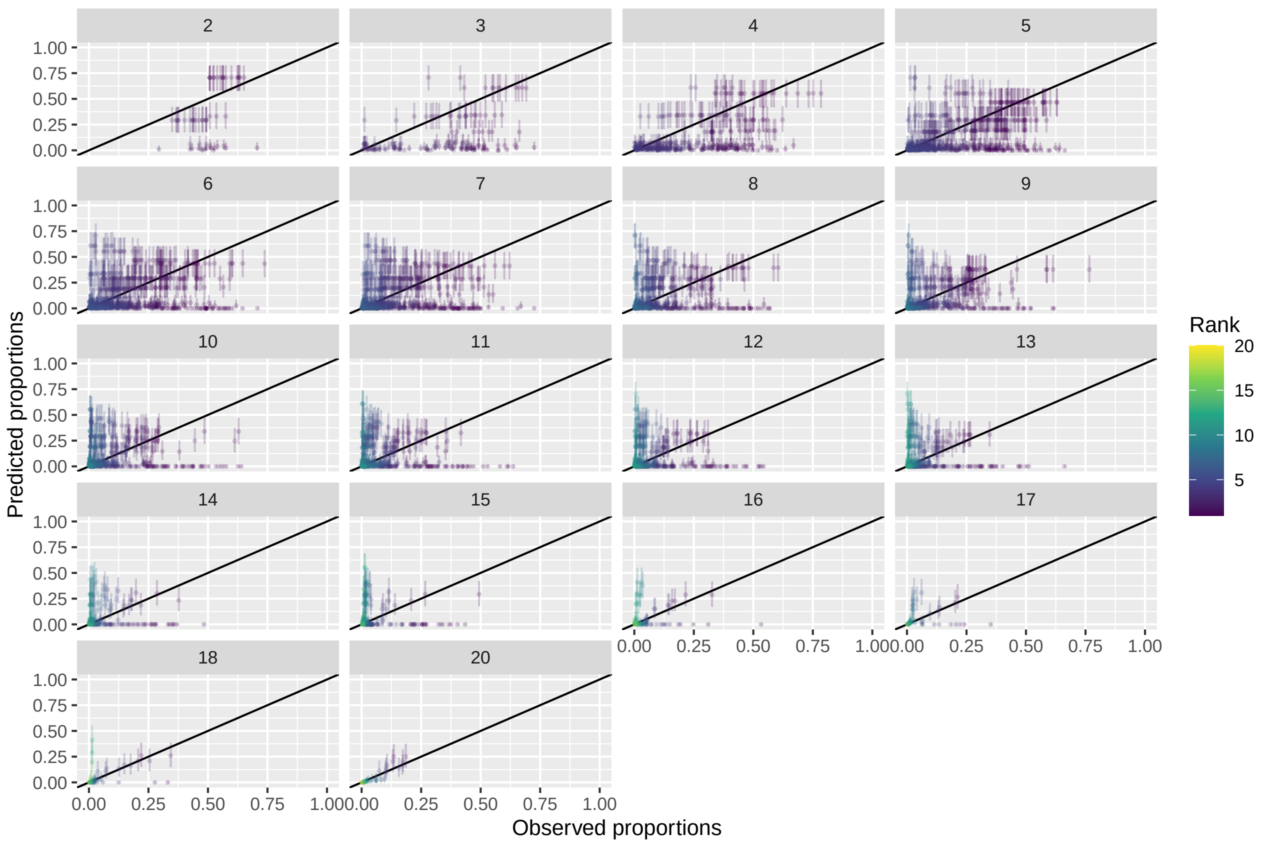

Figure A5.4: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.5 Political model, Ordered-Dirichlet

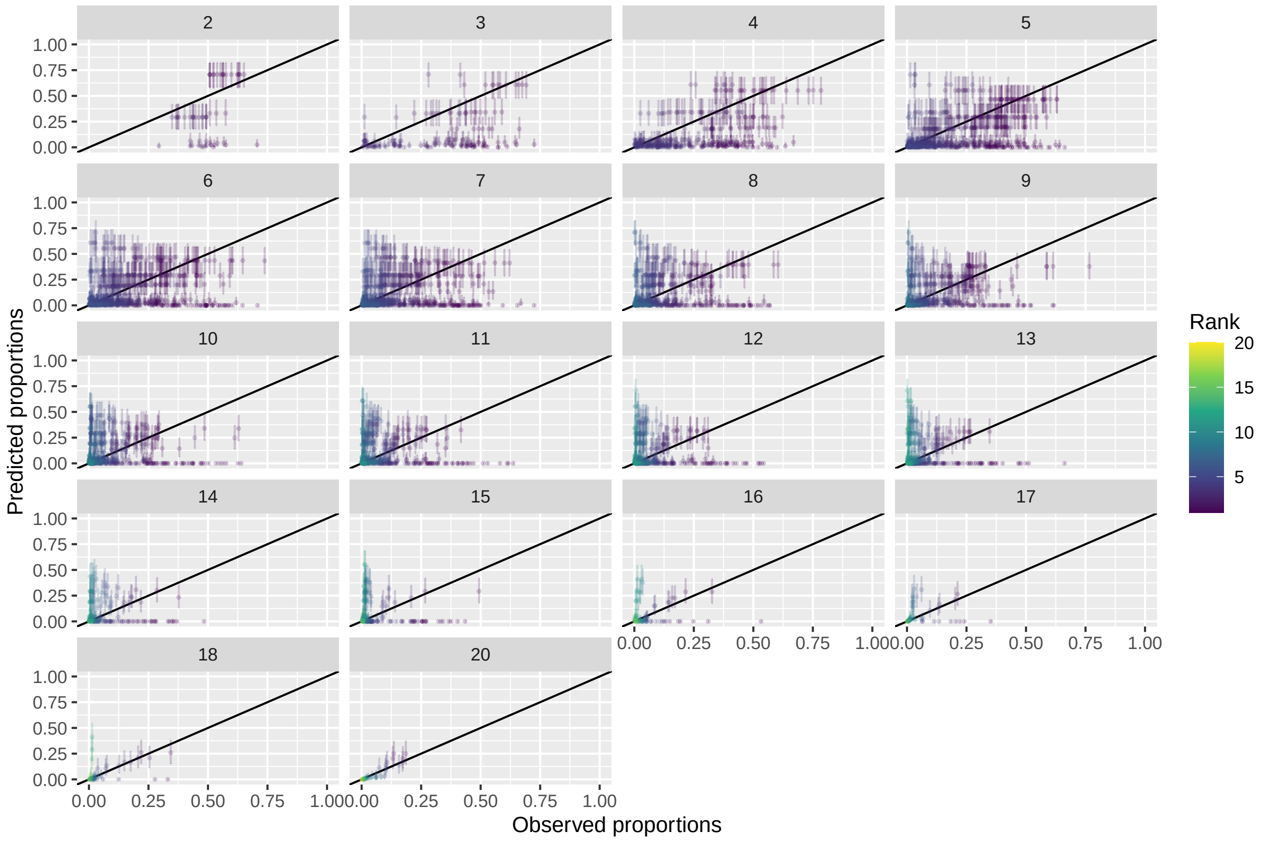

Figure A5.5: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.6 Saturated model, Dirichlet

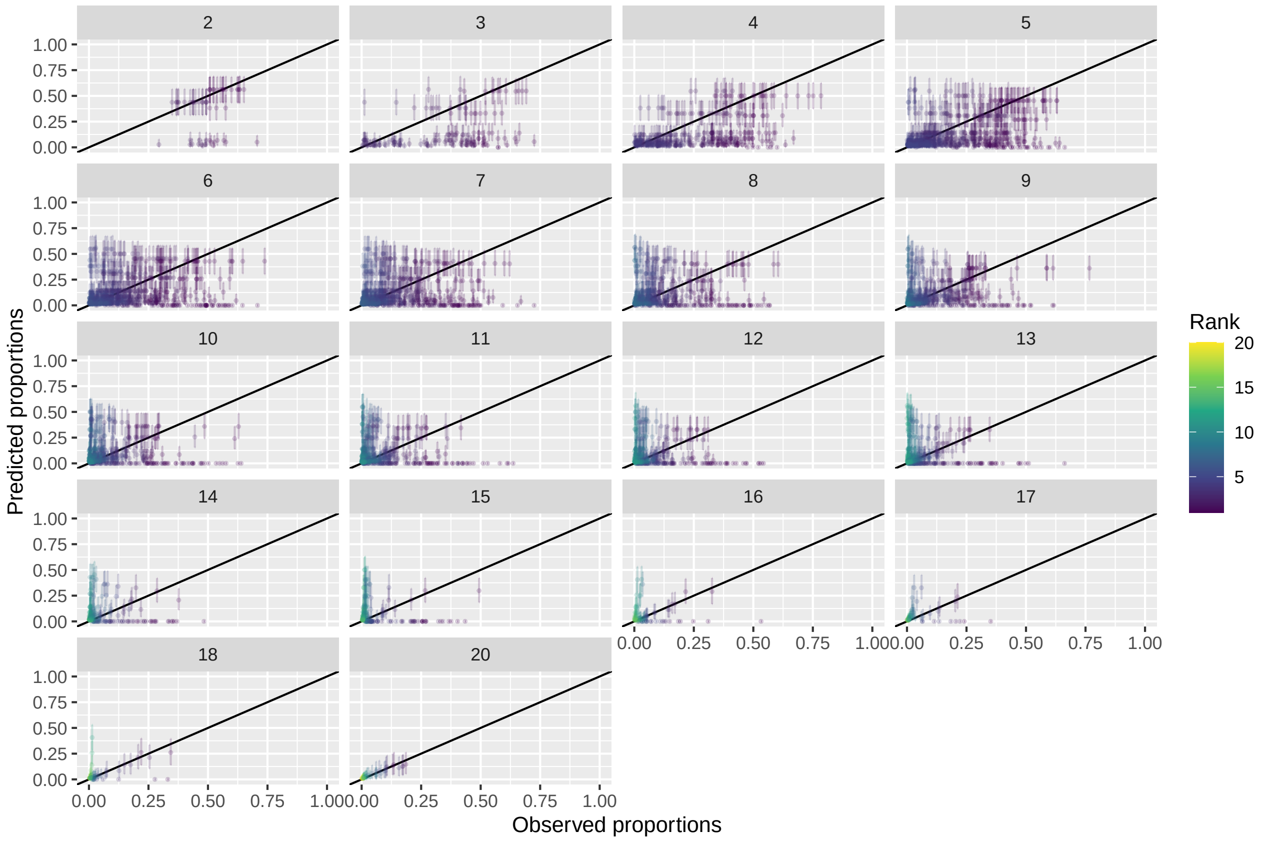

Figure A5.6: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.1.7 Saturated model, Ordered-Dirichlet

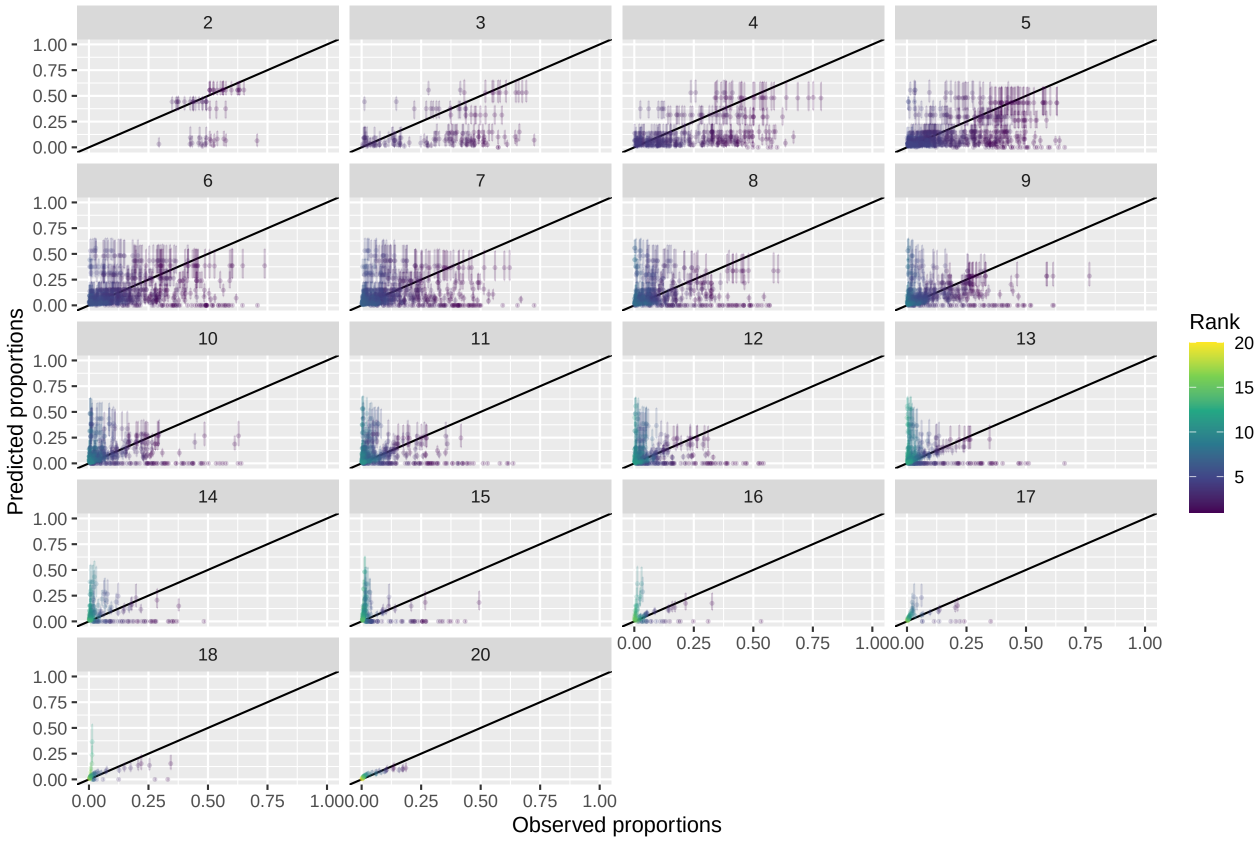

Figure A5.7: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2 Vote shares

A5.2.1 Null model, Dirichlet

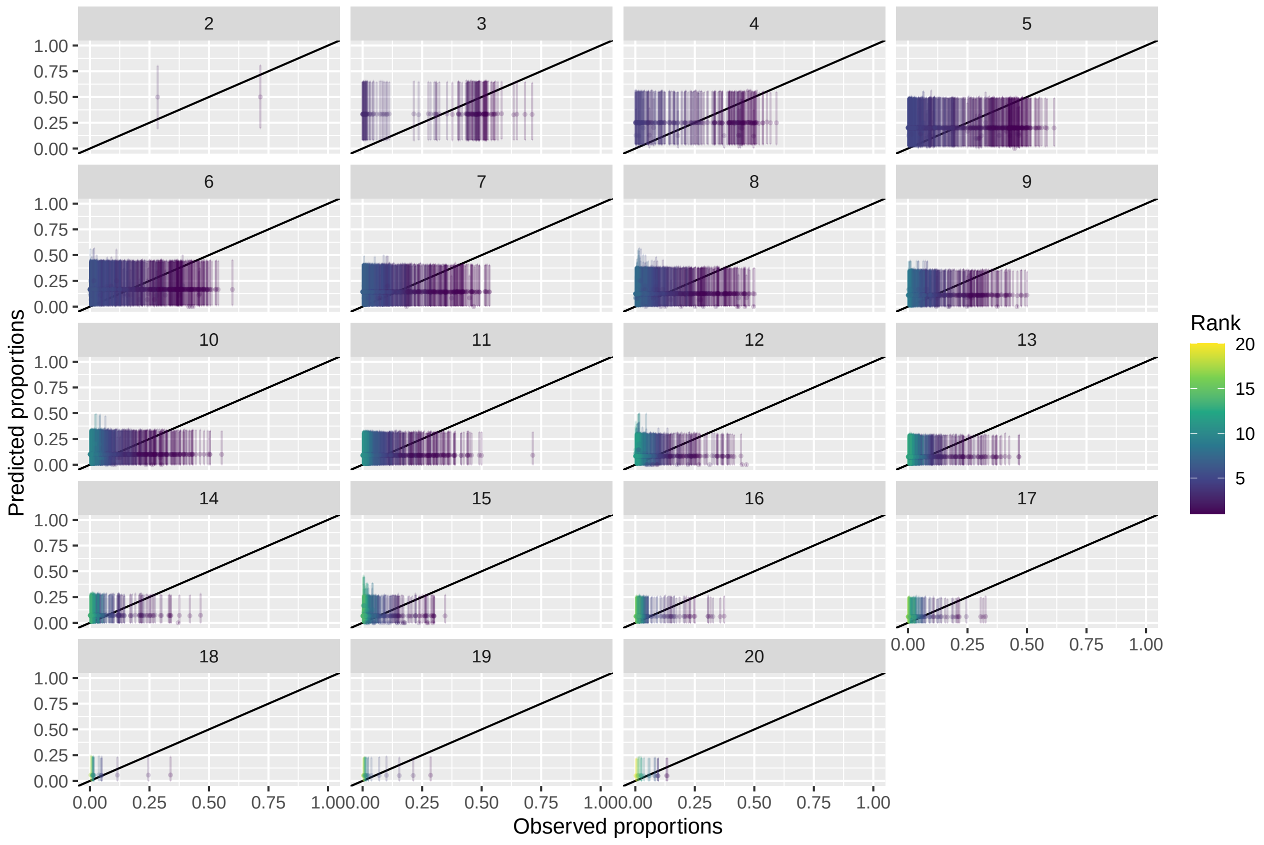

Figure A5.8: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.2 Logical model, Dirichlet

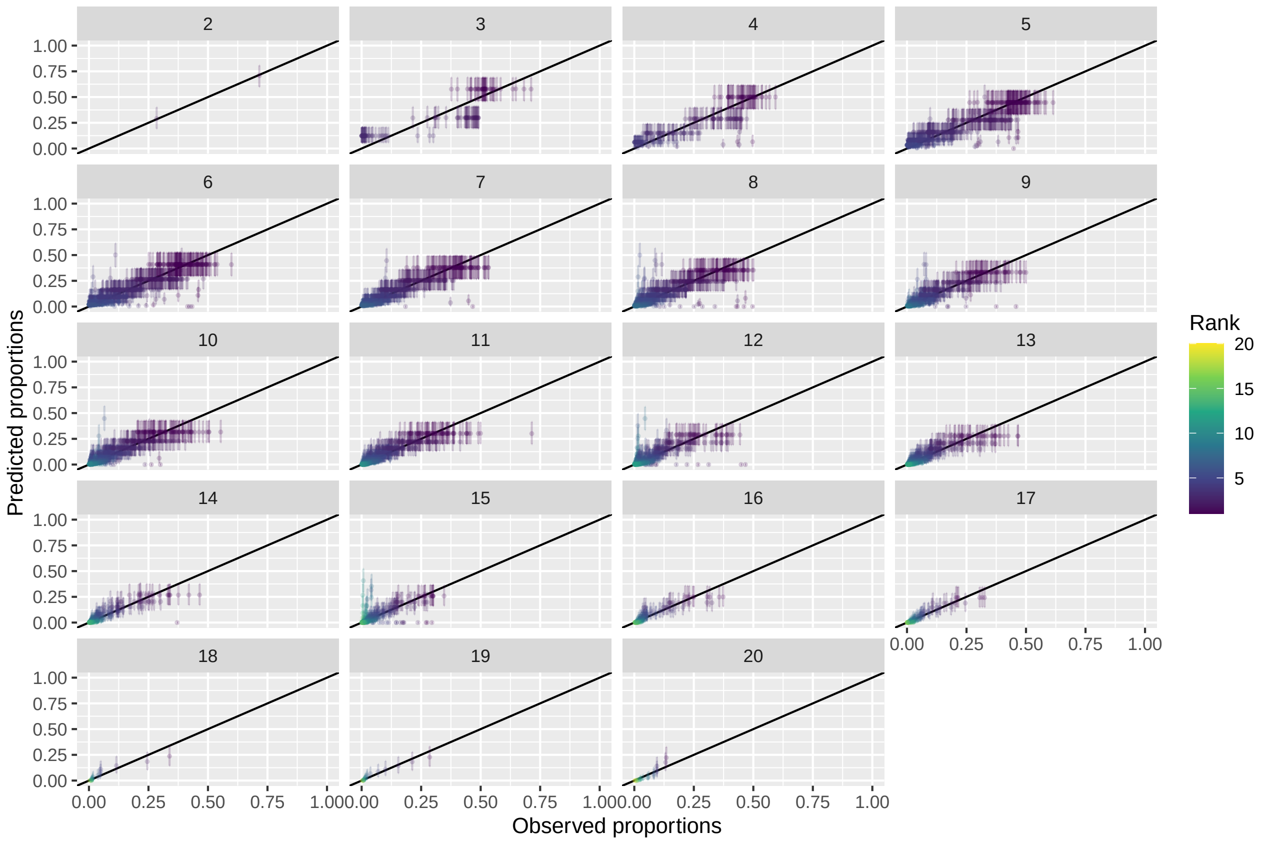

Figure A5.9: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.3 Logical model, Ordered-Dirichlet

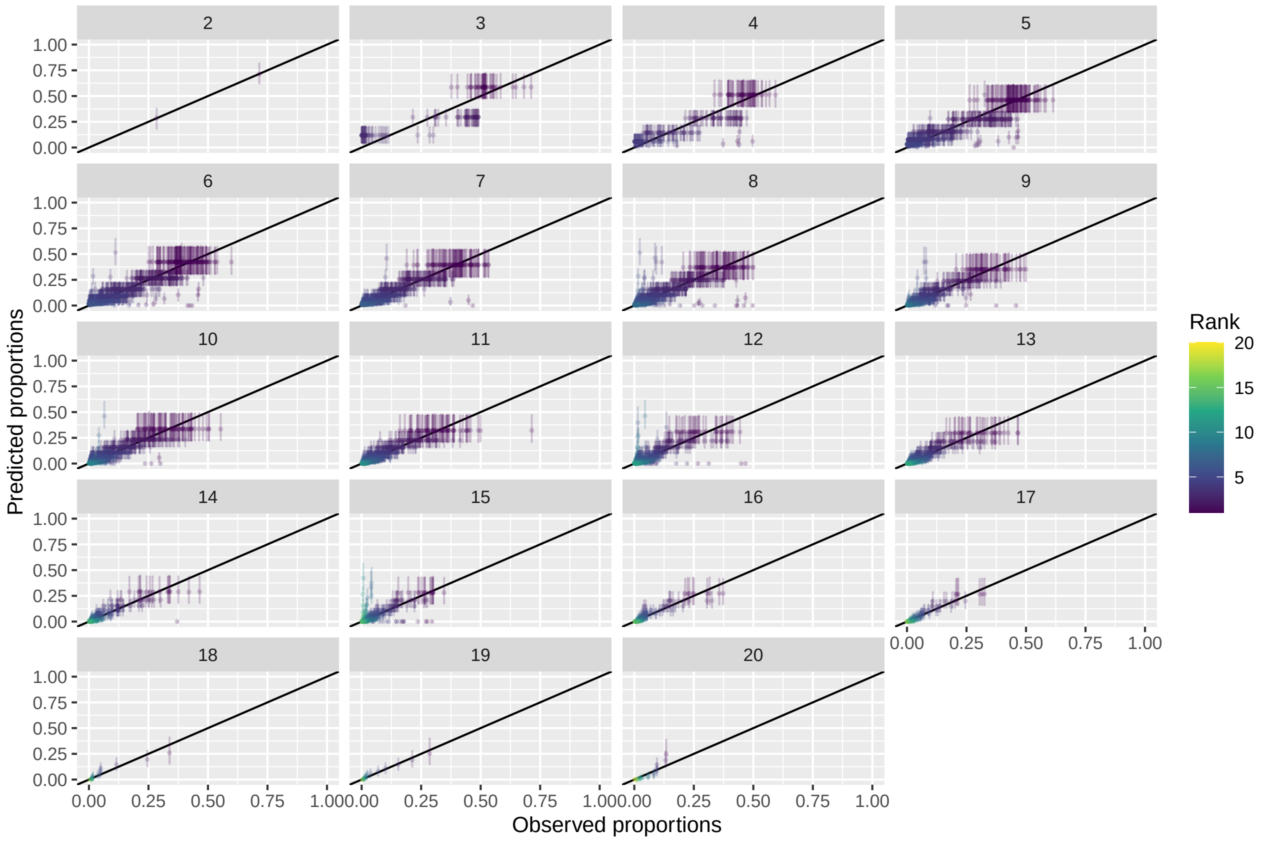

Figure A5.10: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.4 Political model, Dirichlet

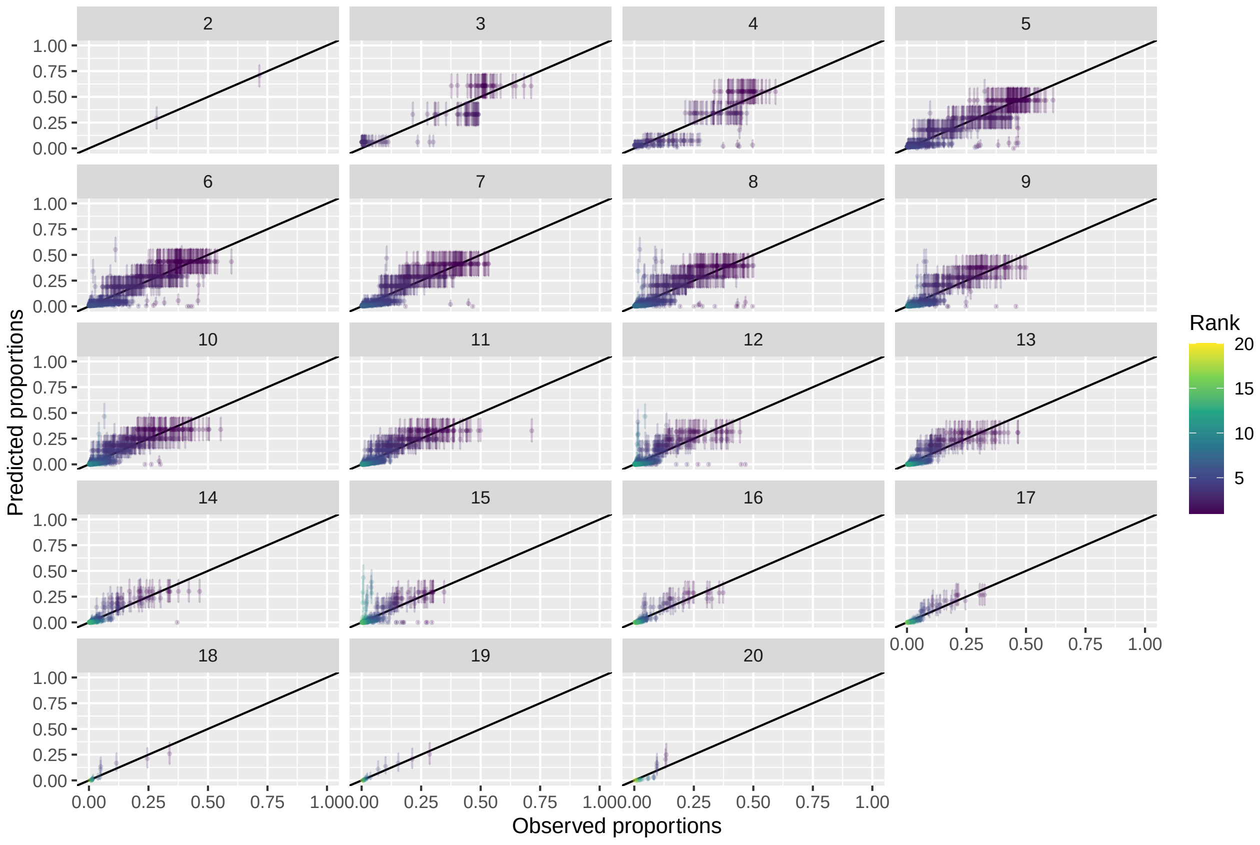

Figure A5.11: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.5 Political model, Ordered-Dirichlet

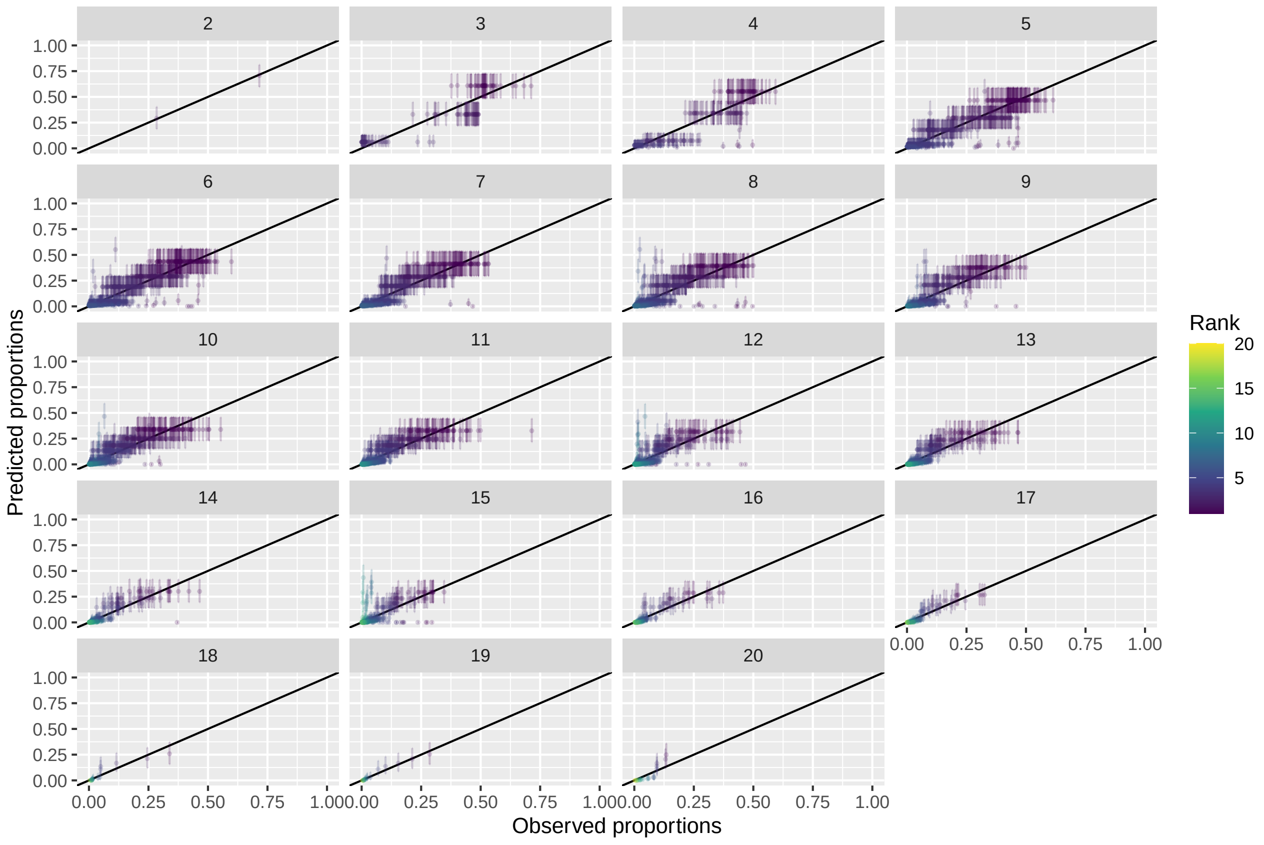

Figure A5.12: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.6 Saturated model, Dirichlet

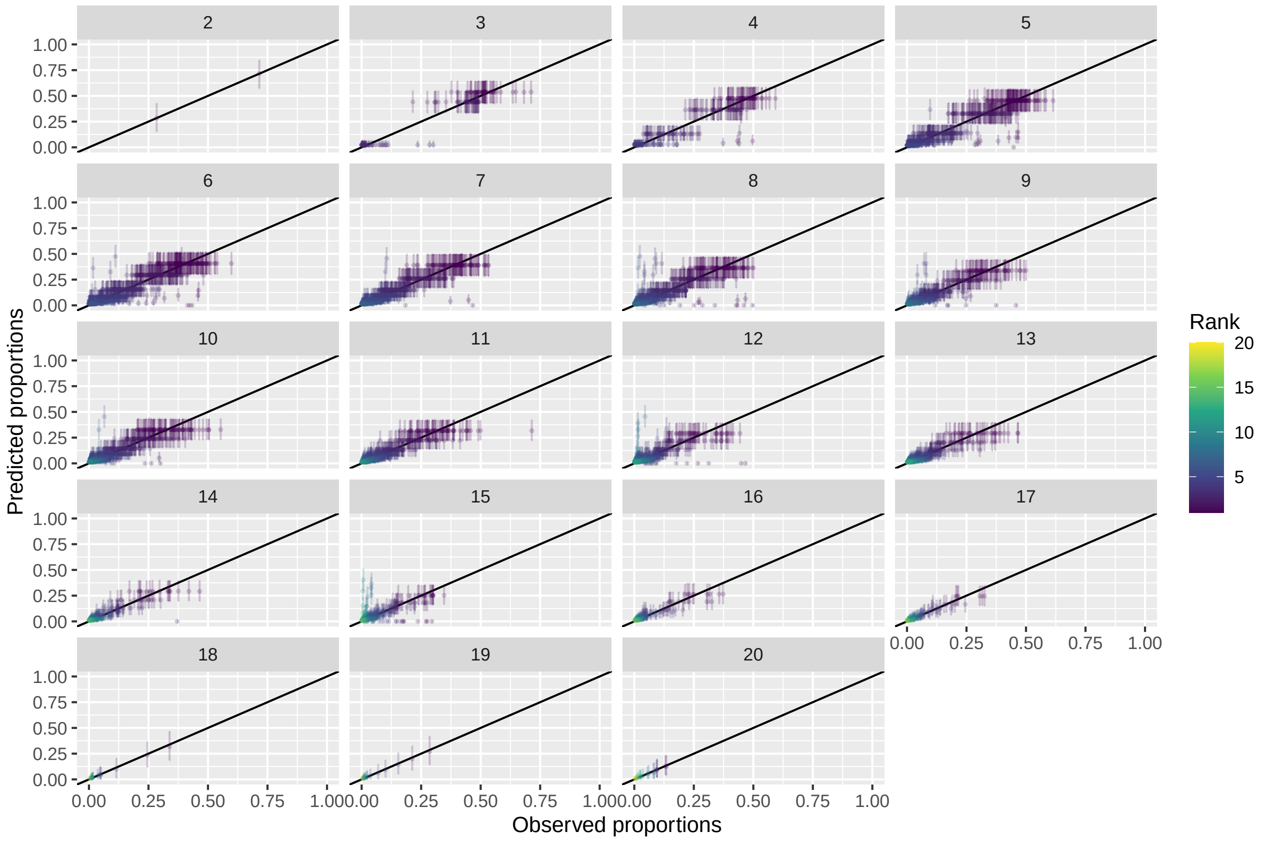

Figure A5.13: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A5.2.7 Saturated model, Ordered-Dirichlet

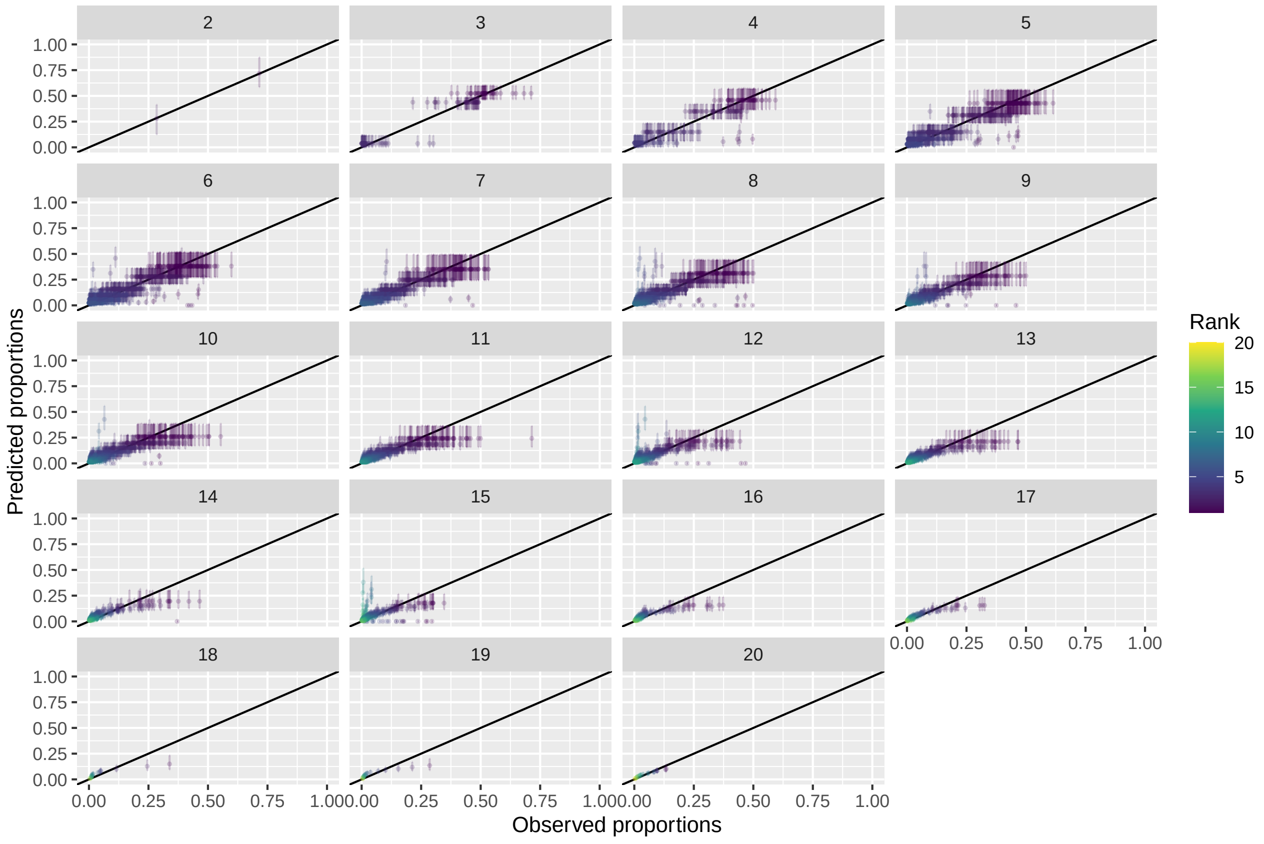

Figure A5.14: Observed vs. predicted proportions (with 90% posterior prediction intervals). The facets disaggregate the plot by different party system sizes N_{S0}. Party ranks are represented by colours, with lighter colours representing lower ranks.

A6 Heterogeneity in concentration parameters: Parliamentary vs. presidential systems

Our main text analyses use ParlGov data, which we have chosen due to its unparalleled coverage of both vote *and* seat shares. ParlGov does, however, have certain limitations. Most notably, it lacks information on seat- and vote-shares in presidential regimes. As our estimates and recommendations for simulating party shares are thus exclusively informed by parliamentary systems, one may question if our results equally extend to presidential systems. One reason to expect that this may not be the case is that in presidential democracies, the presidential party system spills over to the legislative party system, with the result that legislative systems will tend to have fewer parties than an otherwise comparable parliamentary party system (e.g., Hicken and Stoll 2011).

First, we note that this argument has less force when conditioning on the number or seat- or vote-winning parties (as our approach does). The argument of Hicken and Stoll (2011) is that where there are few (relevant) presidential candidates, this exerts a downward pressure on the (effective) number of legislative parties. If we assume for the moment that more presidential systems are characterized by having a few relevant candidates than by having many relevant candidates, one way in which the effective number of parties could come down is if some parties drop out entirely or fail to win seats. This would reduce the raw number of seat- or vote-winning parties. If we are able to condition on this smaller number, our simulations would include the effect that Hicken and Stoll (2011) identify.

Another way the effective number of parties could come down, however, is if the same parties compete and win seats as before, but if their shares are more concentrated. If *all* of the reduction in the effective number came through this route, then our model would produce inaccurate simulations. We would then have to resort to more radical criticisms of Hicken and Stoll (2011; and Neto and Cox 1997), of the kind found in Shugart and Taagepera (2017) (ch. 11, and pp. 309-310).

We address this question empirically by modeling vote shares from 304 election in 20 presidential democracies in the Americas included in the Latin American Electoral Volatility Dataset provided by Mainwaring and Su (2021). After removing non-democratic elections and dropping vote shares below one percent to make the data more comparable to ParlGov, we were able to estimate the same models as before. The results are presented in the table below.

The table shows, as before, that the logical model is preferable to the political model on the basis that it provides a better fit to the data. This is important, because one way that presidential systems might have a lower effective number of parties is through a concentration of share in the "top' ' parties supplying presidential candidates. Although the political model rewards the top parties, it does not do better at explaining vote shares in presidential systems.

The table does show that an ordered Dirichlet distribution offers a better fit to the data than an unordered Dirichlet. However, the differences in fit are not significantly different, in the sense that there is considerable overlap in the 90% credible intervals surrounding RMSE.

Finally, the table does show that vote shares in presidential systems are less tightly clustered around the vector of mean values $\mathbf{p}$. Whilst the estimated value of α in ParlGov vote share data was close to 50, the estimated value of α in this data is closer to 40, or around as predictable as seat shares were. While this 10-point discrepancy in the estimated α might seem substantial, it only results in very modest differences in the dispersion of the simulated shares: For instance, the 90% posterior predictive interval on the vote share of the largest party in a simulated two-party system widens from [0.60, 0.82] when $\alpha = 50$ to [0.58, 0.83] when $\alpha = 40$; in a simulated ten-party system, it widens from [0.21, 0.42] to [0.20, 0.43].

On the basis of this analysis, we think that our main conclusion – realistic seat- and vote- shares for a party system of a given size can be simulated using an unordered Dirichlet distribution with a mean vector given by the rule described in Taagepera and Allik (2006), and with an estimated precision parameter – still stands. There might, however, be grounds for allowing a little more variability when simulating the results of vote shares in presidential systems.

Table A6.3: Evaluation metrics for models of vote shares. RMSE measured in percentage points. Errors on N_S, s_1, s_2 are expressed in percentages of the true values [-100, +100]. Figures in square brackets are 90% credible intervals.

Model		α	RMSE	Calibration	Error N_v	Error v_1	Error v_2
Null	Dirichlet	7.16	17.6	89.3	16.6	0.0328	0.179
		[6.85, 7.48]	[17.2, 18.1]		[13.8, 19.4]	[-0.522, 0.611]	[-0.413, 0.763]
Logical	Dirichlet	37.2	8.16	82.1	6.07	-0.154	0.000803
		[35.5, 39.0]	[7.89, 8.44]		[4.61, 7.51]	[-0.387, 0.0843]	[-0.246, 0.252]
	Ordered	16.0	8.06	81.2	5.42	-0.213	0.00665
		[15.4, 16.6]	[7.76, 8.36]		[3.08, 7.85]	[-0.439, 0.0106]	[-0.235, 0.257]
Political	Dirichlet	36.3	8.26	73.2	-7.78	-0.229	-0.0239
		[34.7, 38.0]	[7.99, 8.52]		[-8.89, -6.68]	[-0.455, 0.00109]	[-0.261, 0.220]
	Ordered	15.1	8.15	69.3	-7.72	-0.269	-0.0202
		[14.5, 15.7]	[7.87, 8.43]		[-9.60, -5.76]	[-0.476, -0.0561]	[-0.241, 0.206]
Saturated	Dirichlet	60.1	6.42	89.8	4.72	-0.190	0.159
		[57.5, 63.4]	[6.21, 6.63]		[3.16, 6.29]	[-0.378, -0.00444]	[-0.0408, 0.362]
	Ordered	18.7	6.92	82.9	27.9	-0.0561	0.104
		[17.7, 19.7]	[6.68, 7.17]		[24.9, 31.0]	[-0.227, 0.116]	[-0.0855, 0.296]

More generally, we note that whenever practitioners expect substantial heterogeneity in α – across different executive systems, regions, time periods, etc. – they can flexibly use the statistical models we implemented to estimate α across any subsets of elections that is of interest to them and that they have data for. The corresponding Stan model code is included in Online Appendix A4 and made available along with our data-processing routine as part of the replication materials. Practitioners can use these tools to explore potential heterogeneity in alpha and use the resulting estimates within the simulation tools we provide to simulate realistic party systems that are specific to, e.g., particular regions, eras, or executive systems.

References

Dorp, J.R. van, and T.A. Mazzuchi. 2004. “Parameter Specification of the Beta Distribution and Its Dirichlet Extensions Utilizing Quantiles.” *Statistics Textbooks and Monographs* 174: 283–318.

Hicken, A., and H. Stoll. 2011. “Presidents and Parties: How Presidential Elections Shape Coordination in Legislative Elections.” *Comparative Political Studies* 44 (7): 854–83.

Mainwaring, S., and Y.-P. Su. 2021. “Replication Data for: Electoral Volatility in Latin America, 1932–2018.” Harvard Dataverse.

Neto, O.A., and G.W. Cox. 1997. “Electoral Institutions, Cleavage Structures, and the Number of Parties.” *American Journal of Political Science* 41 (1): 149–74.

Shugart, M.S., and R. Taagepera. 2017. *Votes from Seats: Logical Models of Electoral Systems.* Cambridge University Press.

Taagepera, R., and M. Allik. 2006. “Seat Share Distribution of Parties: Models and Empirical Patterns.” *Electoral Studies* 25 (4): 696–713.