Online Appendix

Contents

A1. Review of empirical political science literature

As we mention in the main text, we conducted a review of subnational empirical research published in top political science journals since 2010. We utilized a keyword search of the *Web of Science* search engine to find a total of 100 peer-reviewed articles that utilized some form of subnational/geospatial empirical research. We make no claims that this subset of articles contains the entire population of subnational/geospatial empirical articles within political science, but we are confident that this provides a representative sample of the extant literature within the field. Below we enumerate the articles in our sample, and note which ones include steps taken by researchers to change the support of key variables.

Author & Year	DOI	Journal	CoS by Authors
Gehring (2021)	10.1017/S0003055418000709	APSR	
Harris & Posner (2019)	10.1017/S0003055418000709	APSR	✓
Hankinson (2018)	10.1017/S0003055418000035	APSR	
Tajima et al. (2018)	10.1017/S0003055418000138	APSR	
Braun (2016)	10.1017/S0003055415000544	APSR	
Cederman et al. (2011)	10.1017/S0003055411000207	APSR	✓
Bohmelt et al. (2020)	10.1111/ajps.12494	AJPS	
Nall et al. (2018)	10.1111/ajps.12305	AJPS	
Knutsen et al. (2017)	10.1111/ajps.12268	AJPS	
Monogan et al. (2017)	10.1111/ajps.12278	AJPS	
Stokes (2016)	10.1111/ajps.12220	AJPS	✓
Nyhan & Montgomery (2015)	10.1111/ajps.12143	AJPS	

Williams & Whitten (2015)	10.1111/ajps.12124	AJPS	
Branton et al. (2015)	10.1111/ajps.12159	AJPS	✓
Wallace et al. (2014)	10.1111/ajps.12060	AJPS	
Bhavani et al. (2014)	10.1111/ajps.12045	AJPS	✓
Mukherjee & Singer (2010)	10.1111/j.1540-5907.2009.00417.x	AJPS	
Cho & Gimpel (2010)	10.1111/j.1540-5907.2009.00419.x	AJPS	
Wimpy et al. (2021)	10.1086/710089	JOP	
Montgomery & Nyhan (2017)	10.1086/690301	JOP	
Bove & Bohmelt (2016)	10.1086/684679	JOP	
Nall (2015)	10.1086/679597	JOP	✓
Benmelech et al. (2015)	10.1086/678765	JOP	✓
Boehmke et al. (2012)	10.1017/S0022381612000321	JOP	✓
Bell et al. (2012)	10.1017/S0022381611001642	JOP	
Weidmann (2011)	10.1017/S0022381611000831	JOP	✓
Clemens et al. (2015)	10.1111/lsq.12067	LSQ	
Cortina (2020)	10.1177/1065912919854135	PRQ	
Briggs (2019)	10.1177/1065912918798489	PRQ	
Croicu & Kreutz (2017)	10.1177/1065912916670272	PRQ	
Minkoff & Lyonos (2019)	10.1177/1532673X17733799	APR	✓
Smith & Weinberg (2016)	10.1177/1532673X15602755	APR	
Gill (2021)	10.1177/1532440020930197	SPPQ	
Darmofal et al. (2019)	10.1177/1532440019851806	SPPQ	
Pacheco (2017)	10.1177/1532440017705150	SPPQ	
Parinandi (2013)	10.1177/1532440013484477	SPPQ	
Boehmke & Skinner (2012)	10.1177/1532440012438890	SPPQ	
Carson et al. (2012)	10.1177/1532440012438892	SPPQ	
Gilardi & Wasserfallen (2014)	10.1017/S0007123414000246	BJPS	
Bell et al. (2013)	10.1017/S0007123413000100	BJPS	
Gibler & Braithwaite (2013)	10.1017/S000712341200052X	BJPS	
Gatesman & Unwin (2021)	10.1017/pan.2020.22	PA	
Juhl (2021)	10.1017/pan.2020.23	PA	
Betz et al. (2021)	10.1017/pan.2020.26	PA	
Saxon (2020)	10.1017/pan.2019.45	PA	✓
Vande Kamp (2020)	10.1017/pan.2019.35	PA	
Juhl (2020)	10.1017/pan.2019.12	PA	
Betz et al. (2018)	10.1017/pan.2018.10	PA	

Harbers & Ingram (2017)	10.1017/pan.2017.4	PA	
Goplerud (2016)	10.1093/pan/mpv029	PA	✓
Franzese et al. (2012)	10.1093/pan/mpr049	PA	
Steinwand (2011)	10.1093/pan/mpr026	PA	
Abramson & Carter (2021)	10.1017/S0020818320000545	IO	
Christensen (2019)	10.1017/S0020818318000413	IO	
Sommerer & Tallberg (2019)	10.1017/S0020818318000450	IO	
Cunningham & Sawyer (2017)	10.1017/S0020818317000200	IO	
Branch (2016)	10.1017/S0020818316000199	IO	
Steinwand (2015)	10.1017/S0020818314000381	IO	
Chaudoin et al. (2015)	10.1017/S0020818314000356	IO	
Neumayer et al. (2014)	10.1017/S0020818313000362	IO	
Neumayer & Pluemper (2010)	10.1017/S0020818309990191	IO	
Brazys & Kotsdam (2020)	10.1093/isq/sqaa072	ISQ	
Jones & Zeitz (2019)	10.1093/isq/sqz068	ISQ	
Reeder (2018)	10.1093/isq/sqy016	ISQ	✓
Bohmelt et al. (2017)	10.1093/isq/sqx067	ISQ	
Zhukov & Stewart (2013)	10.1111/isqu.12008	ISQ	
Barthel & Neumayer (2012)	10.1111/j.1468-2478.2012.00757.x	ISQ	
Cao (2010)	10.1111/j.1468-2478.2010.00611.x	ISQ	
Kosec & Mogues (2020)	10.1017/S0043887120000027	WP	
Wilfahrt (2018)	10.1017/S0043887117000363	WP	✓
Baccini et al. (2014)	10.1017/S0043887114000124	WP	
Obinger & Schmitt (2011)	10.1017/S0043887111000025	WP	
Cammett & Issar (2010)	10.1017/S0043887110000080	WP	✓
Schvitz et al. (2021)	10.1177/00220027211013563	JCR	
Echevarria-Coco et al. (2021)	10.1177/0022002720958470	JCR	
Polo (2020)	10.1177/0022002720930811	JCR	
Ito & Elliot (2020)	10.1177/0022002719885428	JCR	
Koren (2019)	10.1177/0022002719833160	JCR	✓
Moro & Sberna (2018)	10.1177/0022002717693049	JCR	
Bohnet et al. (2018)	10.1177/0022002716665209	JCR	
Miller et al. (2018)	10.1177/0022002716649232	JCR	
Minhas & Radford (2017)	10.1177/0022002716639100	JCR	
Schultz (2017)	10.1177/0022002715620470	JCR	
Osorio (2015)	10.1177/0022002715587048	JCR	

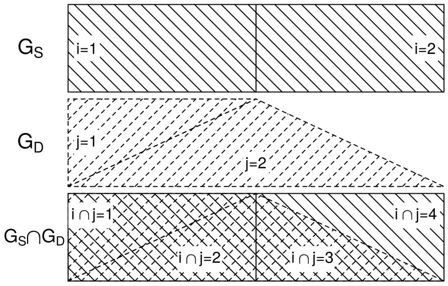
Schutte (2015)	10.1177/0022002713520534	JCR	
Baccini & Duer (2015)	10.1177/0022002713516844	JCR	
Fjelde & Hultman (2014)	10.1177/0022002713492648	JCR	✓
McDoom (2014)	10.1177/0022002713484282	JCR	
Althaus et al. (2012)	10.1177/0022002711422340	JCR	
Buhaug et al. (2011)	10.1177/0022002711408011	JCR	✓
Weidmann & Ward (2010)	10.1177/0022002710371669	JCR	
Chang & Manion (2021)	10.1177/0010414021989762	CPS	
Spater & Tranvik (2019)	10.1177/0010414019830721	CPS	
Lechler & McNamee (2018)	10.1177/0010414018758760	CPS	
Ejdemyr et al. (2018)	10.1177/0010414017730079	CPS	
De Juan (2017)	10.1177/0010414016688006	CPS	✓
Maehler & Pierskalla (2015)	10.1177/0010414014545012	CPS	✓
Ward & Cao (2012)	10.1177/0010414011434007	CPS	
Neumayer & Pluemper (2012)	10.1177/0010414011429066	CPS	
Elkink (2011)	10.1177/0010414011407474	CPS	

A2. Scale and nesting metrics

This section provide details on the nesting and scale metrics used in the main text, as well as several alternative metrics included in the SUNGEO R package (`nesting()` function).

Let $\mathcal{G}_S$ be a set of source polygons, indexed $i = 1, \ldots, N_S$, and $\mathcal{G}_D$ be a set of destination polygons, indexed $j = 1, \ldots, N_D$. Let $\mathcal{G}_{S \cap D}$ be the intersection of polygons 1 and 2, indexed $i \cap j = 1, \ldots, N_{S \cap D} : N_{S \cap D} \geqslant \max(N_S, N_D)$. Let a_i be the area of source polygon i, and a_j be the area of destination polygon j. Let $a_{i \cap j}$ be the area of $i \cap j : a_{i \cap j} \leqslant \min(a_i, a_j)$. Let $1(\cdot)$ be a Boolean operator, equal to 1 if statement "$\cdot$" is true, and 0 otherwise.

Each intersection $i \cap j$ can be mapped to its parent polygons i and j, using a $N_{S \cap D} \times 3$ matrix of indices $M_{S \cap D}$. For illustrative purposes, consider the stylized example below:



$$M_{S \cap D} = \begin{array}{|c|c|c|} \hline i \cap j & i & j \\ \hline 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 2 \\ 4 & 2 & \varnothing \\ \hline \end{array} \quad \text{where } N_{S \cap D} = 4$$

$M_{i\cap D}$ is the subset of $M_{S\cap D}$ indexing the $N_{i\cap D}$ intersections of polygon i. For $i = 1$:

$$M_{1\cap D} = \begin{array}{|c|c|c|}\hline i\cap j & i & j \\ \hline 1 & 1 & 1 \\ 2 & 1 & 2 \\ \hline \end{array} \quad \text{where } N_{1\cap D} = 2$$

Similarly, $M_{S\cap j}$ is the subset corresponding to destination polygon j. For $j = 2$:

$$M_{S\cap 2} = \begin{array}{|c|c|c|}\hline i\cap j & i & j \\ \hline 2 & 1 & 2 \\ 3 & 2 & 2 \\ \hline \end{array} \quad \text{where } N_{S\cap 2} = 2$$

If a polygon (or a part of a polygon) from $\mathcal{G}_S$ does not overlap with any features from $\mathcal{G}_D$, the corresponding row in $M_{S\cap D}$ will have an empty value for j (and vice versa):

$$M_{2\cap D} = \begin{array}{|c|c|c|}\hline i\cap j & i & j \\ \hline 3 & 2 & 2 \\ 4 & 2 & \varnothing \\ \hline \end{array}$$

where $i \cap j = 4$ corresponds to a part of source polygon $i = 2$ that does not overlap with any polygons from $\mathcal{G}_D$. Note that this intersection is not empty; it just has a "single parent" and cannot be mapped to any destination features j.

We can now define our nesting and scale metrics.

- *Relative nesting* (RN). Captures how closely source and destination boundaries align:

$$RN = \frac{1}{N_S} \sum_{i}^{N_S} \sum_{i\cap j}^{N_{i\cap D}} \left(\frac{a_{i\cap j}}{a_i} \right)^2 \tag{A2.1}$$

  which is the share of source units that cannot be split across multiple destination units. Values of 0 indicate no nesting (every source unit can be split across multiple destination units) and values of 1 indicate full nesting (no source unit can be split across multiple destination units).

- *Relative scale* (RS). Captures whether a task is one of aggregation or disaggregation:

$$RS = \frac{1}{N_{S\cap D}} \sum_{i\cap j}^{N_{S\cap D}} 1(a_i < a_j) \tag{A2.2}$$

which is the share of source units that are smaller than destination units. Its range is from 0 to 1, where values of 1 indicate pure aggregation (all source units are smaller than destination units) and values of 0 indicate no aggregation (all source units are at least as large as destination units). Values between 0 and 1 indicate a hybrid (i.e. some source units are smaller, others are larger than target units).

- *Relative nesting, symmetric* ($RN\text{-}sym$). Alternative measure of RN, ranges from -1 to 1:

$$RN\text{-}sym = \frac{1}{N_S}\sum_{i}^{N_S}\sum_{i\cap j}^{N_{i\cap D}}\left(\frac{a_{i\cap j}}{a_i}\right)^2 - \frac{1}{N_D}\sum_{j}^{N_D}\sum_{i\cap j}^{N_{S\cap j}}\left(\frac{a_{i\cap j}}{a_j}\right)^2 \tag{A2.3}$$

  which is the difference between the nesting of source units within destination units, $1/N_S \sum_i^{N_S} \sum_{i\cap j}^{N_{i\cap D}} (a_{i\cap j}/a_i)^2$ (i.e. RN from standpoint of $\mathcal{G}_S$), and the nesting of destination units within source units, $1/N_D \sum_j^{N_D} \sum_{i\cap j}^{N_{S\cap j}} (a_{i\cap j}/a_j)^2$ (RN from standpoint of $\mathcal{G}_D$). Values of 1 indicate that source units are perfectly nested within destination units; -1 indicates that destination units are perfectly nested within source units.

- *Relative scale, symmetric* ($RS\text{-}sym$). Alternative measure of RS, ranges from -1 to 1:

$$RS\text{-}sym = \frac{1}{N_{S\cap D}}\sum_{i\cap j}^{N_{S\cap D}} 1(a_i < a_j) - 1(a_i > a_j) \tag{A2.4}$$

  which is a difference between two proportions: $1/N_{S\cap D}\sum 1\,(a_i < a_j)$, or the share of source units that is smaller than destination units (i.e. RS from standpoint of $\mathcal{G}_S$), and $1/N_{S\cap D}\sum 1\,(a_i > a_j)$, the share that is larger (i.e. RS from standpoint of $\mathcal{G}_D$). Its range is from -1 (pure disaggregation, all source units are larger than target units) to 1 (pure aggregation, all source units are smaller than target units). Values of 0 indicate that all source units are the same size as target units.

- *Relative nesting, conditional* ($RN\text{-}nn$). RN for source units that are not fully nested:

$$RN^{(\text{nn})} = \frac{1}{N_{S^\star}}\sum_{i}^{N_{S^\star}}\sum_{i\cap j}^{N_{i\cap D}}\left(\frac{a_{i\cap j}}{a_i}\right)^2 \tag{A2.5}$$

  where $S^\star$ denotes the set of source units with $\frac{1}{N_{i\cap D}}\sum_{i\cap j}^{N_{i\cap D}}\frac{a_{i\cap j}}{a_i} < 1$.

- *Relative scale, conditional* (RS-nn). RS for source units that are not fully nested:

$$RS^{(\text{nn})} = \frac{1}{N_{S^\star \cap D}} \sum_{i \cap j}^{N_{S^\star \cap D}} 1(a_i < a_j) \tag{A2.6}$$

where $S^\star$ denotes the set of source units with $\frac{1}{N_{i \cap D}} \sum_{i \cap j}^{N_{i \cap D}} \frac{a_{i \cap j}}{a_i} < 1$.

- *Proportion intact* (PI). A nesting metric that requires no area calculations at all:

$$PI = \frac{1}{N_S} \sum_{i}^{N_S} 1 \left(N_{i \cap D} - \sum_{i \cap j}^{N_{i \cap D}} 1 \left(\mathcal{G}_{i \cap j} = \varnothing \right) = 1 \right) \tag{A2.7}$$

This measure ranges from 0 to 1, where 1 indicates full nesting (i.e. every source unit is intact/no splits), and 0 indicates no nesting (i.e. no source unit is intact/all are split).

- *Proportion fully nested* (PFN). A stricter version of PI, which also requires that source units are fully contained within destination units (in PI, source units outside the boundaries of the destination layer are considered "intact"; in PFN, they are not).

$$PFN = \frac{1}{N_S} \sum_{i}^{N_S} 1 \left(\frac{1}{N_{i \cap D}} \sum_{i \cap j}^{N_{i \cap D}} \frac{a_{i \cap j}}{a_i} = 1 \right) \tag{A2.8}$$

This measure ranges from 0 to 1, where 1 indicates full nesting (i.e. every source unit is intact AND is fully contained within a single destination unit), and 0 indicates no nesting (i.e. no source unit is intact OR none are contained within destination units).

- *Relative overlap* (RO). Assesses extent of spatial overlap between source and destination polygons. Let α_S be the combined area of all source polygons. Let α_D be the combined area of all destination polygons. Let $\alpha_{S(-D)}$ be the combined area of all source polygons, excluding the area covered by destination polygons. Let $\alpha_{D(-S)}$ be the combined area of all destination polygons, excluding the area covered by source polygons.

$$RO = \frac{\alpha_{S(-D)}}{\alpha_S} - \frac{\alpha_{D(-S)}}{\alpha_D} \tag{A2.9}$$

this measure is scaled between -1 and 1. Values of 0 indicate perfect overlap (there is no part of source units that fall outside of destination units, and vice versa). Values between 0 and 1 indicate a source "underlap" (some parts of source polygons fall outside of destination polygons; more precisely, a larger part of source polygon area falls outside destination polygons than the other way around). Values between -1 and 0 indicate a destination "underlap" (some parts of destination polygons fall outside of source polygons; a larger part of destination polygon area falls outside source polygons than the other way around). Values of -1 and 1 indicate no overlap (all source units fall outside destination units, and vice versa). This is a theoretical limit only; in the R package, the function returns an error if there is no overlap.

Table A2.2 reports the pairwise correlations between these metrics, for the Monte Carlo simulations described in the main text. As the table suggests, the correlations are generally strongly positive, but not always perfect — especially in the case of PI and RO, which are capturing conceptually different properties of changes of support.

Metric	RN	RS	RN-sym	RS-sym	RN-nn	RS-nn	PI	PFN	RO
RN	1.00	0.97	1.00	0.97	1.00	0.97	0.80	0.77	0.20
RS	0.97	1.00	0.97	1.00	0.97	0.99	0.73	0.69	0.18
RN-sym	1.00	0.97	1.00	0.97	0.99	0.97	0.80	0.77	0.20
RS-sym	0.97	1.00	0.97	1.00	0.97	0.99	0.73	0.69	0.18
RN-nn	1.00	0.97	0.99	0.97	1.00	0.98	0.74	0.71	0.20
RS-nn	0.97	0.99	0.97	0.99	0.98	1.00	0.68	0.64	0.18
PI	0.80	0.73	0.80	0.73	0.74	0.68	1.00	0.99	0.17
PFN	0.77	0.69	0.77	0.69	0.71	0.64	0.99	1.00	0.15
RO	0.20	0.18	0.20	0.18	0.20	0.18	0.17	0.15	1.00

Table A2.2: Correlation between alternative nesting metrics.

A3. The relationship between RN and RS

The strong correlations reported in Table A2.2 raise several important questions about how these metrics relate to each other. First, are some of these metrics more strongly predictive of transformation quality than others? Second, are these metrics redundant? After we condition on RN, for example, does RS add any explanatory value in accounting for variation in transformation quality? Third, how frequently do these metrics diverge, and what are the implications of such divergence for analysis?

Our Monte Carlo simulations confirm that some nesting metrics — particularly RN and its variants — have particularly strong explanatory power as predictors of transformation quality. Table A3.3 reports a "horse race" evaluation of the nesting metrics' ability to explain transformation quality, as measured by RMSE, Spearman's correlation and OLS estimation bias. Specifically, we replicated the semi-parametric regressions in equation (5), each time with a different nesting metric on the right-hand side, and compared goodness-of-fit diagnostics across these specifications. Across all three fit diagnostics —

Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and the sum of squared residuals (Deviance) — RN consistently outperforms RS. Moreover, some variants of RN (e.g. RN-sym, RN-nn) offer marginal performance gains over the original.

Diagnostic	Metric	AIC	BIC	Deviance
log(NRMSE)	RN-sym	418288.80	418461.79	67176.92
	RN	418795.50	418968.49	67269.85
	RN-nn	419522.91	419695.90	67403.49
	RS	425622.19	425795.18	68534.52
	RS-sym	425622.19	425795.18	68534.52
	RS-nn	425641.08	425814.07	68538.05
	PI	438001.14	438174.12	70888.70
	PFN	442979.61	443152.60	71858.13
	RO	516004.72	516177.71	87700.58
Spearman's correlation	RN-sym	-606350.72	-606177.73	4103.34
	RS	-605773.83	-605600.84	4109.80
	RS-sym	-605773.83	-605600.84	4109.80
	RS-nn	-605123.09	-604950.10	4117.11
	RN	-604566.82	-604393.83	4123.36
	RN-nn	-604525.32	-604352.33	4123.83
	PI	-597065.65	-596892.66	4208.62
	PFN	-596480.28	-596307.29	4215.34
	RO	-547282.36	-547109.37	4820.89
OLS estimation bias	RN-sym	310210.25	310383.24	50021.78
	RS-sym	311564.65	311737.64	50206.97
	RS	311564.65	311737.64	50206.97
	RN	311685.51	311858.50	50223.52
	RN-nn	311779.58	311952.56	50236.42
	RS-nn	312556.44	312729.43	50343.01
	PI	320795.66	320968.65	51487.49
	PFN	323451.97	323624.96	51861.98
	RO	396262.88	396435.87	63258.93

Table A3.3: Relative performance of nesting metrics in explaining transformation quality.

If RN generally "outperforms" RS as a predictor of transformation quality, then why should we bother with RS at all? Is there any added value in calculating and reporting RS scores, once we account for RN? We compared the performance of several nested models, including:

1. RN: a baseline specification with just RN in the spline function, as in equation (5).

2. $RN + RS$: an expanded, additive specification with separate splines for RN and RS.

3. $RN \times RS$: an expanded, interactive specification with separate splines for RN and RS, and a multiplicative interaction between the two splines.

To assess whether including RS in these specifications improves model fit, we performed a series of Likelihood Ratio Tests, reported in Tables A3.4-A3.5. The null hypothesis in all cases is that the more parsimonious model (e.g. RN only) fits the data just as well as the expanded model (e.g. $RN + RS$). The alternative hypothesis is that the expanded model fits the data significantly better than the restricted model. We were able to reject the null hypothesis for all of the three diagnostic measures (RMSE, correlation, OLS bias), in simulations with both intensive and extensive variables. In every instance, the ratio of the likelihoods is significantly different from 1; adding RS to the baseline specification results in lower residual deviance, and (generally) lower BIC scores.

Outcome	Model	BIC	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
log(RMSE)	RN	156451.40	91665.00	29515.85			
	RN+RS	156360.09	91662.00	29475.45	3	40.41	<0.001
	RN×RS	156409.62	91653.00	29458.31	9	17.13	<0.001
Spearman's correlation	RN	-229698.15	91665.00	437.36			
	RN+RS	-230198.60	91662.00	434.82	3	2.54	<0.001
	RN×RS	-230471.67	91653.00	433.04	9	1.78	<0.001
OLS estimation bias	RN	175014.76	91665.00	36140.25			
	RN+RS	174777.61	91662.00	36033.42	3	106.84	<0.001
	RN×RS	174342.49	91653.00	35822.60	9	210.82	<0.001

Table A3.4: Likelihood ratio tests (intensive variable).

Outcome	Model	BIC	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
log(NRMSE)	RN	104818.44	91617.00	16807.33			
	RN+RS	104794.42	91614.00	16796.64	3	10.69	<0.001
	RN×RS	104763.94	91605.00	16772.22	9	24.42	<0.001
Spearman's correlation	RN	-150966.90	91617.00	1030.84			
	RN+RS	-151417.88	91614.00	1025.40	3	5.44	<0.001
	RN×RS	-152008.39	91605.00	1017.67	9	7.73	<0.001
OLS estimation bias	RN	78096.19	91617.00	12555.88			
	RN+RS	77818.22	91614.00	12513.17	3	42.71	<0.001
	RN×RS	77733.59	91605.00	12487.60	9	25.57	<0.001

Table A3.5: Likelihood ratio tests (extensive variable).

One of the reasons why RS does not appear redundant in the Likelihood Ratio Tests may be that RS and RN are capturing conceptually different geometric properties — (dis)aggregation vs. nesting — and the two scores occasionally numerically diverge. As we have seen in Table 1 of the main text, it is possible to obtain a (near-)perfect RS score in the absence of perfect nesting. Such cases, judging by our simulations and real-world examples, are not uncommon in practice. They can arise due to both measurement error (e.g. small misalignments due to an imprecise representation of border features) and structural differences between source and destination units (e.g. as in the grid-to-constituency example in Table 1).

Figure A3.1 shows histograms of the distributions of RS and RN values across our Monte Carlo simulations, along with a scatterplot of RS as a function of RN. The two distributions have very different shapes. RS has a bimodal distribution, with peaks around $RS = 0$ and $RS = 1$. RN appears more normally distributed, with a single mode around $RN = 0.5$ and almost no values at the extremes of $RN = 0$ or $RN = 1$. The relationship between the two measures resembles a logistic curve, in which $RS < RN$ for values of $RN < 0.5$ and $RS > RN$ for $RN > 0.5$. We can parameterize this relationship as follows,

$$\widehat{RS} = \left(1 + e^{6.2 - 12.75 \cdot RN}\right)^{-1}$$

where -6.2 and 12.75 are intercept and slope estimates from a logit regression of RS on RN. This fitted curve appears as a solid black line in the rightmost pane of Figure A3.1.

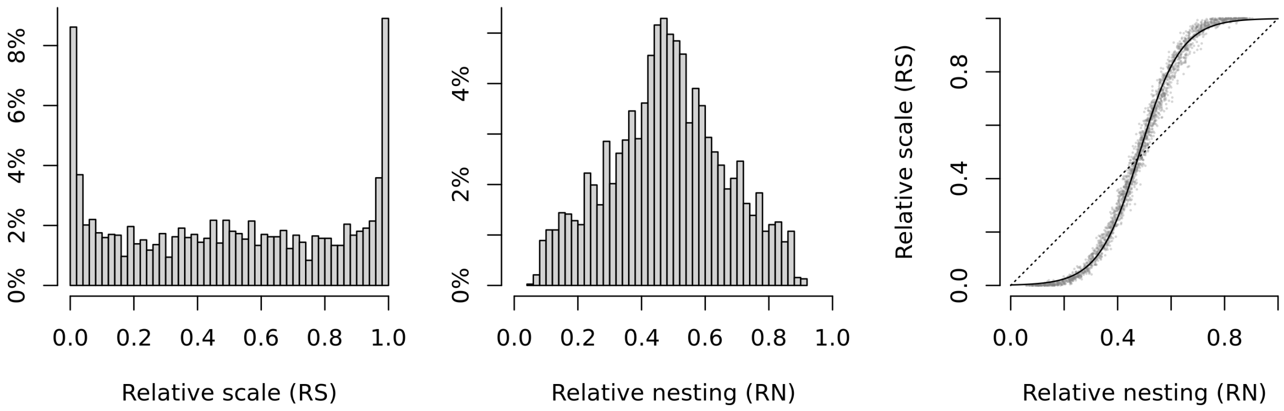


Figure A3.1: How the distributions of RS and RN differ from each other.

This analysis suggests that divergence between RN and RS, while systematic, tends to be limited to a relatively narrow range. The largest absolute differences $|RS - RN|$ are below 0.3 in our simulations (0.35 in our analysis of election data), and there are no cases in which $RS > 0.5$ and $RN < 0.5$ (or vice versa).

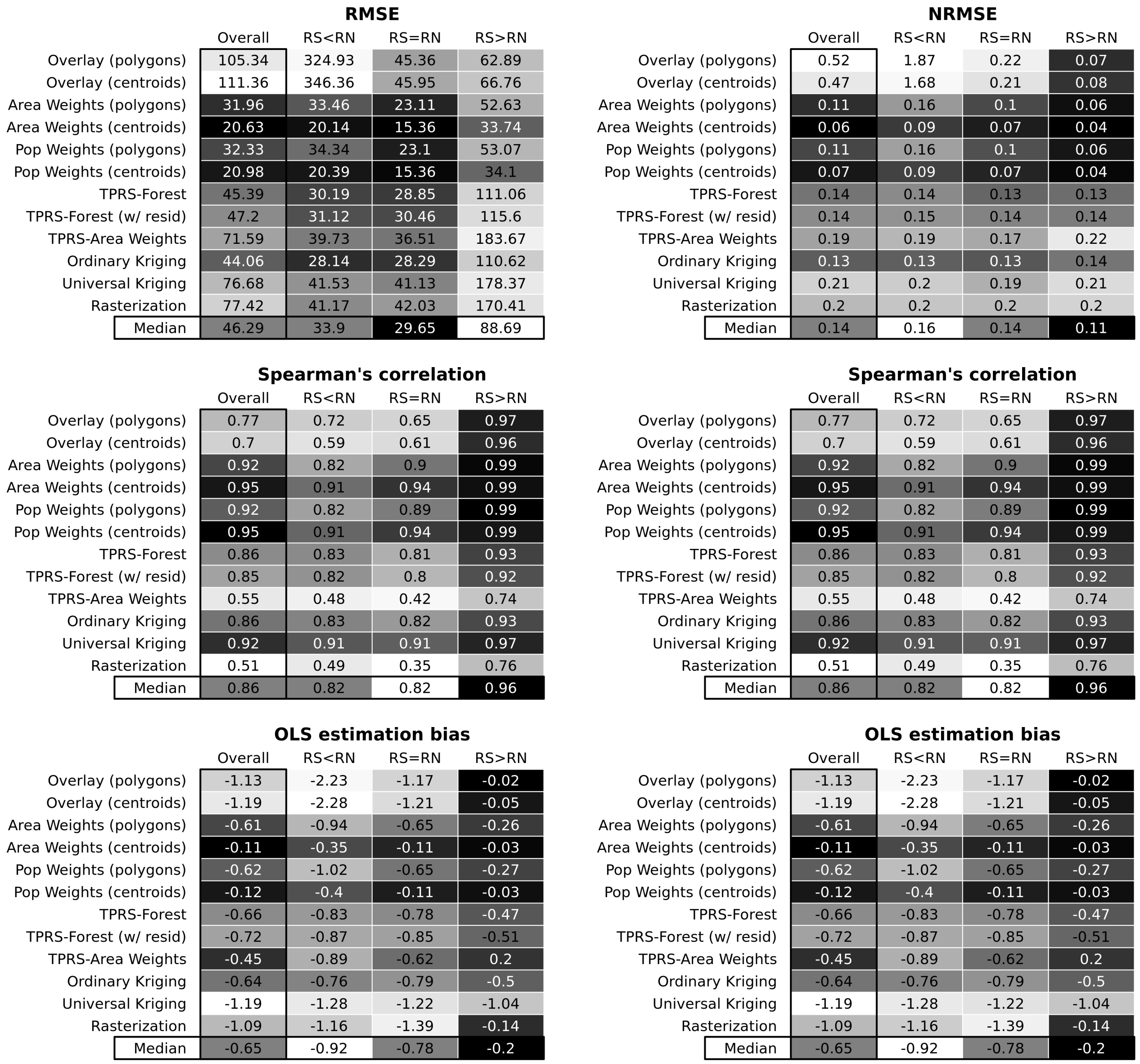
Figure A3.2 takes stock of how such divergences might impact analysis — and whether some CoS methods outperform others in such instances. The values in the cells represent average (N)RMSE, correlation and OLS bias in three subsets of simulations:

1. $RS < RN$, where $RS = 0$ *and* $RN \geqslant 0.16$ (90th percentile) (i.e. bottom-left of the scatterplot in Figure A3.1, where the curve begins to turn upward).

2. $RS = RN$, where $0.45 \leqslant RS \leqslant 0.55$ *and* $0.45 \leqslant RN \leqslant 0.55$.

3. $RS > RN$, where $RS = 1$ *and* $RN \leqslant 0.79$ (10th percentile) (i.e. top-right of the scatterplot in Figure A3.1, where the curve begins to flatten).

Figure A3.2 suggests that no CoS method has a clear comparative advantage in cases where RN and RS diverge. As we have already established, transformation quality generally improves as RN and RS increase, so the statistics where $RS = 1 > RN$ unsurprisingly look more favorable than those where $RS = 0 < RN$. The relative performance of the CoS methods, moreover, does not radically change across these subsets of simulations. With some minor exceptions (e.g. simple overlays with centroids when $RS < RN$), no CoS method emerges as a local winner. Methods that fare relatively poorly overall, like population-weighted interpolation, also perform quite poorly in these more specific instances of divergence as well.



RMSE

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	105.34	324.93	45.36	62.89
Overlay (centroids)	111.36	346.36	45.95	66.76
Area Weights (polygons)	31.96	33.46	23.11	52.63
Area Weights (centroids)	20.63	20.14	15.36	33.74
Pop Weights (polygons)	32.33	34.34	23.1	53.07
Pop Weights (centroids)	20.98	20.39	15.36	34.1
TPRS-Forest	45.39	30.19	28.85	111.06
TPRS-Forest (w/ resid)	47.2	31.12	30.46	115.6
TPRS-Area Weights	71.59	39.73	36.51	183.67
Ordinary Kriging	44.06	28.14	28.29	110.62
Universal Kriging	76.68	41.53	41.13	178.37
Rasterization	77.42	41.17	42.03	170.41
Median	46.29	33.9	29.65	88.69

Spearman's correlation

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	0.77	0.72	0.65	0.97
Overlay (centroids)	0.7	0.59	0.61	0.96
Area Weights (polygons)	0.92	0.82	0.9	0.99
Area Weights (centroids)	0.95	0.91	0.94	0.99
Pop Weights (polygons)	0.92	0.82	0.89	0.99
Pop Weights (centroids)	0.95	0.91	0.94	0.99
TPRS-Forest	0.86	0.83	0.81	0.93
TPRS-Forest (w/ resid)	0.85	0.82	0.8	0.92
TPRS-Area Weights	0.55	0.48	0.42	0.74
Ordinary Kriging	0.86	0.83	0.82	0.93
Universal Kriging	0.92	0.91	0.91	0.97
Rasterization	0.51	0.49	0.35	0.76
Median	0.86	0.82	0.82	0.96

OLS estimation bias

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	-1.13	-2.23	-1.17	-0.02
Overlay (centroids)	-1.19	-2.28	-1.21	-0.05
Area Weights (polygons)	-0.61	-0.94	-0.65	-0.26
Area Weights (centroids)	-0.11	-0.35	-0.11	-0.03
Pop Weights (polygons)	-0.62	-1.02	-0.65	-0.27
Pop Weights (centroids)	-0.12	-0.4	-0.11	-0.03
TPRS-Forest	-0.66	-0.83	-0.78	-0.47
TPRS-Forest (w/ resid)	-0.72	-0.87	-0.85	-0.51
TPRS-Area Weights	-0.45	-0.89	-0.62	0.2
Ordinary Kriging	-0.64	-0.76	-0.79	-0.5
Universal Kriging	-1.19	-1.28	-1.22	-1.04
Rasterization	-1.09	-1.16	-1.39	-0.14
Median	-0.65	-0.92	-0.78	-0.2

(a) Intensive variable (Gaussian)

NRMSE

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	0.52	1.87	0.22	0.07
Overlay (centroids)	0.47	1.68	0.21	0.08
Area Weights (polygons)	0.11	0.16	0.1	0.06
Area Weights (centroids)	0.06	0.09	0.07	0.04
Pop Weights (polygons)	0.11	0.16	0.1	0.06
Pop Weights (centroids)	0.07	0.09	0.07	0.04
TPRS-Forest	0.14	0.14	0.13	0.13
TPRS-Forest (w/ resid)	0.14	0.15	0.14	0.14
TPRS-Area Weights	0.19	0.19	0.17	0.22
Ordinary Kriging	0.13	0.13	0.13	0.14
Universal Kriging	0.21	0.2	0.19	0.21
Rasterization	0.2	0.2	0.2	0.2
Median	0.14	0.16	0.14	0.11

Spearman's correlation

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	0.77	0.72	0.65	0.97
Overlay (centroids)	0.7	0.59	0.61	0.96
Area Weights (polygons)	0.92	0.82	0.9	0.99
Area Weights (centroids)	0.95	0.91	0.94	0.99
Pop Weights (polygons)	0.92	0.82	0.89	0.99
Pop Weights (centroids)	0.95	0.91	0.94	0.99
TPRS-Forest	0.86	0.83	0.81	0.93
TPRS-Forest (w/ resid)	0.85	0.82	0.8	0.92
TPRS-Area Weights	0.55	0.48	0.42	0.74
Ordinary Kriging	0.86	0.83	0.82	0.93
Universal Kriging	0.92	0.91	0.91	0.97
Rasterization	0.51	0.49	0.35	0.76
Median	0.86	0.82	0.82	0.96

OLS estimation bias

	Overall	RS<RN	RS=RN	RS>RN
Overlay (polygons)	-1.13	-2.23	-1.17	-0.02
Overlay (centroids)	-1.19	-2.28	-1.21	-0.05
Area Weights (polygons)	-0.61	-0.94	-0.65	-0.26
Area Weights (centroids)	-0.11	-0.35	-0.11	-0.03
Pop Weights (polygons)	-0.62	-1.02	-0.65	-0.27
Pop Weights (centroids)	-0.12	-0.4	-0.11	-0.03
TPRS-Forest	-0.66	-0.83	-0.78	-0.47
TPRS-Forest (w/ resid)	-0.72	-0.87	-0.85	-0.51
TPRS-Area Weights	-0.45	-0.89	-0.62	0.2
Ordinary Kriging	-0.64	-0.76	-0.79	-0.5
Universal Kriging	-1.19	-1.28	-1.22	-1.04
Rasterization	-1.09	-1.16	-1.39	-0.14
Median	-0.65	-0.92	-0.78	-0.2

(b) Extensive variable (Poisson)

Figure A3.2: **Transformation quality when RN and RS diverge**

A4. Overview of change-of-support methods

Simple overlays

We consider two types of overlay methods: polygon-to-polygon and point-to-polygon. For point coordinates in the latter transformation, we used polygon centroids.

- *Simple overlay (polygons).* The source layer $\mathcal{G}_S$ is a set of polygons, representing administrative units, constituencies or other discrete areas of interest. The destination layer $\mathcal{G}_D$ is a second, different set of polygons. For each destination polygon $j \in \{1, \ldots, N_S\}$, the algorithm identifies the source polygon $i \in \{1, \ldots, N_S\}$ with which it overlaps. If i overlaps with multiple destination polygons, we assign it to the destination polygon with which it shares the largest areal overlap. For each destination unit j, the algorithm then computes statistics (e.g. sum, mean) for the source polygon $i \cap j$ assigned to it.

- *Simple overlay (centroids).* The source layer $\mathcal{G}_S$ is a set of points, representing the centroids of polygons or other fixed address (e.g. event location). The destination layer $\mathcal{G}_D$ is a set of polygons. For each polygon j, the algorithm identifies the set of points that fall within it, and calculates statistics for these overlapping points $i \cap j$.

Simple overlays are the industry standard for the aggregation of event data, which are typically stored as point locations. Its primary advantages are its speed and ease of implementation, which requires no re-weighting or geostatistical modeling. Its primary disadvantage is its one-to-one or many-to-one mapping of source-to-destination units, which can generate missing values in $\mathcal{G}_D$, particularly when $\mathcal{G}_S$ are points, when $N_S \ll N_D$, or when destination units are smaller in area relative to source units. This is less of a problem when missing values can be treated as "true zeroes" (e.g. event counts). It is more of a problem for most other social science applications (e.g. votes, surveys).

Area-weighted interpolation

We consider two variants of areal interpolation: polygon-to-polygon and point-to-polygon. For point coordinates in the latter transformation, we used polygon centroids.

- *Area weights (polygons).* The source layer $\mathcal{G}_S$ and destination layer $\mathcal{G}_D$ are sets of (different) polygons, representing administrative units, constituencies or other discrete areas of interest. The algorithm intersects the two polygon layers, creating a third polygon layer $\mathcal{G}_{S \cap D}$, where each feature $i \cap j \in \{1, \ldots, N_{S \cap D}\}$ is a part of source polygon i that falls inside destination polygon j. The algorithm then computes area weights, proportional to the share of j's area contributed by each source polygon. Each intersection $i \cap j$ receives weight $w_{i \cap j}^{\text{(area)}} = \frac{a_{i \cap j}}{a_j}$, where $a_{i \cap j}$ is the area of $i \cap j$ and a_j is the area of j. For each polygon j, the algorithm calculates weighted statistics for overlapping source features. For intensive variables, these statistics are typically weighted averages of values in intersections, $x_j = \sum_{i \cap j} w_{i \cap j}^{\text{(area)}} x_{i \cap j}$, where $x_{i \cap j}$ is the value of some variable x in intersection $i \cap j$. For extensive variables, these statistics are typically sums of values in all constituent intersections, $x_j = \sum_{i \cap j} x_{i \cap j}$, adjusted so as to satisfy the pycnophylactic (mass-preserving) property.

- *Area weights (centroids).* The source layer $\mathcal{G}_S$ is a set of points, representing the centroids of polygons or other fixed address (e.g. event location). The destination layer $\mathcal{G}_D$ is a set of polygons. This method includes an additional, intermediate step to convert the point features into polygons, by creating a Voronoi tessellation of the study area. During the tessellation stage, the algorithm creates N_S polygons, such that for any polygon l_i corresponding to point i, all points inside l_i are closer to i than to any other point $-i$. This is followed by a polygon-to-polygon interpolation stage, as described in the previous paragraph.

Areal weighting is the default CoS method built in to many commercial and open-source Geographic Information Systems. In contrast to simple overlays, interpolation by design leaves no gaps or missing regions. It is also easy to implement and requires information only on the geometries of source and destination units, with no need for ancillary data. Its point-to-polygon variant is particularly attractive if the boundaries of source units are unknown. However, this method rests on several important assumptions, which are fully satisfied in only very rare cases. Most notably, it assumes that the phenomenon of interest is uniformly distributed in source polygons. The point-to-polygon variant is also sensitive to assumptions about boundary placement made in the creation of the synthetic tessellated polygons.

Population-weighted interpolation

We consider two variants of population-weighted interpolation: polygon-to-polygon and point-to-polygon. As above, we used polygon centroids as point coordinates.

- *Population weights (polygons).* This method requires three spatial data layers. The source layer $\mathcal{G}_S$ and destination layer $\mathcal{G}_D$ are sets of (different) polygons, representing administrative units, constituencies or other discrete areas of interest. The third, ancillary layer is a raster of population levels Π, which fully overlaps with the area of $\mathcal{G}_S$ and $\mathcal{G}_D$. The algorithm intersects the two polygon layers, creating a third polygon layer $\mathcal{G}_{S \cap D}$, where each feature $i \cap j \in \{1, \ldots, N_{S \cap D}\}$ is a part of source polygon i that falls inside destination polygon j. The algorithm then computes population weights, proportional to the share of j's population contributed by each source polygon. Each intersection $i \cap j$ receives weight $w^{(\text{pop})}_{i \cap j} = \frac{p_{i \cap j}}{p_j}$, where $p_{i \cap j}$ is the population count of intersection $i \cap j$ and p_j is the population of j. For each polygon j, the algorithm calculates weighted statistics for overlapping source features, as described above.

- *Population weights (centroids).* The source layer $\mathcal{G}_S$ is a set of points, representing the centroids of polygons or other fixed address (e.g. event location). The destination layer $\mathcal{G}_D$ is a set of polygons. The third, ancillary layer is the raster of population levels Π. As with the second, point-based area-weighting method, this method includes an intermediate step to convert the point features into tessellated polygons, and calculates population levels for each intersection between these polygons and $\mathcal{G}_D$. This is followed by a population-weighted polygon-to-polygon interpolation stage, as described in the previous paragraph.

This is an extension of the area weighting method, which dispenses with the uniformity assumption and seeks to account for variation in the underlying distribution of x. Although we use the term "population" here, this method is extensible to any set of ancillary data that researchers consider to be predictive of this distribution. The primary disadvantages are data scarcity (e.g. contemporaneous population data are not always available), the assumption that this ancillary layer is indeed predictive of x, as well as assumptions made during the tessellation stage in the point-based version.

Thin-plate regression spline methods

We consider two methods that employ thin-plate regression splines: TPRS-Forest estimation and TPRS-Areal Weighting. These methods allow researchers to use external variables when interpolating between source and designation polygons. In addition, these methods provide a way to construct uncertainty measures for the designation unit by exploiting geographic variation in the XY-coordinates of the source polygon units. Finally, these methods provide a way to interpolate spatial data with non-Gaussian error distributions, such as binary, count, or categorical data. These two methods differ, however, in how they handle information from the source polygon. TPRS-Forest estimation excludes random noise fluctuations, or non-systematic variation, when mapping from the source to designation polygon. TPRS-Areal Weighting, on the other hand, interpolates all information, including random noise fluctuations, from the source to designation polygon.

Thin-plate regression splines (TPRS) (Duchon, 1977; Wood, 2003) estimate a nonparametric smooth function $f(\cdot)$ — in our case, $f(\text{Long}, \text{Lat})$ — by minimizing

$$||\mathbf{y} - \mathbf{f}|| + \lambda J_{md}(f)$$

where $\mathbf{y}$ is a vector of y_i's, $\mathbf{f} = |f(x_1), \ldots, f(x_n)|'$, $\mathbf{x}$ is an $N_{S \cap D} \times d$ matrix of predictors (in this case, longitude and latitude), $||\cdot||$ is the Euclidean norm, and λ is a smoothing parameter governing the model degrees of freedom, which can be selected through generalized cross-validation or the Akaike Information Criterion. J_{md} is a "wiggliness penalty" for f:

$$J_{md} = \int \ldots \int_{\mathcal{R}_d} \sum_{\upsilon_S! \ldots \upsilon_d! = m} \frac{m!}{\upsilon_S! \ldots \upsilon_d!} \left(\frac{\delta^m f}{\delta x_1^{\upsilon_S} \ldots \delta x_d^{\upsilon_d}} \right) dx_1 \ldots dx_d$$

where m is the order of differentiation, satisfying $2m > d$. In our two-predictor case, the wiggliness penalty becomes

$$J_{22} = \int \int \left(\frac{\delta^2 f}{\delta \text{Long}^2} \right)^2 + \left(\frac{\delta^2 f}{\delta \text{Lat}^2} \right)^2 + 2 \left(\frac{\delta^2 f}{\delta \text{Long}^2 \text{Lat}^2} \right)^2 d\text{Long} d\text{Lat}$$

The advantage of thin-plate regression splines is that they avoid the knot placement problems of conventional regression spline modeling, reducing the subjectivity of the model fitting process. They also nest smooths of lower rank within smooths of higher rank.

- *TPRS-Forest Estimation.* This process begins by fitting a thin-plate regression spline to the source polygon and predicting conditional mean and standard error estimates to the designation polygon units. Next, we fit a random forest model to the source polygon and predict conditional mean estimates to the designation polygon units.

- *TPRS-Areal Weighting.* This process begins by intersecting the source and designation polygon units. Next, we fit a thin-plate regression spline to the source polygon and predict conditional mean and standard error estimates to the designation polygon units. Third, we conduct simple areal weighting using the TPRS residuals and the shape boundaries of the source and designation polygon units. Finally, we construct error bounds by bootstrapping the estimated values: conditional mean, areal weighted residuals, and standard error.

Kriging methods

We consider two block kriging methods in the main text: ordinary and universal. The primary difference is that the second requires ancillary data, while the first does not.

- *Ordinary Kriging.* The source layer $\mathcal{G}_S$ is a set of points, representing (in our case) the centroids of polygons. However, the points can represent any other fixed address (e.g. sampling or event location). The destination layer $\mathcal{G}_D$ is a set of polygons. At the point level, ordinary kriging interpolates a value $Z(x_0)$ of random field $Z(x)$ at unobserved location x_0, using data from observed location x_i. The kriging estimator is:

$$\hat{Z}(x_0) = \sum_{i=1}^{n} w_i(x_0) Z(x_i)$$

where $w_i(x_0),\ i = 1, \ldots, n$ is a spatial weight. These weights are based on a variogram model, which describes the degree to which nearby locations have similar values:

$$\hat{\gamma}(d) = \frac{1}{2n(d)} \sum_{d_{iq}=d} (Z(x_i) - Z(x_q))^2$$

where $\gamma\hat{(d)}$ is estimated semivariance, $n(d)$ is number of point pairs (x_i, x_q) separated by distance d, and $Z(x_i)$ is value of a variable at location x_i. As locations become farther apart, they should become more dissimilar and have higher semivariance $\gamma(d)$. We select a variogram model appropriate to the data by minimizing the sum of squared residuals from the sample. To interpolate at point x_0 based on points $x_1, \ldots, x_{N_S}$, the weights $w_1, \ldots w_{N_S}$ must be found, by solving the system of linear equations:

$$\begin{bmatrix} \gamma(d_{11}) & \gamma(d_{12}) & \cdots & \gamma(d_{1N_S}) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(d_{N_S1}) & \gamma(d_{N_S2}) & \cdots & \gamma(d_{N_SN_S}) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_{N_S} \\ \lambda \end{bmatrix} \begin{bmatrix} \gamma(d_{10}) \\ \vdots \\ \gamma(d_{N_S0}) \\ 1 \end{bmatrix}$$

where $\gamma(d_{ij})$ is the semivariance for the distance between points x_i and x_j, and λ is the trend parameter. Ordinary kriging assumes an unknown constant trend: $\lambda(x) = \lambda$. Point-level interpolation by ordinary kriging is given by:

$$\hat{Z}(x_0) = \begin{pmatrix} w_1 \\ \vdots \\ w_{N_S} \end{pmatrix}' \begin{pmatrix} Z(x_1) \\ \vdots \\ Z(x_{N_S}) \end{pmatrix}$$

Ordinary kriging error is:

$$var\left(\hat{Z}(x_0) - Z(x_0)\right) = \begin{pmatrix} w_1 \\ \vdots \\ w_{N_S} \\ \lambda \end{pmatrix}' \begin{pmatrix} \gamma(d_{10}) \\ \vdots \\ \gamma(d_{N_S0}) \\ 1 \end{pmatrix}$$

This approach can be extended to yield predictions for areal units, via *block kriging* (Cressie, 1993; Chiles and Delfiner, 2009). Let B be an area ("block") that forms the spatial support of $Z(B)$. These blocks can be regularly-shaped grid cells, or irregular polygons. In our case, we can specify a separate block B_j for each destination polygon in $\mathcal{G}_D$. The block kriging predictor is a weighted average of point-level measurements

$$\hat{Z}(B_j) = \sum_{i=1}^{n} w_i(B_j) Z(x_i)$$

This approach is equivalent to predicting multiple points in region B_j, and averaging those values over B_j (Young et al., 2009). Using the predicted values of this random field, the algorithm computes statistics (e.g. means, sums) for each destination polygon j, preserving the pycnophylactic property for extensive variables as appropriate.

- *Universal Kriging.* This method requires three or more spatial data layers. As above, the source layer $\mathcal{G}_S$ is a set of points (e.g. centroids) and destination layer $\mathcal{G}_D$ is a set of polygons. The third, ancillary layer is a raster of population levels Π, which fully overlaps with the bounding box of $\mathcal{G}_S$ and $\mathcal{G}_D$. The algorithm interpolates a value $Z(x_0)$ of random field $Z(x)$ at unobserved location x_0, using data from observed location x_i *and* population values observed at x_0 and x_i. We then extend this approach to estimate block averages for each polygon in $\mathcal{G}_D$.

Kriging is widely used in natural and environmental sciences as a solution to the change-of-support problem (Gotway and Young, 2007). Unlike the interpolation and overlay methods, this is a model-based approach, which can ascertain the uncertainty of estimates. However, kriging is highly sensitive to variogram model selection, and some of its assumptions (particularly regarding the smoothness of interpolated values over space) can be problematic for social science.

A5. Analysis of Swedish electoral data

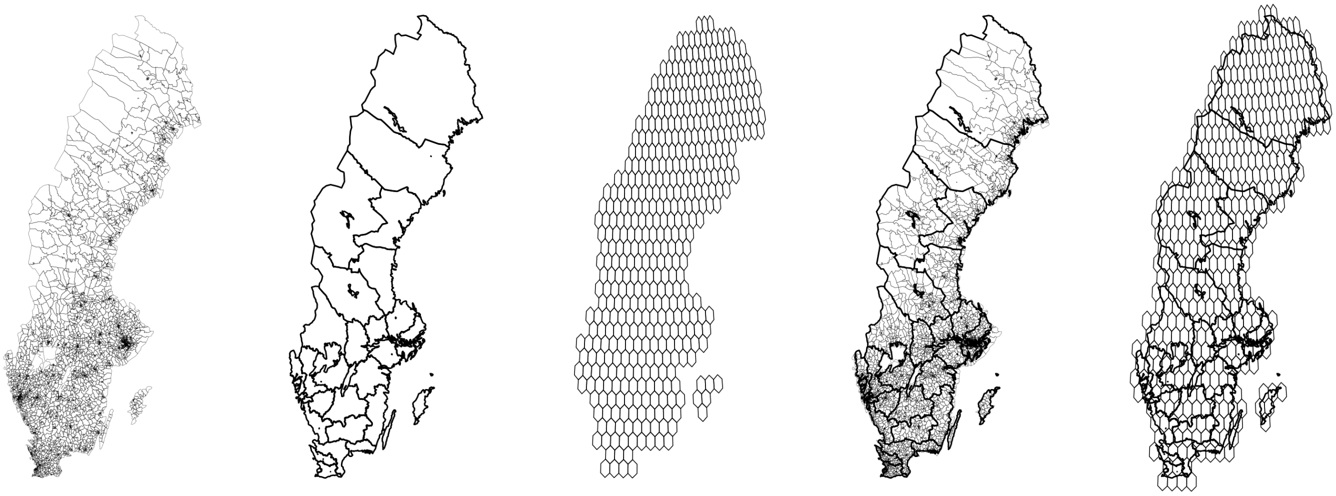
The current section replicates the main text's CoS analysis of electoral data from the U.S. state of Georgia, with analogous data from Sweden's 2010 Riksdag (unicameral legislature) elections. Figure A5.3 shows the spatial data layers used in this analysis, which correspond to those in Figure 1 (precincts, constituencies, 0.5° hexagonal grid cells).[1]

[1] The precincts boundaries are from data.val.se/val/val2010/statistik. The constituency boundaries are from Kollman et al. (2017).

Table A5.6 reports relative scale and nesting coefficients for these polygons (counterpart to Table 1). Notably, as Table A5.6b shows, while precincts should be fully nested within constituencies "in real life," this is not technically the case in the geospatial data ($RN = 0.90$). The nesting coefficient for precincts-constituencies is about the same as it is for precincts-grid, although RS is larger for the former pair. This surprisingly low RN is likely due to measurement error, the differential precision of the two geospatial boundary datasets, and other discrepancies (e.g. coastal features, bodies of water).

Figure A5.4 illustrates several examples of transformed values of Top-2 Competitiveness alongside true values, for (a) precinct-to-constituency and (b) constituency-to-grid CoS. Figure A5.5 reports fit diagnostics for CoS transformations of Swedish election results. The results here are consistent with those for Georgia (Figure 3).

Figure A5.3: **Spatial data layers** (Sweden)



(a) Precincts (b) Constituencies (c) .5° grid (d) a + b (e) c + b

Table A5.6: **Relative scale and nesting of polygons in Figure A5.3.**

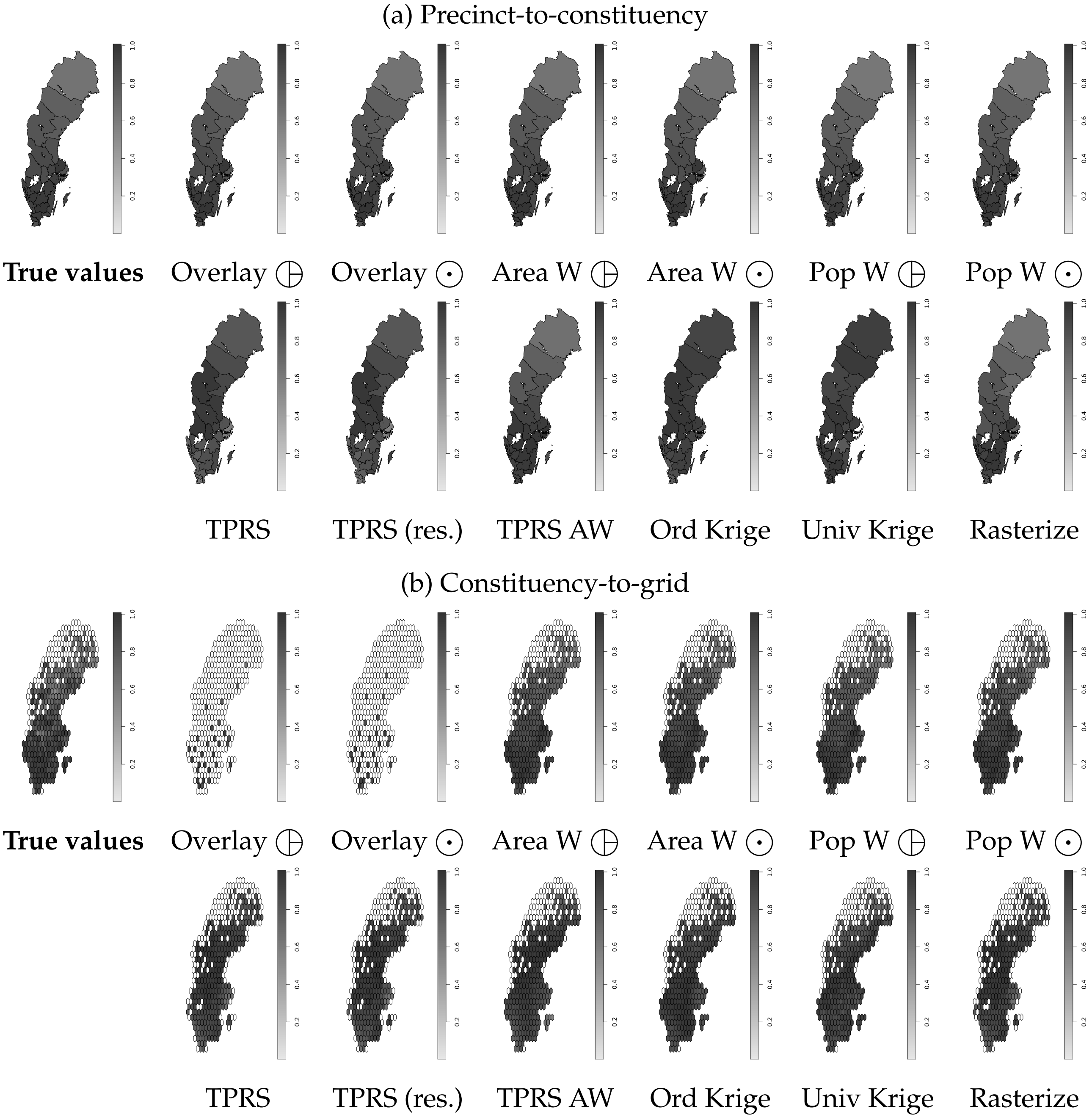
(a) Relative nesting (RN)

Source	Destination		
	(a)	(b)	(c)
(a) Precincts	–	0.90	0.90
(b) Constituencies	0.02	–	0.19
(c) .5° grid	0.27	0.66	–

(b) Relative scale (RS)

Source	Destination		
	(a)	(b)	(c)
(a) Precincts	–	1.00	0.94
(b) Constituencies	0.00	–	0.01
(c) .5° grid	0.09	0.99	–

Figure A5.4: **Output from change-of-support operations** (Sweden). ⊕: source features are polygons. ⊙: source features are polygon centroids.



(a) Precinct-to-constituency

(b) Constituency-to-grid

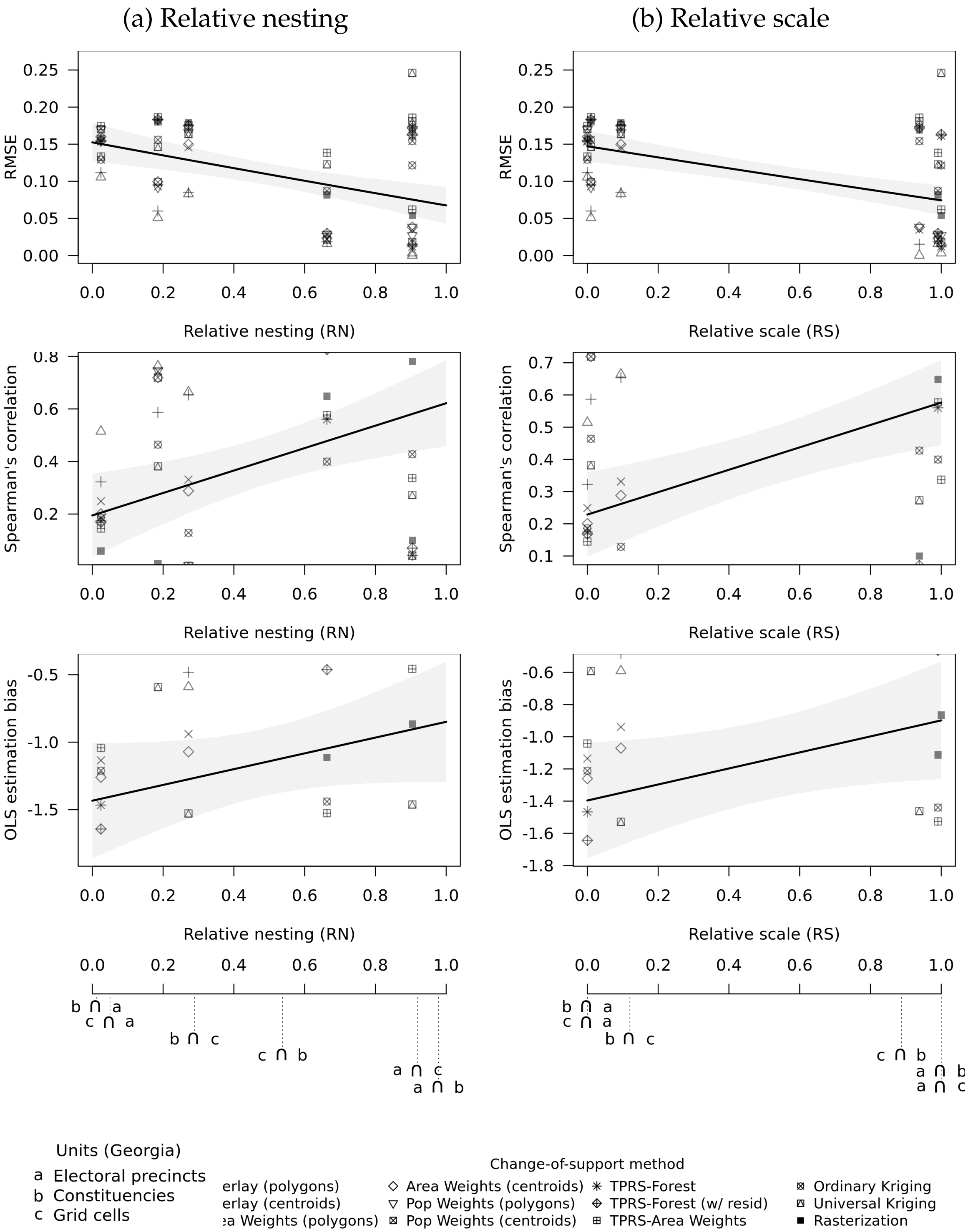


Figure A5.5: **Relative nesting, scale, and transformations of election data** (Sweden)

A6. Direct vs. indirect transformations of intensive variables

The conceptual distinction between intensive and extensive variables can obscure the fact that intensive variables are often composites of multiple extensive variables. While some intensive variables, like air temperature, can be directly observed "in the wild," others are functions of extensive variables divided or normalized by other extensive variables, like volume, area or population count. For example, in order to establish a party's vote share in an election (an intensive variable), one must first observe the total number of valid votes cast (an extensive variable) along with the number of valid votes the party received (another extensive variable), and divide the latter by the former. To take another example, Top-2 Competitiveness — the variable we used in our electoral data analysis — can be calculated from several combinations of components:

$$\begin{aligned}\text{Top-2 Competitiveness} &= 1 - \text{winning party vote share margin}\\ &= \frac{\text{valid votes} - \text{winning party vote count margin}}{\text{valid votes}}\\ &= \frac{\text{valid votes} - (\text{votes for winner} - \text{votes for runner-up})}{\text{valid votes}}\end{aligned}$$

of which only "winning party vote share margin" is an intensive variable, albeit one which is itself a function of multiple constitutive extensive variables. Indeed, some extensive variables are also composites: "vote count margin" is the difference between the vote counts of the winner and runner-up.

What is the appropriate way to handle CoS operations for composite intensive variables? The choice boils down to the following:

1. **Direct transformation**: perform CoS operations directly on the intensive variable, rather than its extensive components. In our example, this means treating competitiveness as a single, self-contained variable, and attempting to calculate its average values in destination units.

2. **Indirect transformation**: perform CoS operations on a variable's extensive components, and use the transformed values of these individual components to reconstruct the intensive variable within destination units. The transformed values for all extensive variables must satisfy the pycnophylactic (mass-preserving) property. In our case, this means calculating sums of valid votes in each destination unit, identifying winners and runners-up in destination units by ranking the parties by aggregate number of votes they each received, and then using these components to calculate competitiveness scores within destination units.

The choice between these two approaches depends in part on data availability. Direct transformation is suitable for a "data-poor" scenario, where the researcher only has access to a composite measure and not the underlying variables used to construct it. Indirect transformation, which we adopted in the main text, is better-suited for a "data-rich" scenario, where the researcher has access to the full complement of component variables. Yet data availability is not the only consideration that is relevant here.

As Table A6.7 shows, the comparative advantages of direct vs. indirect transformations depend on the relative nesting and scale of source and destination units. When RS and RN are high, as in the case of precinct-to-constituency transformations in Georgia (a to b), the indirect approach outperforms the direct one. Errors are generally smaller, and correlations slightly higher, when using multiple CoS operations on individual extensive components rather than a single CoS operation on the composite measure. When RS and RN are lower, as in virtually all other cases in Table A6.7, transforming the single composite measure yielded more reliable results than transforming individual components.

Diagnostic	Source	Destination	RS	RN	Direct	Indirect
RMSE	a. Precincts	b. Constituencies	1.00	0.98	1.20	1.17
	a. Precincts	c. Grid cells	1.00	0.92	1.21	1.18
	b. Constituencies	a. Precincts	0.00	0.01	1.32	1.49
	b. Constituencies	c. Grid cells	0.12	0.29	1.25	1.37
	c. Grid cells	a. Precincts	0.00	0.05	1.31	3.45
	c. Grid cells	b. Constituencies	0.89	0.54	1.19	1.24
Spearman's correlation	a. Precincts	b. Constituencies	1.00	0.98	0.72	0.73
	a. Precincts	c. Grid cells	1.00	0.92	0.85	0.78
	b. Constituencies	a. Precincts	0.00	0.01	0.45	0.08
	b. Constituencies	c. Grid cells	0.12	0.29	0.53	0.35
	c. Grid cells	a. Precincts	0.00	0.05	0.69	0.45
	c. Grid cells	b. Constituencies	0.89	0.54	0.74	0.60
OLS estimation bias	a. Precincts	b. Constituencies	1.00	0.98	0.98	-1.37
	a. Precincts	c. Grid cells	1.00	0.92	0.32	-0.22
	b. Constituencies	a. Precincts	0.00	0.01	0.43	-2.04
	b. Constituencies	c. Grid cells	0.12	0.29	1.31	-1.68
	c. Grid cells	a. Precincts	0.00	0.05	-0.58	-1.41
	c. Grid cells	b. Constituencies	0.89	0.54	1.43	-7.73

Table A6.7: Transformation quality when interpolating component intensive variables directly ("Direct") vs. reconstructing them from extensive components ("Indirect").

This analysis suggests that an indirect strategy of reconstructing secondary statistics from transformed constituent variables pays the most dividends in cases of aggregation across nested units. When units are less nested, the indirect approach can actually backfire — the reconstructed composite becomes less accurate because, at lower values of RN and RS, the transformations of each component themselves become less accurate. In such cases, the researcher may be better off transforming the composite measure directly.

A7. Monte Carlo study design

At each iteration, our simulations executed the following routine:

1. Draw a set of (random) source ($\mathcal{G}_S$) and destination ($\mathcal{G}_D$) polygons. Let N_S be the number of source polygons, and N_D be the number of destination polygons. Create a bounding box $\mathcal{B}$ defined by coordinate set $\{x_{\min}, x_{\max}, y_{\min}, y_{\max}\}$. Within $\mathcal{B}$, sample a random set of N_S points. Create N_S tessellated polygons such that for any polygon $j \in \{1, \ldots, N_S\}$ corresponding to point $i \in \{1, \ldots, N_S\}$, all points inside j are closer to i than to any other point $-i$. Repeat procedure for the N_D destination polygons.

2. Randomly allocate X values (e.g. votes, margins) across $\mathcal{G}_S$ and $\mathcal{G}_D$.

   - For **intensive variables** (scale-independent, like population density or vote shares), we simulated values from a Gaussian Random Field (GRF). For each unit in the intersection $\mathcal{G}_S \cap \mathcal{G}_D$, draw a value of x from a mean zero GRF $\{X_n\}_{n \in G}$, simulated with the sequential simulation algorithm (Goovaerts, 1997). We model spatial autocorrelation in this field with a variogram, which describes the degree to which nearby locations have similar values. The semivariance $\gamma(d)$ is formally defined as the squared difference in values between locations:

     $$\hat{\gamma}(d) = \frac{1}{2n(d)} \sum_{d_{ij}=d} \left(X\left(c_i\right) - X\left(c_j\right)\right)^2 \tag{A7.10}$$

     where $n(d)$ is the number of point pairs separated by distance d, and $X(c_i)$ is the value of variable $x(\mathbf{c})$ at location c_i. The variogram can be used for spatial prediction by fitting a parametric model to the variogram, specifying the model type (e.g. exponential, spherical, Matern), partial sill (magnitude of variation), range (maximum distance d), and nugget variance (micro-variability, measurement error). The range parameter here governs the degree of spatial correlation (because $n(d)$ is increasing in d, more observations influence each other as range increases).

     We fit two variogram models:

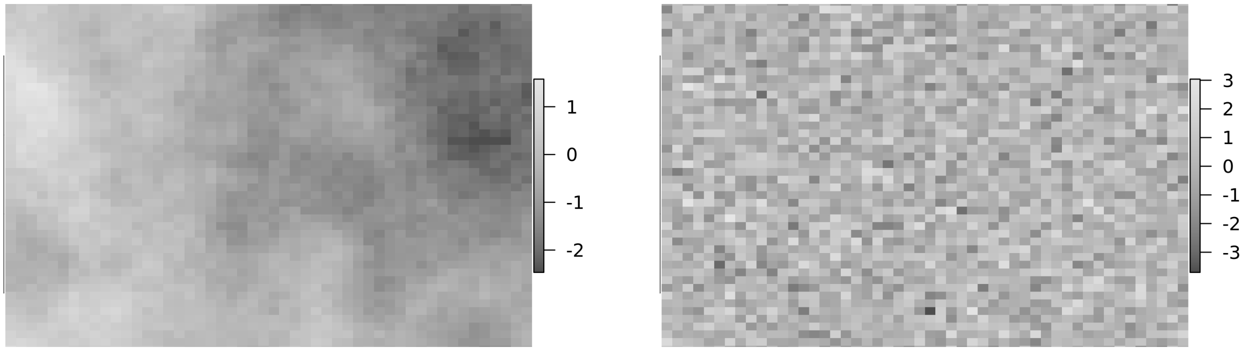
(a) *Spatially autocorrelated data*. Matern covariance semi-variogram model, with range of 2000, partial sill of 1, nugget variance of 0.

(b) *Spatially random data*. Matern covariance semi-variogram model, with range of 1, partial sill of 1, nugget variance of 0.

The sequential simulation algorithm follows a random path through locations $c_1, \dots, c_n$. At each location c_i (e.g. centroid of each intersection $\mathcal{G}_S \cap \mathcal{G}_D$), it computes the conditional distribution of $X(c_i)$ given the data and previously simulated values. It draws a value from this distribution, and assigns it to c_i. It then proceeds to the next unvisited location, until all n locations have assigned values.

Figure A7.6 illustrates two realizations of the GRF, simulated using the (a) autocorrelated and (b) spatially random variogram models.

Figure A7.6: Two examples of Gaussian Random Fields used in simulations



(a) Spatially autocorrelated

(b) Spatially random

- For **extensive variables** (scale-dependent, like population counts or event counts), we simulated values from a Poisson point process (PPP). For each unit in the intersection $\mathcal{G}_S \cap \mathcal{G}_D$, count the number of events x from a Poisson point process (PPP) model. Let $\mathcal{C}$ denote a bounded spatial region, let $A(\mathcal{C})$ represent the area of $\mathcal{C}$, let $X(\mathcal{C})$ be the number of events realized in $\mathcal{C}$, and let λ be the intensity parameter. The probability that exactly k events occur in region $\mathcal{C}$ is

$$P(X(S) = k) = \frac{(\lambda A(\mathcal{C}))^k e^{-\lambda A(\mathcal{C})}}{k!} \quad \forall A(\mathcal{C}) > 0, \; k = 0, 1, 2, \dots \qquad \text{(A7.11)}$$

We fit two PPP models:

(a) *Spatially autocorrelated data*. Inhomogeneous Poisson process, where λ(long, lat) is a function of spatial coordinates (assumes intensity is variable over $\mathcal{C}$). We used a spherical functional form, where intensity is highest in the center, and lower on the periphery:

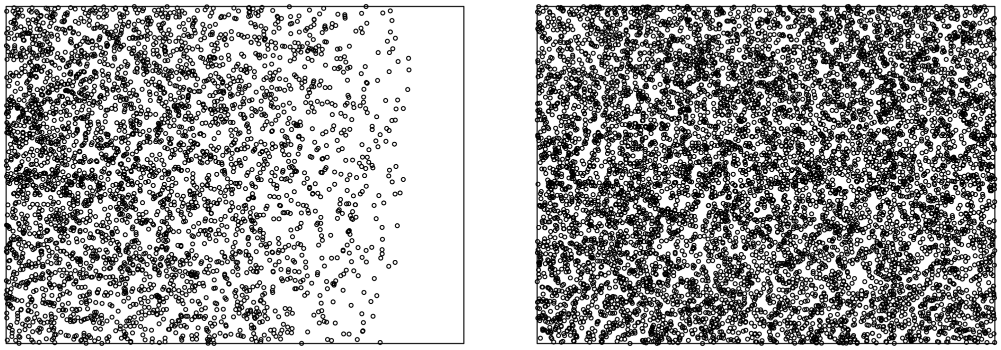
$$\lambda(\text{long}, \text{lat}) = \lambda_{\text{max}} - \lambda_0 \sqrt{(\text{long} - \text{long}_0)^2 + (\text{lat} - \text{lat}_0)^2} \tag{A7.12}$$

where $\lambda_{\text{max}} = 100$ and $\lambda_0 = 10$, and $(\text{long}_0, \text{lat}_0)$ is a central coordinate pair, whose exact location in $\mathcal{C}$ varies randomly across simulations.

(b) *Spatially random data*. Homogeneous Poisson process, where λ is a positive constant (assumes constant intensity over $\mathcal{C}$). We set $\lambda = \lambda_{\text{max}} = 100$.

Figure A7.7 illustrates two realization of the PPP, simulated using the (a) autocorrelated/inhomogeneous and (b) spatially random/homogeneous PPP models.

Figure A7.7: Two examples of Poisson Point Processes used in simulations



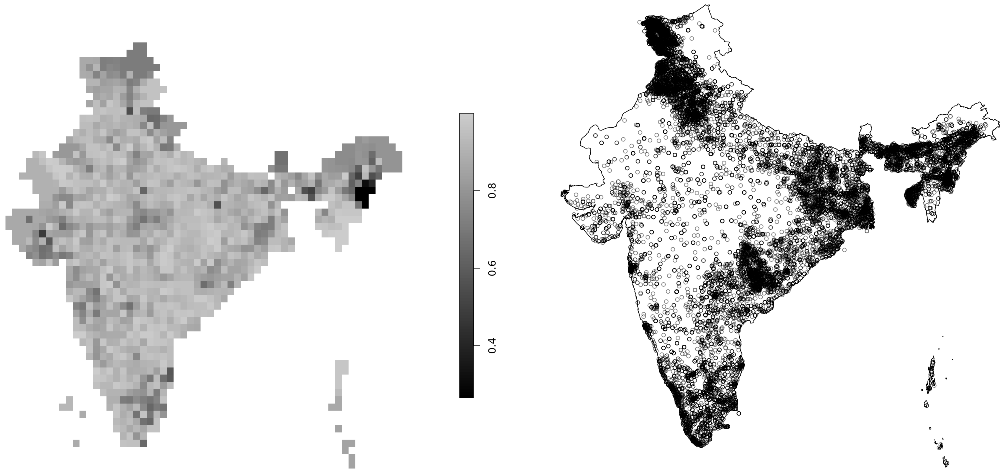
(a) Inhomogeneous PPP

(b) Homogeneous PPP

By way of comparison, Figure A7.8 shows the geographic distribution of electoral competitiveness (A7.8a) and political violence (A7.8b) in India.[2] While real-world data do not neatly align with a particular known distribution (e.g., Gaussian, Poisson), the types of clustering and heterogeneities seen here bear some resemblance to those in the spatially-autocorrelated GRF (Figure A7.6a) and inhomogeneous PPP (Figure A7.7a). For example, the heterogeneous pattern in Figure A7.7a is similar to the differential point densities in the north of India in Figure A7.8b, around Jammu and Kashmir.

[2] For this illustration, we used CLEA data from the 1996 Indian general election (Kollman et al., 2022) and xSub multi-source event data on political violence (Zhukov, Davenport and Kostyuk, 2019).

Figure A7.8: Geographic distribution of electoral and violence data in India



(a) Electoral competitiveness

(b) Political violence

After x_i values are drawn for each of the sub-units in intersection $\mathcal{G}_S \cap \mathcal{G}_D$, calculate synthetic variable $y_i = \alpha + \beta x_i + \epsilon_i$, with parameters $\alpha = 1$, $\beta = 2.5$ and $\epsilon \sim N(0, 1)$.

Calculate aggregated x and y values for each set of polygons:

- Aggregate over $\mathcal{G}_D$ to get "true values" of x, y in $\mathcal{G}_S$: $(x_{\mathcal{G}S}, y_{\mathcal{G}S})$
- Aggregate over $\mathcal{G}_S$ to get "true values" of x, y in $\mathcal{G}_D$: $(x_{\mathcal{G}D}, y_{\mathcal{G}D})$
- Repeat for two types of aggregated variables: intensive (aggregates are means), and extensive (aggregates are sums). Aggregated values for extensive variables must satisfy the pycnophylactic (mass-preserving) property.

3. Change the geographic support of X from $\mathcal{G}_S$ to $\mathcal{G}_D$, compare true values of x in $\mathcal{G}_D$ to these spatially-transformed values of x. Let K be a set of CoS algorithms (e.g. areal interpolation, kriging, etc.). Each $k \in \{1, \ldots, K\}$ specifies a mapping/transformation between geometries $\mathcal{G}_S$ and $\mathcal{G}_D$ Let $x_{\mathcal{G}D}$ be the "true" value of x in destination polygons $\mathcal{G}_D$ Let $\widehat{x_{\mathcal{G}D}}^{(k)} = f_k(x_{\mathcal{G}S})$ be the estimated value of $x_{\mathcal{G}D}$, calculated using CoS algorithm k For each k, calculate

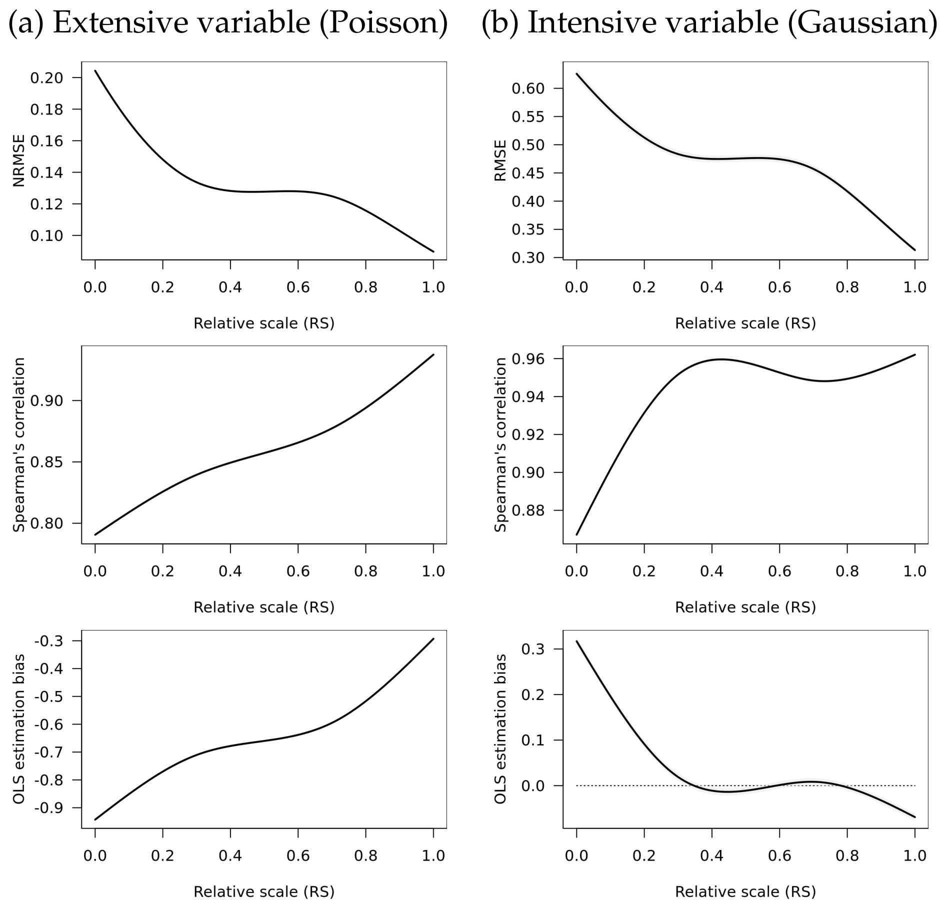
   - Root mean squared error (RMSE): $\sqrt{\sum_j \frac{1}{N_D}(x_{j\mathcal{G}D} - \widehat{x_{j\mathcal{G}D}})^2}$, for intensive variables.
   - Normalized RMSE (NRMSE): $\frac{\sqrt{\sum_j \frac{1}{N_{\mathcal{G}D}}(x_{j\mathcal{G}D} - \widehat{x_{j\mathcal{G}D}})^2}}{\max(x_{\mathcal{G}D}) - \min(x_{\mathcal{G}D})}$, for (scale-dependent) extensive variables.

- Spearman's rank correlation coefficient: $\frac{\sum_i^{N_D}(R_i(x_{\mathcal{G}D})-\bar{R}(x_{\mathcal{G}D}))(R_i(\widehat{x_{\mathcal{G}D}})-\bar{R}(\widehat{x_{\mathcal{G}D}}))}{\sqrt{\sum_i^{N_D}(R_i(x_{\mathcal{G}D})-\bar{R}(x_{\mathcal{G}D}))^2}\sqrt{\sum_i^{N_D}(R_i(\widehat{x_{\mathcal{G}D}})-\bar{R}(\widehat{x_{\mathcal{G}D}}))^2}}$, where $R_i(\cdot)$ is the rank of observation i, and $\bar{R}(\cdot)$ is the sample mean rank.
- OLS estimation bias: difference between "true" value $\beta = 2.5$ and estimate of $\hat{\beta}$ from regression of y on transformed values of x: $y_{j\mathcal{G}D} = \alpha + \beta\widehat{x_{j\mathcal{G}D}} + \epsilon$.

We ran this simulation for all $N_S \in [10, \ldots, 200]$ and $N_D \in [10, \ldots, 200]$, totalling $191^2 = 36481$ potential CoS operations, from aggregation ($N_S = 200, N_D = 10$) to disaggregation ($N_S = 10, N_D = 200$). We repeated this process 100 times, with different random seeds.

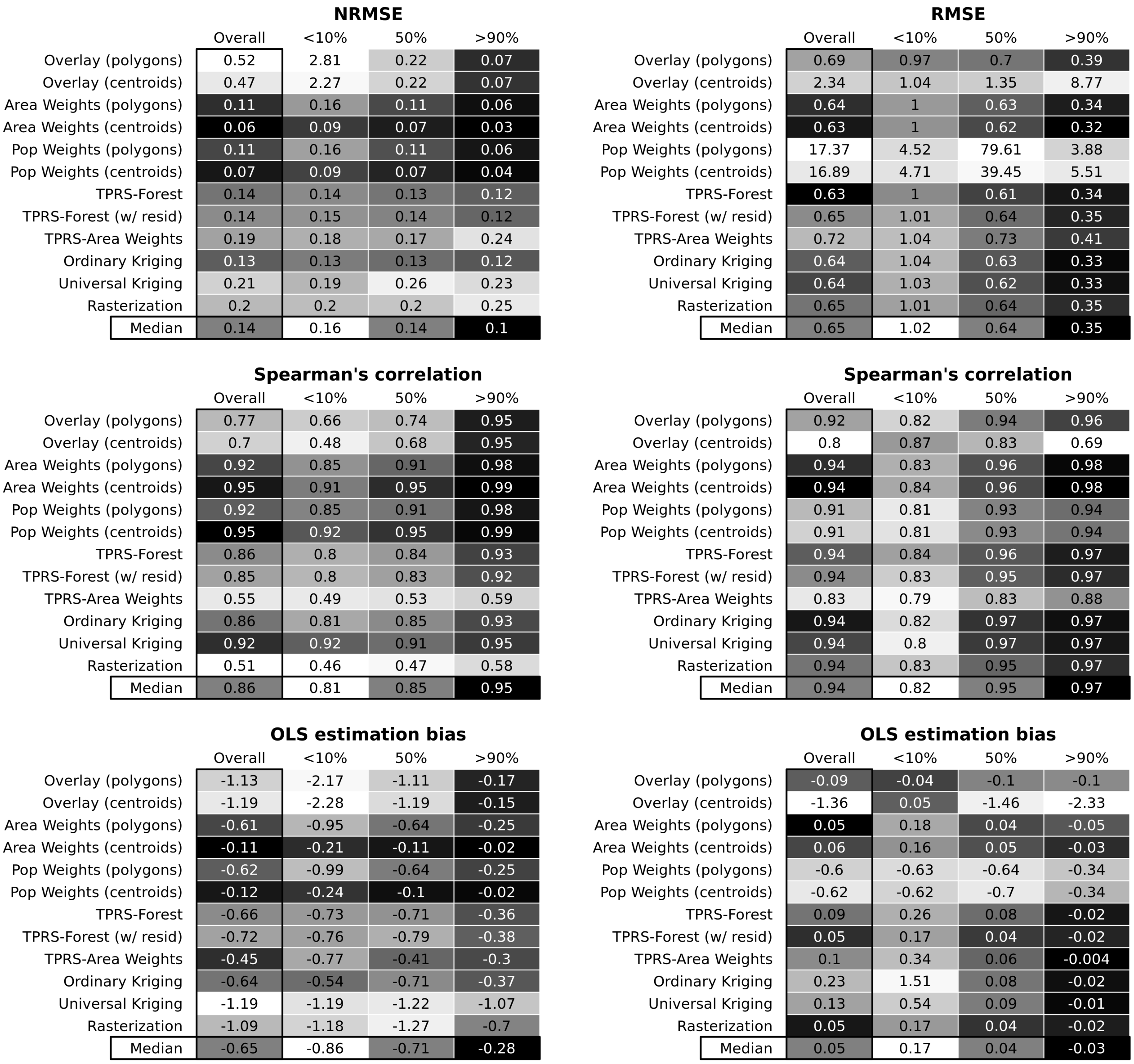
A8. Additional Monte Carlo results

The main text reports Monte Carlo results only for the RN coefficient. Figures A8.9 and A8.10 show an analogous set of results for the RS coefficient.


(a) Extensive variable (Poisson)

(b) Intensive variable (Gaussian)

Figure A8.9: **Relative scale and transformations of synthetic data**



NRMSE

	Overall	<10%	50%	>90%
Overlay (polygons)	0.52	2.81	0.22	0.07
Overlay (centroids)	0.47	2.27	0.22	0.07
Area Weights (polygons)	0.11	0.16	0.11	0.06
Area Weights (centroids)	0.06	0.09	0.07	0.03
Pop Weights (polygons)	0.11	0.16	0.11	0.06
Pop Weights (centroids)	0.07	0.09	0.07	0.04
TPRS-Forest	0.14	0.14	0.13	0.12
TPRS-Forest (w/ resid)	0.14	0.15	0.14	0.12
TPRS-Area Weights	0.19	0.18	0.17	0.24
Ordinary Kriging	0.13	0.13	0.13	0.12
Universal Kriging	0.21	0.19	0.26	0.23
Rasterization	0.2	0.2	0.2	0.25
Median	0.14	0.16	0.14	0.1

Spearman's correlation

	Overall	<10%	50%	>90%
Overlay (polygons)	0.77	0.66	0.74	0.95
Overlay (centroids)	0.7	0.48	0.68	0.95
Area Weights (polygons)	0.92	0.85	0.91	0.98
Area Weights (centroids)	0.95	0.91	0.95	0.99
Pop Weights (polygons)	0.92	0.85	0.91	0.98
Pop Weights (centroids)	0.95	0.92	0.95	0.99
TPRS-Forest	0.86	0.8	0.84	0.93
TPRS-Forest (w/ resid)	0.85	0.8	0.83	0.92
TPRS-Area Weights	0.55	0.49	0.53	0.59
Ordinary Kriging	0.86	0.81	0.85	0.93
Universal Kriging	0.92	0.92	0.91	0.95
Rasterization	0.51	0.46	0.47	0.58
Median	0.86	0.81	0.85	0.95

OLS estimation bias

	Overall	<10%	50%	>90%
Overlay (polygons)	-1.13	-2.17	-1.11	-0.17
Overlay (centroids)	-1.19	-2.28	-1.19	-0.15
Area Weights (polygons)	-0.61	-0.95	-0.64	-0.25
Area Weights (centroids)	-0.11	-0.21	-0.11	-0.02
Pop Weights (polygons)	-0.62	-0.99	-0.64	-0.25
Pop Weights (centroids)	-0.12	-0.24	-0.1	-0.02
TPRS-Forest	-0.66	-0.73	-0.71	-0.36
TPRS-Forest (w/ resid)	-0.72	-0.76	-0.79	-0.38
TPRS-Area Weights	-0.45	-0.77	-0.41	-0.3
Ordinary Kriging	-0.64	-0.54	-0.71	-0.37
Universal Kriging	-1.19	-1.19	-1.22	-1.07
Rasterization	-1.09	-1.18	-1.27	-0.7
Median	-0.65	-0.86	-0.71	-0.28

(a) Extensive variable (Poisson)

RMSE

	Overall	<10%	50%	>90%
Overlay (polygons)	0.69	0.97	0.7	0.39
Overlay (centroids)	2.34	1.04	1.35	8.77
Area Weights (polygons)	0.64	1	0.63	0.34
Area Weights (centroids)	0.63	1	0.62	0.32
Pop Weights (polygons)	17.37	4.52	79.61	3.88
Pop Weights (centroids)	16.89	4.71	39.45	5.51
TPRS-Forest	0.63	1	0.61	0.34
TPRS-Forest (w/ resid)	0.65	1.01	0.64	0.35
TPRS-Area Weights	0.72	1.04	0.73	0.41
Ordinary Kriging	0.64	1.04	0.63	0.33
Universal Kriging	0.64	1.03	0.62	0.33
Rasterization	0.65	1.01	0.64	0.35
Median	0.65	1.02	0.64	0.35

Spearman's correlation

	Overall	<10%	50%	>90%
Overlay (polygons)	0.92	0.82	0.94	0.96
Overlay (centroids)	0.8	0.87	0.83	0.69
Area Weights (polygons)	0.94	0.83	0.96	0.98
Area Weights (centroids)	0.94	0.84	0.96	0.98
Pop Weights (polygons)	0.91	0.81	0.93	0.94
Pop Weights (centroids)	0.91	0.81	0.93	0.94
TPRS-Forest	0.94	0.84	0.96	0.97
TPRS-Forest (w/ resid)	0.94	0.83	0.95	0.97
TPRS-Area Weights	0.83	0.79	0.83	0.88
Ordinary Kriging	0.94	0.82	0.97	0.97
Universal Kriging	0.94	0.8	0.97	0.97
Rasterization	0.94	0.83	0.95	0.97
Median	0.94	0.82	0.95	0.97

OLS estimation bias

	Overall	<10%	50%	>90%
Overlay (polygons)	-0.09	-0.04	-0.1	-0.1
Overlay (centroids)	-1.36	0.05	-1.46	-2.33
Area Weights (polygons)	0.05	0.18	0.04	-0.05
Area Weights (centroids)	0.06	0.16	0.05	-0.03
Pop Weights (polygons)	-0.6	-0.63	-0.64	-0.34
Pop Weights (centroids)	-0.62	-0.62	-0.7	-0.34
TPRS-Forest	0.09	0.26	0.08	-0.02
TPRS-Forest (w/ resid)	0.05	0.17	0.04	-0.02
TPRS-Area Weights	0.1	0.34	0.06	-0.004
Ordinary Kriging	0.23	1.51	0.08	-0.02
Universal Kriging	0.13	0.54	0.09	-0.01
Rasterization	0.05	0.17	0.04	-0.02
Median	0.05	0.17	0.04	-0.03

(b) Intensive variable (Gaussian)

Figure A8.10: **Transformation quality at different percentiles of relative scale**

A9. Sensitivity analyses with multiple CoS methods

One of our central recommendations calls for researchers to perform sensitivity analyses using alternative CoS methods. While using multiple CoS methods avoids reliance on a single potentially idiosyncratic algorithm, this pluralism is not without pitfalls of its own. Different CoS methods may produce divergent results, and adjudicating between these results is not always straightforward. Direct validation is impossible without ground truth data, as we cannot know which set of estimates is closest to true values. The existence of divergent estimates, with no clear hierarchy among them, creates temptations to "cherry pick." Yet the opposite approach — treating all results, including deviant ones, as equally valid — can be just as misleading.

We propose a middle path, in which researchers report the results of multiple CoS methods, along with a measure of how divergent each set of results is from the others. Where traditional cross-validation is not possible, we recommend using outlier detection tests to establish which CoS methods produce similar results, and which ones are deviant. The deviance in this case reflects not how distant a result is from the "truth," but how distant it is from results obtained through other methods. In the example below, we use Rosner (1983)'s generalized extreme Studentized deviate test to identify outliers; yet the same logic can be extended to other outlier tests (e.g. χ^2, Dixon, Grubbs).

Let κ denote the number of CoS methods under consideration, indexed by $k = \{1, \ldots, \kappa\}$. Each of these methods yields an estimate, $\hat{x}_k$, which can represent the sample mean of a spatially-transformed variable, a regression coefficient estimate from transformed data, or any other quantity of interest. We will assume that at least $\kappa - \lfloor \kappa/2 \rfloor$ of these estimates come from the same Gaussian distribution, while up to $\lfloor \kappa/2 \rfloor$ estimates may come from a different distribution (where $\lfloor \kappa/2 \rfloor$ is the largest integer less than or equal to $\kappa/2$).

Let $\hat{x}^{(j)}$ be the j-th most extreme value of $\hat{x}$, such that $\hat{x}^{(1)}$ is the value with the farthest distance from the sample mean $\bar{x}$, $\hat{x}^{(2)}$ is the second-farthest from the mean, and so on. Let $\hat{\mathbf{x}}^{(j)}$ be the set of estimates at least as extreme as $\hat{x}^{(j)}$. Let $\bar{x}^{(j)}$ and $s^{(j)}$ be the mean and standard deviation, respectively, of the $\kappa - j$ estimates that remain after removing the j most extreme values. For each $j = 0, \ldots, \lfloor \kappa/2 \rfloor - 1$, we will compare the test statistic

$$R_{j+1} = \frac{|\hat{x}^{(j)} - \bar{x}^{(j)}|}{s^{(j)}}$$

against its corresponding critical value

$$\lambda_{j+1} = \frac{t_{p,\kappa-j-2}(\kappa - j - 1)}{\sqrt{(\kappa - j - 2 + t_{p,\kappa-j-2})(\kappa - j)}}$$

where $t_{p,\kappa-j-2}$ is the p-th quantile of Student's t-distribution with $\kappa - j - 2$ degrees of freedom, $p = 1 - \frac{\alpha/2}{\kappa-j}$, and Type I error level $\alpha = .05$. If $R_j > \lambda_j$ then the j most extreme values are outliers. Measure k is an outlier if it is among these j values, $\hat{x}_k \in \hat{\mathbf{x}}^{(j)}$.

Let ω_k denote the proportion of tests in which measure k is flagged as an outlier:

$$\omega_k = \frac{1}{\lfloor K/2 \rfloor} \sum_{j}^{\lfloor K/2 \rfloor} 1\left(R_j > \lambda_j\right) \cdot 1\left(\hat{x}_k \in \hat{\mathbf{x}}^{(j)}\right)$$

Table A9.8 reports ω_k values for the CoS methods used in our Monte Carlo simulations. The quantity of interest here is the sample mean of the transformed value of variable X, as generated by each algorithm. The table suggests that simple overlay methods produce the most divergent results of all methods considered, with one algorithm (overlay-centroids) being flagged as an outlier in 86 percent of all tests. Whether this method should therefore be excluded from analysis is at the discretion of the researcher, although the high value of ω_k certainly suggests that relying exclusively on simple overlays could be problematic.

Method	ω_k
Area Weights (polygons)	0.05
TPRS-Forest	0.05
Area Weights (centroids)	0.06
Ordinary Kriging	0.06
TPRS-Forest (w/ resid)	0.06
Rasterization	0.06
Universal Kriging	0.06
TPRS-Area Weights	0.06
Pop Weights (polygons)	0.07
Pop Weights (centroids)	0.07
Overlay (centroids)	0.78
Overlay (polygons)	0.78

Table A9.8: Rosner's outlier tests ($\hat{X}$ estimates in Monte Carlo simulations).

Several caveats are in order. Most emphatically, no set of results should be excluded from analysis solely on the basis of an outlier detection test. In the absence of ground truth data, for all we know, the outlier result may be the only "correct" one, while all the others are truly "wrong." By the same token, just because a result is not an outlier does not mean it is necessarily "correct." For instance, rasterization has the lowest ω_k in Table A9.8, but its performance across most cross-validation exercises was middling at best.

An outlier detection test is not a substitute for cross-validation. Outlier tests can tell us how close a series of results are to each other, but not how close they are to the (unobserved) truth. Our recommendation, therefore, is that researchers use ω_k values in the spirit of transparency and discovery, rather than as a discrete numerical threshold or screening device. At a minimum, we recommend that researchers report ω_k values alongside their main results, to place their findings in context. If the output of a CoS method is frequently flagged as an outlier, further investigation may be warranted into why these results are so deviant. A researcher can then look "under the hood" of the offending algorithm and see whether a well-motivated justification exists for keeping or removing the method from the ensemble. Similarly, researchers whose analysis relies on one primary CoS method can use ω_k to reassure readers that their results are not anomalous.

A10. R package code examples

The SUNGEO R package provides tools to calculate nesting metrics and implement some of the CoS methods described here. The package can be installed and loaded as follows:

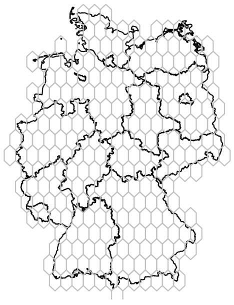
```
# Install package
> install.packages("SUNGEO", dependencies = TRUE)

# Load package:
> library(SUNGEO)
```

Let's begin by loading some spatial data for illustrative purposes: polygons representing German legislative districts and hexagonal grid cells:

```
# Load data
> data(clea_deu2009)
> data(hex_05_deu)

# Preview
> plot(clea_deu2009["geometry"])
> plot(hex_05_deu["geometry"],add=TRUE,border="grey")
```



The grid cells appear generally smaller than legislative districts, and not nested. We can use the `nesting()` function to calculate scale and nesting metrics for the two sets of polygons. Let's calculate nesting metrics for a change of support from district to grid:

```
# Calculate all nesting metrics for a district-to-grid CoS
> nest_1 <- SUNGEO::nesting(
+               poly_from = clea_deu2009,
+               poly_to = hex_05_deu
+               )
> str(nest_1)
List of 12
 $ rs        : num 0.0252
 $ rn        : num 0.16
 $ rs_sym    : num -0.95
 $ rn_sym    : num -0.511
 $ rs_nn     : num 0.0252
 $ rn_nn     : num 0.16
 $ p_intact : num 0
 $ full_nest: num 0
 $ ro        : num -0.175
 $ gmi       : num 0.84
```

In this scenario, $RS = 0.025, RN = 0.16$, indicating disaggregation across non-nested units. Now let's check the opposite direction:

```
# Calculate all nesting metrics for a grid-to-district CoS
> nest_2 <- SUNGEO::nesting(
+               poly_from = hex_05_deu,
+               poly_to = clea_deu2009
+               )
> str(nest_2)
List of 12
 $ rs        : num 0.976
 $ rn        : num 0.67
 $ rs_sym    : num 0.953
 $ rn_sym    : num 0.511
 $ rs_nn     : num 0.97
 $ rn_nn     : num 0.528
 $ p_intact : num 0.588
 $ full_nest: num 0.302
 $ ro       : num 0.175
 $ gmi      : num 0.33
```

Here, $RS = 0.98, RN = 0.67$, indicating aggregation and more (but not perfect) nesting. To save computational time, we can modify the metrix option to extract specific metrics (e.g. just RN) rather the full battery.

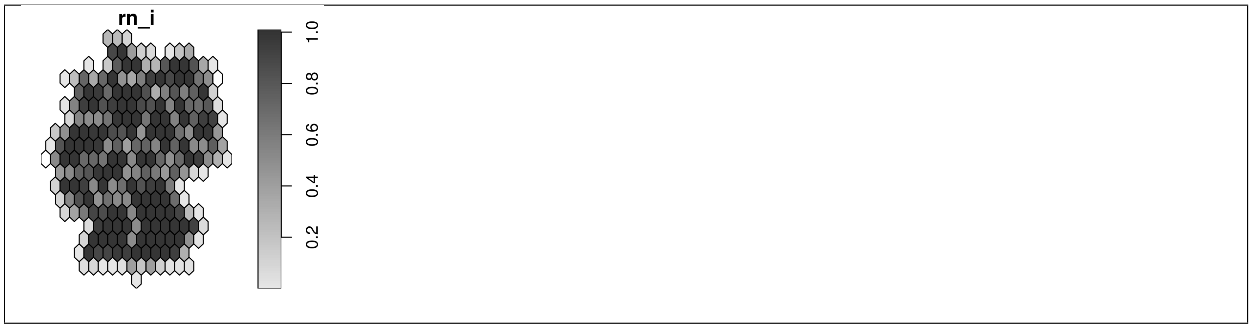
```
# Calculate just RN
> nest_3 <- SUNGEO::nesting(
+               poly_from = hex_05_deu,
+               poly_to = clea_deu2009,
+               metrix = "rn"
+               )
> nest_3
 $rn
 [1] 0.6702956
```

To identify which source units remain intact and which are split (among other quantities), we can use the option by_unit=TRUE to obtain the unit-level components.

```
# Disaggregate nesting metrics by unit (where feasible)
> nest_4 <- SUNGEO::nesting(
+               poly_from = hex_05_deu,
+               poly_to = clea_deu2009,
+               by_unit = TRUE
+               )
nest_4$by_unit
> nest_4$by_unit
     index rs       rn    rs_alt      rn_alt rs_nn    rn_nn p_intact
  1:     1  1 3.19e-06 0.9740827 0.00178652     1 3.19e-06        1
  2:     2  1 4.91e-02 0.9493340 0.22154270     1 4.91e-02        1
  3:     3  1 8.18e-02 0.9493340 0.28602730     1 8.18e-02        1
  4:     4  1 4.88e-03 0.9493340 0.06983717     1 4.87e-03        1
  5:     5  1 4.15e-03 0.9493340 0.06443348     1 4.15e-03        1
 ---
```

```
253:   253  1 1.93e-01 0.9322391 0.43939827     1 1.93e-01        1
254:   254  1 3.43e-01 0.9322391 0.58569486     1 3.43e-01        1
255:   255  1 2.63e-01 0.9019284 0.51319413     1 2.63e-01        1
256:   256  1 2.43e-01 0.9019284 0.49270627     1 2.43e-01        1
257:   257  1 1.01e-01 0.9019284 0.31718422     1 1.01e-01        1
     full_nest       gmi
  1:         0 0.9999968
  2:         0 0.9509188
  3:         0 0.9181884
  4:         0 0.9951228
  5:         0 0.9958483
 ---
253:         0 0.8069292
254:         0 0.6569615
255:         0 0.7366318
256:         0 0.7572405
257:         0 0.8993942

# Visualize rn_i on a map
> hex_05_deu$rn_i <- nest_4$by_unit[,rn]
> plot(hex_05_deu["rn_i"])
```



The SUNGEO package also has routines for batch geocoding of addresses (`geocode_osm()`, `geocode_osm_batch()`), overlays (`point2poly_simp()`), interpolation (`poly2poly_ap()`), and other CoS methods. Please see the package help files for additional information.

References

Chiles, Jean-Paul and Pierre Delfiner. 2009. *Geostatistics: modeling spatial uncertainty*. Vol. 497 2nd ed. John Wiley & Sons.

Cressie, Noel. 1993. *Statistics for spatial data*. John Wiley & Sons.

Duchon, J. 1977. *Construction Theory of Functions of Several Variables*. Berlin: Springer chapter Splines minimizing rotation-invariant semi-norms in Solobev spaces.

Goovaerts, Pierre. 1997. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand.

Gotway, Carol A and Linda J Young. 2007. "A geostatistical approach to linking geographically aggregated data from different sources." *Journal of Computational and Graphical Statistics* 16(1):115–135.

Kollman, Ken, Allen Hicken, Daniele Caramani, David Backer and David Lublin. 2022. *Constituency-Level Elections Archive*. Ann Arbor, MI: Center for Political Studies, University of Michigan.

Kollman, Ken, Allen Hicken, Daniele Caramani, David Backer, David Lublin, Joel Selway and Fabricio Vasselai. 2017. *GeoReferenced Electoral Districts Datasets (Beta)*. Ann Arbor, MI: Center for Political Studies, University of Michigan.

Rosner, Bernard. 1983. "Percentage points for a generalized ESD many-outlier procedure." *Technometrics* 25(2):165–172.

Wood, Simon N. 2003. "Thin plate regression splines." *Journal of the Royal Statistical Society, Series B* 65(1):95–114.

Young, Linda J, Carol A Gotway, Greg Kearney and Chris DuClos. 2009. "Assessing uncertainty in support-adjusted spatial misalignment problems." *Communications in StatisticsTheory and Methods* 38(16-17):3249–3264.

Zhukov, Yuri M, Christian Davenport and Nadiya Kostyuk. 2019. "Introducing xSub: A New Portal for Cross-National Data on Sub-National Violence." *Journal of Peace Research* 56(4):604–614.