Segment Pivotality Index: Digital Appendix*

Pawas Pratikshit

*This paper was developed over the past year (2025), drawing motivation from recent developments in India and building on the quantitative training I received at Sciences Po. I have also developed an open-source R package to enable replication and application of the index in other hierarchical contexts; it can be accessed at https://github.com/PawasPratikshit/SPIr.

1 Legislative Assemblies and Eligibility for the Segment Pivotality Test

Table 1: Indian state and union territory legislative assemblies by size

State / UT	eats in assembly	Ruling party
	Eligible: 59 or more seats	
Andhra Pradesh	175	Telugu Desam Party
Arunachal Pradesh	60	Bharatiya Janata Party
Assam	126	Bharatiya Janata Party
Bihar	243	Janata Dal (United)
Chhattisgarh	90	Bharatiya Janata Party
Delhi	70	Bharatiya Janata Party
Gujarat	182	Bharatiya Janata Party
Haryana	90	Bharatiya Janata Party
Himachal Pradesh	68	Indian National Congress
Jammu and Kashmir	95	Jammu & Kashmir National Conference
Jharkhand	81	Jharkhand Mukti Morcha
Karnataka	224	Indian National Congress
Kerala	140	Communist Party of India
Madhya Pradesh	230	Bharatiya Janata Party
Maharashtra	288	Bharatiya Janata Party
Manipur	60	President's rule
Meghalaya	60	National People's Party
Nagaland	60	Naga People's Front
Odisha	147	Bharatiya Janata Party
Punjab	117	Aam Aadmi Party
Rajasthan	200	Bharatiya Janata Party
Tamil Nadu	234	Dravida Munnetra Kazhagam
Telangana	119	Indian National Congress
Tripura	60	Bharatiya Janata Party
Uttar Pradesh	403	Bharatiya Janata Party
Uttarakhand	70	Bharatiya Janata Party
West Bengal	294	All India Trinamool Congress
	Not eligible: fewer than 59 seats	
Goa	40	Bharatiya Janata Party
Mizoram	40	Zoram People's Movement
Puducherry	33	All India N.R. Congress
Sikkim	32	Sikkim Krantikari Morcha

2 Benford Second-Digit Test Results

Table 2: Benford Second-Digit (2BL) Test Results for AC-Level Largest-Coalition Votes (2024)

State	Size	N	χ^2	df	p-value	Max. Deviation
Madhya Pradesh	Large	230	34.02	9	9.07e-05	0.0977
Maharashtra	Large	288	69.43	9	1.99e-11	0.1370
Uttar Pradesh	Large	403	94.62	9	1.94e-16	0.1190
Gujarat	Medium	175	25.06	9	2.85e-03	0.0803
Odisha	Medium	147	8.47	9	4.88e-01	0.0544
Rajasthan	Medium	200	26.82	9	1.50e-03	0.1000
Arunachal Pradesh	Small	60	5.75	9	7.65e-01	0.0596
Himachal Pradesh	Small	68	8.91	9	4.46e-01	0.0749
Manipur	Small	62	9.37	9	4.04e-01	0.1120

3 Simulation Design for Evaluating the 2BL Test in Uttar Pradesh

This appendix provides the full details of the simulation that is used to evaluate how the second digit Benford test behaves when applied to Assembly Constituency (AC) level data from Uttar Pradesh. The goal of the simulation is to check whether deviations from Benford can arise even in a clean election environment that mirrors the empirical structure of the actual data. In the main paper, the alliance that received the highest number of parliamentary constituencies across the state was the INDIA alliance, and the simulation is based on its AC level performance.

3.1 Setup and notation

Uttar Pradesh has 403 ACs. Index these by $i = 1, \ldots, 403$. For each AC, let

$$E_i > 0$$

denote the number of registered electors. Let V_i denote the observed votes received by the INDIA alliance in AC i. I define the observed rate as

$$\hat{t}_i = \frac{V_i}{E_i}, \qquad 0 < \hat{t}_i < 1.$$

This rate reflects both turnout and coalition strength and is bounded between zero and one. In the simulation, the electorate sizes are kept fixed and only the rate is resampled.

3.2 Modeling the rate on the logit scale

To model the variation in $\hat{t}_i$, I use the logit transformation

$$\operatorname{logit}(x) = \log\left(\frac{x}{1-x}\right),$$

with inverse

$$\operatorname{logit}^{-1}(z) = \frac{e^z}{1+e^z}.$$

For each AC, I compute

$$z_i = \operatorname{logit}(\hat{t}_i),$$

after replacing extremely small or extremely large values of $\hat{t}_i$ with values slightly inside the interval to avoid infinite logits. I then estimate the mean and variance of these logit values:

$$\hat{\mu} = \frac{1}{403}\sum_{i=1}^{403} z_i, \qquad \hat{\sigma}^2 = \frac{1}{402}\sum_{i=1}^{403}(z_i - \hat{\mu})^2.$$

The simulated rate is generated from a logit normal model:

$$z_i^* \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2), \qquad t_i^* = \operatorname{logit}^{-1}(z_i^*).$$

3.3 Generating simulated vote totals

Given the simulated rate t_i^*, the corresponding clean simulated vote count is

$$V_i^{\text{sim}} = \operatorname{round}(E_i \cdot t_i^*),$$

which ensures

$$0 \le V_i^{\text{sim}} \le E_i.$$

This matches the scale of the real data while generating values that contain no manipulation of any kind.

3.4 Extracting second digits

For any positive integer x with at least two digits, write it as

$$x = \sum_{k=0}^{K} d_k 10^k,$$

where $d_K \neq 0$. The second digit from the left is d_{K-1}, which I denote by $\mathrm{sd}(x)$. If x has only one digit, the second digit is undefined and the observation is excluded.

For each simulated vote total,

$$D_i^{\mathrm{sim}} = \begin{cases} \mathrm{sd}(V_i^{\mathrm{sim}}) & \text{if } V_i^{\mathrm{sim}} \text{ has at least two digits,} \\ \text{missing} & \text{otherwise.} \end{cases}$$

3.5 Chi-squared test

For any simulated dataset, let

$$O_d = \sum_{i=1}^{403} \mathbf{1}\{D_i^{\mathrm{sim}} = d\}$$

be the observed frequency of second digit d, and let

$$N = \sum_{d=0}^{9} O_d.$$

The expected counts under Benford are

$$E_d = N p_d.$$

The chi squared test statistic used in the 2BL test is

$$X^2 = \sum_{d=0}^{9} \frac{(O_d - E_d)^2}{E_d}.$$

The p value is

$$p = 1 - F_{\chi^2_9}(X^2),$$

where $F_{\chi^2_9}$ denotes the cumulative distribution function of the chi squared distribution with nine degrees of freedom.

3.6 Monte Carlo procedure

To evaluate the false positive rate of the test under clean elections, I repeat the following steps 500 times:

1. Draw z_i^* from $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$.
2. Transform to t_i^* using the inverse logit.
3. Generate simulated vote totals V_i^{sim}.
4. Extract second digits and compute the chi squared p value.

3.7 Reference to main text

All numerical results of this simulation, including the distribution of p values and the false positive rate, are reported in the main paper. They show that the 2BL test rejects at extremely high rates even when the simulated elections contain no manipulation. This supports the argument that the deviations from Benford observed in the Uttar Pradesh data arise from the structure of meso level vote counts rather than from fraud.

4 Range of the Segment Pivotality Index

This appendix derives the range of values that the Segment Pivotality Index (SPI) can take and clarifies the interpretation of its lower and upper bounds.

4.1 Definition and Notation

Consider a Parliamentary Constituency (PC) composed of n Assembly Constituencies (ACs), indexed by $i = 1, \ldots, n$. Let W_i denote the votes obtained by the PC winner in AC i, and let R_i denote the votes obtained by the PC runner-up in AC i. Define the AC-level margin for the PC winner as

$$m_i = W_i - R_i.$$

Let the total PC margin be

$$M = \sum_{j=1}^{n} m_j = \sum_{j=1}^{n} W_j - \sum_{j=1}^{n} R_j.$$

By definition of the winner, $M > 0$.

The contribution of all ACs other than i to the PC margin is

$$M_{-i} = \left| \sum_{j \neq i} m_j \right| = \left| M - m_i \right|.$$

The Segment Pivotality Index for AC i is defined as

$$\mathrm{SPI}_i = \frac{m_i}{M_{-i}} = \frac{m_i}{|M - m_i|},$$

whenever $M_{-i} > 0$. The case $M_{-i} = 0$ is treated separately below.

4.2 Reparameterisation

For ease of exposition, write

$$x = m_i, \qquad y = \sum_{j \neq i} m_j = M - x.$$

Then $M = x + y$ and $M > 0$ implies $x + y > 0$. The index can be written as

$$\mathrm{SPI}_i = \frac{x}{|y|},$$

with the constraint $x + y > 0$. The denominator $|y|$ is equal to M_{-i}, and $M_{-i} = 0$ occurs if and only if $y = 0$.

4.3 Lower Bound

We show that, under the condition $M > 0$, the SPI is bounded below by -1, in the sense that $\mathrm{SPI}_i \geq -1$ for all admissible configurations, and that values strictly less than -1 are not attainable.

There are two cases to consider.

Case 1: $y > 0$. In this case $|y| = y$, so

$$\mathrm{SPI}_i = \frac{x}{y}.$$

Using $x = M - y$, we can rewrite

$$\mathrm{SPI}_i = \frac{M - y}{y} = \frac{M}{y} - 1.$$

Since $M > 0$ and $y > 0$, it follows that $\frac{M}{y} > 0$. Hence

$$\mathrm{SPI}_i = \frac{M}{y} - 1 > -1.$$

As $y \to \infty$, we have $\frac{M}{y} \to 0$, so $\mathrm{SPI}_i \to -1$ from above. Therefore, in the case $y > 0$, the SPI satisfies $\mathrm{SPI}_i \in (-1, \infty)$ and approaches -1 as y becomes large relative to M.

Case 2: $y < 0$. In this case $|y| = -y$, so

$$\mathrm{SPI}_i = \frac{x}{-y} = -\frac{x}{y}.$$

Using $x = M - y$ and the fact that $y < 0$, we have

$$x = M - y > M > 0,$$

so $x > 0$ and $y < 0$ imply $-\frac{x}{y} > 0$. Thus

$$\mathrm{SPI}_i > 0 > -1.$$

Hence, whenever $y < 0$, the SPI is strictly positive and cannot fall below -1.

Case 3: $y = 0$. If $y = 0$, then $M_{-i} = |y| = 0$, and the SPI is not defined as a finite real number. This corresponds to the limiting case in which

$$M = x + y = x$$

and the entire PC margin is generated by AC i alone. In this situation the AC is perfectly pivotal: removing it eliminates the PC-level margin. In the limit as $y \to 0$ with $M > 0$ fixed, the expression $\frac{x}{|y|}$ grows without bound. This justifies treating the upper range of the SPI as unbounded above.

Combining the three cases, we obtain that for all configurations with $M > 0$ and $M_{-i} > 0$, the SPI satisfies

$$\mathrm{SPI}_i > -1.$$

Values arbitrarily close to -1 can be approached in the limit when $y > 0$ becomes large. Exact equality $\mathrm{SPI}_i = -1$ would require $M = 0$, which contradicts the assumption that the PC has a strictly positive winner margin. It is therefore natural to describe the theoretical lower bound as -1, with the understanding that, for any fixed positive M, the set of attainable values lies in $(-1, \infty)$ but can approach -1 arbitrarily closely across elections.

4.4 Upper Behaviour

To characterise the upper range of the SPI, consider again

$$\mathrm{SPI}_i = \frac{x}{|y|} = \frac{m_i}{M_{-i}}.$$

There is no finite upper bound on this ratio. To see this, fix a positive PC margin M and let y approach zero while keeping $M = x + y$ constant. Then $x = M - y$ approaches M and remains positive, whereas $|y|$ tends to zero. The ratio $\frac{x}{|y|}$ diverges to $+\infty$. In substantive terms, this corresponds to cases where the margin contributed by AC i nearly equals the PC-wide margin, so that the remaining ACs contribute almost nothing net. Such ACs are extremely pivotal: a small change in their local outcome would reverse the PC result.

More generally, even if y is not arbitrarily small, the numerator x can be large in magnitude relative to $|y|$ whenever one AC delivers a very large winner–runner-up margin and the remaining ACs collectively deliver only a modest net margin. The SPI therefore has no finite upper bound and can take arbitrarily large positive values.

4.5 Summary of the Range

The analysis above implies that, under the condition that the PC has a strictly positive winner margin, the SPI satisfies

$$\mathrm{SPI}_i \in (-1, \infty),$$

with -1 as a theoretical lower bound approached in the limit and no finite upper bound. In practice it is convenient to describe the index as taking values on the interval $[-1, \infty)$, with the understanding that exact equality at -1 would correspond to a knife-edge case in which the overall PC margin is zero. Values close to zero indicate that the AC contributes little to the overall PC margin. Negative values arise when the PC winner performs worse in that AC than the runner-up, so that the AC pulls against the PC-level outcome. Values greater than one occur when the margin generated by AC i alone exceeds the net margin generated by all other ACs combined, which identifies assembly segments whose contribution to the parliamentary result is disproportionately large relative to the rest of the constituency.

5 The `SPIr` R Package

5.1 Overview and Motivation

The Segment Pivotality Index (SPI) that I introduce in the paper helps identify how much each Assembly Constituency (AC) contributes to the final result in a Parliamentary Constituency (PC). As discussed in the main text, SPI is shaped by the observed vote totals, which means that manipulation can influence its value as well. It is therefore not a treatment assignment rule and should not be used for causal claims. SPI is better understood as a tool for identifying patterns that may deserve attention when investigating possible irregularities. Sometimes these patterns may simply reflect features of party competition. For example, a party may perform extremely well in certain areas that heavily influence the final margin. For this reason, the results shown in the analysis should be interpreted together with other diagnostic checks.

To make the measure easy to compute in other settings where elections have a hierarchical structure, I created an open source R package called `SPIr`. The package calculates SPI for any dataset where lower level units roll up into a higher level unit, such as ACs within PCs, precincts within districts, or municipalities within provinces. It also includes functions for appending SPI values back into the dataset and for summarizing and visualizing the resulting distribution.

The package is publicly available at:

https://github.com/PawasPratikshit/SPIr

5.2 Package Structure and Main Functions

The `SPIr` package is built using standard R package infrastructure. It uses `usethis`, `devtools`, and `roxygen2` for documentation and structure. The package currently contains four main functions:

- `compute_spi()` which calculates SPI values for each AC within each PC,

- `add_spi()` which adds SPI values directly to the original data,
- `summary_spi()` which provides simple descriptive summaries, and
- `plot_spi_distribution()` which produces a distribution plot for exploration.

All functions follow tidy data conventions. Column names can be passed directly without quotes through nonstandard evaluation. Users can access the documentation through:

```
help(package = "SPIr")
?compute_spi
```

5.3 Installation

Users can install the package directly from GitHub. The recommended method is:

```
# install.packages("remotes")
remotes::install_github("PawasPratikshit/SPIr")
library(SPIr)
```

5.4 Minimal Vignette: Basic Workflow

Suppose a researcher has election data that records candidate vote totals in each AC within each PC. A simple illustration is:

```
library(SPIr)
library(dplyr)

df <- tibble(
    pc    = c(1,1,1,1),
    ac    = c(1,1,2,2),
    cand  = c("A","B","A","B"),
```

```
    votes = c(60,40,55,45)
)
```

To calculate SPI values:

```
spi_df <- compute_spi(
    df,
    pc_id     = pc,
    ac_id     = ac,
    candidate = cand,
    votes     = votes
)
```

To merge the SPI values into the main dataset:

```
df_with_spi <- add_spi(
    df,
    pc_id     = pc,
    ac_id     = ac,
    candidate = cand,
    votes     = votes
)
```

To inspect the distribution:

```
summary_spi(df_with_spi, spi)
```

To plot it:

```
plot_spi_distribution(df_with_spi, spi)
```

5.5 F.5 Reproducibility and Future Extensions

The `SPIr` package is meant to be used in other countries as well. Any dataset with a basic hierarchical structure can be analyzed with SPI. The only requirements are:

1. the higher level unit is composed of multiple lower level units,
2. votes are available at the lower unit level, and
3. the higher level winner is determined by summing votes across lower units.

The code is fully open source, so researchers can modify or extend the functions as needed. This may include adding new visualizations, incorporating additional diagnostics, or adapting the SPI logic for other applications that involve aggregation.

The GitHub repository will include future updates and examples.

Link: `https://github.com/PawasPratikshit/SPIr`