

# Package ‘minorparties’

May 8, 2026

**Type** Package

**Title** Quantitatively Analyze Minor Political Parties

**Version** 1.0.0

**Description** Tools for calculating I-Scores, a simple way to measure how successful minor political parties are at influencing the major parties in their environment. I-Scores are designed to be a more comprehensive measurement of minor party success than vote share and legislative seats won, the current standard measurements, which do not reflect the strategies that most minor parties employ. The procedure leverages the Manifesto Project's NLP model to identify the issue areas that sentences discuss, see Burst et al. (2024) <[doi:10.25522/manifesto.manifestoberta.56topics.context.2024.1.1](https://doi.org/10.25522/manifesto.manifestoberta.56topics.context.2024.1.1)>, and the Wordfish algorithm to estimate the relative positions that platforms take on those issue areas, see Slapin and Proksch (2008) <[doi:10.1111/j.1540-5907.2008.00338.x](https://doi.org/10.1111/j.1540-5907.2008.00338.x)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**Imports** dplyr, purrr, quanteda, quanteda.textmodels, reticulate, rlang, spacyr, stats, stringi, stringr, tibble, tidyr

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1.0)

**URL** <https://gerckentheodore.github.io/minorparties/>

**NeedsCompilation** no

**Author** Theodore Gercken [aut, cre, cph]

**Maintainer** Theodore Gercken <[tgercken@hamilton.edu](mailto:tgercken@hamilton.edu)>

**Repository** CRAN

**Date/Publication** 2025-10-29 20:10:12 UTC

## Contents

calculate_iscores . . . . .	2
install_python . . . . .	4
process_platform_emphasis . . . . .	5
process_platform_position . . . . .	6
sample_data . . . . .	7
sample_data_results . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

calculate_iscores	<i>Calculate minor parties' I-Scores</i>
-------------------	--

---

## Description

calculate\_iscores() takes a tibble of platforms that have already been processed with process\_platform\_emphasis() and process\_platform\_position() and calculates I-Scores for each minor party in the tibble. I-Scores represent the extent to which a minor party influenced the major parties in its political environment on its top issues. Ie-Scores reflect changes in the emphasis major parties place on a minor party's core issues (Ie-Scores measure the change in percentage points; interpreted Ie-Scores measure the change in percent), while Ip-Scores track shifts in the position of major parties on those issues.

## Usage

```
calculate_iscores(
  tibble,
  p_threshold = 0.05,
  core_threshold = 0.05,
  exclude_nonconvergence = TRUE,
  adjust_p_values = TRUE,
  confidence_intervals = FALSE,
  confidence_n = 1000,
  calculation_tables = FALSE
)
```

## Arguments

tibble	Tibble. One row per platform, containing, (this function is designed to work with the output of process_platform_position()): <ul style="list-style-type: none"> <li>• party: Character column. The party's name (this column must be unique for each platform).</li> <li>• sentence_emphasis_scores: List column. A list per sentence in the platform (in order), containing:           <ul style="list-style-type: none"> <li>– sentence: Character. The sentence.</li> <li>– scores: Tibble. The sentence's emphasis score on each issue-area, containing:</li> </ul> </li> </ul>
--------	--

- \* issue: Character column. The issue-area name.
  - \* score: Numeric column. The sentence's score for that issue-area (summing to 1).
  - overall\_emphasis\_scores: List column. A tibble with the platform's overall emphasis scores, containing:
    - \* issue: Character column. The issue-area name.
    - \* score: Numeric column. The platform's score for that issue-area
  - position\_scores: List column. A tibble, containing:
    - \* issue: Character column. The issue-area name.
    - \* position\_score: Numeric column. The platform's position score on the issue-area (NA if the platform did not have enough material about the issue-area to generate a score).
    - \* se: Numeric column. The standard error of the position score.
    - \* convergence: Logical column. Whether the Wordfish model converged (if the estimation algorithm reached a stable set of position scores without divergence).
  - minor\_party: Logical column. Whether the party is a minor party.
  - major\_party\_platforms: List column. Only required for minor parties. A list containing a list for each major party, each of which contains:
    - \* before: Character. The name (as listed in this tibble's party column) of the major party's platform that precedes the minor party.
    - \* after: Character. The name (as listed in this tibble's party column) of the major party's platform that follows the minor party.
    - \* weight: Numeric. The weight assigned to the major party.
- p\_threshold      Numeric. The maximum p-value at which a relationship is considered significant. Defaults to 0.05.
- core\_threshold    Numeric. The minimum emphasis score a minor party must have for an issue-area to be considered a core issue. Defaults to 0.05.
- exclude\_nonconvergence    Logical. Whether to treat issue-areas where the Wordfish model did not converge as if no score had been found when calculating Ip-Scores. Defaults to TRUE.
- adjust\_p\_values    Logical. Whether to adjust p-values to account for the large number of comparisons. Defaults to TRUE.
- confidence\_intervals    Logical. Whether to calculate confidence intervals for the I-Scores. Defaults to FALSE.
- confidence\_n      Numeric. If confidence\_intervals is TRUE, the number of bootstrap samples to take in calculating confidence intervals. Defaults to 1000.
- calculation\_tables    Logical. Whether to include the tables used to calculate I-Scores in the output. Defaults to FALSE.

**Value**

Tibble. The input tibble, filtered to only include minor parties, with the following columns:

- party: Character column. The party's name.
- scores: List column. A list, containing:
  - ie\_score: Numeric. The party's Ie-Score.
  - ie\_score\_interpreted: Numeric. The party's interpreted Ie-Score
  - ip\_score: Numeric. The party's Ip-Score.
- calculation\_tables: List column (present if calculation\_tables argument is TRUE). A list, containing:
  - ie\_score\_table: Tibble. The table used to calculate the party's Ie-Score.
  - ip\_score\_table: Tibble. The table used to calculate the party's Ip-Score.
- confidence\_intervals: List column (present if confidence\_intervals argument is TRUE). A tibble, containing:
  - side: Character column. "lower" or "upper", indicating the bounds of the confidence interval.
  - ie\_score: Numeric column. The bound of the confidence interval for the party's Ie-Score.
  - ie\_score\_interpreted: Numeric column. The bound of the confidence interval for the party's interpreted Ie-Score.
  - ip\_score: Numeric column. The bound of the confidence interval for the party's Ip-Score.

**Examples**

```
tibble <- minorparties::sample_data |>
  minorparties::process_platform_emphasis() |>
  minorparties::process_platform_position()
processed_tibble <- calculate_iscores(tibble)
```

---

install\_python

*Set up the python tools required for calculating I-Scores*

---

**Description**

install\_python() uses reticulate to create or activate a Python virtual environment and install the required packages for running spacyr (sentence tokenization) and huggingfaceR (issue-area classification via ManifestoBERTA).

**Usage**

```
install_python(env_name = "iscores", manifestoberta_model_id = "2024-1-1")
```

**Arguments**

env_name	Character. Name of the Python virtual environment to set up. Defaults to "is-cores".
manifestoberta_model_id	Character. Version of the ManifestoBERTA model to load (must match a valid model). Defaults to "2024-1-1".

**Value**

Invisibly returns TRUE.

---

process\_platform\_emphasis

*Calculate platforms' issue-area emphasis scores*

---

**Description**

process\_platform\_emphasis() takes a tibble of platforms, splits each platform into sentences, and calculates issue-area emphasis scores for each sentence and for the platform as a whole using the ManifestoBERTA model. These issue-area emphasis scores, respectively, represent the probability that each sentence is discussing each issue-area and the proportion of the platform that is devoted to each issue-area.

**Usage**

```
process_platform_emphasis(tibble, cleaning = TRUE)
```

**Arguments**

tibble	Tibble. One row per platform, containing, at minimum: <ul style="list-style-type: none"> <li>• text: Character column. The full text of each platform.</li> </ul>
cleaning	Logical. Whether to apply basic text cleaning before processing each platform. Defaults to TRUE.

**Value**

Tibble. The input tibble with two additional list columns (if a platform cannot be processed due to a lack of text, the function will return an empty list for that platform):

- sentence\_emphasis\_scores: List column. A list per sentence in the platform (in order), containing:
  - sentence: Character. The sentence.
  - scores: Tibble. The sentence's emphasis score on each issue-area, containing:
    - \* issue: Character column. The issue-area name.
    - \* score: Numeric column. The sentence's score for that issue-area (summing to 1).

- overall\_emphasis\_scores: List column. A tibble with the platform’s overall emphasis scores, containing:
  - \* issue: Character column. The issue-area name.
  - \* score: Numeric column. The platform’s score for that issue-area.

### Examples

```
tibble <- minorparties::sample_data
processed_tibble <- process_platform_emphasis(tibble)
```

---

```
process_platform_position
  Calculate platforms' issue-area position scores
```

---

### Description

process\_platform\_position() takes a tibble of platforms that have already been processed with process\_platform\_emphasis() and calculates issue-area position scores for each platform using the Wordfish model. These position scores represent each platform’s relative position on each issue-area compared to the others in the tibble (the positive end of the scale is arbitrarily defined).

### Usage

```
process_platform_position(tibble, inclusion_threshold = 0.2)
```

### Arguments

tibble	<p>Tibble. One row per platform, containing, at minimum (this function is designed to work with the output of process_platform_emphasis()):</p> <ul style="list-style-type: none"> <li>• party: Character column. The party’s name (this column must be unique for each platform).</li> <li>• sentence_emphasis_scores: List column. A list per sentence in the platform (in order), containing:           <ul style="list-style-type: none"> <li>– sentence: Character. The sentence.</li> <li>– scores: Tibble. The sentence’s emphasis score on each issue-area, containing:               <ul style="list-style-type: none"> <li>* issue: Character column. The issue-area name.</li> <li>* score: Numeric column. The sentence’s score for that issue-area (summing to 1).</li> </ul> </li> </ul> </li> </ul>
inclusion_threshold	<p>Numeric. The minimum probability a sentence must have of discussing an issue-area to be included in that issue-area’s Wordfish model. Defaults to 0.2.</p>

**Value**

Tibble. The input tibble with an additional list column:

- **position\_scores**: List column. A tibble, containing:
  - **issue**: Character column. The issue-area name.
  - **position\_score**: Numeric column. The platform’s position score on the issue-area (NA if the platform did not have enough material about the issue-area to generate a score).
  - **se**: Numeric column. The standard error of the position score.
  - **convergence**: Logical column. Whether the Wordfish model converged (if the estimation algorithm reached a stable set of position scores without divergence).

**Examples**

```
tibble <- minorparties::sample_data |>
  minorparties::process_platform_emphasis()
processed_tibble <- process_platform_position(tibble)
```

---

sample\_data

*Sample Data*

---

**Description**

A sample tibble demonstrating the expected input structure for the I-Score pipeline. It includes the platforms of the Republican and Democratic parties from 1948–2024 as well as those of all minor parties that won more than 0.5% of the presidential vote in a given election between those years. Minor parties that won more than 0.5% of the vote in consecutive elections are represented by a single entry spanning those elections. The platform text is that of the election in which the party achieved its highest vote share.

**Usage**

```
sample_data
```

**Format**

A tibble with 53 rows and 4 columns:

**party** Character column. The party’s name (this column must be unique for each platform, e.g., "Democratic Party 1992").

**text** Character column. The full text of each platform.

**minor\_party** Logical column. Whether the party is a minor party.

**major\_party\_platforms** List column. Only required for minor parties. A list containing a list for each major party, each of which contains: **before** (Character. The name (as listed in this tibble’s party column) of the major party’s platform that precedes the minor party), **after** (Character. The name (as listed in this tibble’s party column) of the major party’s platform that follows the minor party), and **weight** (Numeric. The weight assigned to the major party)

## Details

that the package's user would build and feed into the pipeline, containing the platforms of the Republican and Democratic parties from 1948-2024 as well as the platforms of the minor parties that won more than 0.5% of the presidential popular vote in a given election.

---

sample\_data\_results    *Sample Data Results*

---

## Description

A sample tibble demonstrating the expected result structure of the I-Score pipeline. It includes the IScores (along with confidence intervals) of minor parties that won more than 0.5% of the presidential vote in a given election between 1948-2024. These scores were calculated using the sample\_data tibble. Note: Minor parties that won more than 0.5% of the vote in consecutive elections are represented by a single entry spanning those elections.

## Usage

```
sample_data_results
```

## Format

A tibble with 13 rows and 3 columns:

**party** Character column. The party's name (this column must be unique for each platform, e.g., "Democratic Party 1992").

**scores** List column. A list, containing: ie\_score (Numeric. The party's Ie-Score.), ie\_score\_interpreted (Numeric. The party's interpreted Ie-Score), and ip\_score (Numeric. The party's Ip-Score).

**confidence\_intervals** List column. A tibble, containing: side (Character column. "lower" or "upper", indicating the bounds of the confidence interval), ie\_score (Numeric column. The bound of the confidence interval for the party's Ie-Score), ie\_score\_interpreted (Numeric column. The bound of the confidence interval for the party's interpreted Ie-Score), and ip\_score (Numeric column. The bound of the confidence interval for the party's Ip-Score).

# Index

## \* datasets

sample\_data, [7](#)

sample\_data\_results, [8](#)

calculate\_iscores, [2](#)

install\_python, [4](#)

process\_platform\_emphasis, [5](#)

process\_platform\_position, [6](#)

sample\_data, [7](#)

sample\_data\_results, [8](#)