



---

# *Journal of Statistical Software*

June 2011, Volume 42, Issue 6.

<http://www.jstatsoft.org/>

---

## IndElec: A Software for Analyzing Party Systems and Electoral Systems

Francisco Antonio Ocaña  
University of Granada

Pablo Oñate  
University of Valencia

---

### Abstract

**IndElec** is a software addressed to compute a wide range of indices from electoral data, which are intended to analyze both party systems and electoral systems in political studies. Further, **IndElec** can calculate such indices from electoral data at several levels of aggregation, even when the acronyms of some political parties change across districts. As the amount of information provided by **IndElec** may be considerable, this software also aids the user in the analysis of electoral data through three capabilities. First, **IndElec** automatically elaborates preliminary descriptive statistical reports of computed indices. Second, **IndElec** saves the computed information into text files in data matrix format, which can be directly loaded by any statistical software to facilitate more sophisticated statistical studies. Third, **IndElec** provides results in several file formats (text, CSV, HTML, R) to facilitate their visualization and management by using a wide range of application softwares (word processors, spreadsheets, web browsers, etc.). Finally, a graphical user interface is provided for **IndElec** to manage calculation processes, but no visualization facility is available in this environment. In fact, both the inputs and outputs for **IndElec** are arranged in files with the aforementioned formats.

*Keywords:* electoral system, disproportionality, party system, party dimensions.

---

## 1. Introduction

**IndElec** is a software intended to compute a wide range of indices measuring characteristics of party systems and electoral systems in political studies. Among such characteristics, we can briefly mention the disproportionality of an electoral system and some of the main dimensions of a party system, such as fragmentation, effective number of parties, concentration, competitiveness, polarization, regionalism, party linkage and volatility. More detailed information about the indices computed by **IndElec**, including references, is found in Appendix A.

**IndElec** was initially developed to carry out the analysis of all the elections held over 1977–

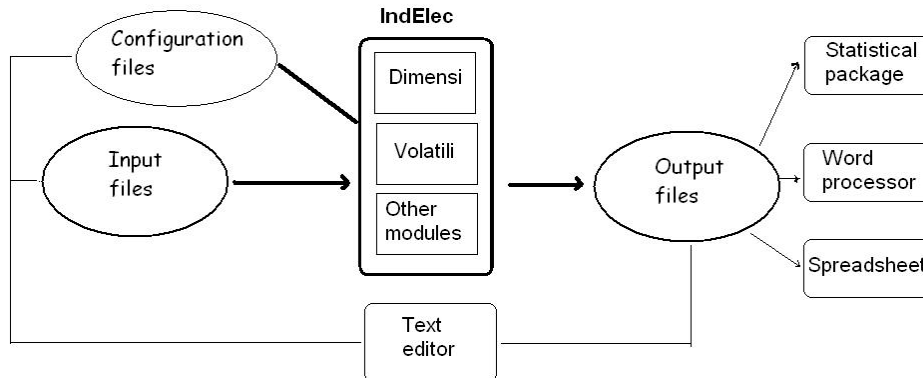


Figure 1: Sketch of the use of **IndElec** in a study.

1999 in Spain, which is available in [Oñate and Ocaña \(1999\)](#). The studied elections were those for the Spanish parliament, the autonomous region parliaments and the European parliament, namely 65 elections in total. The high number of considered elections and the different aggregation levels available in the electoral databases, which were provided by the Spanish Ministry of the Interior, motivated the initial development of **IndElec**. However, this software is now designed to analyze not only the Spanish political system, but also any political system.

From a computational point of view, some of the indices provided by **IndElec** (disproportionality, effective number of parties, fragmentation, etc.) are computed from a data set drawn from an election, which is given by the votes and seats obtained by the competing parties. **IndElec** also computes volatility indices, which depend on data drawn from two (consecutive) elections ([Pedersen 1979](#); [Bartolini and Mair 2007](#)). Apart from its use like a spreadsheet with lot of indices implemented, when the electoral data present several levels of aggregation (state, region, district, etc.), **IndElec** carries out the calculations of such indices for each of the *districts* considered at every level of aggregation. In this data framework, some additional indices are implemented in **IndElec** to compare the effects of data aggregation on some characteristics of the studied political system ([Cox 1999](#); [Oñate and Ocaña 1999](#)), i.e., regionalism and party linkage. Summarizing, more than sixty indices can be calculated by **IndElec** for each electoral distribution. By the way, **IndElec** can even learn to distinguish acronyms of political parties with the user aid, when some political parties present several acronyms across districts. For instance, this practice is common in Spanish elections, like a strategy, when a party wants to catch voters' regionalist feelings ([Lago-Penas 2004](#); [Oñate and Ocaña 1999](#); [Diamandouros and Gunther 2001](#)).

From a technical point of view, **IndElec** consists of several software libraries and a graphical user interface (GUI). Much of **IndElec** is coded in Pascal and its GUI is developed in Object Pascal (an object-oriented extension of Pascal). Though the current Windows binary release of **IndElec** is compiled by using Delphi, **IndElec** but its GUI could be compiled by the classic Borland Pascal compiler or any other freeware version (Free Pascal Compiler–Lazarus, etc.) with minor changes. On the whole, the logic in the programming of **IndElec** distinguishes two modules: **Dimensi** and **Volatili**. **Dimensi** includes the indices depending on an election, and **Volatili** is focused on those indices associated to two elections.

The exchange of information between **IndElec** and the user is conducted mainly through text files, something like the L<sup>A</sup>T<sub>E</sub>X way of work. Figure 1 illustrates this idea by showing a scheme of the use of **IndElec**. Firstly, the input information and some of the settings for **IndElec** (data and other specifications) will be saved into text files by the user. Secondly, the output information obtained by **IndElec**, which is made up of computed indices, statistical analyses and matrices, will also be saved in several text-based files by **IndElec**. This makes the use of **IndElec** easy, because any text editor can manage the files associated to **IndElec**. Moreover, to improve upon the readability and integrability of the **IndElec** output with other softwares, some additional file formats, such as CSV, HTML and R, are considered.

This manuscript is sketched out as follows. The first sections are focused on the module **Dimensi** of **IndElec**. Indeed Sections 2 and 4 explain the management of **Dimensi** for the two considered data frameworks, respectively. In this sense, the implementation of any structure of data aggregation by means of levels is treated in Section 3. The module **Volatili** of **IndElec** is thus described in Section 5. To illustrate some of the details provided in this manuscript, two real data examples will be recurrently considered: the Spanish parliamentary elections held in 2004 and 2000. Finally, the integrability between a statistical software, namely R (R Development Core Team 2011), and **IndElec** is exemplified in Section 6.

## 2. Module **Dimensi** with aggregated data

The aggregated data framework is given when the available electoral data consists of the overall numbers or shares of votes and seats for each of competing parties in a given election. This is the simplest data framework under which **IndElec** can be used. In fact, **IndElec** can thus be viewed like a spreadsheet containing lots of political indices implemented in its code. To illustrate the usage of the module **Dimensi** of **IndElec**, the 2004 Spanish parliamentary election will be considered in what follows.

The aggregated data for a given election must appear in a text file with extension `*.dat`. The information in such a file must be arranged according to the following syntax:

- the first line contains a short description of the electoral data;
- the second line is not taken into account by **IndElec**;
- each of the following lines contains the acronym, the votes and the seats, for each competing party. Any of such quantities for any party can be provided as number, proportion or percentage (the implementation of **IndElec** takes care of such numeric settings).

For example, the aggregated data from the 2004 Spanish parliamentary election, which are contained in the input file `da04.dat`, are arranged as follows:

```
Spanish parliamentary election in 2004-March 14-
Party Vote  Seat
PP 9763144  148
BNG  208688  2
EAJ-PNV 420980  7
PSOE 11026163 164
...
```

As we can see, **IndElec** does not require data alignment by columns.

Under the aggregated data framework, **IndElec** performs the analysis of electoral data and saves the output information in three files with different formats. On the one hand, it generates a text file with extension \*.out and an HTML file (da04.out and da04.htm, in our example). Apart from the indices of disproportionality and those of party dimensions but the volatility, the module **Dimensi** saves the electoral data ordered according to the votes and also their corresponding cumulative distributions of votes and seats. Further, to visualize the disproportionality by parties, it also displays the deviations between votes and seats for each party. For example, in the output of **IndElec** for the data file da04.dat, we can distinguish the following information:

```

                                DATA AND DISTRIBUTIONS OF VOTES AND SEATS
No                               Party      Votes      Seats      %Votes      %Seats
1                               PSOE     11026163   164       43.268     46.857
2                               PP       9763144   148       38.312     42.286
...

```

```

                                CUMULATIVE DISTRIBUTIONS OF VOTES AND SEATS
No      %Cum. Votes      %Cum. Seats
1       43.268         46.857
2       81.579         89.143
3       86.618         90.571
...

```

```

                                PLOT OF DEVIATIONS: %Seats - %Votes
No      Party      %Deviation
1       PSOE      |****      3.59
2       PP        |****      3.97
3       IU        ****|      -3.61
4       CIU       |          -0.42
...

```

On the other hand, to improve the integration with R, **IndElec** generates automatically a R source file which defines some R objects containing the scores of electoral indices computed by **IndElec** (R Development Core Team 2011).

### 3. Defining an aggregation structure in IndElec

Any data aggregation structure given through several levels (discrete aggregation) can be implemented in **IndElec** by the user. Levels of aggregation can be considered in electoral data, when there exists an aggregation structure of *geographic units* or *items* (countries, regions, provinces, districts, etc.) in the *area* where the studied election is held. According to such an aggregation structure, an electoral data distribution is thus gathered for each of those *geographic units*. Indeed such distributions will make up the data set to be provided to **IndElec**.

From a mathematical point of view, let  $R^1$  be the area or *overall region* where a given election took place and  $L$  be the number of aggregation levels to be considered in this region. Each level of aggregation, denoted by  $\ell \in \{1, \dots, L\}$ , is defined by a family  $\mathcal{F}^\ell = \{R_i^\ell : i = 1, \dots, M_\ell\}$  of disjoint *geographic units* such that  $\bigcup_{i=1}^{M_\ell} R_i^\ell = R^1$ , where  $\mathcal{F}^1 = \{R^1\}$  to ensure consistent notation. These families are assumed nested in such a way that  $\forall \ell \in \{2, \dots, L\}$  and  $\forall j \in \{1, \dots, M_\ell\}$ , then there must exist a unique  $i \in \{1, \dots, M_{\ell-1}\}$  such that  $R_i^{\ell-1} \supseteq R_j^\ell$ . Therefore, such an aggregation structure can be viewed as a set of nested layers,  $\{\mathcal{F}^\ell : \ell = 1, \dots, L\}$ , which establish subsequent partitions of the *overall region*,  $R^1$ . Notice that the level of aggregation is determined by  $\ell$  in a decreasing way. Indeed  $\ell$  stands for splitting instead of aggregation.

The aforementioned aggregation structure can be understood by **IndElec**. To this end, the user must implement such an aggregation structure by composing some configuration text files, which must be included in the **IndElec** setup folder. In fact, **IndElec** will not understand an aggregation structure in the provided electoral data, unless such a structure is defined in **IndElec**. So the configuration files for defining an aggregation structure will be detailed in the following paragraphs.

First of all, the main of such configuration files, which must be named `indelec.cfg`, stores a scheme of the aggregation structure to be defined, such as follows:

```
L
nameAgLev1
...
nameAgLevL
```

where `nameAgLev $\ell$`  is a character string which names the aggregation level  $\ell$ ,  $\forall \ell \in \{1, \dots, L\}$ .

Second, for each  $\ell \in \{2, \dots, L\}$ , a configuration file named `nameAgLev $\ell$ .txt` will contain the descriptions of the *geographic units* of the aggregation level  $\ell$ , i.e., the codification of  $\mathcal{F}^\ell = \{R_i^\ell : i = 1, \dots, M_\ell\}$ . To compose a `nameAgLev $\ell$ .txt` file, with  $\ell \in \{2, \dots, L\}$ , the syntax to be considered is given from the following guidelines.

- The first line of the file `nameAgLev $\ell$ .txt` contains the number of *geographic units* for the level  $\ell$ , i.e.,  $M_\ell$ . So the description of *geographic units* starts in the second line of this file.
- Each *geographic unit*  $R_i^\ell$  is identified by the code  $i \in \{1, \dots, M_\ell\}$  and its name (a character string).
- Indeed the description of every *geographic unit*,  $R_i^\ell$ , occupies three lines of the `nameAgLev $\ell$ .txt` file. The first line contains the code  $i$  of  $R_i^\ell$  and also the codes of those *geographic units*, for higher levels of aggregation, containing  $R_i^\ell$ . The name of  $R_i^\ell$  appears in the second line. The third line is always blank to end the description of  $R_i^\ell$ . For example, assume that we have  $R_i^\ell \subseteq R_{i_1}^{\ell-1} \subseteq \dots \subseteq R_{i_{\ell-2}}^2 \subseteq R^1$ . The description of  $R_i^\ell$  is then given by the following three lines:

```
i   i1   ...   iℓ-2
the name of Riℓ
a blank line
```

17	(the number of regions in Spain)
1	(the code of Andalucia)
ANDALUCIA	
	(a blank line)
...	
14	(the code of Pais Vasco)
PAIS_VASCO	
...	

Table 1: A view of the file `CCAA.txt`, which defines the aggregation level given by the 17 autonomous regions in Spain.

Notice that no code is considered for the highest level of aggregation given by  $\mathcal{F}^1 = \{R^1\}$  ( $\ell = 1$ ). Further, apart from the descriptions of *geographic units*, the `nameAgLevel.txt` configuration files specify the nested relationships among the families  $\{\mathcal{F}^\ell : \ell = 1, \dots, L\}$ .

Finally, as reality overcomes theory sometimes, **IndElec** is designed to allow that  $\bigcup_{i=1}^{M_\ell} R_i^\ell \subseteq R^1$ , for some aggregation levels. However, this enters only a slight variation into the logic underlying the theoretic framework considered here.

### 3.1. An example: Spanish parliamentary elections

In the study of Spanish parliamentary elections, it can be worth considering both *regions* and provinces. For the one hand, the provinces are the districts where the electoral rule is applied on. For the other hand, the *regions* are political and cultural unions of provinces (they are called *autonomous regions*). More information on the political map of Spain is available at <http://www.maps.data-spain.com/>

To implement the aggregation structure induced by the Spanish political map, the configuration (text) file `indelec.cfg` will contain the following elements:

```
3
Total
CCAA
Prov
```

This specifies that three aggregation levels ( $L = 3$ ) can be considered, which stand for the aggregation levels given by Spain ( $\mathcal{F}^1 \equiv \text{Total}$ ), the *autonomous regions* ( $\mathcal{F}^2 \equiv \text{CCAA}$ ) and the provinces ( $\mathcal{F}^3 \equiv \text{Prov}$ ). The *geographic units* for the aggregation levels given by `CCAA` and `Prov` are thus defined in the text files `CCAA.txt` and `Prov.txt`, respectively, whose contents are sketched in Tables 1 and 2. For instance, notice how the province *Alava*, which is coded by integer 1 in `Prov.txt`, is defined as included in the autonomous region *Pais Vasco*, which is coded by 14 in `CCAA.txt`.

The Spanish parliamentary elections not only provide an example to illustrate the definition of an aggregation structure in **IndElec**, but also show how **IndElec** can be adapted to real situations partially matching the framework for aggregation structures previously formulated. In fact, the family  $\mathcal{F}^2$ , made up of the Spanish *autonomous regions*, satisfies that  $\bigcup_{i=1}^{17} R_i^2 \subset R^1$  ( $R^1$  is Spain), because two provinces, namely Ceuta and Melilla, are not considered in  $\bigcup_{i=1}^{17} R_i^2$ .

---

52	(the number of provinces or subregions in Spain)
1 14	(the code of Alava is 1; it is included in Pais Vasco)
Alava	
	(a blank line)
...	
4 1	(the code of Almeria is 4; it is included in Andalucia)
Almeria	
...	

---

Table 2: A view of the file `Prov.txt`, which defines the aggregation level given by the 52 provinces in Spain.

Indeed Ceuta and Melilla are endowed by a special legal status (*autonomous cities*), what makes that they are not usually considered in the Spanish political map of *autonomous regions*. However, they are usually included as provinces.

## 4. Module `Dimensi` with levels of data aggregation

In this section, the use of the module `Dimensi` of `IndElec` from data with several aggregation levels will be presented. Roughly speaking, the management of `Dimensi` in this case can be viewed as an interactive process, where the user and `IndElec` exchange information until the final results (the output files) are obtained. By the way, due to the considerable number of input and output files involved in this data framework, it is highly recommended to use a specific folder for each election data set. The exposition in this section will follow the stages to be accomplished in an `IndElec` run under the considered data framework. The step-by-step process so derived is sketched out in Figure 2.

### 4.1. The database

The electoral data with some aggregation levels must be provided to `IndElec` in a (input) text file with extension `*.dab`. Indeed the considered aggregation structure in the data file should be previously defined such as is described in Section 3.

In the electoral data to be provided to `IndElec`, let  $H$  be the number of aggregation levels,  $\ell_1$  be the highest level of aggregation and  $R_{i_1}^{\ell_1}$  be the overall *geographic unit*, where  $H > 1$ ,  $1 \leq \ell_1 < \ell_1 + H - 1 \leq L$  and  $i_1 \in \{1, \dots, M_{\ell_1}\}$ . As we can see, the notation entered in Section 3 will be considered in what follows.

The electoral data must be stored in the `*.dab` text file by following these guidelines.

- The first line of the `*.dab` file contains a short description of the electoral data.
- The integers  $H$  and  $\ell_1$  appear in the following two lines, respectively.
- The integer in the fourth line specifies the overall *geographic unit*. We have two options: it may be the code  $i_1$  or the value zero. The value zero means that the code  $i_1$  will appear in each of the following data *records*; otherwise,  $i_1$  will not appear in those

records. Nevertheless, if  $\ell_1 = 1$ , then any nonzero integer could be considered to name  $R^1$ .

- The fifth line is blank. This establishes the end of the definition of the aggregation structure available in our data. Thus the data records of any of the considered electoral distributions appear sequentially from the sixth line.
- Each party data record occupies  $H + 2$  or  $H + 3$  lines in the `*.dab` file:  $H - 1$  lines, for the  $H - 1$  codes describing the considered geographic unit (if the fourth line contains zero, then an additional line is needed to include  $i_1$ ), and three lines for the acronym, the votes and the seats, respectively, for such a party in such a geographic unit. Finally, we must add a blank line in the data file to establish the end of a party data record.

For instance, let us consider a party with acronym PARTY which obtains  $V$  votes and  $S$  seats in the *geographic unit*  $R_{j_\tau}^{\ell_1+\tau}$ , for any  $\tau < H$  and any  $j_\tau \in \{1, \dots, M_{\ell_1+\tau}\}$ . The figures for  $V$  and  $S$  can be numbers or shares in the file. Further, assume that the *geographic unit*  $R_{j_\tau}^{\ell_1+\tau}$  satisfies that  $R_{j_\tau}^{\ell_1+\tau} \subseteq R_{j_{\tau-1}}^{\ell_1+\tau-1} \subseteq \dots \subseteq R_{j_1}^{\ell_1+1} \subseteq R_{i_1}^{\ell_1}$ , where  $j_s \in \{1, \dots, M_{\ell_1+s}\}$ ,  $\forall s = 1, \dots, \tau$ . Under these settings, its party record in the `*.dab` file is stored as follows:

```

i1 or nil    (level  $\ell_1$ )
j1         (level  $\ell_1 + 1$ )
...
j $\tau-1$ 
j $\tau$       (level  $\ell_1 + \tau$ )
 $\sigma(\ell_1 + \tau + 1)$   (level  $\ell_1 + \tau + 1$ )
...
 $\sigma(\ell_1 + H - 1)$   (level  $\ell_1 + H - 1$ )
PARTY
V
S
a blank line

```

For each level  $\ell$ , the code  $\sigma(\ell)$  is an integer such that  $\sigma(\ell) > M_\ell$ , which stands for the collapse of the aggregation level  $\ell$ . **IndElec** automatically recognizes such codes  $\sigma(\ell)$ ,  $\forall \ell$ , from the `*.dab` file.

To illustrate the structure of a `*.dab` data file, we consider the 2004 Spanish parliamentary election with the aggregation structure defined in Section 3.1. The corresponding electoral data are available in the file `spain4ag.dab`, where its data records are included such as is described in Table 3.

In these electoral data, we can consider some records for the Spanish worker's socialist party (PSOE), which are roughly illustrated in Table 4. This table shows a common practice for some parties in elections in Spain: the acronym of a party changes across regions or districts in order to catch the regionalist feelings of potential voters. This means that PSOE-A, PSOE and PSE-EE, among others, are official acronyms of the same political party. This curious practice presents a serious problem in data analysis, because the parties are usually labeled in official databases by using several official acronyms. **IndElec** provides a way to sort out this problem, which is described in Section 4.2.



---

```

2004 Spanish parliamentary election
3      (the number of considered aggregation levels)
1      (the maximum aggregation level)
1      (the code of Spain)
      (a blank line)
...
1      (begin the PSOE data record in Almeria)
4
PSOE-A
145868
3
      (a blank line: end of the PSOE data record in Almeria)
...
1      (the PSOE data record in Andalucia)
99     (any code greater than 52)
2377455
38
99     (the PSOE data record in Spain; any code greater than 17)
99     (any code greater than 52)
PSOE
11026163
164
14     (the PSOE data record in Alava)
1
PSE-EE
56137
2
...

```

---

Table 3: A view of the text file `spain4ag.dab`, which contains the data from the 2004 Spanish parliamentary election at 3 aggregation levels (Spain, autonomous regions and provinces). PSOE denotes the Spanish worker's socialist party. Source: the Spanish Ministry of the Interior.

Acronym	Unit	# Votes	# Seats
PSOE-A	Almería	145868	3
PSOE-A	Andalucía	2377455	38
PSOE	Spain	11026163	164
PSE-EE	Alava	56137	2

Table 4: Some of the official data records for the Spanish worker's socialist party in the 2004 Spanish parliamentary election. Source: The Spanish Ministry of the Interior.

123	(total number of acronyms)
CC	(Canary Island Coalition)
PANE	(Regional-wide party)
PSE-EE	(Spanish worker's socialist party)
NO-PANE	(State-wide party)
PSOE	(Spanish worker's socialist party)
NO-PANE	(state-wide party)
PSOE-A	(Spanish worker's socialist party)
NO-PANE	(state-wide party)
PP	(Popular Party)
NO-PANE	(state-wide party)
...	(more acronyms)

Table 5: A view of the text file `siglas.txt`, which contains the acronyms and regionalist profile of competing parties in the 2004 Spanish parliamentary election.

## 4.2. Management of acronyms

When the `*.dab` data file is provided to **IndElec** (or **Dimensi**), an information exchange process is performed between the user and **IndElec**. In this step-by-step process, on the one hand, the user teaches **IndElec** by providing information about parties and, on the other hand, **IndElec** eases the user's work by generating preliminary templates of some input files to serve in subsequent steps.

First, **IndElec** extracts all the acronyms from the `*.dab` file in a text file named `siglas.txt`. This file thus contains the acronyms recognized by **IndElec** from the provided data. However, the user must supply to **IndElec** additional information about the parties referred to by the acronyms in `siglas.txt`. In fact, the **IndElec** generated version of `siglas.txt` is just a template, where the user must specify whether the acronym belongs to a state-wide party, labeled by `NO-PANE`, or to a regional-wide party, labeled by `PANE`. To this end, the user will edit `siglas.txt` and then write down `NO-PANE` or `PANE` below each party acronym. After specifying this information in `siglas.txt`, the user version of `siglas.txt` is read by **IndElec** to incorporate the regional-national information. For the 2004 Spanish parliamentary election, the `siglas.txt` file to be provided to **IndElec** is described in Table 5.

Second, the problem of the acronym change across districts is solved through **IndElec**. Mathematically speaking, the solution of the problem consists of establishing the quotient set from the set of party acronyms, which appears in `siglas.txt`, where the equivalence relation establishes that the acronyms are equivalent when they are associated to the same political party. Indeed this quotient set of acronyms is defined from its equivalence classes, which are the subsets of acronyms belonging to the same party. The solution will be implemented by the user in the input text file `siglaso.txt`. In fact, this file will contain the aforementioned equivalence classes by following this guidelines:

96	(the number of equivalence classes)
CC	(an example of class with one acronym)
	(a blank line)
PSOE	(the equivalence class of the PSOE party)
PSOE-A	(the PSOE acronym in Andalucia)
PSE-EE	(the PSOE acronym in the Basque Country)
PSC-PSOE	(the PSOE acronym in Cataluna)
PSDEG-PSOE	(the PSOE acronym in Galicia)
	(a blank line: end of the equivalence class of PSOE)
...	(more equivalence classes)

Table 6: A view of the text file `siglaso.txt`, which identifies the set of acronyms considered for each party competing in the 2004 Spanish parliamentary election.

- the first line of `siglaso.txt` contains the number of equivalence classes of acronyms and,
- for any equivalence class of acronyms, each acronym appears in a line of `siglaso.txt` and the end of its description is points out by a blank line.

For example, in the 2004 Spanish parliamentary election, the final version of `siglaso.txt` to be provided to **IndElec** is described in Table 6.

As the construction of `siglaso.txt` from scratch can be laborious for the user, **IndElec** provides a preliminary version of `siglaso.txt` to be only modified by using any text editor, where the acronyms considered at the highest level of aggregation are distinguished.

Finally, as some polarization indices can be obtained by **IndElec**, the (left–right) ideological scores in the interval  $[0,10]$ , for every party, must be supplied in the input text file `siglapo.txt`. The syntax of this file is inspired on that of `siglaso.txt`. In fact, to easily obtain `siglapo.txt`, we can modify `siglaso.txt` by adding such party scores in the first line of any record, where now each party is viewed as an equivalence class of acronyms in `siglaso.txt`. However, the equivalence classes in `siglapo.txt` are not necessarily equal to those in `siglaso.txt`.

In the 2004 Spanish parliamentary election, the input file `siglapo.txt` is illustrated in Table 7.

### 4.3. Output files

From data with several levels of aggregation, **IndElec** computes lot of political indices for each electoral distribution (set of pairs, votes and seats, for every party) associated to each of the *geographic units* in every aggregation level. Further, **IndElec** computes other political indices quantifying properties of party systems changing across the geographic aggregation (regionalism and party linkage, mainly). For example, in the 2004 Spanish parliamentary election, **IndElec** considers 70 electoral distributions (Spain +17 regions +52 provinces) and carries out 121 comparative studies (regions & Spain, provinces & region, provinces & Spain). **IndElec** stored the vast amount of output information with a statistical report in the text file `result.out`. This report, which includes, among other measures, descriptive statistics, some

---

96	
5.69	(the CC ideological score)
CC	
4.27	(the PSOE ideological score)
PSOE	
PSOE-A	
PSE-EE	
PSC-PSOE	
PSDEG-PSOE	
...	

---

Table 7: A view of the text file `siglapo.txt`, which enters the ideological scores for the parties in the 2004 Spanish parliamentary election.

exploratory statistics (median, quartiles), covariance and correlation matrices, is automatically elaborated by **IndElec** to provide a first approach of the results. Moreover, the contents of `result.out` are available in both CSV and HTML formats. For the HTML format, **IndElec** additionally generates a version of `result.out` with frames which is available in `resultf.htm` (the version without frame is given by `result.htm`).

In order to facilitate the statistical analysis of the results derived by **IndElec**, they are organized in two data matrices (data frames, in the R terminology), which are stored in two kind of files. **IndElec** automatically generates both the text and CSV formats for the aforementioned files. In fact, the output files `matrIREG.*` will contain the computed indices of regionalism and party linkage and the files `matrizDD.*`, the rest of indices derived by **Dimensi**. Therefore, these output files can be loaded as data file to any statistical software (R, S, SPSS, etc.), in order to perform sophisticated statistical analysis from the results derived by **IndElec**.

## 5. The module **Volatili**

**Volatili** is the module of **IndElec** addressed to calculate the volatility indices (Pedersen 1979; Katz, Rattinger, and Pedersen 1997). Associated to two elections held in two dates (years, for instance) rather than to one election, such as is the case with **Dimensi**, the implementation of volatility indices in **IndElec** was carried out in a special software module, which utilizes the internal calculations (binary files, etc.) previously obtained by **Dimensi** for each election. The implemented indices in **Volatility** are the *total volatility* indices proposed by Pedersen (1979) and a generalization of the *bloc volatility* indices suggested by Bartolini and Mair (2007). Moreover, the two data frameworks previously considered (aggregated data and data with aggregation levels) can be also managed by **Volatili**.

In political studies, volatility is a dimension quantifying the changing patterns in a party system, i.e., the total transfer of votes among political parties or blocs of parties between two *consecutive* elections. Pedersen (1979) suggested an index that quantifies such transfers among parties: *the index of total volatility*. The *Pedersen volatility measure* (PVM) became more sophisticated when Bartolini and Mair (2007) tried to explain the electoral change taking

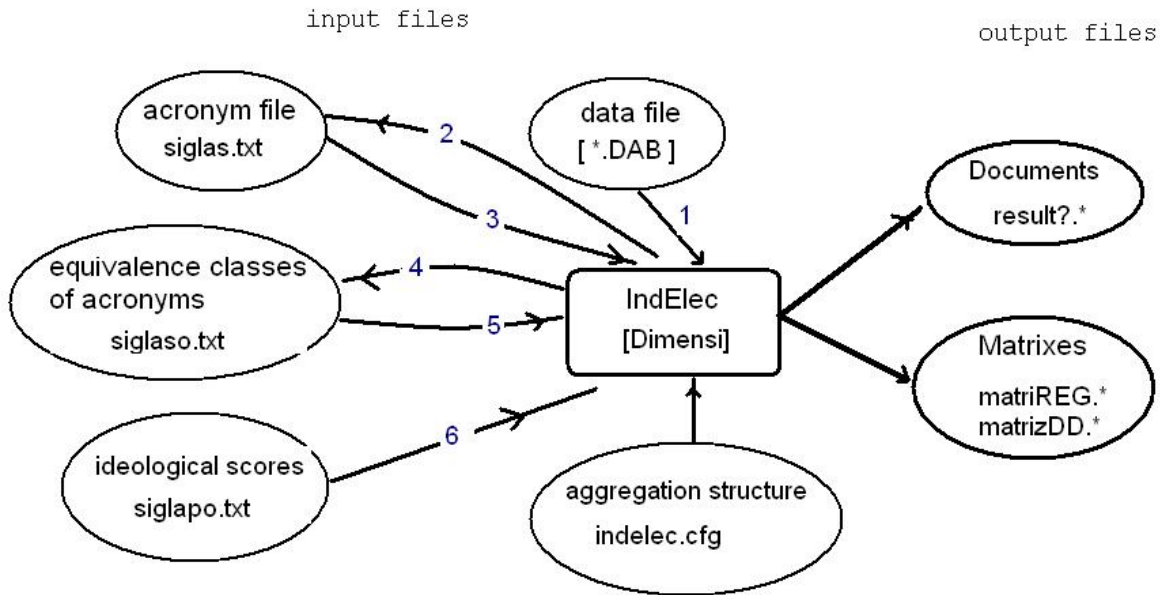


Figure 2: Management of the module **Dimensi** of **IndElec** from data with aggregation levels. The integers stands for the order in the step-by-step process for an **IndElec** run.

into the alignment of parties according to two ideological blocs, namely the left-wing parties and the right-wing parties. These authors thus defined the indices of *(inter) bloc volatility* and *intra-bloc volatility*.

A state of the art of the PVM can be found in [Katz et al. \(1997\)](#), where the broad range of its current applications and some suggestions about this dimension are pointed out. Such suggestions have motivated the generalization of the bloc volatility indices in **IndElec** by letting any number of blocs. To this end, the user will specify both the number of blocs and the character standing for each of such blocs in the configuration text file `simbolos.afi`, which must appear in the **IndElec** setup path. The syntax for `simbolos.afi` is sketched out as follows:

- the first line contains the number of blocs;
- each considered bloc is specified in a line by a character.

For example, if the user wants to consider those blocs in [Bartolini and Mair \(2007\)](#) (the left-wing parties and the right-wing parties), the file `simbolos.afi` will be as follows:

```

2
R
L

```

### 5.1. Party experienced increments for volatility indices

Though the PVM formula is very simple, some computational problems can arise when it is calculated from real data in practice. The main problems appear when the sets of competing parties in both considered elections, respectively, are not identical, such as is theoretically assumed in the PVM formula (Pedersen 1979). This problem arises when, for example, there are changes of party acronyms, merging of parties into coalitions or splitting former parties into new parties, etc. over both consecutive elections. The increments in votes or seats experienced by some parties, between both considered elections, are not so evident in such situations. Therefore, the PVM formula, which depends on such party experienced increments, could not be computed properly from some real data in practice.

These computational problems are solved in Bartolini and Mair (2007, Appendix 1, pp. 311–312) and Ocaña (2007). Bartolini and Mair propose a set of guidelines describing how to do in a wide range of such problematic situations, where the sets of competing parties are not identical. Once these guidelines are applied to our data, the equality of the sets of competing parties in both elections can be assumed in the so transformed electoral data. To sort out this problem, **IndElec** provides the way of implementing the Bartolini and Mair’s rules by means of a input text file with extension `*.ivo`. Moreover, the alternative approximative volatility formulae developed by Ocaña (2007) are also implemented in **IndElec**.

Though the `*.ivo` input file will depend on the considered electoral data framework, it always includes the implementation of the party experienced increments by following a common syntax for both data frameworks. This syntax establishes that any increment for a *party* (party, coalition, etc.) is included in a `*.ivo` file by following these guidelines:

- the first line, for such an increment, contains the character of the bloc where the increment must be included for the bloc volatility indices;
- from the second line, each of the acronyms of parties in the second election, for such an increment, will appear in a line of the input file;
- the end of the above list of acronyms for the second election is established by a blank line (the first blank line);
- after the first blank line, each of acronyms of parties in the first election, for the considered increment, will appear in a line of the input file;
- the end of the above list of acronyms for the first election is given by a blank line (the second blank line);

For example, assume that the  $i$ -th increment experienced by parties between two elections is given by  $p_2$  parties (acronyms) from the second election and  $p_1$  parties (acronyms) from the first election. Further, suppose that such an increment is in the  $b$ -th bloc for the bloc volatility indices. The **IndElec** user can implement such an increment by composing the following contents in the corresponding `*.ivo` input file:

Character of the bloc  $b$

Party <sub>$i_1$</sub> <sup>(2)</sup> (the acronym of the  $i_1$ -th party in the 2nd election)  
 ... (more parties of this increment in the 2nd election)  
 Party <sub>$i_{p_2}$</sub> <sup>(2)</sup> (the acronym of the  $i_{p_2}$ -th party in the 2nd election)  
 (the first blank line)  
 Party <sub>$j_1$</sub> <sup>(1)</sup> (the acronym of the  $j_1$ -th party in the 1st election)  
 ... (more parties of this increment in the 1st election)  
 Party <sub>$j_{p_1}$</sub> <sup>(1)</sup> (the acronym of the  $j_{p_1}$ -th party in the 1st election)  
 (the second blank line)

It makes that **IndElec** incorporates the increment given by

$$\sum_{k=1}^{p_2} F(\text{Party}_{i_k}^{(2)}) - \sum_{h=1}^{p_1} F(\text{Party}_{j_h}^{(1)}),$$

into the volatility formulae, where  $F(\text{Party}_{\ell}^{(t)})$  stands for the vote or seat share of the party named by the acronym  $\text{Party}_{\ell}^{(t)}$ , which is the  $\ell$ -th party in the  $t$ -th election ( $t=1,2$ ). Notice that either  $p_1$  or  $p_2$  may be zero and that two blank lines must always appear for each increment. Moreover, when the electoral data presents several levels of aggregation, it is not necessary to specify the acronyms of a party across the districts. **IndElec** learns such information from the corresponding `siglaso.txt` files for both considered elections, respectively, where **Dimensi** must have been previously applied.

## 5.2. Usage of Volatili

Roughly speaking, the usages of the module **Volatili** for the electoral data frameworks managed by **IndElec**, data aggregated and data with aggregation levels, present nonsignificant differences. However, some big differences arise, whether the programming of **Volatili** is taken into account for both data frameworks. As a user guide of **Volatili**, this section is focused on its usage and, then, it will contain an unified description of **Volatili** as compared to **Dimensi** for both cases.

The aforementioned similarity in the **Volatili** usage is due to the computational design. Indeed **Volatili** requires the previous execution of **Dimensi** for each of both studied elections. The binary files so generated by **Dimensi** for each election, which depend on the electoral data framework, provides the information needed to start **Volatili** calculations. In fact, in order to calculate volatility indices, the only specific information for **Volatili** is given by the party experienced increments, between both studied elections, which are needed to apply the volatility formulae (Bartolini and Mair 2007).

### *The input file*

Such as was established in previous section, the party increments between both elections are implemented into an `*.ivo` text file. Further, the considered electoral data framework enters only a tiny difference in the information saved in such a file, which is located in its first four lines. Indeed the syntax of this header of the `*.ivo` file is given by the following guidelines:

- The first line contains the path of the folder where the data from the second election are stored. In a similar way, the third line contains that path of the first election.

- The fifth line is always blank.
- The party increments are thus arranged from the sixth line.
- The differences by the data framework are found in the second and fourth lines of the header of the \*.ivo file. If the electoral data are aggregated, then the name of the data file (without its extension \*.dat) will appear below its corresponding election working path. If the electoral data presents several aggregation levels, then a number labeling each election will appear below each election path (the year, for instance).

This way the content of any \*.ivo text file is sketched out as follows:

```

the path for the 2nd election
the data file name or a label, for the 2nd election
the path for the 1st election
the data file name or a label, for the 1st election
  (a blank line)
Now, the descriptions of party increments...
```

### *Output files*

The output information of **Volatili** follows the same idea of the output files of **Dimensi**. First, the scores of volatility indices are saved in report style into text and HTML formats; the CSV format is also available for disaggregated data. Such report files are named as the \*.ivo file with the extensions \*.res, \*.htm and \*.csv, respectively. Second, for aggregated data, **IndElec** also generates automatically a R source file which defines some R objects containing the volatility scores computed by **IndElec** (R Development Core Team 2011). Third, the computed volatility indices are saved in data matrix style in text and CSV formats with a common name, `matVolat`.

## 6. Using R and IndElec

This section will illustrate the integration of the statistical software R (R Development Core Team 2011) and **IndElec** through some data examples, according to both electoral data frameworks considered in this paper. Indeed **IndElec** provides a significant level of integrability with any statistical software, such as has been explained across this paper. However, **IndElec** provides some additional facilities to R users, which are illustrated in this section.

Roughly speaking, this section will demonstrate how a data frame can be (1) exported from R, (2) analyzed in **IndElec** and then (3) the so obtained results imported to R. Indeed the emphasis will be on the steps (1) and (3), because the step (2) has already been treated in previous sections. Further, taking into account the two modules of **IndElec**, **Dimensi** and **Volatili**, the step (3) is accomplished in the same way. However, as the electoral data files considered by **Volatili** must have been previously taken by **Dimensi**, the step (1) can only be explained for **Dimensi**. Notice that the relevant information needed by **Volatili** is only provided by the input file of the party increments (see Section 5.1). Therefore, the examples in this section will only illustrate the interactions of the module **Dimensi** of **IndElec** and R.



## 6.1. Aggregated electoral data

Consider the aggregated electoral data from the 2004 Spanish parliamentary election, which are presented in Section 2. Assume that these data are stored in a R data frame named `rdaf`. The R data frame `rdaf` could be easily obtained from the data file `da04.dat` described in Section 2. To this end, the sentence in R

```
R> rdaf <- read.table(file = "da04.dat", header = TRUE, skip = 1)
```

reads `da04.dat` and skips its first line (it is a short data description). The three inherited variables (columns) of `rdaf` are named as `Party`, `Vote` and `Seat`, respectively.

In this framework, the input file for **IndElec** from a given R data frame is derived by the R function `Adata2IndElec`, which is provided in the **IndElec** distribution. This function creates the data file in the form that the module **Dimensi** needs from a standard R data frame containing aggregated electoral data. Its header is given by

```
Adata2IndElec(dataName = "", acronyms, votes, seats, sTitle = "")
```

where `dataName` is a character string containing the input data filename for **IndElec** to be created, `acronyms` is a string vector of the acronyms of the parties competing in the considered election, `votes` and `seats` are numeric vectors containing the votes and seats, respectively, of the considered parties, and `sTitle` is a character string containing a short description of the electoral data. For instance, taking into account the proposed example, the R sentence

```
R> Adata2IndElec("da04n", rdaf$Party, rdaf$Vote, rdaf$Seat,
+ "2004 Spanish parliamentary election")
```

will generate the input data file `da04n.dat` for **IndElec** in the R working directory from the data frame `rdaf`.

The contents of `da04n.dat` and `da04.dat` are equal. Therefore, the same conclusion holds for their corresponding output files.

The outputs of **IndElec** from `da04n.dat` would be arranged in several files with different formats, such as is explained in Section 2. Indeed **IndElec** would derive the following output files: `da04n.out`, `da04n.htm` and `da04n.R`, which all contain the same results but with different formats. Particularly, `da04n.R` would be a R source file which defines the results by **IndElec** from `rdaf` as a R list.

## 6.2. Disaggregated electoral data

This section will illustrate how **IndElec** can be applied on data with aggregation levels stored as a data frame in R. In this data framework, two different situations can be considered.

1. The available electoral data, which are saved in the given R data frame, are only those of the lowest level of aggregation. This means that an aggregation process is needed to obtain the electoral data for the rest of levels of aggregation.
2. The electoral data for all aggregation levels are available in the given R data frame.

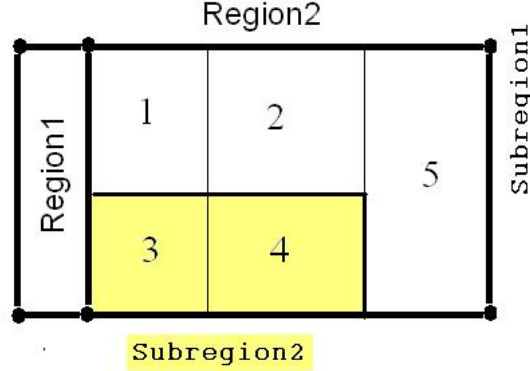


Figure 3: Map of the artificial state considered in the example of Section 6.2. Possible levels of aggregation: state, region, subregion and district.

To obtain the input data file for **IndElec**, two R functions, namely `DA2IndElec` and `D02IndElec`, are provided in the **IndElec** distribution, for both situations, respectively. Nevertheless, the aforementioned situations on the available data determine only which of the provided R function must be considered. In fact, the rest of steps to accomplish the task proposed in this section are the same for both situations. Because of this, we will only present an example of the first situation, which is artificial to make extensive use of R.

Consider a state which consists of two regions, named by *Region1* and *Region2*. *Region2* is divided into *Subregion1* and *Subregion2* (yellow color in Figure 3). Further, these subregions are split in five districts, which are labeled by an index: three districts (1, 2 and 5) in *Subregion1* and two districts (3 and 4) in *Subregion2*. To visualize the so defined aggregation structure in this artificial state, its map is depicted in Figure 3.

Theoretically, four levels of aggregation are assumed in such a state, namely  $\mathcal{F}^1 \equiv \text{State}$ ,  $\mathcal{F}^2 \equiv \text{Region}$ ,  $\mathcal{F}^3 \equiv \text{Subregion}$  and  $\mathcal{F}^4 \equiv \text{District}$ , where the notation in Section 3 is considered. However, to illustrate the sophistication of **IndElec**, we will consider that the election under study was held only in *Region2*, and then that its corresponding electoral data are drawn from each of its districts. Taking into account the general framework in Section 4, in our problem, the highest level of aggregation is thus *Region*, with  $\ell_1 = 2$ , for the regional unit *Region2*, labeled by  $i_1 = 2$ , and three aggregation levels are considered in the electoral data,  $H = 3$ , namely *Region*, *Subregion* and *District*. Nevertheless, we will assume that the available electoral data are only given by those of the *District* level (the first situation above).

**Step 0.** For the election held in *Region2*, its electoral data drawn from each of its districts are going to be generated in R. Consider 3 parties competing across the five districts of *Region2*, where the parties are labeled by  $Pa$ , for any  $a = 1, 2, 3$ , for instance. Assume that the distributions of the numbers of votes and seats are Poisson with parameters 20 and 3, respectively, for instance. As the information (description) on the considered districts must be joined to each data record, the variables of the electoral data frame can be generated in R as follows:

```
R> noDat <- 3 * 5
R> Parties <- gl(3, 1, label = c("P1", "P2", "P3"), length = noDat)
```

```
R> LDistri <- gl(5, 3, length = noDat)
R> LSubreg <- gl(2, 2 * 3, length = noDat)
R> LRegion <- rep(2, noDat)
R> v <- rpois(n = noDat, lambda = 20)
R> s <- rpois(n = noDat, lambda = 3)
R> rdatD <- data.frame(LRegion, LSubreg, LDistri, Parties, v, s)
```

This way the R data frame `rdatD` contains the electoral data obtained by each of parties in each of the five districts, i.e.,

	LRegion	LSubreg	LDistri	Parties	v	s	
1	2	1	1	P1	19	8	(P1 in district 1)
2	2	1	1	P2	33	4	(P2 in district 1)
3	2	1	1	P3	22	5	(P3 in district 1)
4	2	1	2	P1	19	6	(P1 in district 2)
	...						...
14	2	1	5	P2	19	1	(P2 in district 5)
15	2	1	5	P3	28	5	(P3 in district 5)

This data generation process is generalized in the R source file `exampAg.R`, which is available in the **IndElec** distribution.

**Step 1.** Once the disaggregated electoral data are available in a data frame, namely `rdatD`, the input data file for **IndElec** can be made by using one of the *ad hoc* R functions, `DA2IndElec` or `D02IndElec`. These functions are managed in the same way. In fact, the only difference between both R functions is found in the electoral data in the R data frame. On the one hand, when the available electoral data are only those of the lowest level of aggregation, and thus a data aggregation process must be carried out to obtain the electoral data for the rest of levels, `DA2IndElec` must be executed. On the other hand, when all electoral data are available, `D02IndElec` must be executed instead. Therefore, in our example we must consider

```
DA2IndElec(dataName = "", 11, agLevels, parties, votes, seats, sTitle = "")
```

where `dataName` is a character string containing the name of the input file to be created, `11` is the index of the highest level of aggregation, `agLevels` is a list containing the aggregation levels sorted in decreasing order of aggregation (each aggregation level is coded by a R factor), `parties` is a vector of strings (R factor) of party acronyms, `votes` and `seats` are numeric vectors containing the votes and seats of the competing parties, respectively, and `sTitle` is a short description of the electoral data, which will be included in the first line of the input file to be generated. For instance, taking into account the considered example, the R sentence

```
R> DA2IndElec("Regi2D", 2, list(rdatD$LRegi, rdatD$LSubreg, rdatD$LDistri),
+   rdatD$Parties, rdatD$v, rdatD$s, sTitle="Region2 election")
```

will generate the data file `Regi2D.dab` for **IndElec** in the R working directory from the disaggregated electoral data contained in the data frame `rdatD`.

**Step 3.** **IndElec** must be prepared to understand the aggregation structured in the electoral data to be analyzed, such as is established in Section 3. To this end, we must modify the file `indelec.cfg` to define the potential levels of aggregation to be considered in the artificial state (Figure 3), such as follows

```

4
State
Region
Subreg
District

```

This implies that the four levels of aggregation are defined in the configuration files named as `Region.txt`, `Subreg.txt` and `District.txt`. In fact, the region level is defined in `Region.txt` as follows

```

2
1
Region1

2
Region2

```

The subregion level is defined in `Subreg.txt` as follows

```

3
1 2
Subregion1

2 2
Subregion2

3 1          (it is not necessary)
Region1

```

Finally, the district level is defined in `District.txt` as follows

```

6
1 1 2
District1

...
3 2 2
District3

...
5 1 2
District5

6 3 1      (it is not necessary)
Region1

```

These configuration files make possible the analysis of the data file `Regi2D.dab` by using **IndElec** (see Section 4).

**Step 4.** After the **Dimensi** step-by-step run (Figure 2), a lot of output files are generated by **IndElec** (see Section 4.3) with different purposes. Among such output files, the `matriREG.*` and `matrizDD.*` files let import the **IndElec** results to R. For instance, taking into account the CSV matrix files, the results are stored in two R data frames as follows:

```
R> rOutDD <- read.csv("matrizDD.csv")
R> rOutRL <- read.csv("matriREG.csv")
```

where `rOutRL` contains of the regional and party linkage indices and `rOutDD`, the rest of indices computed by **IndElec**.

## 7. Conclusions

This paper presents a software devoted to help the political researcher in the analysis of party systems and electoral systems. **IndElec** can calculate more than fifty political indices measuring characteristics of electoral systems and party systems, from electoral data. However, **IndElec** is flexible, because it can be adapted with the user aid to several situations arising when real electoral data are considered in a study (the presence of aggregation levels in data, party with several acronyms across districts, among others). Nevertheless, its development is always in progress (Oñate and Ocaña 2000, 2005; Ocaña and Oñate 2006; Ocaña 2007).

Finally, an important point is the integrability of the **IndElec** output with other softwares (word processor, spreadsheet, statistical softwares, etc.), which is achieved through the considered output file styles. On the one hand, the readability of the **IndElec** output is provided through the report-style files. Apart from providing an inspection tool to the user, they also let composing texts in the input files. On the other hand, the vast amount of scores obtained from disaggregated electoral data can be analyzed by any statistical software through the matrix-style output files. Moreover, R's users can easily manage the **IndElec** output derived such as described in Section 6.

## Acknowledgments

The authors are indebted to Micah Altman, the referees and Achim Zeileis for constructive criticism and useful comments in the original versions of both the software and the manuscript.

## References

- Arian A, Weiss S (1969). "Split-Ticket Voting in Israel." *Western Political Quarterly*, **24**, 375–389.
- Bartolini S, Mair P (2007). *Identity, Competition and Electoral Availability: The Stabilization of European Electorates 1885–1985*. 2nd edition. ECPR Press, Essex.
- Chhibber P, Kollman K (1998). "Party Aggregation and the Number of Parties in India and the United States." *American Political Science Review*, **92**, 329–342.

- Cox GW (1999). "Electoral Rules and Electoral Coordination." *Annual Review of Political Science*, **2**, 145–161.
- Cox GW, Shugart MS (1996). "Strategic Voting under Proportional Representation." *Journal of Law Economics and Organisations*, **12**(2), 299–324.
- Diamandouros N, Gunther R (eds.) (2001). *Parties, Politics and Democracy in the New Southern Europe*. Johns Hopkins University Press, Baltimore.
- Gallagher M (1991). "Proportionality, Disproportionality and Electoral Systems." *Electoral Studies*, **10**(1), 33–51.
- Hazan RY (1997). *Centre Parties: Polarization and Competition in European Parliamentary Democracies*. Pinter, London.
- Katz RS, Rattinger H, Pedersen MN (1997). "The Dynamics of European Party Systems." *European Journal of Political Research*, **31**(1), 83–97.
- Kesselman M (1966). "French Local Politics: A Statistical Examination of Grass Root Consensus." *American Political Science Review*, **60**(4), 963–973.
- Laakso M, Taagepera R (1979). "Effective Number of Parties. A Measure with Application to West Europe." *Comparative Political Studies*, **12**(1), 3–27.
- Lago-Penas I (2004). "Cleavages and Thresholds: The Political Consequences of Electoral Laws in the Spanish Autonomous Communities, 1980–2000." *Electoral Studies*, **23**, 23–43.
- Lijphart A (1994). *Electoral Systems and Party Systems: A Study of Twenty-Seven Democracies, 1945–1990*. Oxford University Press, Oxford.
- Loosemore J, Hanby VJ (1971). "The Theoretical Limits of Maximum Distortion: Some Analytic Expressions for Electoral Systems." *British Journal of Political Science*, **1**(4), 467–477.
- Mackie TT, Rose R (1982). *The International Almanac of Electoral History*. McMillan, London.
- Mackie TT, Rose R (1991). *The International Almanac of Electoral History*. Congressional Quarterly Press, Washington.
- Moenius J, Kasuya Y (2004). "Measuring Party Linkage across Districts. Some Party System Inflation Indices and Their Properties." *Party Politics*, **10**(5), 543–564.
- Molinar J (1991). "Counting the Number of Parties: An Alternative Index." *American Political Science Review*, **85**(4), 1383–1391.
- Ocaña FA (2007). "An Approximation Problem in Computing Electoral Volatility." *Applied Mathematics and Computation*, **192**(2), 299–310.
- Ocaña FA, Oñate P (2006). "Las Arenas Electorales en España y la Normalidad de la Convocatoria de Marzo de 2004." In J Molins, P Oñate (eds.), *Elecciones y Competición Electoral en la España Multinivel*, pp. 23–77. Centro de Investigaciones Sociológicas, Madrid.

- Oñate P, Ocaña FA (1999). *Análisis de Datos Electorales*, volume 27 of *Cuadernos Metodológicos*. Centro de Investigaciones Sociológicas, Madrid.
- Oñate P, Ocaña FA (2000). “Elecciones de Marzo de 2000: ¿Cuánto Cambio Electoral?” *Revista de Estudios Políticos*, **110**, 297–336.
- Oñate P, Ocaña FA (2005). “Las Elecciones de Marzo de 2004 y los Sistemas de Partidos en España: ¿Tanto Cambio Electoral?” *Revista Española de Ciencia Política*, **13**, 159–182.
- Pedersen MN (1979). “The Dynamics of European Party Systems: Changing Patterns of Electoral Volatility.” *European Journal of Political Research*, **7**, 1–26.
- Pennisi A (1998). “Disproportionality Indexes and Robustness of Proportional Allocation Methods.” *Electoral Studies*, **17**(1), 3–19.
- Rae DW (1971). *The Political Consequences of Electoral Laws*. 2nd edition. Yale University Press, New Haven.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- Sani G, Sartori G (1983). “Polarization, Fragmentation and Competition in Western Democracies.” In H Daalder, P Mair (eds.), *Western European Party Systems: Continuity and Change*, pp. 307–340. Sage Publications, Beverly Hills.
- Sartori G (2005). *Parties and Party Systems: A Framework of Analysis*. 2nd edition. ECPR Press, Essex.
- Taagepera R, Grofman B (2003). “Mapping the Indices of Seats-Votes Disproportionality and Inter-Election Volatility.” *Party Politics*, **9**(6), 659–677.
- Theil H (1969). “The Desired Political Entropy.” *American Political Science Review*, **63**(2), 521–525.
- Tukey JW (1977). *Exploratory Data Analysis*. Addison–Wesley, Reading Massachusetts.
- Wildgen JK (1971). “The Measurement of Hyperfractionalization.” *Comparative Political Studies*, **4**(2), 233–243.

## A. Indices computed by IndElec: Formulae and references

This appendix gathers information on the main political indices computed by **IndElec**, namely their formulae and some of their references.

First of all, consider a given election under study. Let  $\mathcal{I} = \{1, \dots, N\}$  be the set of parties competing in such an election and  $\{(V_i, S_i) : i \in \mathcal{I}\}$  be the joint distribution of votes and seats, expressed in percentages, which summarizes the electoral results obtained by these parties. Further, to ease notation,  $p_i$  will denote the proportion of votes or seats, indistinctively, for the  $i$ -th political party, for any  $i \in \mathcal{I}$ .

Though only necessary for some indices, we will consider that the competing parties are ordered according to their obtained votes such as follows:  $V_i \geq V_{i+1}, \forall 1 \leq i < N$ . However, the distortion yielded by the electoral system makes that this order is not necessarily maintained for their seats.

In some elections, the number of parties,  $N$ , can be extremely high and, thus, lot of parties have no seat. Under such circumstances, some indices may present inappropriate behaviors (Lijphart 1994), since a high proportion of parties with no seat is involved in the calculation of such indices. Because of this, some alternatives for  $N$  into the formulas of some political indices, which are proposed in the literature, have been implemented in **IndElec**.

- Lijphart (1994) suggests  $N_L = \max\{J \in \mathcal{I} : V_J > 0.5\}$ .
- Oñate and Ocaña (1999) consider two alternatives:  $N_S = \min\{J \in \mathcal{I} : \sum_{i=1}^J S_i = 100\}$  and  $N_+ = \max\{J \in \mathcal{I} : S_i > 0, \forall i = 1, \dots, J\}$ .

### Indices of disproportionality

- Sainte-Laguë index:  $\sum_{i \in \mathcal{I}} \frac{(V_i - S_i)^2}{V_i}$ .
- Rae index (Rae 1971):  $R(N) = \frac{1}{N} \sum_{i=1}^N |V_i - S_i|$ .
- Modified Rae index (Lijphart 1994):  $R(N_L)$ .
- Loosemore & Hanby index (Loosemore and Hanby 1971):  $LH = \frac{1}{2} \sum_{i \in \mathcal{I}} |V_i - S_i|$ .
- Mackie & Rose index (Mackie and Rose 1982, 1991):  $100 - LH$ .
- Grofman indices (Taagepera and Grofman 2003):  $\frac{1}{N_*} \sum_{i \in \mathcal{I}} |V_i - S_i|$ .
- Largest deviation index (Lijphart 1994):  $\max\{|V_i - S_i| : i \in \mathcal{I}\}$ .
- Least squares index (Gallagher 1991):  $G(N) = \sqrt{\frac{1}{2} \sum_{i=1}^N (V_i - S_i)^2}$ .
- Modified least squares index (Lijphart 1994):  $G(N_L)$ .
- Robust  $L_1$ -norm based index (Pennisi 1998):  $\sum_{i \in \mathcal{I}} \left| \frac{S_i}{V_i} - 1 \right|$ .
- Robust  $L_2$ -norm based index (Pennisi 1998):  $\sum_{i \in \mathcal{I}} \left( \frac{S_i}{V_i} - 1 \right)^2$ .
- Robust  $L_\infty$ -norm based index (Pennisi 1998):  $\max\left\{ \left| \frac{S_i}{V_i} - 1 \right| : i \in \mathcal{I} \right\}$ .



*Bias in the electoral system*

Cox and Shugart (1996) describe the filtering associated to an electoral system through a simple linear regression model given by  $S_i = a + bV_i + \epsilon_i$ ,  $\forall i \in \mathcal{I}$ . Indeed these authors proposed to quantify the bias of an electoral system by the least squares (LS) estimate of the slope  $b$ . However, taking into account that the considered linear model is an excessive simplification in practice (the classic hypotheses on the model residuals may not be satisfied), some alternatives for the estimation of  $b$  were proposed in Oñate and Ocaña (1999). Some of such alternatives, which are based on EDA (Tukey 1977), propose to quantify the bias of the electoral system through the Tukey estimation (T) of  $b$ .

- Cox & Shugart (CS) index (Cox and Shugart 1996):  $\widehat{b}_{LS}(N)$ .
- Modified CS index (Lijphart 1994; Oñate and Ocaña 1999):  $\widehat{b}_{LS}(N_+)$ .
- End–modified CS index (Lijphart 1994; Oñate and Ocaña 1999):  $\widehat{b}_{LS}(N_S)$ .
- EDA version of CS index (Oñate and Ocaña 1999):  $\widehat{b}_T(N)$ .
- Modified EDA CS index (Oñate and Ocaña 1999):  $\widehat{b}_T(N_+)$ .
- End–modified EDA CS index (Oñate and Ocaña 1999):  $\widehat{b}_T(N_S)$ .
- 0.5%–modified EDA CS index (Oñate and Ocaña 1999):  $\widehat{b}_T(N_L)$ .

**Party system dimensions***Fragmentation, Concentration and Competitiveness*

- Fragmentation (Rae 1971):  $1 - \sum_{i \in \mathcal{I}} p_i^2$ .
- Concentration (Sartori 2005):  $p_1 + p_2$ .
- Competitiveness (Sartori 2005):  $p_1 - p_2$ .

*Effective number of parties (ENP)*

(Laakso and Taagepera 1979; Kesselman 1966; Theil 1969; Wildgen 1971; Molinar 1991)

- Laakso & Taagepera indices:  $\mathcal{N}_* = (\sum_{i \in \mathcal{I}} p_i^2)^{-1}$ .
- Kesselman & Wildgen indices:  $\exp(-\sum_{i \in \mathcal{I}} p_i \ln(p_i))$ .
- Molinar indices:  $1 + \mathcal{N}_*^2 \sum_{j=2}^N p_j^2$ .

*Polarization*

- Sartori index (Sartori 2005; Sani and Sartori 1983):  $\text{Range}(\text{ideological scores})$ .

- Weighted polarization (Hazan 1997): Variance(ideological scores).

### *Volatility*

(Arian and Weiss 1969; Pedersen 1979; Bartolini and Mair 2007; Ocaña 2007)

The volatility indices depend upon two elections held in two dates, which are denoted by the superscripts  $[t]$  and  $[t + 1]$ .

- Total volatility:  $TV = 1/2 \sum_i |p_i^{[t]} - p_i^{[t+1]}|$ .
- Bloc volatility:  $BV = TV$  (for the blocs of parties).
- Intra-bloc volatility:  $TV - BV$ .
- Arian & Weiss index:  $\frac{1}{N} \sum_i (p_i^{[t]} - p_i^{[t+1]})^2$ .

### Data with levels of aggregation

**IndElec** can work when electoral data present several levels of aggregation, such as is described in Section 3. In this sense, not only all the aforementioned indices can be re-calculated, but also new indices can be considered.

On the one hand, let  $\ell$  be an aggregation level and  $R_j^\ell$  be one of its *geographic units*, i.e.,  $R_j^\ell \in \mathcal{F}^\ell$ . The electoral data obtained by the parties  $\mathcal{I}$  in the *geographic unit*  $R_j^\ell$  can be summarized by its corresponding joint distribution of votes and seats, expressed in percentages, denoted by  $\{(V_i(R_j^\ell), S_i(R_j^\ell)) : i \in \mathcal{I}\}$ . It follows that the aforementioned indices can be computed for each *unit*  $R_j^\ell$ .

On the other hand, let  $\ell_U < \ell_L$  be any two aggregation levels and  $R_{j_U}^{\ell_U}$  be a *geographic unit* of the aggregation level  $\ell_U$ , i.e.,  $R_{j_U}^{\ell_U} \in \mathcal{F}^{\ell_U}$ . In this framework, we can consider those *geographic units* of the aggregation level  $\ell_L$  which are contained in  $R_{j_U}^{\ell_U}$ , i.e.,

$$\mathcal{F}^{\ell_L}(R_{j_U}^{\ell_U}) = \{R_j^{\ell_L} \in \mathcal{F}^{\ell_L} : R_j^{\ell_L} \subseteq R_{j_U}^{\ell_U}\}.$$

Further, let  $\mathcal{I}^R$  be the set of regionalist parties (they are labeled as PANE in **IndElec**), where  $\mathcal{I}^R \subset \mathcal{I}$ . The following indices are only computed by **IndElec** when the electoral data presents several aggregation levels.

### *Regionalism*

(Oñate and Ocaña 1999)

- Regionalist vote at the upper level item:  $VR(R_{j_U}^{\ell_U}) = \sum_{i \in \mathcal{I}^R} V_i(R_{j_U}^{\ell_U})$ .
- Regionalist vote at a lower level item:  $VR(R_j^{\ell_L})$ .
- Differentiated regionalist vote:  $VR(R_j^{\ell_L}) - VR(R_{j_U}^{\ell_U})$ .
- Differentiated regional vote:  $\frac{1}{2} \sum_{i \in \mathcal{I}} |V_i(R_{j_U}^{\ell_U}) - V_i(R_j^{\ell_L})|$ .

- Modified differentiated regional vote:  $\frac{1}{2} \sum_{i \in \mathcal{I}} |\bar{V}_i(\ell_L; R_{jU}^{\ell_U}) - V_i(R_j^{\ell_L})|$ .

where  $\bar{V}_i(\ell_L; R_{jU}^{\ell_U}) = \text{average} \{V_i(R_j^{\ell_L}) : R_j^{\ell_L} \in \mathcal{F}^{\ell_L}(R_{jU}^{\ell_U})\}$ ,  $\forall i \in \mathcal{I}$ .

### Party linkage

(Chhibber and Kollman 1998; Cox 1999; Moenius and Kasuya 2004)

- ENP at the upper level item:  $\mathcal{N}_e(R_{jU}^{\ell_U})$ .
- Average ENP in the lower aggregation level:  $\bar{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})$ .
- Cox inflation rate:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \bar{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})) / \mathcal{N}_e(R_{jU}^{\ell_U})$ .
- Moenius & Kasuya inflation rate:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \bar{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})) / \bar{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})$ .
- Weighted average ENP in the lower aggregation level:  $\tilde{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})$ .
- Weighted Cox inflation rate:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \tilde{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})) / \mathcal{N}_e(R_{jU}^{\ell_U})$ .
- Weighted Moenius & Kasuya inflation rate:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \tilde{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})) / \tilde{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U})$ .
- Chhibber & Kollman inflation measure:  $\mathcal{N}_e(R_{jU}^{\ell_U}) - \mathcal{N}_e(R_j^{\ell_L})$ .
- Cox inflation measure:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \mathcal{N}_e(R_j^{\ell_L})) / \mathcal{N}_e(R_{jU}^{\ell_U})$ .
- Moenius & Kasuya inflation measure:  $(\mathcal{N}_e(R_{jU}^{\ell_U}) - \mathcal{N}_e(R_j^{\ell_L})) / \mathcal{N}_e(R_j^{\ell_L})$ .

where  $\mathcal{N}_e(R_j^{\ell})$  is the effective number of electoral parties at  $R_j^{\ell}$ ,  $\bar{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U}) = \text{average}\{\mathcal{N}_e(R_j^{\ell_L}) : R_j^{\ell_L} \in \mathcal{F}^{\ell_L}(R_{jU}^{\ell_U})\}$  and  $\tilde{\mathcal{N}}_e(\ell_L; R_{jU}^{\ell_U}) = \text{average}\{\mathcal{N}_e(R_j^{\ell_L}) : R_j^{\ell_L} \in \mathcal{F}^{\ell_L}(R_{jU}^{\ell_U}); \text{weights} = \text{percentages of vote}\}$ .

### Affiliation:

Francisco Antonio Ocaña  
 Department of Statistics and Operations Research  
 University of Granada  
 18071 Granada, Spain  
 E-mail: [focana@ugr.es](mailto:focana@ugr.es)  
 URL: <http://www.ugr.es/~focana/>

Pablo Oñate  
Department of Constitutional Law and Political Science  
University of Valencia  
46071 Valencia, Spain  
E-mail: [Pablo.Onate@uv.es](mailto:Pablo.Onate@uv.es)