

# Simulatore di sistemi elettorali

Davide Maioli

# Introduzione

Il progetto di ingegneria informatica che ho sviluppato riguarda i sistemi elettorali, nello specifico la simulazione della **legge Mattarella**. Ho continuato il lavoro già avviato da un altro studente, che aveva precedentemente creato un framework con il quale è possibile simulare il funzionamento di leggi di questo genere.

Un simulatore di sistemi elettorali può essere utile per molteplici motivi. Innanzitutto può essere un buon strumento per scovare casi particolari non trattati per errore dai legislatori. In più, rende facile studiare come le caratteristiche di queste leggi si riflettono sulla composizione degli organi di governo. Vedremo un esempio di questo nella terzultima sezione, dedicata alla presentazione dei risultati.

Come già accennato, il principale obiettivo di questo progetto era estendere il framework già in essere, aggiungendo la legge sopra citata. Ho iniziato studiando in parallelo il funzionamento del simulatore e dei sistemi elettorali in generale, per poi concentrarmi sulla configurazione del primo e sulla legge Mattarella tra i secondi. Successivamente, ho proceduto con l'implementazione dell'estensione, che ho testato con i dati relativi ai risultati dell'elezione politica del 2001, l'ultima in cui si usò la legge Mattarella, dopo averli acquisiti e integrati. Ho concluso creando alcune presentazioni grafiche dei risultati ottenuti.

## Il simulatore

### Funzionamento

Il simulatore è stato sviluppato in Python seguendo il paradigma di programmazione ad oggetti, usando quindi le classi, messe a disposizione dal linguaggio. La simulazione avviene facendo interagire tra loro due tipi di oggetti: entità geografiche (GeoEnt), il cui obiettivo è quello di determinare il numero di voti ottenuti e il numero di seggi assegnati, e entità politiche (PolEnt) rappresentanti partiti, candidati e coalizioni, che si occupano invece di determinare chi sarà eletto realmente.

Un percorso che parte dal raccoglimento dei voti dell'elezione, passa dall'elaborazione dei questi dati da parte di uno o più *nodi* (solitamente entità geografiche) e si conclude con la presentazione di distribuzioni di seggi si chiama *lane*. Il primo nodo che genera la distribuzione dei seggi si chiama *lane head*, l'ultimo si chiama *lane tail*. In un caso tipico, le entità geografiche di livello più basso raccolgono i dati, poi li trasmettono ai livelli superiori che man mano li elaborano. In seguito i livelli superiori creano delle distribuzioni dei seggi che poi inoltrano ai livelli inferiori, che ricavano le loro distribuzioni locali. In qualunque fase del percorso le entità politiche possono effettuare operazioni di filtro.

## Configurazione

L'estensione del framework si effettua scrivendo opportuni file Python e Yaml. Oltre a creare un file Python in `src/Commons` con funzioni utili ai calcoli che la legge effettuerà, per ogni legge elettorale è necessario creare e popolare le seguenti cartelle:

- **Classes:** conterrà un file (Python o Yaml) per ogni classe che vogliamo definire. Per definire una classe possiamo farla ereditare da diverse metaclassi già disponibili. Per esempio, possiamo far ereditare da **lanes** se vogliamo che la classe faccia parte di una lane, da **superdivision** se è un'entità geografica con dei sottolivelli o da **totals** per permetterle di avere delle funzioni facilmente configurabili che operano su dataframe.
- **Instances:** conterrà un file Yaml per ogni classe definita in Classes. In questo file verranno definite le istanze della classe e alcuni dei loro attributi.
- **Data:** conterrà delle sottocartelle, una per ogni classe che deve caricare dati da file csv. Questi file verranno posizionati all'interno delle sottocartelle e il loro nome dovrà corrispondere ad un attributo della classe.

Per una descrizione più esaustiva di tutte le metaclassi e delle modalità di configurazione si rimanda alla documentazione originale del framework.

## La legge Mattarella

La legge Mattarella, nota anche come Mattarellum, è una legge elettorale della Repubblica Italiana che entrò in vigore nel 1994 e che ci rimase fino al 2005, quando venne abrogata. Fu usata per regolare le elezioni politiche del 1994, 1996 e 2001. Questa legge, che introdusse per la prima volta un sistema misto, prevedeva un comportamento diverso tra Camera e Senato nella parte proporzionale. Essendo la versione della Camera molto più complessa e interessante, mi sono concentrato su quest'ultima.

**Parte maggioritaria** I tre quarti dei seggi della Camera, cioè 475 su 630, venivano assegnati tramite un sistema maggioritario. Il territorio nazionale era diviso in **475 collegi uninominali**, ognuno dei quali assegnava un seggio al **candidato con più voti**. I candidati potevano presentarsi con un simbolo di un partito o di una coalizione e partiti diversi potevano presentarsi da soli in alcuni collegi e in una coalizione in altri. Inoltre, ogni candidato doveva essere *collegato* ad almeno un partito della parte proporzionale.

**Parte proporzionale** I restanti 155 seggi venivano assegnati tramite un sistema proporzionale. Il territorio nazionale era diviso in **26 circoscrizioni plurinominali**, dove ogni partito presentava delle liste bloccate. Di conseguenza, ogni elettore votava due schede per la camera, una per il proprio collegio uninominale e una per la propria circoscrizione plurinominali. La **cifra elettorale circoscrizionale** di ogni partito veniva determinata sommando i voti ottenuti nella circoscrizione e sottraendo a questo valore

la somma dei voti minimi necessari a vincere i collegi di quella circoscrizione vinti da candidati collegati al partito (questa operazione di sottrazione viene chiamata *scorporo*). La **cifra elettorale nazionale** di ogni partito veniva ottenuta sommando le cifre circoscrizionali. Ogni partito veniva sottoposto ad una **soglia di sbarramento del 4%** basata sul numero di voti, poi il numero di seggi per partito a livello nazionale veniva calcolato usando il **metodo Hare** dei quozienti e dei più alti resti. Infine, esisteva un meccanismo abbastanza complesso per trovare le distribuzioni dei seggi a livello circoscrizionale a partire da quella nazionale.

## Implementazione della legge

Di seguito presento come ho progettato la mia soluzione per l'estensione del framework. Per le ipotesi che ho dovuto adottate si rimanda alla sezione sull'acquisizione dei dati.

Ho deciso di utilizzare quattro classi:

- **Collegio**: per rappresentare i collegi uninominali
- **Circoscrizione**: per rappresentare le circoscrizioni plurinominali
- **Nazione**: per rappresentare l'Italia, quindi ne esisterà un'unica istanza
- **Partito**: per rappresentare i partiti politici

e due lane:

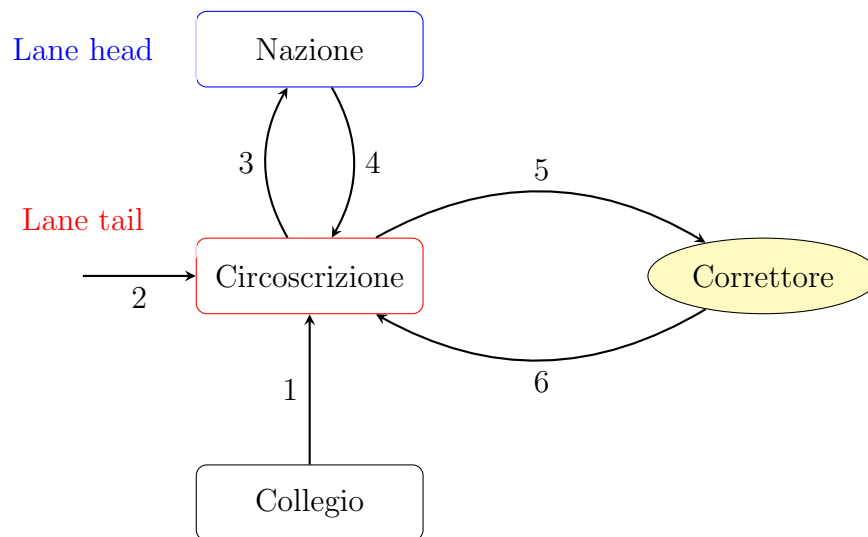
- **uninominal**: che si conclude con l'assegnazione dei 475 seggi della parte maggioritaria del sistema
- **plurinominale** che si conclude con l'assegnazione dei 155 seggi della parte proporzionale del sistema

La lane uninominale è composta da un solo nodo, che è la classe Collegio. Ogni Collegio procede alla seguenti operazioni:

1. Carica i propri voti dal file csv
2. Individua il candidato con più voti
3. Presenta come distribuzione locali dei seggi, composta da un solo partito che riceve l'unico seggio del collegio

La lane plurinomiale è più complessa e coinvolge tutte le classi. E' costituita dai seguenti passi:

1. Ogni Collegio determina il partito del candidato vincente e il numero di seggi necessari a vincere
2. Ogni Circoscrizione carica i propri voti da file csv
3. Ogni Circoscrizione aggrega i dati del passo 1 dei propri collegi e, unendoli ai voti caricati nel passo 2, determina i voti e le cifre elettorali circoscrizionali
4. La Nazione aggrega i dati del passo 3 di ogni Circoscrizione, applica a ogni Partito la soglia di sbarramento del 4% e poi determina la distribuzione nazionale dei seggi.
5. Ogni Circoscrizione riceve la distribuzione nazionale dei seggi e la usa per assegnare i seggi di cui è sicura, ottenendo così una distribuzione locale
6. Tutte le distribuzioni locali vengono corrette tramite una apposita funzione in src/Commons, che ridistribuisce gli ultimi seggi rimasti o dovuti.



## Acquisizione dei dati

### Implementazione del web scraper

Per simulare un'elezione realmente avvenuta con il Mattarellum, necessitavo di risultati tabulari utilizzabili in formato csv. Purtroppo il portale Open Data del Ministero dell'Interno non ha dati su elezioni politiche precedenti al 2013, quindi per poter simulare l'elezione del 2001 ho dovuto programmare un web scraper che andasse ad visitare l'archivio del Ministero dell'Interno. Per farlo ho scelto sempre Python e ho utilizzato la libreria BeautifulSoup4, che permette facilmente di creare oggetti rappresentanti il DOM e che espongono API comode per cercare e indagare il documento.

## Integrazione dei dati

I dati dell'elezione del 2001 presenti nell'archivio sono però incompleti, in quanto non riportano i collegamenti tra candidati uninominali e partiti, necessari per effettuare lo scorporo. Pertanto, sono andato a completarli, sotto due ipotesi:

1. I candidati uninominali sono collegati ad un solo partito della parte proporzionale.
2. I candidati uninominali appartenenti ad un partito sono collegati allo stesso partito nella parte proporzionale; i candidati uninominali appartenenti ad una coalizione sono collegati casualmente ad un partito della coalizione.

Inoltre, le probabilità che un partito sia scelto come collegamento per un candidato appartenente ad una coalizione sono dimensionate in base alla percentuale relativa di voti che quel partito ha ottenuto nella parte proporzionale, rispetto agli altri partiti della coalizione. In questo modo, i partiti più grossi di una coalizione avranno più candidati uninominali rispetto a quelli piccoli.

## Presentazione dei risultati

### Risultati senza liste civetta

Qui di seguito è mostrata la composizione della Camera in seguito alla mia simulazione delle elezioni del 2001. Praticamente tutti i partiti che hanno ottenuto dei seggi erano membri di una delle seguenti due coalizioni della parte uninominale:

- **Casa delle Libertà**, composta da Forza Italia, Alleanza Nazionale, Lega Nord, CCD-CDU, Nuovo PSI e Partito Repubblicano Italiano.
- **L'Ulivo**, composta da Democratici di Sinistra, La Margherita, Il Girasole, Comunisti Italiani, Repubblicani Europei, SVP e Partito Sardo d'Azione.

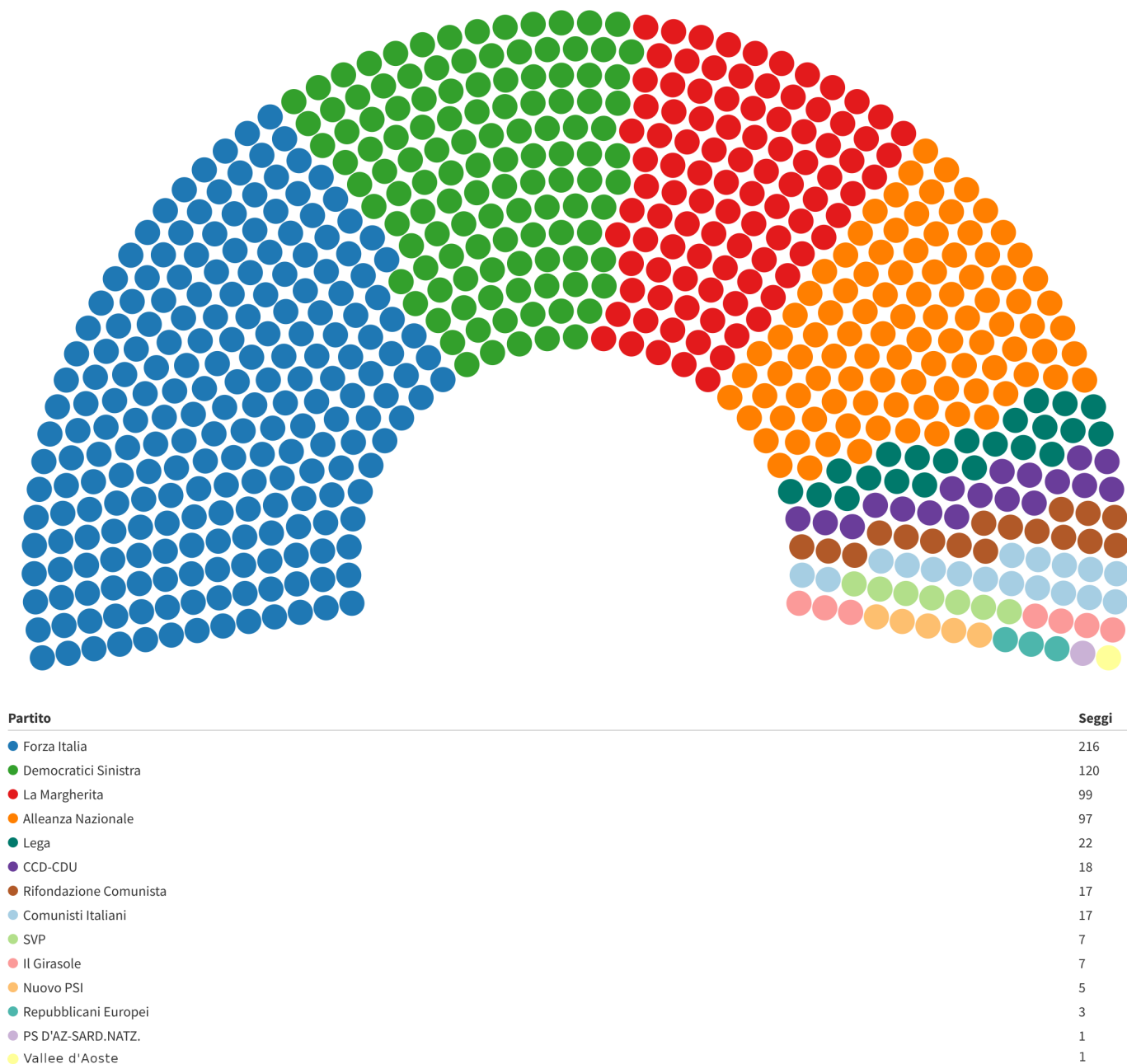


Figura 1: Composizione della Camera dei Deputati

## Risultati con liste civetta

In realtà nelle elezioni del 2001 le due coalizioni sopra citate trovarono un modo per "barare". Collegarono tutti i loro candidati uninominali non ai loro partiti di appartenenza, bensì a due *liste civetta*,

chiamate Abolizione Scorporo (per la Casa delle Libertà) e Paese Nuovo (per L'Ulivo), pensate per non essere votate, in modo che lo scorporo agisse su quest'ultime, senza togliere alcun voto ai partiti "veri".

Ho innanzitutto provato a simulare l'elezione considerando la presenza di entrambe le liste civetta e non ho riscontrato grosse differenze rispetto al caso senza.

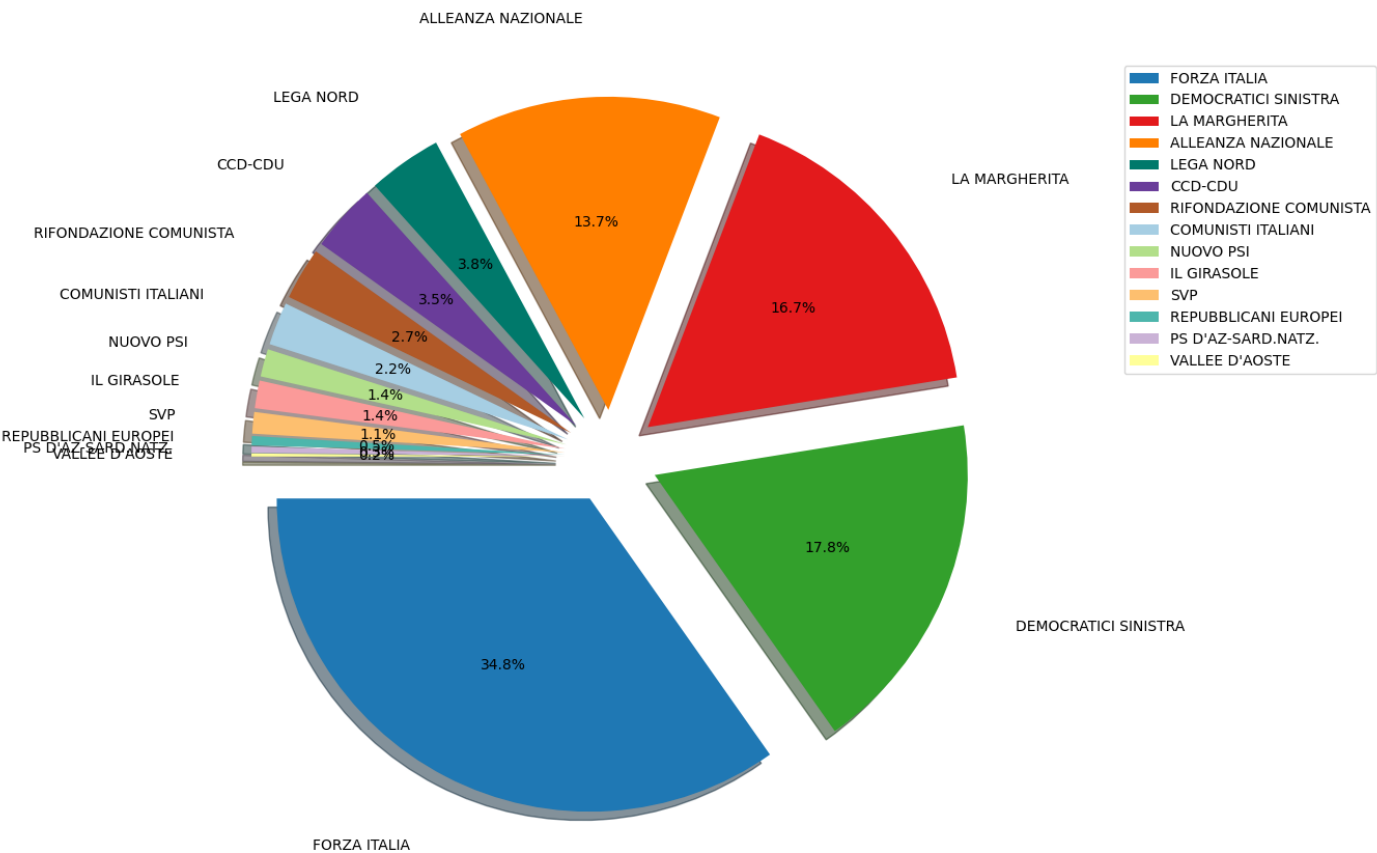


Figura 2: Distribuzione seggi con entrambe le liste civetta

Senza l'utilizzo di liste civetta, nella situazione illustrata dalla Figura 1, la coalizione Casa delle Libertà ottiene il 56.8% dei seggi, mentre L'Ulivo il 43%. Invece con l'utilizzo delle liste civetta, illustrato dalla Figura 2, la coalizione Casa delle Libertà ottiene il 57.2% dei seggi, mentre L'Ulivo il 42.6%. I risultati sono quindi molto simili.



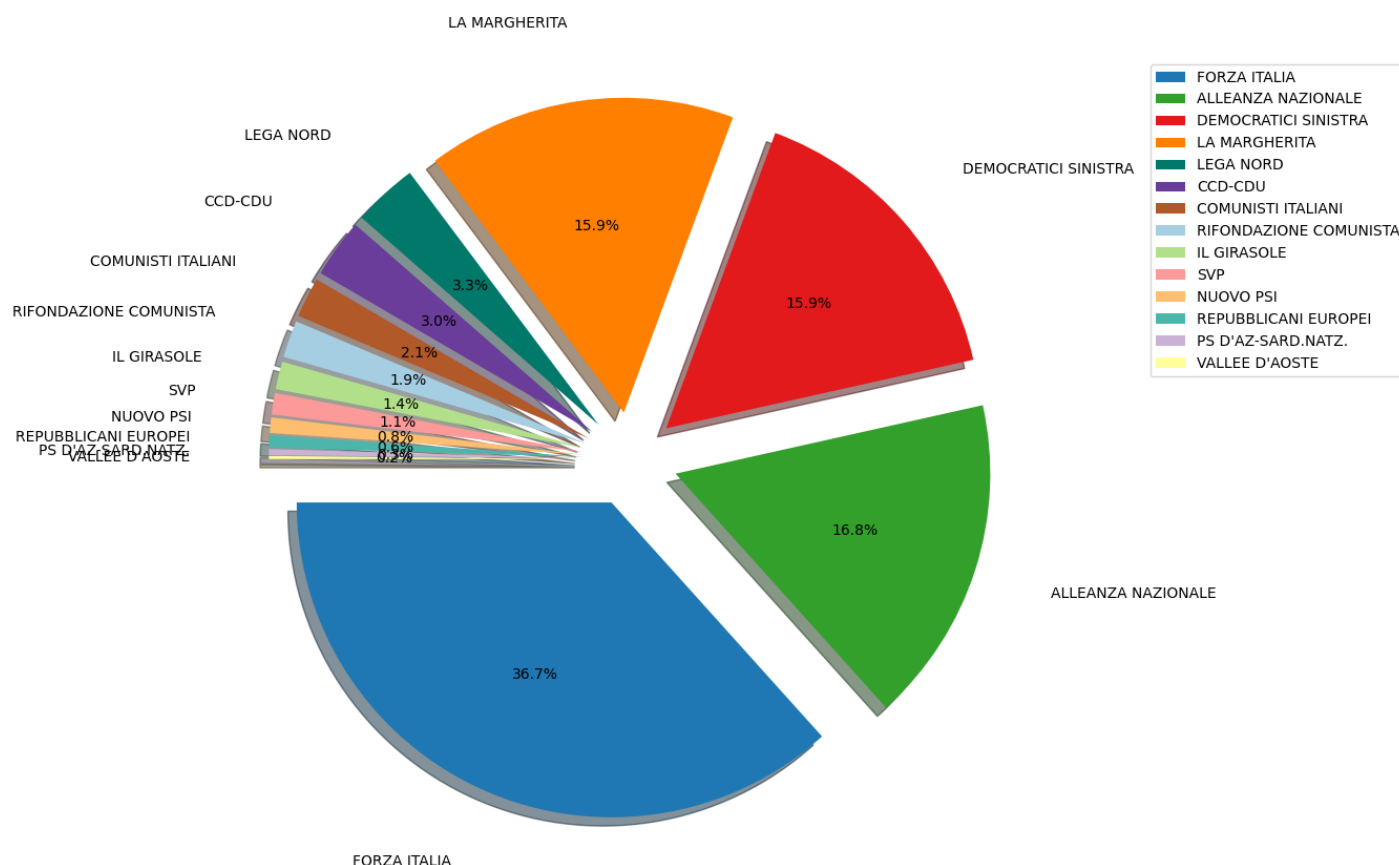


Figura 3: Distribuzione seggi con solamente la lista civetta Abolizione Scorpore

La situazione cambia molto provando a considerare il caso in cui solo una delle due coalizioni bari. Nel caso in Figura 3, si nota che introducendo la sola lista civetta Abolizione Scorpore, la Casa delle Libertà ottiene ben il 60.6% di voti, registrando un aumento del 3.4% rispetto all'assenza delle liste civetta. Al contrario, L'Ulivo si aggiudica il 39.1% dei seggi, con un calo del 3.5%. La differenza tra le due coalizioni passa dal 14.6% al 21.5%. Considerando solo la lista civetta Paese Nuovo si può osservare un fenomeno analogo. Risulta quindi che **la presenza di una sola lista civetta sbilancia la composizione della Camera in modo importante.**

## Utilizzo

Per utilizzare il framework è sufficiente accedere alla directory principale e, tramite una console python, digitare seguenti comandi:

```
>> import src
>> src.run_simulation('LeggiElettoralis/Mattarellum')
```

## Conclusione

Questo progetto è stato sia stimolante che impegnativo. Sicuramente studiare un ambito in cui non avevo alcuna competenza come la legge, anche andando a consultare i testi originali, è stato interessante. Allo stesso tempo, per poter utilizzare al meglio il framework ho dovuto fare un discreto lavoro di "reverse engineering", che è risultato più lungo del previsto.

Riguardo ai risultati ottenuti, ritengo che le problematiche evidenziate dalle analisi dell'utilizzo o meno delle liste civetta siano un esempio chiaro di come un simulatore di questo tipo possa essere utile in situazioni reali. Con il simulatore è stato facile mostrare che le liste civetta sfruttano una problematica del Mattarellum che influenza molto il risultato dell'elezione, ma questo fatto non era emerso nelle elezioni del 2001 perché **entrambe le coalizioni** avevano "barato", annullando lo sbilanciamento.