Package ‘powerindexR’

February 8, 2024

Type Package
Title Measuring the Power in Voting Systems
Version 1.5
Date 2024-02-08
Depends R(>= 3.0.0)
Description This R package allows the determination of some distributions of
the voters’ power when passing laws in weighted voting situations.
License GPL-2
LazyLoad yes
Repository CRAN
NeedsCompilation no
RoxygenNote 7.0.2
Author Livino M. Armijos-Toro [aut, cre],
Jose M. Alonso-Meijide [aut],
Manuel A. Mosquera [aut],
Alejandro Saavedra-Nieves [aut]
Maintainer Livino M. Armijos-Toro <livinoa@gmail.com>
Date/Publication 2024-02-08 18:10:02 UTC

R topics documented:

powerindexR-package ........................................... 2
MWC .............................................................. 3
pi.banzhaf ....................................................... 4
pi.colomermartinez ........................................... 5
pi.johnston ....................................................... 6
pi.johnstoncolomermartinez ................................. 7
pi.shapley ....................................................... 8
powerindex ...................................................... 9
QMWC ............................................................ 10

Index 12
Description

This R package allows the determination of some distributions of the voters’ power when passing laws in weighted voting situations.

Details

The DESCRIPTION file:

Package: powerindexR
Type: Package
Title: Measuring the Power in Voting Systems
Version: 1.5
Date: 2024-02-08
Authors@R: c(person("Livino M.", "Armijos-Toro", role = c("aut","cre"), email="livinoa@gmail.com"), person("Jose M.", "Alonso-Meijide", role = c("aut")), person("Manuel A.", "Mosquera", role = c("aut")), person("Alejandro", "Saavedra-Nieves", role = c("aut")))
Depends: R(>= 3.0.0)
Description: This R package allows the determination of some distributions of the voters’ power when passing laws in weighted voting situations.
License: GPL-2
LazyLoad: yes
Packaged: CRAN
Repository: CRAN
NeedsCompilation: no
RoxygenNote: 7.0.2
Author: Livino M. Armijos-Toro [aut, cre], Jose M. Alonso-Meijide [aut], Manuel A. Mosquera [aut], Alejandro Saavedra-Nieves [aut]
Maintainer: Livino M. Armijos-Toro <livinoa@gmail.com>

Index of help topics:

MWC Obtain the minimal winning coalitions
QMWC Obtain the quasi-minimal winning coalitions
pi.banzhaf Power based on the Banzhaf index.
pi.colomermartinez Power based on the Colomer-Martinez index.
pi.johnston Power based on the Johnston index.
pi.johnstoncolomermartinez Power based on the Johnston-Colomer-Martinez index.
pi.shapley Power based on the Shapley-Shubik index.
powerindex Obtain several measures of power
powerindexR-package Measuring the Power in Voting Systems

This R package allows the determination of some distributions of the voters’ power when passing laws in weighted voting situations.
**MWC**

*Obtain the minimal winning coalitions*

**Description**

This function determines the minimal winning coalitions in a weighted majority game.

**Usage**

```r
MWC(quota, weights)
```

**Arguments**

- `quota` Numerical value that represents the majority in a given voting.
- `weights` Numerical vector of dimension `n` that indicates the weights of `n` agents in a given voting.

**Value**

- **Number of Minimal Winning Coalitions**
  - Total amount of Minimal Winning Coalitions.
- **Minimal Winning Coalitions**
  - Each row indicates a binary representation of each Minimal Winning Coalition.

**Author(s)**

Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

**References**


Examples

```r
weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
MWC(quota,weights)
```

---

**pi.banzhaf**  
*Power based on the Banzhaf index.*

---

**Description**

This function determines the distribution of the power based on the Banzhaf index and the Banzaf-Owen value.

**Usage**

```r
pi.banzhaf(quota, weights, partition = NULL, normalized = FALSE, swing = FALSE)
```

**Arguments**

- **quota**: Numerical value that represents the majority in a given voting.
- **weights**: Numerical vector of dimension $n$ that indicates the weights of $n$ agents in a given voting.
- **partition**: Numerical vector that indicates the partition of voters. Each component indicates the element of the partition to which such voter belongs. If it is not `NULL`, it provides the distribution of the power based on the Banzhaf-Owen value.
- **normalized**: Logical option to obtain the normalized Banzhaf values.
- **swing**: Logical option to obtain the number of swings of each voter.

**Value**

- **Banzhaf value**: The Banzhaf value, if `partition=NULL`.
- **Banzhaf-Owen value**: The Banzhaf-Owen value, if `partition!=NULL`.

**Author(s)**

Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

**References**


Examples

# Example Banzhaf value
weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
pi.banzhaf(quota,weights)
pi.banzhaf(quota,weights,normalized=TRUE)

# Example Banzhaf-Owen value
quota<-30
weights<-c(28, 16, 5, 4, 3, 3)
# Partition={{1},{2,4,6},{3,5}}
pi.banzhaf(quota,weights,partition=c(1,2,3,2,3,2))

pi.colomermartinez

Power based on the Colomer-Martinez index.

Description

This function determines the distribution of the power based on the Colomer-Martinez index.

Usage

pi.colomermartinez(quote, weights, minimal = FALSE)

Arguments

quota Numerical value that represents the majority in a given voting.
weights Numerical vector of dimension n that indicates the weights of n agents in a given voting.
minimal Logical option to obtain the Minimal Winning Coalitions.

Value

Colomer-Martinez
The Colomer-Martinez index.
Number of Minimal Winning Coalitions
Total amount of Minimal Winning Coalitions.
Minimal Winning Coalitions
Each row indicates a binary representation of each Minimal Winning Coalition.

Author(s)

Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

References

Examples
weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
pi.colomermartinez(176,weights,minimal=TRUE)

pi.johnston

Power based on the Johnston index.

Description
This function determines the distribution of the power based on the Johnston index.

Usage
pi.johnston(quota, weights, quasiminimal = FALSE)

Arguments
quota Numerical value that represents the majority in a given voting.
weights Numerical vector of dimension $n$ that indicates the weights of $n$ agents in a given voting.
quasiminimal Logical option to obtain the Quasi-Minimal Winning Coalitions.

Value
Johnston The Johnston index.
Number of Quasi-Minimal Winning Coalitions Total amount of Quasi-Minimal Winning Coalitions.
Quasi-Minimal Winning Coalitions Each row indicates a binary representation of each Quasi-Minimal Winning Coalition.

Author(s)
Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

References

Examples
weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
pi.johnston(176,weights,quasiminimal=TRUE)
pi.johnstoncolomermartinez

Power based on the Jonhston-Colomer-Martinez index.

Description

This function determines the distribution of the power based on the Jonhston-Colomer-Martinez index.

Usage

pi.johnstoncolomermartinez(quota, weights)

Arguments

quota Numerical value that represents the majority in a given voting.
weights Numerical vector of dimension $n$ that indicates the weights of $n$ agents in a given voting.

Value

Jonhston-Colomer-Martinez
The Jonhston-Colomer-Martinez index.

Author(s)

Livino M. Armijos-Toro, Jose M. Alonso-Mejide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

References


Examples

weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
pi.johnstoncolomermartinez(176,weights)
pi.shapley  

**Power based on the Shapley-Shubik index.**

**Description**

This function determines the distribution of the power based on the Shapley-Shubik index and the Owen value.

**Usage**

```
pi.shapley(quota, weights, partition = NULL)
```

**Arguments**

- **quota**: Numerical value that represents the majority in a given voting.
- **weights**: Numerical vector of dimension $n$ that indicates the weights of $n$ agents in a given voting.
- **partition**: Numerical vector that indicates the partition of voters. Each component indicates the element of the partition to which such voter belongs. If it is not NULL, it provides the distribution of the power based on the Owen value.

**Value**

- **Shapley value**: The Shapley value, if `partition` = NULL.
- **Owen value**: The Owen value, if `partition` != NULL.

**Author(s)**

Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

**References**


**Examples**

```r
# Example Shapley value
weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
pi.shapley(quota,weights)

# Example Owen value
quota<-30
```
weights<-c(28, 16, 5, 4, 3, 3)
# Partition=\{(1),(2,4,6),(3,5)\}
pl.shapley(quota,weights,partition=c(1,2,3,2,3,2))

\textbf{Obtain several measures of power}

**Description**

This general function allows the determination of several distributions of the power under different approaches in a weighted voting situation.

**Usage**

\begin{verbatim}
powerindex(quota, weights, index = c("S", "B", "J", "CM", "JCM"),
partition = NULL, quasiminimal = FALSE, minimal = FALSE, normalized = FALSE,
swing = FALSE)
\end{verbatim}

**Arguments**

- \texttt{quota} Numerical value that represents the majority in a given voting.
- \texttt{weights} Numerical vector of dimension \(n\) that indicates the weights of \(n\) agents in a given voting.
- \texttt{index} Character that indicates the used approach. \(S\) and \(B\) denote the Shapley-Shubik index and the Banzhaf index, and the Owen index and the Banzhaf-Owen index if \texttt{partition} exist. \(J\) is used for obtaining the Johnston index, \texttt{CM} determines the Colomer-Martinez index and \texttt{JCM} is used for obtaining the Johnston-Colomer-Martinez index.
- \texttt{partition} Numerical vector that indicates the partition of voters. Each component indicates the element of the partition to which such voter belongs.
- \texttt{quasiminimal} Logical option to obtain the Quasi-Minimal Winning Coalitions.
- \texttt{minimal} Logical option to obtain the Minimal Winning Coalitions.
- \texttt{normalized} Logical option to obtain the normalized Banzhaf values.
- \texttt{swing} Logical option to obtain the number of swings of each voter.

**Value**

See the values of the respective functions.

**Author(s)**

Livino M. Armijos-Toro, Jose M. Alonso-Meijide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.
References


Examples

weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
powerindex(quota,weights,index="S")
powerindex(quota,weights,index="B",swing=TRUE)
powerindex(quota,weights,index="B",partition=c(1,1,2,2,3,3,4,4,4),swing=TRUE)
powerindex(quota,weights,index="J",quasiminimal=TRUE)

QMWC

Obtain the quasi-minimal winning coalitions

Description

This function determines the quasi-minimal winning coalitions in a weighted majority game.

Usage

QMWC(quote, weights)

Arguments

quota Numerical value that represents the majority in a given voting.
weights Numerical vector of dimension $n$ that indicates the weights of $n$ agents in a given voting.

Value

Number of Quasi-Minimal Winning Coalitions
Total amount of Quasi-Minimal Winning Coalitions.
Quasi-Minimal Winning Coalitions
Each row indicates a binary representation of each Quasi-Minimal Winning Coalition.
Author(s)

Livino M. Armijos-Toro, Jose M. Alonso-Mejide, Manuel A. Mosquera, Alejandro Saavedra-Nieves.

Examples

weights<-c(137,85,71,32,9,8,5,2,1)
quota<-176
QMWC(quota,weights)
Index

* powerindex
  powerindexR-package, 2

MWC, 3

pi.banzhaf, 4
pi.colomermartinez, 5
pi.johnston, 6
pi.johnstoncolomermartinez, 7
pi.shapley, 8
powerindex, 9
powerindexR (powerindexR-package), 2
powerindexR-package, 2

QMWC, 10