

# Package ‘vote’

February 9, 2018

**Type** Package

**Title** Election Vote Counting

**Version** 1.1-0

**Date** 2018-02-08

**Author** Hana Sevcikova, Bernard Silverman, Adrian Raftery

**Maintainer** Hana Sevcikova<hanas@uw.edu>

**Description** Counting election votes and determining election results by different methods, including the single transferable vote, approval, score and plurality methods.

**Imports** formattable, knitr

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-02-09 04:04:53 UTC

## R topics documented:

vote-package . . . . .	2
approval . . . . .	3
count.votes . . . . .	4
food_election . . . . .	6
ims_election . . . . .	6
score . . . . .	7
stv . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

## Description

Counting election votes and determining election results by different methods, including the single transferable vote, approval, score and plurality methods.

## Details

The main function of the package is called `count.votes`. If no specific method is passed, it decides on the basis of the available data which method is the most appropriate. Specific methods can also be invoked explicitly. The following voting methods are available:

- `stv`: Single transferable vote (STV) where voters rank candidates in order.
- `score`: Range voting where each voter gives each candidate a score within a specific range.
- `approval`: Voters give each candidate one (approve) or zero (not approve).
- `plurality`: Each voter chooses one candidate.

Output of these functions can be viewed using summary methods, or in a browser using view methods. The summary methods return a data frame which can be stored in a file, see Example below. Functions `invalid.votes` and `valid.votes` can be used to check the validity of ballots for the various methods.

Example datasets are included. The `ims_election` dataset contains anonymized ballots from a past Council election of the Institute of Mathematical Statistics (IMS) which uses the STV method. Modifications of this dataset are available (`ims_approval`, `ims_score`, `ims_plurality`) as examples of data required by the various methods. The `food_election` dataset taken from Wikipedia can be used to test the STV method.

## Author(s)

Hana Sevcikova, Bernard Silverman, Adrian Raftery

Maintainer: Hana Sevcikova

## Examples

```
data(ims_election)
res <- count.votes(ims_election, method = "stv", mcan = 5)
summary(res)

# View invalid votes
invalid.votes(res)

## Not run:
# View results in a browser
view(res)
```

```
# Write election results into a csv file
s <- summary(res)
write.csv(s, "IMSstvresults.csv")
## End(Not run)
```

---

approval

*Approval and Plurality Vote Count*


---

## Description

Count votes using the approval and plurality method. Each voter can select candidates using 1 for a selection and 0 otherwise. In the approval method, any number of candidates can be selected by a voter, while in the plurality method only one candidate can be chosen by a voter. Thus, plurality voting is a special case of approval voting. The winner(s) in either method is/are the most-approved candidate(s).

## Usage

```
approval(votes, mcan = 1, fsep = "\t", ...)
```

```
## S3 method for class 'vote.approval'
summary(object, ...)
```

```
## S3 method for class 'vote.approval'
view(object, ...)
```

```
plurality(votes, mcan = 1, fsep = "\t", ...)
```

```
## S3 method for class 'vote.plurality'
summary(object, ...)
```

```
## S3 method for class 'vote.plurality'
view(object, ...)
```

## Arguments

votes	Matrix or data frame of zeros and ones containing the votes. Rows correspond to the votes, columns correspond to the candidates. If it is a character string it is interpreted as the name of the tab-separated file from which the votes are to be read. Missing values (NA) are interpreted as zeros.
mcan	Number of candidates to be elected.
fsep	If votes is a file name, this argument gives the column separator in the file.
...	Not used.
object	Object of class <code>vote.approval</code> or <code>vote.plurality</code> .

**Value**

Functions `approval` and `plurality` return an object of class `vote`. `approval` and `vote.plurality`, respectively, both of which are lists with the following objects:

<code>elected</code>	Vector of names of the elected candidates in the order in which they were elected.
<code>totals</code>	Vector of total votes in the same order as candidates (columns) in the input data.
<code>data</code>	Input data with invalid votes removed.
<code>invalid.votes</code>	Matrix of invalid votes that were removed from the original dataset.

**Author(s)**

Hana Sevcikova, Adrian Raftery

**References**

[https://en.wikipedia.org/wiki/Approval\\_voting](https://en.wikipedia.org/wiki/Approval_voting)  
[https://en.wikipedia.org/wiki/Plurality\\_voting\\_method](https://en.wikipedia.org/wiki/Plurality_voting_method)

**See Also**

[count.votes](#)

**Examples**

```
# Example using the IMS Council dataset modified for approval voting
data(ims_approval)
approval(ims_approval)

# Example using the IMS Council dataset modified for plurality voting
data(ims_plurality)
pl.ims <- plurality(ims_plurality)
invalid.votes(pl.ims)
```

---

count.votes

*Count Votes*

---

**Description**

Count votes using one of four methods. View valid and invalid ballots.

**Usage**

```
count.votes(votes, method = c("auto", "plurality", "approval", "stv", "score"),
  fsep = "\t", ...)
```

```
invalid.votes(object)
```

```
valid.votes(object)
```

**Arguments**

votes	Matrix or data frame containing the votes. Rows correspond to the votes, columns correspond to the candidates. If it is a character string it is interpreted as a tab-separated file name from which the votes are to be read.
method	Voting method to use. If “auto”, the input data is passed through a checker for each of the methods and the one with the largest number of valid votes is used. In case of the same number of valid votes, it goes by their ordering in the function definition.
fsep	If votes is a file name, this argument gives the column separator in the file.
...	Additional arguments passed to the underlying functions, e.g. <code>mcan</code> , <code>max.score</code> etc.
object	Object returned by one of the functions <a href="#">plurality</a> , <a href="#">approval</a> , <a href="#">stv</a> , <a href="#">score</a> .

**Value**

Depending which method is used, `count.votes` returns an object of class [vote.plurality](#), [vote.approval](#), [vote.stv](#), or [vote.score](#).

Functions `valid.votes` and `invalid.votes` return a subset of the input data with valid records and invalid records, respectively.

**Author(s)**

Hana Sevcikova, Bernard Silverman

**See Also**

[stv](#), [approval](#), [score](#)

**Examples**

```
# Example using the IMS Council dataset modified for score voting
data(ims_score)
# should recognize that it is a dataset with score voting data
count.votes(ims_score, max.score = 9, larger.wins = FALSE)

# All records with score larger than 8 are excluded
res <- count.votes(ims_score, method = "score", max.score = 8)
head(invalid.votes(res))

summary(res)
```

---

food_election	<i>Example Dataset</i>
---------------	------------------------

---

**Description**

Dataset on food election which serves as a simple example for the STV method taken from Wikipedia.

**Usage**

```
data("food_election")
```

**Format**

A data frame with 20 observations and 5 candidates (Oranges, Pears, Chocolate, Strawberries, Sweets). Each record corresponds to one ballot with ranking for each of the candidates.

**Source**

[https://en.wikipedia.org/wiki/Single\\_transferable\\_vote#Example](https://en.wikipedia.org/wiki/Single_transferable_vote#Example)

**Examples**

```
data(food_election)
head(food_election)
```

---

ims_election	<i>Datasets on IMS Election</i>
--------------	---------------------------------

---

**Description**

Datasets containing anonymized votes for a past Council election of the Institute of Mathematical Statistics (IMS). The dataset `ims_election` (named also `ims_stv`) is the original dataset used with single transferable vote, where candidate names have been changed. Each of the other datasets is a modified version of the original data to be used as an example for each of the other voting methods.

**Usage**

```
data("ims_election")
data("ims_stv")

data("ims_approval")
data("ims_score")
data("ims_plurality")
```

**Format**

A data frame with 620 observations and 10 candidates (names were made up). Each record corresponds to one ballot. Values depend on the voting method. The IMS Council voting is done using the STV method, and thus the `ims_election` dataset contains ballots with candidates being ranked between 1 and 10 with zeros allowed.

**Source**

The original dataset (which was randomized and anonymized, with write-in votes removed) was obtained from the the Institute of Mathematical Statistics.

**References**

<http://imstat.org/elections/stvprocess.htm>

**Examples**

```
data(ims_election)
head(ims_election)
```

---

score	<i>Score Vote Count</i>
-------	-------------------------

---

**Description**

Count votes using the score (or range) method. Voters give each candidate a score, the scores are added and the candidate(s) with the highest (or lowest) totals is/are elected.

**Usage**

```
score(votes, mcan = 1, max.score = NULL, larger.wins = TRUE, fsep = "\t", ...)

## S3 method for class 'vote.score'
summary(object, ...)

## S3 method for class 'vote.score'
view(object, ...)
```

**Arguments**

votes	Matrix or data frame containing the votes which should be numbers between 0 and <code>max.score</code> . Rows correspond to the votes, columns correspond to the candidates. If it is a character string it is interpreted as a tab-separated file name from which the votes are to be read. Missing values (NA) are interpreted as zeros.
mcan	Number of candidates to be elected.

<code>max.score</code>	Maximum score allowed. It is used to remove invalid votes. If not given, the maximum value contained in the data is taken and thus, all non-negative votes are valid.
<code>larger.wins</code>	Logical argument indicating whether the winners are the candidates with the highest scores (default) or the lowest scores.
<code>fsep</code>	If <code>votes</code> is a file name, this argument gives the column separator in the file.
<code>...</code>	Not used.
<code>object</code>	Object of class <code>vote.score</code> .

### Value

Function `score` returns an object of class `vote.score` which is a list with the following objects:

<code>elected</code>	Vector of names of the elected candidates in the order in which they were elected.
<code>totals</code>	Vector of total votes in the same order as candidates (columns) in the input data.
<code>larger.wins</code>	Input argument of the same name.
<code>data</code>	Input data with invalid votes removed.
<code>invalid.votes</code>	Number of invalid votes that were removed from the original dataset.

### Author(s)

Hana Sevcikova, Adrian Raftery

### References

[https://en.wikipedia.org/wiki/Range\\_voting](https://en.wikipedia.org/wiki/Range_voting)

### See Also

[count.votes](#)

### Examples

```
# Example using the IMS Council dataset modified for score voting
data(ims_score)
score.ims <- score(ims_score, max.score = 9)
summary(score.ims)
```



---

 stv

*Single Transferable Vote*


---

## Description

Count votes using the single transferable voting method.

## Usage

```
stv(votes, mcan = NULL, eps = 0.001, fsep = '\t', verbose = FALSE, ...)
```

```
## S3 method for class 'vote.stv'
summary(object, ...)
```

```
## S3 method for class 'vote.stv'
view(object, ...)
```

## Arguments

votes	Matrix or data frame containing the votes. Rows correspond to the votes, columns correspond to the candidates. If it is a character string it is interpreted as a tab-separated file name from which the votes are to be read. See below for more details.
mcan	Number of candidates to be elected. By default it is half the number of candidates standing.
eps	Value added to the quota. I.e. the STV quota is computed as $\text{number\_of\_first\_preferences} / (\text{number\_of\_seats} + 1) + \text{eps}$ .
fsep	If votes is a file name, this argument gives the column separator in the file.
verbose	Logical. If TRUE the progress of the count will be printed.
...	Not used.
object	Object of class <code>vote.stv</code> .

## Details

For a description of the single transferable vote system see <http://imstat.org/elections/stvprocess.htm>.

The input data votes is structured as follows: Row  $i$  contains the preferences of voter  $i$  numbered  $1, 2, \dots, r, 0, 0, 0, 0$ , in some order. The columns correspond to the candidates. The dimnames of the columns are the names of the candidates; if these are not supplied then the candidates are lettered A, B, C, .... If the dataset contains missing values (NA), they are replaced by zeros.

Ties are resolved using the forwards tie-breaking method, see Newland and Briton (Section 5.2.5).

**Value**

Function `stv` returns an object of class `vote.stv` which is a list with the following objects:

<code>elected</code>	Vector of names of the elected candidates in the order in which they were elected.
<code>preferences</code>	Matrix of preferences. Columns correspond to the candidates and rows to the counts (i.e. voting rounds).
<code>quotas</code>	Vector of quotas, one for each count.
<code>elect.elim</code>	Matrix of the same shape as preferences. Value 1 means that the corresponding candidate was elected in that round; value -1 means an elimination.
<code>data</code>	Input data with invalid votes removed.
<code>invalid.votes</code>	Matrix of invalid votes that were removed from the original dataset.

**Author(s)**

Bernard Silverman, Hana Sevcikova, Adrian Raftery

**References**

R.A. Newland and F.S. Britton (1997). How to conduct an election by the Single Transferable Vote. ERS 3rd Edition. <http://www.rosenstiel.co.uk/stvrules/index.html>  
<http://imstat.org/elections/stvprocess.htm>  
[https://en.wikipedia.org/wiki/Single\\_transferable\\_vote](https://en.wikipedia.org/wiki/Single_transferable_vote)

**Examples**

```
# Reproducing example from Wikipedia
# https://en.wikipedia.org/wiki/Single_transferable_vote#Example
# Uses eps=1
data(food_election)
stv.food <- stv(food_election, mcan = 3, eps = 1)
summary(stv.food)
## Not run:
view(stv.food)
## End(Not run)

# Example of the IMS Council voting
data(ims_election)
stv.ims <- stv(ims_election, mcan = 5)
## Not run:
view(stv.ims)

# write election results into a csv file
s <- summary(stv.ims)
write.csv(s, "myfile.csv")
## End(Not run)

# Example of a small committee dataset
# with four candidates (C) and four
```

```
# voting committee members (uses tie-breaking)
votes <- data.frame(C1=c(3,2,1,3), C2=c(2,1,2,4),
                   C3=c(4,3,3,1), C4=c(1,4,4,2))
stv(votes, mcan = 2, verbose = TRUE)
```

# Index

- \*Topic **datasets**
  - food\_election, 6
  - ims\_election, 6
- \*Topic **package**
  - vote-package, 2
- \*Topic **tools**
  - approval, 3
  - count.votes, 4
  - score, 7
  - stv, 9
- approval, 2, 3, 5
- count.votes, 2, 4, 4, 8
- food\_election, 2, 6
- ims\_approval, 2
- ims\_approval(ims\_election), 6
- ims\_election, 2, 6
- ims\_plurality, 2
- ims\_plurality(ims\_election), 6
- ims\_score, 2
- ims\_score(ims\_election), 6
- ims\_stv(ims\_election), 6
- invalid.votes, 2
- invalid.votes(count.votes), 4
- plurality, 2, 5
- plurality(approval), 3
- print.summary.vote.approval(approval), 3
- print.summary.vote.plurality(approval), 3
- print.summary.vote.score(score), 7
- print.summary.vote.stv(stv), 9
- score, 2, 5, 7
- stv, 2, 5, 9
- summary.vote.approval(approval), 3
- summary.vote.plurality(approval), 3
- summary.vote.score(score), 7
- summary.vote.stv(stv), 9
- valid.votes, 2
- valid.votes(count.votes), 4
- view(stv), 9
- view.vote.approval(approval), 3
- view.vote.plurality(approval), 3
- view.vote.score(score), 7
- vote(vote-package), 2
- vote-package, 2
- vote.approval, 5
- vote.approval(approval), 3
- vote.plurality, 5
- vote.plurality(approval), 3
- vote.score, 5
- vote.score(score), 7
- vote.stv, 5
- vote.stv(stv), 9