

# Package ‘rcv’

August 11, 2017

**Type** Package

**Title** Ranked Choice Voting

**Version** 0.2.1

**Description** A collection of ranked choice voting data and functions to manipulate, run elections with, and visualize this data and others. It can bring in raw data, transform it into a ballot you can read, and return election results for an RCV contest.

**URL** <https://github.com/ds-elections/rcv>

**BugReports** <https://github.com/ds-elections/rcv/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.3.3)

**Imports** dplyr, tibble, tidyr, readr, stringr

**RoxygenNote** 6.0.1

**Suggests** alluvial, networkD3, knitr, rmarkdown

**NeedsCompilation** no

**Author** Jay Lee [aut],  
Matthew Yancheff [aut, cre],  
Reed College [cph]

**Maintainer** Matthew Yancheff <[yanchefm@reed.edu](mailto:yanchefm@reed.edu)>

**Repository** CRAN

**Date/Publication** 2017-08-11 08:11:33 UTC

## R topics documented:

add_exhausted . . . . .	2
approval . . . . .	3
cambridge_ballot . . . . .	3

cambridge_clean . . . . .	4
cambridge_lookup . . . . .	4
characterize . . . . .	5
clean_ballot . . . . .	5
clean_mn . . . . .	6
elected . . . . .	7
import_data . . . . .	7
label . . . . .	8
make_alluvialdf . . . . .	8
make_d3list . . . . .	9
minneapolis_mayor_2013 . . . . .	10
rcv . . . . .	10
rcv_tally . . . . .	11
readable . . . . .	11
sf_7_results . . . . .	12
sf_bos_ballot . . . . .	12
sf_bos_clean . . . . .	13
sf_bos_lookup . . . . .	13
sf_lookup_labelled . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

add_exhausted	<i>Adds correct exhausted numbers for a 'rcv_tally' dataframe</i>
---------------	-------------------------------------------------------------------

---

## Description

Adds correct exhausted numbers for a 'rcv\_tally' dataframe

## Usage

```
add_exhausted(results)
```

## Arguments

results	A dataframe that comes from 'rcv_tally()'
---------	-------------------------------------------

## Value

The results dataframe with correct counts for the exhausted votes

---

approval	<i>Summarizes the approval rate of the eventual winner(s)</i>
----------	---------------------------------------------------------------

---

**Description**

Counts what proportion of voters approved of the election winners, defining "approved" as a voter listing the candidate on their ballot

**Usage**

```
approval(results, image, rcvcontest, n = 1)
```

**Arguments**

results	the tabulated election results
image	the clean ballot image
rcvcontest	(optional) The election to calculate results for. If the image contains more than one unique contest, this must be supplied.
n	the number of candidates being elected (defaults to 1)

**Value**

a numerical vector of length 1

**Examples**

```
approval(sf_7_results, sf_bos_clean,
         rcvcontest = "Board of Supervisors, District 7", n = 1)
```

---

cambridge_ballot	<i>Raw ballot image data from a Cambridge, MA RCV election</i>
------------------	----------------------------------------------------------------

---

**Description**

The .rda version of a .csv from a 2005 Cambridge, MA City Council multiwinner RCV election. This data is included as an example of another ballot image format for use with the label function. All Cambridge data is to be used with the "ChoicePlus" format.

**Usage**

```
cambridge_ballot
```

**Format**

A data frame with 17959 rows and 26 variables

---

cambridge_clean	<i>Cleaned ballot image data from a San Francisco RCV election</i>
-----------------	--------------------------------------------------------------------

---

### Description

A tidied version of 'cambridge\_ballot' in a "tall" format, readable to see voters' information, including candidate rankings, ward, and precinct. Cleaned with clean\_ballot. clean\_ballot(cambridge\_ballot, b\_header = T, cambridge\_lookup, l\_header = T, format = "ChoicePlus")

### Usage

```
cambridge_clean
```

### Format

A data frame with 108752 rows and 7 variables:

**pref\_voter\_id** a unique key identifying an individual voter

**ward** which city ward the voter voted in

**precinct** which city precinct the voter voted in

**style** which ballot style the voter had; different styles list different candidates first on the ballot to remove that advantage

**contest** which election this candidate ranking applies to

**vote\_rank** the rank given to the candidate by the voter

**candidate** the chosen candidate for the specified vote rank

---

cambridge_lookup	<i>Raw lookup data from a Cambridge, MA RCV election</i>
------------------	----------------------------------------------------------

---

### Description

The .rda version of a .csv from a 2005 Cambridge, MA City Council multiwinner RCV election. This data is included as an example of another master lookup format for use with the label function. This data includes information other than the candidate names and codes, but we only use the candidate names and codes.

### Usage

```
cambridge_lookup
```

### Format

A data frame with 588 rows and 6 variables

---

characterize	<i>Replaces number string codes in ballot with character strings from lookup</i>
--------------	----------------------------------------------------------------------------------

---

**Description**

Matches codes in the 'contest\_id', 'tally\_type\_id', 'precinct\_id', and 'candidate\_id' columns in the labelled ballot with codes from the 'id' column in the labelled lookup, then replaces these codes with character values from the 'description' column in the lookup.

**Usage**

```
characterize(ballot, lookup, format)
```

**Arguments**

ballot	The labelled ballot data
lookup	The labelled lookup data
format	A character string detailing the format. Current supported formats are "WinEDS" and "ChoicePlus" (forthcoming), based on common types of software used. Contact creators with suggestions for more formats.

**Value**

The ballot data, but now "readable" so votes can be understood

**Examples**

```
## Not run:
characterize(ballot = sf_ballot_labelled, lookup = sf_lookup_labelled,
format = "WinEDS")

## End(Not run)
```

---

clean_ballot	<i>Master one-step cleaning function</i>
--------------	------------------------------------------

---

**Description**

Wraps 'import\_data', 'label', and 'characterize' to clean the ballot image in one step.

**Usage**

```
clean_ballot(ballot, b_header, lookup, l_header, format)
```

**Arguments**

ballot	The raw ballot image
b_header	Whether the ballot image has a header line or not
lookup	The raw lookup image
l_header	Whether the lookup image has a header line or not
format	A character string detailing the format. Current supported formats are "WinEDS" and "ChoicePlus" (forthcoming), based on common types of software used. Contact creators with suggestions for more formats.

**Value**

The ballot data, but now "readable" so votes can be understood

**Examples**

```
clean_ballot(ballot = sf_bos_ballot, b_header = TRUE,
             lookup = sf_bos_lookup, l_header = TRUE, format = "WinEDS")
```

---

clean\_mn

*Function for cleaning Minneapolis RCV data*

---

**Description**

The Minneapolis data comes in a different form than SF or Cambridge data. This function optimizes the process for formatting this data.

**Usage**

```
clean_mn(data)
```

**Arguments**

data	The raw RCV data
------	------------------

**Value**

The data formatted for use with rcv\_tally

**Examples**

```
clean_mn(minneapolis_mayor_2013)
```

---

elected	<i>Summarizes which candidates were elected from the results tally</i>
---------	------------------------------------------------------------------------

---

**Description**

Summarizes which candidates were elected from the results tally

**Usage**

```
elected(results, n = 1)
```

**Arguments**

results	the results tabulated from the election in question
n	the number of candidates being elected (defaults to 1)

**Value**

a vector of the candidates successfully elected

**Examples**

```
elected(sf_7_results, n = 1)
```

---

import_data	<i>Imports election data</i>
-------------	------------------------------

---

**Description**

Takes data argument supplied, checks file type, and uses appropriate read-in functions to import the data

**Usage**

```
import_data(data, header)
```

**Arguments**

data	The file, containing ballot or lookup data
header	Whether the first row of the file is a header or not

**Value**

A data frame

**Examples**

```
import_data("http://www.sfelections.org/results/20161108/data/20161206/20161206_masterlookup.txt",
  header = FALSE)
```

---

label	<i>Separates single column election data frames.</i>
-------	------------------------------------------------------

---

**Description**

Takes a data frame of a single column (i.e. sf\_bos\_ballot) and splits it into usable named columns.

**Usage**

```
label(data, image, format)
```

**Arguments**

data	A data frame with a single column
image	Whether the data is a "ballot" or "lookup" image
format	A character string detailing the format. Current supported formats are "WinEDS" and "ChoicePlus" (forthcoming), based on common types of software used. Contact creators with suggestions for more formats.

**Value**

A data frame with multiple columns

**Examples**

```
label(data = sf_bos_ballot, image = "ballot", format = "WinEDS")
```

---

make_alluvialdf	<i>Creates a data frame for use with the alluvial package</i>
-----------------	---------------------------------------------------------------

---

**Description**

Creates a data frame for use with the alluvial package

**Usage**

```
make_alluvialdf(image, rcvcontest)
```

**Arguments**

image	A dataframe containing rcv election data
rcvcontest	The election to calculate results for



**Value**

A dataframe that counts how many ballots follow each unique "path" of candidates through the election rounds

**Examples**

```
## Not run:  
make_alluvialdf(image = sf_bos_clean, rcvcontest = "Board of Supervisors, District 7")  
  
## End(Not run)
```

---

make\_d3list

*Creates a data frame for use with the networkD3 package*

---

**Description**

Creates a data frame for use with the networkD3 package

**Usage**

```
make_d3list(results)
```

**Arguments**

results      A dataframe containing rcv election results in the format produced by 'rcv\_tally()'

**Value**

A list of 2 dataframes that can be used to construct sankey diagrams via the networkD3 package's 'sankeyNetwork()' function

**Examples**

```
make_d3list(results = sf_7_results)
```

---

minneapolis\_mayor\_2013

*Raw ballot data from a Minneapolis, MN RCV election*

---

### Description

The .rda version of a .csv from a 2013 Minneapolis, MN mayoral RCV election. This data is included as an example of another ballot format for use with ballot functions.

### Usage

```
minneapolis_mayor_2013
```

### Format

A data frame with 80101 rows and 5 variables

### Source

<http://vote.minneapolismn.gov/www/groups/public/@clerk/documents/webcontent/2013-mayor-cvr.xlsx>

---

rcv

*Package: rcv*

---

### Description

Ranked Choice Voting

### Details

A collection of ranked choice voting data and functions to manipulate, run elections with, and visualize this data and others. It can bring in raw data, transform it into a ballot you can read, and return election results for an RCV contest.

### rcv Functions

```
import_data(), label(), characterize(), clean_ballot(), clean_mn(), readable(), elected(), approval(),  
rcv_tally(), add_exhausted(), make_alluvialdf(), make_d3list()
```

---

rcv_tally	<i>Determines RCV round results in a dataframe</i>
-----------	----------------------------------------------------

---

**Description**

Determines RCV round results in a dataframe

**Usage**

```
rcv_tally(image, rcvcontest)
```

**Arguments**

image	A dataframe containing rcv election data
rcvcontest	(optional) The election to calculate results for. If the image contains more than one unique contest, this must be supplied. In most cases except for some San Francisco elections, this is unnecessary.

**Value**

A dataframe that contains vote tallies

**Examples**

```
rcv_tally(image = sf_bos_clean, rcvcontest = "Board of Supervisors, District 7")
```

---

readable	<i>Alters ballot dataframe to have each row correspond to a single voter</i>
----------	------------------------------------------------------------------------------

---

**Description**

Alters ballot dataframe to have each row correspond to a single voter

**Usage**

```
readable(clean)
```

**Arguments**

clean	A tidy dataframe that comes from ballot_tidy()
-------	------------------------------------------------

**Value**

A dataframe that can be easily read and understood by humans

**Examples**

```
readable(sf_bos_clean)
```

---

sf_7_results	<i>Tabulated results from a San Francisco RCV election</i>
--------------	------------------------------------------------------------

---

**Description**

Results from the SF "Board of Supervisors, District 7" election in 2016. Tabulated with `rcv_tally.rcv_tally(image = sf_bos_clean, rcvcontest = "Board of Supervisors, District 7")`

**Usage**

```
sf_7_results
```

**Format**

a data frame with 6 rows and 5 variables

---

sf_bos_ballot	<i>Raw ballot image data from a San Francisco RCV election</i>
---------------	----------------------------------------------------------------

---

**Description**

The `.rda` version of a raw `.txt` file, containing ballot data in numeric form for the 2016 San Francisco Board of Supervisors elections in Districts 1, 3, 5, 7, 9, and 11.

**Usage**

```
sf_bos_ballot
```

**Format**

A data frame with 643806 rows and 1 variable

**Source**

[http://www.sfelections.org/results/20161108/data/20161206/20161206\\_ballotimage.txt](http://www.sfelections.org/results/20161108/data/20161206/20161206_ballotimage.txt)

---

sf_bos_clean	<i>Cleaned ballot image data from a San Francisco RCV election</i>
--------------	--------------------------------------------------------------------

---

**Description**

A tidied version of 'sf\_bos\_ballot' in a "tall" format, readable to see voters' information, including candidate rankings, precinct, and over-/under-vote codings. Cleaned with `clean_ballot`. `clean_ballot(sf_bos_ballot, b_header = T, sf_bos_lookup, l_header = T, format = "WinEDS")`

**Usage**

```
sf_bos_clean
```

**Format**

A data frame with 643806 rows and 9 variables:

**contest** which election this candidate ranking applies to

**pref\_voter\_id** a unique key identifying an individual voter

**serial\_number** the serial number of the voting machine used

**tally** contains information about whether the ballot was cast in early voting, was filed provisionally, and other factors

**precinct** which city precinct the voter voted in

**vote\_rank** the rank given to the candidate by the voter; in the San Francisco case can take values of 1, 2, or 3

**candidate** the chosen candidate for the specified vote rank

**over\_vote** a dummy variable, coded 1 if the ballot shows more votes cast than the number of candidates to be elected

**under\_vote** a dummy variable, coded 1 if the ballot shows no valid selection made for a candidate

---

sf_bos_lookup	<i>Raw ballot lookup data from a San Francisco RCV election.</i>
---------------	------------------------------------------------------------------

---

**Description**

The .rda version of a raw .txt file, containing numeric and character information to match the numeric data in the raw ballot image with the objects these numeric strings refer to. Data refers to the 2016 San Francisco Board of Supervisors elections in Districts 1, 3, 5, 7, 9, and 11. All San Francisco data is to be used with the "WinEDS" format.

**Usage**

```
sf_bos_lookup
```

**Format**

A data frame with 644 rows and 1 variable

**Source**

[http://www.sfelections.org/results/20161108/data/20161206/20161206\\_masterlookup.txt](http://www.sfelections.org/results/20161108/data/20161206/20161206_masterlookup.txt)

---

sf_lookup_labelled	<i>Labelled ballot lookup data from a San Francisco RCV election.</i>
--------------------	-----------------------------------------------------------------------

---

**Description**

The .rda version of sf\_bos\_lookup.rda after the 'label()' function has been applied to it.

**Usage**

```
sf_lookup_labelled
```

**Format**

A data frame with 644 rows and 7 variables

# Index

## \*Topic **datasets**

- cambridge\_ballot, [3](#)
- cambridge\_clean, [4](#)
- cambridge\_lookup, [4](#)
- minneapolis\_mayor\_2013, [10](#)
- sf\_7\_results, [12](#)
- sf\_bos\_ballot, [12](#)
- sf\_bos\_clean, [13](#)
- sf\_bos\_lookup, [13](#)
- sf\_lookup\_labelled, [14](#)

- add\_exhausted, [2](#)
- approval, [3](#)

- cambridge\_ballot, [3](#)
- cambridge\_clean, [4](#)
- cambridge\_lookup, [4](#)
- characterize, [5](#)
- clean\_ballot, [5](#)
- clean\_mn, [6](#)

- elected, [7](#)

- import\_data, [7](#)

- label, [8](#)

- make\_alluvialdf, [8](#)
- make\_d3list, [9](#)
- minneapolis\_mayor\_2013, [10](#)

- rcv, [10](#)
- rcv-package (rcv), [10](#)
- rcv\_tally, [11](#)
- readable, [11](#)

- sf\_7\_results, [12](#)
- sf\_bos\_ballot, [12](#)
- sf\_bos\_clean, [13](#)
- sf\_bos\_lookup, [13](#)
- sf\_lookup\_labelled, [14](#)