# Package 'mRchmadness'

April 9, 2017

**Title** Numerical Tools for Filling Out an NCAA Basketball Tournament Bracket

**Version** 1.0.0

**URL** https://github.com/elishayer/mRchmadness

**Imports** dplyr, glmnet, Matrix, rvest, shiny, xml2

**Description** Scrape season results, estimate win probabilities, and find a competitive bracket for your office pool. Additional utilities include: scraping population picks; simulating tournament results; and testing your bracket in simulation.

**Depends** R (>= 3.3.0)

**License** GPL-2

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Eli Shayer [aut, cre], Scott Powers [aut]

**Maintainer** Eli Shayer <eshayer@stanford.edu>

**Repository** CRAN

**Date/Publication** 2017-04-09 13:56:07 UTC

## R topics documented:

---

`bracket.men.2017` *2017 Men's March Madness bracket*

---

## Description

This dataset contains the ESPN team ids of the 64 teams in the 2017 March Madness men's bracket. The teams are ordered by overall seed, such that the first four team ids correspond to the four #1 seeds.

## Format

character vector of length 64

## bracket.women.2017 *2017 Women's March Madness bracket*

### Description

This dataset contains the ESPN team ids of the 64 teams in the 2017 March Madness women's bracket. The teams are ordered by overall seed, such that the first four team ids correspond to the four #1 seeds.

### Format

character vector of length 64

## bradley.terry *Fit a Bradley-Terry model on game score data*

### Description

Fit a Bradley-Terry model on game score data

### Usage

```
bradley.terry(games)
```

### Arguments

| | |
|---|---|
| games | data.frame with the following columns: game.id, home.id, away.id, home.score, away.score, neutral, ot (matched by output of scrape.game.results) |

### Value

matrix of win probabilities, with rows and columns labeled by team. Each entry gives the probability of the team corresponding to that row beating the team corresponding to that column.

### Author(s)

sspowers

### Examples

```
prob = bradley.terry(games = games.men.2017)
```

## draw.bracket *Plot bracket to device*

### Description

Plot bracket to device

### Usage

```
draw.bracket(bracket.empty, bracket.filled = NULL, league = c("men",
  "women"))
```

### Arguments

| | |
|---|---|
| bracket.empty | a length-64 character vector giving the field of 64 teams in the tournament, in order of initial overall seeding |
| bracket.filled | an optional length-63 character vector encoding tournament results (matching output from simulate.bracket) |
| league | which league: "men" (default) or "women". Used for converting team IDs into team names |

### Author(s)

sspowers

### Examples

```
prob.matrix = bradley.terry(games = games.men.2017)
outcome = sim.bracket(bracket.empty = bracket.men.2017,
  prob.matrix = prob.matrix)
draw.bracket(bracket.empty = bracket.men.2017, bracket.filled = outcome)
```

## find.bracket *Fill out a bracket based on some criteria*

### Description

Fill out a bracket based on some criteria

### Usage

```
find.bracket(bracket.empty, prob.matrix = NULL, prob.source = c("pop",
  "Pom", "538"), pool.source = c("pop", "Pom", "538"), league = c("men",
  "women"), year = 2017, num.candidates = 100, num.sims = 1000,
  criterion = c("percentile", "score", "win"), pool.size = 30,
  bonus.round = c(1, 2, 4, 8, 16, 32), bonus.seed = rep(0, 16),
  bonus.combine = c("add", "multiply"))
```

## Arguments

| | |
|---|---|
| bracket.empty | a length-64 character vector giving the field of 64 teams in the tournament, in order of initial overall seeding |
| prob.matrix | a matrix of probabilities, with rows and columns corresponding to teams, matching the output of bradley.terry(). This probabilities are used to simulate candidate brackets and outcomes on which to evaluate the candidates. If NULL, prob.source is used. |
| prob.source | source from which to use round probabilities to simulate candidate brackets and outcomes — "pop": ESPN's population of picks (default), "Pom": Ken Pomeroy's predictions (kenpom.com), or "538": predictions form fivethirtyeight.com. Ignored if prob.matrix is specified. |
| pool.source | source from which to use round probabilities to simulate entries of opponents in pool. Same options as prob.source. |
| league | which league: "men" (default) or "women", for pool.source. |
| year | year of tournament, used for prob.source and pool.source |
| num.candidates | number of random brackets to try, taking the best one (default is 100) |
| num.sims | number of simulations over which to evaluate the candidate brackets (default is 1000) |
| criterion | how to choose among candidate brackets: "percentile" (default, maximize expected percentile within pool), "score" (maximize expected number of points) or "win" (maximize probabilty of winning pool). |
| pool.size | number of brackets in your pool (excluding yours), matters only if criterion == "win" (default is 30) |
| bonus.round | a length-6 vector giving the number of points awarded in your pool's scoring rules for correct picks in each round (default is 2^round) |
| bonus.seed | a length-16 vector giving the bonus awarded for correctly picking winner based on winner's seed (default is zero) |
| bonus.combine | how to combine the round bonus with the seed bonus to get the number of points awarded for each correct pick: "add" (default) or multiply |

## Value

the length-63 character vector describing the filled bracket which performs best according to criterion among all num.candidates brackets tried, across num.sims simulations of a pool of pool.size with scoring rules specified by bonus.round, bonus.seed and bonus.combine

## Author(s)

sspowers

## Examples

```
find.bracket(bracket.empty = bracket.men.2017, prob.source = "538",
  pool.source = "pop", league = "men", year = 2017)
```

## fold *Fold a vector onto itself*

### Description

Fold a vector onto itself

### Usage

```
fold(x, block.size = 1)
```

### Arguments

| | |
|---|---|
| x | a vector |
| block.size | the size of groups in which to block the data |

### Value

a new vector in the following order: first block, last block, second block, second-to-last block, ...

### Author(s)

sspowers

## games.men.2016 *2016 NCAA Men's Basketball Game-by-Game Results*

### Description

This dataset contains the game results of the college men's basketball 2015-2016 season, identifying the game, the home and away teams, (using the ESPN ID system), the home and away scores, whether the game was played at a neutral arena, and whether the game went into overtime. Teams that are not in Division I are uniformly identified by the string 'NA' as their id.

### Format

data frame with 5924 rows and 7 variables

### Source

ESPN

## games.men.2017 *2017 NCAA Men's Basketball Game-by-Game Results*

### Description

This dataset contains the game results of the college men's basketball 2016-2017 season, identifying the game, the home and away teams, (using the ESPN ID system), the home and away scores, whether the game was played at a neutral arena, and whether the game went into overtime. Teams that are not in Division I are uniformly identified by the string 'NA' as their id.

### Format

data frame with 5871 rows and 7 variables

### Source

ESPN

## games.women.2017 *2017 NCAA Women's Basketball Game-by-Game Results*

### Description

This dataset contains the game results of the college women's basketball 2016-2017 season, identifying the game, the home and away teams, (using the ESPN ID system), the home and away scores, whether the game was played at a neutral arena, and whether the game went into overtime. Teams that are not in Division I are uniformly identified by the string 'NA' as their id.

### Format

data frame with 5575 rows and 7 variables

### Source

ESPN

## mRchmadness *March Madness Bracket Package*

### Description

mRchmadness provides utilities to gather NCAA Men's Basketball data, predict the win probability of matchups, and produce a bracket optimized to a specified criterion

## pred.538.men.2017 *2017 March Madness 538 Win Probability Predictions*

### Description

This dataset contains the win probabilities forecasted by the website FiveThirtyEight immediately after the First Four and before any other games had been played in the March Madness tournament of the 2016-2017 season. The columns are the names of the teams, and FiveThirtyEight's projected probability of reaching each round of the tournament.

### Format

data frame with 64 rows and 7 variables

### Source

https://projects.fivethirtyeight.com/2017-march-madness-predictions/

## pred.538.women.2017 *2017 Women's March Madness 538 Win Probability Predictions*

### Description

This dataset contains the win probabilities forecasted by the website FiveThirtyEight for the Women's March Madness tournament of the 2016-2017 season. The columns are the names of the teams, and FiveThirtyEight's projected probability of reaching each round.

### Format

data frame with 64 rows and 7 variables

### Source

https://projects.fivethirtyeight.com/2017-march-madness-predictions/

## pred.pop.men.2016 *2016 March Madness Men's Population Pick Distribution*

### Description

This dataset contains the percent of brackets submitted to ESPN that include each of the 64 teams in Men's March Madness 2016 reaching and winning in each successive round of the tournament.

### Format

data frame with 64 rows and 7 variables

### Source

http://games.espn.com/tournament-challenge-bracket/2016/en/whopickedwhom

## pred.pop.men.2017 *2017 March Madness Population Pick Distribution*

### Description

This dataset contains the percent of brackets submitted to ESPN that include each of the 64 teams in March Madness 2017 reaching and winning in each successive round of the tournament.

### Format

data frame with 64 rows and 7 variables

### Source

http://games.espn.com/tournament-challenge-bracket/2017/en/whopickedwhom

## pred.pop.women.2016 *2016 March Madness Women's Population Pick Distribution*

### Description

This dataset contains the percent of brackets submitted to ESPN that include each of the 64 teams in Women's March Madness 2016 reaching and winning in each successive round of the tournament.

### Format

data frame with 64 rows and 7 variables

### Source

http://games.espn.com/tournament-challenge-bracket-women/2016/en/whopickedwhom

## pred.pop.women.2017 *2017 March Madness Women's Population Pick Distribution*

### Description

This dataset contains the percent of brackets submitted to ESPN that include each of the 64 teams in Women's March Madness 2017 reaching and winning in each successive round of the tournament.

### Format

data frame with 64 rows and 7 variables

### Source

http://games.espn.com/tournament-challenge-bracket-women/2017/en/whopickedwhom

## run.app *Run the Shiny app allowing for interaction with the bracket production given user-entered criteria*

### Description

Run the Shiny app allowing for interaction with the bracket production given user-entered criteria

### Usage

```
run.app()
```

### Author(s)

eshayer

score.bracket *Compute score for bracket given actual result*

## Description

Compute score for bracket given actual result

## Usage

```
score.bracket(bracket.empty, bracket.picks, bracket.outcome,
  bonus.round = c(1, 2, 4, 8, 16, 32), bonus.seed = rep(0, 16),
  bonus.combine = c("add", "multiply"))
```

## Arguments

| | |
|---|---|
| bracket.empty | a length-64 character vector giving the field of 64 teams in the tournament, in order of initial overall seeding |
| bracket.picks | an length-63 character vector encoding the picks (this is the bracket to be evaluated) |
| bracket.outcome | a 63-row matrix encoding the outcome of multiple simulations of the tournament. bracket.picks will be scored against each outcome |
| bonus.round | a length-6 vector giving the number of points awarded in your pool's scoring rules for correct picks in each round (default is 2^round) |
| bonus.seed | a length-16 vector giving the bonus awarded for correctly picking winner based on winner's seed (default is zero) |
| bonus.combine | how to combine the round bonus with the seed bonus to get the number of points awarded for each correct pick: "add" (default) or multiply |

## Value

a vector giving the score for bracket.picks for each outcome in the matrix bracket.oucome

## Author(s)

sspowers

## scrape.game.results *Scrape the game-by-game results of the NCAA MBB seaon*

### Description

Scrape the game-by-game results of the NCAA MBB seaon

### Usage

```
scrape.game.results(year, sex = c("mens", "womens"))
```

### Arguments

| | |
|---|---|
| year | a numeric value of the year, between 2002 and 2017 inclusive |
| sex | either 'mens' or 'womens' |

### Value

data.frame with game-by-game results

### Author(s)

eshayer

## scrape.population.distribution *Scrape the average rate of teams being picked to win across all ESPN brackets*

### Description

Scrape the average rate of teams being picked to win across all ESPN brackets

### Usage

```
scrape.population.distribution(year, sex = c("mens", "womens"))
```

### Arguments

| | |
|---|---|
| year | the numeric year to scrape, either 2016 or 2017 |
| sex | either 'mens' or 'womens' |

### Value

data.frame giving percentage of population picking each team in each round

### Author(s)

eshayer

### Examples

```
populationDistribution = scrape.population.distribution(2017)
```

---

## scrape.team.game.results *Scrape game results for a single team-year combination*

---

### Description

Scrape game results for a single team-year combination

### Usage

```
scrape.team.game.results(year, team.id, sex)
```

### Arguments

| | |
|---|---|
| year | a character value representing a year |
| team.id | an ESPN team id |
| sex | either 'mens' or 'womens' |

### Value

data.frame of game data for the team-year

### Author(s)

eshayer

---

## scrape.teams *Scrape the team names and ids from the ESPN NCAA MBB index*

---

### Description

Scrape the team names and ids from the ESPN NCAA MBB index

### Usage

```
scrape.teams(sex)
```

### Arguments

| | |
|---|---|
| sex | either 'mens' or 'womens' |

### Value

data.frame of team names and ids

### Author(s)

eshayer

---

## sim.bracket *Simulate the full bracket starting with an empty bracket*

---

### Description

Simulate the full bracket starting with an empty bracket

### Usage

```
sim.bracket(bracket.empty, prob.matrix = NULL, prob.source = c("pop", "Pom",
  "538"), league = c("men", "women"), year = 2017, num.reps = 1)
```

### Arguments

| | |
|---|---|
| bracket.empty | a length-64 character vector giving the field of 64 teams in the tournament, in order of initial overall seeding |
| prob.matrix | a matrix of probabilities, with rows and columns corresponding to teams, matching the output of bradley.terry(). If NULL, prob.source is used. |
| prob.source | source from which to use round probabilities for simulation — "pop": ESPN's population of picks (default), "Pom": Ken Pomeroy's predictions (kenpom.com), or "538": predictions form fivethirtyeight.com. Ignored if prob.matrix is specified. |
| league | which league: "men" (default) or "women", for prob.source. Ignored if prob.matrix is specified. |
| year | year of tournament, used for prob.source. Ignored if prob.matrix is specified. |
| num.reps | number of simulations to perform (default is 1) |

### Value

a 63-by-num.reps matrix storing the simulation outcome, each column encoding the outcome for a single simulation in the following order: seeds 1 through 32 after round 1, seeds 1 through 16 after round 2, seeds 1 through 8 after round 3, seeds 1 through 4 after round 4, seeds 1 and 2 after round 5, and finally seed 1 after round 6 (the champion)

### Author(s)

sspowers

### Examples

```
sim.bracket(bracket.empty = bracket.men.2017, prob.source = "538",
  league = "men", year = 2017)
```

---

## sim.bracket.matrix *Simulate the full bracket starting with an empty bracket*

---

### Description

Simulate the full bracket starting with an empty bracket

### Usage

```
sim.bracket.matrix(prob.matrix, league, num.reps, outcome, round,
  teams.remaining, untangling.indices)
```

### Arguments

| | |
|---|---|
| prob.matrix | a matrix of probabilities, with rows and columns corresponding to teams, matching the output of bradley.terry() |
| league | which league: "men" (default) or "women" |
| num.reps | number of simulations to perform |
| outcome | passed in from sim.bracket() |
| round | passed in from sim.bracket() |
| teams.remaining | passed in from sim.bracket() |
| untangling.indices | passed in from sim.bracket() |

### Value

a 63-by-num.reps matrix storing the simulation outcome, each column encoding the outcome for a single simulation in the following order: seeds 1 through 32 after round 1, seeds 1 through 16 after round 2, seeds 1 through 8 after round 3, seeds 1 through 4 after round 4, seeds 1 and 2 after round 5, and finally seed 1 after round 6 (the champion)

### Author(s)

sspowers

`sim.bracket.source` *Simulate the full bracket starting with an empty bracket*

## Description

Simulate the full bracket starting with an empty bracket

## Usage

```
sim.bracket.source(prob.source, league, year, num.reps, outcome, round,
  teams.remaining, untangling.indices)
```

## Arguments

| | |
|---|---|
| `prob.source` | source from which to use round probabilities for simulation — "pop": ESPN's population of picks, "Pom": Ken Pomeroy's predictions (kenpom.com), or "538": predictions form fivethirtyeight.com. |
| `league` | which league: "men" (default) or "women", for prob.source. |
| `year` | year of tournament, used for prob.source. |
| `num.reps` | number of simulations to perform |
| `outcome` | passed in from sim.bracket() |
| `round` | passed in from sim.bracket() |
| `teams.remaining` | passed in from sim.bracket() |
| `untangling.indices` | passed in from sim.bracket() |

## Value

a 63-by-num.reps matrix storing the simulation outcome, each column encoding the outcome for a single simulation in the following order: seeds 1 through 32 after round 1, seeds 1 through 16 after round 2, seeds 1 through 8 after round 3, seeds 1 through 4 after round 4, seeds 1 and 2 after round 5, and finally seed 1 after round 6 (the champion)

## Author(s)

sspowers

## teams.men *NCAA Men's Basketball Teams*

### Description

This dataset the names of each team in NCAA Men's Basketball from ESPN, their ESPN team id, their name as it appears on the Who Picked Whom, and the FiveThirtyEight name. The last two name fields are only available for those teams who played in March Madness in 2016 or 2017, and are otherwise NA.

### Format

data frame with 351 rows and 4 variables

### Source

http://www.espn.com/mens-college-basketball/teams

## teams.women *NCAA Women's Basketball Teams*

### Description

This dataset the names of each team in NCAA Women's Basketball from ESPN, their ESPN team id, their name as it appears on the Who Picked Whom, and their FiveThirtyEight name.

### Format

data frame with 349 rows and 4 variables

### Source

http://www.espn.com/womens-college-basketball/teams

`test.bracket` *Test a bracket*

## Description

Test a bracket

## Usage

```
test.bracket(bracket.empty, bracket.picks, prob.matrix = NULL,
  prob.source = c("pop", "Pom", "538"), pool.source = c("pop", "Pom",
  "538"), league = c("men", "women"), year = 2017, pool.size = 30,
  num.sims = 1000, bonus.round = c(1, 2, 4, 8, 16, 32),
  bonus.seed = rep(0, 16), bonus.combine = c("add", "multiply"))
```

## Arguments

| | |
|---|---|
| `bracket.empty` | a length-64 character vector giving the field of 64 teams in the tournament, in order of initial overall seeding |
| `bracket.picks` | an length-63 character vector encoding your picks (this is the bracket to be evaluated) |
| `prob.matrix` | a matrix of probabilities, with rows and columns corresponding to teams, matching the output of bradley.terry(). This probabilities are used to simulate outcomes on which to evaluate bracket.picks. If NULL, prob.source is used. |
| `prob.source` | source from which to use round probabilities to simulate outcomes — "pop": ESPN's population of picks (default), "Pom": Ken Pomeroy's predictions (kenpom.com), or "538": predictions form fivethirtyeight.com. Ignored if prob.matrix is specified. |
| `pool.source` | source from which to use round probabilities to simulate entries of opponents in pool. Same options as prob.source. |
| `league` | which league: "men" (default) or "women", for pool.source. |
| `year` | year of tournament, used for prob.source. Ignored if prob.matrix is specified. |
| `pool.size` | number of brackets in your pool (excluding yours), matters only if criterion == "win" (default is 30) |
| `num.sims` | number of simulations over which to evaluate the candidate brackets (default is 1000) |
| `bonus.round` | a length-6 vector giving the number of points awarded in your pool's scoring rules for correct picks in each round (default is 2^round) |
| `bonus.seed` | a length-16 vector giving the bonus awarded for correctly picking winner based on winner's seed (default is zero) |
| `bonus.combine` | how to combine the round bonus with the seed bonus to get the number of points awarded for each correct pick: "add" (default) or multiply |

### Author(s)

sspowers

### Examples

```
prob.matrix = bradley.terry(games = games.men.2017)
my.bracket = find.bracket(bracket.empty = bracket.men.2017,
  prob.matrix = prob.matrix, pool.source = "pop", league = "men",
  year = 2017)
result = test.bracket(bracket.empty = bracket.men.2017,
  bracket.picks = my.bracket, prob.matrix = prob.matrix,
  pool.source = "pop", league = "men", year = 2017)
```

---

## unfold *Unfold a vector (the inverse of the fold function)*

---

### Description

Unfold a vector (the inverse of the fold function)

### Usage

```
unfold(x, block.size = 1)
```

### Arguments

| | |
|---|---|
| x | a vector |
| block.size | the size of groups in which to block the data |

### Value

a vector in the following order: block 1, block 3, ..., block n-1, block n, block n-2, ..., block 2.

### Author(s)

sspowers

# Index