Package ‘RMAT’

May 7, 2026

Version 0.2.0

Title Random Matrix Analysis Toolkit

Maintainer Ali Taqi <alif.taqi00@gmail.com>

Description
Simulate random matrices and ensembles and compute their eigenvalue spectra and dispersions.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports purrr, dplyr, stats, magrittr

Suggests tidyverse

NeedsCompilation no

Author Ali Taqi [aut, cre],
Jonathan Wells [ths]

Repository CRAN

Date/Publication 2021-04-28 07:50:05 UTC

Contents

`dispersion` *Obtain the eigenvalue spacings of a matrix or ensemble of matrices.*

Description

Returns a vector of the eigenvalue spacings of a random matrix or ensemble.

Usage

```
dispersion(
  array,
  pairs = NA,
  norm_order = TRUE,
  singular = FALSE,
  pow_norm = 1
)
```

Arguments

`array`	a square matrix or matrix ensemble whose eigenvalue spacings are to be returned
`pairs`	a string argument representing the pairing scheme to use
`norm_order`	sorts the eigenvalue spectrum by its norms if TRUE, otherwise sorts them by sign
`singular`	return the singular values of the matrix or matrix ensemble
`pow_norm`	power to raise norm to - defaults to 1 (the standard absolute value); otherwise raises norm to the power of argument (beta norm)

Value

A tidy dataframe with the real & imaginary components of the eigenvalues and their norms along with a unique index.

Examples

```
# Eigenvalue dispersion of a normal matrix using the lower pair scheme
P <- RM_norm(N = 5)
disp_P <- dispersion(P, pairs = "lower")

# Eigenvalue dispersion of a stochastic matrix (using the consecutive pair scheme)
Q <- RM_stoch(N = 5)
disp_Q <- dispersion(Q)

# Eigenvalue dispersion of an normal matrix ensemble, ordering by sign instead of norm.
ens <- RME_beta(N = 10, beta = 2, size = 10)
disp_ens <- dispersion(ens, norm_order = FALSE)
```

RME_beta *Generate an ensemble of random beta matrices*

Description

Given the same arguments as RM_norm, this function returns an ensemble of that particular class of matrix. While random matrices usually do not exude unique properties on their own, they do indeed have deterministic properties at the ensemble level in terms of their spectral statistics.

Usage

```
RME_beta(N, beta, size)
```

Arguments

N	number of dimensions of the square matrix
beta	the value of the beta parameter for the beta ensemble
size	the size of the ensemble (i.e. number of matrices)

Value

An ensemble (list) of beta matrices as specified by the matrix arguments.

Examples

```
# Generate an ensemble of 10x10 beta matrices with beta = 4 of size 100.
ensemble <- RME_beta(N = 10, beta = 4, size = 100)
```

RME_erdos *Generate an ensemble of Erdos-Renyi transition matrices*

Description

Given the same arguments as RM_norm, this function returns an ensemble of random Erdos-Renyi stochastic matrices. While random matrices usually do not exude unique properties on their own, they do indeed have deterministic properties at the ensemble level in terms of their spectral statistics.

Usage

```
RME_erdos(N, p, size)
```

Arguments

N	number of dimensions of the square matrix
p	the probability two vertices are connected in an Erdos-Renyi graph.
size	the size of the ensemble (i.e. number of matrices)

Value

An ensemble (list) of Erdos-Renyi transition matrices as specified by the matrix arguments.

Examples

```
# Generate an ensemble of 10x10 Erdos-Renyi transition matrices of size 50 with p = 0.7
ensemble <- RME_erdos(N = 10, p = 0.7, size = 50)
```

RME_norm *Generate an ensemble of normal random matrices*

Description

Given the same arguments as RM_norm, this function returns an ensemble of random normal matrices. While random matrices usually do not exude unique properties on their own, they do indeed have deterministic properties at the ensemble level in terms of their spectral statistics.

Usage

```
RME_norm(N, mean = 0, sd = 1, ..., size)
```

Arguments

N	number of dimensions of the square matrix
mean	mean of the normal distribution of entries
sd	standard deviation of the normal distribution of entries
...	any default-valued parameters taken as arguments by RM_norm()
size	the size of the ensemble (i.e. number of matrices)

Value

An ensemble (list) of normal matrices as specified by the matrix arguments.

Examples

```
# Generate an ensemble of standard normal 3x3 matrices of size 20
ensemble <- RME_norm(N = 3, size = 20)
```

RME_stoch *Generate an ensemble of stochastic matrices*

Description

Given the same arguments as RM_stoch, this function returns an ensemble of random stochastic matrices. While random matrices usually do not exude unique properties on their own, they do indeed have deterministic properties at the ensemble level in terms of their spectral statistics.

Usage

```
RME_stoch(N, ..., size)
```

Arguments

`N`	number of dimensions of the square matrix
`...`	pass any default-valued parameters taken as arguments by RM_stoch()
`size`	the size of the ensemble (i.e. number of matrices)

Value

An ensemble (list) of stochastic matrices as specified by the matrix arguments.

Examples

```
# Generate an ensemble of random 5x5 transition matrices of size 20.
ensemble <- RME_stoch(N = 5, size = 20)

# Generate an ensemble of symmetric random 5x5 transition matrices of size 20.
ensemble <- RME_stoch(N = 5, symm = TRUE, size = 20)
```

RME_unif *Generate an ensemble of normal random matrices*

Description

Given the same arguments as RM_norm, this function returns an ensemble of random normal matrices. While random matrices usually do not exude unique properties on their own, they do indeed have deterministic properties at the ensemble level in terms of their spectral statistics.

Usage

```
RME_unif(N, min, max, ..., size)
```

Arguments

N	number of dimensions of the square matrix
min	minimum of the uniform distribution to be sampled from
max	maximum of the uniform distribution to be sampled from
...	any default-valued parameters taken as arguments by RM_norm()
size	the size of the ensemble (i.e. number of matrices)

Value

An ensemble (list) of normal matrices as specified by the matrix arguments.

Examples

```
# Generate an ensemble of standard normal 3x3 matrices of size 20
ensemble <- RME_norm(N = 3, size = 20)
```

RM_beta *Generate a Hermite β-matrix*

Description

Hermite-β ensemble matrices are matrices with normal entries and beta real number components. Using Dumitriu's tridiagonal model, this function is an implementation of the generalized, but not necessarily invariant, beta ensembles for $\beta > 0$.

Usage

```
RM_beta(N, beta)
```

Arguments

N	number of dimensions of the square matrix
beta	the value of the beta parameter for the beta ensemble

Value

A random Hermite beta matrix with any integer parameter beta

Examples

```
# Generate a 3x3 random beta matrix with beta = 4
P <- RM_beta(N = 3, beta = 4)

# Generate a 10x10 random beta matrix with beta = 25
P <- RM_beta(N = 10, beta = 25)
```

RM_erdos *Generate a random stochastic matrix for a walk on an Erdos-Renyi graph*

Description

An Erdos-Renyi Graph is a graph whose edges are connected ~ Bern(p). Hence, its transition matrix will have nonzero entries with that probability. So, we can alternatively think of the transition matrix for such walk as a stochastic matrix with parameterized sparsity.

Usage

```
RM_erdos(N, p)
```

Arguments

N	number of dimensions of the square matrix
p	the probability two vertices are connected in an Erdos-Renyi graph.

Value

A random stochastic matrix corrosponding to a walk on an Erdos-Renyi graph with probability p.

Examples

```
# Very sparse graph
P <- RM_erdos(N = 3, p = 0.2)

# Slightly sparse graph
P <- RM_erdos(N = 9, p = 0.6)

# Completely connected graph
P <- RM_erdos(N = 5, p = 1)
```

RM_norm *Generate a normal random matrix*

Description

Normal random matrices are matrices with normally distributed entries. These matrices are extensively studied in random matrix theory.

Usage

```
RM_norm(N, mean = 0, sd = 1, symm = FALSE, cplx = FALSE, herm = FALSE)
```

Arguments

N	number of dimensions of the square matrix
mean	mean of the normal distribution of entries
sd	standard deviation of the normal distribution of entries
symm	indicates whether the matrix should be symmetric (equal to its transpose). Reserved for when cplx = FALSE, otherwise use herm = TRUE.
cplx	indicates whether the matrix should have complex entries.
herm	indicates whether the matrix should be hermitian (equal to its conjugate transpose). Reserved for when cplx = TRUE, otherwise use symm = TRUE.

Value

A random matrix with normally distributed entries.

Examples

```
# N(1,2) distributed matrix
P <- RM_norm(N = 3, mean = 1, sd = 2)

# N(0,5) distributed matrix with real symmetric entries
P <- RM_norm(N = 7, sd = 5, symm = TRUE)

# 7x7 standard normal matrix with complex entries
Q <- RM_norm(N = 7, cplx = TRUE)

# N(2,1) distributed matrix with hermitian complex entries
Q <- RM_norm(N = 5, mean = 2, cplx = TRUE, herm = TRUE)
```

RM_stoch *Generate a random stochastic matrix*

Description

A (row-)stochastic matrix is a matrix whose rums sum to 1. There is a natural one-to-one corrospondence between stochastic matrices and Markov Chains; this is so when its i,j entry represent the transition probability from state i to state j.

Usage

```
RM_stoch(N, symm = FALSE, sparsity = FALSE)
```

Arguments

N	number of dimensions of the square matrix
symm	indicates whether the matrix should be symmetric; equal to its transpose.
sparsity	indicates whether the matrix should add some arbitrary sparsity (zeros)

Value

A random stochastic matrix.

Examples

```
P <- RM_stoch(N = 3)
P <- RM_stoch(N = 9, sparsity = TRUE)
Q <- RM_stoch(N = 9, symm = TRUE)
Q <- RM_stoch(N = 9, symm = TRUE, sparsity = TRUE)
```

RM_trid — *Generate a tridiagonal matrix with normal entries*

Description

Generate a tridiagonal matrix with normal entries

Usage

```
RM_trid(N, symm = FALSE)
```

Arguments

N	number of dimensions of the square matrix
symm	indicates whether the matrix should be symmetric; equal to its transpose.

Value

A random tridiagonal matrix with N(0,2) diagonal and N(0,1) band.

Examples

```
# Generate a 3x3 standard normal tridiagonal matrix
P <- RM_trid(N = 3)

# Symmetric tridiagonal matrix
P <- RM_trid(N = 9, symm = TRUE)
```

RM_unif *Generate a uniform random matrix*

Description

Uniform random matrices are matrices with uniformly distributed entries. They are an elementary type of random matrix.

Usage

```
RM_unif(N, min, max, symm = FALSE, cplx = FALSE, herm = FALSE)
```

Arguments

N	number of dimensions of the square matrix
min	minimum of the uniform distribution to be sampled from
max	maximum of the uniform distribution to be sampled from
symm	indicates whether the matrix should be symmetric (equal to its transpose).
cplx	indicates whether the matrix should have complex entries.
herm	indicates whether the matrix should be hermitian (equal to its conjugate transpose). Reserved for when cplx = TRUE, otherwise use symm = TRUE.

Value

A random matrix with uniformly distributed entries.

Examples

```
# Unif(1,2) distributed matrix
P <- RM_unif(N = 3, min = 1, max = 2)

# Unif(0,5) distributed matrix with real symmetric entries
P <- RM_unif(N = 7, min = 0, max = 5, symm = TRUE)

# Unif(0,1) distributed matrix with complex entries
Q <- RM_unif(N = 7, min = 0, max = 1, cplx = TRUE)

# Unif(2,10) distributed matrix with hermitian complex entries
Q <- RM_unif(N = 5, min = 2, max = 10, cplx = TRUE, herm = TRUE)
```

spectrum — *Obtain the ordered eigenvalue spectrum of a matrix or ensemble of matrices.*

Description

Returns a tidied dataframe of the eigenvalues of a random matrix or ensemble.

Usage

```
spectrum(
  array,
  norm_order = TRUE,
  singular = FALSE,
  components = TRUE,
  order = NA
)
```

Arguments

`array`	a square matrix or matrix ensemble whose eigenvalues are to be returned
`norm_order`	sorts the eigenvalue spectrum by its norms if TRUE, otherwise sorts them by sign
`singular`	return the singular values of the matrix or matrix ensemble
`components`	returns the array with resolved real and imaginary components if TRUE, otherwise returns complex-valued eigenvalues
`order`	an integer or integer vector of which eigenvalue orders to return; order 1 representing the largest, order N represents smallest (where N is the number of eigenvalues). If uninitialized, defaults to returning the entire spectrum.

Value

A tidy dataframe with the real & imaginary components of the eigenvalues and their norms along with a unique index.

Examples

```
# Eigenvalue spectrum of a random normal matrix
P <- RM_norm(N = 5)
spec_P <- spectrum(P)

Q <- matrix(runif(2^2), ncol = 2)
spec_Q <- spectrum(Q)

# Eigenvalue spectra of ensemble matrices
ens <- RME_norm(N = 3, size = 10)
spec_ens <- spectrum(ens)
```

Index