Package ‘wishmom’

August 27, 2024

Type Package

Title Compute Moments Related to Beta-Wishart and Inverse Beta-Wishart Distributions

Version 1.1.0

Description Provides functions for computing moments and coefficients related to the Beta-Wishart and Inverse Beta-Wishart distributions.
It includes functions for calculating the expectation of matrix-valued functions of the Beta-Wishart distribution,
coefficient matrices C_k and H_k, expectation of matrix-valued functions of the inverse Beta-Wishart distribution,
and coefficient matrices \tilde{C}_k and \tilde{H}_k. For more details, refer Hillier and Kan (2024) <https://www-2.rotman.utoronto.ca/~kan/papers/wishmom.pdf>,
``On the Expectations of Equivariant Matrix-valued Functions of Wishart and Inverse Wishart Matrices".

Imports roxygen2

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Raymond Kan [aut, cre],
Preston Liang [aut]

Maintainer Raymond Kan <raymond.kan@rotman.utoronto.ca>

Repository CRAN

Date/Publication 2024-08-27 11:20:06 UTC

Contents

`denpoly` *Coefficients of the Denominator Polynomial for $\tilde{H}$_k and $\tilde{C}$_k*

Description

This function computes the coefficients of the denominator polynomial for the elements of $\tilde{H}_k$ and $\tilde{C}_k$. The function returns a vector containing the coefficients in descending powers of $\tilde{n}$, with the last element being the coefficient of $\tilde{n}$.

Usage

```
denpoly(k, alpha = 2)
```

Arguments

`k`	The order of the polynomial (a positive integer)
`alpha`	The type of Wishart distribution $(\alpha = 2/\beta)$: • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

A vector containing the coefficients of the denominator polynomial in descending powers of $\tilde{n}$ for the elements of $\tilde{H}_k$ and $\mathcal{C}_k$.

Examples

```
# Example 1: Compute the denominator polynomial for k = 3, alpha = 2
# Output corresponds to the polynomial n1^5-3n1^4-8n1^3+12n1^2+16n1,
# where n1 is \eqn{\tilde{n}}
denpoly(3)

# Example 2: Compute the denominator polynomial for k = 2, alpha = 1
# Output corresponds to the polynomial n1^3-n1, where n1 is \eqn{\tilde{n}}
denpoly(2, alpha = 1)
```

---

dkmap — *Mapping Matrix that maps Q_$k+1$ to Q_k for a beta-Wishart Distribution, but without n on the diagonal*

Description

This function computes the matrix that maps Q_{k+1} to Q_k when $W \sim W_m^\beta(n, \Sigma)$.

Usage

```
dkmap(k, alpha = 2)
```

Arguments

k	The order of the mapping matrix D_k (a positive integer)
alpha	The type of beta-Wishart distribution ($\alpha = 2/\beta$): • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

A matrix that maps Q_{k+1} to Q_k, but without n on the diagonal.

Examples

```
# Example 1: Compute the mapping matrix for k = 2, real Wishart
dkmap(2)
# Example 2: Compute the mapping matrix for k = 1, complex Wishart
dkmap(1, 1)
# Example 3: Compute the mapping matrix for k = 2, quaternion Wishart
dkmap(2, 1/2)
```

`ip_desc` *Generate Integer Partitions in Reverse Lexicographical Order*

Description

This function generates all integer partitions of a given integer `k` in reverse lexicographical order. The function is adapted from "Algorithm ZS1" described in Zoghbi and Stojmenovic (1998), "Fast Algorithms for Generating Integer Partitions", International Journal of Computer Mathematics, Volume 70, Issue 2, pages 319-332.

Usage

```
ip_desc(k)
```

Arguments

`k`	An integer to be partitioned

Value

A matrix where each row represents an integer partition of `k`. The partitions are listed in reverse lexicographical order.

References

Zoghbi, A., & Stojmenovic, I. (1998). Fast Algorithms for Generating Integer Partitions. International Journal of Computer Mathematics, 70(2), 319-332. DOI: 10.1080/00207169808804755

Examples

```
# Example 1:
ip_desc(3)

# Example 2:
ip_desc(5)
```

`iwishmom` *Expectation of a Matrix-valued Function of an Inverse beta-Wishart Distribution*

Description

When `iw = 0`, the function calculates $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j}]$, where $W \sim W_m^{\beta}(n, S)$. When `iw != 0`, the function calculates $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j} W^{-iw}]$.

Usage

```
iwishmom(n, S, f, iw = 0, alpha = 2)
```

Arguments

- `n` The degrees of freedom of the beta-Wishart matrix W
- `S` The covariance matrix of the beta-Wishart matrix W
- `f` A vector of nonnegative integers f_j that represents the power of $\text{tr}(W^{-j})$, where $j = 1, \ldots, r$
- `iw` The power of the inverse beta-Wishart matrix W^{-1} (0 by default)
- `alpha` The type of Wishart distribution $(\alpha = 2/\beta)$:
  - 1/2: Quaternion Wishart
  - 1: Complex Wishart
  - 2: Real Wishart (default)

Value

When `iw = 0`, it returns $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j}]$. When `iw != 0`, it returns $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j} W^{-iw}]$.

Examples

```
# Example 1: For E[tr(W^{-1})^2] with W ~ W_m^1(n,S),
# where n and S are defined below:
n <- 20
S <- matrix(c(25, 49,
              49, 109), nrow=2, ncol=2)
iwishmom(n, S, 2) # iw = 0, for real Wishart distribution

# Example 2: For E[tr(W^{-1})^2*tr(W^{-3})W^{-2}] with W ~ W_m^1(n,S),
# where n and S are defined below:
n <- 20
S <- matrix(c(25, 49,
              49, 109), nrow=2, ncol=2)
iwishmom(n, S, c(2, 0, 1), 2, 2) # iw = 2, for real Wishart distribution

# Example 3: For E[tr(W^{-1})^2*tr(W^{-3})] with W ~ W_m^2(n,S),
# where n and S are defined below:
# Hermitian S for the complex case
n <- 20
S <- matrix(c(25, 49 + 2i,
              49 - 2i, 109), nrow=2, ncol=2)
iwishmom(n, S, c(2, 0, 1), 0, 1) # iw = 0, for complex Wishart distribution

# Example 4: For E[tr(W^{-1})*tr(W^{-2})^2*tr(W^{-3})^2*W^{-1}] with W ~ W_m^2(n,S),
# where n and S are defined below:
n <- 30
S <- matrix(c(25, 49 + 2i,
              49 - 2i, 109), nrow=2, ncol=2)
iwishmom(n, S, c(1, 2, 2), 1, 1) # iw = 1, for complex Wishart distribution
```

iwishmom_sym — *Symbolic Expectation of a Matrix-valued Function of an Inverse beta-Wishart Distribution*

Description

When `iw = 0`, the function returns an analytical expression of $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j}]$, where $W \sim W_m^\beta(n, S)$. When `iw != 0`, the function returns an analytical expression of $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j} W^{-iw}]$. For a given `f`, `iw`, and `alpha`, this function provides the aforementioned expectations in terms of the variables $\tilde{n}$ and Σ.

Usage

```
iwishmom_sym(f, iw = 0, alpha = 2, latex = FALSE)
```

Arguments

`f`	A vector of nonnegative integers f_j that represents the power of tr(W^{-j}), where $j = 1, \ldots, r$
`iw`	The power of the inverse beta-Wishart matrix W^{-1} (0 by default)
`alpha`	The type of Wishart distribution $(\alpha = 2/\beta)$: • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)
`latex`	A Boolean indicating whether the output will be a LaTeX string or dataframe (FALSE by default)

Value

When `iw = 0`, it returns an analytical expression of $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j}]$. When `iw != 0`, it returns an analytical expression of $E[\prod_{j=1}^{r} \text{tr}(W^{-j})^{f_j} W^{-iw}]$. If `latex = FALSE`, the output is a data frame that stores the coefficients for calculating the result. If `latex = TRUE`, the output is a LaTeX formatted string of the result in terms of $\tilde{n}$ and Σ.

Examples

```
# Example 1: For E[tr(W^{-1})^4] with W ~ W_m^1(n,Sigma), represented as a dataframe:
iwishmom_sym(4) # iw = 0, for real Wishart distribution

# Example 2: For E[tr(W^{-1})*tr(W^{-2})W^{-1}] with W ~ W_m^1(n,S), represented as a dataframe:
iwishmom_sym(c(1, 1), 1) # iw = 1, for real Wishart distribution

# Example 3: For E[tr(W^{-1})^4] with W ~ W_m^2(n,S), represented as a LaTeX string:
# Using writeLines() to format
writeLines(iwishmom_sym(4, 0, 1, latex=TRUE)) # iw = 0, for complex Wishart distribution
```

```
# Example 4: For E[tr(W^{-1})*tr(W^{-2})W^{-1}] with W ~ W_m^2(n,S), represented as a LaTeX string:
# Using writeLines() to format
writeLines(iwishmom_sym(c(1, 1), 1, 1, latex=TRUE)) # iw = 1, for real Wishart distribution
```

---

iwish_ps *Inverse of a Coefficient Matrix $\tilde{\mathcal{H}}$_k*

Description

This function computes the inverse of a coefficient matrix $\tilde{\mathcal{H}}_k$ that allows us to compute the expected value of a power-sum symmetric function of W^{-1}, where $W \sim W_m^\beta(n, \Sigma)$.

Usage

```
iwish_ps(k, alpha = 2)
```

Arguments

k	The order of the $\tilde{\mathcal{H}}_k$ matrix (a positive integer)
alpha	The type of Wishart distribution ($\alpha = 2/\beta$): • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

Inverse of a coefficient matrix $\tilde{\mathcal{H}}_k$ that allows us to compute the expected value of a power-sum symmetric function of W^{-1}, where $W \sim W_m^\beta(n, \Sigma)$. The matrix is represented as a 3-dimensional array where each slice along the third dimension represents a coefficient matrix of the polynomial in descending powers of $\tilde{n}$.

Examples

```
# Example 1:
iwish_ps(3) # For real Wishart distribution with k = 3

# Example 2:
iwish_ps(4, 1) # For complex Wishart distribution with k = 4

# Example 3:
iwish_ps(2, 1/2) # For quaternion Wishart distribution with k = 2
```

iwish_psr *Coefficient Matrix $\tilde{\mathcal{H}}$_k*

Description

This function computes the coefficient matrix $\tilde{\mathcal{H}}_k$ for $W \sim W_m^\beta(n, \Sigma)$.

Usage

```
iwish_psr(k, alpha = 2)
```

Arguments

k	The order of the $\tilde{\mathcal{H}}_k$ matrix (a positive integer)
alpha	The type of Wishart distribution ($\alpha = 2/\beta$): • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

A list with two elements:

- c: A 3-dimensional array containing the coefficient matrices of the numerator of $\tilde{\mathcal{H}}_k$ in descending powers of $n1$, where $n1 = n - m + 1 - \alpha$
- den: A vector containing the coefficients of the denominator of $\tilde{\mathcal{H}}_k$, in descending powers of $n1$

Examples

```
# Example 1:
iwish_psr(2) # For real Wishart distribution with k = 2

# Example 2:
iwish_psr(4, 1) # For complex Wishart distribution with k = 4

# Example 3:
iwish_psr(2, 1/2) # For quaterion Wishart distribution with k = 2
```

qkn_coeff *Inverse of a Coefficient Matrix $\tilde{\mathcal{C}}$_k*

Description

This function computes the inverse of the coefficient matrix $\tilde{\mathcal{C}}_k$

Usage

```
qkn_coeff(k, alpha = 2)
```

Arguments

k The order of the $\tilde{\mathcal{C}}_k$ matrix

alpha The type of beta-Wishart distribution ($\alpha = 2/\beta$):
- 1/2: Quaternion Wishart
- 1: Complex Wishart
- 2: Real Wishart (default)

Value

Inverse of a coefficient matrix $\tilde{\mathcal{C}}_k$ that allows us to obtain $E[p_\lambda(W^{-1})W^{-r}]$, where $r + |\lambda| = k$ and $W\ W_m^\beta(n, \Sigma)$. The matrix is represented as a 3-dimensional array where each slice along the third dimension represents a coefficient matrix of the polynomial in descending powers of $\tilde{n}$.

Examples

```
# Example 1:
qkn_coeff(2) # For real Wishart distribution with k = 2
# Example 2:
qkn_coeff(3, 1) # For complex Wishart distribution with k = 3

# Example 3:
qkn_coeff(2, 1/2) # For quaternion Wishart distribution with k = 2
```

qkn_coeffr *Coefficient Matrix $\tilde{C}$_k*

Description

This function computes the coefficient matrix for $\tilde{\mathcal{C}}_k$ for $W \sim W_m^\beta(n, \Sigma)$.

Usage

```
qkn_coeffr(k, alpha = 2)
```

Arguments

k	The order of the $\tilde{\mathcal{C}}_k$ matrix (a positive integer)
alpha	The type of Wishart distribution ($\alpha = 2/\beta$): • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

A list with two elements:

- c: A 3-dimensional array containing the coefficient matrices of the numerator of $\tilde{\mathcal{C}}_k$ in descending powers of $n1$, where $n1 = n - m + 1 - \alpha$.
- den: A vector containing the coefficients of the denominator of $\tilde{\mathcal{C}}_k$, in descending powers of $n1$.

Examples

```
# Example 1:
qkn_coeffr(2) # For real Wishart distribution with k = 2

# Example 2:
qkn_coeffr(3, 1) # For complex Wishart distribution with k = 3

# Example 3:
qkn_coeffr(2, 1/2) # For quaternion Wishart distribution with k = 2
```

---

qk_coeff *Coefficient Matrix C_k*

Description

This function computes the coefficient matrix $\mathcal{C}_k$, which is a matrix of constants that allows us to obtain $E[p_\lambda(W)W^r]$, where $r + |\lambda| = k$ and $W \sim W_m^\beta(n, \Sigma)$.

Usage

```
qk_coeff(k, alpha = 2)
```

Arguments

k	The order of the $\mathcal{C}_k$ matrix
alpha	The type of Wishart distribution ($\alpha = 2/\beta$): • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

$\mathcal{C}_k$, a matrix that allows us to obtain $E[p_\lambda(W)W^r]$, where $r + |\lambda| = k$ and $W \sim W_m^\beta(n, \Sigma)$. The matrix is represented as a 3-dimensional array where each slice along the third dimension represents a coefficient matrix of the polynomial in descending powers of n.

Examples

```
# Example 1:
qk_coeff(2) # For real Wishart distribution with k = 2

# Example 2:
qk_coeff(3, 1) # For complex Wishart distribution with k = 3

# Example 3:
qk_coeff(2, 1/2) # For quaternion Wishart distribution with k = 2
```

---

wishmom — *Expectation of a Matrix-valued Function of a beta-Wishart Distribution*

Description

When `iw = 0`, the function calculates $E[\prod_{j=1}^r \text{tr}(W^j)^{f_j}]$, where $W \sim W_m^\beta(n, S)$. When `iw != 0`, the function calculates $E[\prod_{j=1}^r \text{tr}(W^j)^{f_j} W^{iw}]$

Usage

```
wishmom(n, S, f, iw = 0, alpha = 2)
```

Arguments

`n`	The degrees of freedom of the beta-Wishart matrix W
`S`	The covariance matrix of the beta-Wishart matrix W
`f`	A vector of nonnegative integers f_j that represents the power of $\text{tr}(W^j)$, where $j = 1, \ldots, r$
`iw`	The power of the inverse beta-Wishart matrix W (0 by default)
`alpha`	The type of Wishart distribution $(\alpha = 2/\beta)$: • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)

Value

When `iw = 0`, it returns $E[\prod_{j=1}^r \text{tr}(W^j)^{f_j}]$. When `iw != 0`, it returns $E[\prod_{j=1}^r \text{tr}(W^j)^{f_j} W^{iw}]$.

Examples

```
# Example 1: For E[tr(W)^4] with W ~ W_m^1(n,S),
# where n and S are defined below:
n <- 20
S <- matrix(c(25, 49,
              49, 109), nrow=2, ncol=2)
wishmom(n, S, 4) # iw = 0, for real Wishart distribution

# Example 2: For E[tr(W)^2*tr(W^3)W^2] with W ~ W_m^1(n,S),
# where n and S are defined below:
n <- 20
S <- matrix(c(25, 49,
              49, 109), nrow=2, ncol=2)
wishmom(n, S, c(2, 0, 1), 2, 2) # iw = 2, for real Wishart distribution

# Example 3: For E[tr(W)^2*tr(W^3)] with W ~ W_m^2(n,S),
# where n and S are defined below:
# Hermitian S for the complex case
n <- 20
S <- matrix(c(25, 49 + 2i,
              49 - 2i, 109), nrow=2, ncol=2)
wishmom(n, S, c(2, 0, 1), 0, 1) # iw = 0, for complex Wishart distribution

# Example 4: For E[tr(W)*tr(W^2)^2*tr(W^3)^2*W] with W ~ W_m^2(n,S),
# where n and S are defined below:
n <- 20
S <- matrix(c(25, 49 + 2i,
              49 - 2i, 109), nrow=2, ncol=2)
wishmom(n, S, c(1, 2, 2), 1, 1) # iw = 1, for complex Wishart distribution
```

`wishmom_sym` *Symbolic Expectation of a Matrix-valued Function of a beta-Wishart Distribution*

Description

When `iw = 0`, the function returns an analytical expression of $E[\prod_{j=1}^r \mathrm{tr}(W^j)^{f_j}]$, where $W \sim W_m^\beta(n,S)$. When `iw != 0`, the function returns an analytical expression of $E[\prod_{j=1}^r \mathrm{tr}(W^j)^{f_j} W^{iw}]$. For a given `f`, `iw`, and `alpha`, this function provides the aforementioned expectations in terms of the variables n and Σ.

Usage

```
wishmom_sym(f, iw = 0, alpha = 2, latex = FALSE)
```

Arguments

`f`	A vector of nonnegative integers f_j that represents the power of tr(W^{-j}), where $j = 1, \ldots, r$
`iw`	The power of the beta-Wishart matrix W^{-1} (0 by default)
`alpha`	The type of Wishart distribution $(\alpha = 2/\beta)$: • 1/2: Quaternion Wishart • 1: Complex Wishart • 2: Real Wishart (default)
`latex`	A Boolean indicating whether the output will be a LaTeX string or a dataframe (FALSE by default)

Value

When `iw = 0`, it returns an analytical expression of $E[\prod_{j=1}^{r} \mathrm{tr}(W^j)^{f_j}]$. When `iw != 0`, it returns an analytical expression of $E[\prod_{j=1}^{r} \mathrm{tr}(W^j)^{f_j} W^{iw}]$. If `latex = FALSE`, the output is a data frame that stores the coefficients for calculating the result. If `latex = TRUE`, the output is a LaTeX formatted string of the result in terms of n and Σ.

Examples

```
# Example 1: For E[tr(W)^4] with W ~ W_m^1(n,Sigma), represented as a dataframe:
wishmom_sym(4) # iw = 0, for real Wishart distribution

# Example 2: For E[tr(W)*tr(W^2)W] with W ~ W_m^1(n,S), represented as a dataframe:
wishmom_sym(c(1, 1), 1) # iw = 1, for real Wishart distribution

# Example 3: For E[tr(W)^4] with W ~ W_m^2(n,S), represented as a LaTeX string:
# Using writeLines() to format
writeLines(wishmom_sym(4, 0, 1, latex=TRUE)) # iw = 0, for complex Wishart distribution

# Example 4: For E[tr(W)*tr(W^2)W] with W ~ W_m^2(n,S), represented as a LaTeX string:
# Using writeLines() to format
writeLines(wishmom_sym(c(1, 1), 1, 1, latex=TRUE)) # iw = 1, for real Wishart distribution
```

`wish_ps` *Coefficient Matrix $\mathcal{H}$_k*

Description

This function computes the coefficient matrix $\mathcal{H}_k$ that allows us to compute the expected value of a power-sum symmetric function of W, where $W \sim W_m^{\beta}(n, \Sigma)$.

Usage

```
wish_ps(k, alpha = 2)
```

Arguments

k
The order of the $\mathcal{H}_k$ matrix

alpha
The type of Wishart distribution ($\alpha = 2/\beta$):

- 1/2: Quaternion Wishart
- 1: Complex Wishart
- 2: Real Wishart (default)

Value

A coefficient matrix $\mathcal{H}_k$ that allows us to compute the expected value of a power-sum symmetric function of W, where $W \sim W_m^\beta(n, \Sigma)$. The matrix is represented as a 3-dimensional array where each slice along the third dimension represents a coefficient matrix of the polynomial in descending powers of n.

Examples

```
# Example 1:
wish_ps(3) # For real Wishart distribution with k = 3

# Example 2:
wish_ps(4, 1) # For complex Wishart distribution with k = 4

# Example 3:
wish_ps(2, 1/2) # For quaternion Wishart distribution with k = 2
```

Index