Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Invariant Moments of the Wishart Distribution: A Sage Package and Website Implementation

Carlos Antunes Percíncula
Universidad Nacional del Litoral

Liliana Forzani
Universidad Nacional del Litoral

Ricardo Toledano
Universidad Nacional del Litoral

Abstract

The moments (positive and negative) of order k of a Wishart distribution are functions of its parameters (i.e., degrees of freedom n and variance-covariance matrix Σ). These moments can be expressed symbolically as linear combinations of products of powers of Σ, $\mathrm{tr}\Sigma^k$ and n. The formulas require the computation of the partitions of the integer k and the coefficients of a certain class of Jack polynomials in the monomial and power-sum bases. These formulas are implemented in a SAGE package `WishartMoments` and the invariant moments are obtained at the website https://antunescarles.github.io/wishart-moments-calculator/ Antunes Percíncula (2022a). The formulas were developed by Letac and Massam (2004).

Keywords: Moments of Wishart and inverse Wishart, SAGE, website implementation.

1. Introduction

The Wishart distribution plays an essential role in multivariate statistics and, more recently, in random matrix theory. Almost 100 years after the original publication of Wishart (1928), it is still a subject of research and further study. Although the density of the Wishart distribution (when it exists) is very well known, the same is not true of the moments, which are usually needed by researchers in the field. Their problems often concern functions of Wishart matrices, such as the trace, moments of the trace, moments of the variable or combinations of them.

The efforts to compute these moments have a long history, beginning with Haff (1979, 1981), who, using a very clever identity for Wisharts, find the second moments and second inverse moments of a Wishart variable. Von Rosen (1988a) was among the first to work on formulas for moments of any order. He generalized an idea of Guiard (1986) and, with the help of differentiating the moment generating function, was able to explicitly compute moments of the form $E(\otimes_{i=1}^{j} W)$ for $j = 2, 3, 4$, giving recursive expressions for the higher order moments. It is important to clarify that computing these kind of moments, $E(\otimes_{i=1}^{j} W)$, allows us to get, in particular, $E(W^j)$ for $j \in \mathbb{N}$ and the expectation of other combinations of $W^i \mathrm{tr}^k(W^l)$ for $i+k*l = j$. He extended this result to the moments of the inverse of the Wishart distribution for the case of $n > 2k + r - 1$ in Von Rosen (1988b), where r is the dimension of the random matrix. Later, Von Rosen (1997) obtained formulas with the help of a factorization theorem, again giving recursive expressions for them. The history is much longer—for example, Sultan and Tracy (1996) gave explicit formulas for moments of order up to 4; while Lu and Richards (2001), using MacMahon's master theorem, obtained an explicit formula for the moments of arbitrary polynomials in the entries of a Wishart W.

All of these formulas, although explicit, are almost impossible to use for orders greater than, say, 4. Graczyk, Letac, and Massam (2003) and Letac and Massam (2004) compute invariant moments of the form $E(Q(W))$, where $Q(W)$ is a polynomial with respect to the entries of a Wishart W. Their moments are invariant in the sense that they depend only on the eigenvalues of W. In particular, they are able to compute the expected value of any power of W or its inverse W. Because they do not rely on traditional combinatorial methods but instead rely on the interplay between two bases of the space of invariant polynomials in W, all of the moments can be obtained through the multiplication of three matrices whose entries are obtained after computing the partitions of the integer k and the coefficients of a certain class of Jack polynomials in the monomial and power-sum bases. Although the authors were able to implement a Maple program to compute them, it is no longer available.

Later, Kuriki and Numata (2010) introduced another formula for the moments of the Wishart distribution. In their formula, the moments are described as special values of the weighted generating function of matchings of graphs. Matsumoto (2012) derived a formula for a general moment of Wishart and inverse of Wishart as a function of orthogonal Weingarten functions, giving again explicit formulas for moments up to order 3. Kim and Kang (2015) also gave explicit formulas for moments up to order 3. Bishop, Moral, and Angèle (2018) derived a polynomial formula for the invariant moments in a different way than that presented by Letac and Massam (2004). Nevertheless, if higher order moments are needed, they are still not available in the literature.

Moreover, there was a resurgence of the need for these moments due to the importance of random matrix theory. For example, for the proof of the consistency of partial least square regression in the context of r large and n small, Cook and Forzani (2018, 2019) rely on the computation of higher order moments of the Wishart distribution. According to Holgersson and Pielaszkiewicz (2020): *…the Wishart distribution is generally not closed under the transformation being conducted, and one may search the literature for available results on some specific moment. However, there presently does not seem to be any single source to consult in the matter, and it can be tedious to find the result sought for.* Consequently, they list a number of moments of functions of Wishart moments in a convenient format for easy access and in particular they give the invariant moments of order 3 in the list.

The purpose of this work is the implementation of symbolic computation, using the open source mathematical software SAGE (The Sage Developers 2022), of the formulas for the invariant moments of the Wishart distribution and its inverse, as given by Letac and Massam (2004). The formulas for the invariant models are implemented in the `WishartMoments` Antunes Percíncula (2022b). Using this package, the desired k moment and the symbolic moment are given in both LaTeX and pdf formats for a general Σ. Numerical results can be obtained when the numerical values for the parameters (n and Σ) are given.

Moreover, we provide a graphical user interface to compute the formulas for the moments in Antunes Percíncula (2022a). Given a desired value of k, a symbolic formula for the invariant moments can be obtained, while an explicit formula can be obtained when numerical values for the parameters are given.

2. Formulas for the invariant Wishart moments

We present now the formulas (see Theorem 1 below) which will be used for the computation of the invariant moments of order k of a Wishart distribution and its inverse in SAGE. We consider here a symmetric random matrix W of dimension r, a matrix $\Sigma \in \mathbb{S}^r_{\{+\}}$ (the set of positive definite matrices of dimension r) and $n \in \mathbb{N}$ such that $W \sim W_r(n, \Sigma)$. These formulas were derived by Letac and Massam (2004).

Let $\mathbb{S}_k$ be the symmetric group of permutations of $\{1, \ldots, k\}$ (Wikipedia contributors 2022). If $\pi \in \mathbb{S}_k$, let i_j be the number of cycles of π of length j for $j = 1, \ldots, k$. The k-tuple $(i) = (i_1, \ldots, i_k)$ is called the *portrait* of π and satisfies the equality $i_1 + 2i_2 + \cdots + ki_k = k$. The set of all portraits associated to $\mathbb{S}_k$ is denoted by I_k, that is

$$I_k = \{(i) = (i_1, \ldots, i_k) : \; i_j \in \mathbb{N}_0 \text{ such that } i_1 + 2i_2 + \cdots + ki_k = k\} \tag{1}$$

Let us note that the cardinal of I_k, that we denote by s, is smaller or equal than the number of permutations.

A *partition* λ of an integer k is a sequence of non-negative integers $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = k$. We will write $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ and the number $l = l(\lambda)$ of non-zero components λ_i of λ is called the *length* of the partition λ. It is a common practice to write down only the non zero components of a partition. Notice that for each portrait $(i) \in I_k$ we have a partition of k associated to (i), namely

$$\lambda_{(i)} := (\overbrace{k, \ldots, k}^{i_k}, \overbrace{k-1, \ldots, k-1}^{i_{k-1}}, \ldots, \overbrace{1, \ldots, 1}^{i_1}) \tag{2}$$

where $\overbrace{j, \ldots, j}^{i_j}$ indicates that j appears i_j times, and if $i_j = 0$ then j is excluded.

In this work we will use two orderings for the set of partitions of an integer k: let $\lambda = (\lambda_1, \ldots, \lambda_k)$ and $\mu = (\mu_1, \ldots, \mu_k)$ be two partitions of k. On the one hand we say that $\mu \leq \lambda$ in the *lexicographic* order if either $\mu = \lambda$ (that is $\mu_i = \lambda_i$ for $i = 1, \ldots, k$) or, if for the largest integer $j < k$ such that $\mu_i = \lambda_i$ for $i = 1, \ldots, j$ we have that $\mu_{j+1} < \lambda_{j+1}$. On the other hand, we say that $\mu \preccurlyeq \lambda$ in the *dominance* order if

$$\mu_1 + \cdots + \mu_i \leq \lambda_1 + \cdots + \lambda_i,$$

for $i = 1, \ldots, k$. Notice that the lexicographic order is a total order but the dominance order is not: for example $\lambda = (5, 1, 1, 1)$ and $\mu = (4, 3, 1)$ are two partitions of 8 that can not be compared with the dominance order, that is neither $\lambda \preccurlyeq \mu$ nor $\mu \preccurlyeq \lambda$.

For each $(i) \in I_k$ and $h \in \mathbb{S}^r_{\{+\}}$, we define

$$r_{(i)}(h) = \prod_{j=1}^{k} (\mathrm{tr} h^j)^{i_j} \quad (3)$$

$$L_{r_{(i)}}(h) = r_{(i)}(h) \sum_{j=1}^{k} j i_j \frac{h^j}{\mathrm{tr}(h^j)} \quad (4)$$

where $\mathrm{tr}(A)$ indicates the trace of the matrix A. For instance, if we consider $(i) = (0, \dots, 0, 1) \in I_k$ we have that $\lambda_{(0\dots,0,1)} = (k)$ and then

$$r_{(0\dots,0,1)}(h) = (\mathrm{tr} h)^k \quad (5)$$

$$L_{r_{(0\dots,0,1)}}(h) = k h^k. \quad (6)$$

Now, for a partition λ of $k \in \mathbb{N}$ and integers $d, q \in \mathbb{N}$, we define

$$(d)_\lambda = \prod_{j=1}^{l(\lambda)} \prod_{t=1}^{\lambda_j} (d + t - 1 - \frac{j-1}{2}) \quad (7)$$

$$(q)^*_\lambda = [\prod_{j=k-l(\lambda)+1}^{k} \prod_{t=1}^{\lambda_{k-j+1}} (q + \frac{k-j+1}{2} - t)]^{-1}. \quad (8)$$

We define now three matrices which are the key ingredients of the formulas appearing in Theorem 1 below. These formulas are the ones we will use to compute the invariant moments (of a given order) of a Wishart distribution. First we have the matrix $D_k(d)$ which is the diagonal matrix with diagonal $(d)_\lambda$, where the λ´s are all the partitions of k, ordered in ascending lexicographic order. Then we have the diagonal matrix $D^*_k(q)$ which is analogous to $D_k(d)$ but now $(q)^*_\lambda$ is used. Finally we have an invertible matrix denoted by B_k, whose definition depends on the coefficients of a certain class of symmetric polynomials called Jack polynomials. The precise definition of B_k is given in the next section. We are now in a position to present the theoretical result, given in Letac and Massam (2004)[Theorem 4], that we use to compute the invariant moments of order k of a Wishart distribution and its inverse.

Theorem 1 *Let $W \sim W_r(n, \Sigma)$. For a given $k \in \mathbb{N}$, let $\{L_{r_{(i)}}(h)\}_{I_k}$ be the column vector of components $L_{r_{(i)}}(h)$ with $(i) \in I_k$, ordered according to the ascending lexicographic order of the associated partitions. Then for $n \in \mathbb{N}$,*

$$\{E(L_{r_{(i)}})(W)\}_{I_k} = B_k^{-1} D_k(n/2) B_k \{L_{r_{(i)}}(2\Sigma)\}_{I_k} \quad (9)$$

$$\{E(L_{r_{(i)}})(W^{-1})\}_{I_k} = B_k^{-1} D^*_k((n-r)/2) B_k \{L_{r_{(i)}}((2\Sigma)^{-1})\}_{I_k} \text{ for } n > 2k + r - 1 \quad (10)$$

where $E(A)$ represents the expectation of the random variable A.

In particular by (5) and (6), the last row of (9) gives $kE(W^k)$ and the last row of (10) gives $kE(W^{-k})$.

3. The matrix B_k

A polynomial f in the variables $x_1, \ldots, x_k$ (with real coefficients) is called *symmetric* if it is invariant under any permutation of its variables, that is $f(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$ for any $\pi \in \mathbb{S}_k$. It is called *homogeneous of degree k* if $f(\alpha x_1, \ldots, \alpha x_k) = \alpha^k f(x_1, \ldots, x_k)$ for any $0 \neq \alpha \in \mathbb{R}$. We denote by $\mathbb{V}_k$ the set of all homogeneous symmetric polynomials of degree k with real coefficients together with the zero polynomial. In what follows we mention some well known facts about $\mathbb{V}_k$ and we refer the reader to Macdonald (2015) not only for these facts but also for a comprehensive treatment of symmetric polynomials and functions. First of all we have that $\mathbb{V}_k$ is an $\mathbb{R}$-vector space. Second there are several standard bases for this vector space which are frequently used by the specialists in this field, but here we will just need two of them: the monomial and the powe-sum basis. The so called power-sum basis is the set $\{p_\lambda : \lambda \text{ partition of } k\}$ where

$$p_\lambda = \sum_{j=1}^{k} x_j^k,$$

if the length $l(\lambda) = 1$ (so that $\lambda = (k)$) and

$$p_\lambda = \prod_{i=1}^{l(\lambda)} \sum_{j=1}^{k} x_j^{\lambda_i},$$

if the length $l(\lambda) > 1$. On the other hand we have the so called monomial basis of $\mathbb{V}_k$ formed by the monomials

$$m_\lambda = \sum_{\sigma \in S_\lambda} x_{\sigma(1)}^{\lambda_1} x_{\sigma(2)}^{\lambda_2} \cdots x_{\sigma(k)}^{\lambda_k},$$

where λ runs over all partitions of k and $S_\lambda \subset \mathbb{S}_k$ is the set of permutations giving distinct terms in the sum.

Given a real parameter $\alpha > 0$ there is an inner product $\langle , \rangle_\alpha$ in $\mathbb{V}_k$ such that

$$\langle p_\lambda, p_\mu \rangle_\alpha = \delta_{\lambda\mu} z_\lambda \alpha^{l(\lambda)},$$

where $\delta_{\lambda\mu} = 0$ if $\lambda \neq \mu$ and $\delta_{\lambda\mu} = 1$ if $\lambda = \mu$ and

$$z_\lambda = (1^{m_1} 2^{m_2} \cdots k^{m_k}) m_1! m_2! \cdots m_k!,$$

where $m_i = |\{j : \lambda_j = i\}|$.

We introduce next the so called Jack polynomials $J_\lambda^{(\alpha)}$. It is proved in Chapter VI of Macdonald (2015) that there are unique polynomials $J_\lambda^\alpha \in \mathbb{V}_k$ such that

$$\langle J_\lambda^{(\alpha)}, J_\mu^{(\alpha)} \rangle_\alpha = 0,$$

if $\lambda \neq \mu$ and

$$J_\lambda^{(\alpha)} = \sum_{\mu \preccurlyeq \lambda} c_{\lambda\mu} m_\mu, \tag{11}$$

with $c_{\lambda 1^k} = k!$. Notice that in the sum (11) the dominance ordering for partitions is used.

One of the important properties of Jack polynomials is that several classical families of symmetric polynomials are obtained by choosing specific values of the parameter α. In particular, when $\alpha = 2$ the well known Zonal polynomials Z_λ (with the appropriate normalization as in James (1964)) are obtained.

Now we describe the construction of the matrix B_k that appears in formulas (9) and (10): roughly speaking it is a matrix formed by the coefficients of the Jack polynomials with $\alpha = 2$ expressed in terms of the power sum basis. First we consider the set of all partitions of k and we use the lexicographic ordering to write them in ascending order from left to right

$$\lambda^{(1)} = (1^k) < \lambda^{(2)} < \cdots < \lambda^{(s)} = (k),$$

where (1^k) is the partition $(1, 1, \ldots, 1)$ of k. Next we consider the Jack polynomials $J^{(2)}_{\lambda^{(i)}}$ for $i = s, s-1, \ldots, 1$ and from (11) we can write

$$\begin{aligned} J^{(2)}_{\lambda^{(s)}} &= k! m_{\lambda^{(1)}} + c_{1,2} m_{\lambda^{(2)}} + c_{1,3} m_{\lambda^{(3)}} + \cdots + c_{1,s-1} m_{\lambda^{(s-1)}} + c_{1,s} m_{\lambda^{(s)}}, \\ J^{(2)}_{\lambda^{(s-1)}} &= k! m_{\lambda^{(1)}} + c_{2,2} m_{\lambda^{(2)}} + c_{2,3} m_{\lambda^{(3)}} + \cdots + c_{2,s-1} m_{\lambda^{(s-1)}}, \\ &\vdots \\ J^{(2)}_{\lambda^{(1)}} &= k! m_{\lambda^{(1)}}. \end{aligned} \tag{12}$$

Then we change to the power sum basis and we get expressions of the form

$$\begin{aligned} J^{(2)}_{\lambda^{(s)}} &= b_{1,1} p_{\lambda^{(1)}} + b_{1,2} p_{\lambda^{(2)}} + b_{1,3} p_{\lambda^{(3)}} + \cdots + b_{1,s} p_{\lambda^{(s)}}, \\ J^{(2)}_{\lambda^{(s-1)}} &= b_{2,1} p_{\lambda^{(1)}} + b_{2,2} p_{\lambda^{(2)}} + b_{2,3} m_{\lambda^{(3)}} + \cdots + b_{2,s} p_{\lambda^{(s)}}, \\ &\vdots \\ J^{(2)}_{\lambda^{(1)}} &= b_{s,1} p_{\lambda^{(1)}} + b_{s,2} p_{\lambda^{(2)}} + b_{s,3} p_{\lambda^{(3)}} + \cdots + b_{s,s} p_{\lambda^{(s)}}. \end{aligned} \tag{13}$$

The matrix B_k is the $s \times s$ matrix $B_k = (b_{ij})$. In our algorithms we compute all the Jack polynomials in (12) using the Zonal polynomials expressed in terms of the monomial basis. In SAGE the Zonal polynomials and the coefficients in the monomial basis can be obtained using the `SymmetricFunctions` module. Once they are calculated, we normalize each Zonal polynomial dividing it by the coefficient corresponding to the partition (1^k), obtaining the polynomials defined in (11). We then use an already implemented feature of the `SymmetricFunctions` module to make the change to the power-sum basis and thus (13) is obtained.

4. SAGE package to compute all invariant Wishart moments

The main component of the `WishartMoments` package consists of the `Expectations` class. This class is designed to keep all of the information needed to compute the Wishart moment associated to an integer k.

To instantiate an object of the `Expectations` class, the user is required to give a positive integer k (the order of the moments that the user wants to compute) as an argument to its constructor.

This class internally computes and stores the expressions $L_{r_{(i)}}(h)$ for every portrait $(i) \in I_k$, as well as the matrices B_k, B_k^{-1}, $D_k(n/2)$ and $D_k((n-r)/2)$, which are later used to symbolically compute the expressions for every invariant moment of order k of a symbolic variable $W \sim W_r(n, \Sigma)$ or those of its inverse W^{-1}. The dimension r and the parameters n and Σ of the distribution will appear in the expression for the moment as symbolic variables and they need not be given. Although the matrices B_k, B_k^{-1}, $D_k(n/2)$ and $D_k((n-r)/2)$ need to be computed when an instance is constructed, the expressions for the moments are only computed when they are required by the user for the first time. This means that several calls to the method that calculate them are only able to retrieve the stored result of the first call.

The results of the class `Expectations` can be requested through the methods `expressions`, `moment`, `pretty_print_moment`, `evaluate_moment` and `pretty_print_eval_moment`.

The `expressions` method returns a list of all the expressions $L_{r_{(i)}}(W)$, to let the user know which Wishart moments can be computed for a given k. The output is a list of 2-element lists, where the first element is the index of a given portrait (i) with respect to the order induced by the ascending lexicographic order of the associated partitions, and the second element is the symbolic expression of $L_{r_{(i)}}(W)$ itself for the portrait (i). There are as many lists as elements of I_k.

The computation of the moments is carried by the `moment` method. This is needed because it gives the index of the portrait (i) (the one appearing in the result of the `expressions` method) for which the user wants to compute the expressions for the expectation of $L_{r(i)}(W)$. This method also allows the user to compute the expectation of $L_{r(i)}(W^{-1})$ by passing the optional argument `inverse` set to `True`. The built-in methods in SAGE to deal with Zonal polynomials are used to compute the matrix B_k that is needed for this step. The `moment` method returns the symbolic expression corresponding to $L_{r_{(i)}}(W)$ and $E[L_{r_{(i)}}(W)]$ as values of a dictionary corresponding to the keys `'var'` and `'moment'`, respectively. If the argument `inverse=True` is passed to the method, then it will return $L_{r_{(i)}}(W^{-1})$ and $E[L_{r_{(i)}}(W^{-1})]$. The results were validated by comparison with the moments available in the literature.

We can obtain the string with the LaTeX code representing any of these expressions by using the built-in function `latex`.

A nicer presentation of the moments can be displayed when the package is used inside a Jupyter Notebook, using `pretty_print_moment`. This method takes an index of a portrait as argument and displays the formula corresponding to the `moment` method as it would appear in a document when copying its LaTeX code.

It is also possible to get numerical results for the moment of a variable $W \sim W_r(n, \Sigma)$ when concrete values for n and Σ are provided. This can be done by using the `evaluate_moment` method after giving the index of the desired moment, as in the previously explained methods, and concrete values of k, n and Σ. The value for Σ is given as a Numpy `ndarray`, and the return value is a dictionary with the corresponding variable given as a value for the `'var'` key, and a `ndarray` for the its expectation given as a value of the `'moment'` key. A nicer presentation of the results, when using a Jupyter Notebook, can be obtained by the `pretty_print_eval_moment` method.

5. Installation of the package

This section assumes that the user has already installed a version of SAGE, which can be downloaded from https://www.sagemath.org/. To install the `WishartMoments` package, the user must open a Sage shell and install the package via the `pip` utility:

```
(sage-sh)   sage -pip install WishartMoments
```

To use the package, it is only necessary to import it as any other Python package

```
import WishartMoments as wm
```

6. Examples of computing the Wishart moments

We will now show how to compute the following Wishart moments of order 3:

$$\mathbb{E}(W^3) = (n^3 + 3\,n^2 + 4\,n)\,\Sigma^3 + \left(2\,(n^2+n)(\operatorname{tr}\Sigma)\right)\Sigma^2 + \Big(n(\operatorname{tr}\Sigma)^2 + (n^2+n)(\operatorname{tr}\Sigma^2)\Big)\,\Sigma$$
$$\mathbb{E}(W(\operatorname{tr}W)^2) = 8\,n\Sigma^3 + 4\,n^2(\operatorname{tr}\Sigma)\Sigma^2 + \Big(n^3(\operatorname{tr}\Sigma)^2 + 2\,n^2(\operatorname{tr}\Sigma^2)\Big)\,\Sigma$$

and

$$\mathbb{E}(W^{-3}) = \frac{(n-r-1)\Sigma^{-3}}{(n-r+1)(n-r)(n-r-3)(n-r-5)}\frac{2\,\Sigma^{-2}(\operatorname{tr}\Sigma^{-1})}{(n-r+1)(n-r)(n-r-3)(n-r-5)} + \frac{\Big(2\,(\operatorname{tr}\Sigma^{-1})^2 + n(\operatorname{tr}\Sigma^{-2}) - r(\operatorname{tr}\Sigma^{-2}) - (\operatorname{tr}\Sigma^{-2})\Big)\Sigma^{-1}}{(n-r+1)(n-r)(n-r-1)(n-r-3)(n-r-5)}$$

After importing the Wishart moments package (`import WishartMoments as wm`), we have to create an instance of the class `Expectations`.

```
sage: k=3
sage: expec = wm.Expectations(k)
```

We need to know how to reference the expressions of which we want to compute their expectations. We can get a list the expressions of order k by using the method `expressions` of `Expectations`, which returns a list of 2-element lists with the index of the portrait and the the expression for expectation of the moment corresponding to it.

```
sage: expec.expressions()
```

```
[0, W*tr(W, 1)^2]
[1, 2/3*W^2*tr(W, 1) + 1/3*W*tr(W, 2)]
[2, W^3]
```

Here `tr(A,j)` represents $\operatorname{tr}(A^j)$. Therefore, to get $W(\operatorname{tr}W)^2$ we call the method `moment` with the index `0`

```
sage: expec.moment(0)
```

```
{
        'var': W*tr(W, 1)^2 ,
        'moment': 8*n*S^3 + 4*n^2*tr(S, 1)*S^2 + (n^3*tr(S, 1)^2
        + 2*n^2*tr(S, 2))*S
}
```

Similarly we use the index `2` to get W^3.

```
sage: expec.moment(2)
```

```
{
        'var': W^3 ,
        'moment': (n^3 + 3*n^2 + 4*n)*S^3 + (2*(n^2 + n)*tr(S, 1))*S^2
        + (n*tr(S, 1)^2 + (n^2 + n)*tr(S, 2))*S
}
```

As for the moment of the inverse, we can call `moment` with the argument `inverse` set to `True`.

```
sage: expec.expressions(inverse = True)
```

```
[0, inv(W, 1)*tr(W, -1)^2]
[1, 2/3*inv(W, 2)*tr(W, -1) + 1/3*inv(W, 1)*tr(W, -2)]
[2, inv(W, 3)]
```

Here `inv(A,j)` represents A^{-j}. We use the index `2` to get W^{-3}.

```
sage: expec.moment(2, inverse = True)
```

```
{
        'var': inv(W, 3) ,
        'moment': (n - r - 1)*inv(S, 3)/((n - r + 1)*(n - r)
          *(n - r - 3)*(n - r - 5))
        + 2*inv(S, 2)*tr(S, -1)/((n - r + 1)*(n - r)*(n - r - 3)*(n - r - 5))
        + (2*tr(S, -1)^2 + n*tr(S, -2) - r*tr(S, -2)
        - tr(S, -2))*inv(S, 1)/((n - r + 1)*(n - r)*(n - r - 1)
          *(n - r - 3)*(n - r - 5))
}
```

We can obtain the string with the LaTeX code representing these expressions by using the built-in function `latex`. For instance, if we want to get the code for the variable $W(\mathrm{tr}\, W)^2$, we should use the following commands

```
sage: latex(expec.moment(0)['var'])
```

```
W {(\mathrm{tr} \, W)}^{2}
```

and for its expectation, $\Sigma n^3(\mathrm{tr}\,\Sigma)^2 + 8\,\Sigma^3 n + 2\,(2\,\Sigma^2(\mathrm{tr}\,\Sigma) + \Sigma(\mathrm{tr}\,\Sigma^2))n^2$,

```
sage: latex(expec.moment(0)['moment'])
```

```
8 \, {n} \Sigma^{3} + 4 \, {n}^{2} {(\mathrm{tr} \, {\Sigma})} \Sigma^{2}
+ \left({n}^{3} {(\mathrm{tr} \, {\Sigma})}^{2}
+ 2 \, {n}^{2} {(\mathrm{tr} \, {\Sigma}^{2})}\right) \Sigma
```

Notice that an instance of the form `wm.Expectations(3)` only permits to compute moments of order 3. To compute a moment of a different order, say $k = 4$, the user has to instantiate a new object of the class `wm.Expectations(4)`. We will continue with the examples using the same parameter as we were doing so far, that is `k=3`, so that can keep using the same object `expec`.

Now we show how to compute the numerical value of the moment $E(W\mathrm{tr}(W^2))$ for a Wishart distributions with parameters $n = 10$ and

$$\Sigma = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$$

We want to remark that neither the package nor the website will check if the matrix Σ is positive definite.

We first set the matrix Σ:

```
sage: Sigma = np.array([[4,1],[1,3]]);
```

To evaluate the moment we use the `evaluate_moment` where the parameters are `t` (the index of the expression in the list `expec.expressions()` for which we require its expectation), `n_param` (the numerical value for the parameter n), `Sigma` (the numerical value for the matrix Σ) and the boolean parameter `inverse` (`False` will compute the moment of W and `True` the moment for W^{-1}). Here we need to compute $E(W\mathrm{tr}(W^2))$ and therefore the parameters are passed as follows.

```
sage: ev = expec.evaluate_moment(t=0, n_param=10, Sigma=Sigma, inverse=False);
```

As `ev` is a dictonary, we can retrieve the variable by using the key `'var'`

```
sage: ev['var']
```

```
W*tr(W, 1)^2
```

To get the moment we use the key `'moment'`

```
sage: ev['moment']
```

```
array([[813600., 231120.],
       [231120., 582480.]], dtype=object)
```

7. The web interface

The main page is devoted to the calculation of the symbolic expressions. To compute the moment of order k of a random variable $W \sim W_r(n, \Sigma)$, it is necessary to provide the input k in the box. The interface provides a slider that allows the user to choose among the possible portraits associated with k. After that, the user can decide whether to require the Wishart moment associated to the selected partition for W, W^{-1} or both. The expression $L_{r_{(i)}}(W)$ and its expectations in terms of n and Σ are then displayed. This result is the same as that obtained by calling the `pretty_print_moment` method. By default, the options in both pages are set to compute the moment of order $k = 2$ for the greatest portrait $(0, 0, \dots, 1)$ and only for the variable W.

A link to the page where these expressions can be evaluated is presented on the main page. On that page, the user is required to specify the value of the matrix Σ as a Numpy `ndarray`. Once this code is submitted with the `compute` button, a new interactive box is displayed, and the user must provide the integers k and n, and then select whether to compute the moment for W, W^{-1} or both. If the user wants to compute the moment for W^{-1}, then the interface will require the user that the value for n satisfies the condition $n > 2k + (r - 1)$ for the inverse W^{-1} to exist.

Acknowledgement

We would like to thank Magister Laura Badela who greatly improved the presentation of the website for computation of the moments.

References

Antunes Percíncula CS (2022a). "Wishart Moments Calculator." https://antunescarles.github.io/wishart-moments-calculator/.

Antunes Percíncula CS (2022b). "Wishart Moments Calculator." URL https://github.com/antunescarles/wishart-moments-calculator.

Bishop A, Moral PD, Angèle N (2018). "An Introduction to Wishart Matrix Moments." *Found. Trends Mach. Learn.*, **11**, 97–218.

Cook RD, Forzani L (2018). "Big data and partial least-squares prediction." *Canadian Journal of Statistics*, **46**(1), 62–78. doi:https://doi.org/10.1002/cjs.11316. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cjs.11316, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cjs.11316.

Cook RD, Forzani L (2019). "Partial least squares prediction in high-dimensional regression." *The Annals of Statistics*, **47**(2), 884 – 908. doi:10.1214/18-AOS1681. URL https://doi.org/10.1214/18-AOS1681.

Graczyk P, Letac G, Massam H (2003). "The Complex Wishart Distribution and the Symmetric Group." *The Annals of Statistics*, **31**(1), 287–309. ISSN 00905364. URL http://www.jstor.org/stable/3448376.

Guiard V (1986). "A general formula for the central mixed moments of the multivariate normal distribution." *Statistics*, **17**(2), 279–289. doi:10.1080/02331888608801937. https://doi.org/10.1080/02331888608801937, URL https://doi.org/10.1080/02331888608801937.

Haff L (1979). "An identity for the Wishart distribution with applications." *Journal of Multivariate Analysis*, **9**(4), 531–544. ISSN 0047-259X. doi:https://doi.org/10.1016/0047-259X(79)90056-3. URL https://www.sciencedirect.com/science/article/pii/0047259X79900563.

Haff LR (1981). "Further identities for the Wishart distribution with applications in regression." *Canadian Journal of Statistics*, **9**(2), 215–224. doi:https://doi.org/10.2307/3314615. https://onlinelibrary.wiley.com/doi/pdf/10.2307/3314615, URL https://onlinelibrary.wiley.com/doi/abs/10.2307/3314615.

Holgersson T, Pielaszkiewicz J (2020). "A Collection of Moments of the Wishart Distribution." In *Recent Developments in Multivariate and Random Matrix Analysis : Festschrift in Honour of Dietrich von Rosen*, pp. 147–162. Springer International Publishing. ISBN 9783030567736. doi:10.1007/978-3-030-56773-6_9. URL http://libris.kb.se/bib/kxp8t113hwfcr2tv.

James AT (1964). "Distributions of matrix variates and latent roots derived from normal samples." *Ann. Math. Statist.*, **35**, 475–501. ISSN 0003-4851. doi:10.1214/aoms/1177703550. URL https://doi.org/10.1214/aoms/1177703550.

Kim C, Kang C (2015). "Determinantal Expression and Recursion for Jack Polynomials." *Applied Mathematical Sciences*, **9**(73), 3643–3649. URL http://dx.doi.org/10.12988/ams.2015.52173.

Kuriki S, Numata Y (2010). "Graph presentations for moments of noncentral Wishart distributions and their applications." *Annals of the Institute of Statistical Mathematics*, **62**(4), 645–672.

Letac G, Massam H (2004). "All Invariant Moments of the Wishart Distribution." *Scandinavian Journal of Statistics*, **31**(2), 295–318. doi:https://doi.org/10.1111/j.1467-9469.2004.01-043.x. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9469.2004.01-043.x, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9469.2004.01-043.x.

Lu I, Richards D (2001). "MacMahon's master theorem, representation theory, and moments of Wishart distributions." *Advances in Applied Mathematics*, **27**(2-3), 531–547. ISSN 0196-8858. doi:10.1006/aama.2001.0748. Funding Information: 1Research supported in part by a grant to the Institute for Advanced Study from the Bell Fund and by the National Science Foundation under Grant DMS-9703705.

Macdonald IG (2015). *Symmetric functions and Hall polynomials*. Oxford Classic Texts in the Physical Sciences, second edition. The Clarendon Press, Oxford University Press, New York. ISBN 978-0-19-873912-8. With contribution by A. V. Zelevinsky and a foreword by Richard Stanley, Reprint of the 2008 paperback edition [MR1354144].

Matsumoto S (2012). "General Moments of the Inverse Real Wishart Distribution and Orthogonal Weingarten Functions." *Journal of Theoretical Probability*, **25**(3), 798–822. doi:10.1007/s10959-011-0340-0. URL https://doi.org/10.1007/s10959-011-0340-0.

Sultan SA, Tracy DS (1996). "Moments of wishart distribution." *Stochastic Analysis and Applications*, **14**(2), 237s–243. doi:10.1080/07362999608809436. https://doi.org/10.1080/07362999608809436, URL https://doi.org/10.1080/07362999608809436.

The Sage Developers (2022). *SageMath, the Sage Mathematics Software System (Version 9.0)*. https://www.sagemath.org.

Von Rosen D (1988a). "Moments for matrix normal variables." *Statistics*, **19**(4), 575–583. doi:10.1080/02331888808802132. https://doi.org/10.1080/02331888808802132, URL https://doi.org/10.1080/02331888808802132.

Von Rosen D (1988b). "Moments for the Inverted Wishart Distribution." *Scandinavian Journal of Statistics*, **15**(2), 97–109. ISSN 03036898, 14679469. URL http://www.jstor.org/stable/4616090.

Von Rosen D (1997). "On Moments of the Inverted Wishart Distribution." *Statistics*, **30**(3), 259–278. doi:10.1080/02331889708802613. https://doi.org/10.1080/02331889708802613, URL https://doi.org/10.1080/02331889708802613.

Wikipedia contributors (2022). "Permutation — Wikipedia, The Free Encyclopedia." [Online; accessed 21-February-2022], URL https://en.wikipedia.org/w/index.php?title=Permutation&oldid=1072383198.

Wishart J (1928). "The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population." **20A**(1/2), 32–52. ISSN 0006-3444 (print), 1464-3510 (electronic). doi:https://doi.org/10.2307/2331939. URL http://www.jstor.org/stable/2331939.

Affiliation:

Carlos Antunes Percíncula
Departamento de Matemática
Facultad de Ingeniería Química
Universidad Nacional del Litoral
Santiago del Estero 2829
3000 Santa Fe, Argentina
E-mail: antunes.p.carlos@hotmail.com

Liliana Forzani
Departamento de Matemática
Facultad de Ingeniería Química
Universidad Nacional del Litoral and CONICET
Santiago del Estero 2829
3000 Santa Fe, Argentina
E-mail: liliana.forzani@gmail.com

Ricardo Toledano
Departamento de Matemática
Facultad de Ingeniería Química
Universidad Nacional del Litoral
Santiago del Estero 2829
3000 Santa Fe, Argentina
E-mail: ridatole@gmail.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd