L3a Eigenvalue Decomposition - Definitions and Facts

Ivan Slapničar

April 9, 2018

1 Eigenvalue Decomposition - Definitions and Facts

1.1 Prerequisites

The reader should be familiar with basic linear algebra concepts.

1.2 Competences

The reader should be able to understand and check the facts about eigenvalue decomposition.

1.3 Selected references

There are many excellent books on the subject. Here we list a few:

References

[Dem97] J.W. Demmel, 'Applied Numerical Linear Algebra', SIAM, Philadelphia, 1997.

[GV13] G. H. Golub and C. F. Van Loan, 'Matrix Computations', 4th ed., The John Hopkins University Press, Baltimore, MD, 2013.

[Hig02] N. Higham, 'Accuracy and Stability of Numerical Algorithms', SIAM, Philadelphia, 2nd ed., 2002.

[Hog14] L. Hogben, ed., 'Handbook of Linear Algebra', CRC Press, Boca Raton, 2014.

[Par80] B. N. Parlett, 'The Symmetric Eigenvalue Problem', Prentice-Hall, Englewood Cliffs, NJ, 1980, also SIAM, Philadelphia, 1998.

[Ste01] G. W. Stewart, 'Matrix Algorithms, Vol. II: Eigensystems', SIAM, Philadelphia, 2001.

[TB97] L. N. Trefethen and D. Bau, III, 'Numerical Linear Algebra', SIAM, Philadelphia, 1997.

[Wil65] J. H. Wilkinson, 'The Algebraic Eigenvalue Problem', Clarendon Press, Oxford, U.K., 1965.

1.4 General matrices

For more details and the proofs of the Facts below, see Section [DeA14] and the references therein.

References

[DeA14] L. M. DeAlba, Determinants and Eigenvalues, in: L. Hogben, ed., 'Handbook of Linear Algebra', pp. 4.1-4.15, CRC Press, Boca Raton, 2014.

1.4.1 Definitions

We state the basic definitions:

Let $F = \mathbb{R}$ or $F = \mathbb{C}$ and let $A \in F^{n \times n}$ with elements $a_{ij} \in F$.

An element $\lambda \in F$ is an **eigenvalue** of A if $\exists x \in F$, $x \neq 0$ such that

$$Ax = \lambda x,$$

and x is an **eigenvector** of λ.

Characteristic polynomial of A is $p_A(x) = \det(A - xI)$.

Algebraic multiplicty, $\alpha(\lambda)$, is the multiplicity of λ as a root of $p_A(x)$.

Spectrum of A, $\sigma(A)$, is the multiset of all eigenvalues of A, with each eigenvalue appearing $\alpha(A)$ times.

Spectral radius of A is

$$\rho(A) = \max\{|\lambda|, \lambda \in \sigma(A)\}.$$

Eigenspace of λ is

$$E_\lambda(A) = \ker(A - \lambda I).$$

Geometric multiplicity of λ is

$$\gamma(\lambda) = \dim(E_\lambda(A)).$$

λ is **simple** if $\alpha(\lambda) = 1$.

λ is **semisimple** if $\alpha(\lambda) = \gamma(\lambda)$.

A is **nonderogatory** if $\gamma(\lambda) = 1$ for all λ.

A is **nondefective** if every λ is semisimple.

A is **diagonalizable** if there exists nonsingular B matrix and diagonal matrix D such that $A = BDB^{-1}$.

Trace of A is

$$\operatorname{tr}(A) = \sum_i a_{ii}.$$

$Q \in \mathbb{C}^{n \times n}$ is **unitary** if $Q^*Q = QQ^* = I$, where $Q^* = (\bar{Q})^T$.

Schur decomposition of A is $A = QTQ^*$, where Q is unitary and T is upper triangular.
A and B are **similar** if $B = QAQ^{-1}$ for some nonsingular matrix Q.
A is **normal** if $AA^* = A^*A$.

1.4.2 Facts

There are many facts related to the eigenvalue problem for general matrices. We state some basic ones:

1. **Cayley-Hamilton Theorem.** $\lambda \in \sigma(A) \Leftrightarrow p_A(\lambda) = 0$.
2. $p_A(A) = 0$.
3. A simple eigenvalue is semisimple.
4. $\mathrm{tr}(A) = \sum_{i=1}^n \lambda_i$.
5. $\det(A) = \prod_{i=1}^n \lambda_i$.
6. A is singular $\Leftrightarrow \det(A) = 0 \Leftrightarrow 0 \in \sigma(A)$.
7. If A is triangular, $\sigma(A) = \{a_{11}, a_{22}, \ldots, a_{nn}\}$.
8. For $A \in \mathbb{C}^{n\times n}$, $\lambda \in \sigma(A) \Leftrightarrow \bar{\lambda} \in \sigma(A^*)$.
9. **Corollary of the Fundamental theorem of algebra.** For $A \in \mathbb{R}^{n\times n}$, $\lambda \in \sigma(A) \Leftrightarrow \bar{\lambda} \in \sigma(A)$.
10. If (λ, x) is an eigenpair of a nonsingular A, then $(1/\lambda, x)$ is an eigenpair of A^{-1}.
11. Eigenvectors corresponding to distinct eigenvalues are linearly independent.
12. A is diagonalizable $\Leftrightarrow$ A is nondefective $\Leftrightarrow$ A has n linearly independent eigenvectors.
13. Every A has Schur decomposition. Moreover, $T_{ii} = \lambda_i$.
14. If A is normal, matrix T from its Schur decomposition is normal. Consequently:
    - T is diagonal, and has eigenvalues of A on diagonal,
    - matrix Q of the Schur decomposition is the unitary matrix of eigenvectors,
    - all eigenvalues of A are semisimple and A is nondefective.
15. If A and B are similar, $\sigma(A) = \sigma(B)$. Consequently, $\mathrm{tr}(A) = \mathrm{tr}(B)$ and $\det(A) = \det(B)$.
16. Eigenvalues and eigenvectors are continous and differentiable: if λ is a simple eigenvalue of A and $A(\varepsilon) = A + \varepsilon E$ for some $E \in F^{n\times n}$, for small ε there exist differentiable functions $\lambda(\varepsilon)$ and $x(\varepsilon)$ such that
    $$A(\varepsilon)x(\varepsilon) = \lambda(\varepsilon)x(\varepsilon).$$
17. Classical motivation for the eigenvalue problem is the following: consider the system of linear differential equations with constant coefficients,
    $$\dot{y}(t) = Ay(t).$$
    If the solution is $y = e^{\lambda t}x$ for some constant vector x, then $\lambda e^{\lambda t}x = Ae^{\lambda t}x$, or $Ax = \lambda x$.

1.4.3 Examples

We shall illustrate above Definitions and Facts on several small examples, using symbolic computation.

```
In [1]: using SympPy
```

```
In [2]: A=[-3 7 -1; 6 8 -2; 72 -28 19]
```

```
Out[2]: 3×3 Array{Int64,2}:
         -3    7  -1
          6    8  -2
         72  -28  19
```

```
In [3]: @vars x
```

```
Out[3]: (x,)
```

```
In [4]: A-x*I
```

Out[4]:

$$\begin{bmatrix} -x-3 & 7 & -1 \\ 6 & -x+8 & -2 \\ 72 & -28 & -x+19 \end{bmatrix}$$

```
In [5]: # Characteristic polynomial p_A(λ)
        p(x)=det(A-x*I)
        p(x)
```

Out[5]:

$$(-x-3)\left(-x+8-\frac{42}{-x-3}\right)\left(-x-\frac{\left(-28-\frac{504}{-x-3}\right)\left(-2+\frac{6}{-x-3}\right)}{-x+8-\frac{42}{-x-3}}+19+\frac{72}{-x-3}\right)$$

```
In [6]: # Characteristic polynomial in nicer form
        p(x)=factor(det(A-x*I))
        p(x)
```

Out[6]:

$$-(x-15)^2(x+6)$$

```
In [7]: λ=solve(p(x),x)
```

Out[7]:

$$\begin{bmatrix} -6 \\ 15 \end{bmatrix}$$

The eigenvalues are $\lambda_1 = -6$ and $\lambda_2 = 15$ with algebraic multiplicities $\alpha(\lambda_1) = 1$ and $\alpha(\lambda_2) = 2$.

```
In [8]: g=nullspace(A-λ[1]*I)
```

```
Out[8]: 1-element Array{Array{SymPy.Sym,1},1}:
         SymPy.Sym[-1/4, 1/4, 1]
```

```
In [9]: h=nullspace(A-λ[2]*I)
```

```
Out[9]: 1-element Array{Array{SymPy.Sym,1},1}:
         SymPy.Sym[-1/4, -1/2, 1]
```

The geometric multiplicities are $\gamma(\lambda_1) = 1$ and $\gamma(\lambda_2) = 1$. Thus, λ_2 is not semisimple, therefore A is defective and not diagonalizable.

```
In [10]: # Trace and determinant
         trace(A), λ[1]+λ[2]+λ[2]
```

```
Out[10]: (24, 24)
```

```
In [11]: det(A), λ[1]*λ[2]*λ[2]
```

```
Out[11]: (-1350.0000000000002, -1350)
```

```
In [12]: # Schur decomposition
         T,Q=schur(A)
         T
```

```
Out[12]: 3×3 Array{Float64,2}:
          -6.0  25.4662       -72.2009
           0.0  15.0          -12.0208
           0.0   1.48587e-15   15.0
```

```
In [13]: Q
```

```
Out[13]: 3×3 Array{Float64,2}:
          -0.235702  -0.0571662  -0.970143
           0.235702  -0.971825   -5.90663e-16
           0.942809   0.228665   -0.242536
```

```
In [14]: @which schur(A)
```

```
Out[14]: schur(A::Union{Base.ReshapedArray{T,2,A,MI} where MI<:Tuple{Vararg{Base.MultiplicativeI
```

```
In [15]: F=schurfact(A)
```

```
Out[15]: Base.LinAlg.Schur{Float64,Array{Float64,2}} with factors T and Z:
         [-6.0 25.4662 -72.2009; 0.0 15.0 -12.0208; 0.0 1.48587e-15 15.0]
         [-0.235702 -0.0571662 -0.970143; 0.235702 -0.971825 -5.90663e-16; 0.942809 0.228665 -0.
         and values:
         Complex{Float64}[-6.0+0.0im, 15.0+1.33647e-7im, 15.0-1.33647e-7im]
```

```
In [16]: fieldnames(F)
```

```
Out[16]: 3-element Array{Symbol,1}:
          :T
          :Z
          :values
```

```
In [17]: F.Z
```

```
Out[17]: 3×3 Array{Float64,2}:
          -0.235702  -0.0571662  -0.970143
           0.235702  -0.971825   -5.90663e-16
           0.942809   0.228665   -0.242536
```

```
In [18]: F[:Z]
```

```
Out[18]: 3×3 Array{Float64,2}:
          -0.235702  -0.0571662  -0.970143
           0.235702  -0.971825   -5.90663e-16
           0.942809   0.228665   -0.242536
```

```
In [19]: println(diag(T))
```

```
[-6.0, 15.0, 15.0]
```

```
In [20]: Q'*Q
```

```
Out[20]: 3×3 Array{Float64,2}:
          1.0          1.11022e-16  2.77556e-17
          1.11022e-16  1.0          1.52656e-16
          2.77556e-17  1.52656e-16  1.0
```

```
In [21]: Q*Q'
```

```
Out[21]: 3×3 Array{Float64,2}:
          1.0           1.35877e-16  0.0
          1.35877e-16  1.0           3.22346e-17
          0.0           3.22346e-17  1.0
```

```
In [22]: # Similar matrices
         M=rand(-5:5,3,3)
         B=M*A*inv(M)
         eigvals(B), trace(B), det(B)
```

```
Out[22]: ([-6.0, 15.0, 15.0], 23.999999999999986, -1350.000000000001)
```

1.4.4 Example

This matrix is nondefective and diagonalizable.

```
In [23]: A=[57 -21 21; -14 22 -7; -140 70 -55]
```

```
Out[23]: 3×3 Array{Int64,2}:
            57  -21   21
           -14   22   -7
          -140   70  -55
```

```
In [24]: p(x)=factor(det(A-x*I))
         p(x)
```

```
Out[24]:
```

$$-\left(x-15\right)^{2}\left(x+6\right)$$

```
In [25]: λ=solve(p(x),x)
```

```
Out[25]:
```

$$\begin{bmatrix}-6\\15\end{bmatrix}$$

```
In [26]: h=nullspace(A-λ[2]*I)
```

```
Out[26]: 2-element Array{Array{SymPy.Sym,1},1}:
          SymPy.Sym[1/2, 1, 0]
          SymPy.Sym[-1/2, 0, 1]
```

1.4.5 Example

Let us try some random examples of dimension $n = 4$ (the largest n for which we can compute eigevalues symbolically).

```
In [27]: A=rand(-4:4,4,4)
```

```
Out[27]: 4×4 Array{Int64,2}:
           1   0   4  -4
           0  -3   4   0
          -3  -1   1   1
           1   3  -1  -2
```

```
In [28]: p(x)=factor(det(A-x*I))
         p(x)
```

```
Out[28]:
```

$$x^4 + 3x^3 + 18x^2 + 63x - 37$$

```
In [29]: λ=solve(p(x),x)
```

```
Out[29]:
```

$$\left[\begin{array}{l}
-\frac{3}{4} + \frac{\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}}{2} - \frac{\sqrt{-\frac{315}{4\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}} - \frac{39}{2} - 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}} + \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{3032}}{18}}}}}{2} \\
-\frac{3}{4} + \frac{\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}}{2} + \frac{\sqrt{-\frac{315}{4\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}} - \frac{39}{2} - 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}} + \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{3032}}{18}}}}}{2} \\
-\frac{\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}}{2} - \frac{3}{4} + \frac{\sqrt{-\frac{39}{2} - 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}} + \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + \frac{315}{4\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{3032}}{18}}}}}}{2} \\
-\frac{\sqrt{-\frac{39}{2} - 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}} + \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + \frac{315}{4\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}}}}{2} - \frac{\sqrt{-\frac{39}{4} - \frac{229}{6\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}} + 2\sqrt[3]{\frac{2355}{8}+\frac{\sqrt{30328437}}{18}}}}{2} -
\end{array}\right.$$

```
In [30]: length(λ)
```

```
Out[30]: 4
```

Since all eigenvalues are distinct, they are all simple and the matrix is diagonalizable. With high probability, all eigenvalues of a random matrix are simple.

Do not try to use `nullspace()` here.

In [31]:
```julia
A=rand(4,4)
p(x)=factor(det(A-x*I))
p(x)
```

Out[31]:

$$\frac{6.89461187104567\left(0.14504079688656x^8 - 0.668803822573383x^7 + 1.0x^6 - 0.55124280370586x^5 + 0.1162848121708\ldots\right.}{(1.0x - 0.874820667009007)^2\left(1.0x^2 - 0.93928\ldots\right.}$$

In this case, symbolic computation does not work well with floating-point numbers - the degree of $p_A(x)$ is 8 instead of 4.

Let us try `Rational` numbers:

In [32]:
```julia
A=map(Rational,A)
```

Out[32]:
```
4×4 Array{Rational{Int64},2}:
 1969921014978887//2251799813685248  …  3903413676503757//4503599627370496
 1043807032273379//2251799813685248     3590728183568295//4503599627370496
 2112048884840731//2251799813685248      944606236044501//2251799813685248
  158954728713921//2251799813685248     3144110323860347//4503599627370496
```

In [33]:
```julia
p(x)=factor(det(A-x*I))
p(x)
```

Out[33]:

$$\frac{10284403483257537763468557390983440656142099160209874145928806 4x^4 - 197688141826136999758874598644 6\ldots}{\ldots}$$

In [34]:
```julia
λ=solve(p(x),x)
```

Out[34]:

$$\left[\begin{array}{l}
\frac{8656877798631013}{18014398509481984} + \sqrt{-\frac{\ldots}{30853210449772613290405672172950321968426297480629622437786419 2\sqrt[3]{\frac{14201270538028849311278267879556213385914161911707036492449329441568651193 33\ldots}{16687398718132110018711107079449625895333629080911349765211262561111091607661\ldots}}}}\ldots \\
-\sqrt{-\frac{64117210520248774846133 5\ldots}{30853210449772613290405672172950321968426297480629622437786419 2\sqrt[3]{\frac{142012705380288493112782678795562133859141619117070364924493294415686511933316950321338596307}{166873987181321100187111070794496258953336290809113497652112625611110916076612542970543913041 92}+\frac{185\sqrt{7\ldots}}{\ldots}}}}\ldots \\
-\sqrt{-\frac{64117210520248774846133 5\ldots}{30853210449772613290405672172950321968426297480629622437786419 2\sqrt[3]{\frac{142012705380288493112782678795562133859141619117070364924493294415686511933316950321338596307}{166873987181321100187111070794496258953336290809113497652112625611110916076612542970543913041 92}+\frac{185\sqrt{7\ldots}}{\ldots}}}}\ldots \\
-\sqrt{-2\sqrt[3]{\frac{142012705380288493112782678795562133859141619117070364924493294415686511933316950321338596307}{166873987181321100187111070794496258953336290809113497652112625611110916076612542970543913041 92}+\frac{185\sqrt{78708315040110128366278506745878773940703844336657476148261604\ldots}}{15018658846318899016 8\ldots}}\ldots}\ldots
\end{array}\right.$$

```
In [35]: length(λ)
```

```
Out[35]: 4
```

1.4.6 Example - Circulant matrix

For more details, see Section [Bot14] and the references therein.

References

[Bot14] A. Böttcher and I. Spitkovsky, Special Types of Matrices, in: L. Hogben, ed., 'Handbook of Linear Algebra', pp. 22.1-22.20, CRC Press, Boca Raton, 2014.

Given $a_0, a_1, \ldots, a_{n-1} \in \mathbb{C}$, the **circulant matrix** is

$$C(a_0, a_1, \ldots, a_{n-1}) = \begin{bmatrix} a_0 & a_{n-1} & a_{n-2} & \cdots & a_1 \\ a_1 & a_0 & a_{n-1} & \cdots & a_2 \\ a_2 & a_1 & a_0 & \cdots & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_0 \end{bmatrix}.$$

Let $a(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_{n-1} z^{n-1}$ be the associated complex polynomial.

Let $\omega_n = \exp\left(\frac{2\pi i}{n}\right)$. The **Fourier matrix** of order n is

$$F_n = \frac{1}{\sqrt{n}} \left[\omega_n^{(j-1)(k-1)}\right]_{j,k=1}^{n} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots \omega_n^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots \omega_n^{(n-1)(n-1)} \end{bmatrix}.$$

Fourier matrix is unitary. Fourier matrix is a Vandermonde matrix, $F_n = \frac{1}{\sqrt{n}} V(1, \omega_n, \omega_n^2, \ldots, \omega_n^{n-1})$.

Circulant matrix is normal and, thus, unitarily diagonalizable, with the eigenvalue decomposition

$$C(a_0, a_1, \ldots, a_{n-1}) = F_n^* \operatorname{diag}[(a(1), a(\omega_n), a(\omega_n^2), \ldots, a(\omega_n^{n-1})] F_n.$$

We shall use the package SpecialMatrices.jl.

```
In [36]: using SpecialMatrices
         using Polynomials
```

```
In [37]: whos(SpecialMatrices)
```

```
           Cauchy     40 bytes  UnionAll
        Circulant     40 bytes  UnionAll
        Companion     40 bytes  UnionAll
        Frobenius     40 bytes  UnionAll
           Hankel     40 bytes  UnionAll
          Hilbert     40 bytes  UnionAll
            Kahan     80 bytes  UnionAll
          Riemann     40 bytes  UnionAll
  SpecialMatrices   6671 bytes  Module
           Strang     40 bytes  UnionAll
         Toeplitz     40 bytes  UnionAll
      Vandermonde     40 bytes  UnionAll
            embed      0 bytes  SpecialMatrices.#embed
```

In [38]:
```
n=6
a=rand(-9:9,n)
```

Out[38]:
```
6-element Array{Int64,1}:
 -2
  0
  6
 -3
  3
  5
```

In [39]:
```
C=Circulant(a)
```

Out[39]:
```
6×6 SpecialMatrices.Circulant{Int64}:
 -2   5   3  -3   6   0
  0  -2   5   3  -3   6
  6   0  -2   5   3  -3
 -3   6   0  -2   5   3
  3  -3   6   0  -2   5
  5   3  -3   6   0  -2
```

In [40]:
```
# Check for normality
full(C)*full(C)'-full(C)'*full(C)
```

Out[40]:
```
6×6 Array{Int64,2}:
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
```

```
In [41]: p1=Polynomials.Poly(a)

Out[41]: Poly(-2 + 6*x^2 - 3*x^3 + 3*x^4 + 5*x^5)

In [42]: ω=exp(2*π*im/n)

Out[42]: 0.5000000000000001 + 0.8660254037844386im

In [43]: v=map(Complex,[ω^k for k=0:n-1])
         F=Vandermonde(v)

Out[43]: 6×6 SpecialMatrices.Vandermonde{Complex{Float64}}:
          1.0+0.0im   1.0+0.0im           1.0+0.0im           …   1.0+0.0im
          1.0+0.0im   0.5+0.866025im     -0.5+0.866025im          0.5-0.866025im
          1.0+0.0im  -0.5+0.866025im     -0.5-0.866025im         -0.5-0.866025im
          1.0+0.0im  -1.0+3.88578e-16im   1.0-7.77156e-16im      -1.0+1.94289e-15im
          1.0+0.0im  -0.5-0.866025im     -0.5+0.866025im         -0.5+0.866025im
          1.0+0.0im   0.5-0.866025im     -0.5-0.866025im      …   0.5+0.866025im

In [44]: Fn=full(F)/sqrt(n)
         Λ=Fn*full(C)*Fn'

Out[44]: 6×6 Array{Complex{Float64},2}:
                   9.0+0.0im           …   5.55112e-15-5.10703e-15im
          -1.22125e-15-1.02904e-18im      -9.99201e-16+2.22045e-16im
          -1.02827e-15+1.33227e-15im      -1.51334e-15+5.05199e-15im
           2.22045e-16+1.7486e-15im       -1.94749e-15-1.33227e-15im
           8.19498e-16+3.55271e-15im        8.6254e-15-1.71901e-15im
           5.10703e-15+4.95572e-15im   …          -1.0+1.73205im

In [45]: [diag(Λ) p1(v) eigvals(full(C))]

Out[45]: 6×3 Array{Complex{Float64},2}:
            9.0+0.0im         9.0+0.0im            -12.0+6.9282im
           -1.0-1.73205im    -1.0-1.73205im        -12.0-6.9282im
          -12.0-6.9282im    -12.0-6.9282im           9.0+0.0im
            5.0+0.0im         5.0-3.10862e-15im      5.0+0.0im
          -12.0+6.9282im    -12.0+6.9282im          -1.0+1.73205im
           -1.0+1.73205im    -1.0+1.73205im         -1.0-1.73205im
```

1.5 Hermitian and real symmetric matrices

For more details and the proofs of the Facts below, see Section [Bar14] and the references therein.

References

[Bar14] W. Barrett, Hermitian and Positive Definite Matrices, in: L. Hogben, ed., 'Handbook of Linear Algebra', pp. 9.1-9.13, CRC Press, Boca Raton, 2014.

1.5.1 Definitions

Matrix $A \in \mathbb{C}^{n\times n}$ is **Hermitian** or **self-adjoint** if $A^* = A$, or element-wise, $\bar{a}_{ij} = a_{ji}$. We say $A \in \mathcal{H}_n$.

Matrix $A \in \mathbb{R}^{n\times n}$ is **symmetric** if $A^T = A$, or element-wise, $a_{ij} = a_{ji}$. We say $A \in \mathcal{S}_n$.

Rayleigh qoutient of $A \in \mathcal{H}_n$ and nonzero vector $x \in \mathbb{C}^n$ is

$$R_A(x) = \frac{x^* A x}{x^* x}.$$

Matrices $A, B \in \mathcal{H}_n$ are **congruent** if there exists nonsingular matrix C such that $B = C^* A C$.

Inertia of $A \in \mathcal{H}_n$ is the ordered triple

$$\operatorname{in}(A) = (\pi(A), \nu(A), \zeta(A)),$$

where $\pi(A)$ is the number of positive eigenvalues of A, $\nu(A)$ is the number of negative eigenvalues of A, and $\zeta(A)$ is the number of zero eigenvalues of A.

Gram matrix of a set of vectors $x_1, x_2, \ldots, x_k \in \mathbb{C}^n$ is the matrix G with entries $G_{ij} = x_i^* x_j$.

1.5.2 Facts

Assume A is Hermitian and $x \in \mathbb{C}^n$ is nonzero. Then

1. Real symmetric matrix is Hermitian, and real Hermitian matrix is symmetric.
2. Hermitian and real symmetric matrices are normal.
3. $A + A^*$, $A^* A$, and AA^* are Hermitian.
4. If A is nonsingular, A^{-1} is Hermitian.
5. Main diagonal entries of A are real.
6. Matrix T from the Schur decomposition of A is Hermitian. Consequently:
   - T is diagonal and real, and has eigenvalues of A on diagonal,
   - matrix Q of the Schur decomposition is the unitary matrix of eigenvectors,
   - all eigenvalues of A are semisimple and A is nondefective,
   - eigenvectors corresponding to distinct eigenvalues are orthogonal.
7. **Spectral Theorem.** To summarize:

- if $A \in \mathcal{H}_n$, there is a unitary matrix U and real diagonal matrix Λ such that $A = U\Lambda U^*$. The diagonal entries of Λ are the eigenvalues of A, and the columns of U are the corresponding eigenvectors.
- if $A \in \mathcal{S}_n$, the same holds with orthogonal matrix U, $A = U\Lambda U^T$.
- if $A \in \mathcal{H}_n$ with eigenpairs (λ_i, u_i), then

$$A = \lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^* + \cdots + \lambda_n u_n u_n^*.$$

- similarly, if $A \in \mathcal{S}_n$, then

$$A = \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T + \cdots + \lambda_n u_n u_n^T.$$

8. Since all eigenvalues of A are real, they can be ordered:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n.$$

9. **Rayleigh-Ritz Theorem.** It holds:

$$\begin{aligned}
\lambda_n &\leq \frac{x^*Ax}{x^*x} \leq \lambda_1, \\
\lambda_1 &= \max_x \frac{x^*Ax}{x^*x} = \max_{\|x\|_2=1} x^*Ax, \\
\lambda_n &= \min_x \frac{x^*Ax}{x^*x} = \min_{\|x\|_2=1} x^*Ax.
\end{aligned}$$

10. **Courant-Fischer Theorem.** It holds:

$$\begin{aligned}
\lambda_k &= \max_{\dim(W)=k} \min_{x\in W} \frac{x^*Ax}{x^*x} \\
&= \min_{\dim(W)=n-k+1} \max_{x\in W} \frac{x^*Ax}{x^*x}.
\end{aligned}$$

11. **Cauchy Interlace Theorem.** Let B be the principal submatrix of A obtained by deleting the i-th row and the i-th column of A. Let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_{n-1}$ be the eignvalues of B. Then

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \lambda_3 \geq \cdots \geq \lambda_{n-1} \geq \mu_{n-1} \geq \lambda_n.$$

12. **Weyl Inequalities.** For $A, B \in \mathcal{H}_n$, it holds:

$$\begin{aligned}
\lambda_{j+k-1}(A+B) &\leq \lambda_j(A) + \lambda_k(B), && \text{for } j+k \leq n+1, \\
\lambda_{j+k-n}(A+B) &\geq \lambda_j(A) + \lambda_k(B), && \text{for } j+k \geq n+1,
\end{aligned}$$

and, in particular,

$$\lambda_j(A) + \lambda_n(B) \leq \lambda_j(A+B) \leq \lambda_j(A) + \lambda_1(B), \quad \text{for } j = 1, 2, \ldots, n.$$

13. $\pi(A) + \mu(A) + \zeta(A) = n$.
14. $\text{rank}(A) = \pi(A) + \mu(A)$.
15. If A is nonsingular, $\text{in}(A) = \text{in}(A^{-1})$.
16. If $A, B \in \mathcal{H}_n$ are similar, $\text{in}(A) = \text{in}(B)$.
17. **Sylvester's Law of Inertia.** $A, B \in \mathcal{H}_n$ are congruent if and only if $\text{in}(A) = \text{in}(B)$.
18. **Subadditivity of Inertia.** For $A, B \in \mathcal{H}_n$,

$$\pi(A+B) \leq \pi(A) + \pi(B), \qquad \nu(A+B) \leq \nu(A) + \nu(B).$$

19. Gram matrix is Hermitian.

1.5.3 Example - Hermitian matrix

```
In [46]: # Generating Hermitian matrix
         n=4
         A=rand(n,n)+im*rand(n,n)
         A=A+A'
```

```
Out[46]: 4×4 Array{Complex{Float64},2}:
          0.175006+0.0im        1.11516+0.657242im  …    1.7725-0.36993im
           1.11516-0.657242im   0.61824+0.0im           1.41848+0.270531im
           1.11377+0.19234im   0.645817-0.529582im     0.802366-0.301999im
            1.7725+0.36993im    1.41848-0.270531im     0.332017+0.0im
```

```
In [47]: ishermitian(A)
```

```
Out[47]: true
```

```
In [48]: # Diagonal entries
         diag(A)
```

```
Out[48]: 4-element Array{Complex{Float64},1}:
          0.175006+0.0im
           0.61824+0.0im
          0.735742+0.0im
          0.332017+0.0im
```

```
In [49]: # Schur decomposition
         T,Q=schur(A)
```

```
Out[49]: (Complex{Float64}[3.97417-1.1106e-16im 7.59684e-17+3.1225e-16im -4.48192e-16-2.82466e-1
```

```
In [50]: λ,U=eig(A)
```

```
Out[50]: ([-2.06705, -0.520821, 0.474706, 3.97417], Complex{Float64}[-0.564883+0.357615im -0.060
```

```
In [51]: # Spectral theorem
         norm(A-U*diagm(λ)*U')
```

```
Out[51]: 9.359483730626927e-15
```

```
In [52]: # Spectral theorem
         A-sum([λ[i]*U[:,i]*U[:,i]' for i=1:n])
```

```
Out[52]: 4×4 Array{Complex{Float64},2}:
          -4.55191e-15+6.93889e-18im  ...   4.44089e-16+0.0im
            1.9984e-15-2.88658e-15im       4.44089e-16+1.11022e-16im
           2.88658e-15+1.58207e-15im       3.33067e-16-1.11022e-16im
           4.44089e-16+5.55112e-17im      -1.11022e-15+0.0im
```

```
In [53]: λ
```

```
Out[53]: 4-element Array{Float64,1}:
          -2.06705
          -0.520821
           0.474706
           3.97417
```

```
In [54]: # Cauchy Interlace Theorem (repeat several times)
         @show i=rand(1:n)
         μ=eigvals(A[[1:i-1;i+1:n],[1:i-1;i+1:n]])
```

```
i = rand(1:n) = 3
```

```
Out[54]: 3-element Array{Float64,1}:
          -1.96694
          -0.179044
           3.27125
```

```
In [55]: # Inertia
         inertia(A)=[sum(eigvals(A).>0), sum(eigvals(A).<0), sum(eigvals(A).==0)]
```

```
Out[55]: inertia (generic function with 1 method)
```

```
In [56]: inertia(A)
```

```
Out[56]: 3-element Array{Int64,1}:
          2
          2
          0
```

```
In [57]: # Similar matrices
         C=rand(n,n)+im*rand(n,n)
         B=C*A*inv(C)
         eigvals(B)

Out[57]: 4-element Array{Complex{Float64},1}:
            3.97417-3.61283e-15im
           -2.06705+1.57799e-15im
           0.474706+3.21583e-15im
          -0.520821-7.84798e-16im
```

This did not work numerically due to rounding errors!

```
In [58]: # Congruent matrices - this does not work either, without some preparation
         B=C'*A*C
         inertia(A)==inertia(B)


        MethodError: no method matching isless(::Int64, ::Complex{Float64})
    Closest candidates are:
      isless(::PyCall.PyObject, ::Any) at /home/slap/.julia/v0.6/PyCall/src/pyoperators.jl:71
      isless(::SymPy.Sym, ::Number) at /home/slap/.julia/v0.6/SymPy/src/logical.jl:106
      isless(::Real, ::AbstractFloat) at operators.jl:97
      ...



        Stacktrace:

         [1] (::##5#8)(::Complex{Float64}) at ./<missing>:0

         [2] broadcast_t(::Function, ::Type{Any}, ::Tuple{Base.OneTo{Int64}}, ::CartesianRange{C

         [3] broadcast_c at ./broadcast.jl:321 [inlined]

         [4] broadcast(::Function, ::Array{Complex{Float64},1}) at ./broadcast.jl:455

         [5] inertia(::Array{Complex{Float64},2}) at ./In[55]:2

         [6] include_string(::String, ::String) at ./loading.jl:522


In [59]: # We need to symmetrize B
         inertia((B+B')/2)

Out[59]: 3-element Array{Int64,1}:
          2
```

```
 2
 0
```

```
In [60]: # Weyl's Inequalities
         B=rand(n,n)+im*rand(n,n)
         B=(B+B')/10
         @show λ
         μ=eigvals(B)
         γ=eigvals(A+B)
         μ,γ
```

```
λ = [-2.06705, -0.520821, 0.474706, 3.97417]
```

```
Out[60]: ([-0.0908422, 0.043951, 0.124584, 0.506866], [-2.04727, -0.506073, 0.541985, 4.45692])
```

```
In [61]: # Theorem uses different order!
         j=rand(1:n)
         k=rand(1:n)
         @show j,k
         if j+k<=n+1
             @show sort(γ,rev=true)[j+k-1], sort(λ,rev=true)[j]+sort(μ,rev=true)[k]
         end
         if j+k>=n+1
             sort(γ,rev=true)[j+k-n], sort(λ,rev=true)[j]+sort(μ,rev=true)[k]
         end
```

```
(j, k) = (3, 4)
```

```
Out[61]: (-0.5060729475044469, -0.6116633579741112)
```

```
In [62]: sort(λ,rev=true)
```

```
Out[62]: 4-element Array{Float64,1}:
           3.97417
           0.474706
          -0.520821
          -2.06705
```

1.5.4 Example - Real symmetric matrix

```
In [63]: # Generating real symmetric matrix
         n=6
         A=rand(-9:9,n,n)
         A=A+A'
```

```
Out[63]: 6×6 Array{Int64,2}:
           -6  -3  -1  -11    3   7
           -3  16   9    5   -9  10
           -1   9  10   -5    3  -1
          -11   5  -5   16  -14   4
            3  -9   3  -14    8   6
            7  10  -1    4    6  -6
```

```
In [64]: issymmetric(A)
```

```
Out[64]: true
```

```
In [65]: T,Q=schur(A)
```

```
Out[65]: ([-20.7392 -3.55271e-14 ... 7.0198e-15 -1.03824e-15; 0.0 35.0551 ... -5.3653e-15 -3.368
```

```
In [66]: T
```

```
Out[66]: 6×6 Array{Float64,2}:
          -20.7392  -3.55271e-14   6.01947e-15  ...   7.0198e-15   -1.03824e-15
            0.0     35.0551       -6.23788e-15      -5.3653e-15   -3.36847e-15
            0.0      0.0          22.6115           -1.05284e-15   3.60359e-15
            0.0      0.0           0.0               7.81625e-16  -1.45886e-15
            0.0      0.0           0.0               6.53279       4.26412e-16
            0.0      0.0           0.0          ...   0.0           1.63791
```

```
In [67]: Q
```

```
Out[67]: 6×6 Array{Float64,2}:
          -0.52912    -0.238701    0.107513   0.683696   0.339459   -0.26234
          -0.277743    0.483725    0.657415  -0.36792    0.281135   -0.2056
           0.0674099  -0.0132337   0.611388   0.377824  -0.617287    0.312557
          -0.307918    0.661594   -0.33301    0.233929   0.0428889   0.547743
          -0.326755   -0.516467    0.152099  -0.332148   0.254162    0.654552
           0.660882    0.0664266   0.220048   0.298995   0.59854     0.250507
```

```
In [68]: λ,U=eig(A)
         λ
```

```
Out[68]: 6-element Array{Float64,1}:
          -20.7392
           -7.09809
            1.63791
            6.53279
           22.6115
           35.0551
```

```
In [69]: U
```

```
Out[69]: 6×6 Array{Float64,2}:
          -0.52912    -0.683696   0.26234    0.339459    0.107513   0.238701
          -0.277743    0.36792    0.2056     0.281135    0.657415  -0.483725
           0.0674099  -0.377824  -0.312557  -0.617287    0.611388   0.0132337
          -0.307918   -0.233929  -0.547743   0.0428889  -0.33301   -0.661594
          -0.326755    0.332148  -0.654552   0.254162    0.152099   0.516467
           0.660882   -0.298995  -0.250507   0.59854     0.220048  -0.0664266
```

```
In [70]: A-sum([λ[i]*U[:,i]*U[:,i]' for i=1:n])
```

```
Out[70]: 6×6 Array{Float64,2}:
          -3.01981e-14   1.86517e-14  -3.63043e-14  …   1.95399e-14  -7.10543e-15
           1.86517e-14  -2.84217e-14   1.59872e-14      0.0          -1.24345e-14
          -3.63043e-14   1.59872e-14  -5.68434e-14      5.32907e-15  -4.55191e-15
          -1.24345e-14   2.30926e-14  -1.5099e-14       3.55271e-15  -3.55271e-15
           1.95399e-14   0.0           5.32907e-15     -1.59872e-14  -9.76996e-15
          -7.10543e-15  -1.24345e-14  -4.996e-15    …  -8.88178e-15   1.23457e-13
```

```
In [71]: inertia(A)
```

```
Out[71]: 3-element Array{Int64,1}:
          4
          2
          0
```

```
In [72]: C=rand(n,n)
         inertia(C'*A*C)
```

```
Out[72]: 3-element Array{Int64,1}:
          4
          2
          0
```

1.6 Positive definite matrices

These matrices are an important subset of Hermitian or real symmteric matrices.

1.6.1 Definitions

Matrix $A \in \mathcal{H}_n$ is **positive definite** (PD) if $x^* Ax > 0$ for all nonzero $x \in \mathbb{C}^n$.

Matrix $A \in \mathcal{H}_n$ is **positive semidefinite** (PSD) if $x^* Ax \geq 0$ for all nonzero $x \in \mathbb{C}^n$.

1.6.2 Facts

1. $A \in \mathcal{S}_n$ is PD if $x^T A x > 0$ for all nonzero $x \in \mathbb{R}^n$, and is PSD if $x^T A x \geq 0$ for all $x \in \mathbb{R}^n$.
2. If $A, B \in \text{PSD}_n$, then $A + B \in \text{PSD}_n$. If, in addition, $A \in \text{PD}_n$, then $A + B \in \text{PD}_n$.
3. If $A \in \text{PD}_n$, then $\text{tr}(A) > 0$ and $\det(A) > 0$.
4. If $A \in \text{PSD}_n$, then $\text{tr}(A) \geq 0$ and $\det(A) \geq 0$.
5. Any principal submatrix of a PD matrix is PD. Any principal submatrix of a PSD matrix is PSD. Consequently, all minors are positive or nonnegative, respectively.
6. $A \in \mathcal{H}_n$ is PD iff *every leading* principal minor of A is positive. $A \in \mathcal{H}_n$ is PSD iff *every* principal minor is nonnegative.
7. For $A \in \text{PSD}_n$, there exists unique PSD k-th **root**, $A^{1/k} = U\Lambda^{1/k}U^*$.
8. *(Cholesky Factorization)* $A \in \mathcal{H}_n$ if PD iff there is an invertible lower triangular matrix L with positive diagonal entries such that $A = LL^*$.
9. Gram matrix is PSD. If the vectors are linearly independent, Gram matrix is PD.

1.6.3 Example - Positive definite matrix

In [73]:
```
# Generating positive definite matrix as a Gram matrix
n=5
A=rand(n,n)+im*rand(n,n)
A=A*A'
```

Out[73]:
```
5×5 Array{Complex{Float64},2}:
 1.83633+0.0im         2.0928+0.494729im  …  2.42437+0.149122im
  2.0928-0.494729im   3.24377+0.0im          3.56763-0.666931im
 1.86835-0.68901im    2.84287+0.134274im     4.76557-0.447265im
 1.67368-0.0564416im  2.13082+0.481664im     3.53916-0.0324904im
 2.42437-0.149122im   3.56763+0.666931im     5.73632+0.0im
```

In [74]:
```
ishermitian(A)
```

Out[74]:
```
true
```

In [75]:
```
eigvals(A)
```

Out[75]:
```
5-element Array{Float64,1}:
  0.0881005
  0.159667
  0.665919
  1.44659
 15.8314
```

In [76]:
```
# Positivity of principal leading minors
[det(A[1:i,1:i]) for i=1:n]
```

Out[76]:
```
5-element Array{Complex{Float64},1}:
   1.83633+0.0im
   1.33206+1.11022e-16im
   2.45674+5.00537e-16im
  0.990398+6.02389e-16im
  0.214526+1.95767e-16im
```

In [77]:
```
# Square root
λ,U=eig(A)
Ar=U*diagm(sqrt.(λ))*U'
A-Ar*Ar
```

Out[77]:
```
5×5 Array{Complex{Float64},2}:
 8.88178e-16+1.11022e-16im  …  1.77636e-15+4.44089e-16im
 8.88178e-16-5.55112e-17im     2.22045e-15-4.44089e-16im
 1.33227e-15-5.55112e-16im     2.66454e-15-9.99201e-16im
 8.88178e-16-7.07767e-16im     4.44089e-16+2.77556e-16im
 1.33227e-15-3.60822e-16im     3.55271e-15-2.77556e-17im
```

In [78]:
```
# Cholesky factorization - the upper triangular factor is returned
L=chol(A)
```

Out[78]:
```
5×5 UpperTriangular{Complex{Float64},Array{Complex{Float64},2}}:
 1.35511+0.0im  1.54437+0.365083im  …   1.78906+0.110044im
         ⋅       0.8517+0.0im          0.897592-0.215716im
         ⋅              ⋅               1.16429+0.00427023im
         ⋅              ⋅              0.120929-0.290623im
         ⋅              ⋅              0.465409+0.0im
```

In [79]:
```
norm(A-L'*L)
```

Out[79]:
```
1.0031043987764443e-15
```

1.6.4 Example - Positive semidefinite matrix

In [80]:
```
# Generating positive semidefinite matrix as a Gram matrix, try it several times
n=6
m=4
A=rand(-9:9,n,m)
```

Out[80]:
```
6×4 Array{Int64,2}:
  6  5  5   2
  0  6  6  -3
```

```
  3  -9   1   6
 -2   6  -8   3
 -7  -4  -5   3
  6   5  -1  -1
```

In [81]:
```
A=A*A'
```

Out[81]:
```
6×6 Array{Int64,2}:
  90   54  -10  -16  -81   54
  54   81  -66  -21  -63   27
 -10  -66  127  -50   28  -34
 -16  -21  -50  113   39   23
 -81  -63   28   39   99  -60
  54   27  -34   23  -60   63
```

In [82]:
```
# There are rounding errors!
eigvals(A)
```

Out[82]:
```
6-element Array{Float64,1}:
   8.85451e-15
   3.37781e-14
  22.3934
  85.4559
 177.756
 287.394
```

In [83]:
```
@which rank(A)
```

Out[83]:
```
rank(A::AbstractArray{T,2} where T) in Base.LinAlg at linalg/generic.jl:723
```

In [84]:
```
# Cholesky factorization - this can fail
L=chol(A)
```

```
Base.LinAlg.PosDefException(5)



Stacktrace:

 [1] _chol!(::Array{Float64,2}, ::Type{UpperTriangular}) at ./linalg/cholesky.jl:55

 [2] chol!(::Hermitian{Float64,Array{Float64,2}}) at ./linalg/cholesky.jl:124

 [3] chol(::Hermitian{Int64,Array{Int64,2}}) at ./linalg/cholesky.jl:145
```

```
[4] chol(::Array{Int64,2}) at ./linalg/cholesky.jl:186

[5] include_string(::String, ::String) at ./loading.jl:522
```

1.6.5 Example - Covariance and correlation matrices

Covariance and correlation matrices are PSD.

Correlation matrix is diagonally scaled covariance matrix.

```
In [85]: x=rand(10,5)
```

```
Out[85]: 10×5 Array{Float64,2}:
          0.454492  0.559037   0.33877    0.84708    0.10335
          0.484327  0.220427   0.860757   0.501913   0.39751
          0.840928  0.0771441  0.158538   0.658027   0.378485
          0.081144  0.591003   0.125599   0.739654   0.695591
          0.584567  0.766753   0.832055   0.451284   0.787183
          0.235022  0.954204   0.499114   0.026292   0.604332
          0.307502  0.349188   0.137916   0.745453   0.681559
          0.151295  0.379964   0.585945   0.0868066  0.283561
          0.839135  0.720298   0.596365   0.808623   0.122719
          0.930813  0.458035   0.0544393  0.767189   0.596098
```

```
In [86]: A=cov(x)
```

```
Out[86]: 5×5 Array{Float64,2}:
           0.0921992   -0.0166857  -0.00627706   0.0423799  -0.0173615
          -0.0166857    0.07034     0.019833    -0.0209628   0.0125028
          -0.00627706   0.019833    0.0893095   -0.0419064  -0.0112602
           0.0423799   -0.0209628  -0.0419064    0.0874753  -0.0120296
          -0.0173615    0.0125028  -0.0112602   -0.0120296   0.0592643
```

```
In [87]: # Covariance matrix is a Gram matrix
         (x.-mean(x,1))'*(x.-mean(x,1))/9-A
```

```
Out[87]: 5×5 Array{Float64,2}:
           0.0           0.0          -8.67362e-19   0.0           0.0
           0.0           0.0           3.46945e-18  -3.46945e-18   0.0
          -8.67362e-19   3.46945e-18   0.0          -6.93889e-18   0.0
           0.0          -3.46945e-18  -6.93889e-18   1.38778e-17   0.0
           0.0           0.0           0.0           3.46945e-18   0.0
```

```
In [88]: B=cor(x)
```

```
Out[88]: 5×5 Array{Float64,2}:
           1.0         -0.207195   -0.0691742   0.471904   -0.234869
          -0.207195     1.0         0.250229   -0.267243    0.193646
          -0.0691742    0.250229    1.0        -0.474121   -0.154776
           0.471904    -0.267243   -0.474121    1.0        -0.167075
          -0.234869     0.193646   -0.154776   -0.167075    1.0
```

```
In [89]: # Diagonal scaling
         D=1./sqrt.(diag(A))
```

```
Out[89]: 5-element Array{Float64,1}:
          3.29334
          3.7705
          3.34619
          3.38109
          4.10774
```

```
In [90]: diagm(D)*A*diagm(D)
```

```
Out[90]: 5×5 Array{Float64,2}:
           1.0         -0.207195   -0.0691742   0.471904   -0.234869
          -0.207195     1.0         0.250229   -0.267243    0.193646
          -0.0691742    0.250229    1.0        -0.474121   -0.154776
           0.471904    -0.267243   -0.474121    1.0        -0.167075
          -0.234869     0.193646   -0.154776   -0.167075    1.0
```

```
In [91]: eigvals(A)
```

```
Out[91]: 5-element Array{Float64,1}:
          0.0296396
          0.0451806
          0.0626813
          0.09504
          0.166047
```

```
In [92]: eigvals(B)
```

```
Out[92]: 5-element Array{Float64,1}:
          0.344896
          0.633831
          0.845555
          1.22093
          1.95479
```

```
In [93]: C=cov(x')
```

```
Out[93]: 10×10 Array{Float64,2}:
  0.075391    -0.00850587  …  -0.0256449    0.0654914    0.0291172
 -0.00850587   0.0547036       0.0210803    0.00280885  -0.0407367
  0.0255895   -6.65218e-5     -0.0537012    0.0324019    0.0927577
  0.0199654   -0.0438513      -0.0231166   -0.031639     0.0204377
 -0.0340216    0.0052068       0.0279152   -0.0280476   -0.0411634
 -0.043481    -0.0340047   …   0.0435742   -0.0385472   -0.0595662
  0.0151072   -0.0263838      -0.0357439   -0.0233638    0.0462068
 -0.0256449    0.0210803       0.0389907   -0.0158487   -0.0631515
  0.0654914    0.00280885     -0.0158487    0.0853681    0.0260316
  0.0291172   -0.0407367      -0.0631515    0.0260316    0.111928
```

```
In [94]: eigvals(C)
```

```
Out[94]: 10-element Array{Float64,1}:
 -5.028e-17
 -3.95996e-17
 -9.60291e-18
 -2.16957e-19
  6.40788e-18
  2.37509e-17
  0.0825783
  0.113017
  0.209325
  0.387483
```

```
In [95]: inertia(C)
```

```
Out[95]: 3-element Array{Int64,1}:
 6
 4
 0
```

```
In [96]: rank(C)
```

```
Out[96]: 4
```

Explain the function `rank()`.

```
In [97]: @which rank(C)
```

```
Out[97]: rank(A::AbstractArray{T,2} where T) in Base.LinAlg at linalg/generic.jl:723
```