IMPORTANT Notes

Indices start at 0
When specifying rows, columns, or entries of a matrix, count by starting with 0.

Row vs Column Vectors
Sage does not differentiate between row and column vectors. Vectors are interpreted as needed so operations are defined, if possible.

Specify the ring
Sage needs to know which space our matrices and vectors are defined in, even when defining a particular matrix. The Sage keys for common sets are
`RR` = reals, `QQ` = rationals, `ZZ` = integers
Specifying the ring is optional but highly recommended.

DEFINING MATRICES AND VECTORS

- `A = Matrix(FF, [[row0],[row1],...])`
  Defines specific matrix with entries in the ring F
  ex: `A = Matrix(RR, [[1,2,3],[4,5,6]])`
  defines $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
- `v = vector(FF, list)`
  Defines specific vector with entries in the ring F
  ex: `v = vector(RR, [1,0,8,6])`

MATRIX Property Testing

Return boolean value (true or false)

- `A.is_invertible()`
- `A.is_symmetric()`
- `A.is_singular()`
- `A.is_orthogonal()`
- `A.is_scalar()`
- `A.is_one()`
- `A.is_zero()`
- `A.is_diagonalizable()`

MATRIX/VECTOR Entries and Components

- `v[i]`
  Entry in position i of vector v
- `A[i,j]`
  Entry in row i and column j of A
- `A.row(m)`
  mth row of A as vector
- `A.column(m)`
  mth column of A as vector
- `A.rows()`
  List of rows of A as list
- `A.columns()`
  List of columns of A as list
- `A.submatrix(m,n,nrows,ncols)`
  $nrows \times ncols$ submatrix of A beginning at entry $A[m, n]$
- `A.matrix_from_rows(list)`
  Forms new matrix from rows of A listed in list
- `A.matrix_from_columns(list)`
  Forms new matrix from columns of A listed in list

SPECIAL Matrices

- `random_matrix(FF,m,n)`
  Random $m \times n$ in field F
  ex: `A = random_matrix(QQ,2,4)`
  A is random 2×4 matrix with entries in $\mathbb{Q}$
- `identity_matrix(n)`
  $n \times n$ identity matrix:
  ex: `I4 = identity_matrix(4)`
  $I4$ is 4×4 identity matrix
- Augment matrices A and B: `A.augment(B)`
  ex: `A = Matrix([[1,2],[3,4]])`
  `N = A.augment(identity_matrix(2))`
  returns

$$N = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{bmatrix}$$

MATRIX Operations

- `A*B`
  Product of matrices A and B, if defined
- `A+B`
  Sum of matrices A and B, if defined
- `A.echelon_form()`
  Echelon form of matrix A
- `A.rref()`
  Reduced echelon form of matrix A
- `A.inverse()`
  Inverse of matrix A (if defined)
- `A.determinant()`
  Returns determinant of matrix A
- `A.transpose()`
  Returns transpose of matrix A
- `A.rank()`
  Returns rank of matrix A

VECTOR Operations

- `v.norm()`
  Norm of vector v
- `v.norm(1)`
  Sum of entries of vector v
- `v.len()`
  Number of entries of v
- `v.dot_product(u)`
  Dot produce of vectors v and u
- `v.cross_product(u)`
  Cross produce of vectors v and u

Row Operations

- `A.swap_rows(n,m)`
  Interchange rows n and m
- `A.rescale_row(n, c)`
  Multiply entries of row n by c
- `A.add_multiple_of_row(n,m,c)`
  Add c times row m to row n

JOINING Matrices

- `A.augment(B)`
  Augment matrices A and B
  ex: `A = Matrix([[1,2],[3,4]])`
  `N = A.augment(identity_matrix(2))`
  returns

$$N = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{bmatrix}$$

- `A.stack(B)`
  Stack matrix A over B
  ex: `A = Matrix([[1,2],[3,4]])`
  `N = A.stack(identity_matrix(2))`
  returns

$$N = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

VECTOR Spaces and Properties

- `VectorSpace(FF, n)`
  n dimensional vector space over F
  ex: `V = VectorSpace(RR,2)`
  defines $V = \mathbb{R}^2$
- `V.dimension()`
  Dimension of space V
- `V.basis()`
  Canonical basis of space V
- `V.is_subspace(W)`
  Test if W is a subspace of V
- `MatrixSpace(FF, m, n)`
  Space of $m \times n$ matrices over F
  ex: `V = MatrixSpace(QQ,2,3)`
  space of 2×3 matrices with rational entries

EIGENVECTORS and Eigenvalues

- `A.eigenvalues()`
  List of eigenvalues of A
- `A.eigenvectors_right()`
  List of triples:
  (eval, evector, multiplicity)
- `A.characteristic_polynomial()`
  Characteristic polynomial of A

SOLVING a System of Equations

Example system with solution $x = -1, y = -5$

$$3x - y = 2$$
$$-x - y = 6$$

Method 1: Define augmented matrix and reduce

```
A = Matrix([[3,-1,2],[-1,-1,6]])
S = A.rref()
S
```

prints

```
[ 1 0 -1 ]
[ 0 1 -5 ]
```

Method 2: Define coefficient matrix and vector, then solve

```
A = Matrix([[3,-1],[-1,-1]])
b = vector([2,6])
S = A.solve_right(b)
S
```

prints

```
(-1, -5)
```

CONSTANTS and Basic Math

- Sage constants: `pi`, `e`, `i`, `oo`, `log2`
- Products
  `a = 4*9`
- Powers
  `a = 8^2`
- Roots
  `a=sqrt(16)`

SYNTAX

- List: `[1,2,3]`
- Tuple: `(1,2,3)`
- Set: `{1,2,3}`
- `range(n)`
  List $0, 1, 2, \ldots, n-1$
- `range(m,n)`
  List $m, m+1, \ldots, n-1$
- Comments: Start line with #
- Comparison: ==, <, >, <=, >=, !=

OTHER Useful Functions

- Solve equation
  `solve(x^2-4==0, x)`
  `solve(x^2-4==0, x)`
- Roots of polynomial
  `(x^2-3*x).roots(x)`
- Factor polynomial
  `(x^2+8*x-9).factor()`
- Limits
  `limit(x^2+5*x, x=2)`
- Derivatives
  `diff(x^2+4*x, x)`
- Antiderivative
  `integral(x^2+4*x, x)`
- Definite Integral
  `integral(x^2+4*x, x, 0, 4)`
- Simplify Expressions
  `((x^2-4)/(x+2)).simplify_rational()`
  `((x-2)*(x+6)).expand()`

PLOTTING

See Sage documentation for a complete list of plot options, such as titles, colors, line widths, fills, and much more.

- `plot(x^2,(x,-4,20))`
- `circle( (3,5), 2 )`
- `line([(1,5),(3,7)])`
- `polygon([(0,0), (1,3), (2,5), (0,4)])`
- ```
  L = plot(x^2, (x,-2,2), rgbcolor=(1,0,0))
  N = plot(x^4, (x,-2,2), rgbcolor=(0,0,1))
  show(L+N)
  ```
- `bar_chart([3,5,-1,2,8,4,3,2,5] )`
- ```
  M = [[1,3,4,2],[2,4,1,0]]
  matrix_plot(M, colorbar=True)
  ```
- ```
  L = [[1,3],[2,5],[2,7],[1.8,2.9]]
  scatter_plot(L)
  ```
  ```