

### Vectors, Matricies, Indexing

| MATLAB  | R  | Description  |
|---|--|--|
| <code>a=[1 2 3 4];</code>                               | <code>a &lt;- c(1,2,3,4)</code>  | Row vector   |
| <code>a=[1;2;3;4]</code> OR <code>a=[1 2 3 4]'</code> ; | <code>a &lt;- t(c(1,2,3,4))</code>   | Column vector  |
| <code>a=[1 2 3; 4 5 6];</code>                          | <code>matrix(c(1,2,3,4,5,6),nrow=2,byrow=TRUE)</code>  | Matrix, enter by row   |
| <code>a=[1 2 3; 4 5 6];</code>                          | <code>matrix(c(1,4,2,5,3,6),nrow=2)</code>   | Matrix, enter by column  |
| <code>a(3)</code>                                       | <code>a[3]</code>  | Access element #3  |
| <code>a=[2:7];</code>                                   | <code>a &lt;- 2:7</code> OR <code>a &lt;- c(2:7)</code>  | Vector [2 3 4 5 6 7]   |
| <code>a=[7:-1:2];</code>                                | <code>a &lt;- 7:2</code>   | Vector [7 6 5 4 3 2]   |
| <code>a=[2:3:14];</code>                                | <code>a &lt;- seq(2,14,3)</code>   | Vector [2 5 8 11 14]   |
| <code>x=linspace(a,b,n);</code>                         | <code>x &lt;- seq(a,b,length.out=n)</code> OR <code>seq(a,b,len=n)</code>                        | Vector with n equally spaced values between a and b, inclusive                           |
| <code>a=zeros(a,1)</code> OR <code>a=zeros(1,a)</code>  | <code>a &lt;- rep(0,k)</code>  | Vector of length k with all zeros  |
| <code>a=jones(a,1)</code> OR <code>a=jones(1,a)</code>  | <code>a &lt;- rep(j,k)</code>  | Vecot of length k with all values j  |
| <code>a=zeros(m,n)</code>                               | <code>matrix(0,nrow=m,ncol=n)</code> OR <code>matrix(0,m,n)</code>                               | Matrix of all zeros, size m by n   |
| <code>a=j*ones(m,n)</code>                              | <code>matrix(j,m,n)</code>   | Matrix of all j, size m by n   |
| <code>a=[a1 a2];</code>                                 | <code>a &lt;- cbind(a1,a2)</code>  | "Glue" two matricies a1 and a2 together (same # of rows)                                 |
| <code>[X,Y]=meshgrid(x,y)</code>                        | <code>m=length(x); n=length(y); X=matrix(rep(x,each=n),nrow=m); Y=matrix(rep(y,m),nrow=n)</code> | X rows are copies of x, Y columns are copies of y  |
| <code>A(:,2)</code>                                     | <code>A[,2]</code>   | Column 2 of matrix A   |
| <code>A(7,:)</code>                                     | <code>A[7,]</code>   | Row 7 of matrix A  |
| <code>flipr(A)</code>                                   | <code>t(apply(A,1,rev))</code>   | Flip the order of elements in each row of matrix A                                       |
| <code>flipud(A)</code>                                  | <code>apply(A,2,rev)</code>  | Flip the order of elements in each column of matrix A                                    |
| <code>v(a:end)</code>                                   | <code>v[a:length(v)]</code>  | Extract elements of v from positon a to end  |
| No simple way   | <code>v[-j,-k]</code>  | All but the jth and kth elements of v  |
| <code>A = reshape(A,m,n)</code>                         | <code>dim(A) &lt;- c(m,n)</code>   | Reshape matrix A into an m by n matrix (take elements columnwise from original matrix A) |



By **[deleted]**  
[cheatography.com/deleted-30394/](https://cheatography.com/deleted-30394/)

Not published yet.  
 Last updated 3rd November, 2017.  
 Page 1 of 3.

Sponsored by **ApolloPad.com**  
 Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>

### Cell Array/List

| MATLAB  | R  | Description  |
|---|--|--|
| <code>v=cell(1,n); v{1}=12; v{2}='hi there'; v{3}=rand(3);</code> | <code>v &lt;- vector('list',n); v[[1]]=12; v[[2]]='hi there'; v[[3]] = matrix(runif(9),3)</code> | Vector of length n capable of containing different data types in different elements (aka cell or list) |
| <code>w=v{1}</code>   | <code>w=v[[1]]</code>  | Extract the ith element of cell/list vector  |
| No names associated with elements of cell arrays                  | <code>names(v)[3] &lt;- 'myrandmatrix'</code>  | Set the name of the ith element in a list  |
| No names associated with elements of cell arrays                  | <code>names(v)</code>  | See all names of elements in list  |
| No names associated with elements of cell arrays                  | <code>names(v) = NULL</code>   | Clear all names  |

### Structures/Data Frames

|   |  |  |
|---|--|--|
| <code>avals=2*ones(1,6); yvals=6:-1:1; v=[1 5 3 2 3 7]; d = struct('a',avals,'y',yvals,'fac',v);</code> | <code>v &lt;- c(1,5,3,2,3,7); d &lt;- data.frame(cbind(a=2, y=6:1,v))</code> | Create a matrix-like object with different named columns ( <i>structure</i> in MATLAB, <i>data frame</i> in R) |
|---|--|--|

### Basic Computation

| MATLAB   | R  | Description  |
|--|--|--|
| <code>a=1; b=2;</code>                           | <code>a &lt;- 1; b &lt;- 2;</code>               | Variable assignment                                      |
| <code>a+b; a*b; a^b; etc</code>                  | <code>a+b; a*b; a^b; etc.</code>                 | Add, subtract, multiply, divide, power                   |
| --   | <code>a %/% b</code>                             | Integer division   |
| <code>abs(a); exp(a); log(a); log10(a);</code>   | <code>abs(a); exp(a); log(a); log10(a);</code>   | absolute value, e, ln, log base 10                       |
| <code>sin(a); asin(a); sinh(a); asinh(a);</code> | <code>sin(a); asin(a); sinh(a); asinh(a);</code> | sin, inverse sin, hyperbolic sin, inverse hyperbolic sin |
| <code>mod(n,k)</code>                            | <code>n %% k</code>                              | Remainder  |
| <code>round(x), floor(x), ceil(x)</code>         | <code>round(x), floor(x), ceil(x)</code>         | Round, round down, round up                              |
| <code>sign(x)</code>                             | <code>sign(x)</code>                             | Sign of x (+1, 0, or -1)                                 |

### Matrix Computations

| MATLAB                           | R  | Description   |
|----------------------------------|--|---|
| <code>dot(x,y)</code>            | <code>sum(x*y)</code>                        | Vector dot product  |
| <code>cross(x,y)</code>          | <code>xprod</code> in RSEIS package          | Vector cross product  |
| <code>A*B</code>                 | <code>A %*% B</code>                         | Matrix multiplication AB  |
| <code>A.*B</code>                | <code>A*B</code>                             | Element-by-element multiplication of A and B                                  |
| <code>A\b</code>                 | <code>solve(A,b)</code>                      | Solve Ax=b  |
| <code>mean(v), mean(A(:))</code> | <code>mean(v)</code> or <code>mean(A)</code> | Mean of all elements in vector or matrix                                      |
| <code>mean(A), sum(A)</code>     | <code>colMeans(A), colSums(A)</code>         | Means or sums of columns in a matrix  |
| <code>mean(A,2), sum(A,2)</code> | <code>rowMeans(A), rowSums(A)</code>         | Means or sums of rows in a matrix   |
| <code>std(v), std(A(:))</code>   | <code>sd(v), sd(c(A))</code>                 | Standard deviation of all elements in a vector of matrix, normalized by (n-1) |
| <code>std(A)</code>              | <code>sd(A)</code>                           | Standard deviations of columns of a matrix                                    |



### Matrix Computations (cont)

|                                    |   |   |
|------------------------------------|---|---|
| <code>std(A,2)</code>              | <code>apply(A,1,sd)</code>  | Standard deviations of rows of a matrix   |
| <code>min(v),<br/>min(A(:))</code> | <code>min(v), min(A)</code>   | Minimum of all elements in vector or matrix   |
| <code>min(A)</code>                | <code>apply(A,2,min)</code>   | Minimum value in each column of A   |
| <code>min(A,[],2)</code>           | <code>apply(A,1,min)</code>   | Minimum value in each row of matrix A   |
| <code>min(A,c)</code>              | <code>pmin(A,c)</code>  | Given matrix A and scalar c, compute a matrix where each element is the minimum of c and corresponding element of A |
| <code>[y,ind]=min(v)</code>        | <code>ind = which.min(v)</code>   | Find index of the first time min(v) appears in v and store that index as ind  |
| <code>size(A,1)</code>             | <code>nrow(A)</code>  | Number of rows in A   |
| <code>size(A,2)</code>             | <code>ncol(A)</code>  | Number of columns in A  |
| <code>size(A)</code>               | <code>dim(A)</code>   | Dimensions of A, listed as a vector   |
| <code>length(v)</code>             | <code>length(v)</code>  | Number of elements in vector  |
| <code>numel(A)</code>              | <code>length(A)</code>  | Number of elements in matrix  |
| <code>sort(v)</code>               | <code>sort(v)</code>  | Sort values in vector v   |
| <code>[s,idx]=sort(v)</code>       | <code>tmp &lt;- sort(v,index,return=TRUE); s &lt;- tmp\$s; idx=tmp\$ix</code>   | Sort values in v, putting sorted values in s and indices in idx, in teh sense that s[k] = x[idx[k]]                 |
| <code>find(v&gt;5)</code>          | <code>which(v&gt;5)</code>  | List of indices of each element of v which are greater than 5   |
| <code>[r,c] = find(A&gt;5)</code>  | <code>w &lt;- which(A&gt;5, arr.ind=TRUE); r &lt;- w[,1]; c &lt;- w[,2];</code> | Generate r and c giving rows and columns of elements of A which are greater than 5                                  |

### Relationals and Logicals

| MATLAB   | R  | Description          |
|--|--|----------------------|
| <code>a == b; a &lt; b; a &gt;= b; etc.</code> | <code>a == b; a &lt; b; a &gt;= b; etc.</code> | eq, gt, lt, gte, lte |
| <code>a ~= b</code>                            | <code>a != b</code>                            | Not equal            |
| <code>a &amp;&amp; b; a    b;</code>           | <code>a &amp;&amp; b; a    b;</code>           | AND, OR              |
| <code>~a</code>                                | <code>!a</code>                                | Not                  |

### GUI Commands

| MATLAB                 | R                           | Description         |
|------------------------|-----------------------------|---------------------|
| <code>clear all</code> | <code>rm(list=ls())</code>  | Clear all variables |
| <code>close all</code> | <code>graphics.off()</code> | Close all figures   |
| <code>clc</code>       | <code>ctrl+L</code>         | Clear console       |

