

Examples of Using R with Linear Algebra

S. K. Hyde

August 28, 2007

Contents

1	Introduction	2
2	Creating Vectors and Matrices	3
2.1	Create a Vector: <code>c</code>	3
2.2	Create a Sequence of Numbers: <code>:</code>	3
2.3	Create a Sequence of Numbers: <code>seq</code>	3
2.4	Create a Vector by Replication: <code>rep</code>	4
2.5	Create a Matrix: <code>matrix</code>	4
2.6	Combine Rows or Columns: <code>rbind</code> , <code>cbind</code>	6
3	Extracting Elements from a Vector or Matrix	6
3.1	Vectors	6
3.1.1	Extract an Element: <code>x[n]</code>	6
3.1.2	Extract All Elements Except an Element: <code>x[-n]</code>	7
3.1.3	Extract a Sequence of Elements: <code>x[1:n]</code>	7
3.1.4	Extract a Sequence of Elements: <code>x[-(1:n)]</code>	7
3.1.5	Extract Many Specific Elements: <code>x[c(1,5,9)]</code>	8
3.1.6	Extract Elements Meeting a Criteria: <code>x[x>2]</code>	8
3.1.7	Extract Elements Meeting a Criteria: <code>x[x>2 & x<5]</code>	8
3.2	Matrices	8
3.2.1	Extract an Element of a Matrix: <code>A[i,j]</code>	9
3.2.2	Extract a Row of a Matrix: <code>A[i,]</code>	9
3.2.3	Extract a Column of a Matrix: <code>A[,j]</code>	9

3.2.4	Extract a Series of Rows: <code>A[5:n,]</code>	9
3.2.5	Extract Specific Columns: <code>A[,c(1,4,5)]</code>	10
3.3	General Information Commands	10
3.3.1	Dimension of a Matrix: <code>dim</code>	10
3.3.2	Number of Elements in a Matrix or Vector: <code>length</code>	11
3.3.3	Number of Rows or Columns: <code>nrow</code> , <code>ncol</code>	11
3.3.4	Dimension Names of a Matrix: <code>dimnames</code>	11
4	Vector and Matrix Operations	12
4.1	Add Vectors of Matrices: <code>+</code>	12
4.2	Subtract Vectors or Matrices: <code>-</code>	13
4.3	Multiply Vectors or Matrices: <code>%*%</code> , <code>*</code>	13
4.4	Transpose of a Matrix: <code>t</code>	14
4.5	Extract a Diagonal of a Matrix, or Create a Diagonal Matrix: <code>diag</code>	14
4.6	Inverse of a Matrix or Solve a System of Equations: <code>solve</code>	14
4.7	Create $\mathbf{A}^\top \mathbf{A}$ efficiently: <code>crossprod</code>	15
4.8	Sum the Elements of a Vector or Matrix: <code>sum</code>	15
4.9	Multiply the Elements of a Vector or Matrix: <code>prod</code>	16
4.10	Cumulative Sum of a Vector: <code>cumsum</code>	16
4.11	Cumulative Product of a Vector: <code>cumprod</code>	16
4.12	Minimum Value of a Vector or Matrix: <code>min</code>	16
4.13	Maximum Value of a Vector or Matrix: <code>max</code>	17
4.14	Determinant of a Matrix: <code>det</code>	17
5	Matrix Factorizations	17

1 Introduction

The R statistical programming environment is a free implementation of the S computer language. In addition to being used to analyze statistical problems, it can also be used to do linear algebra, as it has matrix and array capabilities. There are other programs that can be used, such as Matlab, Mathematica, and Maple. In our class, we will use the program R.

To install R, consult our class website links. There is a link there to give you instructions on installing R for windows, which I'll assume that most of you have. If you don't have windows, talk with me and

we'll work on installing it for you.

To start up R, you either type R at a command prompt (Linux), click on the R icon on the desktop (windows), or find R in the start menu (windows). Once an R window has started, you type commands into the command window. For example,

```
> 23 + 49
```

```
[1] 72
```

The goal of this document is to give you a summary of the commands that can be used for Linear Algebra in R. These will be categorized into creation of matrices and vectors, extracting elements from matrices and vectors, operating on vectors and matrices, and matrix factorizations. For any of these commands, you can type `help(command)` in the command window to receive more detailed help.

2 Creating Vectors and Matrices

2.1 Create a Vector: `c`

To create a vector, the `c` command is used, which combines its arguments (think `c` for combine or concatenate). For example, to create a vector `v`, which consists of the numbers from 2 to 7, then

```
> v = c(2, 3, 4, 5, 6, 7)
> v
```

```
[1] 2 3 4 5 6 7
```

The `[1]` that is before the elements of the vector indicates that this vector is one dimensional. It might make more sense when you see how R denotes dimensions of a matrix later on. Note that vectors created in this way are typically thought of as column vectors, even though they are not written in that manner.

2.2 Create a Sequence of Numbers: `:`

The `:` command is useful to create a sequence of numbers that are separated from each other by 1. For example, a simpler way to create the vector which consists of the numbers from 2 to 7 would be

```
> v = 2:7
> v
```

```
[1] 2 3 4 5 6 7
```

2.3 Create a Sequence of Numbers: `seq`

Another command to create vectors is `seq`, which is short for sequence. The syntax of `seq` is `seq(from,to,by,length,along)`. Not all the arguments must be used, the only required arguments are `from` and `to`. For example, to create a vector of the even numbers from 2 to 20,

```
> v = seq(2, 20, by = 2)
> v
```

```
[1]  2  4  6  8 10 12 14 16 18 20
```

The `along` argument is used to specify the length of the vector you want. For example:

```
> v = seq(2, 20, along = 1:15)
> v
```

```
[1]  2.000000  3.285714  4.571429  5.857143  7.142857  8.428571  9.714286
[8] 11.000000 12.285714 13.571429 14.857143 16.142857 17.428571 18.714286
[15] 20.000000
```

This created a vector with 15 elements that are equally spaced from 2 to 20.

2.4 Create a Vector by Replication: `rep`

The `rep` command is used to create a vector in which elements are replicated. The simplest way to create a 10 dimensional vector of ones is

```
> v = rep(1, 10)
> v
```

```
[1] 1 1 1 1 1 1 1 1 1 1
```

You can also create more complicated repeating structures, such as

```
> v = rep(c(1, 2, 3), 3)
> v
```

```
[1] 1 2 3 1 2 3 1 2 3
```

which replicates the vector (1,2,3) three times. The `each` argument replicates each element before proceeding to the next element

```
> v = rep(c(1, 2, 3), each = 3)
> v
```

```
[1] 1 1 1 2 2 2 3 3 3
```

2.5 Create a Matrix: `matrix`

To create a 3×2 matrix that contains in the first column 2, 3, 4 and the second column 5, 6, 7,

```
> A = matrix(c(2, 3, 4, 5, 6, 7), 3, 2)
> A
```

```
      [,1] [,2]
[1,]    2    5
[2,]    3    6
[3,]    4    7
```

The [3,] indicates this is the third row, and the [,2] indicates it is the second column.

You need to make sure that the the number of entries in the `c()` equals the number of rows times the number of columns. If they aren't, R will either hack off the rest, or it will recycle arguments until the asked for size is reached. Don't assume it will give you an error or a warning! However, sometimes it might be exactly what you wanted.

```
> A = matrix(1:100, 2, 5)
> A
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]     1     3     5     7     9
[2,]     2     4     6     8    10
```

```
> B = matrix(c(2, 5, 7), 3, 8)
> B
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]     2     2     2     2     2     2     2     2
[2,]     5     5     5     5     5     5     5     5
[3,]     7     7     7     7     7     7     7     7
```

Notice that the creation of matrices works by first filling in a column before proceeding to the next column. To enter a matrix by the rows instead of the columns, then add the argument `byrow=TRUE` to the command parameters. That results with:

```
> A = matrix(c(2, 3, 4, 5, 6, 7), 3, 2, byrow = TRUE)
> A
```

```
      [,1] [,2]
[1,]     2     3
[2,]     4     5
[3,]     6     7
```

Matrices can also be entered by only specifying how many rows or columns instead of specifying the exact dimensions of the matrix. For example:

```
> A = matrix(c(2, 3, 4, 5, 6, 7), nrow = 3, byrow = TRUE)
> A
```

```
      [,1] [,2]
[1,]     2     3
[2,]     4     5
[3,]     6     7
```

2.6 Combine Rows or Columns: rbind, cbind

To combine vectors into a matrix, use the commands `cbind` or `rbind`. `cbind` combines them together as columns, whereas `rbind` combines them together as rows. The vectors should have the same dimension. If they don't, you will receive a warning, and it will attempt to recycle the arguments.

```
> u = 1:6
> v = 1:6
> w = c(1, 3, 4, 8, 9, 0)
> cbind(u, v)
```

```
      u v
[1,] 1 1
[2,] 2 2
[3,] 3 3
[4,] 4 4
[5,] 5 5
[6,] 6 6
```

```
> rbind(v, w)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
v      1    2    3    4    5    6
w      1    3    4    8    9    0
```

3 Extracting Elements from a Vector or Matrix

3.1 Vectors

The following commands can be used to extract elements from a vector. Examples of these commands follow.

<code>x[n]</code>	Extract the n^{th} element of <code>x</code> .
<code>x[-n]</code>	Extract everything except the n^{th} element of <code>x</code> .
<code>x[1:n]</code>	Extract elements 1 through n of <code>x</code> .
<code>x[-(1:n)]</code>	Extract all elements other than elements 1 through n
<code>x[c(1,5,9)]</code>	Extract elements 1, 5 and 9 from <code>x</code>
<code>x[x>2]</code>	Extract all elements of <code>x</code> that are larger than 2.
<code>x[x>2 & x<5]</code>	Extract all elements of <code>x</code> that are between 2 and 5.

3.1.1 Extract an Element: x[n]

To extract the n^{th} element of `x` use the command `x[n]`:

```
> x = seq(2, 20, by = 2)
> x
```

```
[1]  2  4  6  8 10 12 14 16 18 20
```

```
> x[5]
```

```
[1] 10
```

3.1.2 Extract All Elements Except an Element: $x[-n]$

To extract everything except the n^{th} element of x use the command $x[-n]$:

```
> x
```

```
[1] 2 4 6 8 10 12 14 16 18 20
```

```
> x[-5]
```

```
[1] 2 4 6 8 12 14 16 18 20
```

3.1.3 Extract a Sequence of Elements: $x[1:n]$

To extract elements 1 through n of x use the command $x[1:n]$:

```
> x
```

```
[1] 2 4 6 8 10 12 14 16 18 20
```

```
> x[1:5]
```

```
[1] 2 4 6 8 10
```

3.1.4 Extract a Sequence of Elements: $x[-(1:n)]$

To extract all elements other than elements 1 through n use the command $x[-(1:n)]$. In other words, this extracts the elements from $n + 1$ until the end of the vector.

```
> x
```

```
[1] 2 4 6 8 10 12 14 16 18 20
```

```
> x[-(1:5)]
```

```
[1] 12 14 16 18 20
```

3.1.5 Extract Many Specific Elements: $x[c(1,5,9)]$

To extract elements 1, 5 and 9 from x use the command $x[c(1,5,9)]$.

```
> x

[1]  2  4  6  8 10 12 14 16 18 20

> x[c(1, 5, 9)]

[1]  2 10 18
```

3.1.6 Extract Elements Meeting a Criteria: $x[x>2]$

To extract all elements of x that are larger than 2 use the command $x[x>2]$.

```
> x

[1]  2  4  6  8 10 12 14 16 18 20

> x[x > 2]

[1]  4  6  8 10 12 14 16 18 20
```

3.1.7 Extract Elements Meeting a Criteria: $x[x>2 \ \& \ x<5]$

To extract all elements of x that are between 2 and 5 use the command $x[x>2 \ \& \ x<5]$:

```
> x

[1]  2  4  6  8 10 12 14 16 18 20

> x[x > 2 & x < 5]

[1] 4
```

3.2 Matrices

The following commands can be used to extract elements, rows, etc. from a matrix. Examples of these commands follow.

$A[i,j]$	Extract the element in row i , column j .
$A[i,]$	Extract the i^{th} row of A .
$A[,j]$	Extract the j^{th} column of A .
$A[5:n,]$	Extract rows 5 through n of A .
$A[,c(1,4,5)]$	Extract columns 1, 4, and 5 from A

3.2.1 Extract an Element of a Matrix: A[i,j]

To extract the element in row i , column j , use the command `A[i,j]`:

```
> A = matrix(1:20, 4, 5)
> A
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

```
> A[3, 5]
```

```
[1] 19
```

3.2.2 Extract a Row of a Matrix: A[i,]

To extract the i^{th} row of `A`, use the command `A[i,]`:

```
> A[2, ]
```

```
[1]  2  6 10 14 18
```

3.2.3 Extract a Column of a Matrix: A[,j]

To extract the j^{th} column of `A`, use the command `A[,j]`:

```
> A[, 5]
```

```
[1] 17 18 19 20
```

3.2.4 Extract a Series of Rows: A[5:n,]

To extract rows 5 through n of `A`, use the command `A[5:n,]`:

```
> A[3:4, ]
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    3    7   11   15   19
[2,]    4    8   12   16   20
```

3.2.5 Extract Specific Columns: `A[,c(1,4,5)]`

To extract columns 1, 4, and 5 from `A`, use the command `A[,c(1,4,5)]`:

```
> A[, c(1, 4, 5)]

      [,1] [,2] [,3]
[1,]    1   13   17
[2,]    2   14   18
[3,]    3   15   19
[4,]    4   16   20
```

3.3 General Information Commands

The following commands can be used to gather information about matrices or vectors.

<code>dim</code>	Displays or sets the dimension of a matrix
<code>length</code>	Displays the number of elements in a matrix or vector.
<code>nrow</code>	Displays the number of rows in a matrix.
<code>ncol</code>	Displays the number of columns in a matrix.
<code>dimnames</code>	Displays or sets the name of the the dimensions of the vector or matrix.

3.3.1 Dimension of a Matrix: `dim`

To display or set the dimension of a matrix, use the command `dim`:

```
> A = matrix(1:10, 5, 2)
> A
```

```
      [,1] [,2]
[1,]    1    6
[2,]    2    7
[3,]    3    8
[4,]    4    9
[5,]    5   10
```

```
> dim(A)
```

```
[1] 5 2
```

```
> dim(A) = c(2, 5)
> A
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

Note that this is **NOT** the transpose of ***A***!

3.3.2 Number of Elements in a Matrix or Vector: length

To display the number of elements in a matrix or vector, use the command `length`:

```
> A

      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

```
> length(A)
```

```
[1] 10
```

```
> v
```

```
[1] 1 2 3 4 5 6
```

```
> length(v)
```

```
[1] 6
```

3.3.3 Number of Rows or Columns: nrow, ncol

To display the number of rows or columns in a matrix, use the commands `nrow` or `ncol`:

```
> nrow(A)
```

```
[1] 2
```

```
> ncol(A)
```

```
[1] 5
```

3.3.4 Dimension Names of a Matrix: dimnames

To display or set the name of the the dimension of the vector or matrix, use the command `dimnames`:

```
> dimnames(A)
```

```
NULL
```

```
> u = c(1, 2, 3)
> v = c(4, 5, 6)
> W = cbind(u, v)
> W
```

```

      u v
[1,] 1 4
[2,] 2 5
[3,] 3 6

> dimnames(W)

[[1]]
NULL

[[2]]
[1] "u" "v"

> dimnames(W) = c(NULL, NULL)
> W

      [,1] [,2]
[1,]     1     4
[2,]     2     5
[3,]     3     6

```

4 Vector and Matrix Operations

The following commands perform operations on matrices and vectors.

<code>+</code>	Add two matrices/vectors
<code>-</code>	Subtract two matrices/vectors
<code>%*%</code>	Multiply two matrices
<code>*</code>	Elementwise multiplication of two matrices/vectors
<code>t</code>	Transpose of the matrix.
<code>diag</code>	Extracts the diagonal of a matrix or creates a diagonal matrix from a vector.
<code>solve</code>	Finds an inverse of a matrix or solves a system of equations)
<code>crossprod</code>	Creates the crossproduct $\mathbf{A}^\top \mathbf{A}$ in an efficient manner.
<code>sum</code>	sum of the elements
<code>prod</code>	product of the elements
<code>cumsum</code>	cumulative sum of the elements
<code>cumprod</code>	cumulative product of the elements
<code>min</code>	Finds the minimum value in a matrix
<code>max</code>	Finds the maximum value in a matrix
<code>det</code>	Calculates the determinant of a matrix

An example of each of the above commands follows.

4.1 Add Vectors of Matrices: `+`

To add matrices or vectors elementwise, the operator `+` is used:

```

> A = matrix(c(1:5, 11:14), 3, 3)
> B = matrix(1:9, 3, 3)
> A

```

```

      [,1] [,2] [,3]
[1,]    1    4   12
[2,]    2    5   13
[3,]    3   11   14

```

```
> B
```

```

      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

```

```
> A + B
```

```

      [,1] [,2] [,3]
[1,]    2    8   19
[2,]    4   10   21
[3,]    6   17   23

```

4.2 Subtract Vectors or Matrices: -

To subtract two matrices/vectors, use the command, use the command -:

```
> A - B
```

```

      [,1] [,2] [,3]
[1,]    0    0    5
[2,]    0    0    5
[3,]    0    5    5

```

4.3 Multiply Vectors or Matrices: %*%, *

To multiply two matrices together, the %*% operator is used.

```
> A %*% B
```

```

      [,1] [,2] [,3]
[1,]   45   96  147
[2,]   51  111  171
[3,]   67  151  235

```

Note that this is different than the *, which performs an elementwise multiplication:

```
> A * B
```

```

      [,1] [,2] [,3]
[1,]    1   16   84
[2,]    4   25  104
[3,]    9   66  126

```

4.4 Transpose of a Matrix: `t`

To transpose a matrix, use the command `t`:

```
> t(A)

      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5   11
[3,]   12   13   14
```

4.5 Extract a Diagonal of a Matrix, or Create a Diagonal Matrix: `diag`

To extract the diagonal of a matrix or create a diagonal matrix from a vector, use the command `diag`:

```
> diag(A)

[1]  1  5 14

> diag(c(1, 1, 1))
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

```
> diag(diag(A))
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    5    0
[3,]    0    0   14
```

4.6 Inverse of a Matrix or Solve a System of Equations: `solve`

To find the inverse of a matrix or solve a system of equations, use the command `solve`:

```
> solve(A)

      [,1]      [,2]      [,3]
[1,] -1.3272727  1.38181818 -0.14545455
[2,]  0.2000000 -0.40000000  0.20000000
[3,]  0.1272727  0.01818182 -0.05454545

> b = c(2, 3, 4)
> x = solve(A, b)
> x
```

```
[1] 9.09091e-01 -4.79411e-17 9.09091e-02
```

You can also use the `fractions` command from the `MASS` package to convert the answer to fractions:

```
> require(MASS)
> fractions(x)
```

```
[1] 10/11      0 1/11
```

4.7 Create $A^T A$ efficiently: `crossprod`

To create the crossproduct $A^T A$ in an efficient manner, use the command `crossprod`:

```
> crossprod(A)

      [,1] [,2] [,3]
[1,]    14    47    80
[2,]    47   162   267
[3,]    80   267   509

> t(A) %% A

      [,1] [,2] [,3]
[1,]    14    47    80
[2,]    47   162   267
[3,]    80   267   509
```

4.8 Sum the Elements of a Vector or Matrix: `sum`

To sum the elements of a vector or a matrix, use the command `sum`:

```
> b

[1] 2 3 4

> sum(b)

[1] 9

> A

      [,1] [,2] [,3]
[1,]     1     4    12
[2,]     2     5    13
[3,]     3    11    14

> sum(A)

[1] 65
```

4.9 Multiply the Elements of a Vector or Matrix: prod

To multiply the elements of a vector or a matrix, use the command `prod`:

```
> prod(1:5)
```

```
[1] 120
```

```
> prod(c(2, 3, 5))
```

```
[1] 30
```

4.10 Cumulative Sum of a Vector: cumsum

To create a vector of cumulative sums of the elements of a vector, use the command `cumsum`. If used on a matrix, the matrix is first changed to a vector.

```
> b
```

```
[1] 2 3 4
```

```
> cumsum(b)
```

```
[1] 2 5 9
```

4.11 Cumulative Product of a Vector: cumprod

To create a vector of cumulative products of the elements of a vector, use the command `cumprod`. If used on a matrix, the matrix is first changed to a vector.

```
> b
```

```
[1] 2 3 4
```

```
> cumprod(b)
```

```
[1] 2 6 24
```

4.12 Minimum Value of a Vector or Matrix: min

To find the minimum value in a vector or matrix, use the command `min`:

```
> min(A)
```

```
[1] 1
```


4.13 Maximum Value of a Vector or Matrix: `max`

To find the maximum value in a vector or matrix, use the command `max`:

```
> max(A)
```

```
[1] 14
```

4.14 Determinant of a Matrix: `det`

To calculate the determinant of a square matrix, use the command `det`:

```
> det(A)
```

```
[1] 55
```

5 Matrix Factorizations

To be completed!