An Introduction to R

© 2009 Dan Nettleton

Preliminaries

- Throughout these slides, red text indicates text that is typed at the R prompt or text that is to be cut from a text file and pasted to the R prompt.
- Blue text indicates R output.
- Black text indicates notes about the code and output.
- To read the help file for any R command type `?command` at the R prompt. For example, to read about the mean function:

```r
> ?mean
```

> 7

[1] 7

The 1 in square brackets marks the element number in the output. For example, the next command generates 20 standard normal random numbers. The [8] indicates that -0.6693018 is the 8th number of the 20.

```r
> rnorm(20,0,1)
```

[1] 0.7471442 -0.1930792 -1.4760422 -0.3172517 -0.3997149

[6] -0.6693018 0.4165647 0.8298968 1.0748766 1.3025841

[11] 0.4675156 -0.5274312 0.2418231 0.9012613 0.6674558

[16] -0.2037051 1.0914826 -1.5400979 -1.3983107 -1.0940496

> 6+4/2

[1] 8

> (6+4)/2

[1] 5

> (6+4)/+

[1] 5

If you give R 7, it will give you back 7.

If you press the enter key before completing a command, the R prompt becomes a +, which indicates that anything typed after the + will be appended to the line above for evaluation.

```r
dx=c(4,-1,5,9,4.5)
dx
```

[1] 4.0 -1.0 5.0 9.0 4.5

This command combines the numbers into a vector and assigns the vector to an object called `x`. The vector is printed to the screen when `x` is typed at the R prompt.

```r
> x=c(4,-1,5,9,4.5)
x
```

[1] 4.0 -1.0 5.0 9.0 4.5

```r
> x+1
```

[1] 5.0 0.0 6.0 10.0 5.5

This command creates (but does not store) a new vector whose elements are the elements `x` with 1 added to each. Note that the command does not change `x`.

This command requests the 1st and 5th elements of the vector `x`.

```r
> x=sample(1:5)
x
```

[1] 2 1 4 3 5

The vector `x` was not changed by the previous commands.

```r
> y=sample(1:5)
y
```

[1] 2 1 4 3 5

```r
> y[x<0]=0
```

y

[1] 2 0 1 3 5

This command finds the elements of `y` that correspond to elements of `x` that are less than 0, and then replaces all those elements of `y` with 0. In this case, only the 2nd element of `x` is less than 0, so only the 2nd element of `y` gets replaced with 0.

```r
> y[x<0]=99
```

y

[1] 2 99 4 3 5

This command finds the elements of `y` that are exactly equal to 0 and then replaces all those elements of `y` with 99.

```r
> y[x<99]=1
```

y

[1] 2 1 3 4 5

This command finds the elements of `y` that are less than 99 and replaces them all with 1.

```r
> y[y<0]=1:4
```

y

[1] 1 99 2 3 4

This command finds the elements of `y` that are less than 0 and replaces the first such element with 1, the second with 2, the third with 3, and the fourth with 4.
### Example 1: Finding Elements in a Matrix

**Problem:**
Find the value in the second row and second column of the matrix `m`.

**Solution:**
```r
> m = matrix(1:9, nrow=3)
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```
Find the value in the second row and second column of the matrix m.

**Answer:**
```r
> m[2,2]
[1] 5
```

### Example 2: Sorting a Vector

**Problem:**
Sort the vector `x`.

**Solution:**
```r
> x = c(-1.0, 4.0, 4.5, 5.0, 9.0)
> x
[1] -1.0  4.0  4.5  5.0  9.0
> x = sort(x)
> x
[1]  4.0 -1.0  5.0  9.0  4.5
```
Sort the vector x.

**Answer:**
```r
> x
[1]  4.0 -1.0  5.0  9.0  4.5
```

### Example 3: Rank of a Vector

**Problem:**
Find the rank of the vector x.

**Solution:**
```r
> rank(x)
[1] 2 1 5 3 4
```
Find the rank of the vector x.

**Answer:**
```r
> rank(x)
[1] 2 1 5 3 4
```

### Example 4: Component-wise Multiplication

**Problem:**
Find the product of the vector `x` and its elements.

**Solution:**
```r
> x*(x <= 0)
[1]  4.0 -2.0 15.0 36.0 22.5
```
Find the product of the vector x and its elements.

**Answer:**
```r
> x*(x <= 0)
[1]  4.0 -2.0 15.0 36.0 22.5
```

### Example 5: Finding Submatrices

**Problem:**
Find the submatrix consisting of the first two rows and the first and third columns of the matrix m.

**Solution:**
```r
> m = matrix(1:9, nrow=3)
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```
Find the submatrix consisting of the first two rows and the first and third columns of the matrix m.

**Answer:**
```r
> m[1:2, c(1,3)]
     [,1] [,2]
[1,]    1    3
[2,]    4    6
```
Find the submatrix consisting of the first two rows and the first and third columns of the matrix m.

**Answer:**
```r
> m[1:2, c(1,3)]
     [,1] [,2]
[1,]    1    3
[2,]    4    6
```
> m
[1,]   -4   -3   -2
[2,]   -1    0    1
[3,]    2    3    4

> sum(m)
[1] 0

> apply(m,1,sum)
[1] -9  0  9

> apply(m,2,median)
[1] -1  0  1

> sweep(m,1,c(2,9,4),"")
[,1]    -8   -6   -4
[,2]    -9    0    9
[,3]    8   12   16

Add up all the elements of the matrix m.
Separately for each row of m, sum the values.
Separately for each column of m, find the median of the values.
(1 means rows and 2 means columns.)
This command creates a new matrix whose 1st row is the 1st row of m multiplied by 2,
whose 2nd row is the 2nd row of m multiplied by 9,
and whose 3rd row is the 3rd row of m multiplied by 4.

> m=matrix(sample(1:9),nrow=3)
> m
[,1]    8    4    1
[,2]    5    6    3
[,3]    9    2    7

> sweep(m,2,apply(m,2,median))
[,1]   -6  -12   -6
[,2]   -3   -9   -3
[,3]    0   -6    0

This command combines sweep and apply to subtract the column median from the values in each column.
This command creates a new matrix whose elements are a random shuffling of the integers 1,2,...,9.
This command creates a new matrix whose 1st row is the 1st row of m multiplied by 2,
whose 2nd row is the 2nd row of m multiplied by 9,
and whose 3rd row is the 3rd row of m multiplied by 4.

> x=c(3,3,9,6,12,15,18,12)
> c(9,4,3,8)%in%x
[1] TRUE FALSE TRUE FALSE
> table(x)
x    3  6  9 12 15 18
n    2  1  1  2  1  1
> unique(x)
[1]  3  9  6 12 15 18
> max(x)
[1] 18
> max(x)
[1] 18
> which.max(x)
[1] 7
> which(x<7)
[1] 1 2 4

A vector is created that is TRUE for each element in the first vector that is contained in the second vector, and FALSE for each element in the first vector that is not in the second vector.
The number of times that each value appears in x is reported in a table.  For example, 3 appears 2 times in x, 6 appears 1 time, 9 appears 1 time, 12 appears 2 times, etc.
A new vector is created that has only one value for each different value in x.  For example, although x has two 3's, unique(x) has only one 3.
The largest value of x is returned.
The element number (index) of the largest value of x is returned.
The indices of the values of x that are less than 7 are returned.