R Tutorial-01

T.Scofield

R is

- a programming language
- equipped to do large-scale scientific computing, particularly statistical
- interpreted, not compiled
- readily supplemented by packages

Built-in functions for mathematical computations

- usual binary arithmetic operators

```
r = 2
volume = 4/3 * pi * r^3
volume
```

```
[1] 33.51032
```

- absolute value

```
abs(-3)
```

```
[1] 3
```

- square roots

```
sqrt(2)
```

```
[1] 1.414214
```

- exponentials and logarithms

```
exp(2)
```

```
[1] 7.389056
```

```
log( exp(2) )
```

```
[1] 2
```

- counting functions: `factorial()`, `choose()`

```
factorial(0)
```

```
[1] 1
```

```
factorial(6)
```

```
[1] 720
```

```
choose(5,2)
```

```
[1] 10
```

Vectors in R

The `c()` command is a convenient constructor of a vector.

```
visitors = c(211, 172, 194, 318, 325)
visitors
```

```
[1] 211 172 194 318 325
```

```
sort(visitors)   # shows entries sorted lowest-to-highest
```

```
[1] 172 194 211 318 325
```

Vectors, or arrays, in R, are one-indexed, but can be accessed by a criterion:

```
visitors[1]
```

```
[1] 211
```

```
visitors[c(3,2,5)]
```

```
[1] 194 172 325
```

```
visitors[visitors > 300]
```

```
[1] 318 325
```

As to what the criterion does, specifically:

```
visitors > 300
```

```
[1] FALSE FALSE FALSE  TRUE  TRUE
```

```
which(visitors > 300)
```

```
[1] 4 5
```

```
which(visitors == 194)
```

```
[1] 3
```

There are some other convenient ways to create vectors. Here are some examples:

```
seq(2,3,.1)
```

```
 [1] 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0
```

```
5:2
```

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

```
rep(6.5, 4)
```

```
[1] 6.5 6.5 6.5 6.5
```

and one can combine these methods:

```
c(2:7, 6:2)
```

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

```
c(rep("red",2), rep("black",3), rep("red",3))
```

```
[1] "red"   "red"   "black" "black" "black" "red"   "red"   "red"
```

Data frames (like 2D arrays)

- many are supplied through loaded packages
- to learn the layout of a data frame one can use `names()`, `dim()`, `nrow()`, `head()`, `help()`, `glimpse()`, `inspect()`

```
names(iris)  # shows column/variable names
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```
nrow(iris)   # tells the number of cases/rows
```

```
[1] 150
```

```
head(iris)   # shows, by default, first 6 cases/rows
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

- `data()` displays data sets currently available
- each row is a case/subject/unit

```
iris[61, 3]   # the entry on row 61, column 3
```

```
[1] 3.5
```

```
iris[61,]     # all measurements taken on case 61
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
61            5           2          3.5           1 versicolor
```

```
iris[, 3]     # vector of all values in column 3
```

```
  [1] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 1.5 1.6 1.4 1.1 1.2 1.5 1.3 1.4
 [19] 1.7 1.5 1.7 1.5 1.0 1.7 1.9 1.6 1.6 1.5 1.4 1.6 1.6 1.5 1.5 1.4 1.5 1.2
 [37] 1.3 1.4 1.3 1.5 1.3 1.3 1.3 1.6 1.9 1.4 1.6 1.4 1.5 1.4 4.7 4.5 4.9 4.0
 [55] 4.6 4.5 4.7 3.3 4.6 3.9 3.5 4.2 4.0 4.7 3.6 4.4 4.5 4.1 4.5 3.9 4.8 4.0
 [73] 4.9 4.7 4.3 4.4 4.8 5.0 4.5 3.5 3.8 3.7 3.9 5.1 4.5 4.5 4.7 4.4 4.1 4.0
 [91] 4.4 4.6 4.0 3.3 4.2 4.2 4.2 4.3 3.0 4.1 6.0 5.1 5.9 5.6 5.8 6.6 4.5 6.3
[109] 5.8 6.1 5.1 5.3 5.5 5.0 5.1 5.3 5.5 6.7 6.9 5.0 5.7 4.9 6.7 4.9 5.7 6.0
[127] 4.8 4.9 5.6 5.8 6.1 6.4 5.6 5.1 5.6 6.1 5.6 5.5 4.8 5.4 5.6 5.1 5.1 5.9
[145] 5.7 5.2 5.0 5.2 5.4 5.1
```

```
iris$Petal.Length   # same result as last command
```

```
  [1] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 1.5 1.6 1.4 1.1 1.2 1.5 1.3 1.4
 [19] 1.7 1.5 1.7 1.5 1.0 1.7 1.9 1.6 1.6 1.5 1.4 1.6 1.6 1.5 1.5 1.4 1.5 1.2
 [37] 1.3 1.4 1.3 1.5 1.3 1.3 1.3 1.6 1.9 1.4 1.6 1.4 1.5 1.4 4.7 4.5 4.9 4.0
 [55] 4.6 4.5 4.7 3.3 4.6 3.9 3.5 4.2 4.0 4.7 3.6 4.4 4.5 4.1 4.5 3.9 4.8 4.0
 [73] 4.9 4.7 4.3 4.4 4.8 5.0 4.5 3.5 3.8 3.7 3.9 5.1 4.5 4.5 4.7 4.4 4.1 4.0
 [91] 4.4 4.6 4.0 3.3 4.2 4.2 4.2 4.3 3.0 4.1 6.0 5.1 5.9 5.6 5.8 6.6 4.5 6.3
[109] 5.8 6.1 5.1 5.3 5.5 5.0 5.1 5.3 5.5 6.7 6.9 5.0 5.7 4.9 6.7 4.9 5.7 6.0
[127] 4.8 4.9 5.6 5.8 6.1 6.4 5.6 5.1 5.6 6.1 5.6 5.5 4.8 5.4 5.6 5.1 5.1 5.9
[145] 5.7 5.2 5.0 5.2 5.4 5.1
```

```
iris$Petal.Length[61]   # like iris[61, 3]
```

```
[1] 3.5
```

Loading already-installed packages

- Can use the Packages tab and check desired library
- Can use `require()` or `library()`

```
require(fosdata)
```

```
Loading required package: fosdata
```

```
data(package="fosdata")
```

- packages to load regularly include **fosdata** and **mosaic**