

TUTORIAL 1

STA437 WINTER 2015

AL NOSEDAL

CONTENTS

1. An Introduction to R	1
1.1. What is R?	1
1.2. Installing R	1
1.3. Help and Documentation	2
2. Vectors and Matrices	2
3. Problems	4
4. Solutions	6

1. AN INTRODUCTION TO R

1.1. **What is R?** The R system for statistical computing is an environment for data analysis and graphics. The base distribution of R and a large number of user contributed extensions are available under the terms of the Free Software Foundation's GNU General Public License in source code form. The main source of information about the R system is the world wide web with the official home page of the R project being

<http://www.R-project.org>

All resources are available from this page: the R system itself, a collection of add-on packages, manuals, documentation and more.

1.2. **Installing R.** The R system for statistical computing consists of two major parts: the base system and a collection of user contributed add-on packages. A package is a collection of functions, examples and documentation. Both the base system and packages are distributed via the Comprehensive R Archive Network (CRAN) accessible under

<http://CRAN.R-project.org>

1.2.1. *The Base System and the First Steps.* The base system is available in source form and in precompiled form for various Unix systems, Windows platforms and Mac OS X. For us, it will be sufficient to download the precompiled binary distribution and install it locally. Just go to

`http://CRAN.R-project.org`

download the corresponding file (Download R for Linux or Download R for (Mac) OS X or Download R for Windows), execute it locally and follow the instructions given by the installer. Depending on the operating system, R can be started either by typing 'R' on the shell (Unix systems) or by clicking on the R symbol created by the installer (Windows).

1.2.2. *Packages.* Packages not included in the base distribution can be installed directly from the R prompt. Given that an Internet connection is available, a package is installed by supplying the name of the package to the function **install.packages**.

If, for example, we need the *MVA* package, it can be downloaded and installed via

```
install.packages("MVA")
```

The package functionality is available after attaching the package by `library("MVA")`

1.3. **Help and Documentation.** The help system is a collection of manual pages describing each user-visible function and data set that comes with R. A manual page is shown in a pager or web browser when the name of the function we would like to get help for is supplied to the help function

```
help("mean")
```

An overview on documented topics in an add-on package is given, for example for the *MVA* package, by

```
help(package="MVA")
```

2. VECTORS AND MATRICES

Two of the most commonly used types of R object are vectors and matrices. Vectors may be created in several ways, of which the most common is via the `c` function which combines all values from all arguments to the function. A simple example is:

```
x<-c(1,2,3,4)
```

```
x
```

```
y<-c(6,7,8)
```

```
c(y,x,y)
```

A powerful feature of R is that the usual arithmetic operators can be applied to vector objects with operations being performed element by element. So, for example:

```
# lines preceded by # are comments and ignored by R
```

```
x*x
```

```
# you should get: 1 4 9 16
```

```
x/x
```

```
# you should get: 1 1 1 1
```

The number of elements in a vector is extracted by the **length** function:

```
length(x)
```

```
# you should get 4
```

Matrix objects can be created from vectors by use of the **matrix** function. Some examples are:

```
x<-c(1,2,3,4,5,6)
```

```
X<-c(10,11,12,100,-5,-6)
```

```
M<-matrix(c(x,X),ncol=2)
```

```
# the vectors x and X are combined using the c function
```

```
# and then converted to a 2-column matrix by the matrix function
```

```
M
```

```
matrix(c(1,2,3,4,5,1,2,3,4),ncol=3)
```

As with vectors, arithmetic operations apply element by element when applied to matrices. So, for example

```
M*M
```

```
M/M
```

Matrix multiplication can be achieved using the `%*%` operator. An example is:

```
M1<-matrix(c(1,2,1,2),ncol=2)
```

```
# M1 will be a 2 x 2 matrix
```

```
M2<-matrix(c(3,4,5,6,7,8),ncol=3)
```

```
# M2 will be a 2 x 3 matrix
```

```
M1%*%M2
```

```
# the transpose of M2 multiplied by M1 can be obtained using
```

```
# the t function
```

```
t(M2)%*%M1
```

The `[i,]` and `[,j]` nomenclature is used in R to refer to complete rows or columns of a matrix, and can be used to extract particular rows or columns as illustrated in the following example.

```
Mrow1<-M[1, ]
```

```
Mrow1
```

```
M[ ,2]
```

Sub matrices can be selected in a similar fashion with the help of the **c** function:

```
M3<-matrix(1:10,ncol=5)
```

```
# the command 1:10 generates a vector containing the elements 1 to 10
```

```
M3
```

```
M3sub<-M3[ ,c(1,3,5)]
```

```
# selects columns 1,3, and 5
```

M3sub

Particular elements of a matrix can be accessed using the **[i,j]** nomenclature. For example:

```
x<-M3[1,2]
# element located at the first row, second column
x
```

Elements and subsets of elements of vectors and matrices can be excluded by specifying a negative integer or a set of negative integers. For example:

```
M3.sub<-M3[ ,c(1,5)]
# excludes columns 1 and 5
M3.sub
```

More complex 'subsetting' of vectors and matrices can be achieved by using the logical operators such as **>**, **<**, **<=** (less than or equal to), **>=** (greater than or equal to), **==** (equal to), etc.

```
X<-2*(1:20)
X[X <= 10]
Y<-rep(0,20)
# rep function is used to obtain a vector of zeros of length 20
Y[c(1,20)]<-1
#sets elements 1 and 20 to unity
X[Y>0]
```

Matrices can be amalgamated using the two functions **rbind**, which combines by rows, and **cbind**, which combines by columns. The following examples illustrate the use of the two functions.

```
X1<-matrix(1:10,ncol=2)
# X1 is 5 x 2
Y1<-matrix(1:20,ncol=4)
# Y1 is 5 x 4
Z1<-cbind(X1,Y1)
Z1
rbind(X1,Y1[ ,1:2] )
```

3. PROBLEMS**2.3** Let

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & -1 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 2 & 0 \\ 1 & 5 \end{pmatrix}$$

a. Find **AB** and **BA**.

b. Find $|\mathbf{AB}|$, and $|\mathbf{A}|$, and $|\mathbf{B}|$ and verify that $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$ holds in this case.

2.4 Use the matrices \mathbf{A} and \mathbf{B} in Problem 2.3:

a. Find $tr(\mathbf{A} + \mathbf{B})$.

b. Find $tr(\mathbf{A})$ and $tr(\mathbf{B})$ and show that $tr(\mathbf{A} + \mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B})$.

2.5 Let

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & -1 & 1 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 3 & -2 \\ 2 & 0 \\ -1 & 1 \end{pmatrix}$$

a. Find \mathbf{AB} and \mathbf{BA} .

b. Compare $tr(\mathbf{AB})$ and $tr(\mathbf{BA})$ and confirm that $tr(\mathbf{AB}) = tr(\mathbf{BA})$ holds here.

2.19 Let

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & -2 \\ -1 & 2 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

a. Find the eigenvalues and associated normalized eigenvectors.

b. Find $tr(\mathbf{A})$ and $|\mathbf{A}|$ and show that $tr(\mathbf{A}) = \sum_{i=1}^3 \lambda_i$ and $|\mathbf{A}| = \prod_{i=1}^3 \lambda_i$.

2.20 Let

$$\mathbf{A} = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix}$$

a. Find the normalized eigenvectors and use them as columns in an orthogonal matrix \mathbf{C} .

b. Show that $\mathbf{C}'\mathbf{A}\mathbf{C} = \mathbf{D}$, where \mathbf{D} is diagonal with the eigenvalues of \mathbf{A} on the diagonal.

c. Show that $\mathbf{A} = \mathbf{C}\mathbf{D}\mathbf{C}'$.

3.10 Use the calcium data in Table 3.4:

a. Calculate \mathbf{S} (sample covariance matrix).

b. Obtain \mathbf{R} (sample correlation matrix).

3.11 Use the calcium data in Table 3.4:

- a. Find the generalized sample variance $|\mathbf{S}|$.
- b. Find the total sample variance $tr(\mathbf{S})$.

3.14 For the variables in Table 3.4, define $z = 3y_1 - y_2 + 2y_3$. Find \bar{z} and s_z^2 .

3.15 For the variables in Table 3.4, define $w = -2y_1 + 3y_2 + y_3$ and define z as in Problem 3.14. Find r_{zw} .

3.17 Define the following linear combinations for the variables in Table 3.4:

$$z_1 = y_1 + y_2 + y_3$$

$$z_2 = 2y_1 - 3y_2 + 2y_3$$

$$z_3 = -y_1 - 2y_2 - 3y_3$$

Find \bar{z} , \mathbf{S}_z and \mathbf{R}_z .

4. SOLUTIONS

PROBLEM 2.3

```
A<-matrix(c(1,2,3,-1),nrow=2,ncol=2)
```

A

```
B<-matrix(c(2,1,0,5),nrow=2,ncol=2)
```

B

a) AB and BA

```
A%*%B
```

```
B%*%A
```

b) $|AB|$, $|A|$, $|B|$

```
det(A%*%B)
```

```
det(A)
```

```
det(B)
```

```
det(A)*det(B)
```

```
## PROBLEM 2.4
```

```
# tr(A+B)
sum(diag(A+B))
```

```
# tr(A)
sum(diag(A))
```

```
# tr(B)
sum(diag(B))
```

```
## PROBLEM 2.5
```

```
A<-matrix(c(1,2,2,-1,3,1),nrow=2,ncol=3)
```

```
A
```

```
B<-matrix(c(3,2,-1,-2,0,1),nrow=3,ncol=2)
```

```
B
```

```
A%%B
```

```
B%%A
```

```
sum(diag(A%%B))
```

```
sum(diag(B%%A))
```

```
## PROBLEM 2.19
```

```
## a) Finding eigenvalues and eigenvectors
```

```
A<-matrix(c(1,-1,0,1,2,1,-2,1,-1),nrow=3,ncol=3)
```

```
A
```

```
# d = vector with eigenvalues
```

```
d<-eigen(A)$val
```

```
# C = matrix with normalized eigenvectors
```

```
C<-eigen(A)$vec
C

## b)

## trace of A
sum(diag(A))
sum(d)

## determinant of A
det(A)
prod(d)

## PROBLEM 2.20

## a)
A<-matrix(c(3,1,1,1,0,2,1,2,0),nrow=3,ncol=3)
A

# eigenvalues
d<-eigen(A)$val
d

# eigenvectors
C<-eigen(A)$vec
C

## b)

t(C)%*%A*%C

## c)
```



```
C%%diag(d)%*%t(C)
```

```
## Problem 3.10
```

```
## First, create a new folder, let us call it sta437  
## Next, make sta437 your working directory  
## Then, save T3_4_CALCIUM.DAT in sta437
```

```
data<-read.table(file="T3_4_CALCIUM.DAT")
```

```
## Only variables y1, y2, and y3
```

```
vars<-data[ , -1]
```

```
vars
```

```
## a) Finding S
```

```
S<-var(vars)
```

```
S
```

```
## b) Finding R
```

```
R<-cor(vars)
```

```
R
```

```
## Problem 3.11
```

```
## a)
```

```
det(S)
```

```
## b)
```

```
sum(diag(S))
```

```
## Problem 3.14
```

```
## mean vector for y1, y2, and y3
```

```
y.bar<-apply(vars,2,mean)
```

```
y.bar<-matrix(y.bar,ncol=1)
```

```
## linear combo z
```

```
b<-matrix(c(3,-1,2),ncol=1)
```

```
## mean for linear combo
```

```
t(b)%*%y.bar
```

```
## variance for linear combo z
```

```
var.z<-t(b)%*%S%*%b
```

```
var.z
```

```
## Problem 3.15
```

```
## linear combo w
```

```
a<-matrix(c(-2,3,1),ncol=1)
```

```
## covariance between z and w
```

```
cov.z.w<-t(b)%*%S%*%a
```

```
cov.z.w
```

```
## variance for linear combo w
```

```
## variance for linear combo z
```

```
var.w<-t(a)%*%S%*%a
```

```
var.w
```

```
## correlation between z and w
cor.z.w<-cov.z.w/sqrt(var.z*var.w)

cor.z.w

## PROBLEM 3.17

## B matrix of linear transformation
B<-matrix(c(1,2,-1,1,-3,-2,1,2,-3),nrow=3,ncol=3)

## mean of linear transformation
B%*%y.bar

## covariance matrix of linear transformation
S.B<-B%*%S%*%t(B)

S.B

## correlation matrix of linear transformation
D<-diag(1/sqrt(diag(S.B)))

D%*%S.B%*%D
```