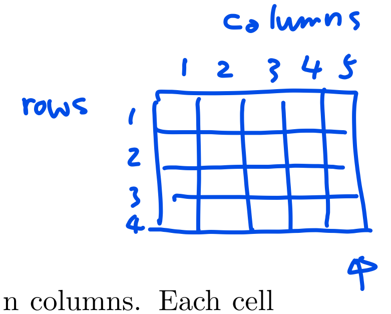
Single values , vectors

Chapter 2.5: Matrices and Data Frames in R

Learning Outcomes: We are going to go over the following:

- Defining a matrix in R.
- Naming the rows and columns of a matrix in R.
- Commands to sum the columns and rows of a matrix.
- How to add another row or column to a matrix.
- How to select specific elements from a matrix.
- The difference between the as.matrix() and the matrix() command.
- What is the difference between a matrix and a data frame?



4x5 matrix.

Definition

An m×n **matrix** is a 2-dimensional rectangular table with m rows and n columns. Each cell in the matrix is identified by its row and column indices or names. In a **matrix** all the elements must be the **same type of data**. A matrix in R is like a mathematical matrix, containing all the same type of values (usually numbers).

Defining a Matrix in R: In order to create a matrix in R, you use the command matrix(). The input for this function is a vector and then the number of rows or columns you want. For example:

e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$ or $\begin{bmatrix} \text{"Tom"} & \text{"Mary"} \\ \text{"0"} & \text{"1"} \end{bmatrix}$

3x2 matrix with number for 1 to 6

```
x <- matrix(c(1:6), nrow = 3)
or  x <- matrix(c(1:6), ncol = 2)
```

$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & 4 \\ 2 & 2 & 5 \\ 3 & 3 & 6 \end{array}$$

R fills the matrix by columns (default)

You might want to specify how the matrix should be filled (that is, should it be filled by row or by column). There is an extra input that you can type in the matrix() function that accomplishes this:

```
x <- matrix(c(1:6), nrow = 3, byrow = TRUE)
x <- matrix(c(1:6), byrow = TRUE, nrow = 3)
```

Try:

```
x <- matrix(c(1:7), nrow = 3)
```

← warning but fill in missing values using

vectors
(reuse)

Often times, the matrix is organizing data in a useful way. Perhaps the columns represent the values of certain variables and the rows represent the individuals in the sample. It is nice to associate meaning to the values in the matrix by naming the rows and the columns. For example:

Suppose we have a matrix with 4 rows and 2 columns and the rows represent the individuals A, B, C, D and the first column represents the variable Weight and the second column represents the variable Age. Suppose the pairs of data are as follows: (80,12), (40,3), (20,0.8), (25,1)

4x2 matrix

weight age
of A

(a) Create a matrix M with these values.

```
M= matrix(c(80, 12, 40,3, 20, 0.8, 25, 1), byrow= TRUE,
          nrow= 4 )
```

(b) Create a vector called *individuals* which contains the individuals A,B,C,D and create a vector called *variables* which contains the two variables Weight and Age.

```
individuals = c("A", "B", "C", "D")
variables   = c("weight", "Age")
```

(c) Use the R commands colnames() and rownames() to set the names of the matrix M to the appropriate vectors from part (b).

```
colnames(M) = variables
rownames(M) = individuals
```

Adding up the rows and columns of a matrix: There are commands rowSums() and colSums() which add up the rows and columns of a matrix.

For example, the commands:

rowSums (M) ← total of the rows

↑ capital letter

colSums (M) ← total of the columns

Adding another row or another column to an existing Matrix: Sometimes, you may wish to add more information to your matrix. Rather than re-create the matrix, we can add another row using the command rbind() and add another column using the command cbind().

For example, suppose that we wished to add another individual E with weight 30 and age 1.5 to our matrix. This amounts to adding another row. We could do this by typing in:

newM = rbind (M , c(30, 1.5))

Selecting an element or a row or a column from a Matrix:

Matrices are indexed by row position and column position. Suppose we have a matrix that is called M, then:

R index starts at 1, not zero.

- M[1,2] = 12

  ↑ row 1
  column 2

- M[4,3] gives error

- M[,2] or M[,2] = 12
  3
  0.8
  1
  2nd column

- M[3,] third row of matrix M

- M[1:3,] first 3 rows of M

- M[,1:4] first 4 columns or columns 1 to 4

Finding and locating max and min in a matrix or vector:

R commands: max(M)

min (M)

Note:

Difference between the matrix() and as.matrix() commands:

- The command matrix() is to create a matrix from a vector.

- The command as.matrix() is to convert a compatible object to a matrix.

Data Frame

Recall: In a **matrix** all the elements must be the **same type of data**.

A data frame is similar to a matrix, The structure of a data frame is more flexible. The columns of a data frame are allowed to contain **different types of data**,

Many data sets that we download or that are available to us in R are data frames.

In R, if you type "mtcars" without the quotation marks, you will display the built-in dataset mtcars. If you type ?mtcars, R will display the description of the data frame.

identifier not a variable

cat.

```
                  mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

each row is one observation of an individual

each column contains values of a variable.

What are the individuals? car name

What are the variables? those 11 variables

Which variable(s) is (are) categorical? am , vs (see descriptions)

Which variable(s) is (are) numerical? the rest

Working with individual variable(s) within a data frame

To display the variable **hp**, you can type:

mtcars $ hp

Matrix indexing also works with data frames:

mtcars [6, 2] → 6 (valiant , cyl)

6th individual 2nd variable

or

mtcars $ cyl [6]

There are many ways to create a data frame

Example: Suppose we want to create a data frame below:

```
> sample.df
   Name Age  Vote
1  Juan  22  TRUE
2 Maria  15 FALSE
3  Mark  19  TRUE
```

Example: You may want to convert a data frame to a matrix (or vice versa). Here is an example. Consider the data frame PlantGrowth that is one of the built-in data sets in R.

(a) Print out the first 2 rows of the data frame using the head() command.

(b) Use the as.matrix command to create a matrix called $weightMatrix$ containing only the first column of the data frame.

(c) Use the class() function to confirm that $weightMatrix$ is in fact a matrix, and that $PlantGrowth$ is a data frame.