Data Structures

Michael Malick

Updated: 15 May 2013

Data Structures

A *data structure* in R is a means of storing information

- Vector
- Matrix
- Dataframe
- List
- Array (not covered)

Vectors

Vectors

A *vector* is a one dimensional, ordered collection of numbers, strings, etc.

- 1, 2, 3, 4, 5
- A, B, C, D, E, F
- TRUE, FALSE, FALSE, TRUE
- blue, red, purple, orange

Creating Vectors

```
x <- c(1, 2, 3, 4, 5)
y <- c("A", "B", "C", "D")
u <- c(TRUE, FALSE, FALSE, TRUE)
```

```
z <- seq(from = 1, to = 5, by = 1)
z <- 1:5

r <- rep(y, times = 10)
```

Vector Arithmetic

R does computations on whole vectors

```
x <- rep(1, times = 3)
y <- rep(2, times = 3)
x + y

z <- 1:3
z*2
z/3
```

Subsetting Vectors

Subsetting is when you extract a certain portion (subset) of the data structure

```
 1  2  3  4  5  6  7  8  9 10
```

```
x <- 1:10
x[3]
x[5:8]
x[x > 5]
x[x != 2]
```

Classes

Every object in R has a class

```
x <- c(1, 2, 3, 4, 5)
y <- c("A", "B", "C", "D")
u <- c(TRUE, FALSE, FALSE, TRUE)

class(x)
class(y)
class(u)
```

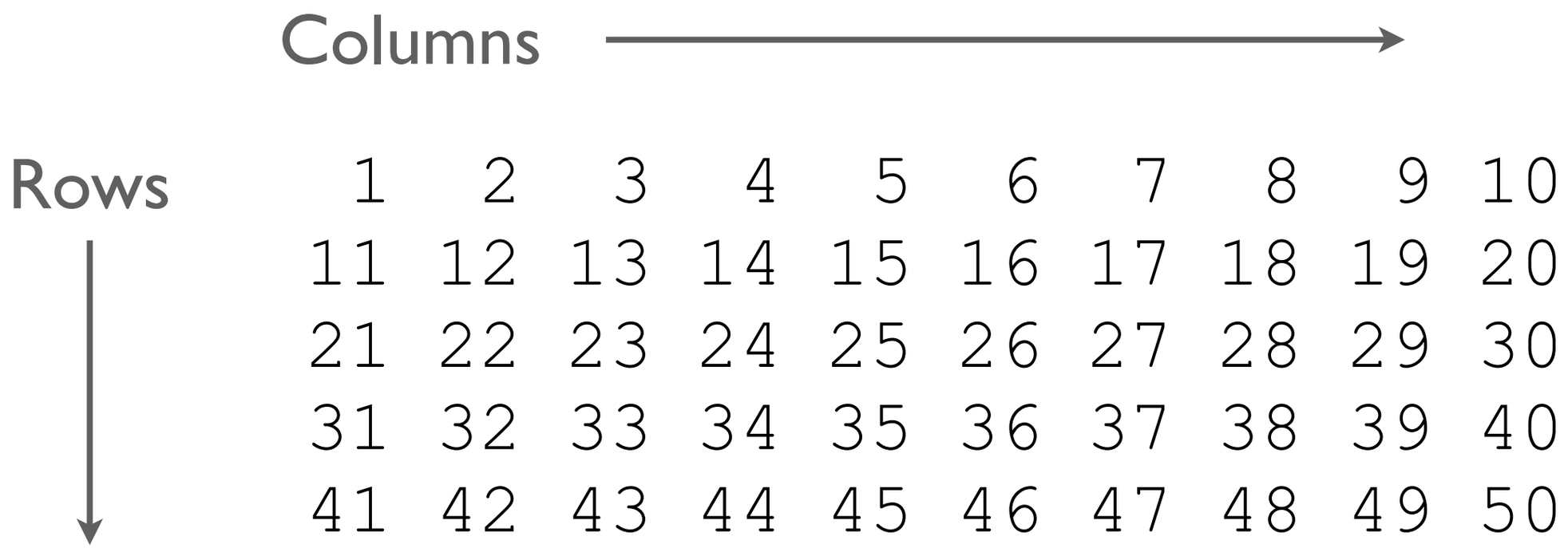
You Try...

1. Create a sequence of numbers from 1 to 10 by 0.1 and assign it to the variable z
2. What is the mean of z?
    - Hint: use the `mean()` function
3. What is the 87th number in the sequence?

Matrices

Matrix

A *matrix* is a two dimensional data structure composed of rows and columns



Creating a Matrix

```
x <- rnorm(25)

mat <- matrix(data = x)

mat <- matrix(data = x, nrow = 5,
    ncol = 5)
```

```
mat <- matrix(data = x, nrow = 5,
    ncol = 5, dimnames = list(c("A", "B",
    "C", "D", "E"), c("1", "2", "3", "4",
    "5")))

mat.na <- matrix(NA, 10, 10)
```

Subsetting Matrices

SomeMatrix[row, column]

Matrix z

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
z[1, ]
```

```
z[ ,2]
```

```
z[4,4]
```

Subsetting Matrices Cont'd

```
x <- 1:25
mat <- matrix(data = x, nrow = 5, ncol = 5)

mat
mat[1, ]      # first row
mat[ ,2]      # second column
mat[4, 3]     # 4th row and 3rd column
mat[mat > 5] # all values greater than 5
```

Matrix Arithmetic

Matrices can be added, subtracted, multiplied, etc. just like vectors

```
mat1 <- matrix(1:25, 5, 5)
mat2 <- matrix(1:25, 5, 5)
```

```
mat1 + mat2
mat1 * mat2
mat1^2

summary(mat1)
```

You Try...

1. Create a sequence of numbers from 1 to 100 by 1 and assign it the variable v
2. Using v create a matrix with 10 rows and 10 columns and assign it to the variable m
3. What is the number in the 4th row and 7th column of the matrix?

Dataframes

Dataframe

A *dataframe* is similar to a matrix, but each column can be of a different type of data (e.g., character, numeric, logical).

Columns →

Rows ↓

Species	Sex	Count
A	M	10
B	F	19
C	M	22
D	M	16
E	F	12

Creating a Dataframe

```
number <- 1:5
weight <- c(1.2, 2.0, 1.1, 1.9, 2.9)
length <- c(25, 29, 26, 24, 30)
sex    <- c("M", "M", "F", "M", "F")

dat <- data.frame(number, weight, length, sex)
```

Subsetting Dataframes

SomeDataframe[row, column]

SomeDataframe$variablename

Dataframe z

Species	Sex	Count
A	M	10
B	F	19
C	M	22
D	M	16
E	F	12

```
z[1, ]
```

```
z[ ,2]
```

```
z$Sex
```

Subsetting Dataframes Cont'd

```
iris


iris[1, ]
iris[ ,1:2]
iris$Species

iris[iris$Species == "virginica", ]

iris[iris$Sepal.Length < 5, ]
subset(iris, Sepal.Length < 5)

iris[iris$Species == "virginica" &
     iris$Sepal.Length < 6, ]
```

Dataframes: Factors

By default, variables in a dataframe that consist of strings are stored as factors

```
head(iris)

class(iris$Species)
iris$Species
levels(iris$Species)

summary(iris)
```

You Try...

Using the iris dataset:

1. Subset the 17th row, what is the Petal.Width for this record?
2. What is the mean of Petal.Length?
3. What is the median of Petal.Length for the setosa Species?

Lists

Lists

A *list* is an arbitrary collection of R objects

- Different objects within a list are known as components
- Lists can have components made up of vectors, matrices, arrays, dataframes, and other lists

Creating a List

```
mat    <- matrix(1:25, 5, 5)
number <- 1:5
sex    <- c("M", "M", "F", "M", "F")
```

```
lst <- list(mat, number, sex)
```

Subsetting Lists

SomeList[[component]]

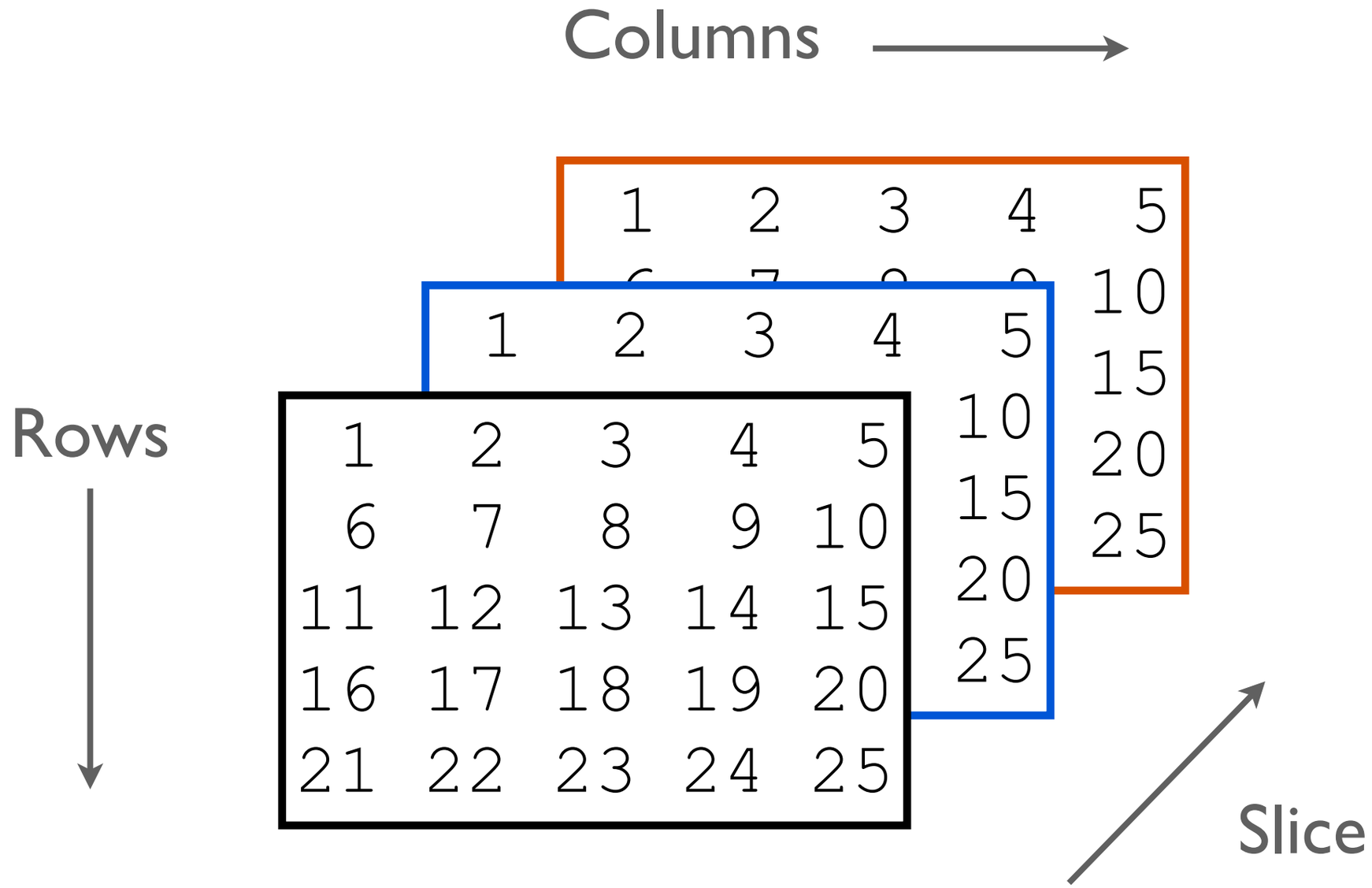
SomeDataframe$component

```
mat    <- matrix(1:25, 5, 5)
number <- 1:5
sex    <- c("M", "M", "F", "M", "F")

lst <- list(mat = mat, number = number,
    sex = sex)

lst[[2]]
lst$mat
```

Arrays

Array

An *array* is a three dimensional structure that consists of multiple matrices



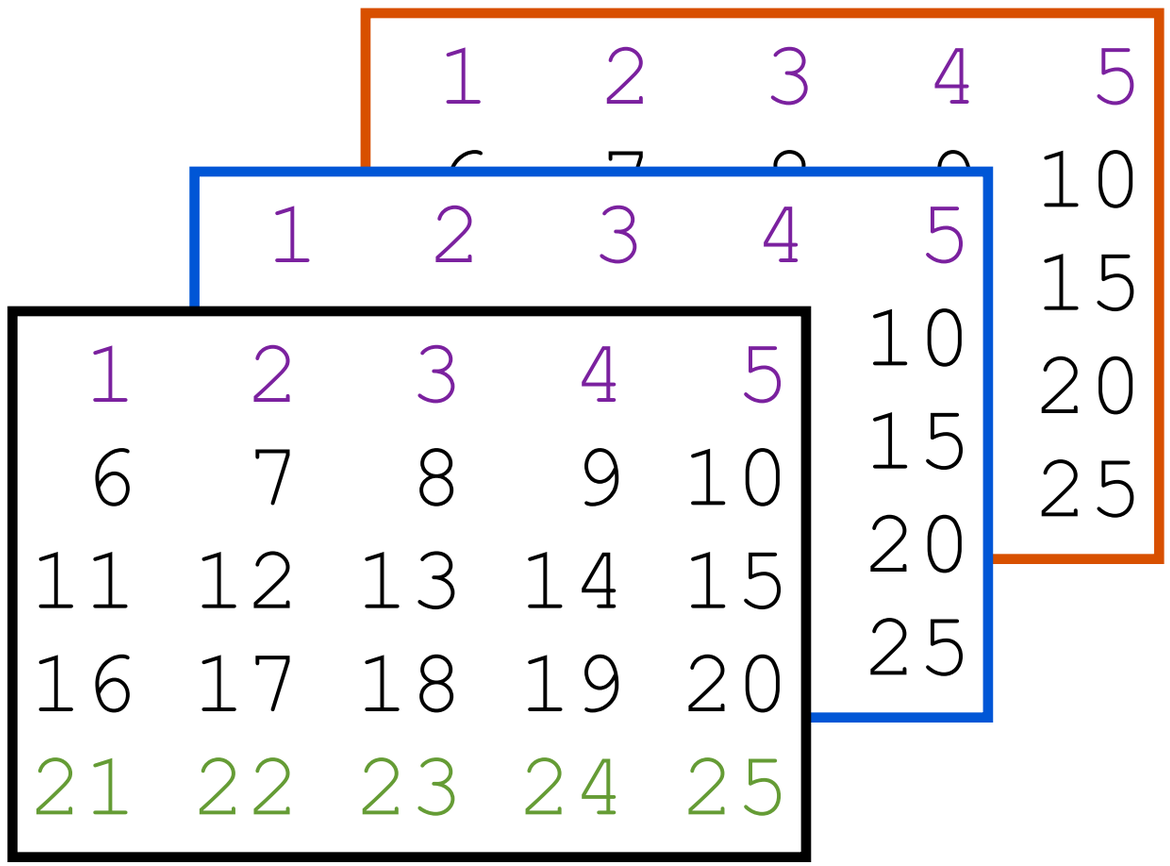
Creating an Array

```
x <- 1:10
y <- 21:30
z <- 41:50

arr <- array(data = c(x, y, z), dim =
    c(5, 2, 3))

arr.na <- array(NA, dim = c(5, 5, 3))
```

Subsetting Arrays

SomeArray[row, column, slice]

Array z




```
z[1, , ]

z[ , ,2]

z[5, ,1]
```

Subsetting Arrays Cont'd

```
x <- rep(1:25, 3)
arr <- array(x, dim = c(5, 5, 3))
```

```
arr[1, , ] # first row
arr[ , ,2] # second slice
arr[5, ,1] # 5th row in 1st slice
```

Array Arithmetic

Arrays can be added, subtracted, multiplied, etc. just like matrices

```
x <- rep(1:25, 3)
arr1 <- array(x, dim = c(5, 5, 3))
arr2 <- array(x, dim = c(5, 5, 3))

arr1 + arr2
arr1 * arr2
arr^2
```