R development master class

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

June 2011

1. Important info
2. Course outline
3. Revision

HELLO

my name is

Hadley

Charlotte Wickham
Assistant Professor
Department of Statistics
Oregon State University

Barret Schloerke
Engineer
Metamarkets

http://had.co.nz/
courses/11-masterclass

Course outline

Day one

- First class functions
- Controlling evaluation
- Object oriented programming
- Best practices

Day two

- Introduction to packages
- Documentation
- Testing
- Releasing your package

Tips

Tips

Ask questions!

Practice consciously: make a prediction, then test it, then reflect.

Keep an electronic copy of the slides open so you can copy and paste code.

Revision

Your turn

What are the four basic types of atomic vectors?

Brainstorm with your neighbour for 1 minute.

character

numeric
integer

logical

```
as.character(c(T, F))
as.character(seq_len(5))
as.logical(c(0, 1, 100))
as.logical(c("T", "F", "a"))
as.numeric(c("A", "100"))
as.numeric(c(T, F))
```

When vectors of different types occur in an expression, they will be automatically coerced to the same type: character > numeric > logical

`mode()`

`names()` Optional, but useful

`length()` A scalar is a vector of length 1

Technically, these are all **atomic** vectors

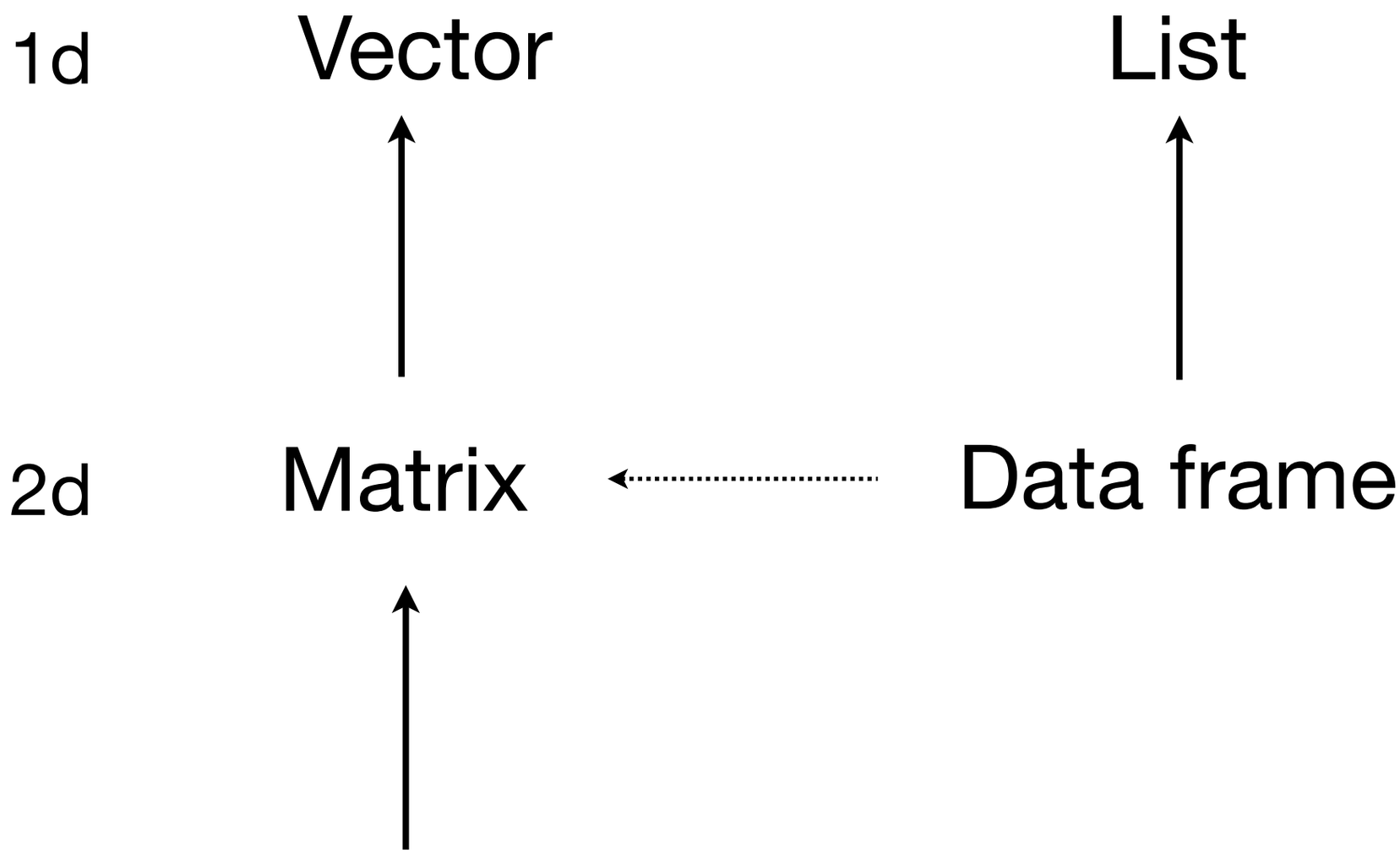
Your turn

How is a list different from an atomic vector?

How is a data frame different from a matrix?

How do you examine the structure of an object?

Brainstorm with your neighbour for 1 minute.


1d
Vector
List
2d
Matrix
Data frame
nd
Array
Same types
Different types

str()

Your turn

What are the five types of object that you can subset with?

What's the difference between [, [[and $?

Brainstorm with your neighbour for 2 minutes.

blank	include all
integer	**+ve**: include **-ve**: exclude
logical	keep TRUEs
character	lookup by name

Vectors	`x[1:4]`	–
Matrices Arrays	`x[1:4, ]` `x[, 2:3, ]`	`x[1:4, , drop = F]`
Lists	`x[[1]]` `x$name`	`x[1]`

Your turn

What are the three ways arguments supplied to a function are matched to the formal arguments?

What does ... do ?

Argument matching

full name

partial name

position

...

captures all other arguments
can pass on to other functions