# http://courses.had.co.nz R development masterclass

### **Hadley Wickham**

Assistant Professor / Dobelman Family Junior Chair Department of Statistics / Rice University

**June 2012** 



1. Introduction

2. Course outline

3. Revision

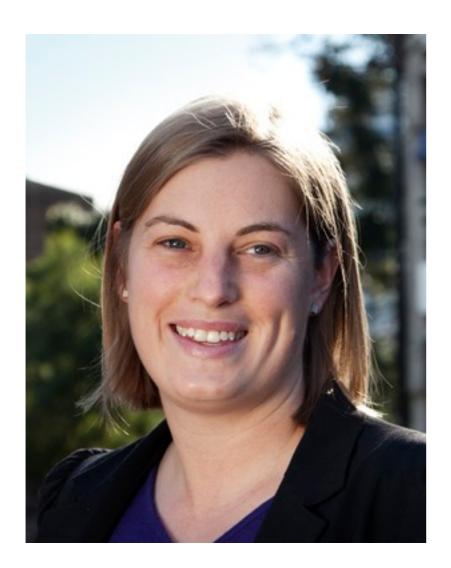
## Introductions

## HELLO my name is

# Hadley



Winston Chang
Research Associate
Department of Statistics
Rice University



Charlotte Wickham
Assistant Professor
Department of Statistics
Oregon State University

Who are you and what are you using R for?

# Course outline

## Day one

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

## Day two

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

## 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

What are the four common types of atomic vectors? (Bonus points for the two uncommon types)

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

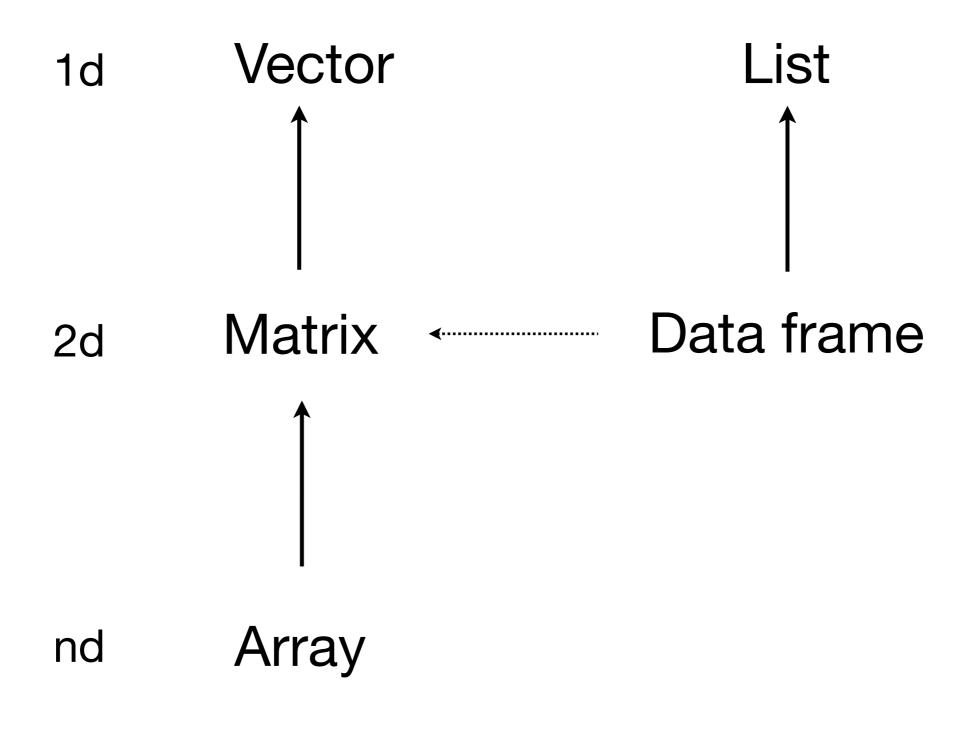
Technically, these are all atomic vectors

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.



Same types

Different types

# str()

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

What's the difference between [, [[ and \$? When might you use drop = F?

Brainstorm with your neighbour for 2 minutes.

blank include all

integer +ve: include

-ve: exclude

logical keep TRUEs

character lookup by name

	Simplifying	Preserving
Vectors	x[[1]]	x[1:4]
Matrices/ Data frame	x[1:4,]	x[1:4, , drop = F]
Lists	x[[1]] x\$name	x[1]

If list x is a train carrying objects, then x[[5]] is the object in car 5; x[4:6] is a train of cars 4-6.

What are the three ways arguments supplied to a function are matched to the formal arguments? In what situations should you use each?

What does ... do?

#### **Argument matching**

full name partial name position

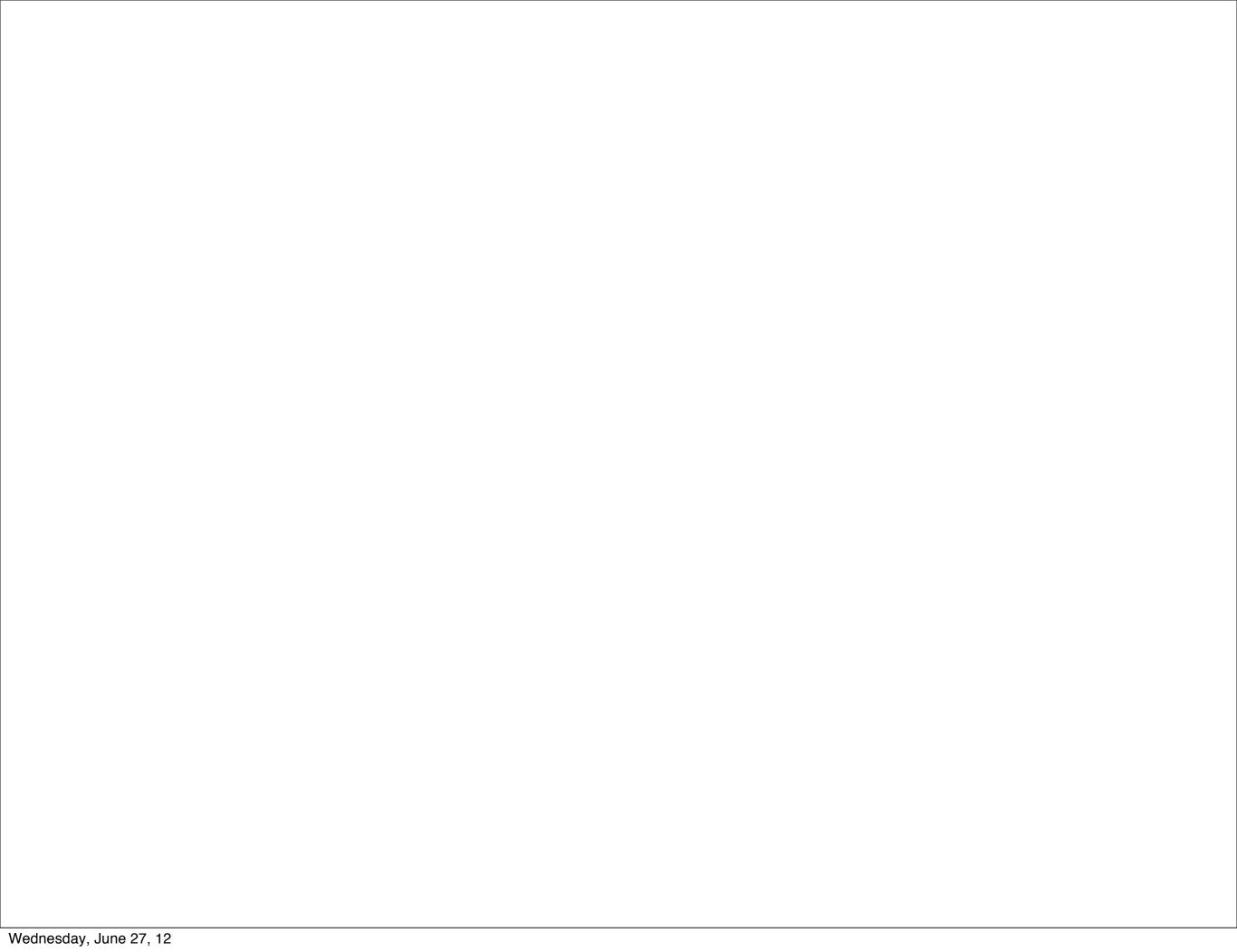
captures all other arguments can pass on to other functions

```
x <- 5
f <- function() {</pre>
  y <- 10
  c(x = x, y = y)
f()
g <- function() {</pre>
  x <- 20
  y <- 10
  c(x = x, y = y)
}
g()
h <- function() {</pre>
  y <- 10
  i <- function() {</pre>
    z <- 20
    c(x = x, y = y, z = z)
  i()
h()
```

## What do these functions return?

```
j <- function() {
  if (!exists("a")) {
    a <- 5
  } else {
    a <- a + 1
  }
  print(a)
}</pre>
```

What does this function return the first time you run it? The second time?



This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/us/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.