Stat405

Advanced topics

Hadley Wickham

1. Math on the computer
2. Debugging
3. Professional development
4. Feedback

Posters

Make sure your poster is ready by 4pm.
Print on Wednesday!

Dress up.

There will be food and beverages. Partake appropriately.

Make sure you've read the poster rubric.

Math on the computer

Your turn

Perform the following calculations in R.
Are the answers what you expect?

```
seq(0.1, 0.9, by = 0.1) - 1:9 / 10
sqrt(2)^2 - 2
```

What is the property of these numbers that might cause the problem?

```
# Each number must be stored in a finite amount of
# space => each number can only have a finite number
# of digits => floating point math does not work
# like normal math

(1e-16 + 1) == 1
(1e-16 + 1) * 10 == 1e-16 * 10 + 1 * 10
```

```
1e10 + 1.9 - 1e10
1e11 + 1.9 - 1e11
1e12 + 1.9 - 1e12
1e13 + 1.9 - 1e13
1e14 + 1.9 - 1e14
1e15 + 1.9 - 1e15
1e16 + 1.9 - 1e16
```

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

$$a + (b + c) = (a + b) + c$$

$$a + b - b = a$$

```
# By default R only shows 7 significant digits
# If the trailing digits are zero, the number will
be rounded
(1 / 237)
(1 / 237) * 237
(1 / 237) * 237 - 1
```

```
seq(0.1, 0.9, by = 0.1)
seq(0.1, 0.9, by = 0.1) - 1:9 / 10

# Tricky to get to print exactly:
formatC((1 / 237) * 237, digits = 20)
formatC(seq(0.1, 0.9, by = 0.1), digits = 20)
```

```
# When working with floating point numbers (numeric)
# (but not integers, this is the one place where the
# difference is important) never test for equality
# with ==

a <- seq(0.1, 0.9, by = 0.1)
b <- 1:9 / 10


all(a == b)
all.equal(a, b)
all(abs(a - b) < 1e-6)

# Similarly, need to be careful with < and > etc
```

```
# Places where this matters:
#
# * sums
# * calculating the standard deviation
# * inverting a matrix (condition)
#   * linear models!
# * maximum likelihood estimation
```

Debugging

Pop quiz

Which is the best way to solve a problem?

1) Write a giant function that tries to do everything, and then when it doesn't work you have no idea where the problem is.

2) Write small functions that each do a single task and can be tested easily. If there is a problem, you can localize it to a few lines of code

Tools of

last resort

If you're using them all the time, something is wrong with your basic approach

```
traceback()
browser()
recover()
```

```
options(error)
```

```
f <- function(x) {
   a <- 1
   i(x)
   g(h(x))
   i(x)
}
g <- function(x) {
  b <- 2
  x
}
h <- function(x) {
  c <- 3
  i(x)
  i(x)
}
i <- function(x) {
  d <- 4
  if (sample(10, 1) == 1) stop("This is an error!")
}

f()
traceback()  # This is called the call stack
```

Traceback

Shows the call stack: the path of functions R called between starting your function and encountering an error.

```
f(3) # rerun until you get an error

traceback()
```

If your previous function worked successfully, `traceback` will show the most recent error

Browser

- Pauses function and creates an interactive prompt in a function's environment
- Four special commands (no brackets):
  - `n` = next line
  - `c` = continue (or just press return)
  - `where` = where am I in the call stack?
  - `Q` = quit
- Can also use any regular R function (e.g. `ls()`)

```
j <- function(x, y = 10) {
  k(x, y)
}

k <- function(x, y) {
  z <- 3
  browser()
  x + y
}

j(10)
```

Your turn

Familiarise yourself with `browser()`

Try using `ls()` while you are browsing. What do you see?

Try modifying the values inside the function. What happens to the result?

Think about how you could have used it in the project

Browser

Most useful when you know where the problem is

`debug(f)` automatically adds a browser statement to the start of `f`, `undebug(f)` removes it.

If the error occurs only under some conditions you might want to put `browser()` inside an `if` statement

Recover

- Works like `browser()`, but lets you jump in anywhere in the call stack
- Most useful in conjunction with `options(error = recover)`

```
# Can change the default behaviour when an error
# occurs
options(error = recover)
f()

# Set to NULL to return to the default
# (i.e. do nothing)
options(error = NULL)

# Another useful option: turn warnings into errors
options(warn = 2)
# and turn them back
options(warn = 0)
```

Your turn

Use `options(error = recover)` and explore the call stack of f.

Use `ls()` to explore what variables are defined in each environment

```
# Your turn
# Use the tools you have just learned about to debug
# this function and create a version that works

larger <- function(x, y) {
  y.is.bigger <- y > x
  x[y.is.bigger] <- y[y.is.bigger]
  x
}
larger(c(1, 5, 10), c(2, 4, 11))
larger(c(1, 5, 10), 6)
```

```
larger <- function(x, y) {
  if (length(x) != length(y))
    stop("x and y don't have same length")

  y.is.bigger <- y > x
  x[y.is.bigger] <- y[y.is.bigger]
  x
}
```

Professional Development

Professional development

The aspects of being a statistician, apart from knowing statistics. Principally communication: written, spoken, visual and electronic.

Take every opportunity you can to practice these skills.

A small investment now can have a big pay off in the future.

Learn your tools

- Touch typing
- Text editor
- Command line
- Caffeine
- R

R: Mailing list

Sign up to R-help: https://stat.ethz.ch/mailman/listinfo/r-help

Make sure to set up filters

Skim interesting subjects and read them

Don’t be afraid to post
(use a pseudonym if necessary)

R: Books

R in a nutshell, *Joseph Adler.*
http://amzn.com/059680170X

Data manipulation with R, *Phil Spector.*
http://amzn.com/0387747303

Software for Data Analysis: Programming with R, *John Chambers*.
http://amzn.com/0387759352

R: Books

Regression Modeling Strategies, *Frank Harrell*. http://amzn.com/0387952322

Mixed-Effects Models in S and S-PLUS, *Jose Pinheiro and Douglas Bates*. http://amzn.com/1441903178 and http://lme4.r-forge.r-project.org/book/

Data Analysis Using Regression and Multilevel/Hierarchical Models, *Andrew Gelman and Jennifer Hill*. http://amzn.com/052168689X

R: Journals

The R Journal,
http://journal.r-project.org/

The Journal of Statistical Software,
http://www.jstatsoft.org/

Statistical computing and graphics newsletter, *http://stat-computing.org/newsletter/*

Writing

Every job, academic and industry, requires you to communicate your work.

Best book: “Style: Lessons in Clarity and Grace”. *http://amzn.com/0205747469*

Take every opportunity you can to practice. Develop a regular habit

Speaking

Seize every opportunity to practice.

Tracy Volz (tmvolz@rice.edu) is a fantastic resource – if you had to pay for her, you wouldn't be able to afford it.

Team work

Will be a part of your future job.
Incredibly, incredibly important skill.

Useful tools: cc / reply all, google docs, dropbox, http://www.stypi.com, https://trello.com/,

More reading

http://www.huffingtonpost.com/linda-stone (email articles in 2009)

http://www.alistapart.com/articles/habit-fields/

Feedback

Your thoughts

In the last few minutes of class, please give us some feedback about the class.

I've set up an anonymous survey at:

http://hadley.wufoo.com/forms/stat405-feedback/