Data Science: Lets All Be Ninjas

Matthew J. Denny

mzd5530@psu.edu – www.mjdenny.com – @MatthewJDenny

www.mjdenny.com/ICPSR_Data_Science_2015.html

July 27, 2015
Data Science

1. Means lots of things...

2. Data: Collection, curation, exploration, prediction.

3. Rewards programming skills.

4. Yay for outside options!
Skills

1. Data management.
   ▶ Funky data, many sources, replicability.

2. Working at large scale and high speed.
   ▶ Approximate methods, HPC, resource management.

3. Description and interpretation.
   ▶ Visualization, summarization, causal inference.

4. Collaboration and open source.
   ▶ Fluency, Extensibility, distributability.
What We Will Cover

1. Scientific Programming in R.


5. Big Data – Memory management.

6. Hardware.

7. R and C++.

8. Web Scraping.

Overview

1. General Programming in R.
2. Super Friends!
3. Super Tools!
1. General Programming in R.
Motivation – The Gardener

- How many plants to water?
- Which plants to water?
Overview

- for( ) and while( ) loops.
- if( ) statements.
- Nested loops
- Lists
Preliminaries

# create a vector
my_vector <- c(1:10)
print(my_vector)

# get the length of the vector
length(my_vector)

# comparison operators
5 < 6
5 > 6
5 == 5
5 != 6
5 <= 5
The for( ) Loop
The for( ) Loop

- Do something N times.

```r
my_vector <- c(20:30)
for(i in 1:length(my_vector)){
    my_vector[i] <- sqrt(my_vector[i])
}

my_vector

[6] 5.000000 5.099020 5.196152 5.291503 5.385165
```
The `while()` Loop
The **while( )** Loop

- Do something until a condition is met.
- Useful if you do not know the number of iterations ahead of time.

```r
my_vector <- c(20:30)
counter <- 1
while(counter <= length(my_vector)){
  my_vector[counter] <- sqrt(my_vector[counter])
  counter <- counter + 1
}

my_vector
```
The if( ) Statement
The if() Statement

- Do something if some condition is met.
- Can be built into a loop.

```r
my_vector <- c(20:30)

for(i in 1:length(my_vector)){
  if(my_vector[i] == 25){
    print("The square root of 25 is 5!")
  }
}
```
The else Statement

- Do something if some condition is not met.

```r
my_vector <- c(20:30)

for(i in 1:length(my_vector)){
  if(my_vector[i] == 25){
    print("I am 25!")
  }else{
    print("I am not 25!")
  }
}
```
Traversing A Matrix

```r
> matrix(1:25,5,5)

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

- Can loop over entries in higher dimensional data structures.

```r
my_matrix <- matrix(1:100, ncol=10, nrow=10)

for(i in 1:length(my_matrix[,1])){
    for(j in 1:length(my_matrix[1,])){
        if(my_matrix[i,j] %% 2 == 0){
            my_matrix[i,j] <- 0
        }
    }
}

my_matrix
```
Lists

- Flexible, can store any kind of data including another list.
- Good for keeping results together.

```r
# Create an empty list
my_list <- vector("list", length = 10)

# Create a list from objects
my_list <- list(10, "dog", c(1:10))

# Add a sublist to a list
my_list <- append(my_list, list(list(27, 14, "cat")))
```
> my_list
[[1]]
[1] 10

[[2]]
[1] "dog"

[[3]]
[1] 1 2 3 4 5 6 7 8 9 10

[[4]]
[[4]][[1]]
[1] 27

[[4]][[2]]
[1] 14

[[4]][[3]]
[1] "cat"
2. Super Friends!
Overview

- **bash** is a command line terminal.
  - Linux and OS X (Cygwin for Windows).
- **VPN**: access campus network.
- **ssh/PuTTY (Windows)** for remote access.
- **ftp**: for file transfer.
Bash Commands

- **cd**: change the current directory.

- **ls**: prints contents of current directory.

- **edit/vi/emacs**: opens a text editor.
bash commands – continued

- `python` opens python console.
- `R` opens standard R console.
- `cd ..` moves us back up a level in the directory structure.

```
0587377979:Example matthewjdenny$ cd ..
0587377979:RA_Projects matthewjdenny$ cd ..
0587377979:Desktop matthewjdenny$
```
Using VPN

- Provided by university/organization.
- For login to local campus resources.
- Routes all traffic through campus servers.
Using SSH

- Must have account on remote machine.
  
  - `ssh` username@ipaddress
    
    - static ip: 128.114.64.8
    
    - dynamic: somedomain.dyndns.com
  
  - prompt to enter password
FTP Using FileZilla

[Image of FileZilla interface showing local and remote sites, file list with directories and files, and transfer status]
Putting It All Together

1. Connect to campus network using **VPN**

2. **ssh** into remote computing resource

3. Transfer files to/from machine using **ftp**

4. Navigate directories using **bash** and run analysis in **R** or **Python**.
3. Super Tools!
Overview

- Version Control
  - Git or Subversion.
  - Make an account on Github.
- RStudio
  - Lots of setup options.
  - Version control integration.
Version Control

- System for tracking changes to documents.
- Resolving conflicts.
- Reverting changes.
- Sharing your work.
- Highly motivating!
Git/Github

- Start by going to: github.com – make an account.
- Then windows.github.com – or – mac.github.com.
- Check out a tutorial.
- RStudio also has Git integration.
Git Tracks Changes
Motivation

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year
- **726 total**

Longest streak
- **16 days**
  - May 21 – June 5

Current streak
- **3 days**
  - July 22 – July 24

36 contributions on May 26, 2015
RStudio

- Make sure you have the latest version of R.
- Then go to: rstudio.com – free academic download.
- Check out a tutorial.
- Spend time getting to know the interface.
Hands-On Tutorial

www.mjdenny.com/Data_Science_Tools.html