

CS636 Data Analytics with R Programming



Instructor
David Li

Course Logistics

- Basic information
- Requirements
- Goal

CS636 Data Analytics with R Programming

- ❑ **Class Schedule:** Saturday 9:00 am - 11:50 am, Fenster Hall 160
- ❑ **Instructor:** David Li, email: dli@njit.edu, tel: 631-800-3381

- ❑ **TA:** Shih-Chuan Weng, email: sw464@njit.edu, tel: 973-718-1822
- ❑ **Office Hours:** Friday 11:00 am – 12:00 am, GITC 4th floor. Please schedule with Shih-Chuan so that he can reserve a seat.

- ❑ **Textbooks**
 - R Programming for Data Science, by Roger D. Peng
 - Using R for Introductory Statistics, by John Verzani, 2014, ISBN 1466590734
 - Advanced R, by Hadley Wickham, ISBN 9781466586963

- ❑ **Website**
 - <http://moodle.njit.edu/>

Requirements

- ▣ Homework & computing lab exercise (5%)
 - ▣ Quiz (20%)
 - ▣ Term Project (10%)
 - ▣ Midterm (25%)
 - ▣ Final (40%)
-
- ▣ You should sign the attendance sheet at the end of each class. Extra bonus based on attendance will be determined.

Homework (2 %)

□ Homework assignments

- Try to do it independently, discussions allowed, but **copying is forbidden**.

□ Homework Grading Policy

- Your homework: may have several homework assignments, but pick only one (the worst one) to grade. Namely, if you miss one assignment, you get 0.

□ Late homework policy

- 25% penalization per late day;
- Not accepted more than 3 days late

Lab exercise (3 %)

- Have a lab session every week
- Lab exercises
 - Focus on R computing exercises
 - 3 students a group. Please find your group mates as quick as possible.
 - Graded as a group: team work is important!
 - 1~3 groups may be selected for test by the end of lab session.

Two Term Projects (10%)

- ❑ Project 1
 - Scrape data from web
- ❑ Project 2
 - Kaggle competition
<https://www.kaggle.com/competitions>
- ❑ Submit a report to summarize what you have done and results you obtained
- ❑ 1 ~ 5 students a group. It can be same as lab group.
- ❑ More details to be announced soon

Quiz (20%)

- ▣ R programming
- ▣ 5 Quizzes
- ▣ Every other week

Two Exams (65%)

- One midterm and one Final (25% + 40%)
 - In-class; closed book; a cheat sheet is allowed
 - Final is cumulative

Some tips

- ❑ Computer/smartphone is not allowed in quiz/exam
- ❑ You should memorize the basic syntax and the usage of functions
- ❑ Prior to quiz/exam, restudy the slides and Jupyter sample code
- ❑ If I discover cheating, I will report the incident to the Dean of Student's office
Re: Academic Integrity. (TAs report the incident to the course instructor)

Goal

- ❑ Gain programming proficiency of R
- ❑ Familiarize you with the commonly used analytical techniques in Data Science
- ❑ Develop the way of data science thinking
 - Learn how to preprocess, explore and interpret real data
 - Learn how to model real problems using computational techniques

Intro to R



David Li

What is R?

- ❑ Statistical computer language similar to S-plus
- ❑ Interpreted language (like Matlab)
- ❑ Has many built-in (statistical) functions
- ❑ Easy to build your own functions
- ❑ Good graphic displays
- ❑ Extensive help files

Strengths

- ❑ Many built-in functions
- ❑ Can get other functions from the internet by downloading libraries
- ❑ Relatively easy data manipulations

Weaknesses

- ❑ Not as commonly used by non-statisticians
- ❑ Not a compiled language, language interpreter can be very slow, but allows to call own C/C++ code

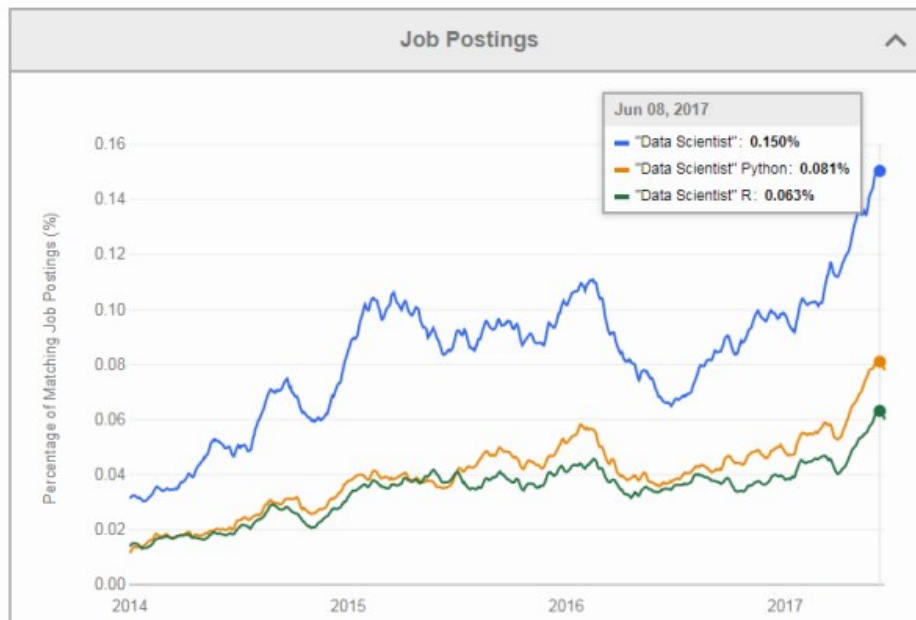
R, Statistics, Data Science

- ❑ Packaging: a crucial infrastructure to efficiently produce, load and keep consistent software libraries from (many) different sources / authors
- ❑ Statistics
 - most packages deal with statistics and data analysis
 - State of the art: many statistical researchers provide their methods as R packages
- ❑ **R vs Python for Data Science: The Winner is ...**
 - <https://www.kdnuggets.com/2017/09/python-vs-r-data-science-machine-learning.html>
 - On the web, you can find many numbers comparing the adoption and popularity of R and Python
 - ❑ You will find R only in a data science environment; As a general purpose language, Python, on the other hand, is widely used in many fields, such as web development.
 - http://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1523009719/main-qimg-9dcf536c501455f073dfbc4e09798a51_vpijr0.png

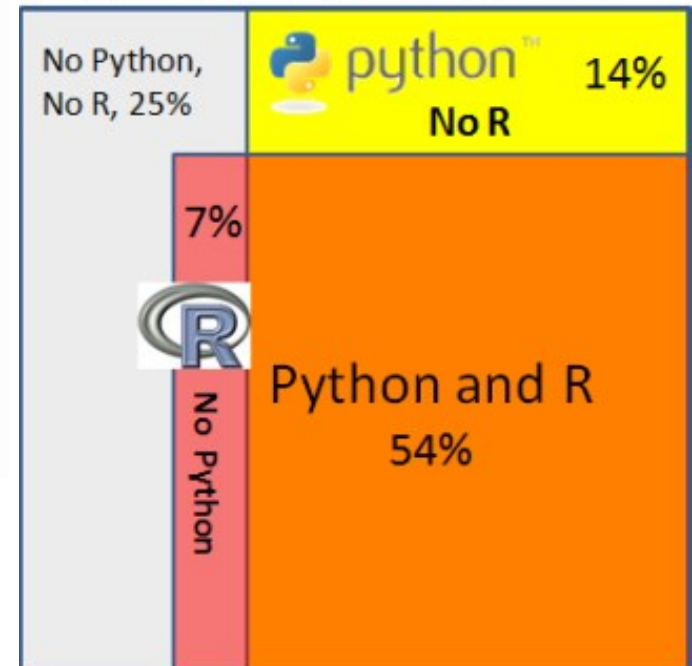
R vs Python

"Data Scientist", "Data Scientist" Python, and "Data Scientist" R Job Trends

"Data Scientist" x "Data Scientist" Python x "Data Scientist" R x + Add Term Find Trends



Data Science Jobs in USA on indeed.com



A sample job opening

Data Science Engineer

Full-Time, Permanent

New York, NY

Finance

- Understanding algorithms, analyzing, and crafting efficient algorithms
- Experience with industry-leading Data Science
- R, NumPy/Python; Big Data and NoSQL technologies such as Spark, Hadoop, MongoDB, and Cassandra required
- A plus - Microsoft software engineering stack

If you are qualified, interested, and planning to make a change, or know of a friend who might have the required qualifications... please respond via email with your current resume and best phone number so I can reach you. Either way, I am looking forward to hearing from you and ask that you please do not hesitate to contact me if there's anything I can do to help.

When to use R?

□ When

- Requires standalone computing or analysis on individual servers.
- Great for exploratory work: it's handy for almost any type of data analysis because of the huge number of packages and necessary tools to get up and running quickly
- R can even be part of a big data solution.

How to use/learn R?

□ How

- (optional) Install and Use Rstudio IDE
- (optional) Install Jupyter with R kernel
- Getting started with R (Basic grammars)
- Get to use/learn those popular packages
 - dplyr, plyr and reshape2 for data manipulation
 - stringr for string operation
 - ggplot2 for data visualization
 - ...
- Do (a lot of) practices including real projects

Install RStudio

- ❑ An integrated development environment (IDE) available for R
 - a nice editor with syntax highlighting
 - there is an R object viewer
 - there are a number of other nice features that are integrated
- ❑ How to install
 - <https://www.youtube.com/watch?v=9-RrkJQQYqY>

Install Jupyter with R kernel

1. Install R and Rstudio
2. Download and install the latest Anaconda at <https://www.anaconda.com/download/>
3. In windows, add your R bin path and Anaconda3 Scripts path to your environmental variable "Path"
 - In my computer the R bin path is C:\Program Files\R\R-3.5.1\bin
 - Anaconda3 Scripts path is C:\ProgramData\Anaconda3\Scripts, the paths in your computer may vary.
 - How to set the path and environment variables in Windows
<https://www.computerhope.com/issues/ch000549.htm>
 - Install R kernel to Jupyter (PLEASE DO THIS STEP IN R CONSOLE, not in Rstudio or RGui)
<https://irkernel.github.io/installation/>
<https://stackoverflow.com/questions/44056164/jupyter-client-has-to-be-installed-but-jupyter-kernelspec-version-exited-wit>
 - Then you can start "Jupyter Notebook" from the start menu.

Starting and stopping R

□ Starting

- Windows: **Double click on the R icon**
- Unix/Linux: **type R (or the appropriate path on your machine)**

□ Stopping

- **Type `q()`**
- `q()` is a function execution
- Everything in R is a function
- `q` merely returns the content of the function

Writing R code

- ❑ Can input lines one at a time into R
- ❑ Can write many lines of code in any of your favorite text editors (including Rstudio) and run all at once
 - Simply paste the commands into R
 - Use function `source("path/yourscript")`, to run in batch mode the codes saved in file "yourscript" (use `options(echo=T)` to have the commands echoed)

R as a Calculator

```
> log2(32)
```

```
[1] 5
```

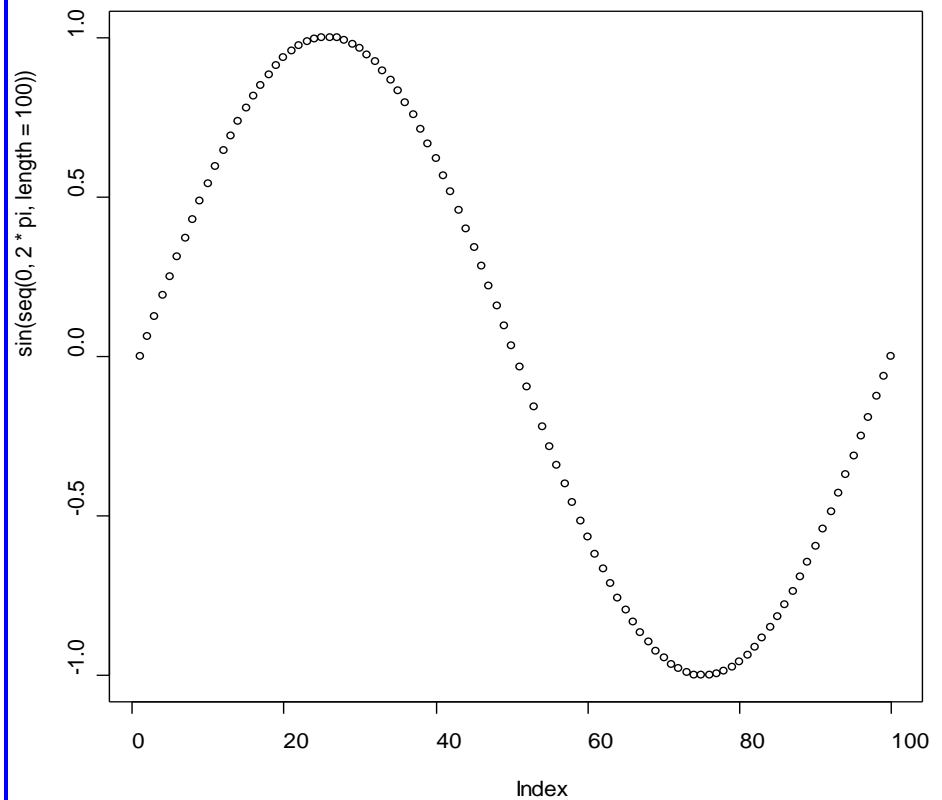
```
> sqrt(2)
```

```
[1] 1.414214
```

```
> seq(0, 5, length=6)
```

```
[1] 0 1 2 3 4 5
```

```
> plot(sin(seq(0,  
  2*pi, length=100)))
```



Recalling Previous Commands

- ❑ In WINDOWS/UNIX one may use the **arrow up key** or the **history** command under the menus
- ❑ Given the history window then one can copy certain commands or else past them into the console window

Language layout

- Three types of statement
 - expression: it is evaluated, printed, and the value is lost ($3+5$)
 - assignment: passes the value to a variable but the result is not printed automatically (`out<-
3+5`)
 - comment: (`#This is a comment`)

Naming conventions

- ❑ Any roman letters, digits, underline, and '.' (non-initial position)
- ❑ Avoid using system names: c, q, s, t, C, D, F, I, T, diff, mean, pi, range, rank, tree, var
- ❑ Hold for variables, data and functions
- ❑ Variable names are case sensitive

Arithmetic operations and functions

- ❑ Most operations in R are similar to Excel and calculators
- ❑ Basic: `+` (add), `-` (subtract), `*` (multiply), `/` (divide)
- ❑ Exponentiation: `^`
- ❑ Remainder or modulo operator: `%%`
- ❑ Matrix multiplication: `%*%`
- ❑ `sin(x)`, `cos(x)`, `cosh(x)`, `tan(x)`, `tanh(x)`, `acos(x)`, `acosh(x)`, `asin(x)`, `asinh(x)`, `atan(x)`, `atan(x,y)` `atanh(x)`
- ❑ `abs(x)`, `ceiling(x)`, `floor(x)`
- ❑ `exp(x)`, `log(x, base=exp(1))`, `log10(x)`, `sqrt(x)`, `trunc(x)` (the next integer closer to zero)
- ❑ `max()`, `min()`, `mean()`, `median()`

Defining new variables

- ❑ Assignment symbol, use "<-" (shortcut: alt -) or =
- ❑ Scalars
 - >scal<-6
 - >value<-7
- ❑ Vectors; using c() to enter data
 - >whales<-c(74,122,235,111,292,111,211,133,16,79)
 - >simpsons<-c("Homer", "Marge", "Bart", "Lisa", "Maggie")
- ❑ Factors
 - >pain<-c(0,3,2,2,1)
 - >fpain<-factor(pain,levels=0:3)
 - >levels(fpain)<-c("none", "Mild", "medium", "severe")

Use functions on a vector

- ❑ Most functions work on vectors exactly as we would want them to do
 - >sum(whales)
 - >length(whales)
 - >mean(whales)
 - sort(), min(), max(), range(), diff(), cumsum()
- ❑ Vectorization of (arithmetic) functions
 - >whales + whales
 - >whales - mean(whales)
 - Other arithmetic funs: sin(), cos(), exp(), log(), ^, sqrt()
 - Example: calculate the standard deviation of whales

$$\text{SD}(X) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}.$$

Functions that create vectors

□ Simple sequences

```
>1:10  
>rev(1:10)  
>10:1  
>c(1:10, 10:1)  
>fractions(1/(2:10))  
>library(MASS) #to have fractions()
```

□ Arithmetic sequence

- $a + (n-1) \cdot h$: how to generate 1, 3, 5, 7, 9?

```
>a=1; h=2; n=5      OR      >seq(1,9,by=2)  
>a+h*(0:(n-1))      >seq(1,9,length=5)
```

□ Repeated numbers

```
>rep(1,10)  
>rep(1:2, c(10,15))
```

- getting help: `?rep` or `help(rep)`
- `help.search("keyword")` or `??keyword`

Next week

- More data structure and R packages
 - Lab 1
 - Homework 1
- Please find your lab group mates and send me email. I expect 13 groups of 39 students.