# CS636 Data Analytics with R Programing

Instructor
David Li

# Course Logistics

- Basic information
- Requirements
- Goal

# CS636 Data Analytics with R Programing

- **Class Schedule:** Saturday 9:00 am – 1:00 pm, WebEx
- **Instructor:** David Li, email: dli@njit.edu

- **TA:** Jianlan Ren, email: jr689@njit.edu
- **Hours**: Friday 1-2pm, or by appointment in other days
- **Suggested Remote Tool**: TeamViewer

- **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**
  - https://njit.instructure.com/courses/12084

# Requirements

- Homework & computing lab exercise (10%)
- Quiz (20%)
- Term Project (20%)
- Midterm (20%)
- Final (30%)

# Homework (10 %)

- Homework assignments
  - Only use R in homework
  - 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

- Have a lab session every week
- Lab exercises
  - Focus on R computing exercises
  - We will solve some simple problems
  - Post your answers by replying on canvas
  - Some answers may be selected for discussion by the end of lab session.
  - Some problems may become part of homework

# Two Term Projects (20%)

- You can choose R or Python for your projects
- Use Jupyter
- Submit code and report to summarize what you have done and results you obtained.
- Prepare for presentation and demo.
- 1~4 students a group.
- More details to be announced soon
- Cheating/Copying is strictly prohibited. I will report to Dean and you will get F in this course.
- If you think your group members don't make contribution, talk to me.

# Quiz (20%)

- Focus on course materials.
- 4 Quizzes
- Every other week
- Only R is allowed

# Two Exams (50%)

- One midterm and one Final (20%+30%)
  - In-class
  - Open book
  - Final is cumulative
  - Only R

# Some tips

- Set up your dev environment for exams. It may require to write code.
- 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

- We use R to teach this class but the content is for generic data science
- Focus on the skills that can be transferred to Python
- 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

# Prerequisites

- Basic programming skills
- Linear algebra
- Probability
- Statistics

# Tentative course topics

(Subject to changes according to progress)

- R libraries for data science. The most common knowledge that you can easily apply to Python.
- Big Data and Graph Analytics
- Visualization with probability and statistics basics
- Regular Expression and NLP for text processing
- Machine learning algorithms (a lot of math)
- Model/Feature Selection(more math)
- May cover advanced big data and deep learning

# 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 packages

- 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

# A sample job opening

- Experience using Statistical and Machine Learning algorithms on real data, in commercial environments. Experience in the Telecom Domain is a big plus.
- Excellent and wide ranging experience in supervised and unsupervised learning. Reinforcement learning a plus.
- Experienced in the use and design of logistic regression, support vector machines, ensemble trees, and neural networks. Optimization problems a plus
- Familiarity and experience with the standard machine learning packages, such as numpy, scipy scikit-learn, TensorFlow, keras and Theano
- Experience in data munging, data cleansing etc.
- Experience in feature selection, feature engineering and development of recommender systems.
- Very Good Knowledge on one and more Statistical Tool like R/Python
- Excellent communication and interpersonal skills, with proven ability to take initiative and build strong, productive relationships.
- Good to have experience in Network Analytics. Building a strong intuitive understanding of the problem domain (Next Generation Access Networks).
- Influence and transform the end-to-end delivery process to maximize the value Customers gain from Analytics
- Should be able to handle multiple projects and liaison with customer different teams to bring overall value add.
- Should have strong people skills and good team management experience.
- Identify testable hypotheses to explain interesting phenomena in this domain
- Constructing an automated system test framework
- Develop and communicate goals, strategies, tactics, project plans, timelines, and key performance metrics to reach goals
- Experience with public cloud (AWS) desirable
- Great communication skills
- Proficiency in using query languages such as SQL
- Good scripting and programming skills, such as R, Python, or Spark

# 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)

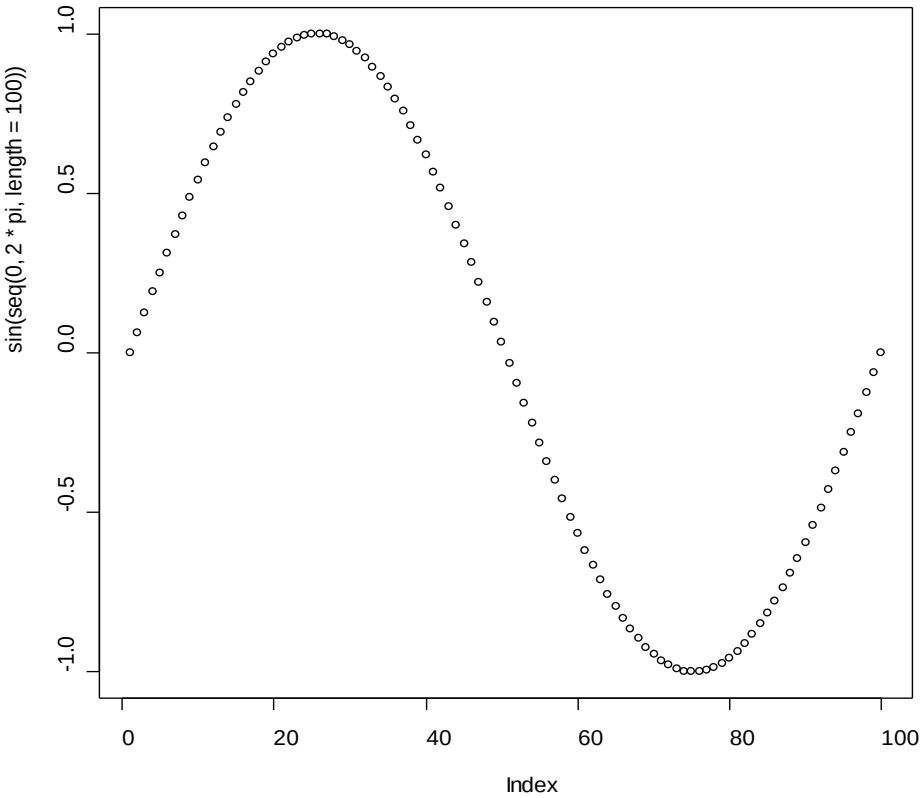
# R as a Calculator

```
> 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

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

# 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)*h: how to generate 1, 3, 5, 7, 9?

```
>a=1; h=2; n=5
>a+h*(0:(n-1))
```

OR

```
>seq(1,9,by=2)
>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

- Quiz 1. Focus on Jupyter and simple coding skills
- More data structure and R packages
- Homework 1
- Please find your project mates.