MIS 413/572:

Introduction to Big Data Analytics

Introduction to R and RStudio

Yihuang Kang

What is "R"?

- R is a software environment for statistical computing and graphics (official definition)

- R is a computer language for scientific computing, data analysis, data management, and data visualization.

- R is object-oriented, functional, cross-platform, and interpreted programming language.

Why R?

- ## R is free
  R is part of GNU projects. R and most of its modules (packages) are open-source and available online.

- ## R is comprehensive
  Thousands of optional packages for data mining, statistical computing, machine learning, operations research, data visualization, ... etc.

- ## R is advanced
  Most update-to-date analytics functions, routines, algorithms are usually implemented in R.

What is "RStudio"?

- [RStudio](#) is a free, cross-platform, and open source integrated development environment (IDE) for R (official definition again).

- A great IDE that make programming in R as easy as programming in other major high-level scientific computing languages.

- RStudio provides both desktop and server versions that enable you to run R code on local machine as an application or on remote Unix-like server.
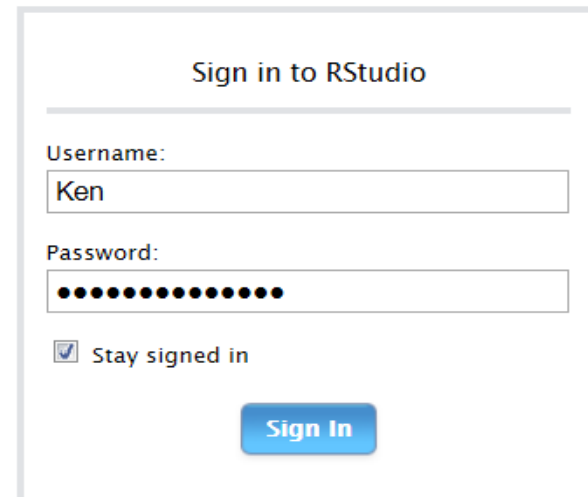
RStudio Desktop Installation

- Download latest R and RStudio from http://www.r-project.org/ and http://www.rstudio.com/.

- Follow the instructions to install. After successful installations, Open RStudio, go to [Tools]-> [Option] then click [Change...] to specify the location of the *R version* you'd like to use, if you use MS Windows.
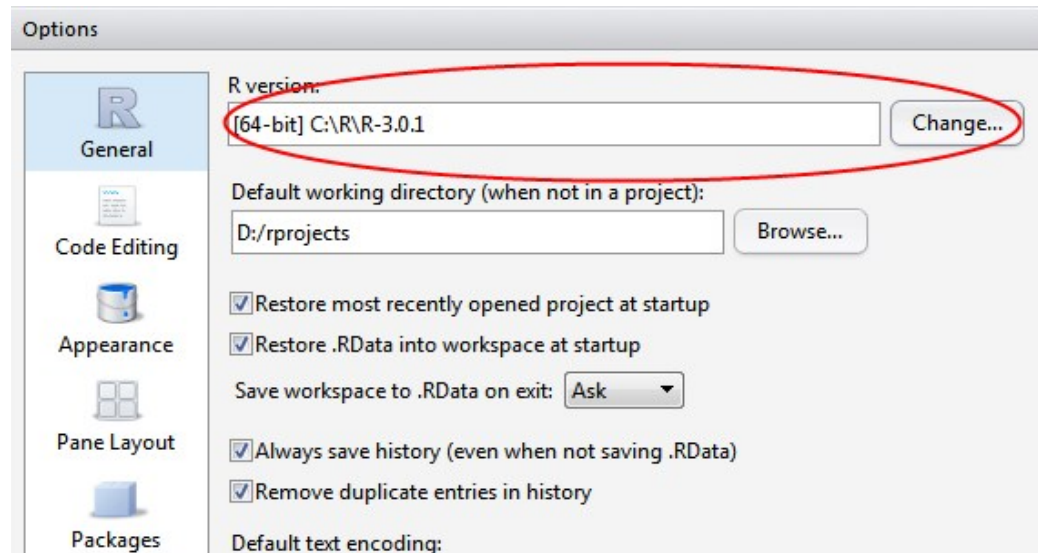
RStudio Server

- Or, you may just open your browser to login your remote server with R and RStudio Server.

Rstudio

Sign in to RStudio

Username:

Ken

Password:

●●●●●●●●●●●●●

☑ Stay signed in

Sign In

Using RStudio Desktop & Server

- Specify your preferred 4 panes of layout in [Tools] -> [Option]->[Panel Layout]

- **Source** panel is an R source code editor.

- **Console** panel is an real-time R statement interpreter. You can also use it as a calculator.

Using RStudio Desktop & Server (cont.)

Environment	Current visible variables/objects in memory.
History	History of executed R commands/statements.
Files	Current local working directory.
Plots	Recent plots/figures.
Package	R package management.
Viewer	Local web content viewer.
Help	R package/function documentations.
Build	Commands to build your own R packages.

Using RStudio Desktop & Server (cont.)

- R as a real-time calculator by typing commands in **Console**. E.g.

```
a = 2 * 2 + 1
a
[1] 5
```

- R for symbolic computing. E.g. compute derivative:
Check **Ryacas** package for more information

```
f = expression(x^2 + x + 1)
D(f,"x")
2 * x + 1
```

Using RStudio Desktop & Server (cont.)

- Note that everything in R is an ***object***. Now there are 2 objects/variables, "a" and "f", shown in your **Environment**. We can also check them by typing function *ls()*:

```
ls()
[1] "a" "f"
```

- Remove objects from memory/ R environment:

```
rm(a) # remove "a"

rm(list = ls() ) # remove all objects
```

Using RStudio Desktop & Server (cont.)

- There are many keyboard shortcuts of RStudio that helps you program in R efficiently. E.g. We can clear the **Console** messages by pressing **Ctrl + L**

- For more keyboard shortcuts, please check: http://www.rstudio.com/ide/docs/using/keyboard_shortcuts

R Documentation

- R documentations are relatively poor compared to other programming languages but are getting better.

- Check the general help files by entering:

```
help.start()
```

- If you know the function names, simply enter:

```
help("functionName")
?functionName
```

- Or, search all documentations by typing a keyword/string:

```
help.search("a_keyword")
??a_keyword
```

R Documentation(cont.)

- Here is an example. If we forget a function that generates normal random numbers, just enter:

```
help.search("norm")
```

to get all function named with "norm".

- If we are interested in the examples of a function, say *mean(),* in R documents, we can ask R to run the examples in the document for the function.

```
example(mean)
```

R Package

- One of exciting part of programming in R is that there're thousands of packages that could suit you needs. For example,

  ✓ writing some SQL codes? – *sqldf*

  ✓ manipulating massive datasets? –*data.table*

  ✓ mining data with Weka? – *RWeka*

  ✓ creating some fancy plots? – *ggplot2*

  ✓ getting ODBC/JDBC DB access? – *RODBC/RJDBC*

  ✓ doing "point-and-click" statistical analyses? – *Rcmdr*

  .... and many more!

R Package(cont.)

- Using RStudio, we can easily download and install R packages from *CRAN* (The Comprehensive R Archive Network). Just click **Package** Tab -> **Install Packages**, then enter the package name.

- Visit http://cran.r-project.org/ for more details about R packages.

- To find packages, you may also consider checking http://www.rseek.org/

- As always, Google is your best friend to find more interesting packages!