Introduction to R Studio

Math 275 Multivariable Calculus

September 13, 2013

Acknowledgement

This deck borrows some background material from Dr. Roger Peng. Check out his YouTube page for lots of videos about how to use R (mostly for statistics).

Development of R

R is a variant or descendant of an earlier language called S.

- ▶ S was developed at Bell Labs by a team led by John Chambers.

Development of R

R is a variant or descendant of an earlier language called S.

- S was developed at Bell Labs by a team led by John Chambers.
- In 1976 the S project began as a set of Fortran libraries for internal use only.

Development of R

R is a variant or descendant of an earlier language called S.

- S was developed at Bell Labs by a team led by John Chambers.
- In 1976 the S project began as a set of Fortran libraries for internal use only.
- In 1988 the system was rewritten in C: this is version 3.

Development of R

R is a variant or descendant of an earlier language called S.

- S was developed at Bell Labs by a team led by John Chambers.
- In 1976 the S project began as a set of Fortran libraries for internal use only.
- In 1988 the system was rewritten in C: this is version 3.
- In 1998 Version 4 was released and is still in use—hasn't changed much.

Development of R

R is a variant or descendant of an earlier language called S.

- S was developed at Bell Labs by a team led by John Chambers.
- In 1976 the S project began as a set of Fortran libraries for internal use only.
- In 1988 the system was rewritten in C: this is version 3.
- In 1998 Version 4 was released and is still in use—hasn't changed much.
- R is an implementation of the S language.

S philosophy

- It is possible to use S (and R) in a solely interactive way.

This philosophy carries down to R.

S philosophy

- It is possible to use S (and R) in a solely interactive way.
- As a user develops her abilities, this naturally turns into programming.

This philosophy carries down to R.

S philosophy

- It is possible to use S (and R) in a solely interactive way.
- As a user develops her abilities, this naturally turns into programming.
- Then the language and system become more important

This philosophy carries down to R.

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ
- 1995: Decision is made to license under GNU General Public License

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ
- 1995: Decision is made to license under GNU General Public License
- 1997: R Core Group is formed; this group controls the source code of R.

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ
- 1995: Decision is made to license under GNU General Public License
- 1997: R Core Group is formed; this group controls the source code of R.
- 2000: v1.0.0

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ
- 1995: Decision is made to license under GNU General Public License
- 1997: R Core Group is formed; this group controls the source code of R.
- 2000: v1.0.0
- 2012: v2.15.1

R development

- 1991: R project initiated by R. Ihaka and R. Gentleman in NZ
- 1995: Decision is made to license under GNU General Public License
- 1997: R Core Group is formed; this group controls the source code of R.
- 2000: v1.0.0
- 2012: v2.15.1
- today: v3.0.1 (I think?)

Features of R

- Similar to S, easy for legacy users to switch

Features of R

- Similar to S, easy for legacy users to switch
- Runs on most any OS (even PS3)

Features of R

- Similar to S, easy for legacy users to switch
- Runs on most any OS (even PS3)
- Modular packages

Features of R

- Similar to S, easy for legacy users to switch
- Runs on most any OS (even PS3)
- Modular packages
- Sophisticated graphics capabilities, better than most stat packages

Features of R

- Similar to S, easy for legacy users to switch
- Runs on most any OS (even PS3)
- Modular packages
- Sophisticated graphics capabilities, better than most stat packages
- Active user community

Free software

- Freedom to run the program for any purpose

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code
- Freedom to redistribute, even for a profit

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

- Freedom to redistribute, even for a profit
  - Usually, the requirement is that copies you make are subject to the same freedoms

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

- Freedom to redistribute, even for a profit
  - Usually, the requirement is that copies you make are subject to the same freedoms

- Freedom to improve and release patches, new packages, etc.

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

- Freedom to redistribute, even for a profit
  - Usually, the requirement is that copies you make are subject to the same freedoms

- Freedom to improve and release patches, new packages, etc.
  - Also impossible without source code.

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

- Freedom to redistribute, even for a profit
  - Usually, the requirement is that copies you make are subject to the same freedoms

- Freedom to improve and release patches, new packages, etc.
  - Also impossible without source code.

- R is without question one of the greatest successes of the free-software movement.

Free software

- Freedom to run the program for any purpose
- Freedom to change the program for your particular needs
  - Impossible without access to the source code

- Freedom to redistribute, even for a profit
  - Usually, the requirement is that copies you make are subject to the same freedoms

- Freedom to improve and release patches, new packages, etc.
  - Also impossible without source code.

- R is without question one of the greatest successes of the free-software movement.
- The Free Software Foundation fsf.org

Reproducible research

R is very widely used worldwide by all kinds of data scientists. Partly, this is because its free nature encourages the development of *reproducible research*.

> *Conclusions whose soundness is a consequence of various calculations must be verifiable.*

- How should the community best verify the calculations?

Reproducible research

R is very widely used worldwide by all kinds of data scientists. Partly, this is because its free nature encourages the development of *reproducible research*.

> *Conclusions whose soundness is a consequence of various calculations must be verifiable.*

- How should the community best verify the calculations?
- Well, if the paper includes all the R code and scripts used to conduct them…

Reproducible research

R is very widely used worldwide by all kinds of data scientists. Partly, this is because its free nature encourages the development of *reproducible research*.

> *Conclusions whose soundness is a consequence of various calculations must be verifiable.*

- How should the community best verify the calculations?
- Well, if the paper includes all the R code and scripts used to conduct them…
- …just "run the paper" and see for yourself.

Reproducible research

R is very widely used worldwide by all kinds of data scientists. Partly, this is because its free nature encourages the development of *reproducible research*.

> *Conclusions whose soundness is a consequence of various calculations must be verifiable.*

- How should the community best verify the calculations?
- Well, if the paper includes all the R code and scripts used to conduct them…
- …just "run the paper" and see for yourself.
- Since even the internals of R are open and free, there is no part of the enterprise that is hidden from view.

Some limitations

- 40-year old technology

Some limitations

- 40-year old technology
- Support for dynamic graphics, animation, or 3D graphics is less than ideal

Some limitations

- 40-year old technology
- Support for dynamic graphics, animation, or 3D graphics is less than ideal
- Loads all objects into memory (this isn't really a problem for us)

Design of the R system

- "base" system, available from CRAN

Design of the R system

- "base" system, available from CRAN
  - **C**omprehensive **R A**rchive **N**etwork

Design of the R system

- "base" system, available from CRAN
  - **C**omprehensive **R A**rchive **N**etwork
  - contains the **base** package which provides the fundamentals

Design of the R system

- "base" system, available from CRAN
  - **C**omprehensive **R A**rchive **N**etwork
  - contains the **base** package which provides the fundamentals
  - also contains packages like **utils**, **datasets**, **graphics**, **gr**Devices, **grid**, **methods**, **tools**, **parallel**, **compiler**, **splines**, **tcltk**, **stats4**.

Design of the R system

- "base" system, available from CRAN
    - **C**omprehensive **R A**rchive **N**etwork
    - contains the **base** package which provides the fundamentals
    - also contains packages like **utils**, **datasets**, **graphics**, **gr**Devices, **grid**, **methods**, **tools**, **parallel**, **compiler**, **splines**, **tcltk**, **stats4**.

- CRAN contains about 4000 packages developed by users around the world

Design of the R system

- "base" system, available from CRAN
  - **C**omprehensive **R A**rchive **N**etwork
  - contains the **base** package which provides the fundamentals
  - also contains packages like **utils**, **datasets**, **graphics**, **gr**Devices, **grid**, **methods**, **tools**, **parallel**, **compiler**, **splines**, **tcltk**, **stats4**.

- CRAN contains about 4000 packages developed by users around the world

- there's no way of knowing how many unofficial packages have been developed and released via private websites, etc.

Literate programming

"Literate programming" is an idea of Donald Knuth. Basically, it's a philosophy of software development.

> *Programs should be written so that they are simultaneously human-readable and computer-readable.*

R is at the center of a developing revival of literate programming, in the context of a cultural focus on reproducibility. Packages like **Sweave** and **knitr** combine with rendering engines like LaTeX and Markdown and the text conversion utility Pandoc to generate extremely portable, flexible, and professional scientific documents with comparatively little effort.

Getting help

There is a very large and vibrant user community online. Chances are, someone has already asked your question and received an answer.

Check out this video http://youtu.be/ZFaWxxzouCY for tips on getting help. Many of the best forums are quite serious and your question will be closed and ignored if it does not measure up to community standards. Make sure to follow proper forums etiquette (check FAQs and archives before posting).