R You Ready?

Paul E. Johnson,
Prof., Political Science
Assoc. Dir, Center for Research Methods and Data Analysis

University of Kansas

Acknowledgment: Thanks to the r-help crowd, especially Pat Burns, Deepayan Sarkar, John Fox, and Sandy Weisberg, for their useful examples
Mission for this talk

- Describe “R”
- Illustrate some of its uses

Future “hands-on” computing sessions can be scheduled.

Alert: KU Summer Stats Camp will offer 1 week-long session on R taught by some well qualified folks :) http://www.quant.ku.edu
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
“R is a little bit like an elephant”
R is

a free/open implementation of S.
a SAS/SPSS replacement for stats and graphs (salvation from Excel)
a statistical toobench for rapid model development by statisticians.
an open community of scholars who cooperate, exchange, and enhance each other’s work product
1 What is R?

2 If You Knew S, you’d Feel Right At Home!

3 OK, What Does It DO?

4 Graphics is a Major Selling Point for R

5 R Handy for Teaching Statistics

6 Packages: Addon Components for R

7 Data Importation Anecdote

8 If You Want To Get Started

9 Appendix 1: Code for Simulation Examples
The S Language was developed at Bell Labs (mid 1970s). See Richard Becker’s “Brief History of S” about the AT&T years.

*S-plus* is a commercial product that answers to S syntax commands (from the Insightful Corporation).

There have been 4 generations of the S language.

- Currently, S3 and S4 are in use.
- In perfect world, transition would not affect users because changes are “under the hood”
What does R Taste Like? Everybody Says "Tastes like S"

- R is a computer language
  - similar to S, but possibly better from a “computer science point of view.”


- R is a program that interprets scripts written in the R language
  - R also can “inter-connect” with other programs.

- R is now the “lingua franca” of research methods development. You Snooze, You Lose.
Does it matter that it is "Open Source"? YES!

- We can inspect, verify, copy, change, fix, and extend R.
- R team also elected to make R available for FREE, without charge.
- R evolves. It is an open, world-wide community of scholars.
- In R-space, nobody can hear (has to listen to) you scream (apologies to Alien)
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
I Don’t Give a Hoot about S. What is R?

- A set of ways to organize data
- All the usual statistical models
- Handy graphs
- Highly “extensible”—open to modular “packages”
- Framework for cooperation with other programs and languages
It's interactive, but not "pointy clicky"

- An interactive session in R looks like this

```
> x <- rnorm(n=1000, mean=10, sd=20)
> mean(x)
[1] 10.07482
> sd(x)
[1] 20.10633
> quantile(x)
   0%   25%   50%   75%  100%
-51.164700 -3.763587 10.293876 22.687147 70.862537
> hist(x)
```

- `>` is the “prompt”. Type stuff there!
There might be some excitement

- A graph pop ups when you type “hist(x)”

- But clicking on the graph doesn’t do anything.
Type more commands to re-draw and beautify the graph.
And a nicer looking histogram pops up

Some GUI do exist (Rcmdr, jagr, rattle, rkward), but....
1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. **Graphics is a Major Selling Point for R**
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
R can create a “blank canvas”

Which can then be decorated with subsidiary plotting commands like:

- lines
- points
- text
- polygon
Recall the old crowd favorite, the Normal Distribution,

\[ x \sim N(\mu, \sigma^2) \]

\( \mu \) is the center point of \( x \)'s range, the expected value, or mean
\( \sigma \) is a dispersion parameter, often called the standard deviation
I warned you. This is one awesome figure!
Getting all Computer-science-ey now:

plot() is magic! It tries to guess what you need, and it gives it to you. R has separate methods to create

- scatterplots
- barplots
- boxplots
- spinograms
- and so forth
plot of 2 numeric variables → get a scatterplot
plot 1 numeric by a categorical variable, get boxplot
plot 2 categorical variables → spineplot

- sex
- pres04
- MALE
- FEMALE
- KERRY
- NADER
- DIDNT VOTE

0.0 0.2 0.4 0.6 0.8 1.0

PJ (KU)
R You Ready?
March 19, 2010 24 / 68
Gender Gap Prettier as a Barplot, IMHO

Presidential Preference, 2004

Respondent Gender

Kerry Bush
Men Women

44.5 52.7

46.6 51.5

PJ (KU)

R You Ready?

March 19, 2010 25 / 68
Best Bar Plot from POLS706 Midterm 2010

- J. Kerry
- G.W. Bush

Strong Dem.
Dem.
Ind Near Dem
Independent
Ind. Near Repub.
Repub.
Strong Repub.

0.0
0.2
0.4
0.6
0.8
1.0

J.Kerry
G.W.Bush
My Best Barplot from the POLS706 Midterm, 2009

Participation in 2000

- VOTED: 93.3%
- DID NOT VOTE: 6.5%
- INELIGIBLE: 0.1%

Participation in 2004

- VOTED: 72.6%
- DID NOT VOTE: 26.9%
- INELIGIBLE: 0.5%

- VOTED: 54.8%
- DID NOT VOTE: 23.3%
- INELIGIBLE: 21.9%
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
R has random variables

- Types of random variable generators (not just Normal, but also many others)
- Calculate theoretical quantities
  - probability density curves
  - cumulative distribution functions
- Draw samples from these distributions
- Conduct simulations (Monte Carlo experiments) easily
  - R has functions to streamline this work.
One Normal Variable, $\mu=50$, $\sigma=20$

A Random Sample from $N(10,400)$

Proportion of Observations

0 20 40 60 80 100

0.000 0.010 0.020 0.030

mean= 49.813
sd= 19.965

A Random Sample from $N(10,400)$
Observed and "True" Probabilities

Possible Values

Density

true under \( N(50,400) \)

observed in sample
Consistent with theory, means should be $\text{Normal}(\mu=50, \sigma = \frac{20}{\sqrt{1500}})$. 

The sampling distribution of the mean is shown with a normal distribution where the mean of means is 50.049 and the standard deviation of means is 0.535.
mean of means = 50.049
sd of means = 0.535
Sample from Exponential is not Normal

An Exponential Random Sample

- Proportion of Observations
- 0, 50, 100, 150, 200, 250, 300
- 0.000, 0.010, 0.020, 0.030
- mean = 51.496
- sd = 51.789

mean = 51.496
sd = 51.789

An Exponential Random Sample
Recall that this is the Central Limit Theorem
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
R Package Writers follow a set of guidelines

Upload packages to CRAN

Available after passing checks & tests

R users can download & install from within R.

```r
> install.packages(c("lmtest","car"), dep=T)
```
What packages do you have already?

```r
> rownames(installed.packages())
```

R provides a set of “recommended” packages, every install will have them.

Wonder what you are missing out on?

```r
> rownames(available.packages())
```

On 2010-03-19, that command returned a list of 2260 packages.

I want it ALL!

I wrote a script that installed them all on a Windows system.

Download and Install took

- 3 hours
- 2.7 Gigabytes of storage

Check for updates periodically

```r
> update.packages( ask=F, checkBuilt=T)
```
I recently learned there is an R package for making and playing SudokU puzzles.

At first, I turned my nose up at the frivolity of it, but then I installed it:

```r
> install.packages("sudoku")
```

After it is installed, run:

```r
> library(sudoku)
```
What is that Sudoku thing?

The first thing I always do after loading a package is find out what is inside it:

```r
> library(help=sudoku)
```
Information on package 'sudoku'

Description:
Package: sudoku
Version: 2.2
Date: 2009-02-02
Title: Sudoku Puzzle Generator and Solver
Author: David Brahm <brahm@alum.mit.edu> and Greg Snow <Greg.Snow@intermountainmail.org>, with contributions from Curt Seeliger <Seeliger.Curt@epamail.epa.gov> and Henrik Bengtsson <hb@maths.lth.se>
Maintainer: David Brahm <brahm@alum.mit.edu>
Suggests: tkrplot
Description: Generates, plays, and solves Sudoku puzzles. The GUI playSudoku() needs package "tkrplot" if you are not on Windows.
License: GPL
Packaged: Mon Feb 2 16:28:15 2009; a215020
Built: R 2.10.1; ; 2010-03-19 06:50:35 UTC; unix
Index:

fetchSudokuUK

fetchSudokuUK

generateSudoku

generateSudoku

hintSudoku

hintSudoku

playSudoku

playSudoku

printSudoku

printSudoku

readSudoku

readSudoku

solveSudoku

solveSudoku

writeSudoku

writeSudoku

Fetch the daily sudoku puzzle from http://www.sudoku.org.uk/

Randomly Generate a Sudoku Puzzle Grid

Give a Hint for a Sudoku Cell

Interactively play a game of Sudoku

Print a Sudoku Grid to the Terminal.

Read a File Containing a Sudoku Grid

Solve a Sudoku Puzzle

Write a Sudoku Grid to a File
Then I use the help feature to find out more on the interesting-looking ones:

```r
d > ?generateSudoku
```

That’s the same as:

```r
d > help(generateSudoku)
```

Perhaps I run the example that is displayed on the help page:

```r
d > example(generateSudoku)
```
When you run a function, the parentheses are required, even if you don’t add any specific arguments. This tells `generateSudoku` to use the default settings.

```r
> generateSudoku()

[1,] 1 0 0 0 0 0 0 0 0
[2,] 7 0 0 0 1 3 5 8 2
[3,] 8 2 0 0 6 0 0 0 0
[4,] 4 0 1 0 2 8 6 0 0
[5,] 0 5 8 0 0 0 4 0 1
[6,] 0 0 0 3 4 0 0 0 0
[7,] 5 0 2 0 7 9 3 1 4
[8,] 0 0 0 0 0 2 0 0 0
[9,] 0 7 0 0 0 0 0 5 0
```
A Nicer Looking Sudoku Puzzle

```r
> myPuzzle <- generateSudoku(Nblank = 20, print.it = F)
> printSudoku(myPuzzle)

+-------+-------+-------+
| 9 3 6 | 1 2 | 5 7 8 |
| 7 5 | 6 9 3 | 2 4 1 |
| 1 2 | 7 5 | 6 |
+-------+-------+-------+
| 8 5 | 9 3 6 | 1 2 |
| 2 4 1 | 5 7 | 9 6 |
| 3 6 | 1 4 | 7 5 |
+-------+-------+-------+
| 5 8 | 3 6 | 1 2 |
| 4 2 | 7 8 | 6 9 |
| 6 3 | 2 4 1 | 8 5 7 |
+-------+-------+-------+
```

PJ (KU)
R You Ready?
March 19, 2010
45 / 68
Torture Yourself with British Sudoku

```r
> printSudoku(fetchSudokuUK())

+-------+-------+-------+
| 2 | 3 |   |
| 4 | 9 | 7 |
|   | 4 | 2 |
+-------+-------+-------+
| 6 | 3 9 | 5 |
| 3 | 8 | 9 |
| 8 | 1 5 | 3 |
+-------+-------+-------+
| 6 | 5 |   |
|   | 9 | 6 8 |
|   | 6 | 1 5 |
+-------+-------+-------+
PJ (KU)
R You Ready?
March 19, 2010
Play Sudoku interactively against R

There is even an interactive on-screen game to be played (with hints for cheaters)
In Some Ways, R is very forgiving

R interprets all of these commands in the same way:

> generateSudoku(Nblank=20, print.it = TRUE)
> generateSudoku(20,T)
> generateSudoku(N=20, p=T)
> generateSudoku(p=T, N=20)

R will try to match up the options with your arguments, but I try to avoid gambling by explicitly naming options. This does not give what you want because the arguments are out of order and unnamed

> generateSudoku(T, 20)
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
> library(memisc)
> idat <- spss.system.file("/home/pauljohn/ps/ps706/DataExample-GSS2006/gss2006.sav")
> idat2 <- as.data.set(idat)
> dat <- as.data.frame(idat2)
> rm(idat2)
> rm(idat)
> table(dat$vote00)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VOTED</td>
<td>1826</td>
</tr>
<tr>
<td>DID NOT VOTE</td>
<td>715</td>
</tr>
<tr>
<td>INELIGIBLE</td>
<td>389</td>
</tr>
<tr>
<td>REFUSED TO ANSWER</td>
<td>0</td>
</tr>
</tbody>
</table>
> library(gmodels)
> CrossTable(dat$vote00)

Cell Contents
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>N / Table Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>VOTED</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>1826</td>
</tr>
<tr>
<td>0.623</td>
</tr>
<tr>
<td>--------</td>
</tr>
</tbody>
</table>

PJ (KU)

March 19, 2010
gmodels package: Tastes Like SPSS in here!

```r
> CrossTable(dat$vote00, dat$sex)

    Cell Contents
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>Chi-square contribution</td>
</tr>
<tr>
<td>N / Row Total</td>
</tr>
<tr>
<td>N / Col Total</td>
</tr>
<tr>
<td>N / Table Total</td>
</tr>
</tbody>
</table>

Total Observations in Table: 2930
<table>
<thead>
<tr>
<th>dat$vote00</th>
<th>MALE</th>
<th>FEMALE</th>
<th>Row Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOTED</td>
<td>779</td>
<td>1047</td>
<td>1826</td>
</tr>
<tr>
<td></td>
<td>0.259</td>
<td>0.199</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.427</td>
<td>0.573</td>
<td>0.623</td>
</tr>
<tr>
<td></td>
<td>0.612</td>
<td>0.632</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.266</td>
<td>0.357</td>
<td></td>
</tr>
<tr>
<td>DID NOT VOTE</td>
<td>317</td>
<td>398</td>
<td>715</td>
</tr>
<tr>
<td></td>
<td>0.130</td>
<td>0.100</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.443</td>
<td>0.557</td>
<td>0.244</td>
</tr>
<tr>
<td></td>
<td>0.249</td>
<td>0.240</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.108</td>
<td>0.136</td>
<td></td>
</tr>
<tr>
<td>INELIGIBLE</td>
<td>177</td>
<td>212</td>
<td>389</td>
</tr>
<tr>
<td></td>
<td>0.378</td>
<td>0.290</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.455</td>
<td>0.545</td>
<td>0.133</td>
</tr>
<tr>
<td></td>
<td>0.139</td>
<td>0.128</td>
<td></td>
</tr>
</tbody>
</table>
I like memisc’s way

```r
> gt <- genTable(percent(vote00) ~ sex, data = dat)
> gt

<table>
<thead>
<tr>
<th>sex</th>
<th>MALE</th>
<th>FEMALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>percent(vote00)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOTED</td>
<td>61.19403</td>
<td>63.18648</td>
</tr>
<tr>
<td>DID NOT VOTE</td>
<td>24.90181</td>
<td>24.01931</td>
</tr>
<tr>
<td>INELIGIBLE</td>
<td>13.90416</td>
<td>12.79421</td>
</tr>
<tr>
<td>REFUSED TO ANSWER</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>N</td>
<td>1273.00000</td>
<td>1657.00000</td>
</tr>
</tbody>
</table>
```
mainly because it easily goes to \LaTeX

<table>
<thead>
<tr>
<th></th>
<th>MALE</th>
<th>FEMALE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOTED</td>
<td>61%</td>
<td>63%</td>
</tr>
<tr>
<td>DID NOT VOTE</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>INELIGIBLE</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>REFUSED TO ANSWER</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>N</td>
<td>1273</td>
<td>1657</td>
</tr>
</tbody>
</table>
1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
My new policy. I won’t help students unless they follow my “Workspace Advice” for R. In essence,

1. Create a “folder”
2. Copy a template R file into that folder
3. Open that R file with the Emacs text editor
4. Launch an R session inside an Emacs window
5. Develop the R code by going back-and-forth between the “program buffer” and the “R buffer”

---

1 I put it in the Emacs wiki, it must be right!

http://www.emacswiki.org/emacs/CategoryESS
Commands on left, R session on Right
Emacs is like Democracy. It's the worst, except for all of the others that have been tried...

- Emacs
  - Free
  - Available on all platforms
  - Highly configurable
  - Useful for many other kinds of projects.
Outline

1. What is R?
2. If You Knew S, you’d Feel Right At Home!
3. OK, What Does It DO?
4. Graphics is a Major Selling Point for R
5. R Handy for Teaching Statistics
6. Packages: Addon Components for R
7. Data Importation Anecdote
8. If You Want To Get Started
9. Appendix 1: Code for Simulation Examples
Draw a Sample from the Normal, Create a Histogram

```r
> var1 <- rnorm(n = 1500, mean = 50, sd = 20)
> hist(x = var1, prob = T, breaks = 20, xlim = c(-10, 110), ylim = c(0, 0.03), xlab = "A Random Sample from N(50, 20^2)", ylab = "Proportion of Observations", main = ")
> den1 <- density(var1)
> lines(den1, lty = 2, col = "red")
> legend("topleft", legend = c(paste("mean=", round(mean(var1), 3)), paste("sd=", round(sd(var1), 3))))
```
> plot(den1, xlim = c(-10, 110), ylim = c(0, 0.03),
    xlab = "Possible Values", type = "l", lty = 2,
    col = "red", main = "")
> possValues <- seq(-10, 110)
> trueProbs <- dnorm(possValues, mean = 50, sd = 20)
> lines(possValues, trueProbs, lty = 1, col = "black")
> legend("topright", legend = c("true under N(50,400)",
    "observed in sample"), lty = c(1, 2), col = c("black",
    "red"))
> samp <- replicate(1000, mean(rnorm(n = 1500, mean = 50, sd = 20)))
> hist(samp, prob = T, breaks = 20, ylim = c(0, 1), xlab = "Normal Sample Means", main = "")
> legend("topleft", legend = c(paste("mean of means=", round(mean(samp), 3)), paste("sd of means=", round(sd(samp), 3))))
Re-scale the Previous Histogram

```r
> hist(samp, prob = T, breaks = 20, xlab = "Normal Sample Means",
    xlim = c(-10, 110), ylim = c(0, 1), main = "")
> legend("topleft", legend = c(paste("mean of means=",
    round(mean(samp), 3)), paste("sd of means=",
    round(sd(samp), 3)))))
```
> var1 <- rexp(n = 1500, rate = 1/50)
> hist(x = var1, prob = T, breaks = 20, xlim = c(-10, 300), ylim = c(0, 0.03), xlab = "An Exponential Random Sample", ylab = "Proportion of Observations", main = "")
> den1 <- density(var1)
> lines(den1, lty = 2, col = "red")
> legend("topleft", legend = c(paste("mean=", round(mean(var1), 3)), paste("sd=", round(sd(var1), 3))))
The Central Limit Theorem is Correct

```r
samp <- replicate(1000, mean(rexp(n = 1500, rate = 1/50)))
hist(samp, prob = T, breaks = 20, ylim = c(0, 0.5), xlab = "Sample Means from Exponentials", main = "")
legend("topleft", legend = c(paste("mean of means=", round(mean(samp), 3)), paste("sd of means=", round(sd(samp), 3)))))
```