A Bit About Me

- PhD in Mathematics – began in theoretical CS
- Mostly self-taught statistics – still learning
- Have used Minitab, Excel, Stata, StatCrunch, R to teach Stats courses.
- Have taught 4.5 different "Intro" courses: Intro Stats – regular and bio-flavored, Stats for Engineers, post-calculus 1-semester course mostly taken by Math Ed and CS majors, and 2-semester Math Stat sequence.
- Author of

Some Advice for Teaching with R

1. Start right away.
   Do something with R on day 1. Do something else on day 2. Have students do something by the end of week 1 at the latest.
   (a) Graphics Early.
   Do graphics very early, so that students see that they can get impressive output from simple commands.
   (b) Sampling and Randomization Early.
   Computers are good at this. Take advantage.

2. Illustrate frequently.
   Have R running every class period and use it as needed throughout the course so students can see what R does. Preview topics by showing before asking students to do things.

3. Teach R as a programming language. (But don’t overdo it.)
   There is a bit of syntax to learn – so teach it explicitly.

4. “Less volume, more creativity.” [Mike McCarthy, head coach, Green Bay Packers]
   Use a few methods frequently and students will learn how to use them well, flexibly, even creatively.

5. Find a way to have computers available for tests.

6. Rethink your course.
   Don’t just do what you have been doing but “with R”. Think about how using R could (should) change what you teach and how you teach it.

7. Anticipate computationally challenged students, but don’t give in.

Some Additional Resources

There are lots of resources to help you teach using R. A good place to start is at the Comprehensive R Archive Network (CRAN, http://cran.r-project.org/).
Here are some relatively kids on the block.
- Project MOSAIC (http://mosaic-web.org/) hosts M-casts (webinars) and other activities focussing on Modeling, Statistics, Calculus, and Computation.
- RStudio (http://rstudio.org/) is a new integrated development environment (IDE) for R. RStudio combines an intuitive user interface with powerful coding tools to help you get the most out of R.
  ◦ Can be run on a server and accessed via a web browser
  ◦ Offers a system for creating manipulable graphics (sliders, check boxes, drop down menus, etc.) that can be built with very little code.
  ◦ R code and Sweave authoring tools
  ◦ Support for sharing files with other users.
  ◦ Check out their booth here at JSM.
- R packages:
  mosaic (Project MOSAIC),
  fastR (companion to Foundations and Applications of Statistics),
  abd (companion to Analysis of Biological Data by Whitlock and Schluter)
### Enough R for Intro Stats

One key to success using R in Intro Stats is keeping the volume low. The commands below are sufficient for an Intro Stats course. In fact, many of these could be excluded. See [http://www.calvin.edu/~rpruim/courses/m143/S11/fromClass/](http://www.calvin.edu/~rpruim/courses/m143/S11/fromClass/) for notes used in an Intro Stats course for biology majors for usage examples. The commands below were culled from those notes. Some commands are

### Help

- `apropos()`
- `?`
- `example()`
- `demo()`

### Numerical Summaries

- `mean()` # mosaic augmented
- `median()` # mosaic augmented
- `sd()` # mosaic augmented
- `var()` # mosaic augmented
- `quantile()` # mosaic augmented
- `favstats()` # mosaic
- `summary()` # Hmisc
- `rank()`
- `IQR()`

### Arithmetic

- `+ - * / ^ ( )`
- `log()`
- `exp()`
- `sqrt()`
- `sum()`
- `log10()`
- `abs()`
- `choose()`
- `factorial()`
- `uniroot()`

### Graphics (mostly lattice)

- `lattice` is not the only option, but I find it works well because (a) it allows for easy multi-variable plots with good default settings, and (b) `lattice` uses the same formula interface that will be needed for `lm()`.

- `bwplot()`
- `xyplot()`
- `histogram()`
- `qqmath()`
- `densityplot()`
- `dotPlot()` # mosaic
- `barchart()`
- `barchart( xtabs() )`
- `mosaic()` # in vcd package
- `xhistogram()` # mosaic
- `xqqmath()` # mosaic

### Randomization/Simulation

- `rflip()` # mosaic
- `do()` # mosaic
- `sample()`
- `shuffle()` # mosaic
- `rbinom()` # mosaic
- `rnorm()` # etc, if needed

### Automated Inference

- `binom.test()`
- `prop.test()`
- `chisq.test()`
- `t.test()`
- `wilcox.test()`
- `fisher.test()`
- `xchisq.test()` # mosaic
- `lm()`
- `summary( lm() )`
- `predict( lm() )`
- `anova( lm() )`
- `resid( lm() )`
- `TukeyHSD( aov() )`
- `plot( TukeyHSD( aov() ) )`
- `power.prop.test()`
- `power.t.test()`
- `interval()` # mosaic
- `pval()` # mosaic

### Distributions

- `pbinom()`
- `pnorm()`
- `pchisq()`
- `pt()`
- `qbinom()`
- `qnorm()`
- `qchisq()`
- `qt()`

### Data

- `$`
- `[]`
- `data()`
- `read.csv()`
- `summary()`
- `str()`
- `names()`
- `head()`
- `tail()`
- `xtabs()`
- `table()`
- `perctable()` # mosaic
- `proptable()` # mosaic
- `with()`
- `subset()`
- `merge()`
- `factor()`
- `c()`
- `cbind()`
- `rbind()`
- `data.frame()` # with predict()