> integrating R into
> introductory statistics

Mine Çetinkaya-Rundel - Duke University  
Andrew Bray - UCLA

JSM 2012 - August 2, 2012
> experience
> first course in statistics for non-majors, mostly social sciences majors
> first course in statistics for non-majors, mostly social sciences majors

> 80-120 students in lecture, 25 students / lab section
first course in statistics for non-majors, mostly social sciences majors

80-120 students in lecture, 25 students / lab section

weekly lab sessions using R

labs designed for an interdisciplinary introductory course, can be modified for discipline-specific courses

can also be used in a first data-analysis course for stats majors, ideally by reducing step-by-step instructions
unlike most software designed specifically for courses at this level, R is

- free and open-source
- powerful and flexible
- relevant beyond the introductory statistics classroom
> Why Not R?
perceived challenge of teaching programming in addition to teaching statistical concepts

> labs and activities that try to find the right balance of standard and custom functions

> consistent syntax highlighting helps
> perceived challenge of teaching programming in addition to teaching statistical concepts

> labs and activities that try to find the right balance of standard and custom functions

> consistent syntax highlighting helps

> working with a command line tends to be more intimidating than traditional GUI based tools

> GUI tools also have a learning curve

> a user-friendly IDE (like RStudio)
> RStudio
what it helps resolve:
  - loading and viewing data
  - saving code
  - code history
  - workspace organization
  - plot history
> what it helps resolve:
> loading and viewing data
> saving code
> code history
> workspace organization
> plot history

> what still remains a challenge:
> working with a command line
> balance
> teach coding as a way of introducing/reinforcing concepts, especially those that are otherwise difficult to convey without computation

> simulations

> sampling distributions

> confidence levels

> bootstrapping

> randomization tests

> ...
teach coding as a way of introducing/reinforcing concepts, especially those that are otherwise difficult to convey without computation

- simulations
- sampling distributions
- confidence levels
- bootstrapping
- randomization tests
- ...

hide implementation issues that are outside the scope of the course
> **teach** coding as a way of introducing/reinforcing concepts, especially those that are otherwise difficult to convey without computation

> simulations

> sampling distributions

> confidence levels

> bootstrapping

> randomization tests

> ...

> **hide** implementation issues that are outside the scope of the course

> **minimize** coding for repeated mechanics that can be unified
concept: confidence levels

resample from the population many times and construct many confidence intervals (loops)
concept: confidence levels

resample from the population many times and construct many confidence intervals (loops)

```
population <- ames$Gr.Liv.Area
```
concept: confidence levels

resample from the population many times and construct many confidence intervals (loops)

```
population <- ames$Gr.Liv.Area

samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60
```
concept: confidence levels

resample from the population many times and construct many confidence intervals (loops)

```r
population <- ames$Gr.Liv.Area

samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60

for (i in 1:50) {
  samp <- sample(population, n)  # obtain a sample of size n = 60 from the population
  samp_mean[i] <- mean(samp)    # save sample mean in ith element of samp_mean
  samp_sd[i] <- sd(samp)        # save sample sd in ith element of samp_sd
}
```
concept: confidence levels

resample from the population many times and construct many confidence intervals (loops)

```r
population <- ames$Gr.Liv.Area

samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60

for (i in 1:50) {
  samp <- sample(population, n)  # obtain a sample of size n = 60 from the population
  samp_mean[i] <- mean(samp)     # save sample mean in ith element of samp_mean
  samp_sd[i] <- sd(samp)         # save sample sd in ith element of samp_sd
}

lower <- samp_mean - 1.96 * samp_sd/sqrt(n)
upper <- samp_mean + 1.96 * samp_sd/sqrt(n)
```
concept: confidence levels

plot these confidence intervals and highlight those that do not contain the true population parameter (custom function)
concept: confidence levels

plot these confidence intervals and highlight those that do not contain the true population parameter (custom function)

plot_ci(lower, upper, mean(population))
concept: confidence levels

plot these confidence intervals and highlight those that do not contain the true population parameter (custom function)

```r
plot_ci(lower, upper, mean(population))
```

mu = 1499.6904
concept: statistical inference
concept: statistical inference

> traditional curriculum for an introductory statistics course includes various statistical inference techniques
concept: statistical inference

- traditional curriculum for an introductory statistics course includes various statistical inference techniques
- when introduced as disconnected topic these can be overwhelming to students
concept: statistical inference

> traditional curriculum for an introductory statistics course includes various statistical inference techniques

> when introduced as disconnected topic these can be overwhelming to students

> to help unify inferential concepts, use one function that does it all, but still requires students to think about the nature of the data and encourages them to conduct exploratory data analysis
concept: statistical inference

function: inference() - theoretical and simulation based inference
concept: statistical inference

function: inference() - theoretical and simulation based inference

```r
inference <- function(data, group = NULL, est = c("mean", "median", "proportion"),
                      success = NULL, order = NULL, nsim = 10000, conflevel = 0.95, null = NULL,
                      alternative = c("less", "greater", "twosided"), type = c("ci", "ht"),
                      method = c("theoretical", "simulation"), drawlines = "yes", simdist = FALSE){
  ...
}
```
concept: statistical inference

function: inference() - theoretical and simulation based inference

```r
code

inference <- function(data, group = NULL, est = c("mean", "median", "proportion"), success = NULL, order = NULL, nsim = 10000, conflevel = 0.95, null = NULL, alternative = c("less", "greater", "twosided"), type = c("ci", "ht"), method = c("theoretical", "simulation"), drawlines = "yes", simdist = FALSE){
...
}
```

> data: response variable, quantitative or categorical
> group: explanatory variable, categorical for grouping (optional)
> type: confidence interval or hypothesis test
> method: theoretical or simulation
> ...
question: compare birth weights of babies born to smoker and nonsmoker mothers (source: north carolina births)
question: compare birth weights of babies born to smoker and nonsmoker mothers (source: north carolina births)

input:

```r
inference(data = nc$weight, group = nc$habit, est = "mean", type = "ht", null = 0, alternative = "twosided", method = "theoretical")
```
**question**: compare birth weights of babies born to smoker and non-smoker mothers (source: north carolina births)

**input**:

```
inference(data = nc$weight, group = nc$habit, est = "mean", type = "ht", null = 0, alternative = "twosided", method = "theoretical")
```

**output**:

One quantitative and one categorical variable
Difference between two means
n_nonsmoker = 873 ; n_smoker = 126
Observed difference between means = 0.3155
H0: mu_nonsmoker - mu_smoker = 0
HA: mu_nonsmoker - mu_smoker != 0
Standard error = 0.134
Test statistic: Z = 2.359
p-value: 0.0184
question: compare numbers of sexual partners of males and females (source: national survey of family growth)
question: compare numbers of sexual partners of males and females (source: national survey of family growth)

input:

inference(data = partners, group = gender, type = "ci", est = "mean", method = "theoretical")
question: compare numbers of sexual partners of males and females (source: national survey of family growth)

input:

```
inference(data = partners, group = gender, type = "ci", est = "mean", method = "theoretical")
```

output:

One quantitative and one categorical variable
Difference between two means
n_female = 12190 ; n_male = 10397
Observed difference between means = -0.432
Standard error = 0.0361
95 % Confidence interval = ( -0.5 , -0.36 )
> resources
openintro.org/stat/labs.php
Data: kobe

To access the data in R, type

```r
download.file("http://www.openintro.org/stat/data/kobe.RData", destfile = "kobe.RData")
load("kobe.RData")
```

Description

Data from the five games the Los Angeles Lakers played against the Orlando Magic in the 2009 NBA finals. Each row represents a shot Kobe Bryant took during these games. Kobe Bryant's performance against the Orlando Magic in the 2009 NBA finals earned him the title of Most Valuable Player and many spectators commented on how he appeared to show a hot hand.

Format

A data frame with 133 observations on the following 6 variables

- **vs**: a categorical vector, ORL if the Los Angeles Lakers played against Orlando
- **game**: a numerical vector, game in the 2009 NBA finals
- **quarter**: a categorical vector, quarter in the game, OT stands for overtime
- **time**: a categorical vector, time at which Kobe took a shot
- **description**: a categorical vector, description of the shot
- **basket**: a categorical vector, H if the shot was a hit, M if the shot was a miss
Data: goog

To access the data in R, type

goog <- read.csv("http://www.openintro.org/stat/data/goog.csv")

Description

The data consist of the number of requests Google received for user account information as part of criminal investigations in the first half of 2011, the rate of compliance, and some other indicators on the countries.

Original data: Google's Transparency Report.

Format

A data frame with 26 observations on the following 8 variables.

- **country**: Name of country
- **compiled**: Percentage of requests Google complied with
- **requests**: Number of requests Google received for user account information as part of criminal investigations
- **pop**: Population in country, in thousands
- **hdi**: Human development index, a composite measure of life expectancy, literacy, education, and standard of living on a scale of 0 (least developed) to 1 (most developed)
- **dem**: Democracy index, categorized into full democracies, flawed democracies, and hybrid regimes
- **internet**: Percentage of internet users in country
- **freepress**: Free press index, scored on a scale from 1 (most free) to 100 (least free)
> reactions
positive:

> “I like them. I feel like in the real world we’ll be using software to do stats, so I’m glad we’re learning how to use it.”

> “I LOVE the labs. They really help cement basic statistic ideas, and I especially love that you can finish them in class.”

> “The labs are a lot of fun. It’s great being able to create our own simulations and watch R Studio calculate everything. I also enjoy learning some code.”

negative:

> “The labs are alright. Sometimes I feel like I’m just plugging in stuff and I feel disconnected from what I’m really doing. It’s also frustrating when the code doesn’t work.”

> “Wish other students focused more.”
positive:

> “Super useful and powerful software. It’s exciting to be introduced to it. Once again, don’t always feel comfortable writing code/understanding what I’m doing.”

> “I like it! I kind of know MATLAB, which has helped with the coding a bit, but it’s a little more intuitive/easier, and very helpful.”

> “I am not a computer person at all, but I find RStudio very easy to use.”

> “I like it better than STATA which we used for [another class]. The user interface is easy and there is plenty of help for it online. Overall, it’s pretty good.”

negative:

> “I am not a fan of coding in general. I used Python before and RStudio is better (for me) than Python was, but I am not a fan of either.”

> “Easy to use, language is not too hard to understand although error messages could be more informative.”

> “I don’t think RStudio will have any use to me outside of this class.”
> also...
Additional Considerations

> Labs should be fully integrated with the curriculum
Labs should be fully integrated with the curriculum

“What concepts, if any, are not covered in the textbook? Have you seen these concepts elsewhere, e.g. lecture, discussion section, previous labs, or homework problems? Be specific in your answer.”
> Labs should be fully integrated with the curriculum

“What concepts, if any, are not covered in the textbook? Have you seen these concepts elsewhere, e.g. lecture, discussion section, previous labs, or homework problems? Be specific in your answer.”

> Works best in a classroom environment where students can collaborate with each other and get immediate support
Labs should be fully integrated with the curriculum

“What concepts, if any, are not covered in the textbook? Have you seen these concepts elsewhere, e.g. lecture, discussion section, previous labs, or homework problems? Be specific in your answer.”

Works best in a classroom environment where students can collaborate with each other and get immediate support

“Collect data on the intervals created by other students in the class and calculate the proportion of intervals that capture the true population mean.”
Labs should be fully integrated with the curriculum

“What concepts, if any, are not covered in the textbook? Have you seen these concepts elsewhere, e.g. lecture, discussion section, previous labs, or homework problems? Be specific in your answer.”

Works best in a classroom environment where students can collaborate with each other and get immediate support

“Collect data on the intervals created by other students in the class and calculate the proportion of intervals that capture the true population mean.”

TAs need to be familiar and comfortable with the material
> thank you
contact: mine@stat.duke.edu

web: stat.duke.edu/~mc301

labs: openintro.org/stat/labs.php