MA386 Course Design

Friday, June 12, 2020 9:15 AM

Course-Level Objectives

(A) **Associate** a computational task with the appropriate function(s) or package(s) (suites of related functions) in a statistical computing language.

(B) Given a script, **describe** the computational task being performed.

(C) Given a computational task from the workflow of a statistical project, **construct** a script to complete the task.

(D) Given a research objective, **integrate** multiple computational tasks to provide a data-driven conclusion.

(E) **Communicate** the solution to a computational task, identifying and describing key steps/chunks within the solution.

(F) **Express the value** of scripting a solution to a computational task.

(G) **Identify resources** that generalize the material covered in class in order to learn new tools for solving a novel computational task.

From <https://moodle.rose-hulman.edu/mod/book/view.php?id=2052449>

These are based on Fink's Significant Learning Outcomes (http://www.buffalo.edu/ubcei/enhance/designing/learning-outcomes/finks-significant-learning-outcomes.html):

- Foundational Knowledge - (A) and (B)
- Application - (C)
- Integration - (D)
- Human Dimension - (E)
- Caring - (F)
- Learning to Learn (G)

Course Alignment

Timing for All Modules:

1. In-Class Activity: 1 hour
2. Reading/Videos/Guided Notes: 3 hours
3. Homework Assignment: 2 hours
4. Portfolio Problem: 3 hours
5. Time towards Programming Project: 2 hours

Total Time per Module: 11 hours
Total Time per Programming Project: 10 hours

Course Objectives D and E are often not directly supported by the module-level objectives but are supported directly through the Portfolio Problem assigned which requires integrating multiple tasks and communicating the process and results.

Course Objective F will be assessed through optional writing assignments which earn tokens for revisions in the course.

Module 1: Reproducible Research

We introduce the concept of reproducible research - being able to trace the conclusions from a study to the data and analysis which resulted in those conclusions. As an example, should we update the data, any computed summaries should update as well. R and Rmarkdown documents will be introduced as tools for easily facilitating reproducible research and collaboration among members. We will also introduce basic functionality in R, such as reading in a CSV file containing a dataset and performing basic computations.

Videos:

1. Overview of Reproducible Research
2. Thinking in vectors/variables
3. Anatomy of an Rmarkdown file (external)

Miscellaneous Course Ideas

In-Class Activities:

Begin each in-class activity with answering any questions from the reading that people may have or highlighting my biggest take-aways.

While the activities are meant to be fun, I imagine a lot of students not finding value in them because they are not directly associated with a grade in the course, and they would probably rather be working on their homework. To try and rectify this a bit, at the end of each activity, do a "minute-paper" to find out where people are still confused on issues, including also key take-aways reinforced by the activity.

Forums:

As part of the homework assignments each week, I can imagine some fun forum discussion to help integrate a topic in a fun way (such as design your own meme, or find an example of…).

I like the idea of having a "personal tip" or "coding hack" type ongoing forum for the course where students let me know what they are learning.

We could also have a "what is your coding playlist" type entry.

- Meme
- Song
- Movie

Portfolio Problems:

With the exception of the first (and possibly last) module, all remaining portfolio problems will be linked through a continuous story. This will help form a portfolio of work at the end of the term. This story can be the Spotify dataset this term. I will pattern these problems off previous final exams.

Programming Projects:

Videos:

1. Overview of Reproducible Research
2. Thinking in vectors/variables
3. Anatomy of an Rmarkdown file (external)

Objectives	*Reading*	*Activities*	*Assessments*
Write and **knit** an Rmarkdown (Rmd) file which includes text, an R code chunk, and inline R code, into an html file or notebook. (Supports Course Objectives C, E)	Ch 1, 6, 26, 27, 29.1 - 29.4	Recreate a knitted file.	Portfolio Problem
Read in a CSV file using an appropriate function. (A, B, C)	Ch 11	Guided Notes	Homework Assignment, Portfolio Problem
Given a function which summarizes a variable, **execute** the function on a variable in a dataset. (A, B, C)	Ch 4	Guided Notes	Homework Assignment, Portfolio Problem
Using R as a calculator, **compute** simple arithmetic expressions. (A, B, C)	Ch 4	Guided Notes	Homework Assignment
Examine a "help" file on a function to determine its purpose. (G)	Ch 4	Guided Notes	Homework Assignment

Recreate a knitted file:
Provide students with an HTML document that contains several elements. Ask students to recreate the document through RStudio. This will be a paired programming assignment in which two students share a screen and work together to develop the output.

Portfolio Problem:
Create a "vignette" (or help-page) for computing the mean and standard deviation of a variable within a dataset. This should include both plain text, a mathematical equation, code block, code output, and inline code block. Students will pick their own dataset on which to illustrate their examples.

Module 2: Tidy Data
Not all spreadsheets are created equal. We introduce principles (known as "tidy data") for storing data that make it easy to work with. We also introduce the "tidyverse" as a suite of packages and tools for working with tidy data and accompanying key verbs.

Videos:

1. Overview of Tidy Data

Objectives	*Reading*	*Activities*	*Assessments*
Describe the key aspects of tidy data. (Supports Course Objective A)	Ch 12		Homework Assignment (Forum)
Describe how the following *verbs* act on a dataset and **identify** and **execute** a function/process which corresponds to each verb: arrange, filter, mutate, select, summarise, join, group_by. (A, B)	Ch 5, Ch 12, Ch 13	Guided Notes	Homework Assignment, Portfolio Problem
Given a dataset and computational task, **combine** multiple verbs together to create a summary of interest. (C)	Ch 5, Ch 12, Ch 13, Ch 18	Tag Team Challenge in creating a summary of a dataset.	Portfolio Problem
Describe the primary data structure in statistical computing languages (vectors) and how they relate to datasets. (A, B)	Ch 20		

Tag Team Challenge:
Students will be given a dataset and a summary to construct. Students will work in pairs; however, one person cannot type and can only coach from the side (probably virtually). After a certain amount of time, the two individuals switch places (email the document to the other individual). So, only one person is programming at a time.

Portfolio Program:
Given the Spotify dataset, compute the number of songs, the average danceability, and the percent of songs in a specific category for songs within a defined "speech-like" group. It would be nice to incorporate a join to determine some information about the corresponding artists.

Module 3: Static Graphics
A good graphical summary is often more valuable than an expertly constructed model. We will

be linked through a continuous story. This will help form a portfolio of work at the end of the term. This story can be the Spotify dataset this term. I will pattern these problems off previous final exams.

Programming Projects:
The projects will span 5-weeks of material each time (second project will span entire term but emphasize the last half of the term). These projects should be large enough in scope that students work on them throughout the term.

Video Descriptions

Overview of Reproducible Research:
We want to adopt a workflow that allows the entire analysis to be reproduced by an outside party. It allows other researchers to validate the analysis. External review, for example, can help strengthen the evidence suggested by a particular study - someone else is able to move from raw data to results and see each step of the original analysis. This can reduce the temptation to fabricate results. Internal review, on the other hand, can help eliminate errors due to copy/paste.

There are a couple of steps we can take toward a reproducible workflow:

- Never alter the original raw dataset. Instead, document (or maintain the code) the steps taken to go from raw data to analytic dataset.
- Script an analysis instead of using point-and-click analysis software.
- Publish the code and data.
- Use software that knits together the publication text and the code which generated it within the same document.

One tool for doing this is Rstudio; it allows R/Python/Stan code to be combined with plain text to produce a final document. This is an example of what is known as literate statistical programming.

Thinking in Vectors/Variables:
R is not a programming language so much as a statistical computing language. We could spend a long time trying to list the ways in which R functions differently than a language like C, but I will sum it up like this: it was written by statisticians, not programmers, for analysis, not general application. What I mean is that it has a specific purpose; so, if we want to get the most out of R, we need to think like its creators. Here are some basic things:

- R indexes beginning at 1, not 0. Why? Because that is how people count; we start indexing most sums in statistics and probability at 1.
- The workhorse of statistical theory is vectors and matrices. So, R also thinks in vectors and matrices (or, as variables and datasets).

As we will see moving forward, instead of thinking in loops that need to be applied to every observation of a dataset, we will think/program in variables. Perform some operation (function) on an entire variable or dataset. R gets its speed by thinking in vectors.

Anatomy of an Rmarkdown File:
There are essentially three components to an Rmarkdown file:

- YAML header code which defines how the output will be generated.
- Plain text which forms the body of the article produced.
- Code chunks which perform the analysis.

These are knitted together to produce a publication-ready document. The YAML code controls what the output looks like. We can accept the defaults when we create a new document. We will be primarily working with Rnotebooks, which generate HTML files.

The plain text is markdown code. It allows you to write plain text with minimal markup (bold, italics, etc.). Instead of worrying about the formatting of the final document, you are focused on the content. The layout is determined during the knitting process. Of course, there are ways to tweak everything, but we don't get into that much.

Code chunks accept various options that allow you to control what is inserted into the document. Code can be displayed directly when useful, or hidden from view. Code can be evaluated, and the results dumped into the document, or it can be suppressed.

We knit the document together, when we are ready. For Rnotebooks, that knitting has a slightly different feel, but the overall process is always similar. The final publication-document is what is submitted.

Overview of Tidy Data:
We often think of data in terms of spreadsheets. However, not all spreadsheets are created

Module 3: Static Graphics
A good graphical summary is often more valuable than an expertly constructed model. We will introduce the grammar of graphics as the foundation for creating a graphical summary. We then introduce the ggplot package as a tool which implements this grammar.

Videos:
1. Overview of the grammar of graphics

Objectives	***Reading***	***Activities***	***Assessments***
Identify the appropriate *layer* to add to a graphic in order to display specific information on a graphic. (Supports Course Objectives A, B, C)	Ch 3, 28	Guided Notes	Homework Assignment
Given a graphical summary of a provided dataset, **recreate** the graphic using a statistical computing language. (A, B, C)	Ch 3, 28	Guided Notes	Portfolio Problem
Given a question of interest and an associated dataset, **construct** an appropriate graphical summary to address the question using a statistical computing language. (A, B, C, E)	Ch 3, 28	Gallery Walk (3-round development)	

Gallery Walk:
Students are broken into small teams; each team is given a prompt and asked to design a graphic to address the question. In round 1, the team develops a graphic. In round 2, the students receive the graphic from a different team (maybe Moodle upload/download?). The team needs to revise the graphic they received in some way. In round 3, the students receive the graphic from a different team again. This time, they are tasked with interpreting what they learned from the graphic.

Portfolio Problem:
Given the Spotify dataset, construct a graphical summary examining the relationship between danceability and tempo while taking into account a quantitative variable (valence/positivity?) and a categorical variable (scale?). Maybe discuss some structure about the nature of the theme as well.

Module 4: Programming
Data is rarely in the format needed for an analysis. The process of manipulating data (wrangling) necessarily involves programming. This involves writing custom functions and applying functions efficiently within various data structures. We will introduce the benefit of vectorizing operations and tools for implementing vectorized code.

Videos:
1. Benefits of vectorization

Objectives	***Reading***	***Activities***	***Assessments***
Rewrite a series of nested functions using the pipe operator. (Supports Course Objectives A, B, C, D, E)	Ch 18	Guided Notes	
Write a function to accomplish a small computational task. (A, B, C, F)	Ch 19	Guided Notes	Homework Assignment, Portfolio Problem
Iterate an operation over elements in a data structure. (A, B, C, D, F)	Ch 21	Guided Notes	Homework Assignment, Portfolio Problem
Construct well-structured scripts which are clean and well-documented. (B, E, F)		Debug an existing set of code.	

Debug an Existing set of Code:
Students will be given a script that does not work. Further, the students will be asked to turn it into a function and clean it up. Part of this process will be making the code cleaner to read as part of the debugging process. They can work in pairs on this.

Portfolio Problem:
Given the Spotify dataset, create a function to summarize the correlation between two variables, and then compute that correlation.

Module 5: String Manipulation
Character data brings with it challenges that numeric data does not. We will introduce some of these

different feel, but the overall process is always similar. The final publication-document is what is submitted.

Overview of Tidy Data:
We often think of data in terms of spreadsheets. However, not all spreadsheets are created equal. For example, it is not uncommon to see engineers and scientists store spreadsheets which look like the following:

[insert example of a complex spreadsheet with each column containing a different observation]

Tidy data principles provides a consistent template for functions operating on data. Essentially, each row represents a single unique observations. Each column represents a single unique variable. Each cell indicates a single unique value. In general, this means we favor long, narrow, datasets over short, wide, ones. We can convert the previous example into

[insert example of tidy data]

Not all data begins as tidy data. However, throughout the term, we will create analytic datasets which adhere to these principles.

Overview of the Grammar of Graphics:
The grammar of graphics is a way of conceptualizing how graphics are constructed. We can think of a graphic as being built up in layers. As an example, consider the following graphic:

[insert graphic]

We can dissect this graphic into the following components:
- There is a background
- There are scales along the axes
- There are aesthetics which correspond to additional information
- There are various methods for presenting/summarizing the data
- There are labels for communicating additional information

The grammar provides a consistent interface for discussing these components. We can then use tools which build on this grammar to improve our programming.

Benefits of Vectorization:
Put simply, *vectorization* is the idea of functions operating on entire vectors, instead of on a scalar. Consider a function that grabs the first word of a sentence. If we had several sentences, we could think about looping over these elements to apply the function to each component. Vectorization essentially writes functions to take into account R's built-in component-wise operations so that the loop is implicit instead of explicit. There are two benefits to vectorization:
- Code becomes more readable: we apply operations to entire variables instead of seeing a bunch of nested loops.
- We gain speed within R. The loop is unavoidable, but vectorization exports that loop from R to C, which handles it much faster.

The workhorse of R is the vector; the more your code operates on vectors, the cleaner and more efficient it will be.

Basic String Operations:
We can classify string manipulation into two broad categories: those which are based on strict positioning, and those which are based on patterns. The first is what we will call "basic string operations." These consist of operations like string concatenation (pasting two strings together), turning an entire string to lowercase letters, or subsetting a string. Notice that each of these ideas has explicit rules.

There is a whole suite of functions which perform basic string operations, and these normally cover the range of what we want to do.

Overview of Regular Expressions:
We can classify string manipulation into two broad categories: those which are based on strict positioning, and those which are based on patterns. In order to perform operations based on patterns, we need a way of describing these patterns; this is where "regular expressions" come in. Regular expressions are essentially a language that allows you to represent character patterns. The classic example is a phone number. Imagine you wanted to search through a transcript and find all the phone numbers. More, the person doing the work has no concept of how to recognize a US phone number. How would you tell them what to look for? You might say, phone numbers are 10 digits (when you include the area code). The first three can be in parentheses, and they tend to come in 3, 3, 4. That is, you are looking for things like:
- (###) ###-####
- ###-###-####

Module 5: String Manipulation
Character data brings with it challenges that numeric data does not. We will introduce some of these challenges as well as common base operations with strings. In addition, we will touch on the use of regular expressions for manipulating character data.

Videos:
1. Basic string operations
2. Overview of regular expressions

Optional Reading:
1. E-book on String manipulation

Objectives	***Reading***	***Activities***	***Assessments***
Given a string, **subset** some portion of the string. (Supports Course Objectives A, B, C)	Ch 14	Guided Notes	Homework Assignment, Portfolio Problem
Given two strings, **concatenate** the strings together. (A, B, C)	Ch 14	Guided Notes	Homework Assignment
Given a simple regular expression, **state** what will be matched. (A, B, C)	Ch 14	Guided Notes	Homework Assignment
Given a string and a pattern, **detect**, **extract**, or **replace** the pattern with alternative text. (A, B, C)	Ch 14	Guided Notes	Homework Assignment, Portfolio Problem
Given a string and a pattern, **split** the string based on the pattern. (A, B, C)	Ch 14	Guided Notes	Homework Assignment, Portfolio Problem
Create a *factor* to represent a categorical variable. (A, B, C)	Ch 15		
Identify key string operations to move from raw data to usable data. (A, B, C, D)		Text Analysis	

Text Analysis:
Students will search online for some text analysis that was constructed (not necessarily in R). They will present the analysis they examined and discuss one step that "must have" been taken at some point during the analysis and link it to a function/operation discussed in the lecture material.

Portfolio Problem:
Given the Spotify dataset, clean up punctuation or something in song titles and then determine number of words? We could also look at number of artists attached to each track?

Programming Project: (Deadly Driving)
Is August 2nd really the deadliest day of the year (as reported on Twitter) for car accidents? Using data from the National Highway Traffic Safety Administration (NHTSA), we can examine all accidents across the US on each day of the year over a span of several years.

This will require stacking data, making group summaries, and making an involved graphic (or set of graphics) to adequately address the question of interest.

If we consider the type of road on which accidents occur, we could have some simple string manipulation.

All datasets are available from NHTSA.gov:
- https://www.nhtsa.gov/node/97996/251 (directory of potential files)
- https://www.nhtsa.gov/filebrowser/download/176821 (online documentation)
- Focus on Accident reports, but could include some of the other files.

Module 6: Data Input / Output
We have discussed reading in tabular data. Not all data is tabular. We will introduce a method of reading in unstructured data. Primarily, we will discuss web-scraping, obtaining data from html/xml sources through CSS selectors.

Videos:
1. Big idea of unstructured data.
2. Anatomy of a web page.
3. Overview of CSS Selectors

how to recognize a US phone number. How would you tell them what to look for? You might say, phone numbers are 10 digits (when you include the area code). The first three can be in parentheses, and they tend to come in 3, 3, 4. That is, you are looking for things like:
- (###) ###-####
- ###-###-####
- ###.###.####

Notice the # cannot be a letter; it must be a digit. We want some way of creating such patterns in the computer so that as it scans through the text, it knows that "1st" or "$30,000" are not phone numbers, even though they are digits.

Regular expressions contain short cuts, known as meta-characters, that are placed within a string in order to tell the computer how to form the pattern. For the phone number example, we might do something like the following:

pattern = "\\([:digit:]{3}\\)\\s[:digit:]{3}-[:digit:]{4}"

[walk through this pattern explaining each of the elements]

There is a learning curve to regular expressions, but as we begin to understand them, they allow us to perform operations on patterns, such as finding all sentences which have a phone number, or extracting all the phone numbers from a script, or replacing phone numbers with pound signs in order to redact a document.

Big Idea of Unstructured Data:
We often think of data as having a tabular structure. Hopefully, at this point you have even begun to think in terms of tidy data principles. But, not all data comes in this form. Data may come in the form of images. Even in this case, we can imagine decomposing an image into a commons structure: size, pixels, coloring, etc. Data may come in the form of social media posts or transcripts. In these cases, we have a long stream of output with no particular structure.

Our basic way of addressing unstructured data is to treat it like text, identify patterns, and extract relevant components. As we extract information, we add structure for further analysis. As an example, consider extracting information from a series of emails. For each email, we can extract the subject line, the sender, the receiver, the body of the text, the signature. We can then use this data to further analyze the sentiment of the email.

Anatomy of a Webpage:
Scraping data from a website can seem overwhelming. But, websites have a common DNA, and if we can understand that DNA, we can leverage that knowledge when extracting information. Consider the following simple site:

[insert simple page]

As we peel back the layers of how this page is constructed, we see that it is basic text surrounded by "tags" which tells the browser how to interpret the page and provide the layout we see on the front end. Some tags are simply for organization; others help determine style and layout. The style is further defined by additional attributes, such as ID and CLASS.

Overview of CSS Selectors:
Learning CSS for styling websites provides a lot of flexibility when doing web development. For our purposes, the more we know, the easier it is to extract information from a website. However, it turns out most things can be extracted using a handful of rules (or selectors):
- Tag names for extracting elements of that type.
- Class selector (.) to grab elements of a specific class.
- ID selector (#) to grab elements with a specific ID.
- We can layer (or cascade) these elements as well.

[insert an example with previous page]

Overview of When Interactivity is Good:
Just because you *can* does not mean you *should*. That applies to interactivity in graphics as well. The key to creating a good visualization is that you do not waste resources. Every element should serve a purpose. If you will have interactivity, it should provide something that a basic graphic could not.

If your research question is about the age of a Starbucks, then plotting the location of Starbucks across the country is not helpful. If your question is about how Starbucks are located with respect to downtown areas, then visualizing the location of Starbucks on a map makes sense because the background map orients the viewers and provides an additional layer of information.

Brushing over points which provide the latitude and longitude of the point on a map provides no

Videos:

1. Big idea of unstructured data.
2. Anatomy of a web page.
3. Overview of CSS Selectors

Objectives	*Reading*	*Activities*	*Assessments*
Read a raw text file. (Supports Course Objectives A, B, C)	readLines() documentation	Guided Notes	Homework Assignment
Identify a CSS selector for isolating an element on a webpage. (A, B, C)	CSS Diner tutorial	Guided Notes, Interactive Demo	Homework Assignment, Portfolio Problem
Use a web-scraping package to **isolate** an element on a webpage via CSS selectors. (A, B, C)	rvest tutorial, httr quickstart guide	Guided Notes, Interactive Demo	Homework Assignment, Portfolio Problem
Save and **load** objects in statistical software. (A, B, C)	save() documentation, load() documentation	Guided Notes	
Read a JSON data file. (A, B, C)	jsonlite vignette	Guided Notes	Homework Assignment
Read package vignettes/tutorials. (G)		Guided Notes	

Interactive Demo:
Using the Rose-Hulman webpage as an example, we will go through the process of scraping key information from faculty bio sketches. Students will help develop the plan by locating key pieces of information at each step.

Portfolio Problem:
For the Spotify dataset, students can scrape artist records in order to obtain relevant information. Summarize the age of artists and number of years a group has been around.

Module 7: Dynamic Graphics

Some graphical summaries are improved by dynamic or interactive elements. This includes maps, time-laps motion, and interactivity (hovering, etc.). We will discuss some common functionality and introduce plotly as a tool for implementation.

Videos:

1. Hans video on GapMinder
2. Overview of when interactivity is good.

Objectives	*Reading*	*Activities*	*Assessments*
Given locations, **geocode** those locations. (Support Course Objectives A, B, C)	tidygeocoder vignette	Guided Notes	Homework Assignment
Given location data, **create** an appropriate visualization using a map background. (A, B, C)	ggmap overview	Interactive Demo, Guided Notes	Homework Assignment, Portfolio Problem
Create an animation which updates at each time step. (A, B, C)	Plotly for R (Ch 14)	Interactive Demo, Guided Notes	Homework Assignment
Create a visualization which has hover-interactivity. (A, B, C)	Plotly for R (Ch 1-4)	Guided Notes	Portfolio Problem

Interactive Demo:
Show the spread of Starbucks over time, having students first sketch out what they think the visualization should do, and then building it up in stages.

Portfolio Problem:
For the Spotify dataset, show the location of the top 100 artists in the US. Try to make some type of statement about areas of the country tend to produce major stars. The graphic should have hover functionality.

Module 8: Simulations

Numerical simulations can be used to investigate complex processes as well as evaluate the performance of various statistical methods. These are particularly useful when we have a firm model for the underlying components of the process and how these components fit together. We introduce the

[illegible] respect to downtown areas, then visualizing the location of Starbucks on a map makes sense because the background map orients the viewers and provides an additional layer of information.

Brushing over points which provide the latitude and longitude of the point on a map provides no additional information, even if it is cool that there is some level of interactivity. Hovering over a point should provide a layer of information we would not otherwise get from the graphic. Similarly, animating the order points appear on a graphic should not be done because it can be (think about annoying animations in a PowerPoint presentation). If points appear in some order, that order should convey information (for example, changes over time).

Adding interactivity is about adding layers to a graphic. Those layers, like our base layers should convey information.

Simulation Study as a Statistical Analysis:
The key to conducting a good simulation study is to think about a good data analysis. When it comes to designing a good study, we think of the following elements:

- Replication: in a simulation study, sample size is relatively cheap; so, we can have very large studies.
- Randomization: random sampling helps to reduce bias; in a simulation, that means generating variables at random from a probability model. Random assignment allows for causal conclusions; in a simulation, that means that we do not confound the comparisons being made with the way the data is generated.
- Comparative Groups: we want our treatment groups to be similar; a popular way of addressing this is blocking. We can employ blocking in simulation studies also.

The way that you design a simulation study changes how we analyze it (think ANOVA versus repeated measures ANOVA or RCBD). Just like with scientific studies, we often have small pilot studies as a proof of concept of how the study will work. Similarly, we might want to run 3 replications to make sure the code works before increasing the number of replications to 10 thousand.

When we go to summarize results, we summarize them in the same way that we would summarize data from a study. We have access to visualizations for comparing groups, examining distributions, etc. We can do the same with the results of a simulation study.

steps in conducting a simulation study and tools for generating random variates.

Videos:
1. Simulation study as a statistical analysis

Objectives	*Reading*	*Activities*	*Assessments*
Generate a random variate from a common distribution. (Supports Course Objectives A, B, C)	Course Notes	Guided Notes	Homework Assignment, Portfolio Problem
Estimate the value of an integral using simulation techniques. (A, B, C)	Course Notes	Guided Notes	Homework Assignment
Given the description of a random process, **conduct** a numerical simulation of the process and estimate an appropriate parameter characterizing the random process. (A, B, C, D)	Course Notes	Guided Notes, Interactive Demo	Homework Assignment, Portfolio Problem

Interactive Demo:
Comparison of the mean or median in the presence of outliers.

Portfolio Problem:
Consider a random Spotify playlist and the length of time it would take me to get through them. We could also investigate the likelihood of a particular artist being put together in a shuffle mode.

Module 9: Randomization-Based Inference

We have seen the use of numerical simulation for investigating a process or examining the properties of a statistical method. In this module, we discuss the applications of simulations to inference. We discuss bootstrapping and randomization-based hypothesis testing.

Videos:
1. Review of Bootstrapping (223 video?)
2. Review of the Null Distribution (223 video?)

Objectives	*Reading*	*Activities*	*Assessments*
Given a dataset, **implement** the nonparametric resampling bootstrap and **compute** a percentile-based confidence interval for a parameter from an introductory statistical analysis. (Supports Course Objectives A, B, C)	Course Notes	Guided Notes	Homework Assignment
Implement the *Bag of Little Bootstraps* procedure. (A, B, C)	Course Notes		
Given a dataset, **implement** a permutation test in an introductory statistical analysis. (A, B, C)	Course Notes	Guided Notes, Interactive Demo	Homework Assignment, Portfolio Problem

Interactive Demo:
Baseball example from Rossman blog

Portfolio Problem:
Give a unique test statistic for computing a common statistical analysis and ask for a p-value comparing two groups of songs. Also ask for an estimate of the parameters using a Bag of Little Bootstraps procedure.

Note:
Bag of Little Bootstraps should be implemented at some point, but given time constraints, not implemented this year.

Module 10: Choose Your Own Adventure

This course only scratches the surface on the topics introduced, and it excludes a vast number of beautiful topics that are part of the statistical analysis pipeline. We provide space for each student to investigate a topic of their choice related to the statistical analysis pipeline. This could include a topic tangential to one discussed, an extension of methods discussed, or a brand new topic.

Potential Topics:

- Sentiment analysis
- Machine learning
- Cross-validation
- Jackknife
- Interactive documents via Shiny
- Parallelization of code
- Computing at scale (sparklyr)
- Algorithms for pseudo-random number generation
- Optimization

Objectives	*Reading*	*Activities*	*Assessments*
Identify resources that generalize the material covered in class in order to learn new tools for solving a novel computational task. (G)			Portfolio Problem

No Homework Assignment or Activity for this module. More time is expected to be devoted to the completion of the Portfolio Problem.

Portfolio Problem:
In this week's assignment, students are asked to provide the tutorial for learning about their chosen topic. They should include a list of readings for learning about the topic, a summary of big ideas, and a list of 3 exercises (with solutions) assessing fundamental components of their topic. They should also illustrate the use of their topic in addressing a question of interest. However, the material presented here cannot be a direct duplication of existing work. They should apply the work to a novel dataset/context.

Programming Project: (Target Locations and Median Income)
Compare the median income of counties with a Target and counties without. Making this a county-level would allow for a nice map. If we do this zip-code wise, then the Census Data is easy to use. This will involve a lot of web scraping. The visualization could be simple or complex. The final result is a two-sample test comparing the two groups.