**Preview Questions**

1. What is sociospatial data science?
2. How is sociospatial data science different from other forms of data science?
3. What is the goal of this text?
4. Why are the course names deceptive?
5. How should we approach the course?
6. What are the most important practices in sociospatial data science?
7. How should I protect my work?
8. Where can I get help?

**Summary Notes**

Preface

* The goal of the text is to provide a reference for various “soft skills” that contribute to success in computational social science.
* The text tries to help the reader make the transition from casual consumer of information to professional developer of knowledge.
* The names of the courses are deceptive because we are more fundamentally concerned with high quality research methods and the process for conducting research.
* Being a successful researcher requires technical methods and mental habits.
* The habits of mind and method taught in the course are largely applicable across disciplines and methodologies.

Chapter 1 – Introduction

* The goal is to increase students’ abilities to answer social science research questions using the tools that were traditionally used predominantly by computer and information scientists.
* Data science is the use of these tools outside of the social sciences.
* Opinionated analysis development guards against common, preventable human errors in data analysis which are more often the result of poor process, not analyst failings.
* The course focuses on process which is atypical in most research methods courses.
* The three core areas of analysis development are:
  + Reproducibility and auditability 🡪 executable scripts with clearly defined dependencies
  + Accuracy 🡪 modular test code; assertively test data, assumptions, and results
  + Collaborativeness 🡪 version control to track issues; easy communication archiving
* Our ability to judge the accuracy of final research products derived from unclear, undefined, or disorganized processes is limited and therefore our faith in the product should be diminished.
* The end does not justify the means; how results were obtained is just as important as whether or not we obtained the right result.
* Robustness of process over ease of process.
* Data science has four (4) key aspects (i.e., “full stack” approach):
  + Statistics 🡪 mathematical techniques to draw inferences from data (i.e., inferential statistics)
  + Programming 🡪 Using programming languages facilitate reproducibility of work
  + Visualization and communication 🡪 make results accessible and easily understood by a wider audience
  + Substantive knowledge 🡪 working well in groups and integrating academic literature into your own research.
* Monospace typewriter typeface references commands (inputs), results (outputs), functions, filenames, filepaths, and GitHub repositories.
* Italicized typewriter typeface text is meant to be replaced.
* Sans serif typeface refers to areas of the user interface, menu items, and buttons; this text uses typewriter face because of limitations of the publishing software.
* Sans serif or typewriter typeface refer to keyboard keys
* Sans serif typeface with a right facing triangle refer to click through actions
* Required data packages (R packages) are available on GitHub.

*Approaching the Courses*

* Recognizing when you are starting to spin your wheels and taking a break is better than trying to power through the problem.
* Being organized is a key to doing well in the course, not just grasping the content.
* Getting Things Done (GTD) methodology helps to stay organized.
* Three-ring binder can be organized temporally or thematically.
* Thematic organization of binder:
  + Syllabus and reading list
  + Workflow diagrams
  + R function sheets
  + ArcGIS process sheets (SOC 4650/5650 only)
  + LaTeX command sheets (SOC 4930/5050 only)
  + Sample outputs
  + Exercise handouts
  + Lab handouts
  + Problem set handouts
* Course flow:
  + Before class (active reading, lecture prep)
  + During class (entry ticket, active lecture, lab)
  + After class (lab, problem set)
* Read the material at least twice before class and try to replicate the examples.
* Use lecture preps and replication videos to gauge understanding of content before class.
* Focus more on when a task should be done rather than how to do a particular task.
* Lab exercises give experience executing tasks.
* Refer to the replication files for lab exercises only as a last resort.
* Problem sets are meant to evaluate course progress.
* Review replication files for problem sets regardless of how well you believe you did.
* Do not wait to begin an assignment until the night before it is due because you won’t have time to walk away from the assignment for a while if you hit a wall.
* Sample work schedule
  + Monday: class
  + Tuesday: finish lab
  + Wednesday: start problem set
  + Thursday: finish problem set
  + Friday: first reading
  + Saturday: second reading
  + Sunday: lecture prep for next class
* Version control number format: {major release}.{minor release}.{patch number} (e.g., 2.2.1)
* Course version control format: {year}.{development status}.{maintenance changes} (e.g., v18.1.1
* For unincorporated changes format adds a 9000 designation at the end (e.g., v18.1.1.9000)

*Good Enough Research Practices*

* Can dramatically improve dissertation experiences.
* Reproducibility can refer to
  + code that can execute without alteration on a person’s computer
  + research designs that can be replicated by other researchers
  + ability to derive similar results or inferences from identical research designs
* The course is concerned with methods reproducibility:
  + Implement exactly the experimental computation procedures with the same data and tools and obtain the same results.
  + Requires full access to the original data and the steps used to produce final research project.
  + Derived from computer code, documentation, and organizational approach
  + Literate programming includes narrative text to describe the function of the code
* Workflows are sets of processes used to approach a given task.
* Scott Long’s workflow for statistical research:
  + Clean data
  + Analyze data
  + Present results
  + Protect files
* Long’s tasks for each step in the workflow:
  + Plan
  + Organize
  + Document
  + Execute
* Save raw data
* Document data source, download date and time, and provide permanent web link.
* It’s better to use open data formats rather than proprietary data formats.
* Use plain text files (e.g., .txt or .md) for writing and for data.
* Markdown files are plain text files with special characters that can be rendered by other applications (i.e., rich text like styling)
* Use version control to provide timeline of work and the ability to revert back to previous versions (i.e., a chain of evidence)

*Protecting Your Work*

* Data threats include losing files, losing flash drives, hard drive failures, theft of hardware, destruction of hardware.
* Create a sustainable file system with a single course directory.
* Save on flash drive, not a synched directory (e.g., Dropbox, Google Drive, etc.)
* Use dedicated subfolders
  + /Core-Documents {for reference only}
  + /DataLibrary {do not alter data}
  + /DoeAssignments
  + /Lectures {for reference only}
  + /Notes
  + /Readings {course readings outside of the main text}
* camelCase, dash-case, snake\_case
* Sample project organization  
  /projectName

/data

rawData.csv  
rawTracts.shp

/doc

joinedTracts.md  
notebook.Rmd  
notebook.nb.html  
researchLog.md

/maps

map.mxd

/results

joinedTracts.shp  
map.png

/src

script.R

LICENSE.md

projectName.Rproj

README.md

* Bootable backups are mirrored images of your entire hard drive and protect against hard drive failure.
* Incremental backups are copies of a single file and protect against data corruption.
* Cloud backups protect against catastrophic events.
* The most successful backup workflows require next to no effort on your part.
* Create reminders for manual backup workflows.

*Getting Help*

* Before posting questions on Slack, waiting for office hours, or posting questions to external resources, do the following:
  + Check spelling
  + Check course resources
  + Check your process (i.e., requirements of workflow)
    - Check course handouts and lecture slides
    - Check official documentation
  + Check outside resources
    - RStudio cheatsheets, RStudio community forums, LaTeX wikibook, StackExchange
  + Construct a reproducible example (i.e., reprex)
* Questions that reference specific parts of a problem set or the final project are not appropriate for Slack.
* When posing a question include the following:
  + The question
  + A reprex
  + A screenshot
  + What you’ve tried

**Additional Questions of Interest**

1. None

**Preview Questions**

**Summary Notes**

Chapter 6 – Opinionated Tools

* Opinionated software is designed with a set of guiding principles that encourages users to follow a certain process.

Chapter 7 – Markdown

* Markdown is a simple markup language.
* Markup languages give computer program direction for processing particular blocks of text.
* Reliability is emphasized above all else in Markdown.
* Markdown’s syntax is comprised of punctuation characters chosen to look like what they mean.
* Markdown was intended to be used as a format for writing for the web.
* Text files can be formatted as WYSIWYG editors but are not dependent on proprietary applications.
* For research documentation, long-term portability is more important than fine-grain control.
* Markdown file extension is .md but most applications recognize it as plain-text files.
* Organize Markdown documents with headings of varying size and paragraph breaks.
* Markdown has six heading levels (# is largest and ###### is the smallest).
* Leave a blank line between paragraphs to create a break.
* Text wrapped with single asterisk \* is italicized.
* Text wrapped with double asterisks \*\* is bold.
* The greater than symbol > is used for quoting text.
* Text wrapped in single backticks ` is used for in-line quotes.
* Code blocks use triple backticks ```.
* Including ‘r’ after the first set of triple backticks indicates for GitHub that the code is written in R programming language; GitHub applies some syntax to the file making it easier to read.
* To create hyperlinks wrap the text in brackets [] and follow with the URL wrapped in parentheses ().
* Format for embedded images is ![](URL)
* Precede each line with a dash – or asterisk \* to create bulleted lists.
* Precede each line with an appropriate number to create enumerated lists.
* Combine bulleted and enumerated lists to create nested lists.
* There are different flavors of Markdown such as GitHub Markdown.
* Wrapping text with two tildes ~~ ~~ creates strikethrough.
* Use pipes “|” and dashes “-“ to create simple tables.
* Precede each item in a list with brackets separated by a space [\_] to create interactive task lists.
* Use the @ symbol to mention other GitHub users in comments.

Chapter 8 – Basic Git

* Git is a free open source version control system.
* A repository or repo is GitHub is a directory with Git tracking.
* You can use Git to version control projects without uploading them to the internet.
* Clone is making an identical copy of a repository on the local drive.
* Commit is approve changes made to a repository.
* Sync is pushing changed files to GitHub after they are committed.
* Rather than creating multiple copies of a document, prepare a single document and commit changes periodically and take a snapshot that tracks the changes.
* Each commit is accompanied by a message.
* GitHub repositories cannot have spaces in their names.
* Public GitHub repositories typically contain at least three core files:
  + LICENSE file since the data is for public consumption.
  + README file describing the purpose and content of the project.
  + .gitignore file which stops certain types of files from syncing.
* DO NOT store GitHub repositories in an area of your computer that syncs with a cloud service.
* GitHub uses Issues for user interactions.
* GitHub Desktop is not a fully functional desktop version of GitHub.
* The GitHub tutorial describes the logic and sequence behind Git workflow.

Chapter 10 – Slack

* Slack facilitates collaboration which is one of the three core components of analysis developed identified by Hilary Parker.
* Slack workspaces are independent from one another; each requires its own login credentials (username and password).
* Slack workspaces are organized around channels that are topic specific.
* Using the channel name in a message notifies all users.
* Messages can be edited after they’ve been sent.
* Use threaded replies if an answer to a message is longer than a few words.
* Shared messages with the original messages content appear similar to quotes in Markdown.
* Can share messages with a link also.
* Direct message can be accessed only by the participants.
* Send questions about Problem Sets as direct messages rather than through a public channel.
* Pinning posts allow them to be easily accessed later.
* In-Line code quotes are included in a sentence and are wrapped in single backticks (`).
* Code blocks use triple backticks (```).
* Do not include the name of the language as in Markdown because Slack doesn’t recognize it.
* Use the code or text snippet to add a longer block of code to a message.