Topic Summary of SPOSM lecture material (2022)

1. Course Introduction and Schedule:

- Introduction to the course and instructor (Joachim Gassen).

- Overview of the course schedule with various sessions and topics.

2. Open Science:

- Definition of Open Science as a practice involving freely available research data, lab notes, and other research processes.

- Emphasis on the importance of data and code repositories for result reproducibility.

3. Fundamentals for Open Science:

- Addressing challenges in result reproducibility.

- Considerations for practicing Open Science, including environmental setup, software licensing, data availability, and code readability.

4. Development Environment and Project Organization:

- Components needed for setting up a development environment.

- Usage of Docker for a containerized development environment.

- Project organization, including naming conventions and the use of Makefiles.

5. Git and GitHub:

- Introduction to using Git and GitHub for version control and collaborative development.

- Creating repositories, commits, branches, and merging code changes.

6. Overview of Statistical Programming Languages:

- Comparison of Julia, Python, R, and Stata as statistical programming languages.

- Evaluation based on criteria like user base, features, learnability, and licensing.

7. Functional vs. Object-Oriented Programming:

- Explanation of differences between functional and object-oriented programming.

- Application of functional programming principles in R.

8. Readable and Reusable Code:

- Use of style guidelines and linting tools.

- Writing clear and understandable code.

- Using functions for reusable code snippets.

- Emphasis on short functions and minimal indentation for better readability.

9. Code Error Checking:

- Importance of error checking in functions.

- Example of a function with unexpected input.

10. Recursion and Object-Oriented Programming:

- Demonstration of recursion in a function (Towers of Hanoi).

- Introduction to object-oriented programming and a comparison with functional programming.

11. Writing Code for Open Science Projects:

- Significance of readable code for Open Science.

- Application of discussed principles to the code.

12. Summary and Outlook:

- Concluding remarks and potential discussion on covered topics.

13. Debugging Tools:

- Principles of debugging, including Google error messages, bug repeatability, bug location, fixing, and testing.

14. Corner Cases:

- Example of a function and its potential issues with small samples.

15. Code Meets Data:

- Examples of reading CSV files and using ggplot for data visualization.

16. Relational Databases and Normalized Data:

- Introduction to relational databases, keys, and different types of joins.

- Case study on normalization with a participant and project data example.

17. Using External RDBMS for Storage:

- Advantages of using external database solutions.

- Introduction to SQLite and data transfer.

18. Data Wrangling and Visualization Fundamentals:

- Overview of data wrangling and cleaning.

- Examples of flawed variable definitions and their impact.

- Principles of data visualization for exploration and presentation.

19. Assignments and Wrap-up:

- Information on assignments, coding sprints, and group projects.

- Announcement of the next physical meeting and upcoming topics.

20. Tidy Data Scraping:

- Practical examples of web scraping using the rvest package.

- Scraping HTML tables, extracting URLs, handling ill-structured tables, and dealing with dynamically generated content.