The document provides guidelines for a final project in a course on Scientific Computing and Programming. Here's a summary of the main points:

**Final Project Guidelines:**

* The project offers a chance to apply programming and scientific computing skills learned in the course.
* Teamwork is permitted, with teams comprising up to 3 students.
* If working in a team, notify the instructor ASAP for Canvas group creation.
* Projects can use the R language or any other tools.

**Options for the Project:**

1. **Convex Optimization in R**:
   * Use CVXR package in R for convex optimization problems.
   * Reproduce an example from a paper and study two or more applications using the CVX solver.
   * Some suggested problems include: sizing of clock meshes, designing a segmented cantilever beam, and others.
2. **Build a Shiny App**:
   * Develop an interactive web app using Shiny in R.
   * Possible apps can illustrate root-finding methods, numerical integration methods, or any other scientific computing, data science, or engineering topic not covered in class.
3. **Proposal**:
   * Submit a one-page proposal explaining how you plan to use programming tools for a research problem.
   * You can incorporate various tools and packages as needed.
   * Examples of past projects are provided, like simplifying a SUMO map using k-means clustering, interactive web app on quantum computing concepts, etc.

**Deliverables:**

* **Project Report**:
  + Submit a cohesive project report showcasing the learned techniques.
  + The report should detail preliminary steps, methodologies used, code testing, web-scrapping, etc.
  + The write-up should be a minimum of 4000 words and can be written in RMarkdown.
  + Supporting documentation and code should be available, preferably on GitHub.
* **Submission**:
  + Submit all supporting materials as a .zip file.
  + The final report and documentation are due by 11:59 PM EST on the last day of classes.
  + Reports should be uploaded to Canvas.
* **Project Presentation**:
  + This accounts for 10% of the final project grade.
  + Present your findings in a short presentation of 5-7 minutes.
  + The presentation should provide a summary of the main findings and a live demo of the code/application.
  + Feedback will be given during the last two class meetings.

**Option 2: Build a Shiny App**

1. **Calculator Interface**: An RShiny app that serves as a comprehensive calculator, making use of R's calculating environment. Users can input expressions, vectors, matrices, etc., and get instant results.
2. **Numerical Methods Visualizer**: An app that visually demonstrates the working of methods like the Newton-Raphson method, Secant method, Trapezoidal rule, etc. Users can input their function and initial parameters, then see step-by-step visual progression of the method.
3. **Optimization Toolkit**: Allow users to input a function, select an optimization method (like Golden-section or Newton’s method for optimization), and get the optimized value. The toolkit can visualize the optimization process.
4. **Data Wrangling & Visualization**: Users can upload a dataset and then utilize various tidyverse functions to manipulate the data and visualize it using different plots.

**Option 3: Proposal**

1. **Advanced Data Wrangling**: Dive deeper into the tidyverse, maybe exploring newer packages or lesser-known functions. Create a guide or set of tools that streamline common data wrangling tasks.
2. **Numerical Methods Deep Dive**: Choose one of the numerical methods (like the Newton-Raphson method or Simpson’s rule) and explore its intricacies. Can it be optimized further? Are there scenarios where it performs exceptionally well or poorly?
3. **Parallel Processing in R**: Given that you've studied loops and parallel processing, propose a project that makes use of R's parallel processing capabilities to speed up a computationally intensive task.
4. **Tidy Data & Non-Standard Evaluation**: Non-standard evaluation (NSE) can be tricky. Propose a project that delves into the intricacies of NSE, providing clearer documentation or tools to help R users navigate it.

**1. Define the App Structure:**

**UI (User Interface)**:

* **Data Upload**: Include a file input where users can upload their dataset (preferably in .csv or .xlsx format).
* **Data Preview**: Display the first few rows of the dataset so users know it has been loaded correctly.
* **Data Wrangling Options**: Create a set of options (dropdowns, checkboxes, etc.) where users can select which tidyverse functions to apply (e.g., filter, select, mutate, arrange).
* **Visualization Options**: Allow users to select a plot type (e.g., scatter plot, histogram, bar chart) and the variables they want to visualize.

**Server**:

* Process the uploaded data and make it available for manipulation.
* Apply the selected tidyverse functions to the dataset.
* Generate and display the selected visualization.

**2. Develop the App:**

* **Initialize**: Set up a new RShiny app and load necessary packages (**shiny**, **tidyverse**, and any other required packages).
* **UI Development**: Using **fluidPage**, **sidebarLayout**, and other Shiny UI functions, set up the structure of the app.
* **Server Logic**: Implement the logic to handle data uploads, data wrangling, and plotting. Use **reactive** functions to make the app responsive to user inputs.

**3. Implement Data Wrangling Options:**

For each tidyverse function you want to include:

* **UI Element**: Add a UI element (like a dropdown or checkbox) for the function.
* **Server Logic**: Implement logic in the server to apply the function to the dataset based on user input.

**4. Implement Visualization:**

* Allow users to select a type of plot and variables to plot.
* Dynamically generate the plot based on user selections.

**5. Test:**

* Regularly test your app to ensure each functionality works as intended.
* Consider edge cases, like what happens if a user tries to apply a function that isn't applicable to the uploaded data type.

**6. Polish and Enhance:**

* **Styling**: Consider adding some CSS to make your app look polished.
* **Feedback**: Once the basic functionalities are implemented, share the app with peers or potential users to gather feedback. This can guide further improvements or additional features.

**7. Documentation:**

* Include a brief guide or help section within the app to guide users on how to use it effectively.

**Final Thoughts:**

Starting with a simple version of the app is key. Get the basics working first – uploading data, a simple manipulation, and one type of plot. Once you have this foundation, you can incrementally add more features and refinements. Good luck with your project!

**Reminder Prompt for Future Session:**

**Project Context:** We've been developing an RShiny app for versatile data analysis and visualization. The app has been set up to handle multiple dataset file types (e.g., .csv, .xlsx, .tsv, .rds, .sav), and we've incorporated initial data wrangling capabilities using the **tidyverse** package.

**Latest Features Added:**

* Implemented data wrangling options in the UI: filtering rows, selecting columns, and adding calculated columns.
* Incorporated an "Exit" button that allows users to safely shut down the app.

**Next Steps:**

1. Dive deeper into advanced data wrangling capabilities.
2. Implement dynamic data visualization options, allowing users to select plot types and input variables.
3. Consider implementing user-driven error handling and notifications for better UX.
4. Think about adding UI styling and themes for a more polished look.

**Objective for Today:** Further development of the RShiny app, focusing on [specific aspects, e.g., "dynamic data visualization"] and refining the user experience based on testing feedback.

**Proposal for Developing an Interactive Data Analysis and Visualization RShiny App in 5 Weeks Using Open-Source Tools**

**To:** [Professor's Name]  
**From:** [Your Name]  
**Date:** [Insert Date]  
**Subject:** Proposal for a 5-Week RShiny Data Analysis App Development Project Using Free Resources

**Introduction:**  
The contemporary digital landscape hinges on swift data processing, analysis, and visualization. I propose building an RShiny application, exclusively utilizing open-source packages, to provide users with a cost-effective and efficient platform for data wrangling and visualization through the tidyverse suite.

**Objectives:**

1. Design a user-friendly interface facilitating data uploading, preview, manipulation, and visualization.
2. Incorporate key tidyverse functions for versatile data wrangling.
3. Enable dynamic visualization generation based on user selections.

**Proposed App Structure:**

1. **User Interface (UI):**
   * **Data Upload:** Facilitate the uploading of datasets in prevalent .csv or .xlsx formats.
   * **Data Preview:** Offer a quick data snapshot post-upload.
   * **Data Wrangling Options:** User-selectable tidyverse functions such as filter, select, and mutate.
   * **Visualization Options:** Customize plot types and variable visuals per user preferences.
2. **Server:**
   * Expedite and streamline data upload processing.
   * Implement user-selected tidyverse functions.
   * Render user-driven visualizations.

**5-Week Development Plan Using Free Tools:**

1. **Week 1:**
   * Initialization: Forge the basic RShiny app framework, focusing on core open-source packages.
   * Commence Basic UI Development: Design initial layout with fluidPage, sidebarLayout, and other Shiny UI tools.
2. **Week 2:**
   * Conclude UI Development: Incorporate user interaction components.
   * Initiate Server Logic: Establish foundational data handling and manipulation.
3. **Week 3:**
   * Perfect Server Logic: Focus on smooth data manipulation and preliminary visualization.
   * Embark on Data Wrangling Implementation: Integrate primary tidyverse functions.
4. **Week 4:**
   * Conclude Data Wrangling Implementation.
   * Launch Visualization Implementation: Facilitate basic user-defined visualizations.
5. **Week 5:**
   * Finalize Visualization Implementation.
   * Testing: Iron out bugs, address edge cases, and ensure error resilience.
   * Quick Polish & Documentation: Optimize aesthetics and integrate a brief user guide.

**Benefits:**

1. **Educational Asset:** This app, rooted in open-source resources, becomes an excellent educational tool, marrying theory with interactive practice.
2. **Research Facilitator:** A robust platform for preliminary data dives and visual explorations in research projects.
3. **Economic & Adaptable:** Built entirely on free tools, this app is not only cost-effective but also modifiable by the wider community.

**Conclusion:**  
An efficient data tool built on free resources can stand shoulder-to-shoulder with premium counterparts in terms of utility. The proposed 5-week timeline is aggressive but achievable with focused development on core functionalities. Your expertise and feedback will be pivotal in steering this project towards success.

Eagerly awaiting your insights and hoping for an affirmative response.

Best wishes,  
[Your Name]