

For the project, you can refer to this pdf, the slides and ask the TA and the professor on Slack for more explanations.

This pdf is a summary of what was presented in class the 21st of November.

Summary for the project

The project encompasses all the different lectures during the semester. In particular:

RMarkdown/Quarto:

- **Vignettes** are RMarkdown documents. This is used to showcase your package and shiny app. This serves as your report for the project. You should make at least one vignette for your project but I recommend to make more: one for showcasing the app, one for the package, one where you can explain some technical choices (for example where you show how efficient your implementation is), one for data acquisition and pre-processing. Vignettes are included within your package and are rendered as an article in the Pkgdown website.

Github:

- You should create one **private** repository in our organization for the project. Please use a meaningful name, e.g., `project_group_1` for Group 1.
- The repo should contain your R-pkg. It should be possible to install the package directly from your repo, using for instance the command `remotes::install_github("ptds2024/your_repo", auth_token=...)` (see the help of this function).
- Make a clear README.md file and add indication such as how to install your package and which external packages must be installed.
- The Github repo should reflect on how your team progress with the project. To this end, you should create issues on Github with TODO list. Issues offer also the possibility to involve the professor and TA. Think about making clear commits. Everyone in the team should make commits.
- The repo should be clean (think about editing `.gitignore`). This includes closing closed issues, deleting merged branches, removing unwanted files (such as `.DS_Store`), ...
- Github actions (GA): they are used to check the package and build the website. Unfortunately we have limited resources to run GA on our organization. To avoid this problem, you can make a copy of your repo to one of your team member's personal account to run these GA in order to make sure they work. We are aware of this issue. We have other ways to make the checks and render the websites. Be aware that we will not look at your personal account so, for example, commits there do not count.

Functions:

- We distinguish between internal functions (used within a R-pkg) and exported functions (made available to external users).
- For exported functions, it is important that you write robust functions that checks arguments and return errors/warnings if necessary. Think of writing examples and tests. They should be well-documented and further explained in the vignette. In particular, the Shiny app is run through an exported function.
- Try as much as you can to write your own functions so to minimize the number of dependency on other packages.
- Write efficient functions using principles from OOP and Functional programming. We will look into that aspect. We suggest you make a comparison in a dedicated vignette of the different stages of the conception of a function (comparing for instance different implementation with `microbenchmark`) so we can better appreciate your efforts.

Data:

- Your project will most likely contain data. Different sources for data: Kaggle, World Bank, OECD, Google BigQuery, ...
- An optional solution is to use webscraping. Be sure that scraping is allowed by checking terms and conditions (the name can slightly vary from website to another one) and `robots.txt` file.
- You are welcome to use SQL query language in your project, this is optional.
- Pre-processing steps (raw data, R scripts, ...) should be included in a dedicated folder `data-raw/` and we suggest you make a dedicated vignette to further explain your different choices (removing some columns of missing values, ...).
- All datasets should be saved in the dedicated folder `data/` and well-documented in the R-pkg (refer to the class on R-pkg). Do not forget to cite your sources.

R-package:

- Think of adding tests (with `testthat`) and examples to your package.
- You can perform a more involved check by adding the `--as-cran` option. Make sure there are no errors and no warnings. Notes are acceptable depending on their natures.
- Edit the `.Rbuildignore` file to ignore all unnecessary files/folders for the R-pkg (e.g., `data-raw/` can be ignored).
- You must include a website build with Pkgdown (you can contact the professor or TA if you want to use blogdown or other template website generator). We will assess the clarity and quality of your website.

Shiny app:

- You must produce a shiny app for your project.
- You are free to choose the UI (dashboard, fluid page, ...).
- Think of adding an explanation page in your app.
- We will assess the quality of your app (is it well written, ...), the quality of your UI (how beautiful the UI looks) and the computational efficiency of your app (is using the app a seamless experience or do you have many errors and bugs). Can some computation be avoided? Should you write a shiny module? You can optionally think of describing all your efforts in a vignette to guide our assessment. Sometimes, we do not realize how difficult it was already to obtain the results that you've obtained.

Project:

- Presentation lasts 15 minutes: 10 for presenting your project + 5 for Q&A and showing your Github repo (if time permits). For presentation, you should at least give motivation (why), showcase what you have achieved and how it works (how), and conclude with further development, directions and indicate hypothetically how you are going to maintain your app.
- Put your presentation in a dedicated folder **presentation/**. You are free to choose how to make the presentation (you can consider RMarkdown/Quarto).
- Every member of the group must present during the presentation.
- Everyone must participate to the project. This will be assessed with the Github repository. For example, has everyone made commits to the project? You can consider explain your team organization in a vignette (with goals, milestones, and how the work is split) or make it apparent on Github Projects.
- Order for the group presenting is: 2 6 3 7 5 1 4. Everyone must be present for all presentations.
- Cite your sources of inspirations. Was ChatGPT useful? How? Or maybe another LLM?
- We will assess the complexity and originality of your project. Is it a "simple" copy-paste + some chatGPT of a pre-existing Shiny app? Or you had to develop many functionalities on your own? Think of describing the complexity and originality of your work in a vignette to guide our assessment (and avoid misinterpretation from our part).