**R Software Development**

**Project: R shiny app**

**Instructional Text on the left panel**

* Revise the texts. [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)

**Section: Study Setup**

**Section: Expected Availability**

* Allow user-uploaded csv file for availability over time
  + Implement this feature (refer to a similar functionality for randomization probability)
  + Provide a template csv file for them to download, if they choose to upload
  + Add a check to make sure the uploaded file is of correct format

**Section: Randomization Probability**

* Allow user-uploaded csv file for time-varying randomization probability
  + The current code implemented this. Make sure the current code works.
  + Improve the current template csv file for them to download. The current one always uses 42 days with 5 decision points per day. It would be nice if we can automatically generate a csv file according to their input # days and # decision points per day.
  + Add a check to make sure the uploaded file is of correct format

**Section: Success Probability Null Curve**

* Allow user to specify log-quadratic form.
  + Do the math to decide what are the inputs (sliding bars) from the user. Probably initial value, end value, maximal decision point, maximal value? [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)
  + Implement this feature. Currently there are only “constant” and “log-linear” options.
* The displayed “Notes” that explains the choice selected by the user.
  + Go through and revise all possible notes to make sure the texts are correct. [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)

**Section: Proximal Treatment Effect**

* Allow user to specify log-quadratic form.
  + Do the math to decide what are the inputs (sliding bars) from the user. Probably initial value, end value, maximal decision point, maximal value? [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)
  + Implement this feature. Currently there are only “constant” and “log-linear” options.
* The displayed “Notes” that explains the choice selected by the user.
  + Go through and revise all possible notes to make sure the texts are correct. [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)

**Section: Result calculation and history display**

* Improve the history display
  + Revise the column names of the history table (possibly by writing up a few lines of text at the end?) [@Tianchen Q](file:////ep/profile/1gg9uA1iUgdDmqkTo2qowQL01HKR6QY8CLBzwd7QhwS15HHR9HBe)
  + Adding the lines of text from the above bullet point to the history display panel.
  + Allow user-uploaded settings in history display (e.g., by showing a cell is “user-uploaded”)

**New Features To Implement (open-ended)**

This feature will also be needed in the R package.

* Allow the user to upload a range of settings
  + How to design this feature? We would implement a functionality to help the user to create a factorial design, with a variety of inputs for each of: availability, randomization probability, success probability null curve, proximal treatment effect.
* Calculate the sample size under each setting, and report to the user either via tables or graphs (or both)
  + What are some useful information to include in a table? The smallest/largest required sample size (along with the corresponding setting). What else?
  + What are some useful information to include in a graph? There are many dimensions of input. How to illustrate the result using graphs?
  + Also report a full table (as csv) with the sample size under all the settings for the user to download.

**Project: R package**

* Write wrapper functions so that people can input to the function as argument such as the following: success\_probability\_null\_curve\_shape = “loglinear”, treatment\_effect = “loglinear”.
  + Play with the Shiny app to get a better understanding of what a user would input.
  + I think we would also want to write some checks on the arguments, to make sure the numbers make sense. (e.g., probabilities should always be between 0 and 1)
* Write documentation for each function that will be included in the package.
  + This requires some good understanding of the setting and paper draft. Tianchen can help with this.
  + This might include writing some Examples, such as the help document for other R funtions.
* Write tests such as unit testing (for each function? or only for main functions).
  + Depending on how much of this is discussed in class, this can be optional.
* Wrap it up as an R package and post on CRAN
* Write a vignette
  + Depending on how much of this is discussed in class, this can be optional. I would really love to at least have some simple tutorial to illustrate the use of the package, though.
* See “New Features To Implement” Section under Shiny section. I would also like this feature to be implemented in the R package.
  + We can build our own summary() and plot() functions some particular objects.