

Lab 5

Due on 04/19/2023 at 11:59 pm

Instructions: For this lab assignment you will create a R Shiny app. You are allowed to work on this lab assignment in groups. You still need to submit an individual lab report if you do work in a group, and you need to list your collaborators. Specific details and a basic grading rubric follow.

The app that you are going to create is largely up to you, but it should be closely related to materials that we covered in class. Some interesting ideas are:

- You can create a Shiny app that displays seasonal stats for each player (or full time players) on a user-selected team and year. This app can also display basic team information such as aggregated seasonal scaled variables: $OPSscale = OPS/avgOPS$, $WHIPscale = avgWHIP/WHIP$, and $FPscale = FP/avgFP$.
- You can create a Shiny app that displays career trajectories for user-selected hitters and pitchers. You can add a widget that allows users to change which stats are displayed. This app can also display career values for the statistic that user selects.
- You can create a Shiny app that fits the Pythagorean formula model (after finding the optimal exponent) to the run-differential and win-loss data for a user-selected time period. The app can then display a top 10 list of managers who most overperformed their Pythagorean winning percentage and a top 10 list of managers who most underperformed their Pythagorean winning percentage over the user-selected time period.
- You can create a Shiny app that uses the `simulate_half_inning` function from the course notes to estimate a 2016 or 1968 team-specific RE24 matrix using Markov Chain simulation.
- You can create a Shiny app that displays annual averages of `release_spin_rate`, `effective_speed`, `plate_x`, `plate_z`, `pfx_x`, `pfx_z`, `release_x`, and `release_z`, and computes the annual pitch type percentages for a user-selected pitcher. The graphic should allow one to clearly see how these annual averages change over time. It is best to display all nine plots in a single grid of plots rather than printing off nine separate plots. This can be achieved using the `grid.arrange` function in the `gridExtra` package.
- Something else that was covered in class.
- A pilot version of your potential final project

Basic rubric details:

- 10 points for all files submitted in the correct directory with the correct naming convention.
- 50 points for an app that successfully compiles from the materials in your directory.
- 10 points will be given for doing something interesting. Yes, this is subjective, but you can ask your instructor if what you are doing is interesting well before you submit your app. Any of the ideas mentioned above are interesting enough to earn these points.
- 15 points will be devoted to bugs. Your app should not break, time out, or display generic warnings/messages. Your app should not take too long to run. How long is too long is a bit subjective and app-specific. You can alleviate time concerns by displaying time warnings.
- 15 points will be devoted to presentation. Plots and inputs should have clear labels. If anything is not clear from these labels, then you need to provide additional clarifying text.

Correct submission of a Shiny app that successfully compiles and displays something interesting at your defaults is automatically worth 70 points. Submission of a Shiny app that is not based on course material or closely related to course material will be deducted 50 points automatically. For example, a correct submission of the Old Faithful Shiny app would be worth 40 points (50 point deduction for submitting a Shiny app not related to course material; 10 point deduction for an app that only changes the number of bins in a histogram which is not interesting).