Agenda:

* ~~Just something to think about, and before asking our clients, how do you guys feel about using AI tools (ChatGPT, GH Copilot) to maximize our progress?~~
  + ~~Pros:~~
    - ~~(potentially) rapid development and iterations~~
    - ~~able to generate code even if we can’t all run the project -> could improve workflow - just brainstorming, but maybe chris could be the chief prompt engineer with andre and i testing, implementing that~~
    - ~~AI probably knows react and js a lot better than we do~~
    - ~~Learning this as a tool is probably the future~~
  + ~~Cons include:~~
    - ~~learning curve~~
    - ~~Not much time for mistakes: if it doesn’t pan out, that time is spent~~
    - ~~potentially more time spent debugging~~
    - ~~may not work ideally with deprecated/unsupported code~~
    - ~~Personal sentiment, but being realistic it’s just a different way to learn this stuff~~
  + ~~Extra Elliott being extra again (hello, fellow kids) - I would prefer not to, but as a means to an end it might enable us to deliver a product that they’d be happy with. I’ve been avoiding ChatGPT and similar AI tools for irrational reasons, but logic suggests this is the way.~~
* ~~Wednesday/Saturday progress/update/check-in/deadlines: maybe one for code reviews and one for development goals?~~
  + Commit often,
  + Aim to push at least weekly? (either wed or sat, flexible)
* ~~Everyone should probably have a task assigned by today (Tuesday)~~
  + Should we change anything?
  + Deadlines for tasks vs. Twice-weekly deadlines
  + Anything to report on your current task?
* ~~How does everyone envision a realistic, final project? I.e. 100% complete? 50% complete? Stretch goals?~~
  + Elliott notes: I know I tend blindly toward optimism, but if it’s not realistic or we aren’t on the same page I’d like to understand. I *am* optimistic and believe we can get the main 4 goals (3x dropdowns, functionally displaying data, and hover text) implemented.
* ~~End of Semester is coming up - consider that in our schedules~~
* ~~add\_graph\_button\_functionality review, ready to pull request for client review~~

Notes:

* “Final exam” - Sprint 4 Demo Day is Final Demo Day
  + Rose Theater during exam time
  + Each team sets up a demo (only provided table)
  + Need to prepare (total length should be <= 10 min):
    - Video - like a commercial, 30-60s
    - Reflection
      * a couple of slides
      * Each team member discuss what they added to the project
      * Suggestions in slides
    - Demo - Live Demo
      * For web apps, kind of like a thinkaloud
  + Will have to upload videos/slides to canvas
* Being able to pull data (from sessions, etc.) and display on the graph
* Need to make all graph dropdowns functional (modify graph accordingly)
* Currently thinking about each dropdown functionality in terms of dot notation, ex (something like…):
  + Data type dropdown: dataset.sessions.teams.persons.typeOfData (set typeOfData = specificTypeOfData)
  + X-axis dropdown: dataset.sessions.teams or dataset.sessions.persons
  + Y-axis dropdown: dataset.sessions.teams.persons.typeOfData (do y-axis things with typeOfData)
* Andre suggests our first week’s goal: hardcoded data for all of our tasks (basically different dropdowns), so probably have something tangible to share for Saturday
* We’ve all got 1 task claimed and will begin working on them.
* Not expecting much to report for our first twice-weekly check-in (Wednesday) but we’ll know more about whether our tasks are properly structured, need to change, or if we need to create different tasks
* Hover text will be completed last, currently unclaimed because it relies on the other 3 major tasks’ completion
* Regarding agenda: discussed the format we’ll use for our weekly deadlines/checki-ns
  + Should expect to push (not necessarily pull request) weekly so we can discuss and see each other’s progress, and provide a status update or actually review the code in the other check-in
* We spent the remainder of time mostly walking through Graph.js paying particular attention to VictoryGraphs and VictoryGraphAxis
* Also discussed the where GraphsContainer.js passes props to Graph, and where Graph sets its states (Graph.js #140-142) specifically mapping x and y
* We’re going to have to approach these tasks as if the data types, axis options won’t change and no new ones will be added, but as a good practice write code that should be easily modifiable for future teams (whenever possible to use parameters and variables in place of hard coding behavior, aim to do that)

Action Items:

* Begin working on tasks
* Next week: should be able to modify graphs for each task using at least hard-coded data; doesn’t necessarily need to work perfectly
* Aim to commit often
* No significant check-in tomorrow, but plan for something substantial on Saturday (progress toward tasks, thoughts on the planned approach and what, if anything, needs to change)