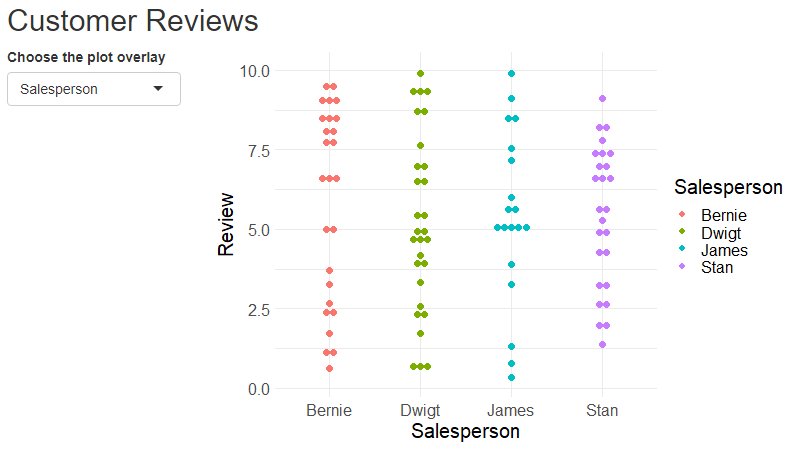
noRth Learnings

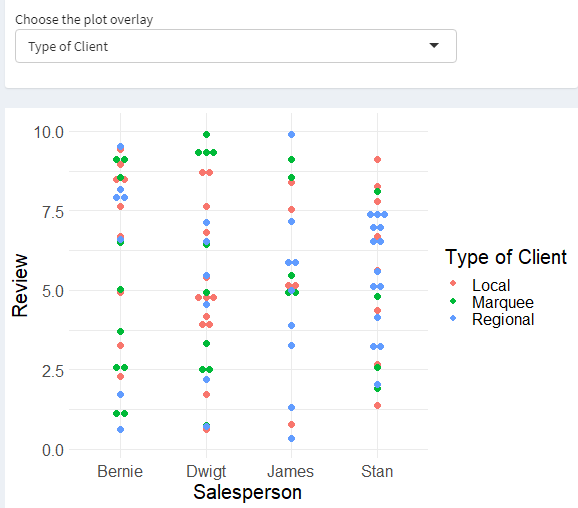
Overview of structure and feedback in **blue**

Structure of tutorial (~ 55 minutes long)

1. Brief overview of what an R application is
2. Explanation of UI, Server, Global
   * **Would have spent more time with an overview of shiny and explaining the functionality of apps. At least at this conference, it seemed that a lot of people wanted to better understand if an R application could work for certain business use cases.**
3. Explanation of some helpful tips when learning R and R shiny
   * Advised them not to worry about the data and it’s ok to learn about the functionality of Shiny with built-in datasets.
   * Provided some resources (Wickham’s book, Rshiny 2016 conference videos, RStudio site)
   * Explained also how shiny has the added element of project management
   * I think a lot of beginners generally ask about where to start – talked about starting with the ui
   * **This section felt ok to me – for me it’s helpful when experienced R users talk about how they learned R. I think learning strategies for new shiny users can be helpful**
4. App 1 : Went through a simple application.
   * I did everything in one file with ui, server, etc on one page.
   * The app had made-up reviews for paper salesmen
   * Some overview
     + Fluid functions (page ,row, column)
     + selectizeInput (this just pulled in static choices)
     + Went over some of the structure of application

(use of commas, how input and output ids work)

* + Covered the overall structure of the function of a server
  + Covered one output (the dot plot)
  + Rendering functions
* **My graphs were no-where as nice as Yu-Ann’s – and I think from the standpoint of our audience this would be helpful if they were more relavent.**
* **It felt okay to start with a simple plot and filter as a starting point. The filters were just static, and I thought that was ok to start with.**
* **I probably would not teach as many concepts within the apps – only because I think it may be helpful for users to see and be aware of some of the following concepts before they look at an application OR I would spend a lot more time on App1 – and then talk about the concepts in a ppt within the context of app1, which I think could be quite helpful.** 
  + **Syntax that’s distinct to shiny programming (commas after elements, ({, () ) .**
  + **I explained input and output ids within the app – I would probably also move this to its own slide prior to the apps.**

1. App 2 : added some complexity to the simple application, mainly structure
   * fileInput (made it so that you had to upload a file)
   * shinydashboard
   * updateSelectizeInput (now this pulled in choices from the data and

helped explain function()

* + introduced the concepts of functions such as req and explained how

shiny doesn’t evaluate code in order as an R script does – passive

* + browser()
  + using observeEvent to make sure a data file was loaded, and req to make sure you had data for the plot.
  + **With req – I showed them how an error would appear for 2 seconds before adding a requirement that a data file existed. This helped me explain imperative programming and I think this was helpful.** 
    - **To that end, I would have also showed a reactive graph (I think this could go either way – if they were familiar with the objects in the application they may not have a difficult time understanding what it is – but it could also be helpful to explain outside of the app).**
  + **I do wonder if adding the concept of shinydashboard should actually be saved until the very end of the tutorial (although if we use our internal functions this might not matter) – but I think adding code that’s not related to the ‘meat’ of what’s happening in rshiny could just add extra code and possibly not help them understand what functions in shiny actually do.**

1. App 3:
   * Explained how ui elements could be written as a function
   * **I’m not really sure this was the best route to go, since a lot of the questions were more business and less tutorial-related, they may be more interested in understanding briefly what might happen after an app is built, or for SIOP we would have more time than north so could go into more complicated concepts. I also just think the idea of adding more to the app is helpful to build complexity.**