Shiny\_ResBazSydney2018 notes

**Lesson 1**

*Hello shiny*

Basic example of reactivity: one input, one output

* User changes the position of the slider, and this alters the binwidth of the histogram
* The user interface updates pretty quickly, gives a seamless experience
* Not all expressions are re-evaluated, only those whose dependent values have changed

UI – user interface

* Input widgets collect information from the user
* Displays outputs, such as plots or tables
* Define how visual elements are laid out
* Written in R syntax, but most functions won’t be familiar

Server

* Where data is processed
* Regular R code combined with special shiny:: functions that implement reactivity
* Reactive expressions are re-evaluated when their dependent values have changed

The sliderInput widget defined in the UI allows the user to set a value, which is retrieved and used to set the binwidth in the Server.

renderPlot is a reactive expression: code wrapped within renderPlot({}) will be re-evaluated any time an input changes

Exercise: identify how the value obtained by sliderInput() in the UI is used by the renderPlot expression in the Server

*Shiny text*