

Shiny App development

- Core
 - `{shinyuieditor}`: Visual tool for organizing Shiny UI
 - `{bslib}`: Custom Bootstrap Sass themes for {shiny} and {rmarkdown}
 - `{shinytest2}`: Unit testing for R Shiny Apps
- Debug
 - `{profvis}`: R Shiny App profiler
 - `{reactlog}`: R Shiny reactivity visualizer
- Performance
 - `shiny::bindCache()` & `{cachem}`: Cache and store any reactive output
 - `{promises}` & `{future}`: Async code execution
- Admin
 - `{shinyloadtest}`: Load testing for R Shiny Apps
- Extra credit
 - `{plumber}`: R web API
 - `{shinymeta}`: Expose Shiny app logic using meta-programming

Shiny performance workflow

From Joe Cheng's Shiny in Production keynote...

1. Use `{shinyloadtest}` to see if it's fast enough
2. If not, use `{profvis}` to see what's making it slow
3. Optimize
 - a. Move work outside of `{shiny}` (very often)
 - b. Make code faster (very often)
 - c. Use caching (sometimes)
 - d. Use `async` (occasionally)
4. Repeat!