SHINY · R · CHEAT SHEET

Shiny module state patterns

Where state lives decides how a Shiny app grows. Two patterns cover most modules; both are tested here with shiny::testServer, the way a module should be tested before it ships.

R Pattern 1: the module returns its state. Tested with testServer.

```
library(shiny)

counter_server <- function(id, step = reactive(1)) {
  moduleServer(id, function(input, output, session) {
    total <- reactiveVal(0)
    observeEvent(input$add, total(total() + step()))
    total    # the caller reads this reactive
  })
}

testServer(counter_server, args = list(step =
reactive(2)), {
  session$setInputs(add = 1)
  session$setInputs(add = 2)
  stopifnot(session$returned() == 4)
  cat("returned after two clicks with step 2:",
session$returned(), "\n")
})
```

Ran when this sheet was built. Output:

```
returned after two clicks with step 2: 4
```

R Pattern 3: per-session shared state through session$userData.

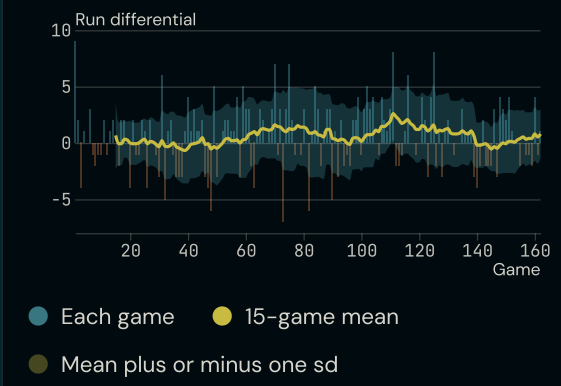
```
library(shiny)

picks_server <- function(id) {
  moduleServer(id, function(input, output, session) {
    if (is.null(session$userData$picks))
session$userData$picks <- reactiveVal(character())
    observeEvent(input$pick,
session$userData$picks(c(session$userData$picks(),
input$pick)))
  })
}

testServer(picks_server, {
  session$setInputs(pick = "FF")
  session$setInputs(pick = "SL")
  cat("shared picks:", session$userData$picks(), "\n")
})
```

Ran when this sheet was built. Output:

```
shared picks: FF SL
```



Run differential game by game (bars), its rolling mean (line) and a band of one rolling sd. Data: Season game log (game_log.csv). Simulated for practice by AthlyticZ (agents/sim_data.py, seed 20261109). License CC BY 4.0.

Return a reactive
The module owns its state and returns it. The caller reads it and passes it on. The default.

Reactive arguments
The caller owns the state and passes a reactive in. Use when several modules read one value.

session$userData
Shared, per-session state any module can reach. Use sparingly, for state that is truly global to a user.

An R6 store
An object with methods and its own reactive trigger. For state with rules, such as undo or validation.

From The No-Look Pass. Every code block on this sheet ran when it was built; the data is simulated for practice.