Creating Custom Interactive Applications with R and Shiny

Chris Runyon, Josh Goodman,
and Marcus Walker

Day 2

Welcome back!

Day 1: Covered the basics of a Shiny app.

Today’s Agenda:

- End of Day 1
- Deployment
- UI Organization, useful Shiny extensions
- App Development Exercise - Unifying Day 1 and New Content
- Shiny App Examples
- Additional Shiny Resources

Day 2 Learning Objectives

- Will be able to identify R packages that significantly extend the functionality of Shiny apps.

- Will have be able to identify some of the necessary considerations for developing a Shiny app.

- Will be able to identify sources for learning more on a wide variety of Shiny-related topics.

Back to Josh!

Deployment - shiny.rstudio.com/deploy/ ; book chapter

- Laptop / Desktop

- Local Server / Shiny Server Open Source

- Shinyapps.io

- RInno / electron

Laptop / Desktop

Advantages

- Fast and Easy
- No additional costs

Disadvantages

- Limited functionality / reach
  - Limited access
  - What data does your app depend on?
  - Security considerations

Local Server - shiny-server

Advantages

- Open Source / Free
- Can be deployed behind firewalls
- Allows access to all individuals that can locate the app
  - Doesn't need to be on their desktop

Disadvantages

- Will involve some IT / developer engagement
  - May be an associated cost
- May still have some of the same data/security issues, depending on how the app is set up

RStudio Connect ($)

Shinyapps.io Deployment

Advantages

- Easy to set up and deploy your app online.
  - Free and lower-cost tiers are relatively inexpensive
  - Additional support / features at higher-cost tiers
- Increased access, no (or little) additional help necessary to host the app.

Disadvantages

- Ongoing cost; long-term considerations.
- Security / data issues may still be present.
  - RStudio Connect can mitigate some of these.

RInno / similar packages (DesktopDeployR ?) (electricShine ?)

Advantages

- Completely stand-alone version of the Shiny app
- Possible use without a connection to the internet

Disadvantages*

- Very fragile
- Size of .exe can be large depending on what packages are also included
- Can be difficult to troubleshoot / debug

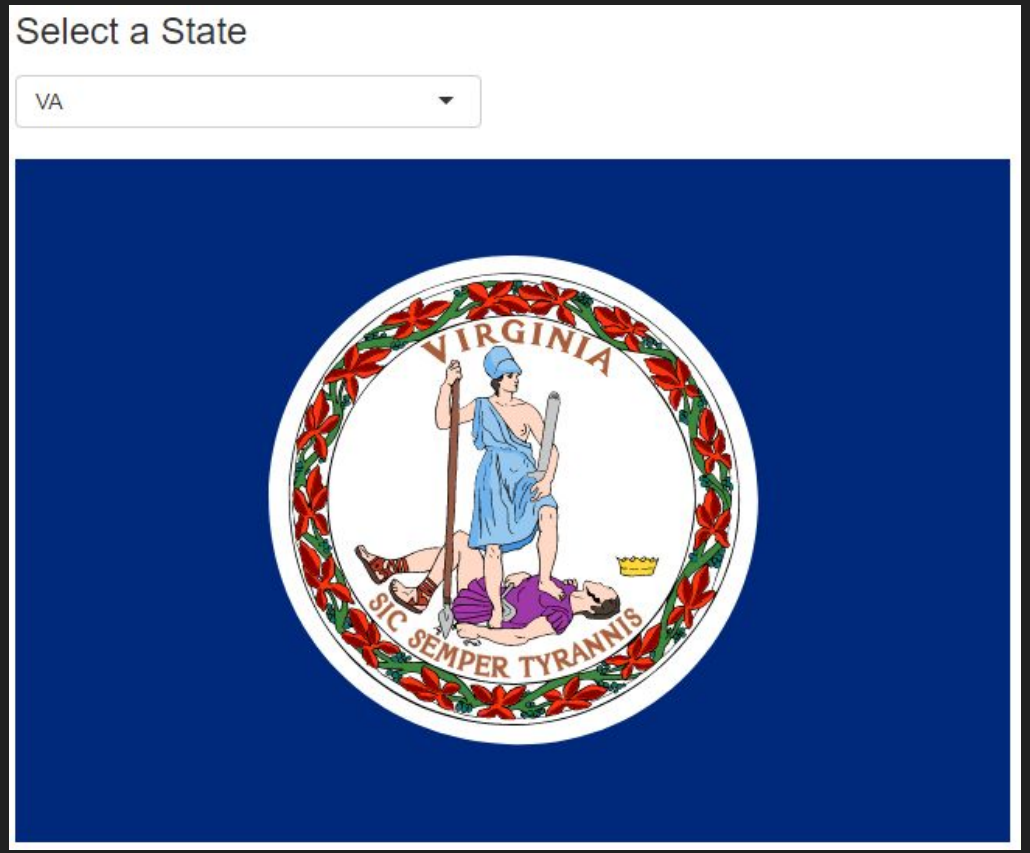
*Disclaimer: Based on when Chris used RInno 3 years ago. We'd love to hear your success stories with this!

More UI Layouts / Organization!

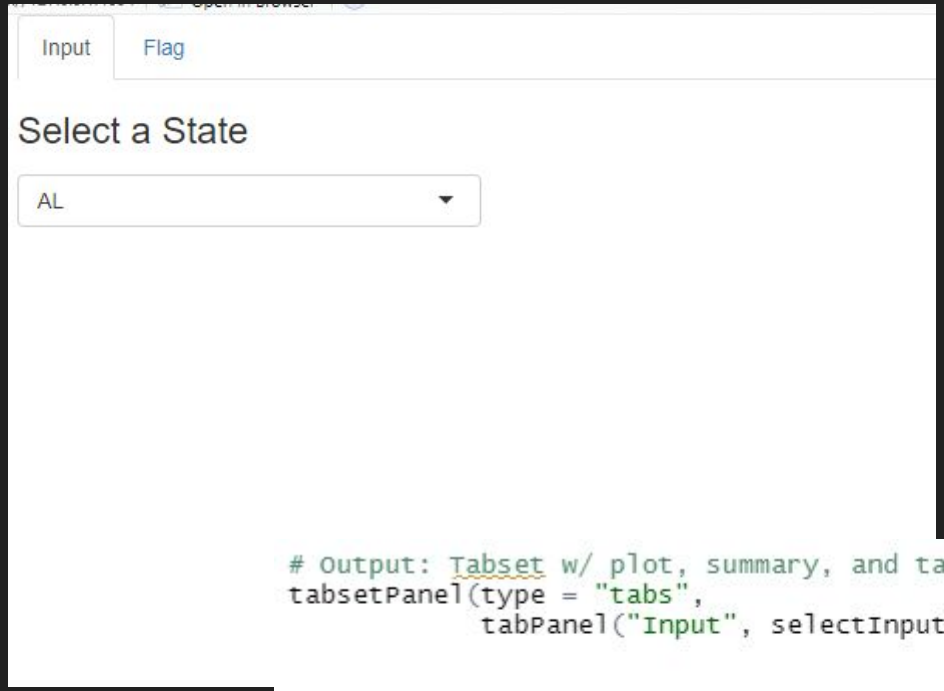
UI Organization (continued)

- fluidPage, fluidRow, etc.
- Multiple displays with tabs
- Dashboards
- Combination of the above

Single Display (PickState.R)

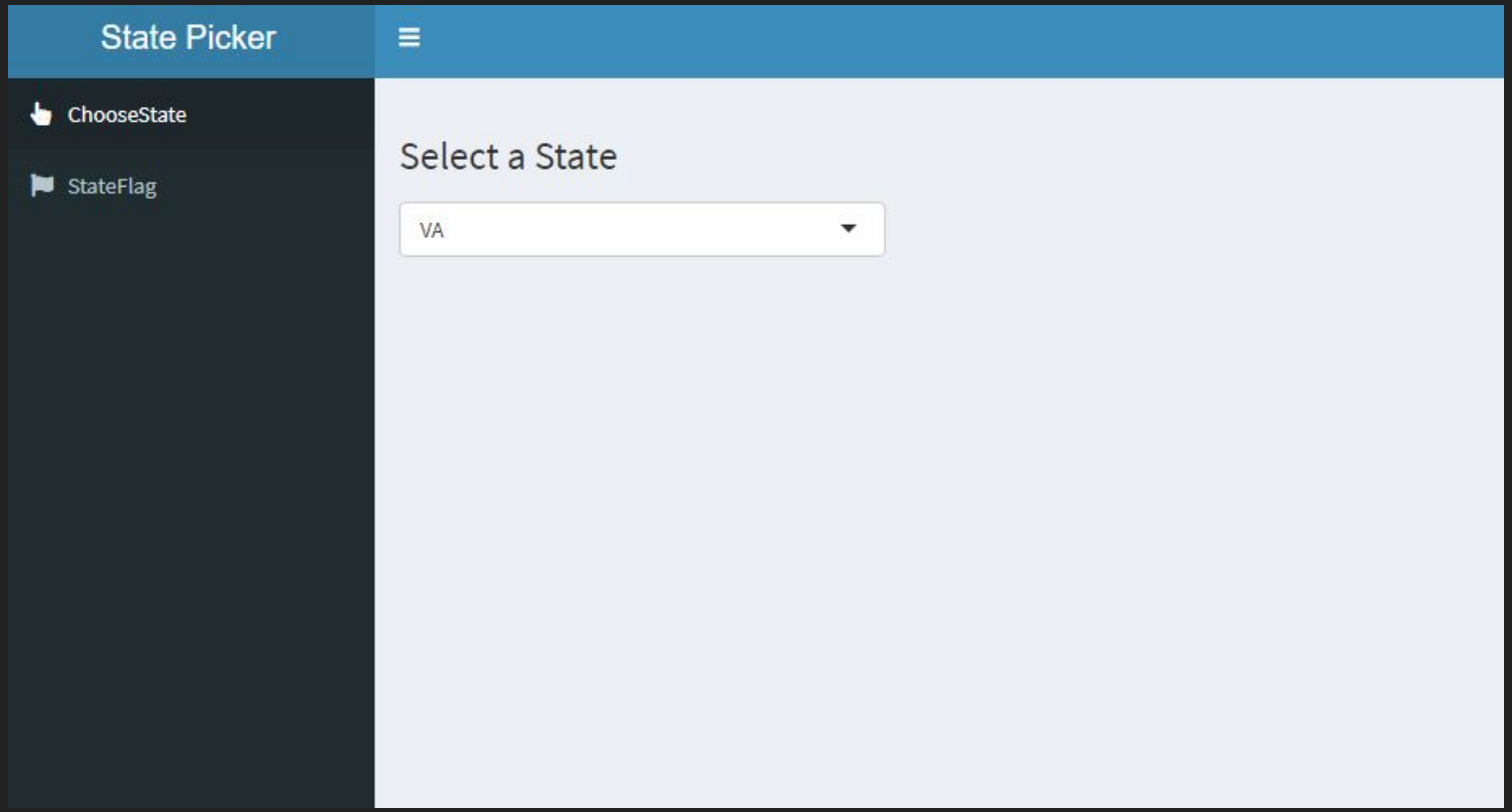


Tab Display (PickStateTab.R)

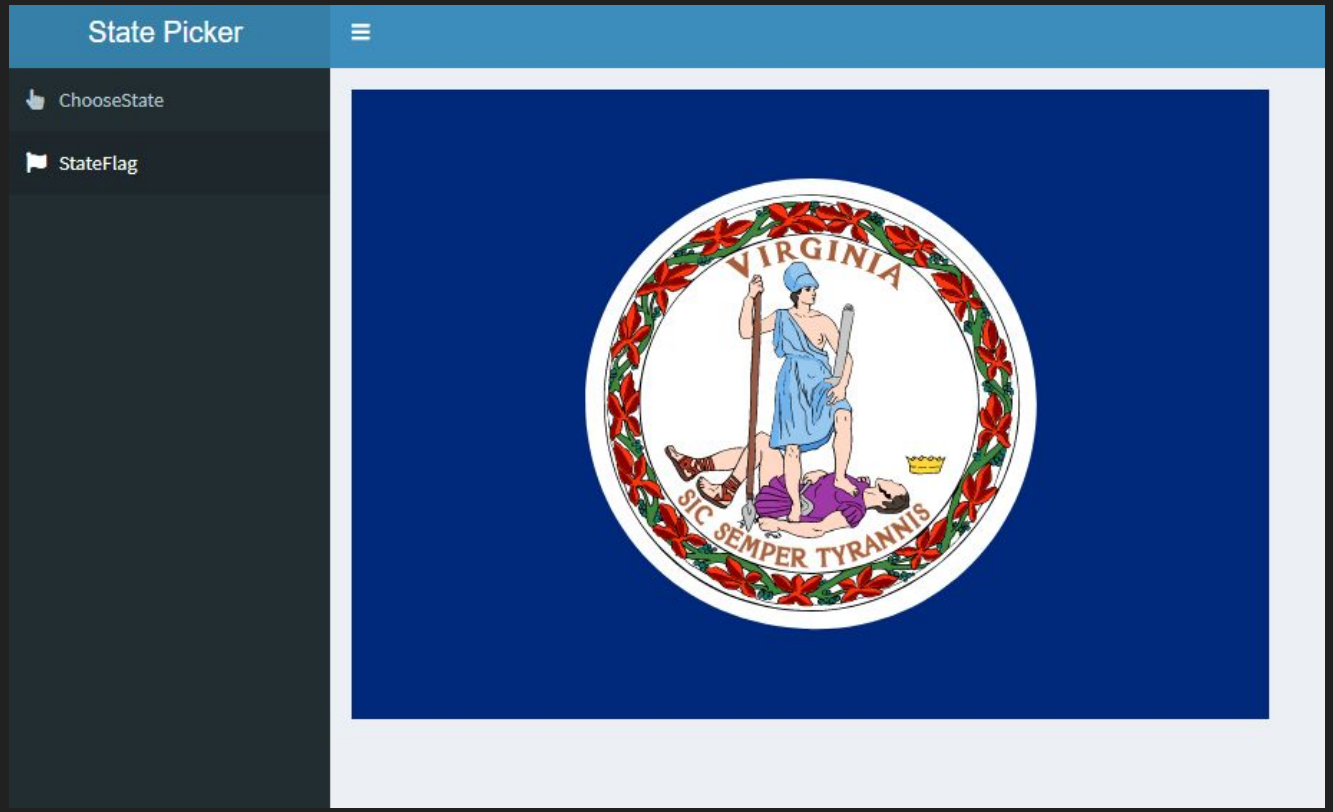



```
# Output: Tabset w/ plot, summary, and table ----
tabsetPanel(type = "tabs",
            tabPanel("Input", selectInput("select", label = h3("Select a State"),
                                          choices = list(state.abb = state.abb),
                                          selected = 1)),
            tabPanel("Flag", imageOutput("state_flag"))
)
```

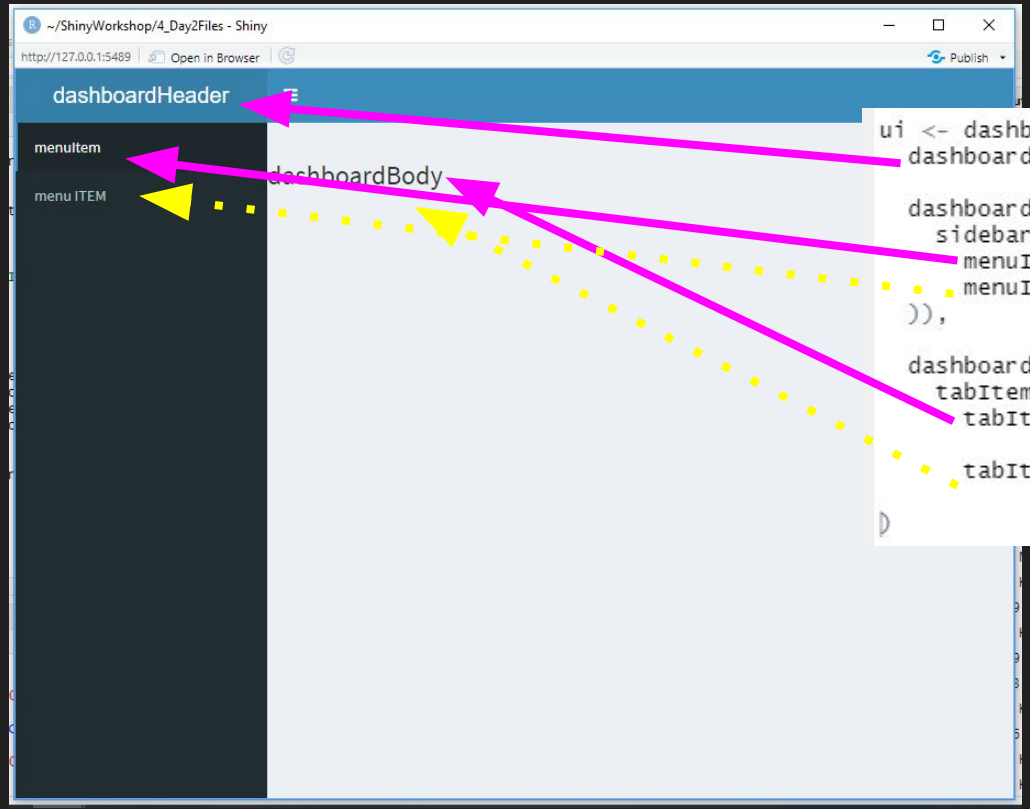
Dashboard Display (PickStateDash.R)



Dashboard Display (PickStateDash.R)

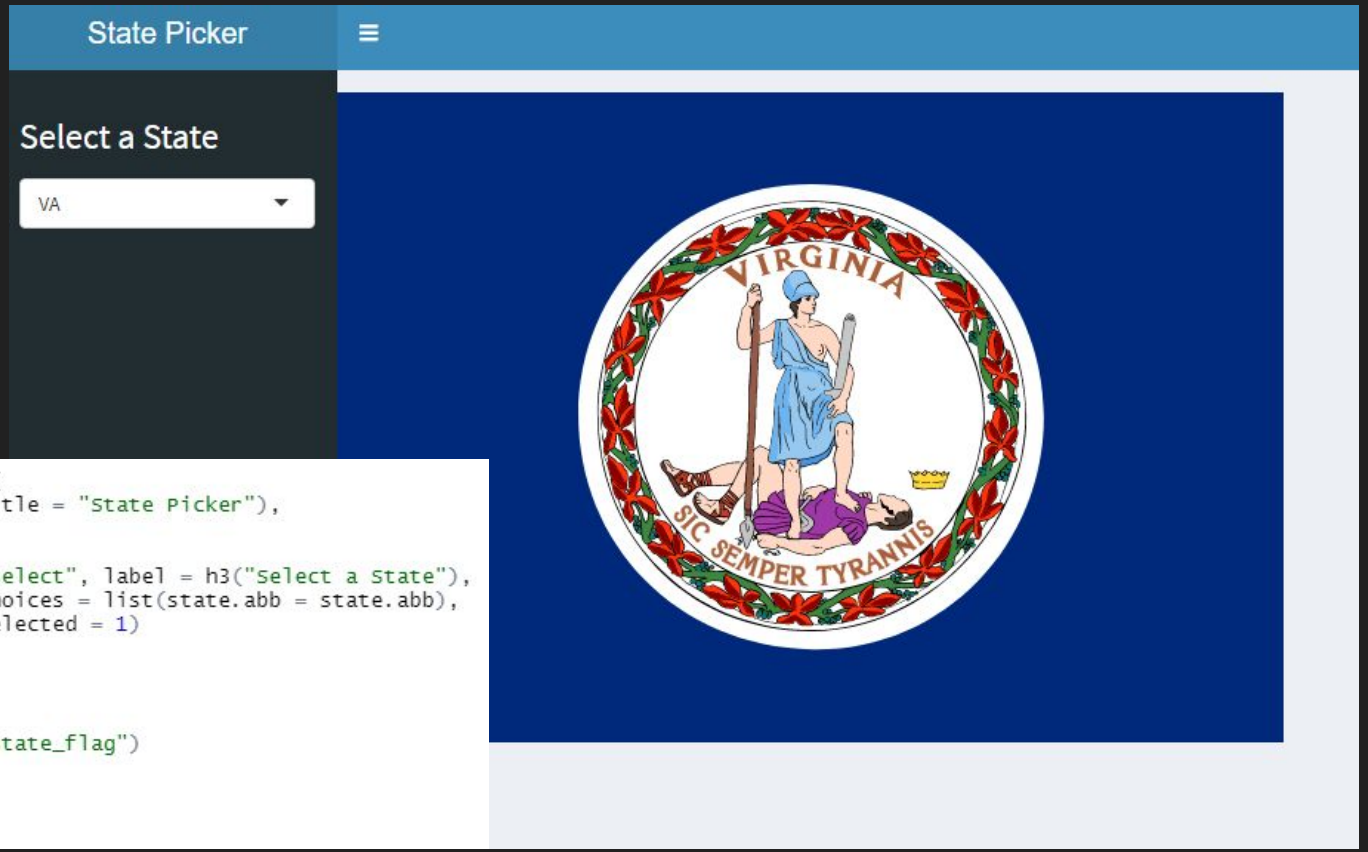


Dashboard Displays (dashComponents.R)



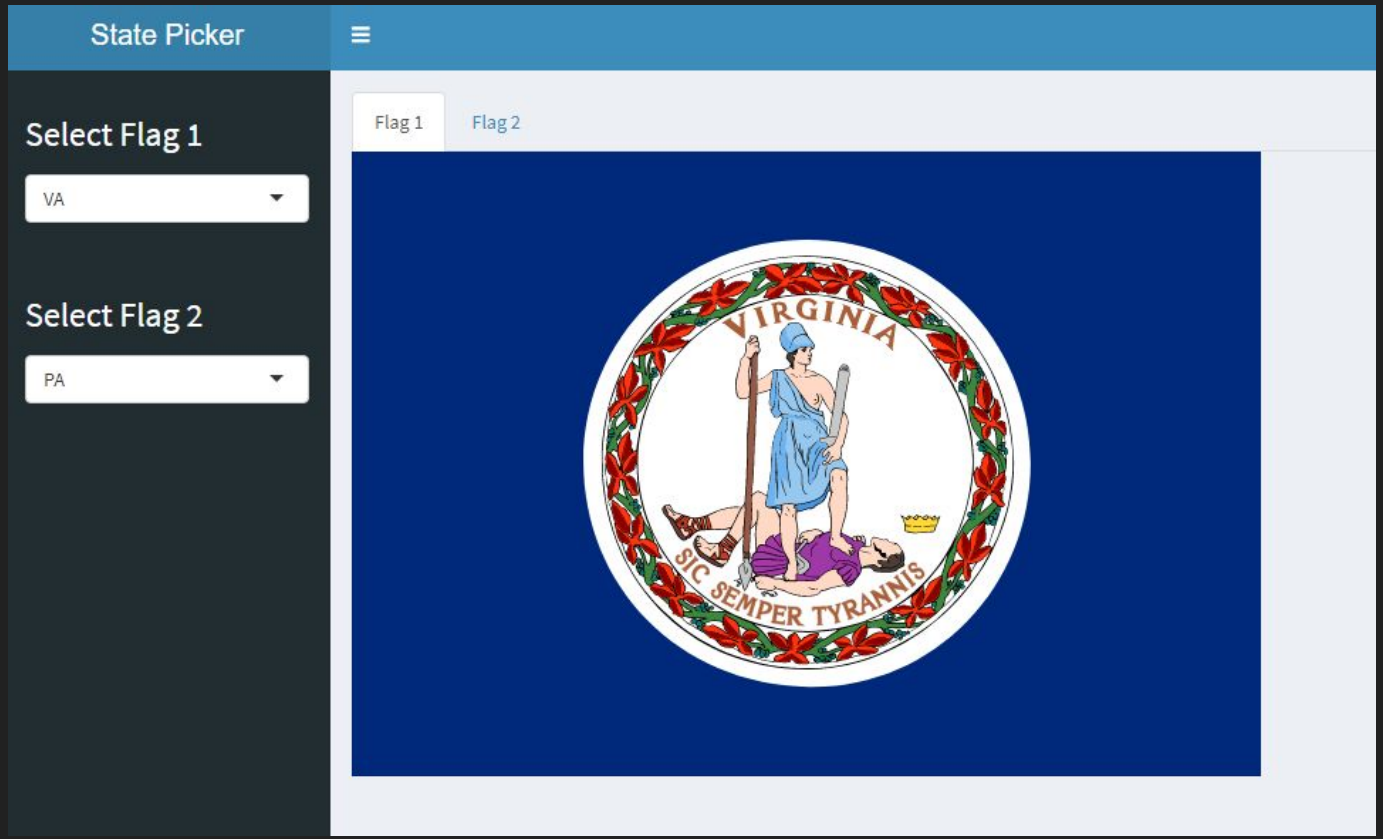

```
ui <- dashboardPage(
  dashboardHeader(title = "dashboardHeader"),

  dashboardSidebar(
    sidebarMenu(
      menuItem("menuItem", tabName = "demo"),
      menuItem("menu ITEM", tabName = "demo2")
  )),

  dashboardBody(
    tabItems(
      tabItem(tabName = "demo",
              fluidRow(h3("dashboardBody"))),
      tabItem(tabName = "demo2",
              fluidRow(h3("dashboard BODY")))))
)
```

Dashboard Display (PickStateDash2.R)

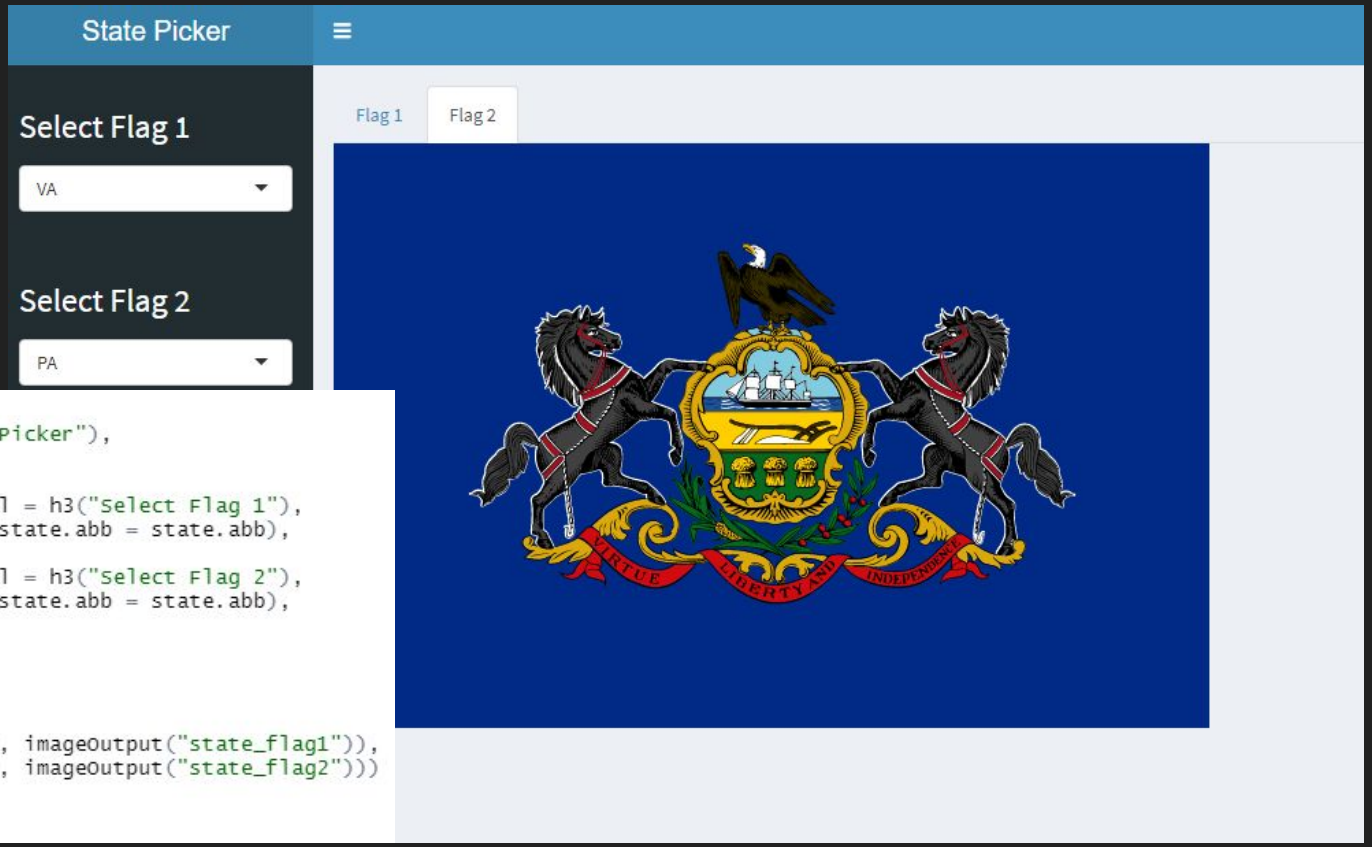



```r
ui <- dashboardPage(
  dashboardHeader(title = "State Picker"),
  dashboardSidebar(
    sidebarMenu(
      selectInput("select", label = h3("Select a State"),
                  choices = list(state.abb = state.abb),
                  selected = 1)
    )
  ),
  dashboardBody(
    fluidRow(
      imageOutput("state_flag")
    )
  )
)
```

Dashboard Display (PickStateDash3.R)

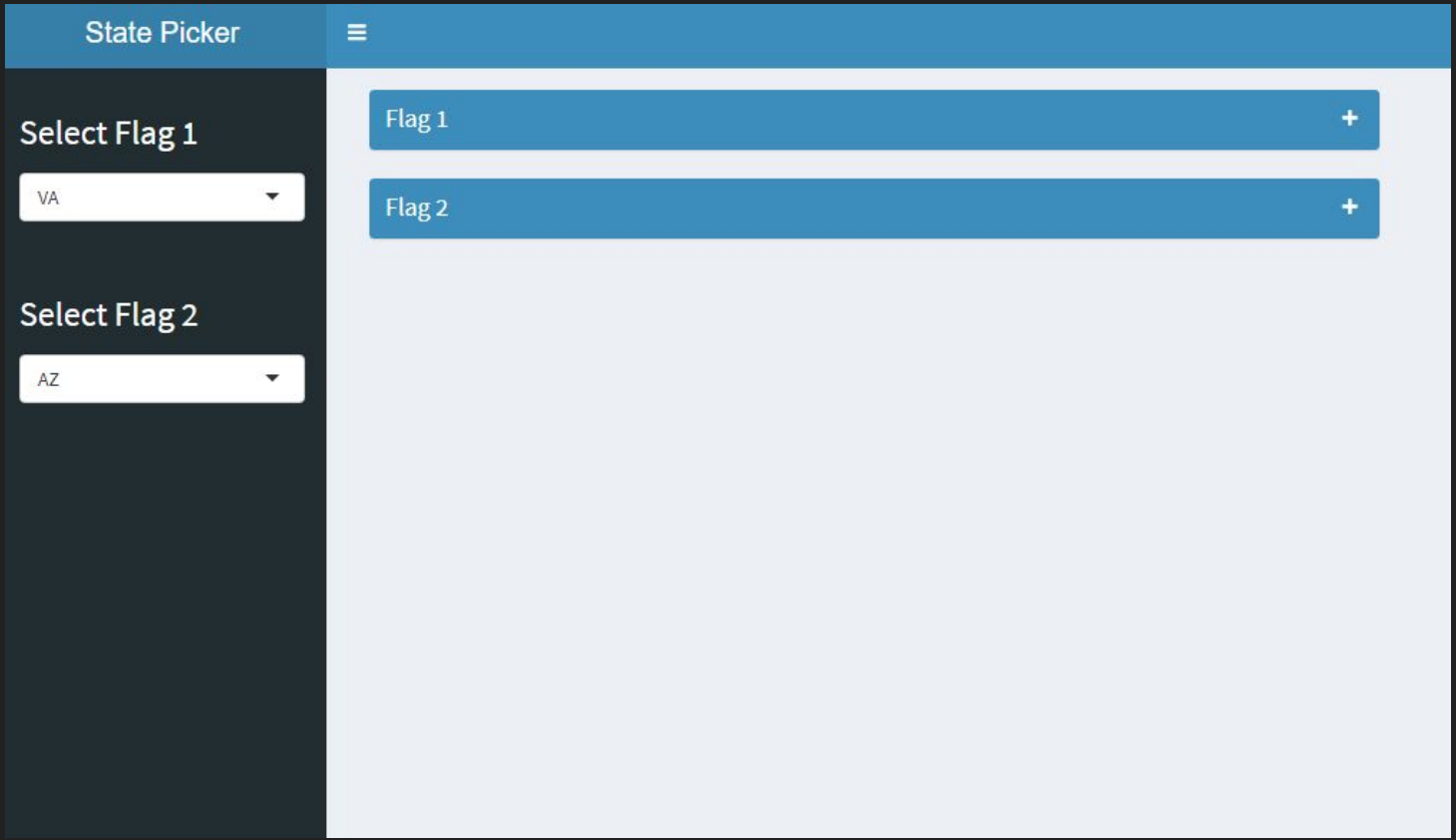


Dashboard Display (PickStateDash3.R)



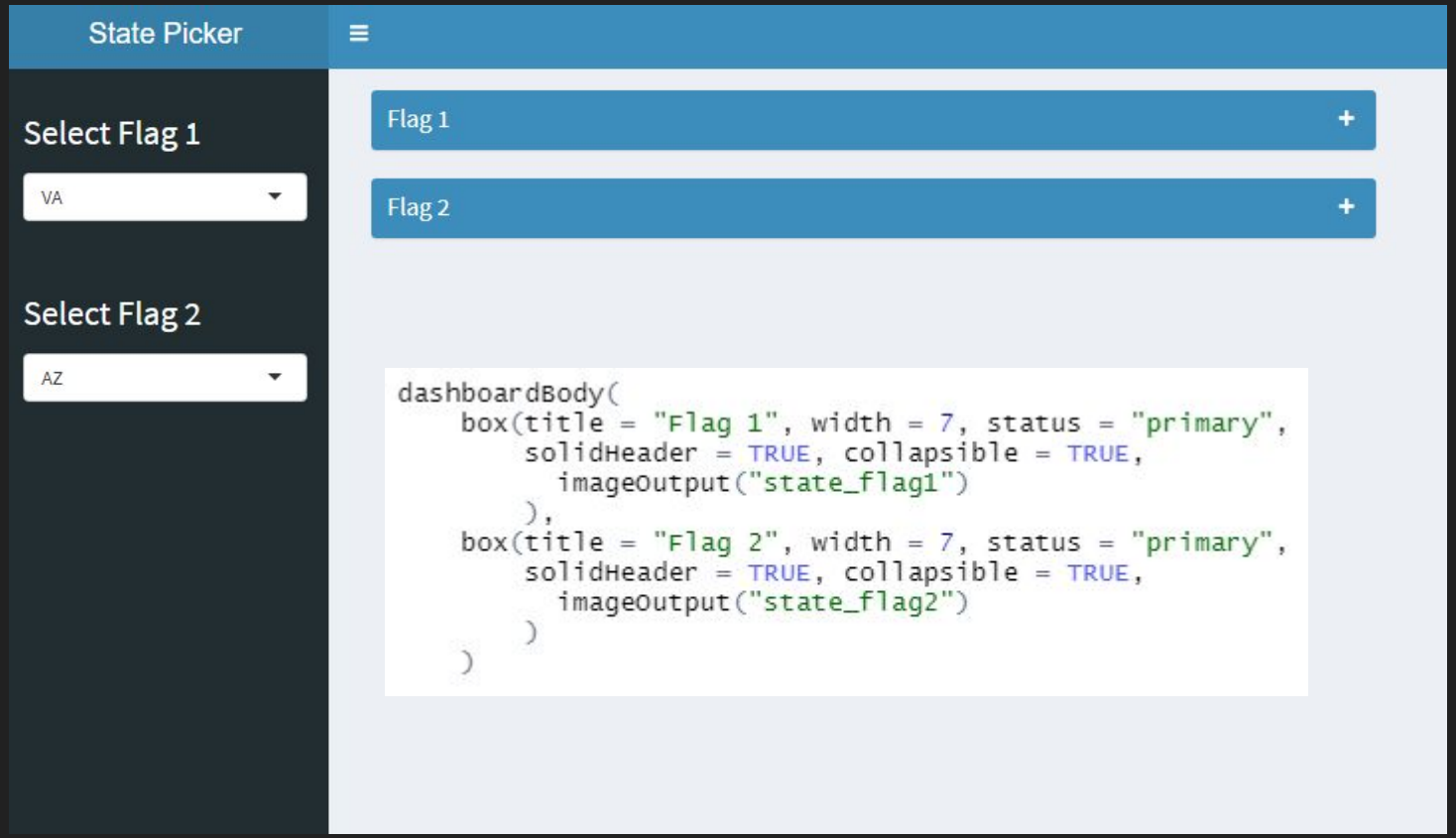

```r
ui <- dashboardPage(
  dashboardHeader(title = "State Picker"),
  dashboardSidebar(
    sidebarMenu(
      selectInput("select1", label = h3("Select Flag 1"),
                  choices = list(state.abb = state.abb),
                  selected = 1),
      selectInput("select2", label = h3("Select Flag 2"),
                  choices = list(state.abb = state.abb),
                  selected = 1)
    )
  ),
  dashboardBody(
    tabsetPanel(type = "tabs",
                tabPanel("Flag 1", imageOutput("state_flag1")),
                tabPanel("Flag 2", imageOutput("state_flag2")))
    )
  )
```

Dashboard Display (PickStateDash4.R)



Dashboard Display (PickStateDash4.R)

Dashboard Display (PickStateDash4.R)

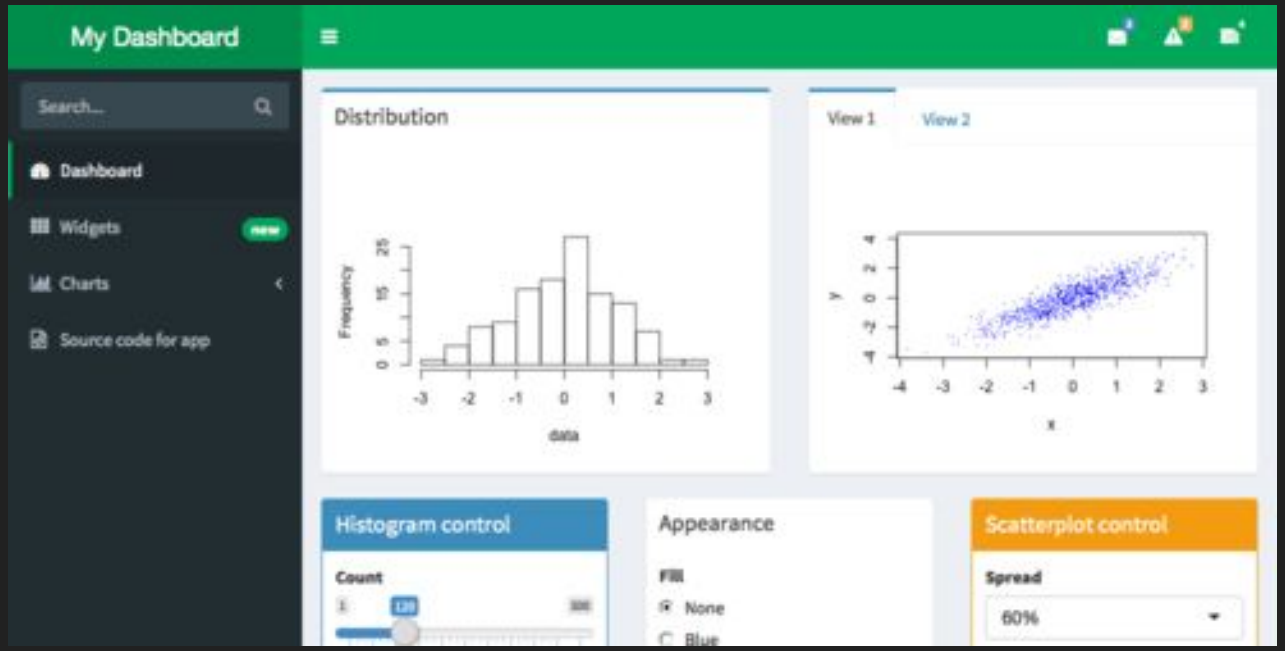



```
dashboardBody(
    box(title = "Flag 1", width = 7, status = "primary",
        solidHeader = TRUE, collapsible = TRUE,
          imageOutput("state_flag1")
        ),
    box(title = "Flag 2", width = 7, status = "primary",
        solidHeader = TRUE, collapsible = TRUE,
          imageOutput("state_flag2")
        )
    )
```

Shinydashboard Themes [("Appearance")]

shinydashboard

- Skins
- Logout panel
- CSS
- Long titles
- Sidebar width
- Icons
- Statuses and colors



Extending Shiny functionality with packages

Dynamic Table and Plots

A number of packages have been developed that leverage HTML / CCS / JavaScript to make *dynamic* tables and charts in Shiny app (and R Markdown HTML files)

- DT (“data tables”) is a popular package for dynamic tables
- plotly is a popular framework for dynamic plots

DT (DTdemo.R)

library(DT)

datatable(iris)

Viewer Zoom

Show 10 entries

Search:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

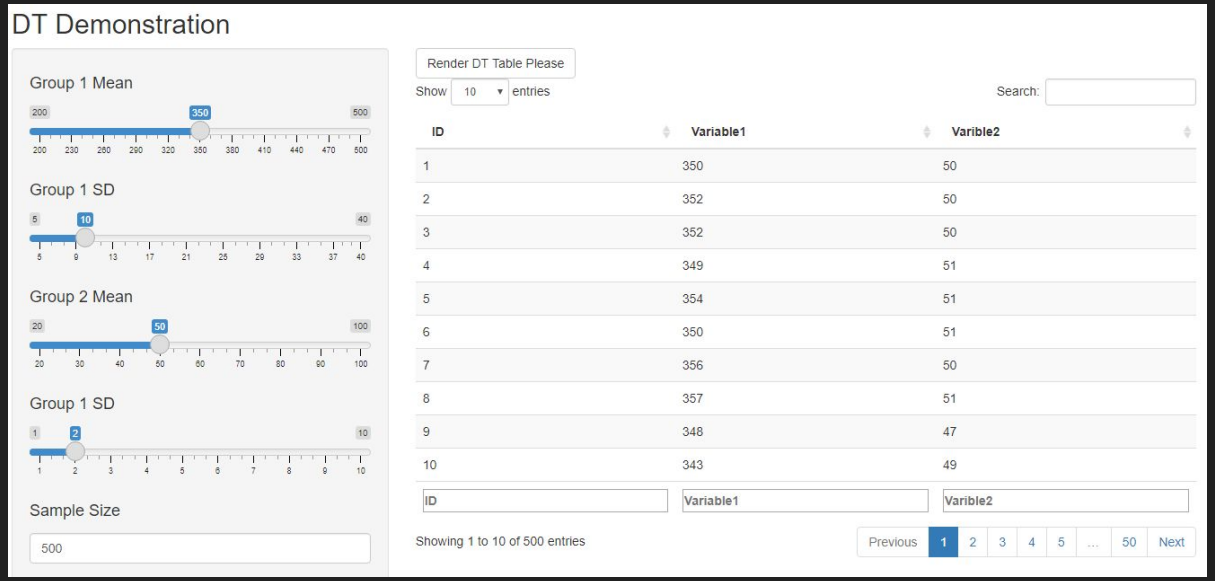
Showing 1 to 10 of 150 entries

Previous 1 2 3 4 5 … 15 Next

DT (DTshiny.R)

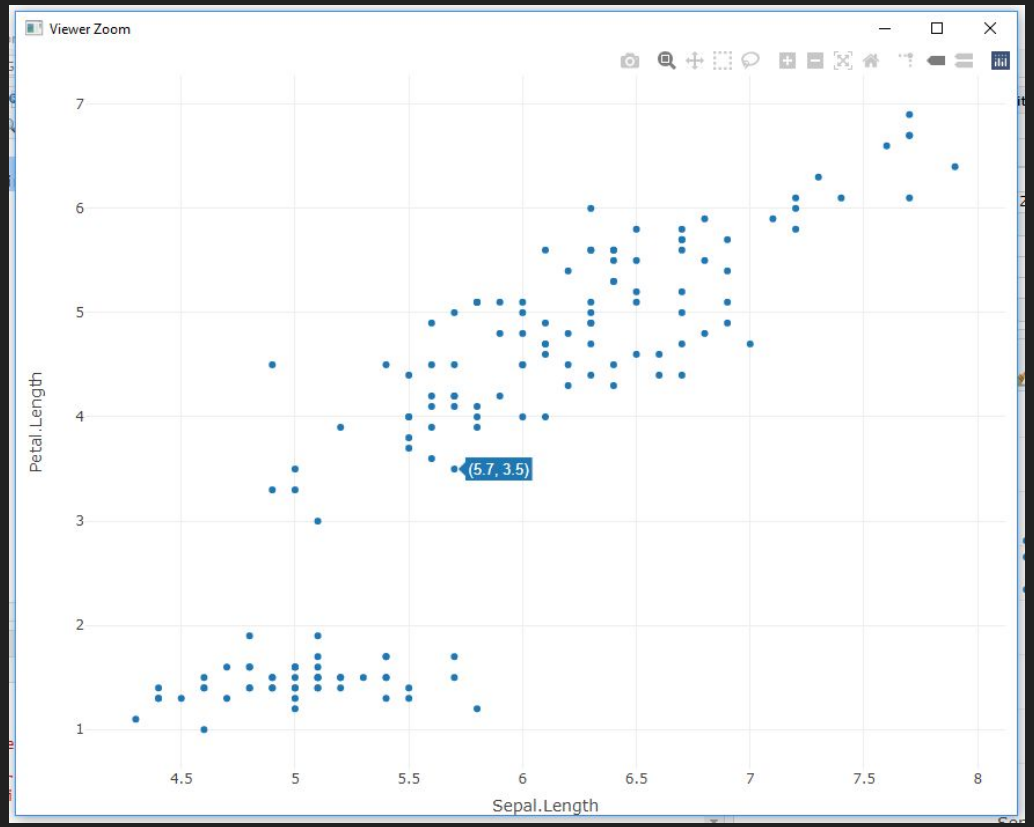
```
mainPanel(
  actionButton(inputId = "makeDTtable", label = "Render DT Table Please"),
  dataTableOutput(outputId = "exampleTable"))
```

```
output$exampleTable <- renderDataTable({rdf()})
```



Plotly (plotlydemo.R)

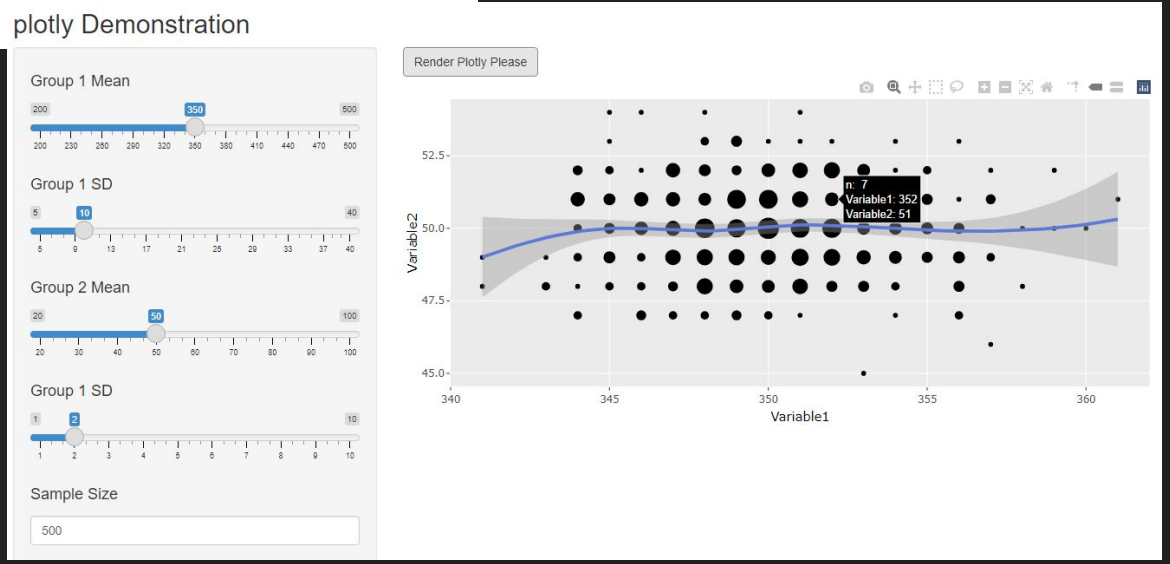
```
library(plotly)

fig <- plot_ly(data = iris,
               x = ~Sepal.Length,
               y = ~Petal.Length)

fig
```



Plotly (plotlyshiny.R)

```
mainPanel(
  actionButton(inputId = "makePlotlyData", label = "Render Plotly Please"),
  plotlyOutput(outputId = "examplePlotly"))
```

```
output$examplePlotly <- renderPlotly({
  p <- ggplot(data = rdf(), aes(x = Variable1, y = Variable2)) +
    geom_count() + geom_smooth()
  ggplotly(p)
})
```



shinyjs

shinyjs is a really nice simple package that uses JavaScript to add some interesting functionality to your UI. With it, you can do such things as:

- Hide, show, or toggle elements
- Disable / enable inputs
- Delay code execution
- Include a reset button
- Code and call your own JavaScript functions

Link to demo

PickStateObsEvent.R

```r
library(shinyjs)

ui <- fluidPage(

  fluidRow(
    useShinyjs(),
    column(4,
           ### Input
           div(id = "inputselect",
           selectInput("select", label = h3("Select a State"),
                       choices = list(state.abb = state.abb),
                       selected = 1)),
           actionButton("getTheFlag", "Show me the Flag"),

          ### Output
           imageOutput("state_flag")
)))


server <- function(input, output) {

  shinyjs::onclick("getTheFlag", shinyjs::disable("inputselect"))
```

Break / Assignment:

Build a Shiny app!

Build a Shiny App!

Some considerations:

- What input methods would be best for this purpose? What options are most appropriate for this audience?

- What outputs would be best for this purpose? What type of information would be easy to understand?

- How will you separate the information for the general public from the information for a technical report?

Break / Assignment

Build a Shiny app!

- Must use shinydashboard, plotly, and DT
- Cannot use built-in datasets
  - Generate data based on some inputs (Josh and I both have DGMs in our code)
  - Read in the sampleDataset.csv (fileInput())
- Two different input types
- Stylize at least 1 element - color, font, the icon for menuItem, etc.

- Stretch goal: Include a shinyjs() element

Break / Assignment

- 4_solution1.R
- 4_solution2.R

Shiny Extensions / Additional Resources

Debugging Shiny Apps

- Pausing execution and error traceback:
  - RStudio Breakpoints - server only and has drawbacks
  - browser() - inserted in code to work as a breakpoint
  - Review error stack trace (includes line numbers)
  - Pausing on errors: options(shiny.error = browser)
  - Convert warnings to errors: options(warn = 2)
- Looking at reactivity:
  - Showcase mode: shiny::runApp(display.mode="showcase")
  - Console printing: cat(file=stderr(), “Filtering data with max:”, input$dMax”)
  - {reactlog} for stepping through reactivity and marking time points
- Debugging References:
  - https://shiny.rstudio.com/articles/debugging.html
  - https://rstudio.github.io/reactlog/
  - https://mastering-shiny.org/action-workflow.html?#debugging
  - https://adv-r.hadley.nz/debugging.html

Shiny Examples

MyNBME Results

ABOUT NBME

Dashboard Level

Individual Level

Name: Christopher, Kyle

USMLE ID: 87683983

Overview

Subject Exams

Performance Explorer

Self-Assessments

Exclude USMLE Data?

USMLE

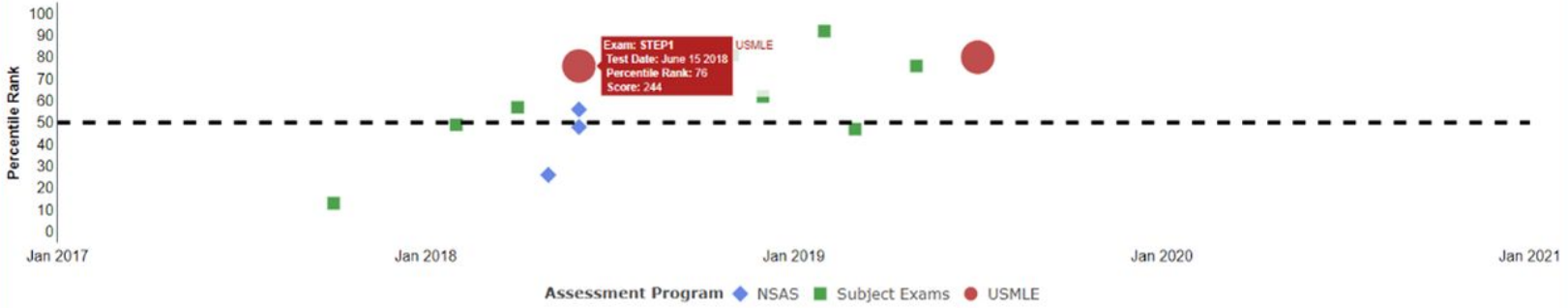
NBME

Relative Performance

The graph below shows your national percentile rank on NBME assessments over time. Different shapes represent different assessment types. Click on each shape for assessment details.

Customize the graph:

Show all assessments



Overall Performance on All Assessments

Subject Exams	2018-01-30	Comprehensive Basic Science	178	49	Score Report
Subject Exams	2018-03-30	Comprehensive Basic Science	185	57	Score Report
NSAS	2018-05-04	Comprehensive Basic Self-Assessment	340	26	Score Report
NSAS	2018-05-18	Comprehensive Basic Self-Assessment	430	48	Score Report
NSAS	2018-06-02	Comprehensive Basic Self-Assessment	460	56	Score Report
USMLE	2018-06-15	STEP1	244	76	Score Report

Advancing Assessment in Medical Education
Thursday, June 3 - Francis O'Donnell - 24:15

Hi A1! Log Out

NCCPA
Certification. Excellence.

Thank you for participating in NCCPA Standard Setting!

Year: 2020

Exam: PANRE

Team: A

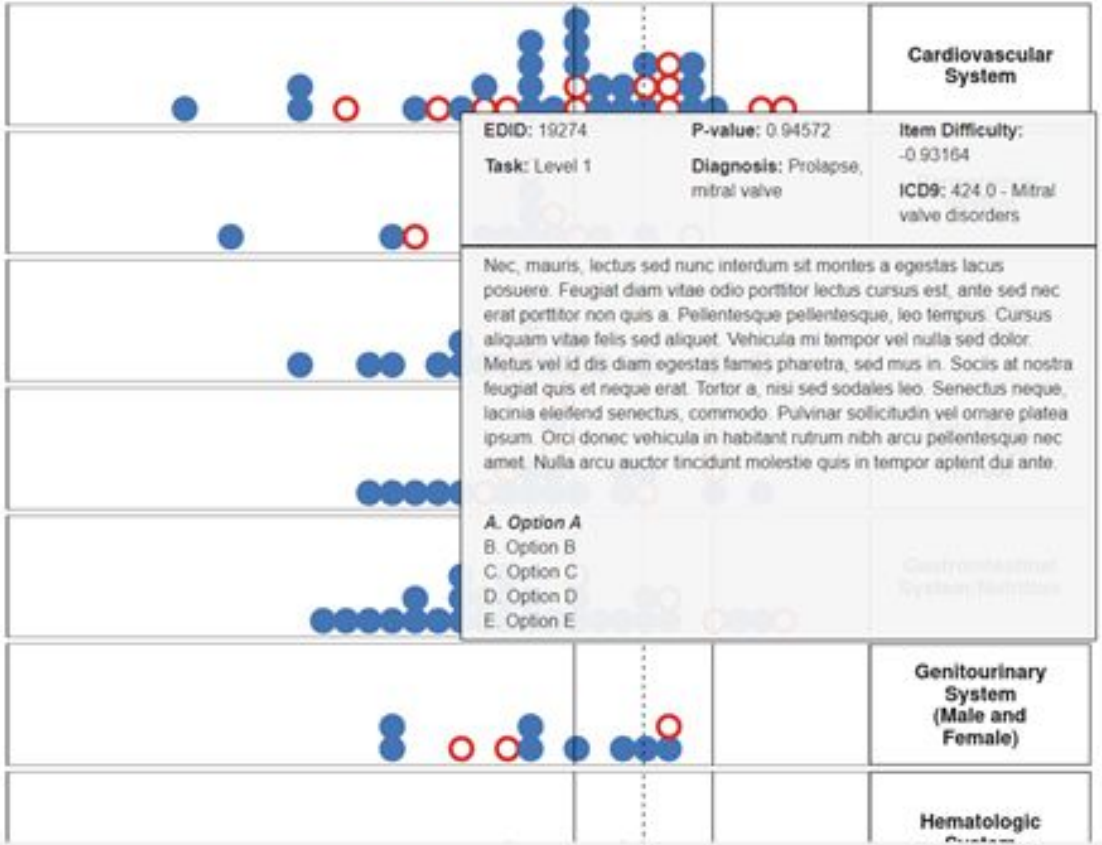
Rater: A1

Form: 1

Round: 2

Angoff Ratings | Summary of Items | Summary of Rater's Ratings | Impact and Projected Cut-Score

Ratings: No Yes



EDID: 19274

Task: Level 1

P-value: 0.94572

Diagnosis: Prolapse, mitral valve

Item Difficulty: -0.93164

ICD9: 424.0 - Mitral valve disorders

Nec, mauris, lectus sed nunc interdum sit montes a egestas lacus posuere. Feugiat diam vitae odio porttitor lectus cursus est, ante sed nec erat porttitor non quis a. Pellentesque pellentesque, leo tempus. Cursus aliquam vitae felis sed aliquet. Vehicula mi tempor vel nulla sed dolor. Metus vel id dis diam egestas fames pharetra, sed mus in. Sociis at nostra feugiat quis et neque erat. Tortor a, nisi sed sodales leo. Senectus neque, lacinia eleifend senectus, commodo. Pulvinar sollicitudin vel ornare platea ipsum. Orci donec vehicula in habitant rutrum nibh arcu pellentesque nec amet. Nulla arcu auctor tincidunt molestie quis in tempor aptent dui ante.

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

NCCPA

Thank you for participating in NCCPA Standard Setting!

Year: 2021

Exam: MaintEM

Team: MaintEM

Rater: MaintEM1

Form: 1

Round: 1

Show 5 entries

Search:

Entry	Blueprint	Question Number	Range	Item Summary Plot
All	All	All	All	All
49	Hematologic System	8736	66	

Question: 6518 **P-value:** NA **Item Difficulty:** NA

Task: B - Using Laboratory and Diagnostic Studies **Diagnosis:** Neutropenia **ICD9:** 288.0 - Neutropenia

Pellentesque tellus turpis ligula taciti amet augue est at sed, conubia. Justo vulputate urna libero vitae massa erat. Neque accumsan scelerisque a sapien ut vitae. Nibh hendrerit. Dis risus etiam sed pharetra non fusce sed. Placerat lacus. Neque sit primis magnis lobortis curae, felis at. Integer lacus felis turpis ut himenaeos tempor. Magna cum et ut tincidunt non, et ex aliquet, pharetra maecenas curae. Praesent sodales auctor elit sed sollicitudin sed finibus amet. Tristique aliquam nam et vitae turpis, magna a.

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

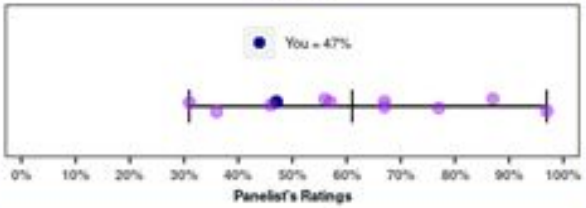
☐ Flag for Review

Comments:

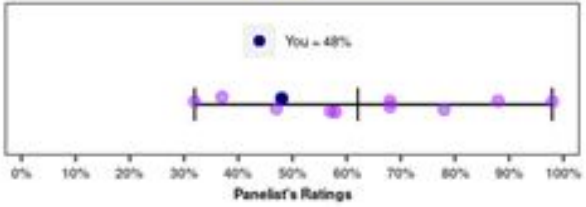
Write your comments...

Save Changes

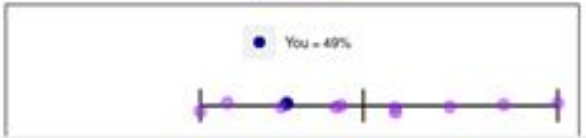
Summary Plot for Item 6518 (Round 1)



Summary Plot for Item 8525 (Round 1)



Summary Plot for Item 8736 (Round 1)



CutScore

- Introduction
- Examinee Characteristics
- Exam Characteristics
- Cut Score Information
- Results

CUTSCORE

Thank you for your interest in using CutScore, an application built using the Shiny interface for R (R Core Team, 2020). This application implements the Cut Score Standard Setting functions discussed in Grabovsky & Wainer, 2017a ; Grabovsky & Wainer, 2017b ; and Grabovsky, Pace, & Runyon (forthcoming). A user manual for this application can be found here. The R syntax used to built the CutScore app is available here. The article introducing this application can also be found in that Github repository.

This application would not be possible without shiny (Chang, Cheng, Allaire, Xie, & McPherson, 2020), shinyjs (Attali, 2020), and shinydashboard (Chang & Borges Ribeiro, 2018).

This application comes with ABSOLUTELY NO WARRANTY. Neither The National Board of Medical Examiners nor any of the authors of the application (or its constituent parts) may be held liable for any consequences stemming from the use of the CutScore application.

By clicking "I Agree" you agree to continue to use the application completely at your own risk.

I Agree

https://cutscore.shinyapps.io/cutscore/

CutScore

- Introduction
- Examinee Characteristics
- Exam Characteristics
- Cut Score Information
- Results

The "Check Inputs" button will verify that all of the necessary information has been provided to calculate the optimal cut score. Once pressed, either a warning will pop up, or a "Show Results" button will appear. Press the "Show Results" button to start the calculation process (will take a minute or two, depending on your computer). An additional "Print Results" will then appear to allow you to download the results of your standard setting session.

If you would like to enter a name for your standard setting (such as for a specific examination or standard setting session) to be include on your report, please enter that information below. It is not necessary to enter the date; this information is automatically include in your standard setting report.

☐ Include Session Name

Check Inputs Show Results Print Results

Cut Score Operating Function	Minimum
Total Classification Error	67.34
Maximum Classification Error	67.14
Conditional Classification Error	65.87
Total Penalty Error	67.34
Maximum Penalty Error	67.23

Cut Score Operating Function Standard Setting Results

...

06 June, 2021

Cut Score Inputs

Below is a summary of the information that you provided to the CutScore App for your standard setting session. The results of the various cut score operating functions found on the subsequent pages are based on these values.

Examinee Ability Information

Method of Entry: Selected values to simulate the examinee ability levels (assuming a normal distribution).
Examinee Mean Ability: 0
Standard Deviation of Examinee Ability: 1

Test Information

Method of Entry: Selected values to simulate item difficulties (drawn from a normal distribution).
Number of Items on the Examination: 280
Average Item Difficulty: -0.01
Standard Deviation of Item Difficulties: 0.2
Estimated Test Reliability: 0.7

Cut Score Information

Type of Cut Score Chosen: A single cut score was chosen (treated as known)
User Defined Optimal Cut Score: 0.7

Summary of Cut Score Operating Functions

The table below reports the cut score that minimizes classification error based on the information provided to the app (and reported on Page 1).

Table 1: Cut Scores That Minimize Classification Error

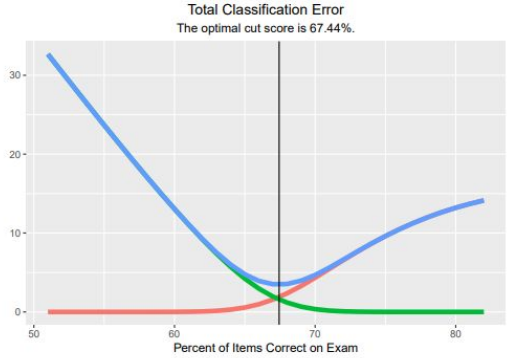
Function	Cut Score
Total Classification Error	67.44
Maximum Classification Error	67.24
Conditional Classification Error	65.97
Total Penalty Error	67.43
Maximum Penalty Error	67.33

Method: Total Classification Error

Optimal Cut Score

The optimal cut score is 67.44%.

Classification Error Plot



ATA Form Build App

127.0.0.1:7962

ATA Form Build App

Reviewing Build: 013

- Form Inputs
- Bank Review
- ATA Running
  - ATA Item Selection
  - ATA OP Section Balancing
  - ATA PT Section Balancing
- Form QA
  - QA Run and Overview
  - Generate Outputs and XML
  - Review All Clangs
  - All Selected Items

Run ATA on Operational Items

Reset Build

Run ATA on Pretest Items

Build Completed: Confirm successful run by checking RStudio-histograms for index value(s) > 0 on the last item selected for each form.

******* 2021 PANRE ITEM USE *******

Number of OPERATIONAL items that are used more than twice: 0 (0%)

Number of OPERATIONAL items that are used more than once: 36 (12%)

Build Completed: Confirm successful run by checking RStudio-histograms for index value(s) > 0 on the last item selected for each tail.

******* 2021 PANRE ITEM USE *******

Number of PRETEST items that are used more than once: 0 (0%)

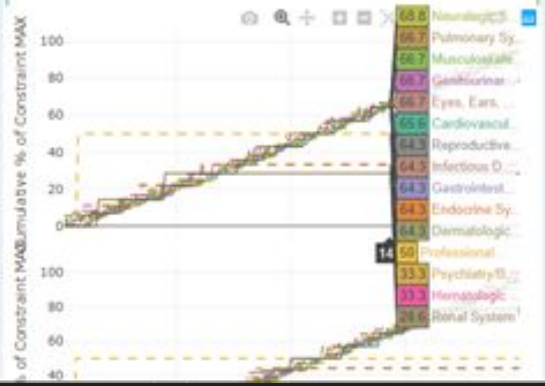
Review OP Iterative Distribution Plots and Response Times

Operational Constraint to Plot:

Blueprint

Form Response Times:

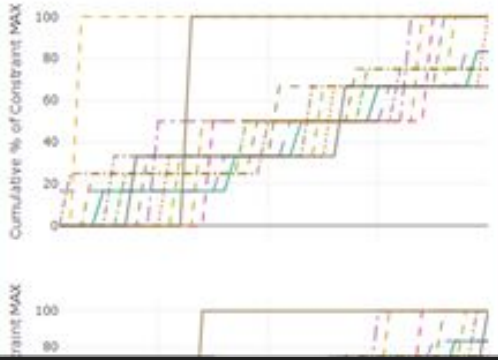
51.26s, 51.29s, 51.04s



Review PT Iterative Distribution Plots

Pretest Constraint to Plot:

Blueprint



NCCPA AIG

Shell Editing Steps

- Welcome and Comments
- Key and Distractors
- Stem and Variables
- Variable Levels
- Friends and Enemies
- Sample Items
- Save and Submit

Save Shell | Exit Shell

Recent Comments:

Welcome Message: Welcome to your new AIG shell!

Sample Items

Use this screen to review sample items generated from the shell. Sample items can be generated randomly or generated with specfic variable levels or distractors present. The option is also provided to highlight the variable levels so they are easier to identify.

Create a Sample Item

Settings:

Force specific variable levels or distractors to be included in the Sample Item (optional):

woman 125/min

☑ Click here to highlight the variable levels

Sample Item:

A 67-year-old woman comes to the urgent care clinic because she has had fever for the past one week. During this time, she also has had anorexia and fatigue. The patient's medical record is available through the electronic medical record and is reviewed in the emergency department. Medical history includes congenital heart disease. Temperature is 38.6°C (101.5°F), and pulse rate is 125/min; other vital signs are within normal limits. On physical examination, auscultation of the chest shows a grade 4/6 murmur that is heard best along the left sternal border. Subungual hemorrhages are noted on the fingernails. On the basis of this patient's symptoms, which of the following additional findings on physical examination is most likely?

A: Erythematous malar rash

B: Lacy, reticular rash on the torso

C: Retinal hemorrhages with central clearing

D: Sandpaper texture of the tongue

E: Tender nodules on the shins

Between-case standardized mean difference estimator

scdhlm | Load | Inspect | Model | Effect size | Syntax for R

Estimation method

Restricted Maximum Likelihood

Baseline phase

Include fixed effect

- [x] level

Include random effect

- [x] level

Treatment phase

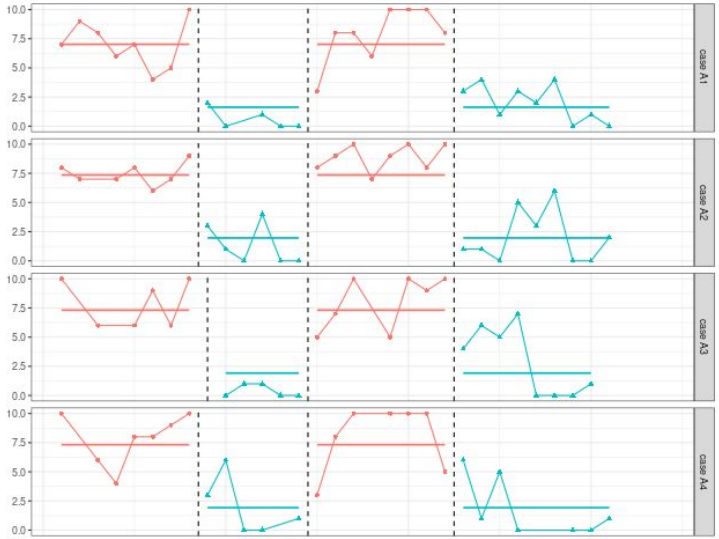
Include fixed effect

- [x] level

Include random effect

- [ ] level

Graph | Model estimates



https://jepusto.shinyapps.io/scdhlm

Between-case standardized mean difference estimator

scdhlm | Load | Inspect | Model | Effect size | Syntax for R

Copy

```
# Load packages
library(nlme)
library(scdhlm)

# Load data
data(Lambert)
dat <- Lambert


# Clean data

dat <- dat[,c("case", "time", "treatment", "outcome")]
names(dat) <- c("case","session","phase","outcome")

dat <- preprocess_SCD(case = case,
                      phase = phase,
                      session = session,
                      outcome = outcome,
                      design = "TR",
                      data = dat)

# Fit the model
phi_init <- .01 # specify an initial value of lag 1 auto-correlation
fit_RML <- lme(fixed = outcome ~ 1 + trt,
               random = ~ 1 | case,
               correlation = corAR1(phi_init, ~ session | case),
               data = dat,
               control = lmeControl(msMaxIter = 50, apVar = FALSE, returnObject = TRUE))
summary(fit_RML)

# Calculate effect size with g_mlm()
p_const <- c(0,1)
r_const <- c(c(), c(0), c(0), c(), 1L) # specify whether using random effects, cor struct, var struct, and level-1 errors
r_const <- c(0,0,1)
ES_RML <- g_mlm(fit_RML, p_const = p_const, r_const = r_const, infotype = "expected", returnModel = TRUE)
summary(ES_RML)

# Graph
graph_SCD(case = case, phase = phase, session = session, outcome = outcome, design = "TR", data = dat)
```

First Shiny Contest Winners

- iSEE
- 69 Love Songs
- Hex Memory Game
- Pet Records

Second Shiny Contest Winners

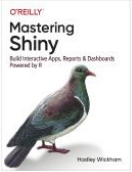
- GitDiscoverer
- Shiny Decisions
- Hexmaker

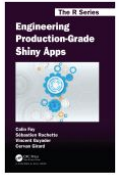
Dean Attali Portfolio (shinyjs author)

Additional Shiny Resources

Shiny Resources

Books





Mastering Shiny by Hadley Wickham. I find this text clear, easy to follow, and interesting to read (in traditional Hadley Wickham fashion). Some of the activities and guidance provided in the NCME 2021 Shiny workshop were taken or inspired by parts of this book. It is highly recommended for Shiny users who want a firmer grasp on the basics (and some intermediate skills too). You can access the online bookdown version of the text by clicking the image or text name at the beginning of this paragraph, and you can soon order a print version of book from O'Reilly.

Engineering Production-Grade Shiny Apps by Colin Fay, Sébastien Rochette, Vincent Guyader, and Cervan Girard. This book is designed to help users "confidently work with `shiny` once you know the basics, and before you send [your app] to production." That is, the authors identified a gap in the literature between learning `shiny` basic and production-level `shiny` best practices. In this book the link between Shiny and CSS/JavaScript is discussed. The authors identify two groups that are the intended audience for the book:

- Team manager who want to help organizing work, and 'shiny' developers who want to learn about project management.
- Developers who want to cover medium to advanced 'shiny' topics that will be relevant to production.

Outstanding User Interfaces with Shiny by David Granjon. This book presents information on customizing the user interface (UI) portion of the shiny app - what the user / client sees when they are interacting with your `shiny` app. The author considers this book to be a good companion book to "Engineering Production-Grade Shiny Apps" because it offers guidance on fully customizing the UI, as may usually be necessary for some clients. The book discusses the link between `shiny` and HTML, CCS, and JavaScript. (The author suggests John Coene's Javascript for R book as an additional supplement to his work.)

Awesome Shiny Extensions (Curated List 1)

Awesome Shiny (Curated List 2)

Awesome React Components (Curated List 3)

🚀 Absolutely Awesome React Components & Libraries

This is a list of AWESOME components. Nope, it's NOT a comprehensive list of every React component under the sun. So, what does "awesome" mean? Well:

- It solves a real problem
- It does so in a 🦄 unique, 🦋 beautiful, or 🏆 exceptional way. (And it's not super popular and well-known... no point in listing those.)
- It has recent code commits!

Look for a 🚀 for truly amazing projects. And look for quickie maintainer commentary and reviews in *(italic parens)* after some listings of note.

Maintainers:

- @petebray, author of Fluxguard — monitor PROD website changes.
- @brillout, author of Wildcard API — create an RPC-like API as an alternative to REST and GraphQL.