Shiny applications without Shiny

Emily Mitchell

October 25, 2018
The views expressed in this presentation are those of the author and no official endorsement by the Department of Health and Human Services, the Agency for Healthcare Research and Quality is intended or should be inferred.
What is Shiny?
### Total expenditures in millions by insurance coverage, United States, 2011-2016

<table>
<thead>
<tr>
<th>Year</th>
<th>Any private, all ages</th>
<th>Public only, all ages</th>
<th>Uninsured, all ages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>1,050,907</td>
<td>524,318</td>
<td>42,306</td>
</tr>
<tr>
<td>2015</td>
<td>1,071,867</td>
<td>496,873</td>
<td>31,167</td>
</tr>
<tr>
<td>2014</td>
<td>989,708</td>
<td>459,348</td>
<td>50,309</td>
</tr>
<tr>
<td>2013</td>
<td>933,725</td>
<td>412,932</td>
<td>53,866</td>
</tr>
<tr>
<td>2012</td>
<td>918,277</td>
<td>383,092</td>
<td>49,353</td>
</tr>
<tr>
<td>2011</td>
<td>912,934</td>
<td>371,142</td>
<td>46,649</td>
</tr>
</tbody>
</table>

-- Estimates suppressed due to inadequate precision (see FAQs for details).
* Relative standard error is greater than 30%
Anatomy of a web page

HTML

```html
<body>
  <div class="container-fluid">
    <h2>Mean expenditure per person, MEPS 2006-2016</h2>
    <form>
      <label for="group">Group By:</label>
      <select id="group" class="form-control">
        <option value="age" selected>Age Group</option>
        <option value="insurance">Insurance Coverage</option>
        <option value="race">Race/Ethnicity</option>
      </select>
    </form>
    <div id="expPlot"></div>
  </div>
</body>
```
Anatomy of a web page

**HTML**

```
<body>
  <div class="container-fluid">
    <h2>Mean expenditure per person, MEPS 2006-2016</h2>
    <form>
      <label for="group">Group By:</label>
      <select id="group" class="form-control">
        <option value="age">Age Group</option>
        <option value="insurance">Insurance Coverage</option>
        <option value="race">Race/Ethnicity</option>
      </select>
    </form>
  </div>

  <div id="expPlot"></div>
</body>
```

**JavaScript (JS)**

```
$('#group').on('change', function(){
  var grp = $('#group').val();
  var data = MEPS_data[grp];
  Plotly.newPlot('expPlot', data);
});

// Initialize default
$('#group').trigger('change');
```
Anatomy of a web page

Cascading Style Sheets (CSS)

```css
table {table-layout: auto;}
td, th {white-space: nowrap;}

footer a {text-decoration: none;}
header a {text-decoration: none;}

.tooltip {
  bottom: auto;
  background: none;
  width: auto;
  font-size: 14px;
}
```
Anatomy of a Shiny app

R – Shiny app

```r
ui <- fluidPage(
  titlePanel("Mean expenditure per person, MEPS 2006-2016"),

  selectInput(
    inputId = "group",
    label = "Group By:",
    choices =
      c("Age Group" = "age",
        "Insurance Coverage" = "insurance",
        "Race/Ethnicity" = "race"),

    plotlyOutput(outputId = "expPlot")
  )
)

server <- function(input, output) {
  output$expPlot <- renderPlotly({
    grp <- input$group
    data <- MEPS_data[[grp]]

    data %>%
      group_by(group) %>%
      plot_ly(x = ~Year, y = ~Expenditure,
        type = "scatter",
        mode = "lines+markers",
        color = ~group)
  })
}
```
Anatomy of a Shiny app

**R – Shiny app**

```r
ui <- fluidPage(
  titlePanel("Mean expenditure per person, MEPS 2006-2016"),
  selectInput(
    inputId = "group",
    label = "Group By:",
    choices = c("Age Group" = "age",
                "Insurance Coverage" = "insurance",
                "Race/Ethnicity" = "race")),
  plotlyOutput(outputId = "expPlot")
)

server <- function(input, output) {
  output$expPlot <- renderPlotly({
    grp <- input$group
    data <- MEPS_data[[grp]]
    data %>%
    group_by(group) %>%
    plot_ly(x = ~Year,
             y = ~Expenditure,
             type = "scatter",
             mode = "lines+markers",
             color = ~group)
  })
}
```

**HTML**

```html
<body>
  <div class="container-fluid">
    <h2>Mean expenditure per person, MEPS 2006-2016</h2>
    
    <form>
      <label for="group">Group By:</label>
      <select id="group" class="form-control">
        <option value="age" selected>Age Group</option>
        <option value="insurance">Insurance Coverage</option>
        <option value="race">Race/Ethnicity</option>
      </select>
    </form>
  
  <div id="expPlot"></div>
</div>
```

**JavaScript (JS)**

```javascript
$('\#group').on("change", function(){
  var grp = $('\#group').val();
  var data = MEPS_data[grp];
  Plotly.newPlot('expPlot', data);
});

// Initialize default
$('\#group').trigger('change');
```
Tips for transitioning

1. Start with UI => HTML
2. Add JavaScript for reactivity
3. Add CSS (optional)
1. Start with UI => HTML

R – Shiny app

```r
ui <- fluidPage(
  titlePanel("Mean expenditure per person, MEPS 2006-2016"),
  selectInput(
    inputId = "group",
    label = "Group By:",
    choices = c("Age Group" = "age",
                "Insurance Coverage" = "insurance",
                "Race/Ethnicity" = "race")),
  plotlyOutput(outputId = "expPlot")
)
```

**HTML**

```html
<h2>Mean expenditure per person, MEPS 2006-2016</h2>
<form>
  <label for="group">Group By:</label>
  <select id="group" class="form-control">
    <option value="age" selected>Age Group</option>
    <option value="insurance">Insurance Coverage</option>
    <option value="race">Race/Ethnicity</option>
  </select>
</form>
<div id="expPlot"></div>
```
2. Add JavaScript for reactivity

R – Shiny app

```r
server <- function(input, output) {
  output$expPlot <- renderPlotly({
    grp <- input$group
    data <- MEPS_data[[grp]]
    data %>%
      group_by(group) %>%
      plot_ly(x = ~Year,
               y = ~Expenditure,
               type = "scatter",
               mode = "lines+markers",
               color = ~group)
  })
}
```

JavaScript (JQuery)

```javascript
$('#group').on("change", function(){
  var grp = $('#group').val();
  var data = MEPS_data[grp];
  Plotly.newPlot('expPlot', data);
});
```
3. Add CSS (optional)
<table>
<thead>
<tr>
<th>Feature</th>
<th>Shiny</th>
<th>JavaScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical programming</td>
<td>🧡</td>
<td>🖕</td>
</tr>
<tr>
<td>Data manipulation</td>
<td>🧡</td>
<td>🖕</td>
</tr>
<tr>
<td>Learning curve</td>
<td>🧡</td>
<td>🖕</td>
</tr>
<tr>
<td>Trouble-shooting</td>
<td>😞</td>
<td>😞</td>
</tr>
<tr>
<td>Customization</td>
<td>🖕</td>
<td>🧡</td>
</tr>
<tr>
<td>Speed</td>
<td>🌐Mozilla Firefox</td>
<td>🌐Google Chrome</td>
</tr>
</tbody>
</table>
You may want to transition if:

- You don’t really need R
- You use Shiny for display, not computation
- You write your own R functions to create UI objects
- You frequently use `shinyBS` and `shinyJS` packages, and find they don’t have all the features you want
- You are writing and embedding custom HTML/JS/CSS into your Shiny app
A happy medium?

Style your apps with CSS
https://shiny.rstudio.com/articles/css.html

Customize your UI with HTML
https://shiny.rstudio.com/articles/html-tags.html

Build your entire UI with HTML
https://shiny.rstudio.com/articles/html-ui.html

How to create custom input bindings
https://shiny.rstudio.com/articles/js-custom-input.html

Use R shiny as a server for JavaScript

Image credits: rstudio.com, deanattali.com/shinyjs/, ebailey78.github.io/shinyBS/
Transition example

https://github.com/e-mitchell/Shiny_JS_transition

R – Shiny app

HTML / JS
# Total expenditures in millions by insurance coverage, United States, 2011-2016

<table>
<thead>
<tr>
<th>Year</th>
<th>Any private, all ages</th>
<th>Public only, all ages</th>
<th>Uninsured, all ages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>1,050,907</td>
<td>524,318</td>
<td>42,306</td>
</tr>
<tr>
<td>2015</td>
<td>1,071,867</td>
<td>496,873</td>
<td>31,167</td>
</tr>
<tr>
<td>2014</td>
<td>989,708</td>
<td>459,348</td>
<td>50,309</td>
</tr>
<tr>
<td>2013</td>
<td>933,725</td>
<td>412,932</td>
<td>53,866</td>
</tr>
<tr>
<td>2012</td>
<td>918,277</td>
<td>383,092</td>
<td>49,353</td>
</tr>
<tr>
<td>2011</td>
<td>912,934</td>
<td>371,142</td>
<td>46,649</td>
</tr>
</tbody>
</table>

--- Estimates suppressed due to inadequate precision (see FAQs for details).
* Relative standard error is greater than 30%

**Source:** Center for Financing, Access and Cost Trends, Agency for Healthcare Research and Quality, Medical Expenditure Panel Survey, 2011-2016
## Resources

<table>
<thead>
<tr>
<th>Shiny tutorials</th>
<th><a href="https://shiny.rstudio.com/articles">https://shiny.rstudio.com/articles</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning HTML / CSS / JS</td>
<td><a href="coursera.org/specializations/web-design">coursera.org/specializations/web-design</a> <a href="w3schools.com">w3schools.com</a></td>
</tr>
<tr>
<td>Bootstrap</td>
<td><a href="getbootstrap.com">getbootstrap.com</a></td>
</tr>
<tr>
<td>US Web Design Standards (USWDS)</td>
<td><a href="standards.usa.gov">standards.usa.gov</a></td>
</tr>
<tr>
<td>Web Accessibility (508 compliance)</td>
<td><a href="webaim.org">webaim.org</a></td>
</tr>
</tbody>
</table>

emily.mitchell@ahrq.hhs.gov