R Shiny

Making Graphics Dynamic

R SHINY

WEB APPS FOR THE COMMON PERSON

STANDARD SHINY WIDGETS (INPUTS)

Function Name	Widget
actionButton	Action Button
checkboxGroupInput	A group of check boxes
checkboxInput	A single check box
dateInput	A calendar to aid date selection
dateRangeInput	A pair of calendars for selecting a date range
fileInput	A file upload control wizard
helpText	Help text that can be added to an input form
numericInput	A field to enter numbers
radioButtons	A set of radio buttons
selectInput	A box with choices to select from
sliderInput	A slider bar
submitButton	A submit button
textInput	A field to enter text

Note each function will store different input values:

textInput = a single character element

selectInput = character elements from a list

sliderInput = two numbers in a range

checkboxInput = T / F

Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Action

Current Value:

```
[1] 0
attr(,"class")
[1] "integer"                "shinyActionButtonValue"
```

See Code

Single checkbox

- [x] Choice A

Current Value:

```
[1] TRUE
```

See Code

Checkbox group

- [x] Choice 1
- [ ] Choice 2
- [ ] Choice 3

Current Values:

```
[1] "1"
```

See Code

Date input

2014-01-01

Current Value:

```
[1] "2014-01-01"
```

See Code

Date range

2019-09-16 to 2019-09-16

Current Values:

```
[1] "2019-09-16" "2019-09-16"
```

See Code

File input

Browse... No file selected

Current Value:

```
NULL
```

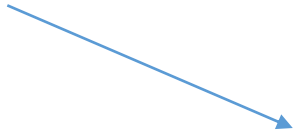
See Code

WIDGET COMPONENTS

- **Name** for the widget. The user will not see this name, but you can use it to access the widget's value. The name should be a character string.

- **Label**. This label will appear with the widget in your app. It should be a character string, but it can be an empty string "".

actionButton(name="submit", label = "Submit Your Form")



Creates an entry at input$submit

How you will access the data:

input$name

Note that you do not create the input object and assign values at input$widget_name. That is done for you by the Shiny package.

RENDERING:
CONVERT R TO DYNAMIC HTML

HOW DOES "KNIT" WORK IN RMD?

Shiny functions work similar to other knitr functions that are used to convert your raw R output into HTML objects that make for nice documents.

Raw R output

```
##
## ===================================================
##                         Dependent variable:
##              ---------------------------------------
##                               heart.rate
##                 (1)       (2)       (3)       (4)
## ---------------------------------------------------
## caffeine     0.087***  0.080***    0.009     0.037
##               (0.021)   (0.008)   (0.121)   (0.047)
##
## gym.time               -1.441***             -1.440***
##                         (0.062)               (0.062)
##
## stress.index                        0.414     0.228
##                                    (0.631)   (0.246)
##
## Constant     68.953*** 116.461*** 68.267*** 116.022***
##               (5.454)   (2.942)   (5.568)   (2.982)
##
## ---------------------------------------------------
## Observations    100       100       100       100
## R2             0.153     0.872     0.157     0.873
## ===================================================
## Note:                     *p<0.1; **p<0.05; ***p<0.01
```

After conversion to HTML table

	Dependent variable:			
	heart.rate			
	(1)	(2)	(3)	(4)
caffeine	0.087***	0.080***	0.009	0.037
	(0.021)	(0.008)	(0.121)	(0.047)
gym.time		-1.441***		-1.440***
		(0.062)		(0.062)
stress.index			0.414	0.228
			(0.631)	(0.246)
Constant	68.953***	116.461***	68.267***	116.022***
	(5.454)	(2.942)	(5.568)	(2.982)
Observations	100	100	100	100
R^2	0.153	0.872	0.157	0.873
Note:				*p<0.1; **p<0.05; p<0.01

SIDE NOTE: THIS IS WHAT THE RAW HTML TABLE LOOKS LIKE

```
<table style="text-align:center"><tr><td colspan="5" style="border-bottom: 1px solid
black"></td></tr><tr><td style="text-align:left"></td><td colspan="4"><em>Dependent
variable:</em></td></tr> <tr><td></td><td colspan="4" style="border-bottom: 1px solid black"></td></tr>
<tr><td style="text-align:left"></td><td colspan="4">heart.rate</td></tr> <tr><td style="text-
align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td></tr> <tr><td colspan="5"
style="border-bottom: 1px solid black"></td></tr><tr><td style="text-
align:left">caffeine</td><td>0.087<sup>***</sup></td><td>0.080<sup>***</sup></td><td>0.009</td><td>0.037
</td></tr> <tr><td style="text-
align:left"></td><td>(0.021)</td><td>(0.008)</td><td>(0.121)</td><td>(0.047)</td></tr> <tr><td
style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td style="text-
align:left">gym.time</td><td></td><td>-1.441<sup>***</sup></td><td></td><td>-
1.440<sup>***</sup></td></tr> <tr><td style="text-
align:left"></td><td></td><td>(0.062)</td><td></td><td>(0.062)</td></tr> <tr><td style="text-
align:left"></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td style="text-
align:left">stress.index</td><td></td><td></td><td>0.414</td><td>0.228</td></tr> <tr><td style="text-
align:left"></td><td></td><td></td><td>(0.631)</td><td>(0.246)</td></tr> <tr><td style="text-
align:left"></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td style="text-
align:left">Constant</td><td>68.953<sup>***</sup></td><td>116.461<sup>***</sup></td><td>68.267<sup>***</
sup></td><td>116.022<sup>***</sup></td></tr> <tr><td style="text-
align:left"></td><td>(5.454)</td><td>(2.942)</td><td>(5.568)</td><td>(2.982)</td></tr> <tr><td
style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td colspan="5"
style="border-bottom: 1px solid black"></td></tr><tr><td style="text-
align:left">Observations</td><td>100</td><td>100</td><td>100</td><td>100</td></tr> <tr><td style="text-
align:left">R<sup>2</sup></td><td>0.153</td><td>0.872</td><td>0.157</td><td>0.873</td></tr> <tr><td
colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-
align:left"><em>Note:</em></td><td colspan="4" style="text-align:right"><sup>*</sup>p<0.1;
<sup>**</sup>p<0.05; <sup>***</sup>p<0.01</td></tr> </table>
```

RENDER FUNCTIONS:

Raw R Version

```
plot( x, y, main="My Plot" )
```

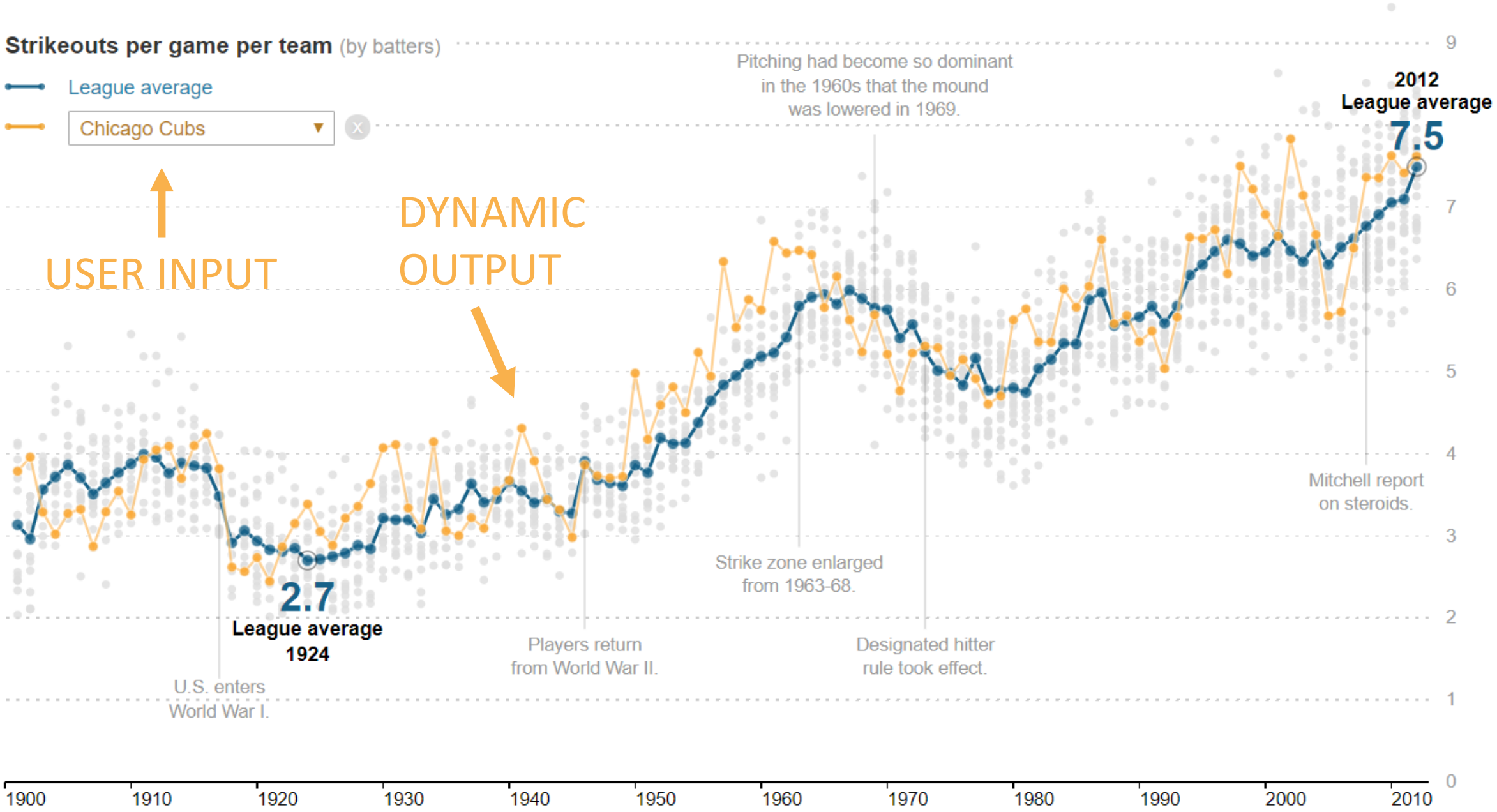
R Shiny Version

```
renderPlot({
  plot( x, y, main="My Plot" )
})
```

Converts this to a shiny object
that can be updated and
re-plotted inside a browser.

Strikeouts on the Rise

There were more strikeouts in 2012 than at any other time in major league history.



RENDER FUNCTIONS:

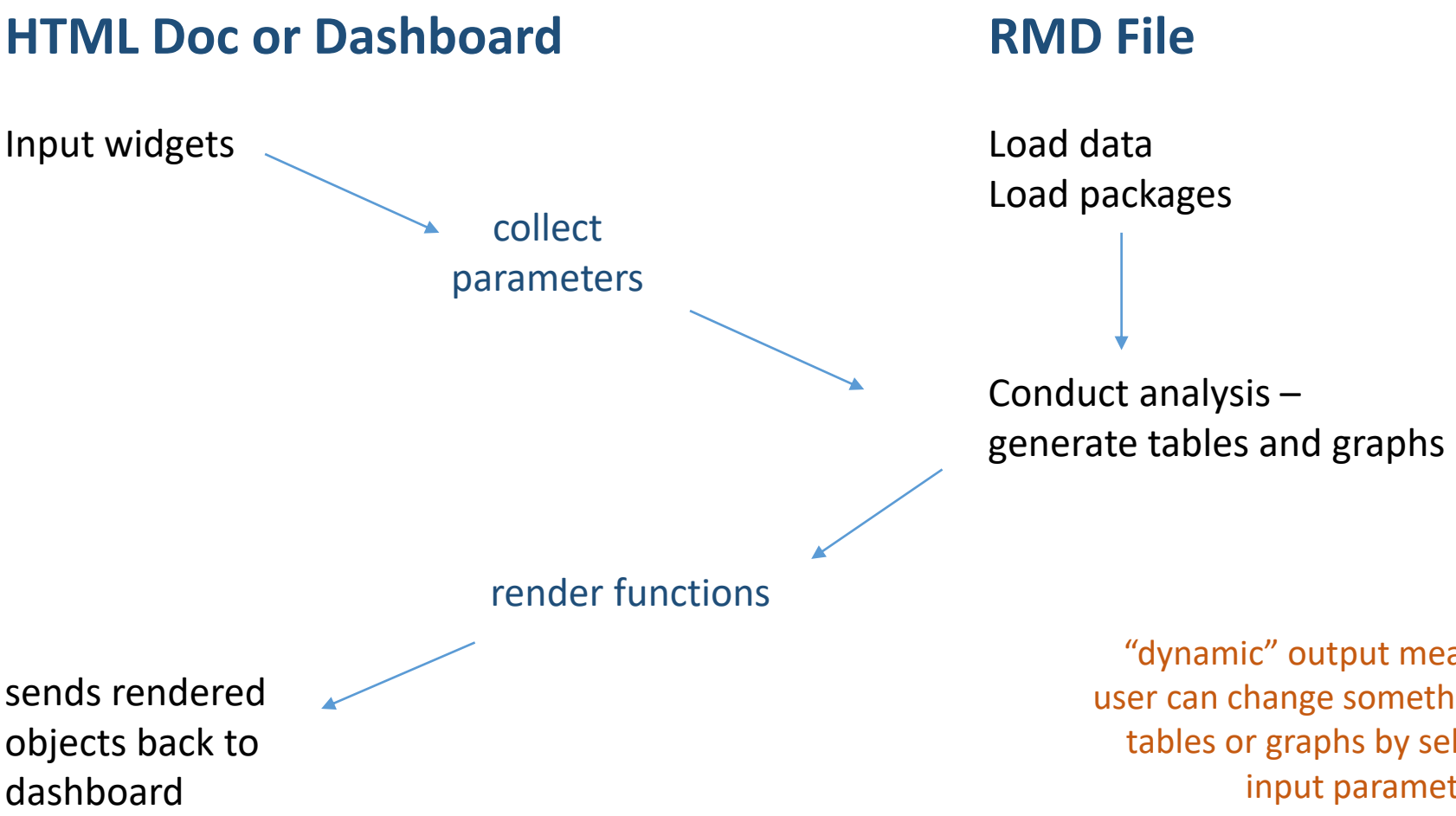
The output functions take R code and “render” it as HTML objects that can be used in web browsers in order to display your dashboard. Shiny functions add some javascript features that allow output to be updated in real-time inside a browser.

Output Functions	**Creates**
renderImage	image
renderPlot	plot
renderTable	table
renderText	text

Note that HTML creates static text, tables, and images in web documents. Any time you are doing something active on a webpage (other than clicking a link), you are using the javascript language. It was created as a way to make web pages more interactive and responsive.

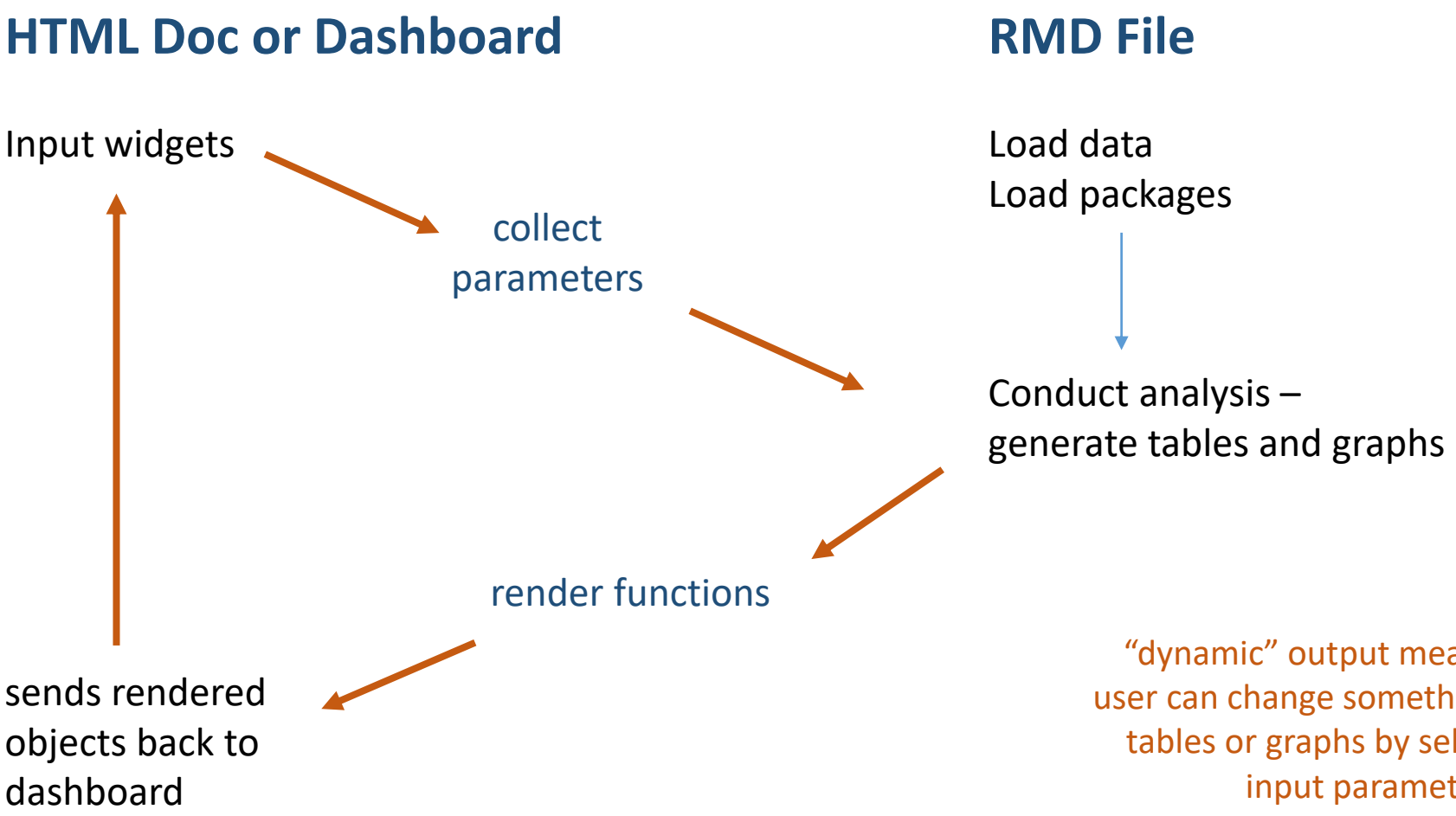
- knitr → converts R to HTML when knitting RMD documents
- shiny functions → convert R to javascript when knitting RMD documents

ANATOMY OF SHINY FUNCTIONS: DATA FLOW



"dynamic" output means that the user can change something about the tables or graphs by selecting new input parameters

ANATOMY OF SHINY FUNCTIONS: USER INPUT



"dynamic" output means that the user can change something about the tables or graphs by selecting new input parameters

INPUT WIDGETS

Building the user interface to gather user inputs

STANDARD SHINY WIDGETS (INPUTS)

Function Name	Widget
actionButton	Action Button
checkboxGroupInput	A group of check boxes
checkboxInput	A single check box
dateInput	A calendar to aid date selection
dateRangeInput	A pair of calendars for selecting a date range
fileInput	A file upload control wizard
helpText	Help text that can be added to an input form
numericInput	A field to enter numbers
radioButtons	A set of radio buttons
selectInput	A box with choices to select from
sliderInput	A slider bar
submitButton	A submit button
textInput	A field to enter text

Note each function will store different input values:

textInput = a single character element

selectInput = character elements from a list

sliderInput = two numbers in a range

checkboxInput = T / F

Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Action

Current Value:

```
[1] 0
attr(,"class")
[1] "integer"                "shinyActionButtonValue"
```

See Code

Single checkbox

- [x] Choice A

Current Value:

```
[1] TRUE
```

See Code

Checkbox group

- [x] Choice 1
- [ ] Choice 2
- [ ] Choice 3

Current Values:

```
[1] "1"
```

See Code

Date input

2014-01-01

Current Value:

```
[1] "2014-01-01"
```

See Code

Date range

2019-09-16 to 2019-09-16

Current Values:

```
[1] "2019-09-16" "2019-09-16"
```

See Code

File input

Browse... No file selected

Current Value:

```
NULL
```

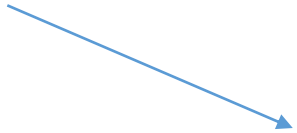
See Code

WIDGET COMPONENTS

- **Name** for the widget. The user will not see this name, but you can use it to access the widget's value. The name should be a character string.

- **Label**. This label will appear with the widget in your app. It should be a character string, but it can be an empty string "".

actionButton(name="submit", label = "Submit Your Form")



Creates an entry at input$submit

How you will access the data:

input$name

Note that you do not create the input object and assign values at input$widget_name. That is done for you by the Shiny package.

ANATOMY OF SHINY FUNCTIONS: USER INPUT