Interactive Documents and Applications with R

Dominic LaRoche

October 21, 2015

Outline

- Introduction

- Examples

- Nuts and Bolts

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Section 1

Introduction

Mel & Enid Zuckerman
College of Public Health

What are interactive documents and applications?

Interactive documents and applications are graphical user interfaces which run analyses in the background.

- Allow user to define the inputs for analyses

- Run pre-defined analyses

- Provide results

How are interactive documents and applications beneficial?

Interactive Applications:

- Streamline repetetive analyses and reduce errors

How are interactive documents and applications beneficial?

Interactive Applications:

- Streamline repetetive analyses and reduce errors

- Increase efficiency of an organization by enabling others to perform and interpret simple analyses

Mel & Enid Zuckerman
College of Public Health

How are interactive documents and applications beneficial?

Interactive Applications:

- Streamline repetetive analyses and reduce errors

- Increase efficiency of an organization by enabling others to perform and interpret simple analyses

- Provide a high-value service to customers

How are interactive documents and applications beneficial?

Interactive Applications:

- Streamline repetetive analyses and reduce errors

- Increase efficiency of an organization by enabling others to perform and interpret simple analyses

- Provide a high-value service to customers

Interactive Documents:

- Provide a tool to both inform *and* educate

How are interactive documents and applications beneficial?

Interactive Applications:

- Streamline repetetive analyses and reduce errors

- Increase efficiency of an organization by enabling others to perform and interpret simple analyses

- Provide a high-value service to customers

Interactive Documents:

- Provide a tool to both inform *and* educate
- Allow consumers to ask and answer their own questions
  - Facilitate discovery
  - Understand the "why" and not just the "what"

How are interactive documents and applications beneficial?

Other uses:

- Teaching

Mel & Enid Zuckerman
College of Public Health

How are interactive documents and applications beneficial?

Other uses:

- Teaching

- Marketing

Mel & Enid Zuckerman
College of Public Health

How are interactive documents and applications beneficial?

Other uses:

- Teaching

- Marketing

- Public health

Mel & Enid Zuckerman
College of Public Health

Interactive Applications Basics

Written in R:

- Can do anything you can do in R

- Large library of pre-written widgets for user interaction

- Can accomodate anything you can write in HTML, Java, Python, etc.

- Can be hosted locally and run behind a firewall

- Use shiny.io for free (up to 5 apps)

Interactive Reports Basics

Written in Rmarkdown:

- Combines report text and code into a single document

- Can do anything you can do in R

- Felixble formatting

- Can easily convert between document types

- Can create a static document from interactive one

- Document must be hosted
  - On the web
  - Locally as application

Section 2

Examples

Examples

- Interactive Application

- Interactive Document

- Shiny Example Library
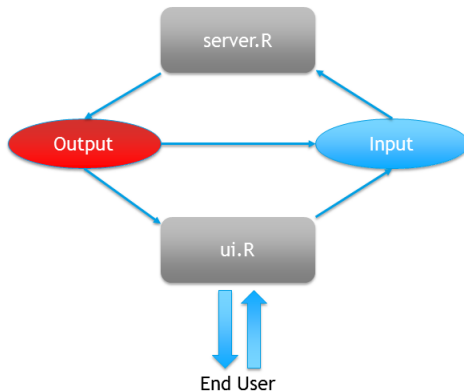
Section 3

Nuts and Bolts

WARNING!

I will be discussing some details of creating these type of applications and documents which will involve talking about code and programming.
Feel free to escape!

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Architecture of an Application

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Building an Application

Applications depend on two main list-like objects:

- input
  - Stores all the user inputs for use in the analysis
  - Data, parameters, etc.

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Building an Application

Applications depend on two main list-like objects:

- input
  - Stores all the user inputs for use in the analysis
  - Data, parameters, etc.

- output
  - Stores all of the outputs from the analysis
  - Everything returned to the user

Building an Application

Applications use two main functions from the shiny package:

- ShinyServer
    - Controls the server logic
        - What happens and in what order
    - Calls sub-functions or other programs
    - Creates dynamic inputs
    - uses "inputs" to create "outputs"

Building an Application

Applications use two main functions from the shiny package:

- ShinyServer
  - Controls the server logic
    - What happens and in what order
  - Calls sub-functions or other programs
  - Creates dynamic inputs
  - uses "inputs" to create "outputs"

- ShinyUI
  - Defines the appearance of the application
  - Interacts with the user to collect "inputs"
  - Displays all the "outputs" of the analysis

Pre-packaged Inputs

Many pre-defined inputs to use:

- textInput()

- numericInput()

- checkboxInput()

- sliderInput()

- selectInput()

- fileInput()

- tableInput()

- Many more...

Users can also define their own inputs with HTML, java, etc.

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Pre-packaged Outputs

An equally large number of predefined outputs for rendering results

- textOutput()
- plotOutput()
- File downloads
- etc.

Outputs are also customizable.

Pre-packaged Outputs

An equally large number of predefined outputs for rendering results

- textOutput()
- plotOutput()
- File downloads
- etc.

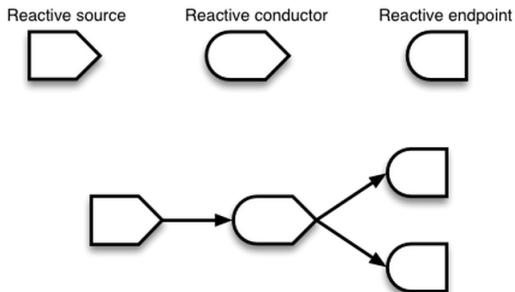Outputs are also customizable.

Two steps for rendering outputs:

1. To assign to the output "list": render...() (renderText, renderTable, etc.)
2. To display in the application: ...Output() (textOutput, tableOutput, etc.)

Reactivity

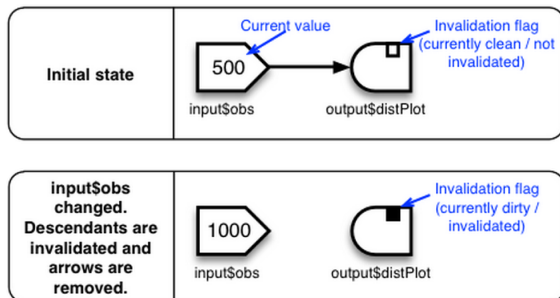The analysis must respond to updated user inputs. This is known as **reactivity**.

- 3 main components for reactivity
    1.) Reactive source (generally an input)
    2.) Reactive conductor (usually a function)*
    3.) Reactive endpoint (usually an output)

- Trickiest part for typical R programmer

- All manipulations of reactive values must be done inside the reactive environment

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Reactivity

From Rstudio folks

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health
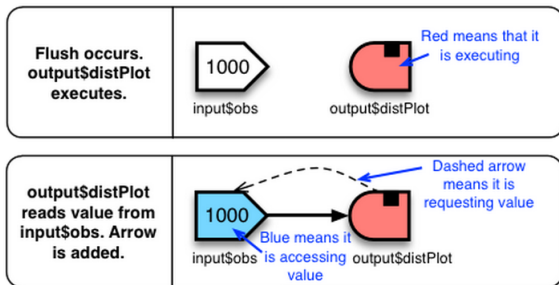
Reactivity: Execution Scheduling

From Rstudio folks

Reactivity: Execution Scheduling

From Rstudio folks

Mel & Enid Zuckerman
College of Public Health

Reactivity

Some of the tricky things about reactivity:

- A reactive function will not execute if there is no reactive endpoint

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Reactivity

Some of the tricky things about reactivity:

- A reactive function will not execute if there is no reactive endpoint

- A reactive value is always reacting to user inputs
  - Some inputs may have no initial value
  - Other inputs may require typing

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Reactivity

Some of the tricky things about reactivity:

- A reactive function will not execute if there is no reactive endpoint

- A reactive value is always reacting to user inputs
  - Some inputs may have no initial value
  - Other inputs may require typing

- Can use observer() function as an endpoint if no output is required

Reactivity

Some of the tricky things about reactivity:

- A reactive function will not execute if there is no reactive endpoint

- A reactive value is always reacting to user inputs
  - Some inputs may have no initial value
  - Other inputs may require typing

- Can use observer() function as an endpoint if no output is required

- isolate() function + action button can control unwanted reactivity

Reactivity

Some of the tricky things about reactivity:

- A reactive function will not execute if there is no reactive endpoint

- A reactive value is always reacting to user inputs
    - Some inputs may have no initial value
    - Other inputs may require typing

- Can use observer() function as an endpoint if no output is required

- isolate() function + action button can control unwanted reactivity

- validate() function can require certain conditions are met before reacting

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Dynamic Applications

Extensive applications should be dynamic

- Limit the number of inputs to focus the user

Dynamic Applications

Extensive applications should be dynamic

- Limit the number of inputs to focus the user

- Ask for inputs as needed and build up options

Dynamic Applications

Extensive applications should be dynamic

- Limit the number of inputs to focus the user

- Ask for inputs as needed and build up options

- Make inputs from reactive outputs

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Building an Interactive Document

Interactive document is a special case of an application

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Building an Interactive Document

Interactive document is a special case of an application

- Written in Rmarkdown
  - Markup language (think LaTeX) that weaves code and text into single document

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Building an Interactive Document

Interactive document is a special case of an application

- Written in Rmarkdown
  - Markup language (think LaTeX) that weaves code and text into single document

- Output controlled by header: word, pdf, latex presentation, io slides, and HTML

Building an Interactive Document

Interactive document is a special case of an application

- Written in Rmarkdown
  - Markup language (think LaTeX) that weaves code and text into single document

- Output controlled by header: word, pdf, latex presentation, io slides, and HTML

- Specify HTML with shiny runtime to get interactive document.

Building an Interactive Document

Rmarkdown:

- Very simple markup (no need to know HTML)

- Can augment with HTML (or other 'languages')

- Easy to switch outputs

- Reproducible and limits errors

Building an Interactive Document

Some differences from a standard application:

Building an Interactive Document

Some differences from a standard application:

- Only 1 file (no separate server.R and ui.R files)

Building an Interactive Document

Some differences from a standard application:

- Only 1 file (no separate server.R and ui.R files)

- Use inputPanel() instead of shinyUI() function

Mel & Enid Zuckerman
College of Public Health

Building an Interactive Document

Some differences from a standard application:

- Only 1 file (no separate server.R and ui.R files)

- Use inputPanel() instead of shinyUI() function

- Outputs are rendered in 2 stages:
    1.) Rendered into HTML by shiny package
    2.) Placed into document via Rmarkdown arguments

THE UNIVERSITY OF ARIZONA
Mel & Enid Zuckerman
College of Public Health

Questions?

"No mom, we didn't do our abc's today, we just reviewed our R, Rmarkdown, and HTML...