# An overview of Shiny applications using R and RStudio

Marcus W. Beck[1]

[1]USEPA National Health and Environmental Effects Research Laboratory, Gulf Ecology Division, beck.marcus@epa.gov

Dec. 10, 2015

# Who am I?

- ORISE post-doc for 2.5 years, fed postdoc since last week
- NHEERL Gulf Ecology Division
- Research focus on water quality assessment and indicator development
- Specific interests in statistical modelling, data assimilation, graphics

# Who am I?

- R user since 2007
- Maintainer of two packages on CRAN:

***SWMPr***

Tools for retrieving, organizing, and analyzing data from the System Wide Monitoring Program of the National Estuarine Research Reserve System.

***NeuralNetTools***

Visualization and analysis tools to aid in the interpretation of neural network models

# Reproducible research workflow

General workflow for ***reproducible research*** - reproduce results from an experiment or analysis conducted by another.

From Wikipedia... ‘The ultimate product is the ***paper along with the full computational environment*** used to produce the results in the paper such as the code, data, etc. that can be ***used to reproduce the results and create new work*** based on the research.’

# Reproducible research workflow

The use of these tools increases transparency and transfer of information = ***better science***

Data prep, analysis, report, and sharing can all be done in RStudio IDE

# Introduction to Shiny

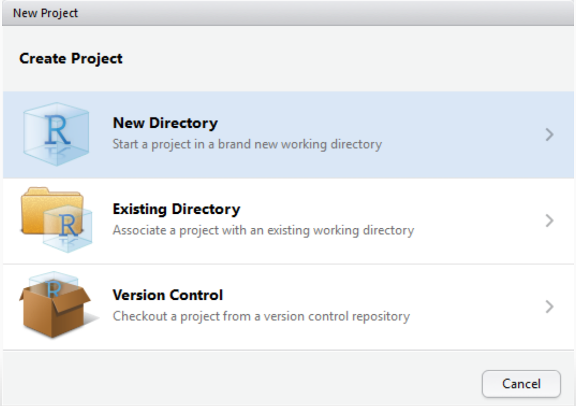
Where does Shiny fit with reproducible research?

Shiny is a web application framework for R

- From the command line to a graphical user interface
- Make your code interactive
- Do not need to know anything about web programming
- Integrated very well with R studio

Tools like Shiny improve ***accessibility*** and ***communication***

## A minimal working example...

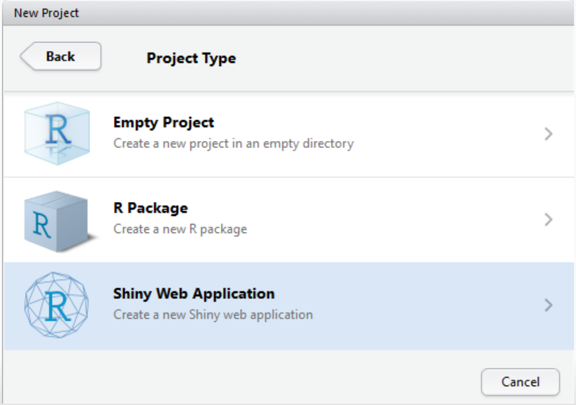


New Project

Create Project

New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

A minimal working example...



New Project

Back

Project Type

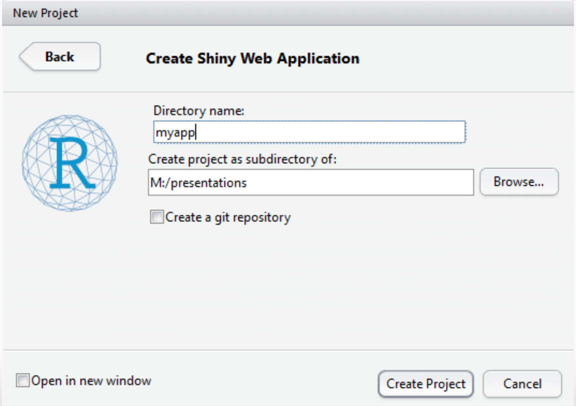
Empty Project
Create a new project in an empty directory

R Package
Create a new R package

Shiny Web Application
Create a new Shiny web application

Cancel

A minimal working example...



New Project

Back

Create Shiny Web Application

Directory name: myapp

Create project as subdirectory of: M:/presentations

Browse...

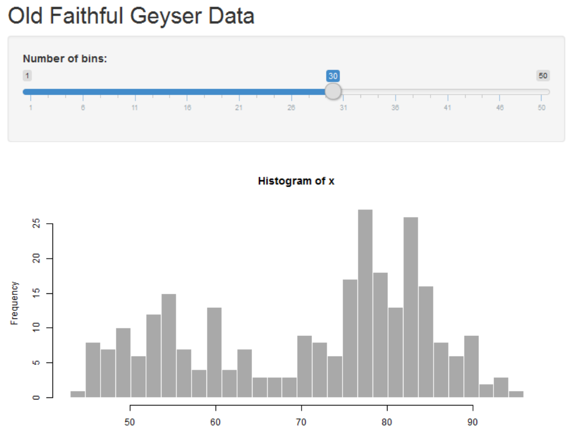
□ Create a git repository

□ Open in new window

Create Project

Cancel

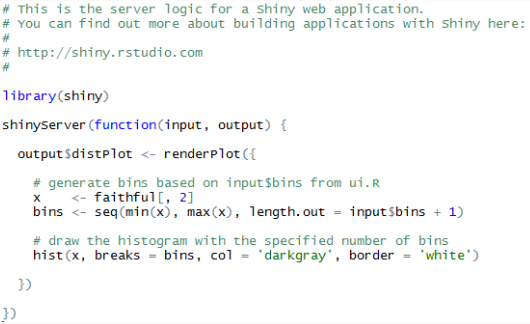
## A minimal working example...



Old Faithful Geyser Data

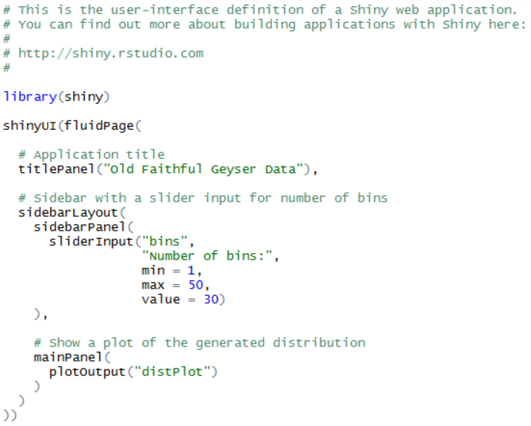
Number of bins: 30

What's under the hood? Two files... ***server.R***



```r
# This is the server logic for a Shiny web application.
# You can find out more about building applications with Shiny here:
#
# http://shiny.rstudio.com
#

library(shiny)

shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    # generate bins based on input$bins from ui.R
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkgray', border = 'white')

  })

})
```

What's under the hood? Two files... ***ui.R***



```
# This is the user-interface definition of a Shiny web application.
# You can find out more about building applications with Shiny here:
#
# http://shiny.rstudio.com
#

library(shiny)

shinyUI(fluidPage(

  # Application title
  titlePanel("Old Faithful Geyser Data"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("distPlot")
    )
  )
))
```

# Introduction to Shiny

The files contain only R code!

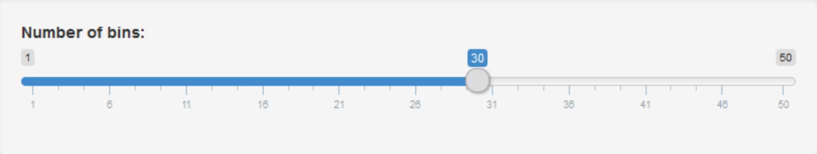
- ***server.R***: Contains instructions to build the content, e.g., plots, functions, etc.

- ***ui.R***: Controls the layout and appearance of the app, i.e., panel types, widgets, etc.

Executing a Shiny app will run both scripts, user input to ***ui.R*** sent to ***server.R***, output from ***server.R*** sent to ***ui.R*** for display

Step 1: User input to ***ui.R***, ‘bins’

```
sliderInput("bins",
            "Number of bins:",
            min = 1,
            max = 50,
            value = 30)
```



Number of bins:

# Introduction to Shiny

Step 2: Input from ***ui.R*** sent to ***server.R***, executed

```
# generate bins based on input from ui.R
x <- faithful[, 2]
bins <- seq(min(x), max(x), length.out = input$bins + 1)

# draw the histogram with the specified number of bins
hist(x, breaks = bins, col = 'darkgray', border = 'white')
```

Step 3: Output from ***server.R*** sent to ***ui.R***, plotted on app

```
plotOutput("distPlot")
```

Step 4: Rinse and repeat

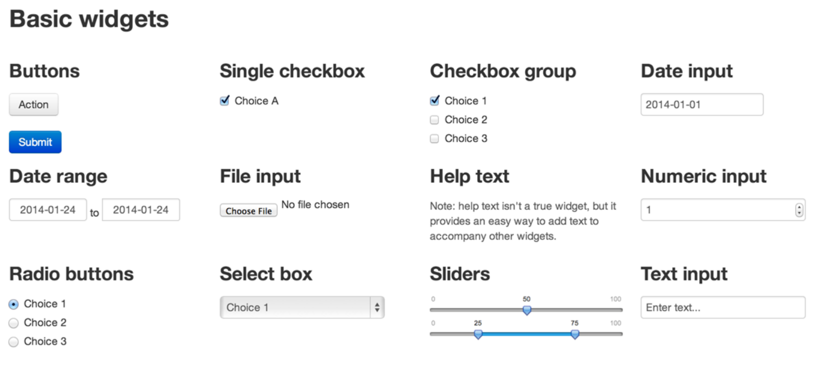
# Introduction to Shiny

This style of programming and execution is ***reactive*** - re-executes automaticallly when inputs change

This has tremendous value:

- Quick code execution after initial setup
- Ease of use for others given application infrastructure
- Ease of use for the developer - no knowledge of web programming needed

## Shiny applications are very flexible: widgets



### Basic widgets

**Buttons**

**Single checkbox**

☑ Choice A

**Checkbox group**

☑ Choice 1
□ Choice 2
□ Choice 3

**Date input**

2014-01-01

**Date range**

2014-01-24 to 2014-01-24

**File input**

No file chosen

**Help text**

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

**Numeric input**

1

**Radio buttons**

● Choice 1
○ Choice 2
○ Choice 3

**Select box**

Choice 1

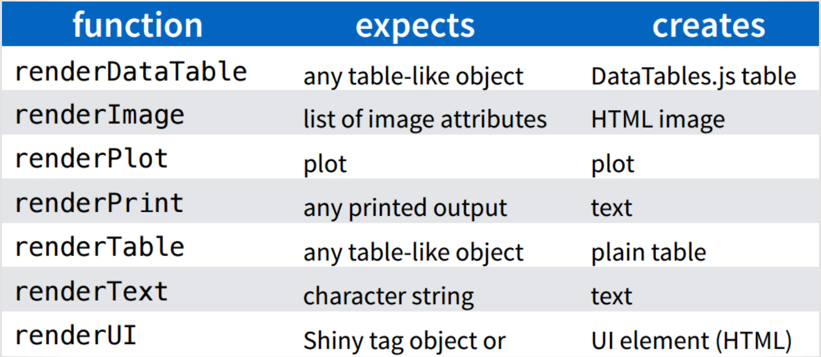
**Sliders**

**Text input**

Enter text...

# Introduction to Shiny

Shiny applications are very flexible: outputs



| function | expects | creates |
|---|---|---|
| `renderDataTable` | any table-like object | DataTables.js table |
| `renderImage` | list of image attributes | HTML image |
| `renderPlot` | plot | plot |
| `renderPrint` | any printed output | text |
| `renderTable` | any table-like object | plain table |
| `renderText` | character string | text |
| `renderUI` | Shiny tag object or | UI element (HTML) |

# Introduction to Shiny

Shiny applications are very flexible: use of HTML or Javascipt libraries

- Refined layouts: shinydashboard, htmlwidgets, shinyBS
- Interacive graphics: dygraphs, metricsgraphics, plotly
- Mapping: leaflet

Use of these libraries can create applications comparable to any other web application for data viz

# Introduction to Shiny

Apps are easily shared....

For the ***single*** user:

- Local RStudio ‘project’ as a standalone working directory

For ***multiple*** users:

- As a web application on your server, requires Shiny Server
- As a web application hosted on `http://www.shinyapps.io/`

# Shiny Examples

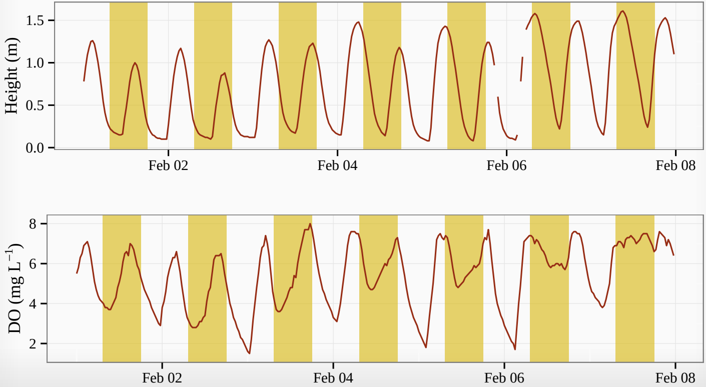
Research application: Evaluating a statistical model to isolate and remove variance components from a time series

***Scenario***: Time series of dissolved oxygen provide information on ecosystem processes in aquatic systems

***Problem***: These time series are assumed to measure biological production/respiration, but noise from tidal cycles in coastal systems

# Shiny Examples

A tidal and dissolved oxygen time series at Sapelo Island, Georgia



# Shiny Examples

Grid-based evaluation of the statistical model using simulated time series and varying model paremeters

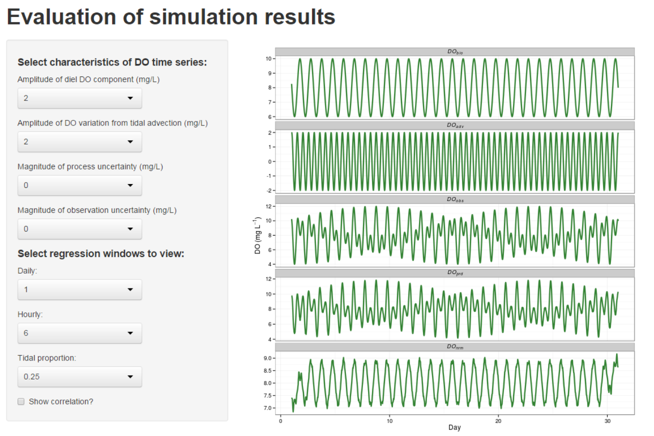
- Time series varying by 4 characteristics, 3 levels per characteristic

- Model parameters varying by 3 characeristics, 3 levels per characterisic

2187 unique combinations: Very challenging to evaluate results, use Shiny!

# Shiny Examples

https://beckmw.shinyapps.io/detiding_sims/



## Evaluation of simulation results

**Select characteristics of DO time series:**

Amplitude of diel DO component (mg/L): 2

Amplitude of DO variation from tidal advection (mg/L): 2

Magnitude of process uncertainty (mg/L): 0

Magnitude of observation uncertainty (mg/L): 0

**Select regression windows to view:**

Daily: 1

Hourly: 6

Tidal proportion: 0.25

□ Show correlation?

# Shiny Examples

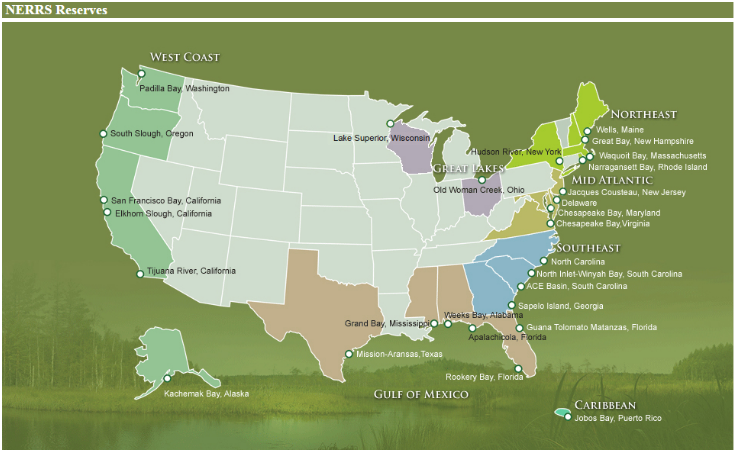
Management application: Spatial and temporal assessment of water quality trends in NOAA estuary reserves

**NERRS**
National Estuarine Research Reserve System, established by Coastal Zone Management Act of 1972. Focus on ***long-term research***, ***monitoring***, ***education***, and ***stewardship*** for more effective coastal management.

**SWMP**
System Wide Monitoring Program, initiated in 1995 to provide ***continuous monitoring*** data at over 140 stations in each of the 28 NERRS reserves

# Shiny Examples

Location of NERRS estuary reserves with SWMP data



# Shiny Examples

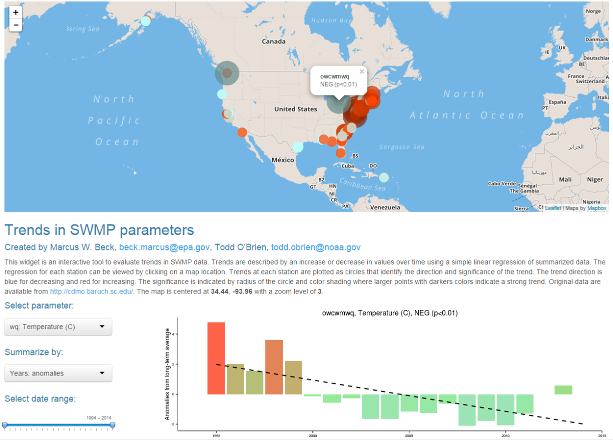
Although SWMP data have been collected and processed using standardized methods...

- Long-term trends have not been evaluated between-reserves in several years
- Tools for simple trend analysis and visualization have been lacking

***Solution***: Use Shiny to bring results to users!

# Shiny Examples

`https://beckmw.shinyapps.io/swmp_comp/`



## Trends in SWMP parameters

Created by Marcus W. Beck, beck.marcus@epa.gov, Todd O'Brien, todd.obrien@noaa.gov

This widget is an interactive tool to evaluate trends in SWMP data. Trends are described by an increase or decrease in values over time using a simple linear regression of summarized data. The regression for each station can be viewed by clicking on a map location. Trends at each station are plotted as circles that identify the direction and significance of the trend. The trend direction is blue for decreasing and red for increasing. The significance is indicated by radius of the circle and color shading where larger points with darkers colors indicate a strong trend. Original data are available from http://cdmo.baruch.sc.edu/. The map is centered at **34.44**, **-93.96** with a zoom level of **3**.

Select parameter: wq: Temperature (C)

Summarize by: Years: anomalies

Select date range: 1994 – 2014

owcwmwq, Temperature (C), NEG (p<0.01)
Anomalies from long-term average
4
2
0
-2
1995
2000
2005
2010
2015

# Conclusions

Shiny has multiple benefits:

- Increased accessibility to information within and outside of the research community
- Available for use with minimal or no experience in web programming
- All open-source, no need for license and under active development

Opportunities for EPA Shiny Server?

# Additional resources

RStudio Shiny tutorial:
`http://shiny.rstudio.com/tutorial/`

RStudio Shiny gallery:
`http://shiny.rstudio.com/gallery/`

Deploy Shiny apps:
`http://www.shinyapps.io/`

This presentation (`beck.marcus@epa.gov`):
`https://github.com/fawda123/shiny_pres/raw/master/shiny_pres.pdf`