



PhUSE US Connect 2020

End to End Interactive TLF using R Shiny



Bayer Inc.

Rohit Banga





Agenda

- // Introduction
- // Shiny App Elements
- // Shiny App Setup & Programs Walkthrough
- // Deploy Shiny App
 - // Internal Server
 - // Shinyapps.io



Introduction

R Shiny + Demo Files

- // R Shiny is a powerful web application framework
- // R Studio's Shiny server makes shiny applications available over the web.
- // GitHub Code -
 - // <https://github.com/rohitbanga/rshinytlf>



Setup Shiny App

Elements

// Working Demo

// <https://phusedemo.shinyapps.io/RShinyTLF/>

5. Dropdown

2. Dashboard Items

3. Button to Load Data

4. Download Files Button

1. Shiny App

6. TLF Output

R Shiny TLF Demo

Study Name:
Demo Study 1

Table 1 - Demographic Information

Copy CSV PDF Print Excel

Search:

	sort	category	type	Placebo	Xanomeline High Dose	Xanomeline Low Dose
1	Gender		F	53	40	50
1	Gender		M	33	44	34
2	Age Group		<65	14	11	8
2	Age Group		>80	30	18	29
2	Age Group		65-80	42	55	47
3	Race		AMERICAN INDIAN OR ALASKA NATIVE		1	
3	Race		BLACK OR AFRICAN AMERICAN	8	9	6
3	Race		WHITE	78	74	78
4	Ethnicity		HISPANIC OR LATINO	3	3	6
4	Ethnicity		NOT HISPANIC OR LATINO	83	81	78

Showing 1 to 10 of 10 entries

Refresh Database

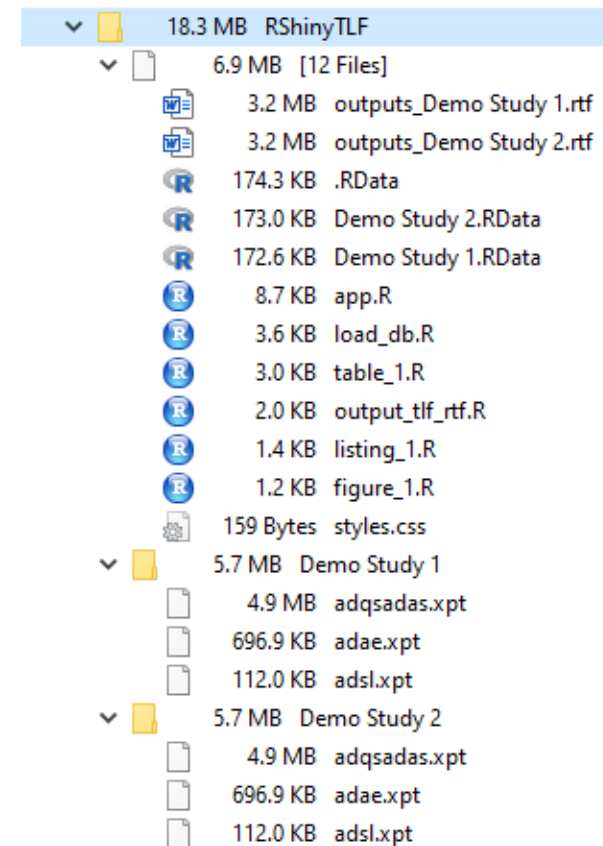
Download All Files



Setup Work Directory

Step 1

- // Make directory to store files - “RShinyTLF”
- // The example directory stores -
 - // Programs –
 - // Main program - app.R
 - // Programs to make TLF - table_1.R, figure_1.R, listing_1.R
 - // Program to load data – load_db.R
 - // Program to convert R data frame to RTF – output_tlf_rtf.R
 - // Study Data as XPT files - In folders Demo Study 1, Demo Study 2
 - // Study Data as R Data Frames – Demo Study 1.RData, Demo Study 2.RData
 - // Output Files – outputs_Demo Study 1.rtf, outputs_Demo Study 2.rtf
 - // Styles.css





Setup Shiny Dashboard (app.R)

Step 2

// Shiny Libraries –

// shiny

// shinydashboard

// shinyWidgets, shinyjs, Shinycssloaders

// Server Components

// observerEvent

// output\$<output name>

// Dashboard Components

// UI Components

// dashboardPage

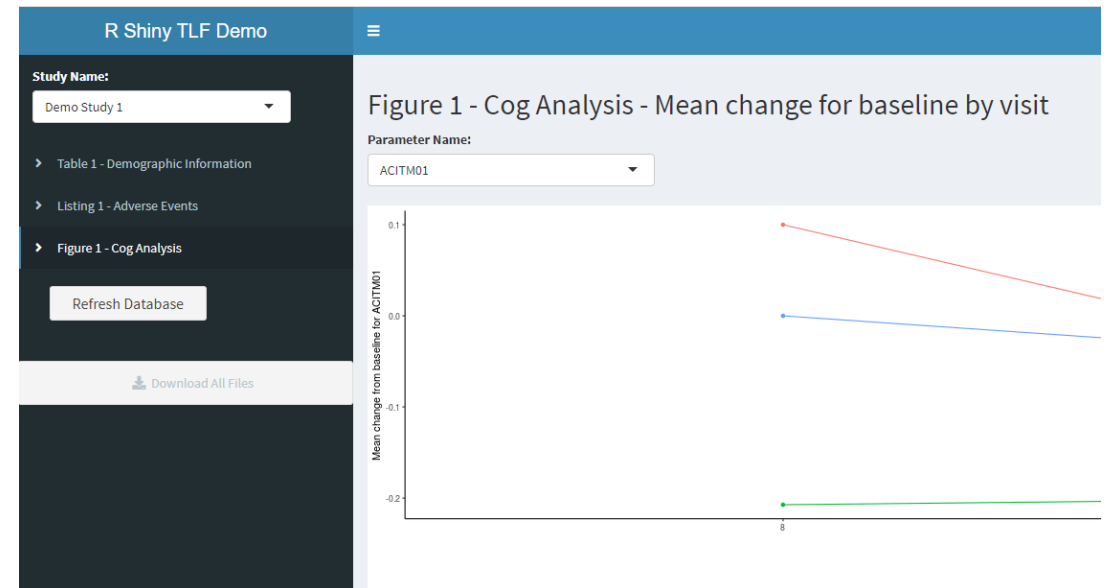
// dashboardSidebar

// sidebarMenu

// menuItem

// dashboardBody

// tabItems





Refresh Data Button

Step 3 - app.R / load_db.R

// app.R

// UI Component

// actionButton

// Server Component

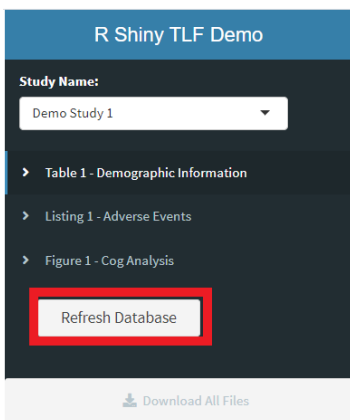
// observerEvent

// input\$refresh_btn

// showModal

// source(load_db.R)

// session\$reload()



// load_db.R

// Load SAS xpt files - sasxport.get function (Hmisc library)

// Run TLF programs - source(table_1.R)

// Save study data image in current directory

// Run code to output TLF as RTF - source("output_tlf_rtf.R")

Global Environment ▾	
Data	
▶ adae	1191 obs. of 55 variables
▶ adqsadas	12463 obs. of 40 variables
▶ ads1	254 obs. of 48 variables
▶ fig_1	List of 9
▶ figure_1	12463 obs. of 13 variables
▶ legend_11	1 obs. of 2 variables
▶ listing_1	1191 obs. of 19 variables
▶ table_1	10 obs. of 6 variables



Download TLF

Step 4 - app.R/output_tlf_rtf.R

// app.R

// UI Component

// downloadButton

// Server Component

// downloadHandler

// file.copy

// output_tlf_rtf.R

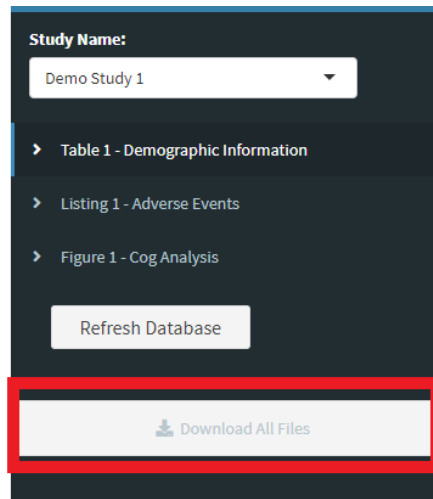
// RTF function from rtf library

// addHeader

// addParagraph

// addTable

// addPlot



Section 1 - Tables

This Section contains all tables

Table 1 - Demographic Data:

category	type	Placebo	Xanomeline	Xanomeline
			High Dose	Low Dose
Gender	F	53	40	50
Gender	M	33	44	34
Age Group	<65	14	11	8
Age Group	>80	30	18	29
Age Group	65-80	42	55	47
Race	AMERICAN INDIAN OR ALASKA NATIVE	-	1	-
Race	BLACK OR AFRICAN AMERICAN	8	9	6
Race	WHITE	78	74	78
Ethnicity	HISPANIC OR LATINO	3	3	6
Ethnicity	NOT HISPANIC OR LATINO	83	81	78



Dropdown for Selecting Studies

Step 5 – app.R

// UI Component

// selectInput

// Server Component

// observeEvent(input\$study) – triggered each time study selected from dropdown changes

// Run load_db.R if study data image not found

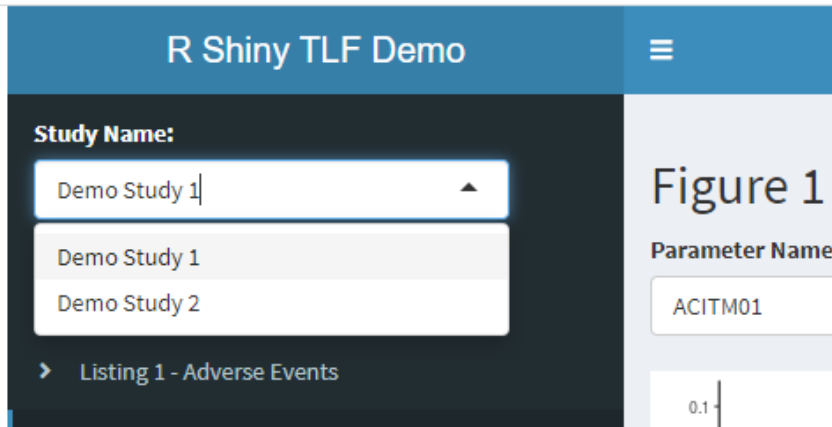
// Load study data Image

// Revisualize outputs

// output\$table_1

// output\$listing_1

// output\$figure_1





Display TLF on Dashboard - UI

Step 6

// app.R

// UI Component

// sidebarMenu

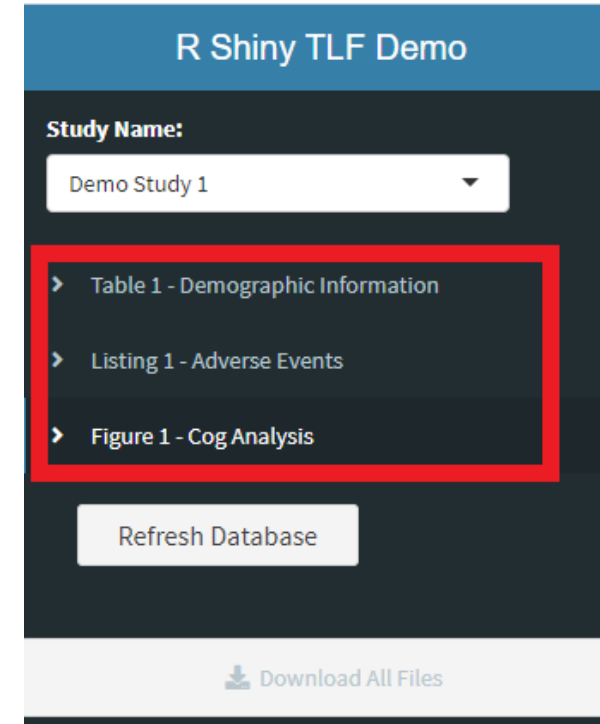
// menuItem – tabName

// dashboardBody

// tabItem - tabName

// h2 - Heading

// DT:dataTableOutput





Display TLF on Dashboard - Server

Step 6

// app.R

// Server Component

// observeEvent(input\$study)

// renderDataTable

// buttons

// scrollX

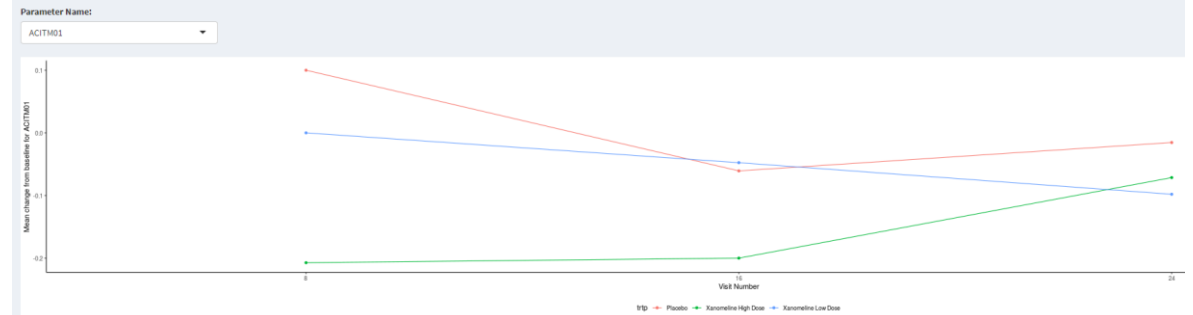
// fixedColumns

// formatStyle – Conditional Formatting

Listing 1 - Adverse Event

usubjid	trta	sex	agegr1	saffl	aeterm	aedecod	aesev	aeser	aescan	aescong	aedisab
01-709-1424	Xanomeline High Dose	M	65-69	Y	SYNCOPE	SYNCOPE	MODERATE	Y	N	N	N
01-718-1170	Xanomeline Low Dose	F	65-69	Y	SYNCOPE	SYNCOPE	SEVERE	Y	N	N	N
01-718-1171	Xanomeline High Dose	F	65-69	Y	PARTIAL SEIZURES WITH SECONDARY GENERALIZATION	PARTIAL SEIZURES WITH SECONDARY GENERALIZATION	SEVERE	Y	N	N	N
01-701-1015	Placebo	F	<65	Y	APPLICATION SITE ERYTHEMA	APPLICATION SITE ERYTHEMA	MILD	N	N	N	N
01-701-1015	Placebo	F	<65	Y	APPLICATION SITE PRURITUS	APPLICATION SITE PRURITUS	MILD	N	N	N	N
01-701-1015	Placebo	F	<65	Y	DIARRHOEA	DIARRHOEA	MILD	N	N	N	N
01-701-1023	Placebo	M	<65	Y	ERYTHEMA	ERYTHEMA	MILD	N	N	N	N
01-701-1023	Placebo	M	<65	Y	ERYTHEMA	ERYTHEMA	MODERATE	N	N	N	N

Figure 1 - Cog Analysis - Mean change for baseline by visit





Publish Shiny App on Self Hosted Server

Step 7a

- // Setup up an Ubuntu Virtual Machine on a cloud provider – AWS, Azure or Google Cloud Platform
- // Open Port 22 (For SSH) and 3838 (For R Studio Shiny Server)
- // Install R and R Shiny Server on virtual machine
- // Check whether you can see welcome to R Shiny Server page on
 - // *<http://<Your Virtual Machine IP Address>:3838/>*
- // Install R Packages
- // Copy RShinyTLF from local desktop to */srv/shiny-server* directory on virtual machine
- // Update shiny server configuration file in */etc/shiny-server/shiny-server.conf*
- // Restart shiny-server
- // Our app is visible on –
 - // *<http://<Your Virtual Machine IP Address>:3838/>*



Publish shiny app on Shinyapps.io

Step 7b

- // Sign up for an account on <https://www.shinyapps.io/>
- // Note down the token and secret from the shinyapps.io admin portal
- // In your local R Studio, click on the blue icon and click publish application
- // In “Publish From Account”, click on “Add new account” → ShinyApps.io and paste the code copied above
- // Select the connected account and click on Publish

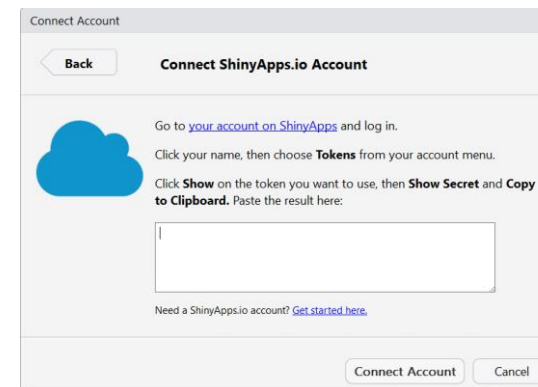
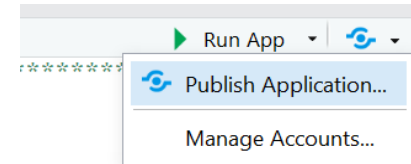
The `shinyapps` package must be authorized to your account using a token and secret. To do this, click the copy button below and we'll copy the whole command you need to your clipboard. Just paste it into your console to authorize your account. Once you've entered the command successfully in R, that computer is now authorized to deploy applications to your shinyapps.io account.

```
rsconnect::setAccountInfo(name="phusedemo",  
  token="DD0763AD00952B3FB63BA9CED0B790C8",  
  secret="<SECRET>")
```

Show secret

Copy to clipboard

OK





Thank you!



Bye-Bye

