

R Shinyの紹介

DS部会2024年度継続タスクフォース11
松永 友貴 (ノバルティス ファーマ株式会社)
生井 伴幸 (中外製薬株式会社)

Shinyとは？

- Shinyは、Posit PBC（旧Rstudio Inc.）によって開発された、対話型ウェブアプリケーションを簡単に構築できるパッケージである。
- 2012年にRパッケージとしてのShinyがリリースされ、2022年にはPythonライブラリとしてもShinyがリリースされている。



Easy web apps for data science without the compromises

No web development skills required



Get started in R

Get started in Python



Here is a Shiny app

Shiny apps are easy to write. Let users interact with your data and your analysis, all with R or Python:

```
R Python
app.R
library(shiny)
library(bslib)
library(dplyr)
library(ggplot2)
library(ggExtra)

penguins_csv <- "https://raw.githubusercontent.com/jcheng5/simplepenguins.R/main/penguins.csv"

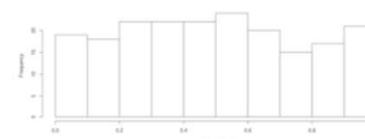
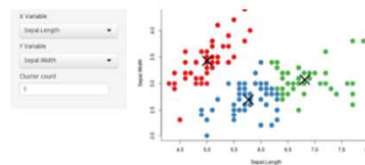
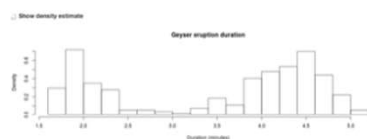
df <- readr::read_csv(penguins_csv)
```

Feature Demos

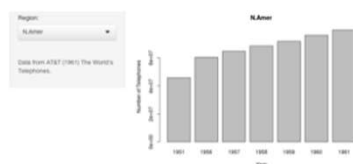
Shiny Demos are a series of apps created by the Shiny developers and are designed to highlight specific features of the shiny package. Each app is presented along with its source code to help you implement these features in your apps. Many of these apps are linked from relevant articles as well.

Start Simple

If you're new to Shiny, these simple but complete applications are designed for you to learn from.



Faithful



Kmeans example



Single-file shiny app





ARTICLES

Start

Build

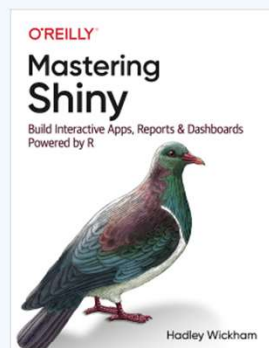
Improve

Share



A comprehensive guide to using Shiny. If you are just starting out, you may want to explore the [Get Started Tutorials](#) to learn the basics. Otherwise, choose your own path with the articles below!

Start



THE BEST PLACE TO START

Mastering Shiny

By Hadley Wickham

A free, online book designed to take you from basic Shiny to creating your own customized apps.

[Read Now](#)

If you are beginning to learn Shiny for R, here are a few resources you'll want to consider:

- Hadley Wickham's [Mastering Shiny book](#)
- Shiny for R's [Get Started Tutorials](#)
- The [Shiny Cheat Sheet](#)
- Posit's [Demos and Tutorials on YouTube](#)

Shiny基本構造

- Shiny アプリを動かすには，基本的な構造として，データ処理のための**server**とパラメータ操作のための**user interface**の 2 つのファイルを作る必要がある。

```
#server
server <-
  shinyServer(function(input, output) {
    output$table = renderDataTable(admiral::admiral_adsl, filter = "top"
  })

#user interface
ui <-
  shinyUI(fluidPage(
    titlePanel("ADSL"),
    dataTableOutput("table")
  ))

# run the app
shinyApp(ui, server)
```

Shiny基本構造

ADSL

Show entries

Search:

	STUDYID	USUBJID	SUBJID	RFSTDTC	RFENDTC	RFXSTDTC	RFXENDTC	RFICDTC	RFPENDTC	DTHDTC	DTHFL
	All	All	A	All	All	All	All	All	All	All	All
1	CDISCPIL0T01	01-701-1015	1015	2014-01-02	2014-07-02	2014-01-02	2014-07-02		2014-07-02T11:45		
2	CDISCPIL0T01	01-701-1023	1023	2012-08-05	2012-09-02	2012-08-05	2012-09-01		2013-02-18		
3	CDISCPIL0T01	01-701-1028	1028	2013-07-19	2014-01-14	2013-07-19	2014-01-14		2014-01-14T11:10		
4	CDISCPIL0T01	01-701-1033	1033	2014-03-18	2014-04-14	2014-03-18	2014-03-31		2014-09-15		
5	CDISCPIL0T01	01-701-1034	1034	2014-07-01	2014-12-30	2014-07-01	2014-12-30				

非常に短いスクリプトで、検索、ソーティング、フィルタリングが行えるデータテーブルのアプリケーションを表示させることができる。

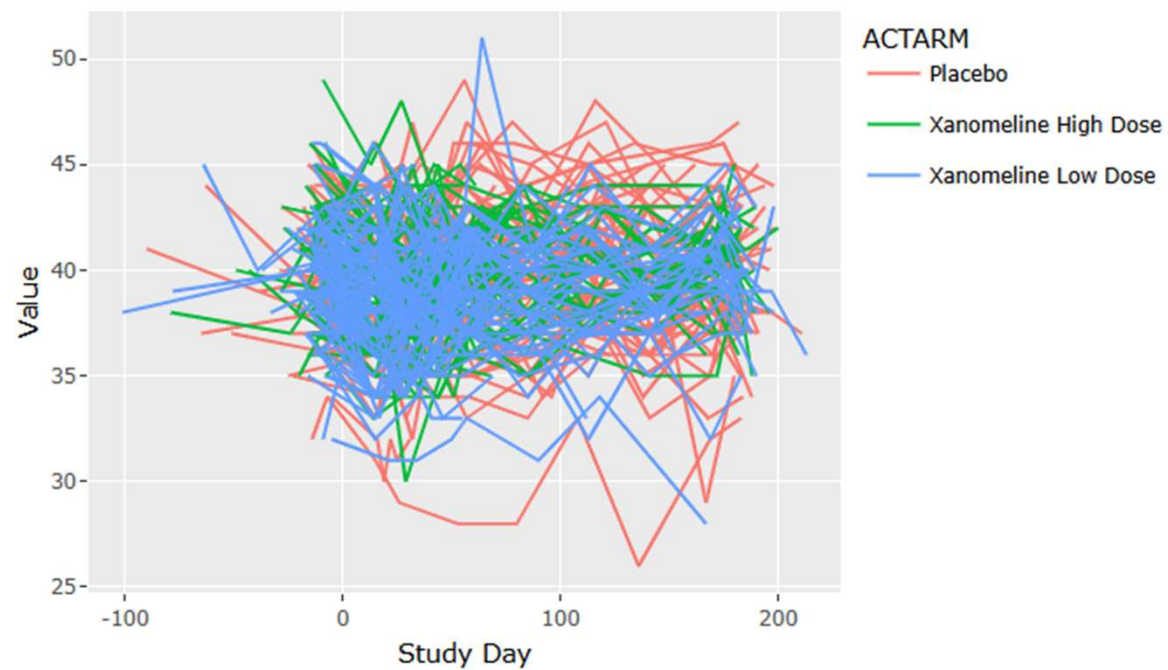
例：臨床検査値のプロット

アプリケーション

Laboratory Data (SDTM.LB)

Lab Test

Albumin ▼



コード

server (server.R)

```
library(dplyr)
library(ggplot2)
library(plotly)
library(pharmaversesdtm)

server <- function(input, output) {
  output$plot <- renderPlotly({
    data <- lb |>
      left_join(dm, by = c("STUDYID", "USUBJID")) |>
      filter(LBTEST == input$select)

    p <- data |>
      ggplot(aes(x = LBDY, y = LBSTRESN, color = ACTARM)) +
      geom_line(aes(group = USUBJID)) +
      labs(x = "Study Day", y = "Value")

    ggplotly(p)
  })
}
```

user interface (ui.R)

```
library(plotly)
library(pharmaversesdtm)

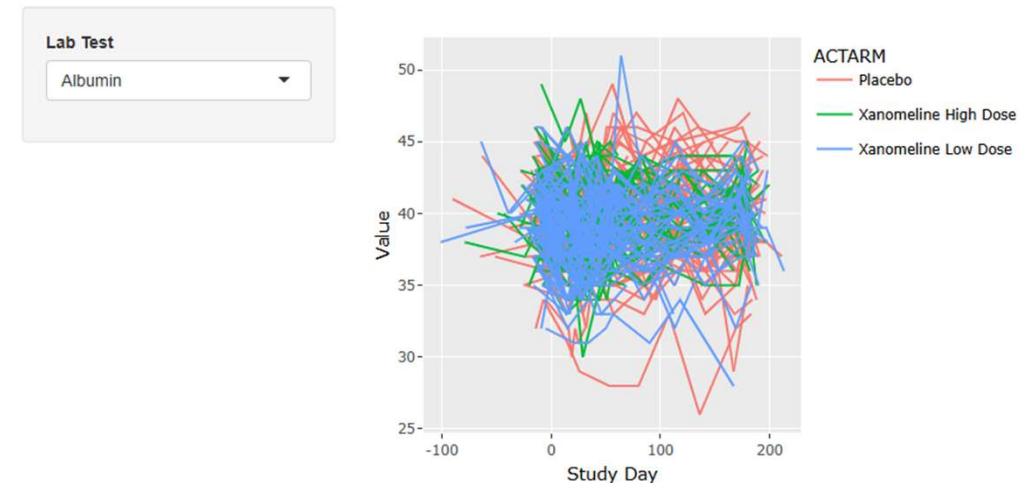
ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

user interface (ui.R)

```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)



user interface (ui.R)

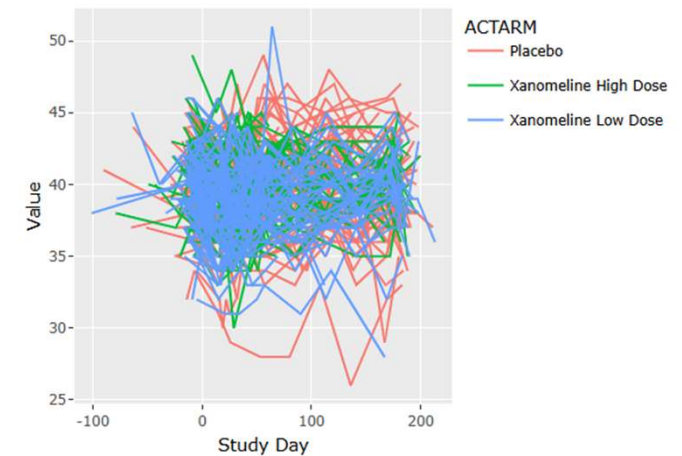
```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)

Lab Test

Albumin

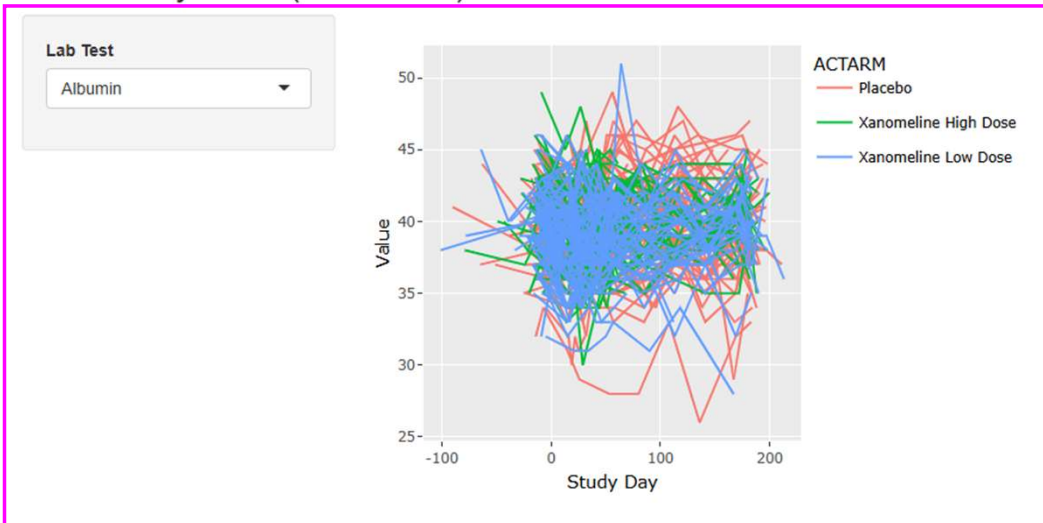


user interface (ui.R)

```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)



user interface (ui.R)

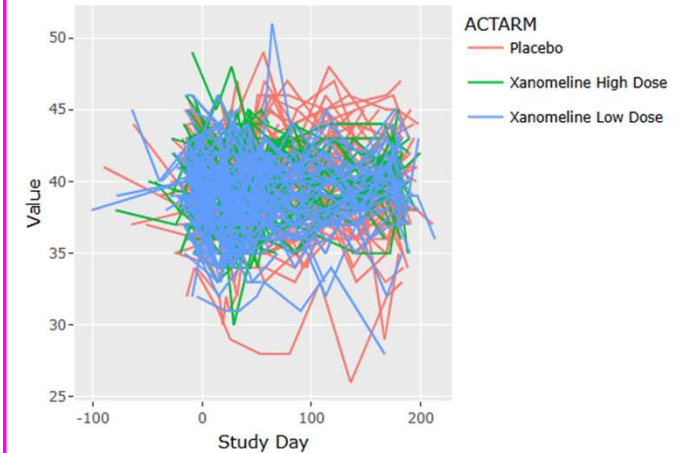
```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)

Lab Test

Albumin

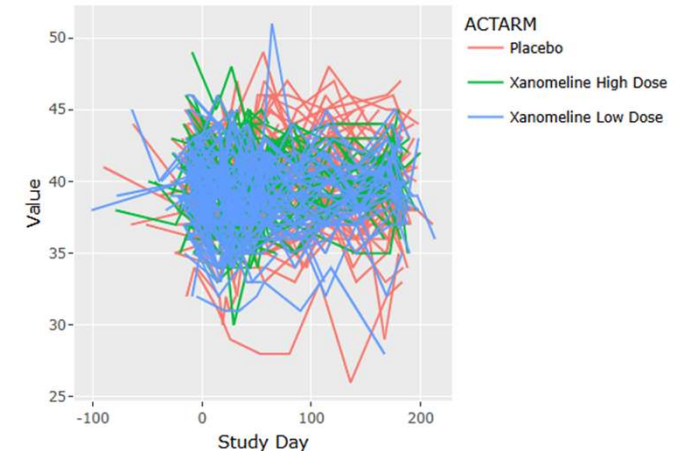
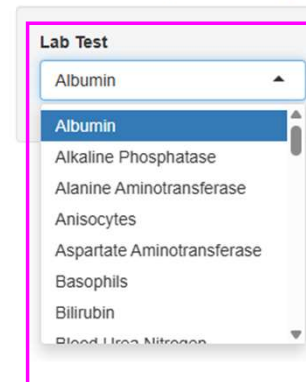


user interface (ui.R)

```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)



user interface (ui.R)

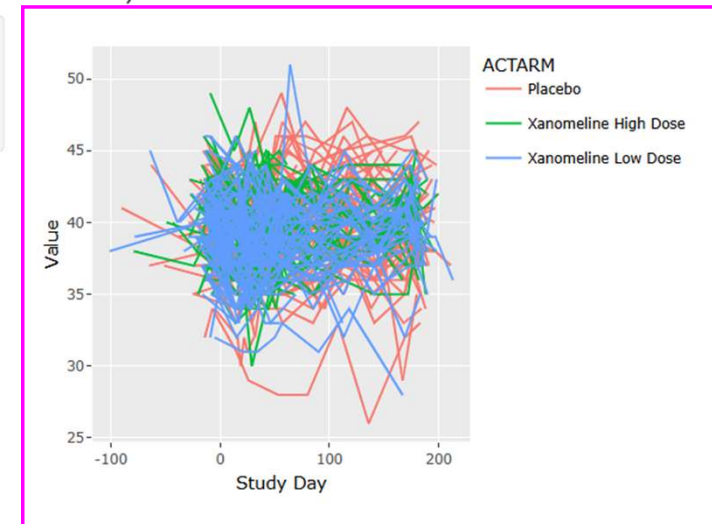
```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

Laboratory Data (SDTM.LB)

Lab Test

Albumin



server (server.R)

```
library(dplyr)
library(ggplot2)
library(plotly)
library(pharmaversesdtm)

server <- function(input, output) {
  output$plot <- renderPlotly({
    data <- lb |>
      left_join(dm, by = c("STUDYID", "USUBJID")) |>
      filter(LBTEST == input$select)

    p <- data |>
      ggplot(aes(x = LBDY, y = LBSTRESN, color = ACTARM)) +
      geom_line(aes(group = USUBJID)) +
      labs(x = "Study Day", y = "Value")

    ggplotly(p)
  })
}
```

server (server.R)

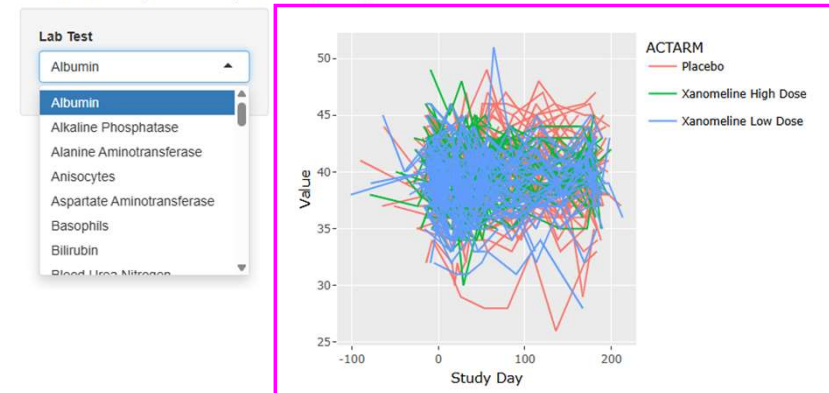
```
library(dplyr)
library(ggplot2)
library(plotly)
library(pharmaversesdtm)

server <- function(input, output) {
  output$plot <- renderPlotly({
    data <- lb |>
      left_join(dm, by = c("STUDYID", "USUBJID")) |>
      filter(LBTEST == input$select)

    p <- data |>
      ggplot(aes(x = LBDY, y = LBSTRESN, color = ACTARM)) +
      geom_line(aes(group = USUBJID)) +
      labs(x = "Study Day", y = "Value")

    ggplotly(p)
  })
}
```

Laboratory Data (SDTM.LB)



server (server.R)

```
library(dplyr)
library(ggplot2)
library(plotly)
library(pharmaversesdtm)

server <- function(input, output) {
  output$plot <- renderPlotly({
    data <- lb |>
      left_join(dm, by = c("STUDYID", "USUBJID")) |>
      filter(LBTEST == input$select)

    p <- data |>
      ggplot(aes(x = LBDY, y = LBSTRESN, color = ACTARM)) +
      geom_line(aes(group = USUBJID)) +
      labs(x = "Study Day", y = "Value")

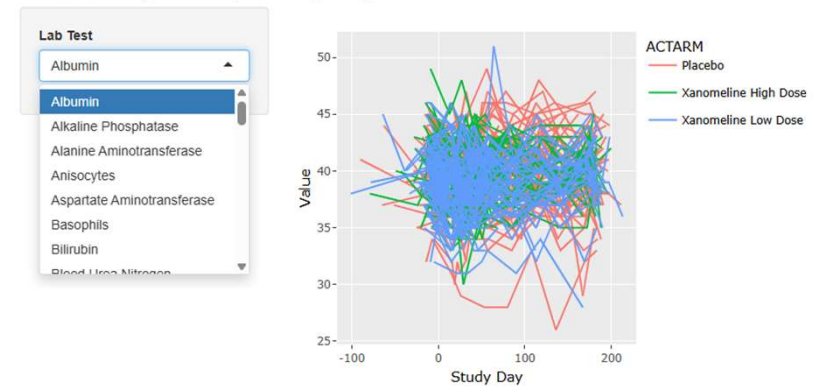
    ggplotly(p)
  })
}
```

```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

ID "plot"

Laboratory Data (SDTM.LB)



server (server.R)

```
library(dplyr)
library(ggplot2)
library(plotly)
library(pharmaversesdtm)

server <- function(input, output) {
  output$plot <- renderPlotly({
    data <- lb |>
      left_join(dm, by = c("STUDYID", "USUBJID")) |>
      filter(LBTEST == input$select)

    p <- data |>
      ggplot(aes(x = LBDY, y = LBSTRESN, color = ACTARM)) +
      geom_line(aes(group = USUBJID)) +
      labs(x = "Study Day", y = "Value")

    ggplotly(p)
  })
}
```

```
library(plotly)
library(pharmaversesdtm)

ui <- fluidPage(
  titlePanel("Laboratory Data (SDTM.LB)"),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "select", label = "Lab Test",
        choices = unique(lb$LBTEST)
      ),
    ),
    mainPanel(
      plotlyOutput("plot")
    )
  )
)
```

ID "select"

Laboratory Data (SDTM.LB)

