



Казанский
Государственный
Медицинский
Университет

Разработка онлайн-приложений в R

Лакман Ирина Александровна

Казань, 2025

План лекции

-  Возможно ли создать онлайн-приложения с использованием среды R?
-  Структура Shiny –скриптов
-  Способы запуска Shiny-приложения
-  Макеты страницы и средства пользовательского интерфейса
-  Добавление виджета в приложение
-  Принципы реактивного программирования в R

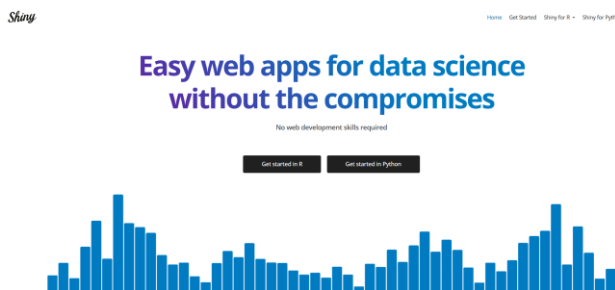


Возможно ли в среде R создавать онлайн-приложения?



- Пакет shiny средство создания интерактивных веб-приложений на языке R.
- В приложении shiny пользовательский веб-интерфейс может быть полностью написан на языке R, но также можно использовать HTML или JavaScript.
- У проекта shiny есть свой сайт <http://shiny.rstudio.com>

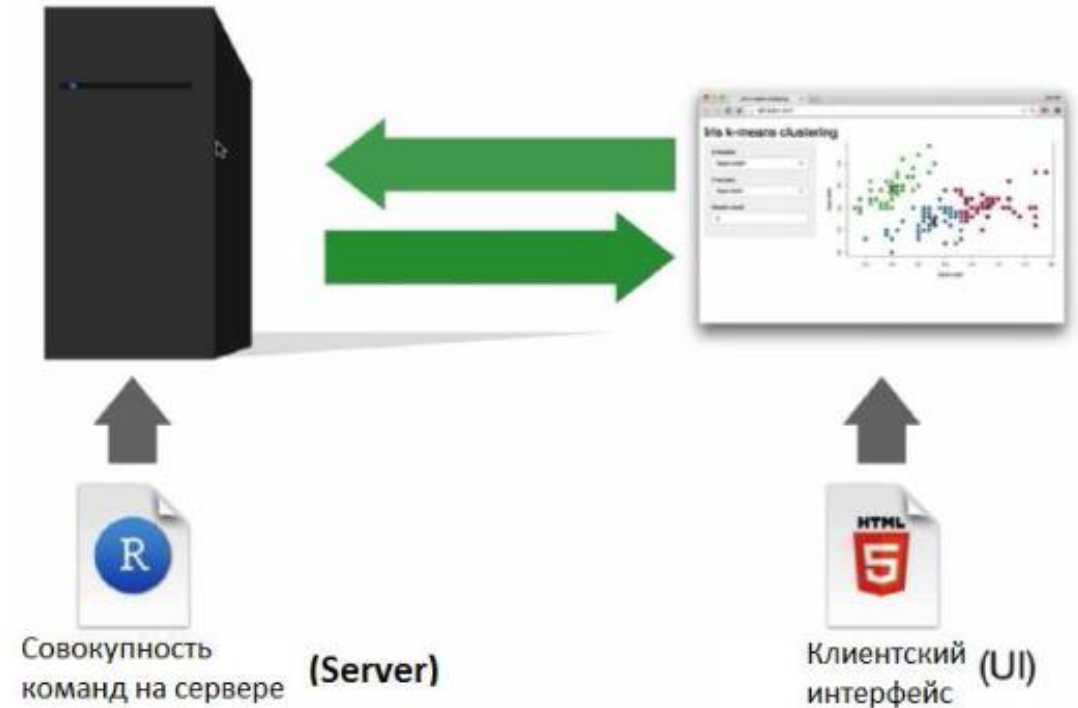
Shiny



при подготовке презентации использованы материалы
В. Шитикова

Структура Shiny –скриптов

Приложение Shiny - это веб-страница (**UI**), связанная с компьютером, на котором выполняется сессия среды R (**Server**)



при подготовке презентации использованы материалы В. Шитикова



Основные библиотеки: shiny и bslib

создание макета пользовательского веб-интерфейса

```
ui <- fluidPage(<определение темы и др.>,  
  titlePanel(<команды вывода текста>), # Панель заголовка  
  sidebarLayout( # Две панели: боковая и главная  
    sidebarPanel(<команды управляющих виджетов>),  
    mainPanel(<определение идентификаторов отображаемых компонент>)))
```

```
install.packages("shiny")  
install.packages("bslib")  
library(shiny)  
library(bslib)
```

Серверная часть

```
server <- function(input, output) {# Обработка сигналов клиентской части и модификация  
серверных объектов
```

```
  reactive(<команды обработки реакций> )
```

Создание или перерисовка графических объектов

```
  output$<идентификаторы> <- renderPlot(<отображение графических объектов>)
```

Отображение или перерисовка текста

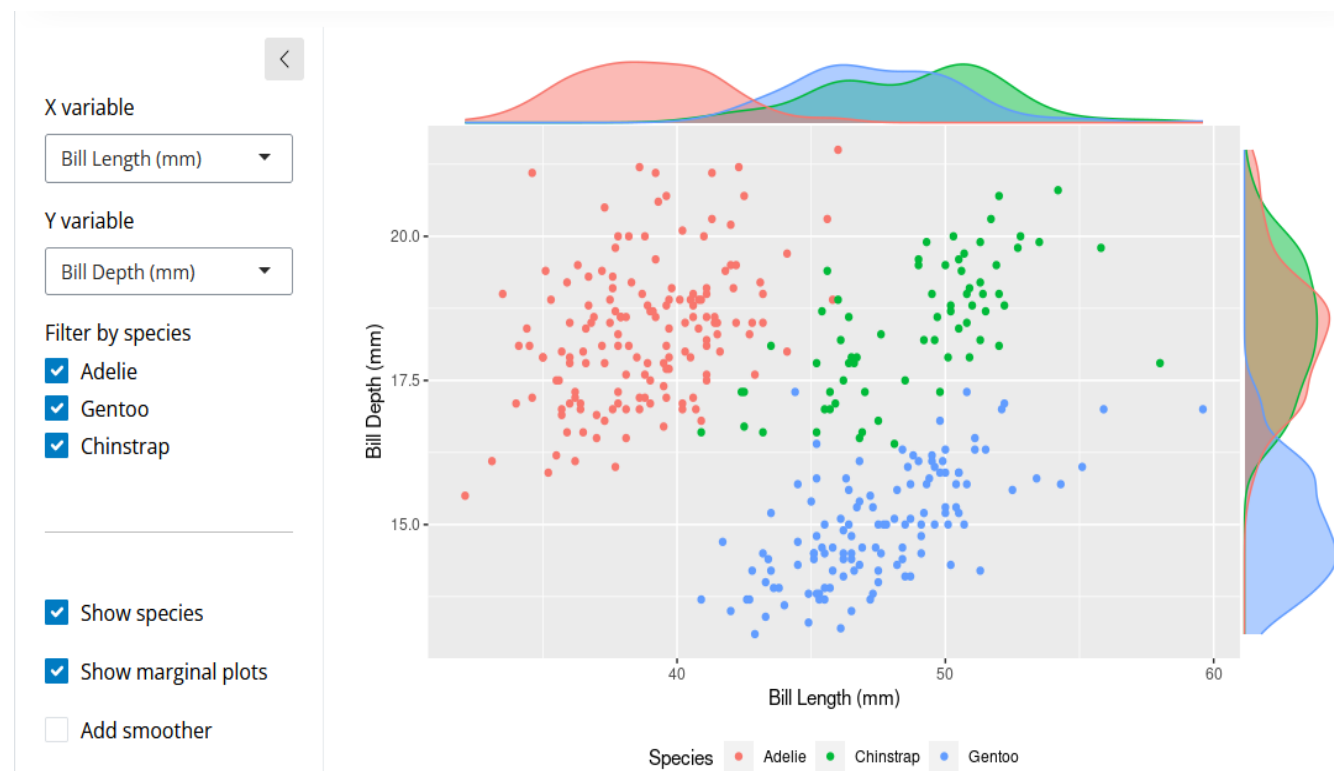
```
  output$<идентификаторы> <- renderText( <отображение текстовых полей>
```

при подготовке презентации использованы
материалы В. Шитикова

Способы запуска Shiny-приложения

- Поместить функции `ui` и `server` в отдельные файлы скриптов (например, `ui.R` и `server.R`) и задать ссылки на них в качестве параметров функции **`shinyApp()`**;
- Объединить функции `ui`, `server` и **`shinyApp()`** в один файл скрипта под названием `app.R`, поместить его в отдельный каталог и задать ссылку на него в качестве параметра функции запуска `runApp()`;
- Запустить `app.R` из командной строки R
`shinyApp(ui = ui, server = server)`

при подготовке презентации
использованы материалы В.
Шитикова



Макеты страницы и средства пользовательского интерфейса

Команды **fluidPage()** или **fluidRow()**, предназначены для создания «растягивающихся» макетов веб-страниц.

Параметры синтаксиса команды **fluidPage()**

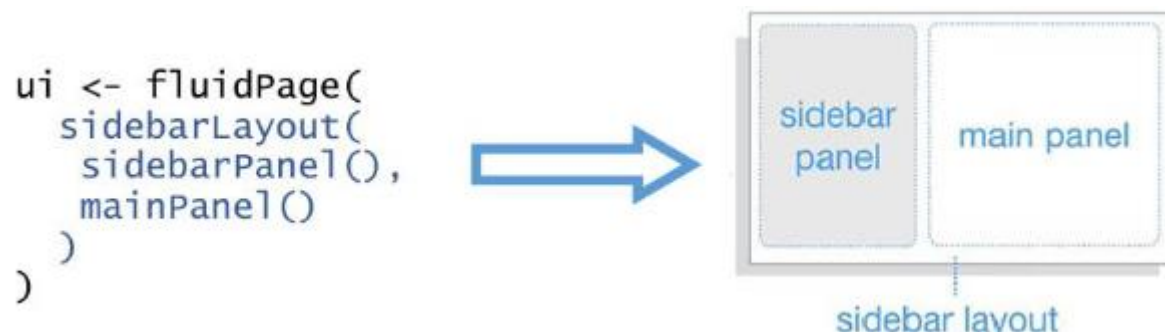
- Простая компоновка с боковой панелью для ввода клиентских команд и большой основной областью для вывода результатов на основе функции **sidebarLayout()**, которая используется по умолчанию в большинстве случаев;
- Сегментированные макеты, созданные с помощью функций **tabsetPanel()** и **navlistPanel()**;
- Приложения с несколькими компонентами верхнего уровня на основе функции **navbarPage()**.
- Функция **titlePanel()**, определяющая заголовок веб-страницы

С использованием функции **fluidRow()** создаются пользовательские макеты приложения, использующие сеточную (grid) структуру Shiny.

Макеты страницы и средства пользовательского интерфейса

Функция **sidebarLayout()**, формирующая макет с боковой панелью, всегда содержит два аргумента:

- **sidebarPanel()**, определяющую компоненты боковой панели (виджеты исходных данных и параметров пользовательского интерфейса);
- функцию **mainPanel()**, определяющую состав графических и текстовых результатов обработки, выводимых на главную панель.



Боковая панель будет размещена по умолчанию в левой стороне веб-страницы. Ее можно перенести в правую сторону путем задания опционального аргумента `position = "right"`.

Виджет (widget) – интерактивный веб-элемент, позволяющий пользователю конструировать приложение.

Варианты создаваемых виджетов

Варианты создаваемых виджетов на сотроне пользователя и соответствующие им команды в R

при подготовке презентации использованы материалы В. Шитикова

Функция	Вид виджета на странице	Выполняемое действие
actionButton	Action	Активная кнопка
numericInput	1	Поле ввода чисел
textInput	Enter text...	Поле ввода текста
dateInput	2014-01-01	Выбор даты по календарю
dateRangeInput	2017-06-21 to 2017-06-21	Выбор диапазона времени по календарю
checkboxInput	☒ Choice A	Простой флажок
checkboxGroupInput	☒ Choice 1 ☐ Choice 2 ☐ Choice 3	Группа флажков
radioButtons	☒ Choice 1 ☐ Choice 2 ☐ Choice 3	Радио-кнопки
selectInput	Choice 1	Выбор варианта из выпадающего списка
sliderInput	050100 0102030405060708090100	Панель движка
fileInput	Browse... No file selected	Управление загрузкой файлов
submitButton	Submit	Кнопка подтверждения

Макеты страницы и средства пользовательского интерфейса

Чтобы добавить необходимый виджет в приложение, нужно разместите функцию виджета в список аргументов функции **sidebarPanel**:

`sidebarPanel(<виджет_1>, < виджет_2>...< виджет_n>).`

У функции каждого виджета есть несколько аргументов. Первые два аргумента для любого виджета являются текстовыми строками и определяют:

- *<Имя виджета>* , которое пользователь не увидит на экране, но его можно использовать для доступа к значению виджета;
- *<Метка>* , которая будет появляться в вашем приложении вместе с виджетом (может быть “пустой”).

```
sidebarPanel( actionButton("action", label = "Action"),
  numericInput("num", label = "Numeric input", value = 1),
  textInput("text", label = "Text input", value = «Введите текст."),
  dateInput("date", label = "Date input", value = "2024-01-01"),
  checkboxInput("checkbox", label = "Choice A", value = TRUE)),
checkboxGroupInput("checkGroup", label = "Checkbox group",
choices = list("Choice 1" = 1, "Choice 2" = 2, "Choice 3" = 3), selected = 1),
radioButtons("radio", label = "Radio buttons", choices = list("Choice 1" = 1, "Choice 2" = 2, "Choice 3" = 3),selected = 1)),
selectInput("select", label = "Select box", choices = list("Choice 1" = 1, "Choice 2" = 2, "Choice 3" = 3), selected = 1),
sliderInput("slider1", label = "Sliders", min = 0, max = 100, value = 50),
fileInput("file", label = "File input"))
```

ПРИМЕР

Добавление виджета в приложение

Также в пользовательском интерфейсе Shiny можно задать непосредственно объекты результатов обработки данных в R (таблицы, графики, гистограммы и т.п.). Они размещаются на главной панели `mainPanel()` в виде функции вывода `*Output`.

Для каждого объекта вывода предназначена своя функции.

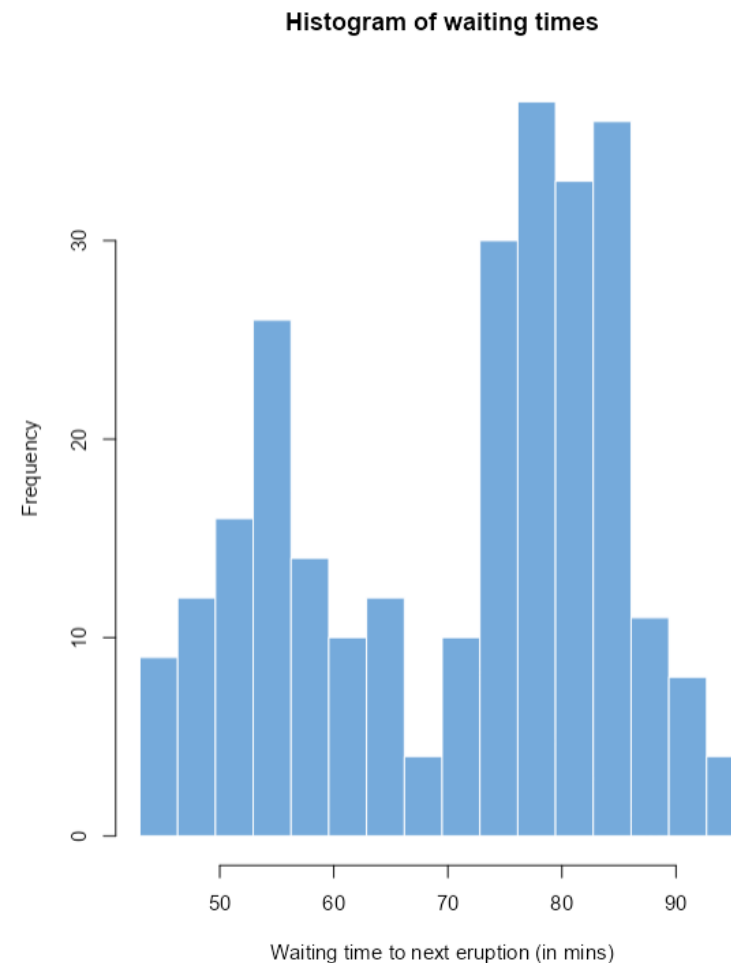
Пример вывода графика (гистограммы):

```
mainPanel(plotOutput(outputId = "distPlot"))
```

ПРИМЕР

Каждая из функций вывода требует только одного аргумента `outputId`, т.е. текстовой строки, которую Shiny воспринимает как идентификатор объекта, который создается функциями серверной части приложения.

при подготовке презентации использованы материалы В. Шитиков



Проектирование серверной части

Для поддержки реакции браузера на изменения пользователем значений виджетов формируется серверная часть приложения.

Для этого в R используется функция `server`, которая принимает на вход объект типа список с именем `input` и создает на выходе объект типа список с именем `output`, который содержит весь код, необходимый для обновления объектов R на формируемой веб-странице. Каждый виджет и каждый объект с идентификатором `outputId` пользовательского интерфейса `ui` должны иметь свою собственную запись в этих списках.

Создание и обновление элементов списка `output` происходит под управлением блоков R-кода, каждый из которых "обернут" функцией `render*()`.

У функции `render` только один аргумент – простое выражение R или несколько строк кода, окруженное фигурными скобками `{}`.

Внимание! Не нужно организовывать возврат результата в последней строке кода: R будет автоматически обновлять возвращаемые значения.

Внимание! Каждой созданной функции вывода `*Output` в блоке `ui` должна соответствовать одна запись функции `render*` соответствующего типа

Проектирование серверной части

Функция вывода ui	render*-функция	Выводимая информация
textOutput	renderText	Текст
imageOutput	renderImage	Изображение
plotOutput	renderPlot	График
tableOutput	renderTable	Таблица (матрица)
uiOutput	renderUI	Виджет, создаваемый на сервере

```
server <- function(input, output) {  
  output$distPlot <- renderPlot({  
    x <- faithful$waiting  
    bins <- seq(min(x), max(x), length.out = input$bins + 1)  
    hist(x, breaks = bins, col = "#75AADB", border = "white",  
    xlab = "Waiting time to next eruption (in mins)",  
    main = "Histogram of waiting times"))})
```

ПРИМЕР

Здесь предполагается, что объект `input` содержит ссылку на виджет с идентификатором `bins`, который позволяет пользователю задать число диапазонов для разбиения на классы. При изменении клиентом этого значения гистограмма будет автоматически перерисовываться

Принципы реактивного программирования в R

Реактивное программирование — парадигма программирования, ориентированная на потоки данных и распространение изменений.

Веб-парадигма Shiny: легко "отслеживать" изменения значений входных параметров на веб-странице, после чего обновленные результаты обработки данных тут же переотображаются браузером.

Поскольку такие веб-приложения являются интерактивными, входные значения могут изменяться в любое время, а выходные значения по всей цепочке выражений, прямо или косвенно зависящие от них, должны быть немедленно обновлены, чтобы отразить эти изменения.

Средства для структурирования логики разрабатываемого приложения;

`render*()`, `reactive()`, `reactiveValues()`, `observeEvent()`, `eventReactive()`.

Сайт разработчиков <https://shiny.rstudio.com>

Распространение Shiny-приложений

Способы распространения Shiny-приложения:

- в форме для локального запуска¹;
- через интернет².

¹ необходимо, чтобы на клиентских компьютерах были установлены R и Shiny,

² нужен только веб-браузер.

Один из способов распространения приложений для локального использования – выложить свой код, состоящий из двух файлов `server.R` и `ui.R`, на `gist.github.com`^{*},

Пусть файлы размещены на <https://gist.github.com/jcheng5/3239667>.

Тогда любой пользователь, у которого установлен R и пакет `shiny`, может запустить сохраненное там приложение просто подав следующую команду:

```
shiny::runGist('3239667')
```

^{*} бесплатный сервис поддержки программных кодов от GitHub



**Спасибо за
внимание!**