

# 我們與 R 的距離



Day 7 : Shiny Application

# Hello !

I am Kristen Chan

Data Scientist

E-Commerce / Telecom

R-Ladies Taipei

Co-Organizer



# Agenda

- Introduction to Shiny
- Let's Play Shiny
  - 資料 : (Kaggle) House Prices: Advanced Regression Techniques  
(Source: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/>)
  - Shiny Application
    - 簡易版
    - 進階版



# Introduction to Shiny

# Introduction to Shiny

- ✓ Shiny is a package from Rstudio
- ✓ It's a web development framework in R.
- ✓ It can be easy to build interactive web applications with R.

You don't need to know following tools:

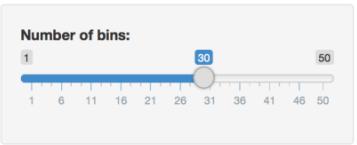
- ✗ Html
- ✗ Javascript
- ✗ CSS

# Hello Shiny

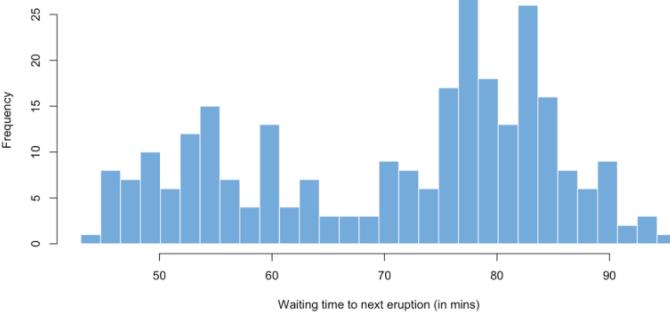


```
library(shiny)
runExample("01_hello")
```

Hello Shiny!



Histogram of waiting times



Hello Shiny!

by RStudio, Inc.

This small Shiny application demonstrates Shiny's automatic UI updates.

Move the *Number of bins* slider and notice how the `renderPlot` expression is automatically re-evaluated when its dependant, `input$bins`, changes, causing a histogram with a new number of bins to be rendered.

app.R

[show with app](#)

```
library(shiny)
```

```
# Define UI for app that draws a histogram ----
ui <- fluidPage(
```

```
  # App title ----
  titlePanel("Hello Shiny!"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(
```

```
    # Sidebar panel for inputs ----
    sidebarPanel(
```

```
      # Input: Slider for the number of bins ----
      sliderInput(inputId = "bins",
                  label = "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
```

).

# A Shiny Project

## ✓ **Ui**

- User interface
- Determine how your app looks

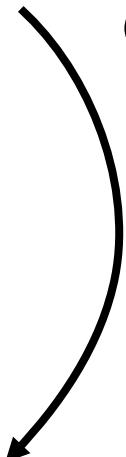
## ✓ **Server**

- Backend or engine of the application
- It's where the data is processed
- Control what your app does

# How does Shiny work



① User 從 Ui 中給一個參數



# How does Shiny work

Ui

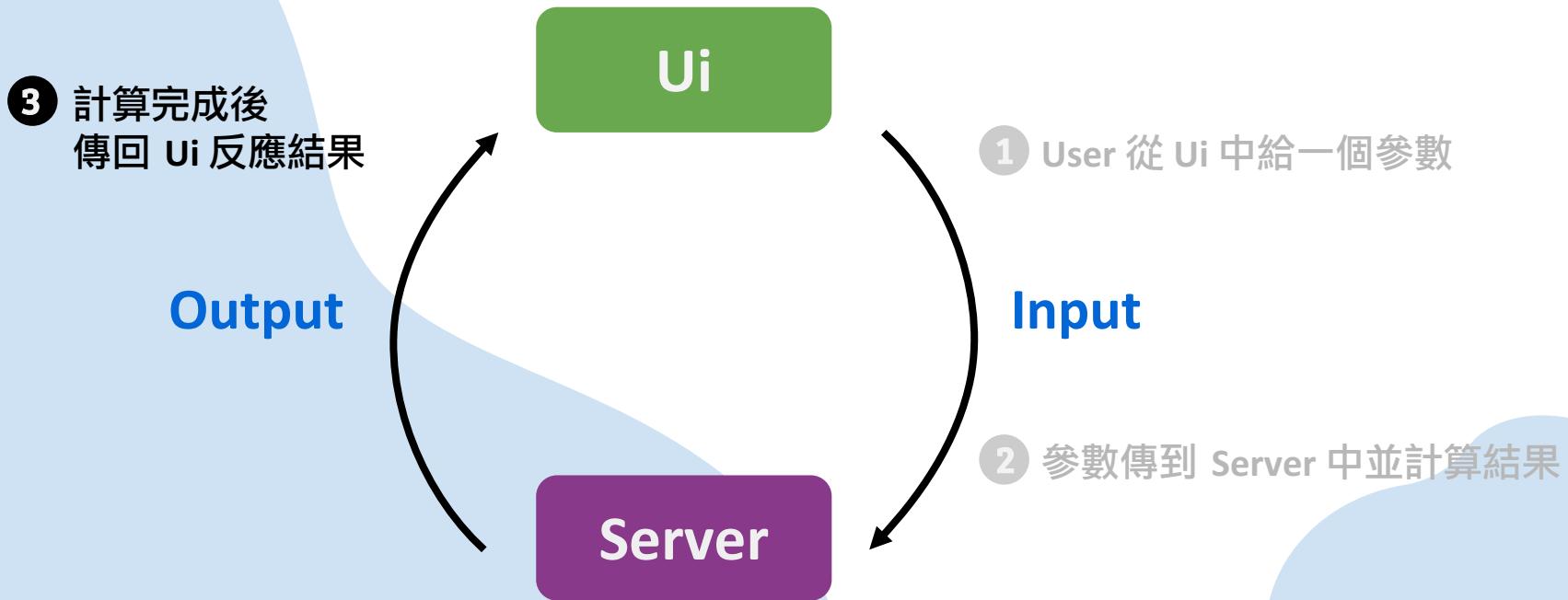
Server

① User 從 Ui 中給一個參數

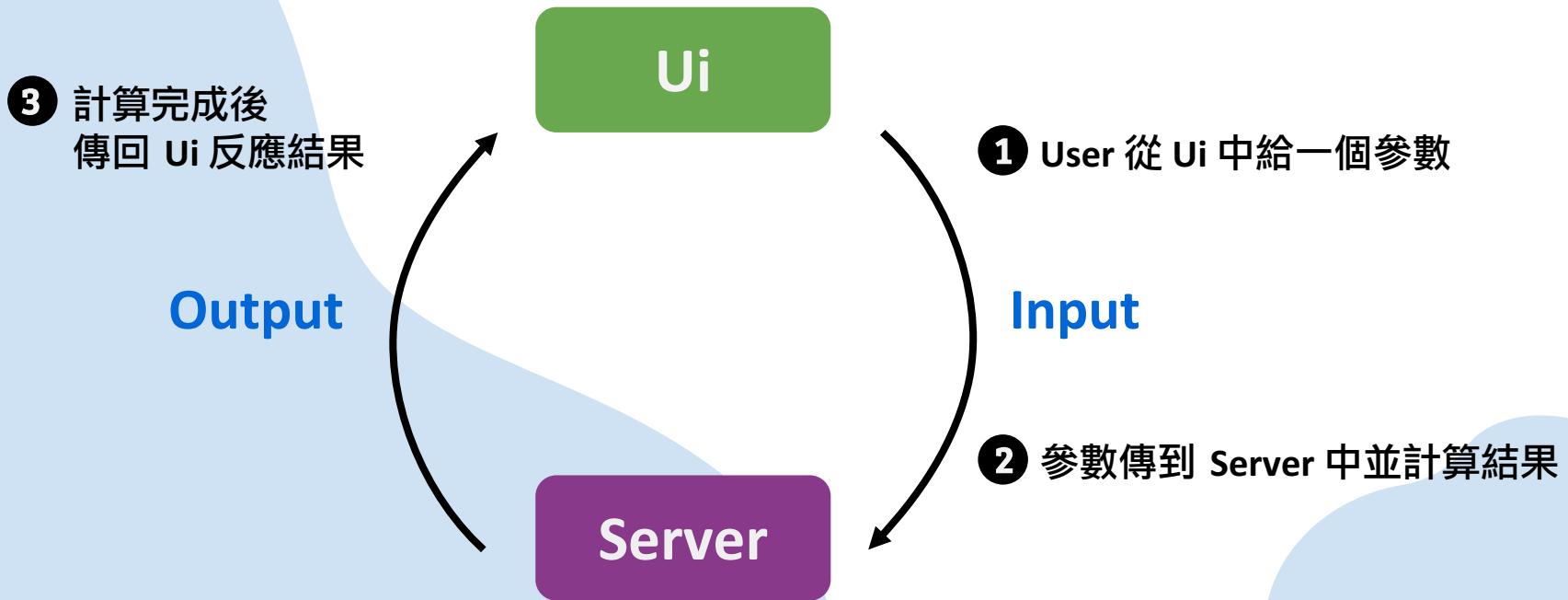
Input

② 參數傳到 Server 中並計算結果

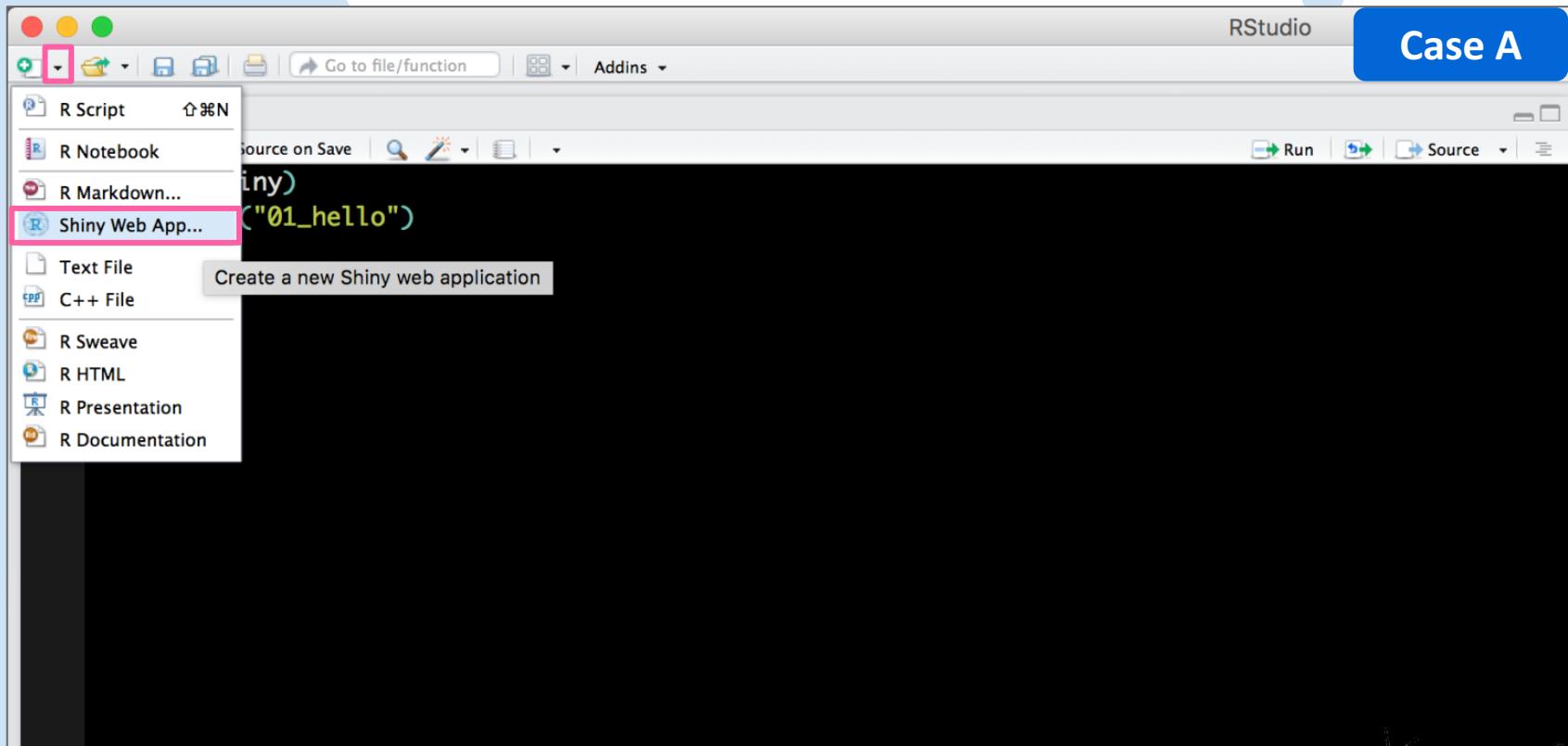
# How does Shiny work



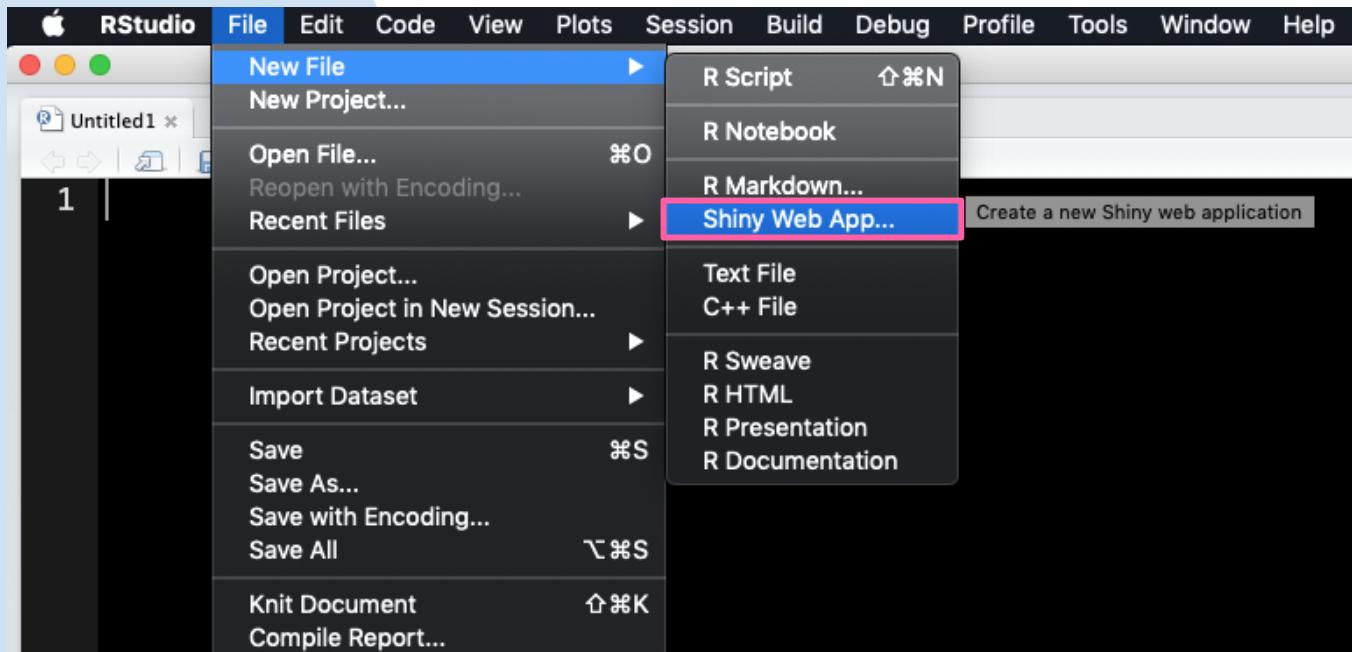
# How does Shiny work



# Your First Shiny App



# Your First Shiny App



Case B

# Your First Shiny App -- Case 1 (Ui.R / Server.R)

New Shiny Web Application

Enter your app's name

Application name: Name

Application type:

Single File (app.R)  
 Multiple File (ui.R/server.R)

Create within directory:

~

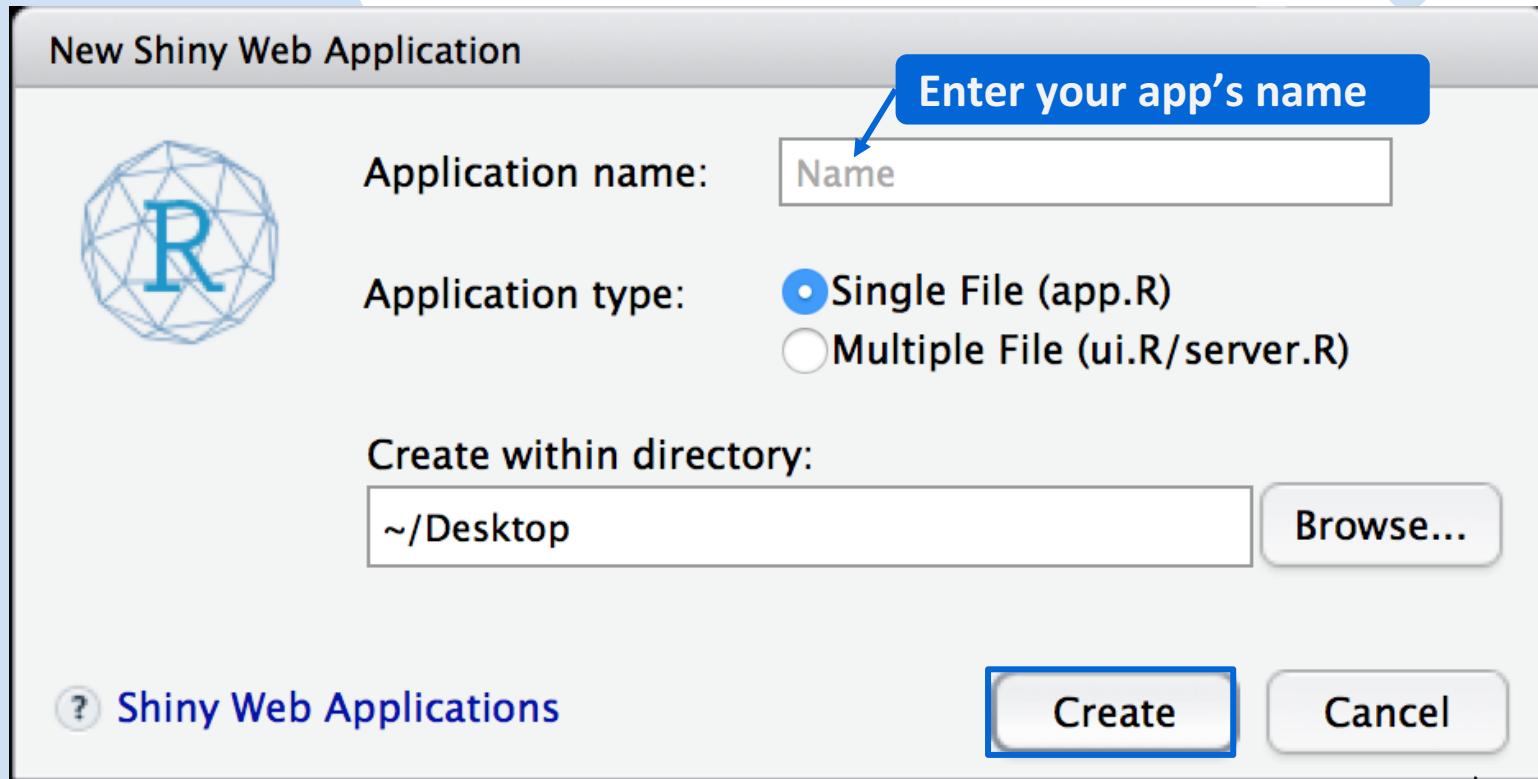
[Shiny Web Applications](#)

# Your First Shiny App -- Case 1 (Ui.R / Server.R)

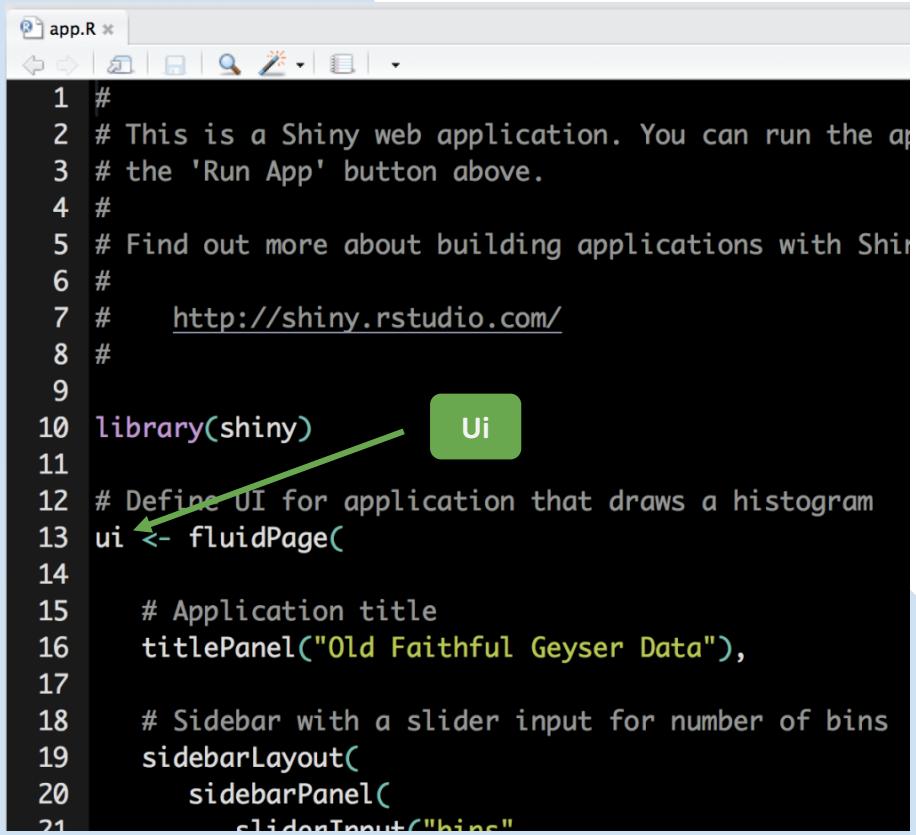
```
ui.R x server.R x
Run App ▾

1 # 
2 # This is the user-interface definition of a Shiny web application. You can
3 # run the application by clicking 'Run App' above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 #     http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 shinyUI(fluidPage(
```

# Your First Shiny App -- Case 2 (app.R)



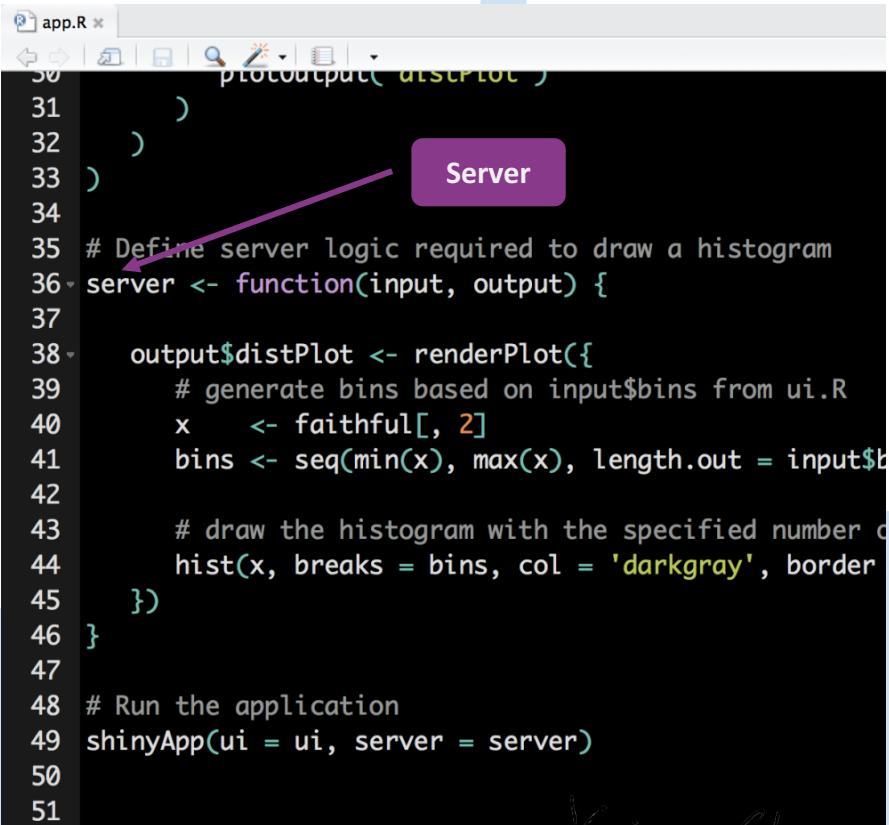
# Your First Shiny App -- Case 2 (app.R)



```

1 #
2 # This is a Shiny web application. You can run the application by
3 # clicking the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny at
6 #
7 #     http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15   # Application title
16   titlePanel("Old Faithful Geyser Data"),
17
18   # Sidebar with a slider input for number of bins
19   sidebarLayout(
20     sidebarPanel(
21       sliderInput("bins",
22
23     )
24   )
25
26   # Main panel for displaying the histogram
27   mainPanel(
28     plotOutput("distPlot")
29   )
30 )
31 )
32 )
33 )
34
35 # Define server logic required to draw a histogram
36 server <- function(input, output) {
37
38   output$distPlot <- renderPlot({
39     # generate bins based on input$bins from ui.R
40     x      <- faithful[, 2]
41     bins <- seq(min(x), max(x), length.out = input$bins)
42
43     # draw the histogram with the specified number of bins
44     hist(x, breaks = bins, col = 'darkgray', border = 'white')
45   })
46 }
47
48 # Run the application
49 shinyApp(ui = ui, server = server)
50
51

```

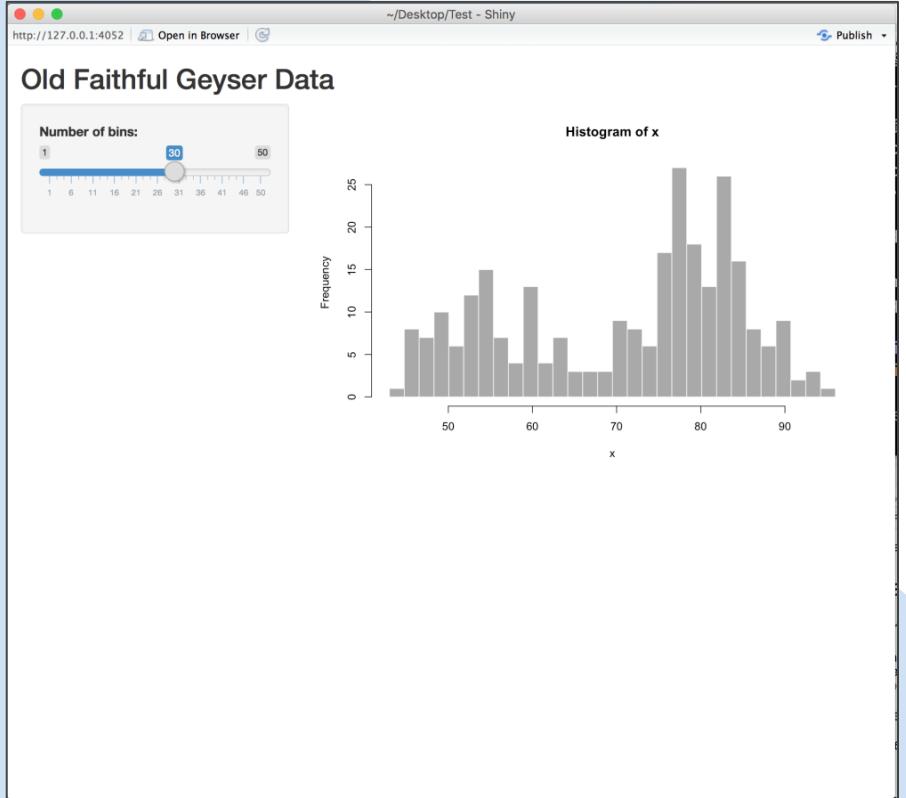


```

50
51

```

# Your First Shiny App



# Let's Play Shiny

# Shiny App

The screenshot shows a Shiny application window titled "我們與 R 的距離". On the left, there is a sidebar with the heading "Decision Tree" containing three input fields: "minsplit" (set to 1), "cp" (set to 0.3), and "maxdepth" (set to 5). The main area displays a table titled "Parameter" with one entry: "Decision Tree ; minsplit : cp : 0.3 maxdepth : 5". To the right of the entry is the "RMSE" value, which is 0.321844683437133. Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom right of the table area, there are "Previous" and "Next" buttons, with the number "1" highlighted between them.

簡易版

# Shiny App

http://127.0.0.1:3525 | Open in Browser |

## 我們與 R 的距離

Model Select

- Linear Regression
- SVR
- Decision Tree
- Random Forest
- XGBoost

nrounds  
1

eta  
0.3

gamma  
0

max\_depth  
3

min\_child\_weight  
1

subsample  
1

colsample\_bytree  
1

Run

~ /R\_DragonBall - Shiny

Show 10 entries Search:

| Parameter   | RMSE              |
|---|-------------------|
| 1 Linear  | 0.160710141107897 |
| 2 SVR ; cost : 1 epsilon : 0.1  | 0.146750160694521 |
| 3 DT ; minsplit : 1 cp : 1 maxdepth : 5   | 0.321844683437133 |
| 4 RF ; ntree : 500 mtr : replace : TRUE   | 0.154809554515217 |
| 5 XGBoost ; 1 eta : 0.3 gamma : 0 max_depth : 3 min_child_weight : 1 subsample : 1 colsample_bytree : 1 | 7.98659755283095  |

Showing 1 to 5 of 5 entries Previous 1 Next

進階版

# House Prices : Advanced Regression Techniques



## House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting  
4,360 teams · Ongoing

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Submit Predictions](#)

[Overview](#)

**Description**

**Start here if...**

You have some experience with R or Python and machine learning basics. This is a perfect competition for data science students who have completed an online course in machine learning and are looking to expand their skill set before trying a featured competition.

**Evaluation**

**Tutorials**

**Frequently Asked Questions**

**Competition Description**



Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

## 預測房屋銷售價格

### ◆ Data

- Train Data : 共 1,460 個房子
- Test Data : 共 1,450 個房子
- ◆ 共 80 個 Feature

# Review -- Day 1 - Day 6

```
##----- 設定環境 -----  
setwd("/Users/rladiestaipei/R_DragonBall/")  
  
library(caTools)  
library(e1071)  
library(rpart)  
library(randomForest)  
library(xgboost)  
library(dplyr)  
library(Metrics)
```

# Review -- Day 1 - Day 6

```
##----- Load Data -----
#讀入資料
dataset <- read.csv("train_new.csv")

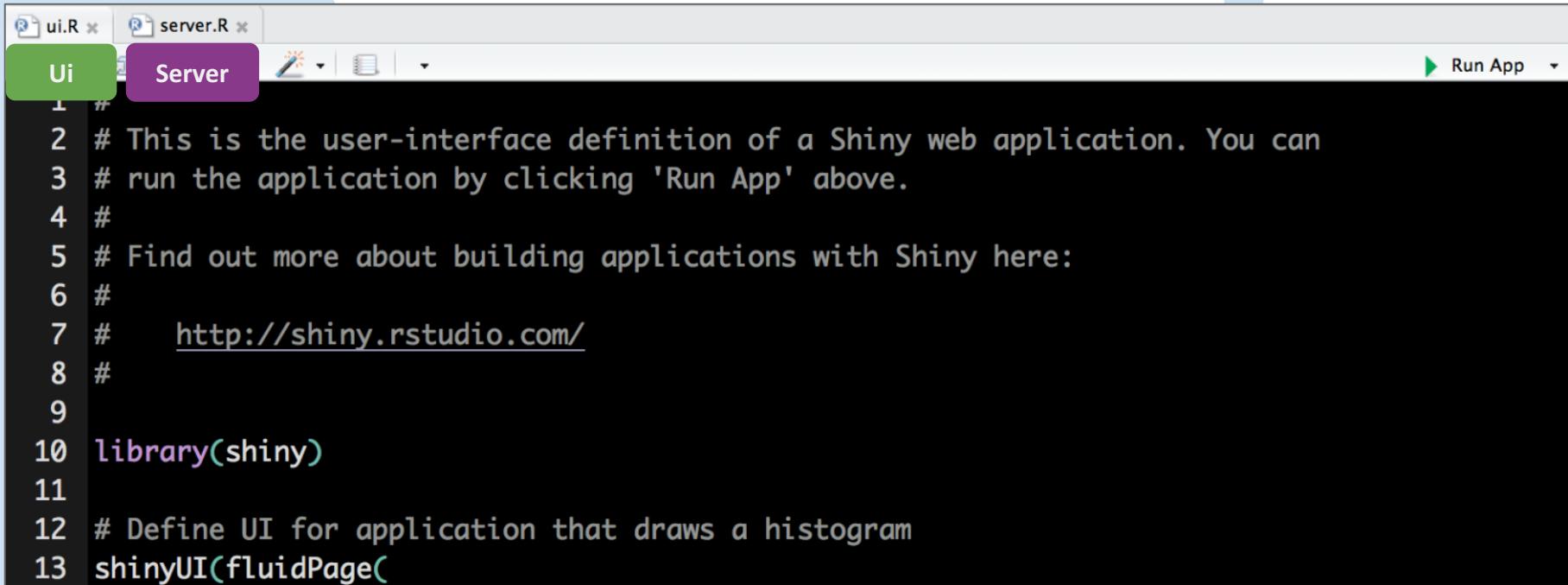
#select features you want to put in models
dataset <- dataset %>% dplyr::select(SalePrice_log, X1stFlrSF, TotalBsmtSF, YearBuilt,
                                         LotArea, Neighborhood, GarageCars, GarageArea, GrLivArea_stand,
                                         MasVnrArea_stand, LotFrontage_log, is_Fireplace, TotalBathrooms,
                                         TotalSF_stand)

#transfer all feature to numeric
cat_index <- which(sapply(dataset, class) == "factor")
dataset[cat_index] <- lapply(dataset[cat_index], as.numeric)

#Splitting the dataset into the Training set and Validation set
set.seed(1)
split <- sample.split(dataset$SalePrice_log, SplitRatio = 0.8)
training_set <- subset(dataset, split == TRUE)
val_set <- subset(dataset, split == FALSE)

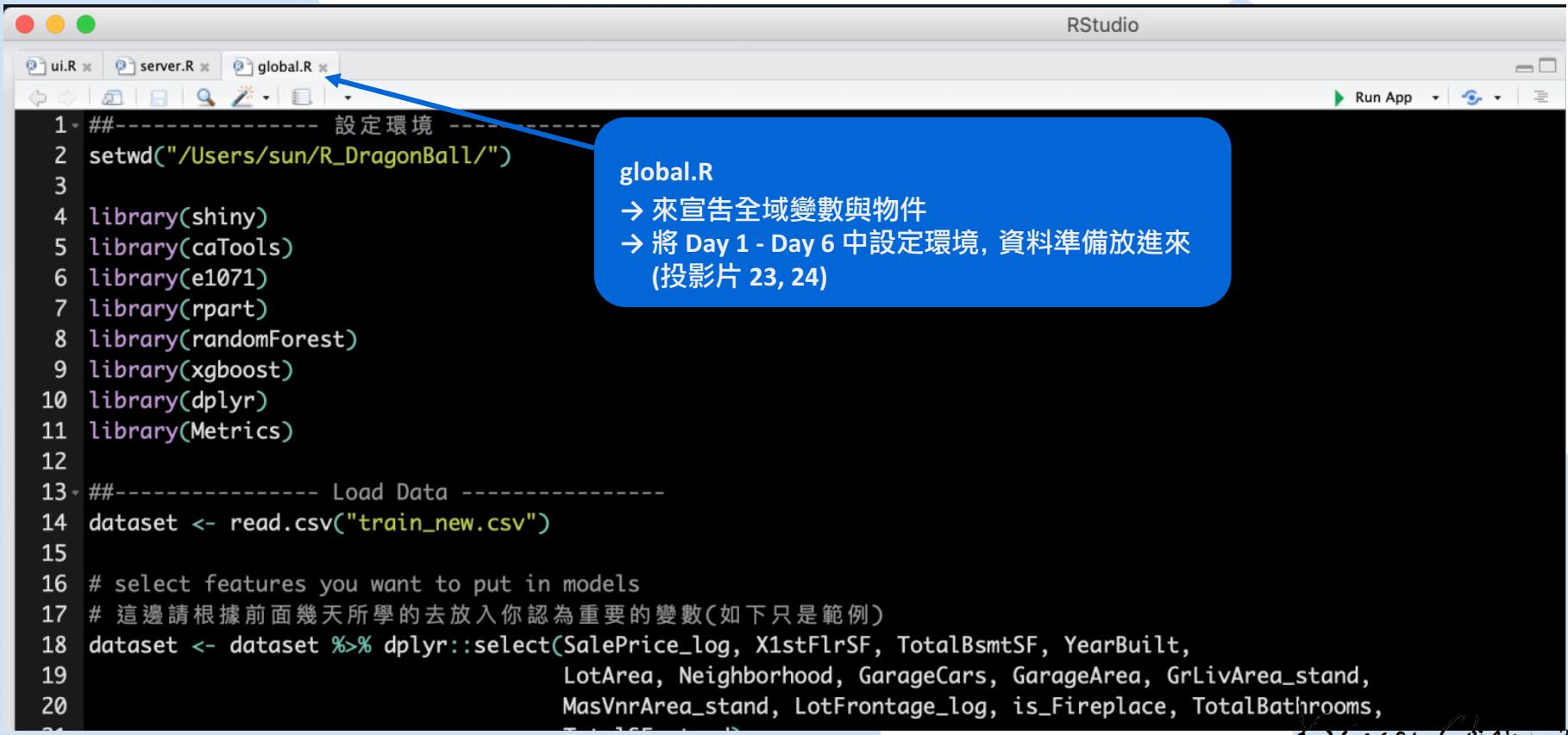
#put testing & training data into two seperates Dmatrix objects
tr_x <- as.matrix(training_set)
tr_y <- training_set$SalePrice_log
val_x <- as.matrix(val_set)
val_y <- val_set$SalePrice_log
```

# Review -- Ui.R / Server.R



```
ui.R * server.R *
Ui Server Run App
1 #
2 # This is the user-interface definition of a Shiny web application. You can
3 # run the application by clicking 'Run App' above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 #     http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 shinyUI(fluidPage(
```

# 新增一個 global.R



RStudio

ui.R x server.R x global.R x

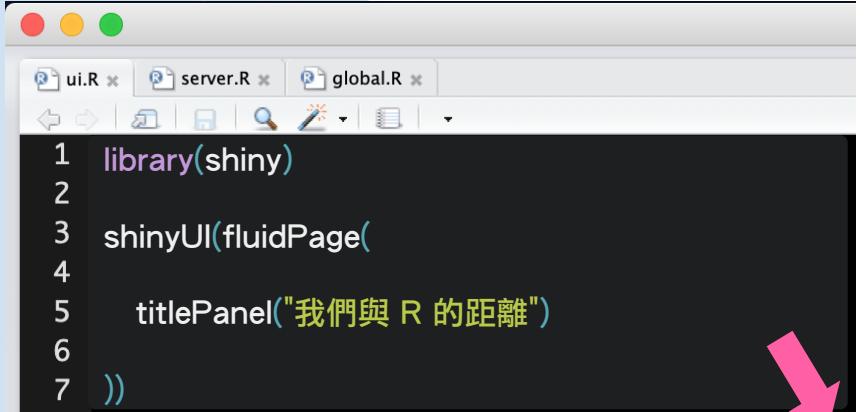
Run App

```
1 ##----- 設定環境 -----#
2 setwd("/Users/sun/R_DragonBall/")
3
4 library(shiny)
5 library(caTools)
6 library(e1071)
7 library(rpart)
8 library(randomForest)
9 library(xgboost)
10 library(dplyr)
11 library(Metrics)
12
13 ##----- Load Data -----
14 dataset <- read.csv("train_new.csv")
15
16 # select features you want to put in models
17 # 這邊請根據前面幾天所學的去放入你認為重要的變數(如下只是範例)
18 dataset <- dataset %>% dplyr::select(SalePrice_log, X1stFlrSF, TotalBsmtSF, YearBuilt,
19                                         LotArea, Neighborhood, GarageCars, GarageArea, GrLivArea_stand,
20                                         MasVnrArea_stand, LotFrontage_log, is_Fireplace, TotalBathrooms,
```

**global.R**

- 來宣告全域變數與物件
- 將 Day 1 - Day 6 中設定環境, 資料準備放進來  
(投影片 23, 24)

# [Code] 我們與 R 的距離 -- Title



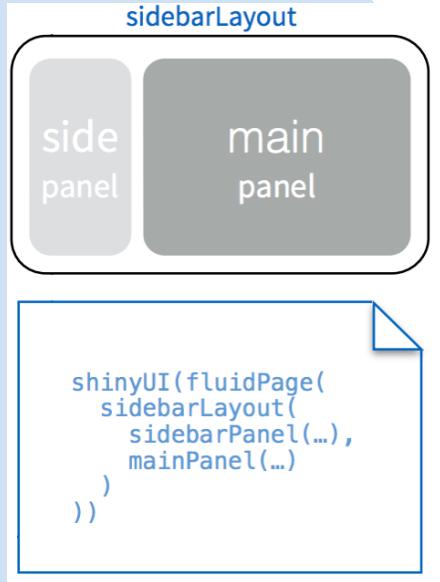
```
1 library(shiny)
2
3 shinyUI(fluidPage(
4
5   titlePanel("我們與 R 的距離")
6
7 ))
```



titlePanel →



# [Layout] 我們與 R 的距離



我們與 R 的距離

Decision Tree

minsplit  
1

cp  
0.3

maxdepth  
5

sidebarLayout

Show 10 entries

Parameter

1 Decision Tree ; minsplit : cp : 0.3 maxdepth : 5

RMSE 0.321844683437133

Search: Previous 1 Next

The screenshot shows a Shiny application window titled "我們與 R 的距離". The UI includes a sidebar with input fields for "Decision Tree" parameters: "minsplit" (1), "cp" (0.3), and "maxdepth" (5). To the right is a table titled "sidebarLayout" showing one entry: "Decision Tree ; minsplit : cp : 0.3 maxdepth : 5" with an RMSE of "0.321844683437133". A pink arrow points from the sidebar in the diagram to the sidebar in the screenshot. Below the screenshot is a code editor window showing the R code for the application:

```
1 library(shiny)  
2  
3 shinyUI(fluidPage(  
4   titlePanel("我們與 R 的距離"),  
5   sidebarLayout(  
6     ))  
7 ))
```

# Access Input Value

Server.R

```

1 library(dplyr)
2 shinyServer( function(input, output) {
3
4   output$outputTable <- DT::renderDataTable([
5     regressor <- rpart(formula = SalePrice_log ~ ., data = training_set,
6                           control = rpart.control(minsplit = input$DT_minsplit,
7                                                 cp = input$DT_cp,
8                                                 maxdepth = input$DT_maxdepth))
9
10  y_pred <- predict(regressor, newdata = val_set)
11
12  data.frame(Parameter = paste0("Decision Tree ; ",
13                            "minsplit : ",input$DT_minsplit,
14                            "maxdepth : ",input$DT_maxdepth),
15    RMSE = rmse(val_set$SalePrice_log, y_pred = ,
16                stringsAsFactors = FALSE)
17  })
18
19 })

```

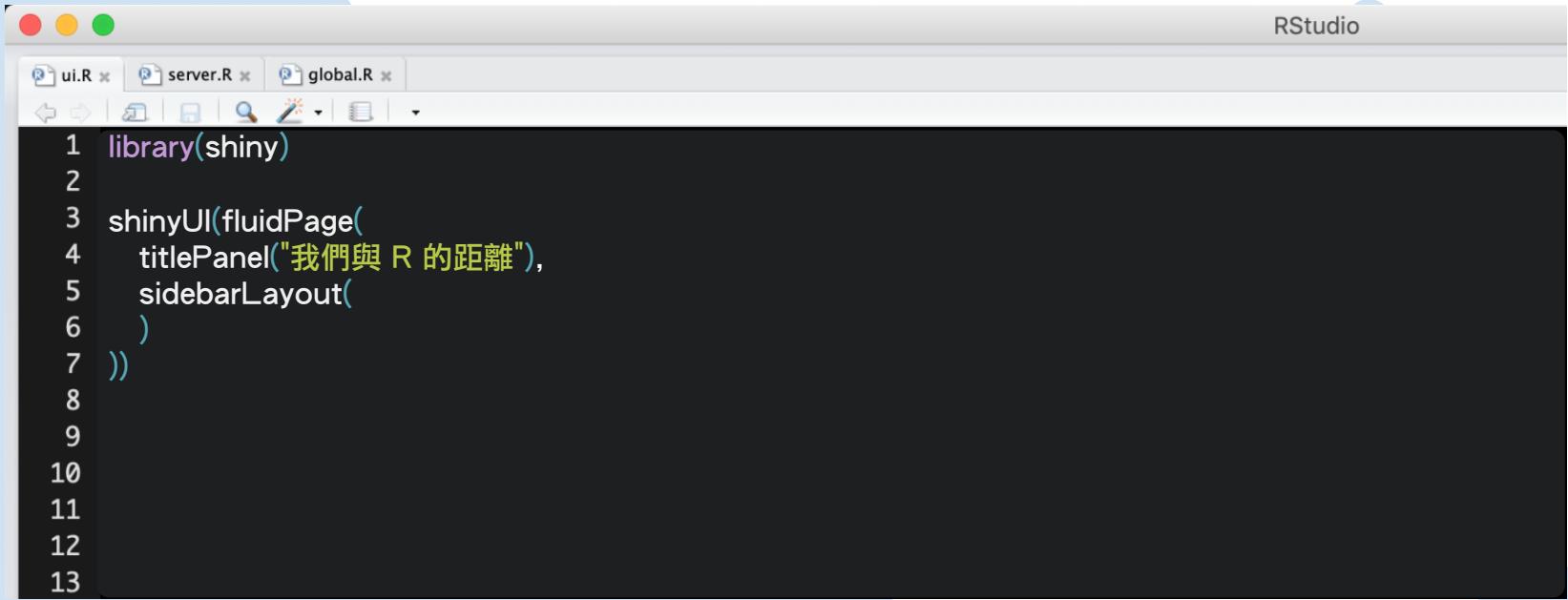
Ui.R

```

3
4
5 sidebarLayout(
6   sidebarPanel(
7
8     h3("Decision Tree"),
9     br(),
10    numericInput("DT_minsplit", "minsplit", value=1,
11    numericInput("DT_cp", "cp", value=0.3, min=0, max=1),
12    numericInput("DT_maxdepth", "maxdepth", value=5)
13  ),
14
15  #Show the RMSE values
16  mainPanel(
17    DT::dataTableOutput("outputTable")
18  )
19
20 )
21
22 )

```

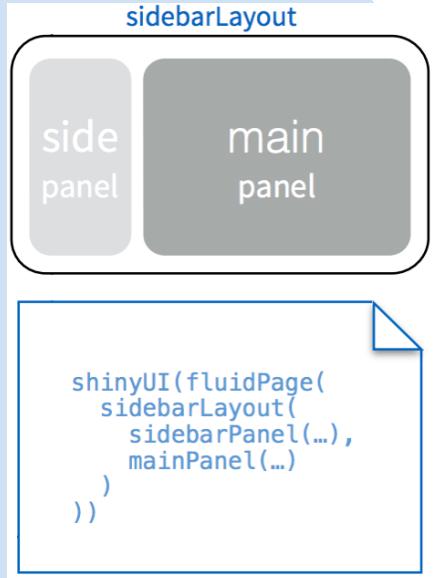
# [Code] 我們與 R 的距離 -- Layout



The screenshot shows the RStudio interface with the title bar "RStudio". The left sidebar lists three files: "ui.R", "server.R", and "global.R". The main workspace is a dark-themed code editor displaying the following R code:

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("我們與 R 的距離"),
5   sidebarLayout(
6     )
7   )))
8
9
10
11
12
13
```

# [Layout] 我們與 R 的距離



我們與 R 的距離

sidebarPanel

mainPanel

Decision Tree

minsplit: 1, cp: 0.3, maxdepth: 5

Parameter

| 1                           | Decision Tree ; minsplit : cp : 0.3 maxdepth : 5 | RMSE : 0.321844683437133 |
|-----------------------------|--|--------------------------|
| Showing 1 to 1 of 1 entries |  |                          |

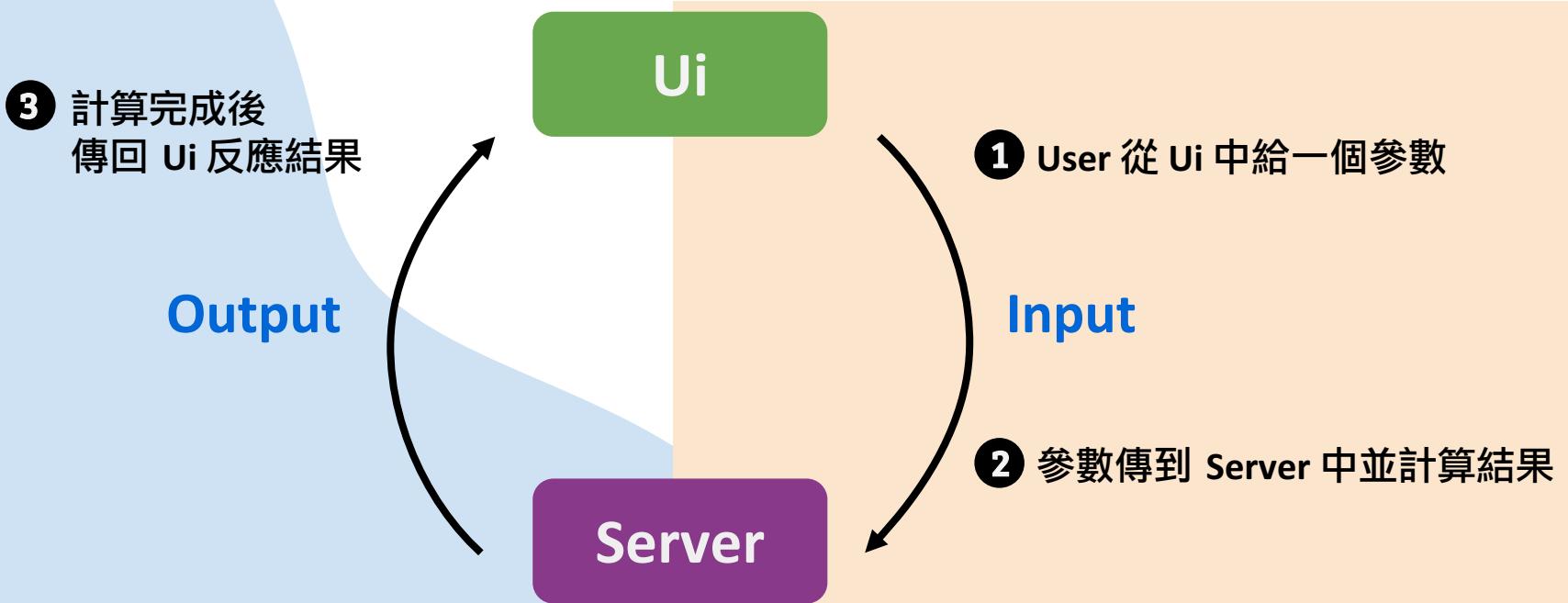
Search: [ ]

Previous [ ] Next [ ]

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("我們與 R 的距離"),
5   sidebarLayout(
6     sidebarPanel(),
7     mainPanel()
8   )
9 ))
10
11
12
13
```

KrisenChan

# Review -- How does Shiny work



# Shiny Input -- sidebarPanel

|   |   |  |   |   |   |
|---|---|--|---|---|---|
| A | Action  | <b>actionButton</b> (inputId, label, icon, ...)  | F | Choose File   | <b>fileInput</b> (inputId, label, multiple, accept)   |
| B | Link  | <b>actionLink</b> (inputId, label, icon, ...)  | G | 1   | <b>numericInput</b> (inputId, label, value, min, max, step)   |
| C | <input checked="" type="checkbox"/> Check me  | <b>checkboxInput</b> (inputId, label, value)   | H | .....   | <b>passwordInput</b> (inputId, label, value)  |
| D | <input checked="" type="checkbox"/> Choice 1<br><input checked="" type="checkbox"/> Choice 2<br><input type="checkbox"/> Choice 3 | <b>checkboxGroupInput</b> (inputId, label, choices, selected, inline)                      | I | <input checked="" type="radio"/> Choice A<br><input type="radio"/> Choice B<br><input type="radio"/> Choice C | <b>radioButtons</b> (inputId, label, choices, selected, inline)   |
| E |    | <b>dateInput</b> (inputId, label, value, min, max, format, startview, weekstart, language) | J |                            | <b>dateRangeInput</b> (inputId, label, start, end, min, max, format, startview, weekstart, language, separator) |

# [Layout] 我們與 R 的距離

The screenshot shows a Shiny application window titled "我們與 R 的距離". On the left, there's a sidebar with a heading "Decision Tree" and three numeric input fields: "minsplit" (value 1), "cp" (value 0.3), and "maxdepth" (value 5). A pink rounded rectangle highlights these three input fields. Below the sidebar, a pink button with the text "numericInput↑" has an upward arrow icon. On the right, there's a table titled "Parameter" with one entry: "1 Decision Tree ; minsplit : cp : 0.3 maxdepth : 5". To the right of the table, the RMSE value is listed as "0.321844683437133". At the bottom of the table, it says "Showing 1 to 1 of 1 entries". Navigation buttons for "Previous" (disabled), "1", and "Next" are also present.

| Parameter  | RMSE              |
|--|-------------------|
| 1 Decision Tree ; minsplit : cp : 0.3 maxdepth : 5 | 0.321844683437133 |

Showing 1 to 1 of 1 entries

Previous 1 Next

# [Code] 我們與 R 的距離 -- Input



The screenshot shows the RStudio interface with the ui.R file open. The code defines a shiny application with a sidebar layout containing three numeric input fields for a decision tree model.

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("我們與 R 的距離"),
5   sidebarLayout(
6     sidebarPanel(
7       h3("Decision Tree"),
8       br(),
9       numericInput("DT_minsplit", "minsplit", value=1, min=0, max=100, step=1) ,
10      numericInput("DT_cp", "cp", value=0.3, min=0, max=1, step=0.01) ,
11      numericInput("DT_maxdepth", "maxdepth", value=5, min=1, max=30, step=1)
12    ),
13    mainPanel(
14    )
15  )
16 ))
17
18
19
20
```

# [Input] 我們與 R 的距離

```
numericInput("DT_minsplit", "minsplit",
            value=1, --> default
            min=0, max=100,
            step=1 )
```

inputId

label

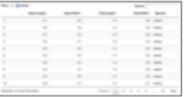
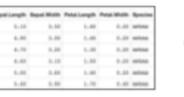
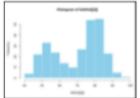
minsplit

cp

maxdepth

|   |     |   |
|---|-----|---|
| 1 | 0.3 | 5 |
|---|-----|---|

# Shiny Output -- mainPanel

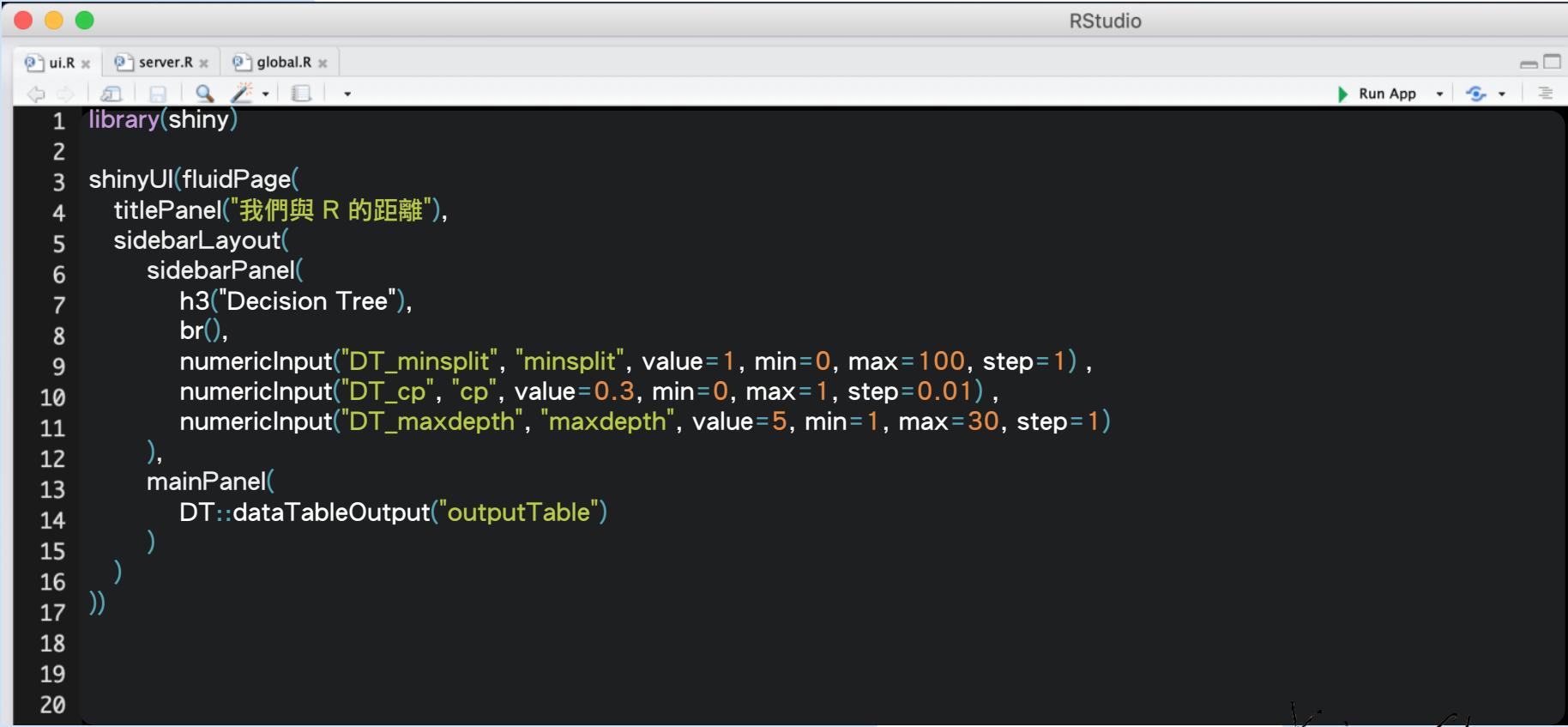
|   |   |   |   |   |  |
|---|---|---|---|---|--|
| a |  | <code>dataTableOutput(outputId, icon, ...)</code>   | e |  | <code>tableOutput(outputId)</code>   |
| b |  | <code>plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)</code> | f |  | <code>imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)</code> |
| c |   |   | g | foo   | <code>textOutput(outputId, container, inline)</code>   |
| d |  | <code>verbatimTextOutput(outputId)</code>   | h |  | <code>leafletOutput(outputId, ...)</code>  |

`dataTableOutput( "plot_name" )`

the type of output  
to display

name to give to the  
output object

# [Code] 我們與 R 的距離



The screenshot shows the RStudio interface with the following code in the ui.R file:

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("我們與 R 的距離"),
5   sidebarLayout(
6     sidebarPanel(
7       h3("Decision Tree"),
8       br(),
9       numericInput("DT_minsplit", "minsplit", value=1, min=0, max=100, step=1),
10      numericInput("DT_cp", "cp", value=0.3, min=0, max=1, step=0.01),
11      numericInput("DT_maxdepth", "maxdepth", value=5, min=1, max=30, step=1)
12    ),
13    mainPanel(
14      DT::dataTableOutput("outputTable")
15    )
16  )
17))
18
19
20
```

The code defines a shiny application with a sidebar containing three numeric input fields for a Decision Tree model: minsplit, cp, and maxdepth. The main panel contains a data table output named "outputTable". The title panel displays the text "我們與 R 的距離".

# [Code] 我們與 R 的距離 -- dataTableOutput

```
DT::dataTableOutput("outputTable")
```



outputId

Show 10 entries

Search:

Parameter

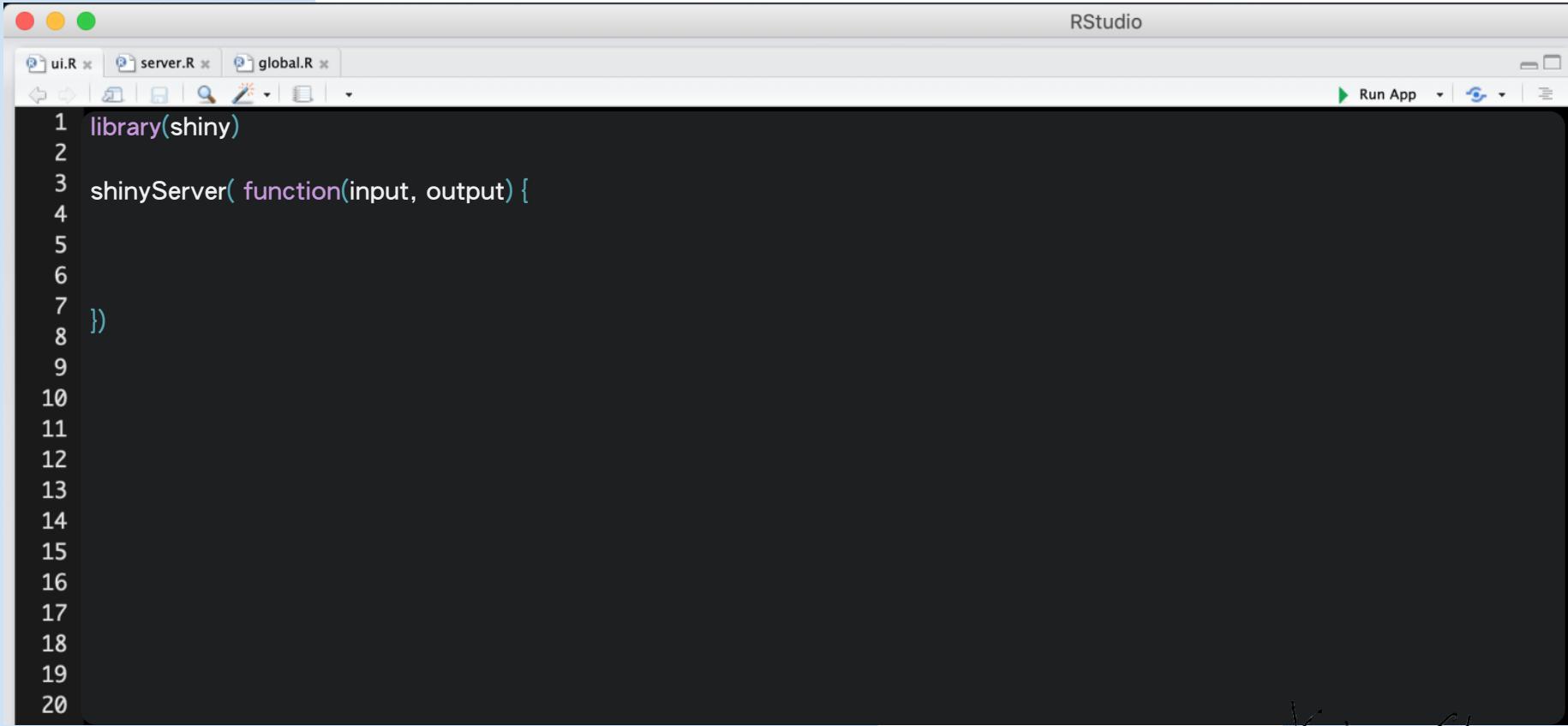
RMSE

|   |  |                   |
|---|--|-------------------|
| 1 | Decision Tree ; minsplit : cp : 0.3 maxdepth : 5 | 0.321844683437133 |
|---|--|-------------------|

Showing 1 to 1 of 1 entries

Previous 1 Next

# [Code] 我們與 R 的距離 -- Original Server.R

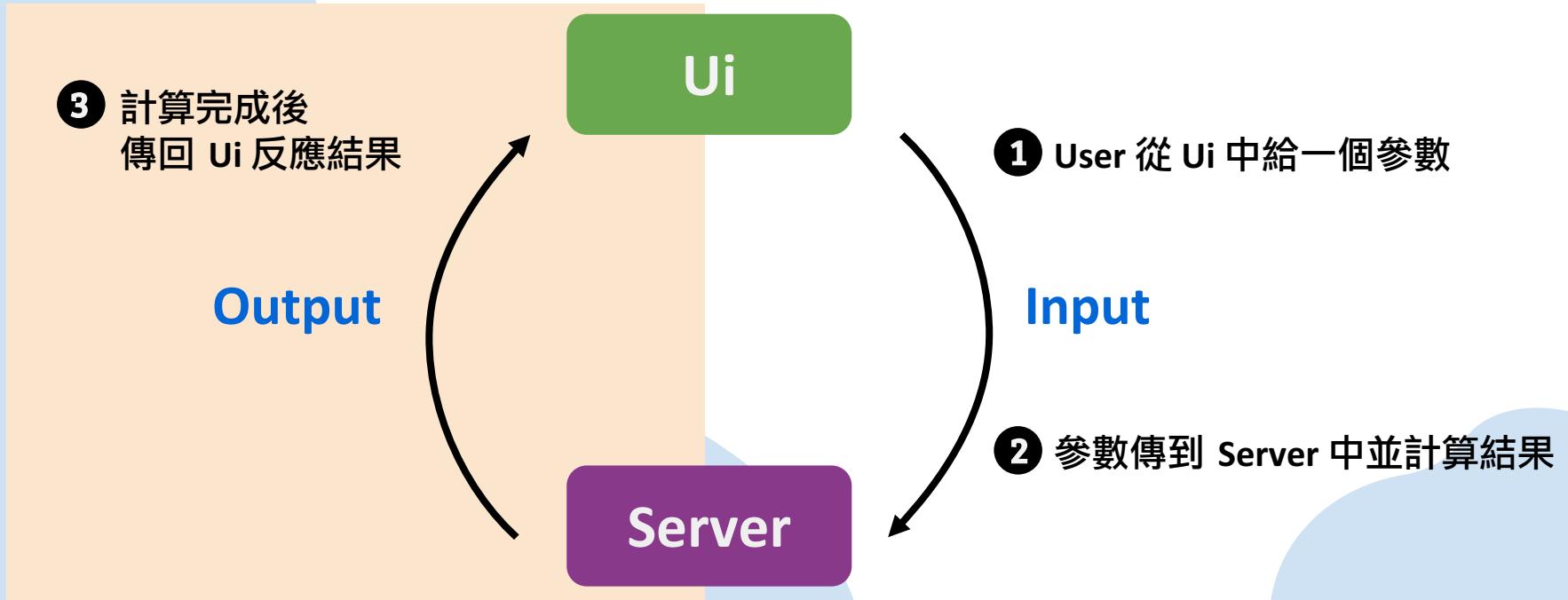


The screenshot shows the RStudio interface with the title bar "RStudio". The left sidebar lists three files: "ui.R", "server.R", and "global.R". The main editor area displays the "server.R" code:

```
1 library(shiny)
2
3 shinyServer( function(input, output) {
4
5
6
7 })
8
9
10
11
12
13
14
15
16
17
18
19
20
```

The "Run App" button is visible in the top right corner of the editor area.

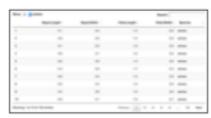
# Review -- How does Shiny work



# render\*() VS. Output()

render\*() and Output() functions work together to add R output to the UI

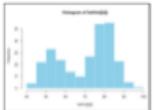
## Server.R



`DT::renderDataTable(expr, options, callback, escape, env, quoted)`



`renderImage(expr, env, quoted, deleteFile)`



`renderPlot(expr, width, height, res, ..., env, quoted, func)`



`renderLeaflet(expr, ...)`

## Ui.R

`dataTableOutput(outputId, icon, ...)`

`imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)`

`plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)`

`leafletOutput(outputId, ...)`

works  
with

# Build Output Object

Server.R

```
1 shinyServer(function(input, output) {  
2  
3   output$outputTable <- DT::renderDataTable({  
4     regressor <- rpart(formula = SalePrice_log ~ ., data = tr,  
5                           control = rpart.control(minsplit = input$minsplit))  
6  
7   # Predicting the Test set results  
8   y_pred <- predict(regressor, newdata = val_set)  
9  
10  data.frame(Parameter = paste0("Decision Tree ; ", " minsplit = ",  
11              RMSE = rmse(val_set$SalePrice_log, y_pred),  
12              stringsAsFactors = FALSE)  
13 } )  
14  
15 } )  
16 } )  
17 }
```

Ui.R

```
1 sidebarLayout(  
2   sidebarPanel(  
3  
4     h3("Decision Tree"),  
5     br(),  
6     numericInput("DT_minsplit", "minsplit", value=1),  
7     numericInput("DT_cp", "cp", value=0.3, min=0, max=1),  
8     numericInput("DT_maxdepth", "maxdepth", value=5)  
9   ),  
10  mainPanel(  
11    DT::dataTableOutput("outputTable")  
12  )  
13 )  
14 )  
15 )  
16 )  
17 )  
18 )  
19 )  
20 )  
21 )  
22 )
```

# Build Output Object

Server.R

```

1 shinyServer(function(input, output) {
2 
3   output$outputTable <- DT::renderDataTable({
4     regressor <- rpart(formula = SalePrice_log ~ .,
5                           control = rpart.control(minsplit=2))
6 
7     # Predicting the Test set results
8     y_pred <- predict(regressor, newdata = val_set)
9 
10    data.frame(Parameter = paste0("Decision Tree ; ", " minsplit =",
11                  RMSE = rmse(val_set$SalePrice_log, y_pred),
12                  stringsAsFactors = FALSE)
13  })
14 
15 }
16 )
17 
```

Ui.R

```

1 ui.R x server.R x global.R x
2 
3 sidebarLayout(
4   sidebarPanel(
5     h3("Decision Tree"),
6     br(),
7     numericInput("DT_minsplit", "minsplit", value=2),
8     numericInput("DT_cp", "cp", value=0.3, min=0, max=1),
9     numericInput("DT_maxdepth", "maxdepth", value=5)
10    ),
11    mainPanel(
12      DT::dataTableOutput("outputTable")
13    )
14  )
15 
16 #Show the RMSE values
17 mainPanel(
18   DT::dataTableOutput("outputTable")
19 )
20 
21 )
22 )|
```

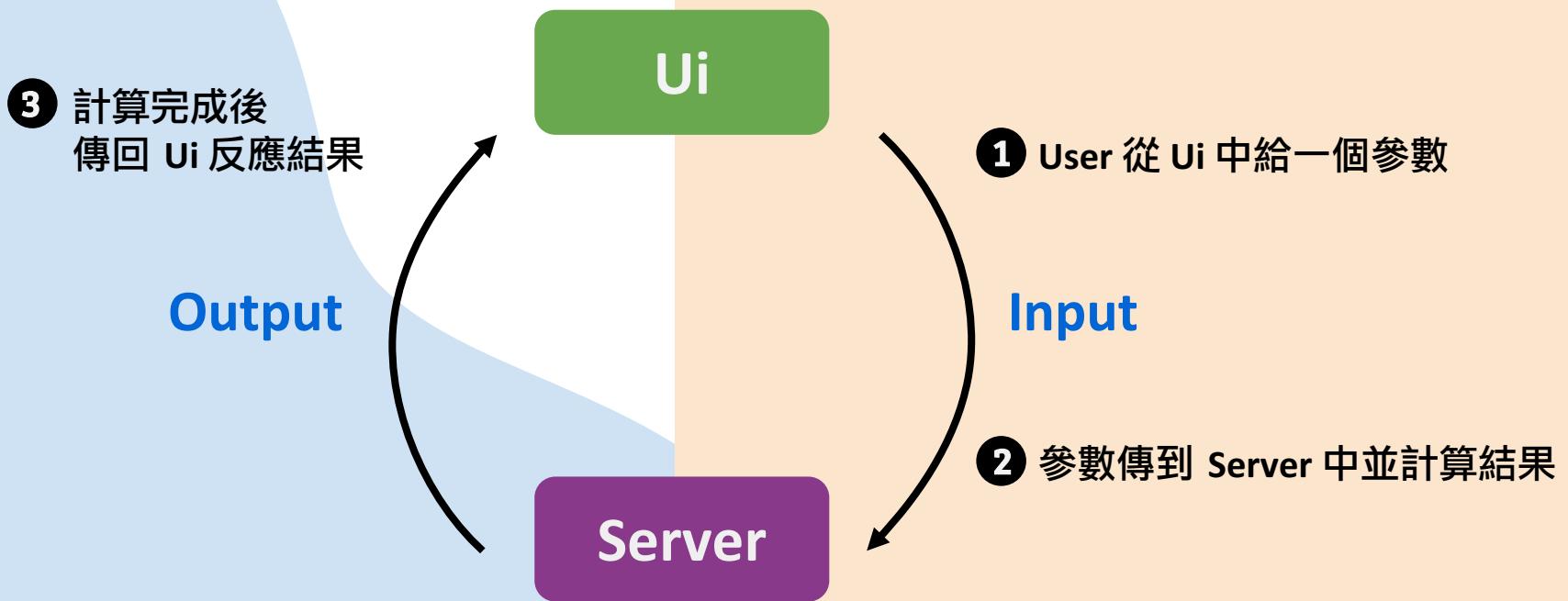
OuputId :  
WaterPlot

# [Code] 我們與 R 的距離 -- Server.R

```
RStudio
ui.R x server.R x global.R x
Run App

1 library(dplyr)
2 shinyServer( function(input, output) {
3
4   output$outputTable <- DT::renderDataTable({
5     regressor <- rpart(formula = SalePrice_log ~ ., data = training_set,
6                           control = rpart.control(minsplit = input$DT_minsplit, cp = input$DT_cp,
7                           maxdepth = input$DT_maxdepth))
8
9     y_pred <- predict(regressor, newdata = val_set)
10
11    data.frame(Parameter = paste0("Decision Tree ; ",
12                                " minsplit : ",input$DT_minsplit," cp : ",input$DT_cp,
13                                " maxdepth : ",input$DT_maxdepth=,
14                                RMSE = rmse(val_set$SalePrice_log, y_pred=,
15                                stringsAsFactors = FALSE)
16
17  })
18})
19
20})
```

# Review -- How does Shiny work



# Shiny App

~/R\_DragonBall - Shiny  
http://127.0.0.1:3525 | Open in Browser

## 我們與 R 的距離

### Decision Tree

Parameter

|   | Parameter  | RMSE              |
|---|--|-------------------|
| 1 | Decision Tree ; minsplit : cp : 0.3 maxdepth : 5 | 0.321844683437133 |

Show 10 entries Search:

Showing 1 to 1 of 1 entries Previous 1 Next

minsplit: 1

cp: 0.3

maxdepth: 5

