

Shiny Application

Your First Shiny App & Cases sharing



Kristen Chan

2018.12.01



#	At/w	Team Name	Score	Entries
1	—	EXL Analytics	0.443793	555
2	—	POWERDOT	0.447651	671
3	—	Dolphin	0.450402	555
4	11	jack3	0.451425	455
5	11	Hopkins Biostat	0.451569	444
6	—	Xing Zhao	0.453081	161

Hello !

I am Kristen Chan

Data Scientist

E-Commerce / Telecom

R-Ladies Taipei

Co-Organizer



Shiny Application

1. **Introduction to Shiny**
2. **Let's Play Shiny**
 - a. **Using Open Data -- 水質監測資料**
(Source: <http://data.taipei/opendata/datalist/datasetMeta?oid=961ca397-4a59-45e8-b312-697f26b059dc>)
 - b. **Azure ML via Shiny**
 - i. What is Azure Machine Learning Studio ?
 - ii. How to connect with Shiny ?
3. **Case Sharing**
 - a. 產生脫貧推薦名單



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:

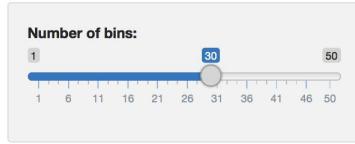
- ✗ Html
- ✗ Javascript
- ✗ CSS

Hello Shiny

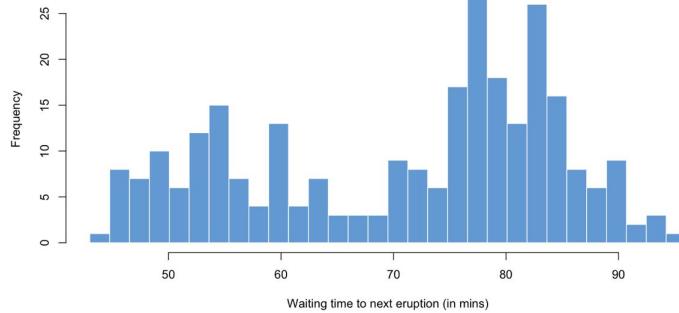


```
library(shiny)
runExample("01_hello")
```

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

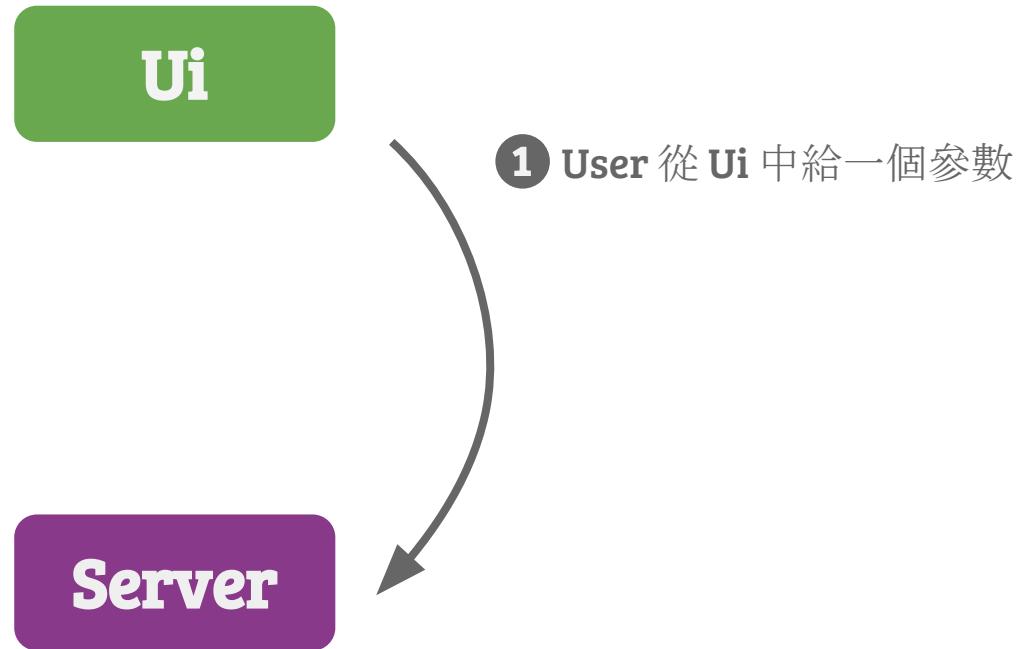
✓ **Ui**

- User interface
- Determine how your app looks

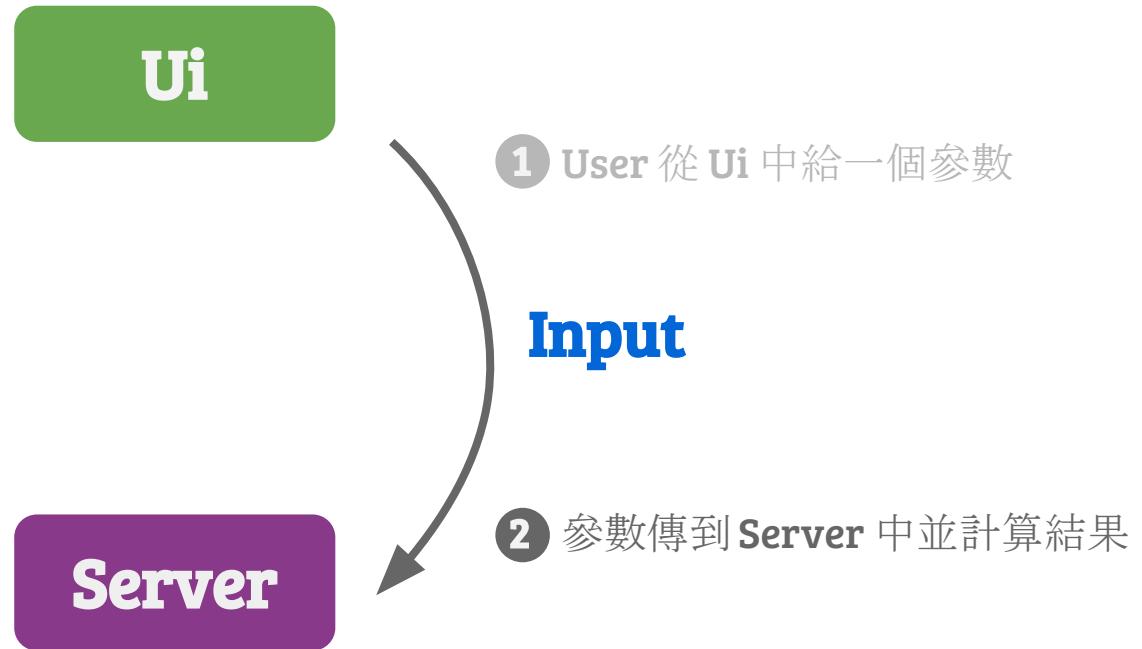
✓ **Server**

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

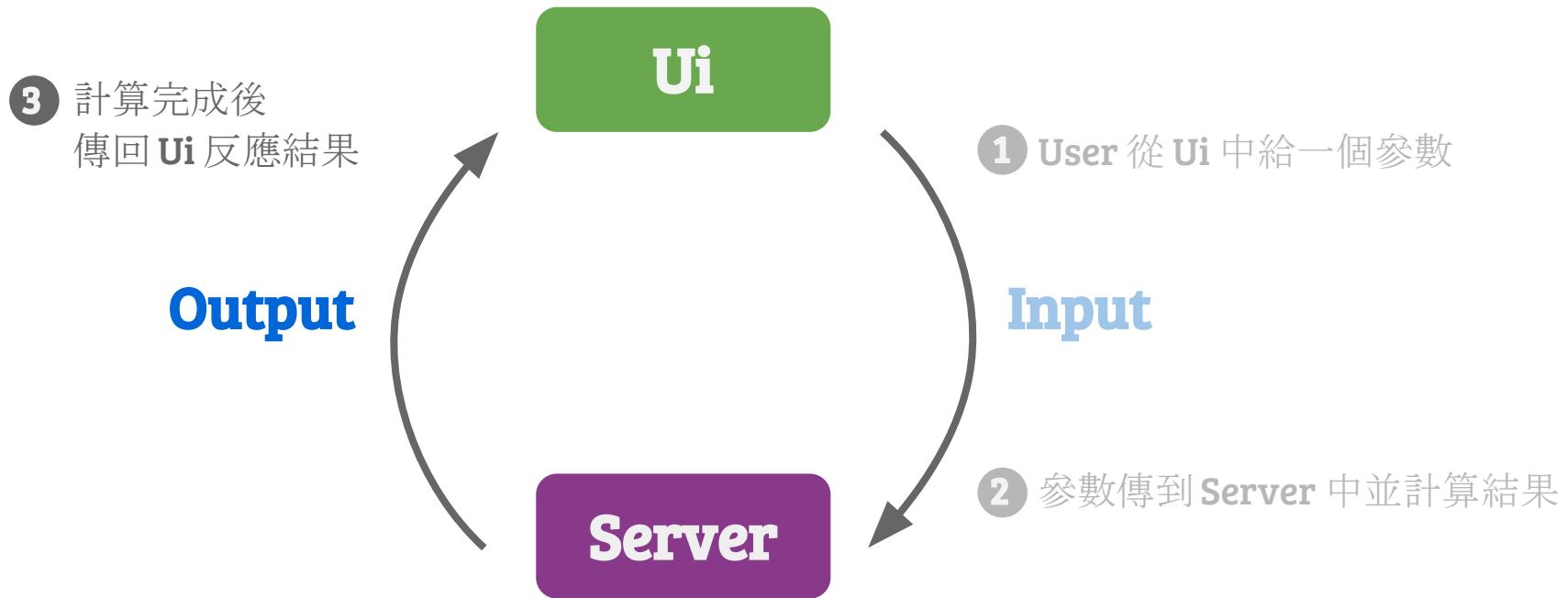
How does Shiny work



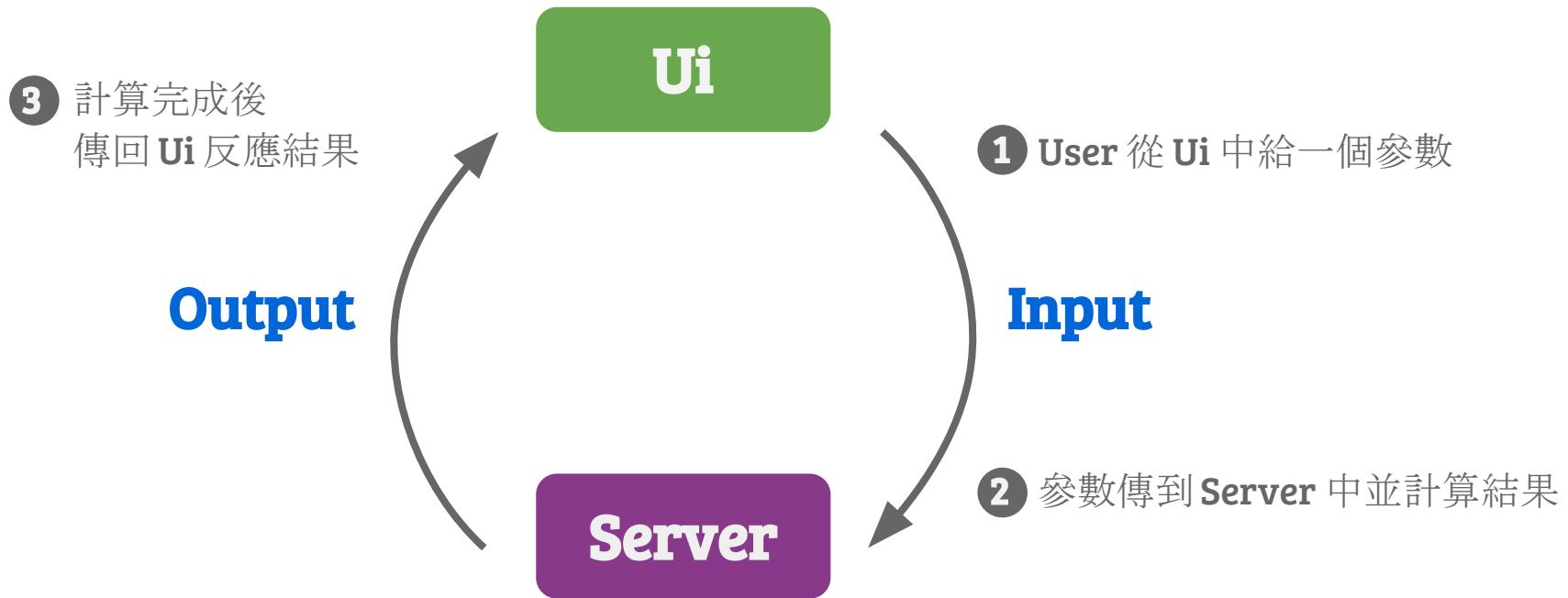
How does Shiny work



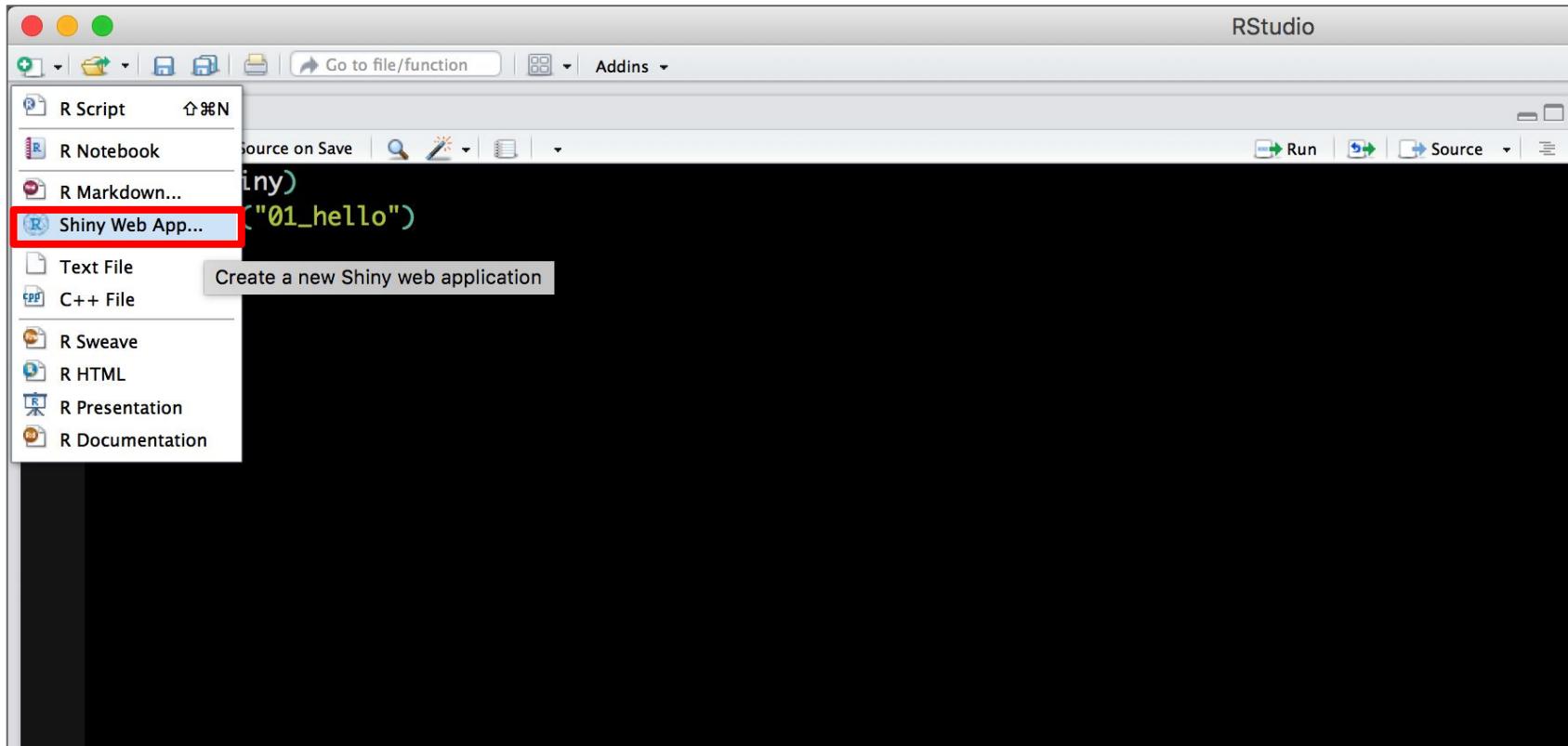
How does Shiny work



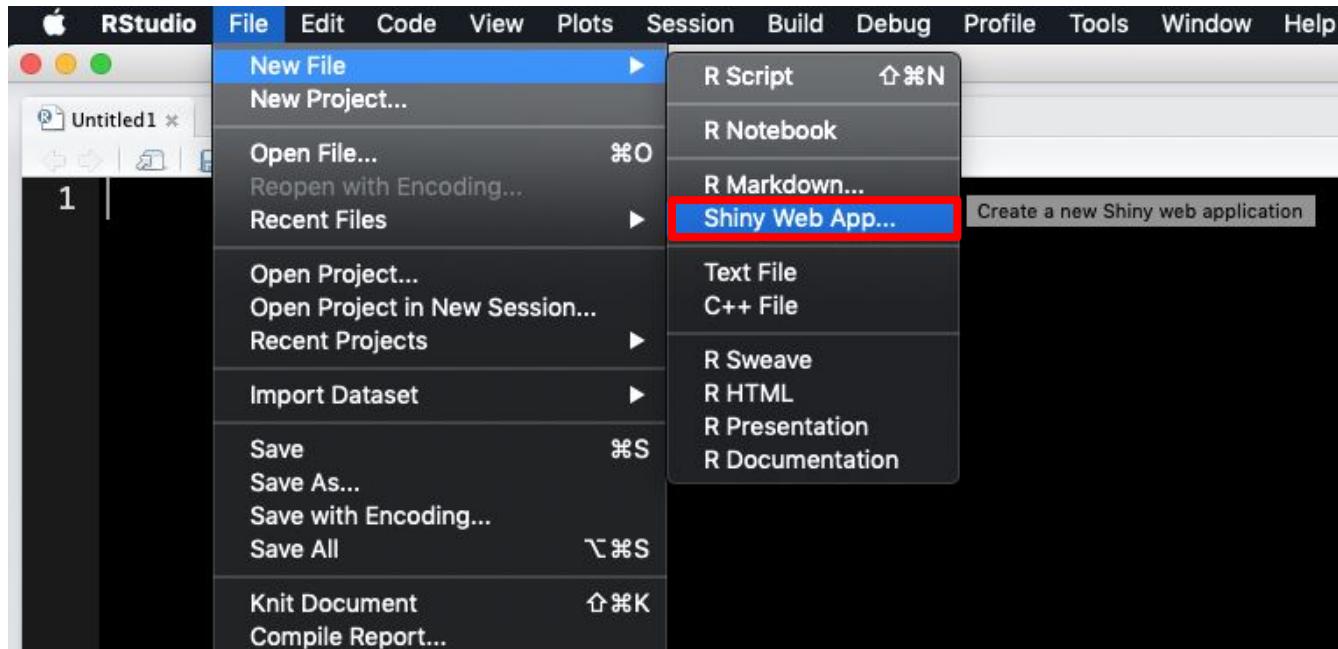
How does Shiny work



Your First Shiny App



Your First Shiny App



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

New Shiny Web Application



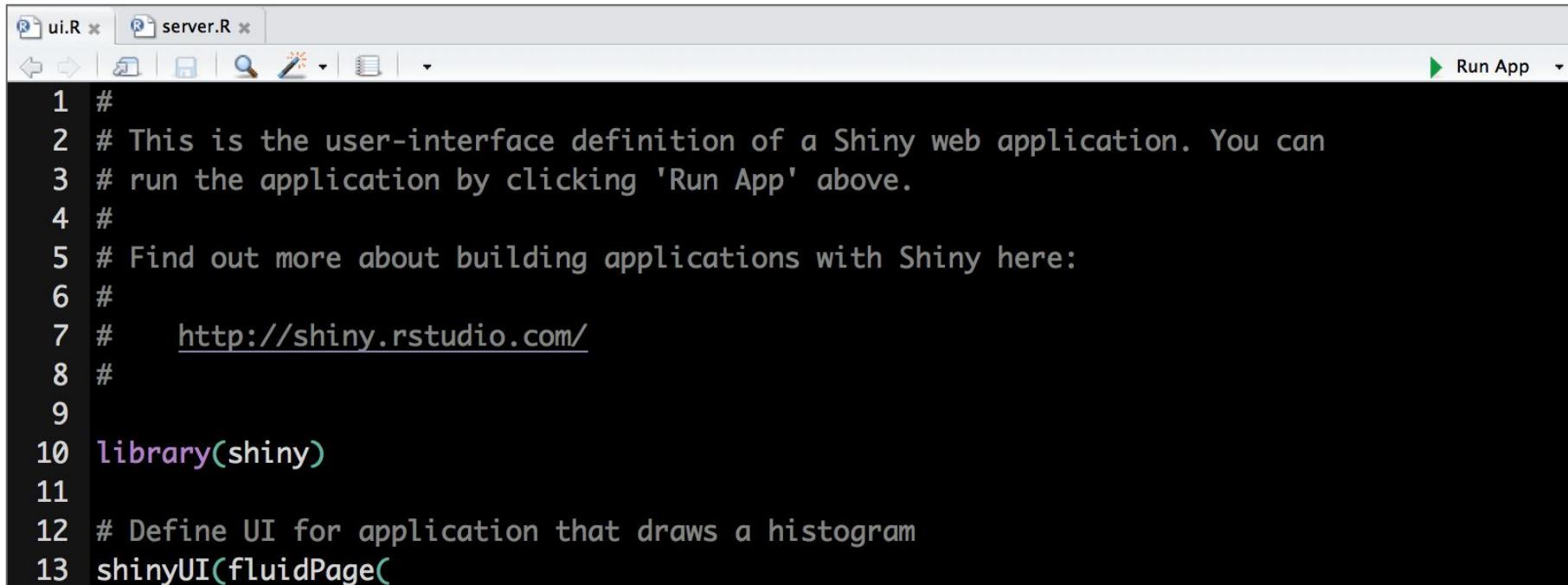
Application name: Enter your app's name

Application type:

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

Create within directory:

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



The screenshot shows the RStudio interface with two files open: ui.R and server.R. The ui.R file contains the user-interface definition for a Shiny application, including comments about running the app and links to the shiny.rstudio.com documentation. The server.R file is partially visible at the top. The RStudio toolbar includes standard icons for file operations, search, and run.

```
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)

New Shiny Web Application



Application name: Enter your app's name

Application type: Single File (app.R) Multiple File (ui.R/server.R)

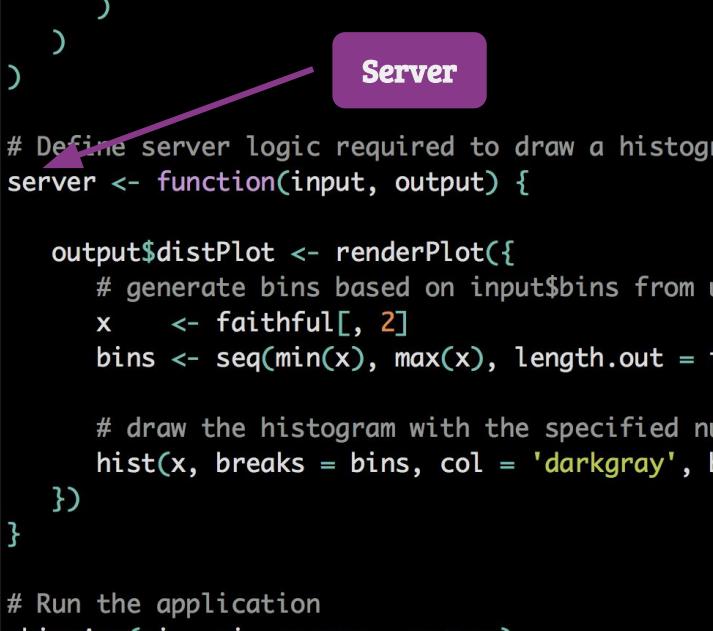
Create within directory:

[? Shiny Web Applications](#)

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

```
1 #  
2 # This is a Shiny web application. You can run the a  
3 # the 'Run App' button above.  
4 #  
5 # Find out more about building applications with Shiny  
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",
```



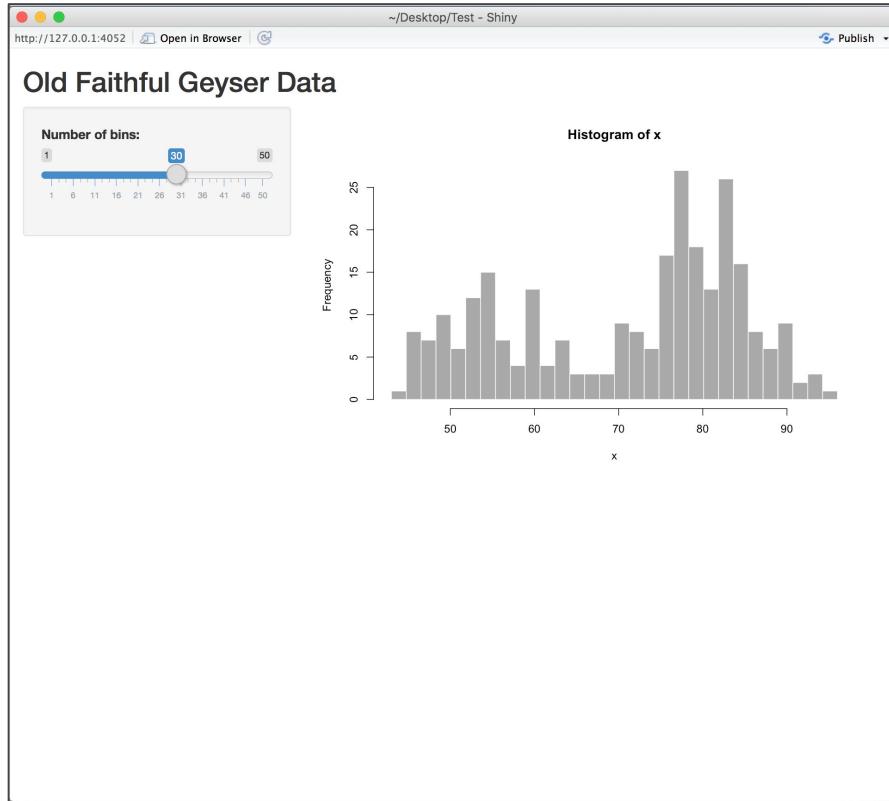


The screenshot shows the RStudio interface with the file `app.R` open. The code defines a Shiny application with a user interface and a server function. A purple callout box labeled "Server" points to the start of the server logic, which begins at line 35.

```
50 plotOutput("distPlot")  
51  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 shinyApp(ui = ui, server = server)  
150 }  
151 }
```



Your First Shiny App





Let's Play Shiny

Open Data -- 水質監測資訊

臺北市水質監測資訊

資料集提供機關



臺北自來水事業處

分享至

Google+

Twitter

Facebook

臺北市水質監測資訊

意見反應
友善列印

分類及編號：生活安全及品質

資料集描述：臺北市自來水事業處各水質監測站，所測得水質之濁度、餘氯、酸鹼度等數據15分鐘平均值

主要欄位說明：1.update_date : 日期 2.update_time : 時間 3.qua_id : 監控站ID 4.code_name : 監控站名稱 5.longitude : 經度 6.latitude : 緯度 7.qua_ntu : 濁度(NTU) 8.qua_cl : 餘氯(mg/L) 9.qua_ph : 酸度(pH) 10.偵測值-9係維修中，暫停偵測。

資料集類型：原始資料
最後更新時間：2018-09-03 17:04:02

資料項目

水質監測資訊	使用資料
水質監測資訊(新版)	使用資料

水質監測

詮釋資料

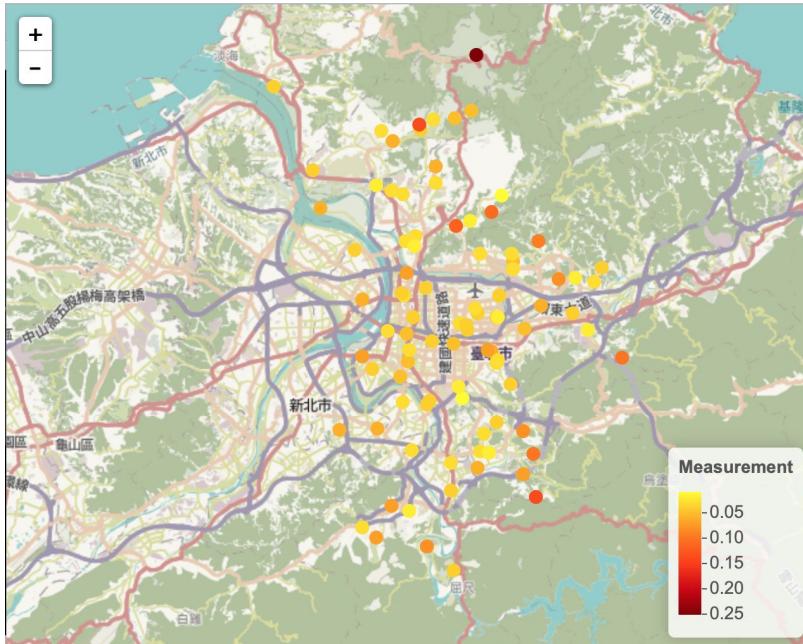
分類及編號	生活安全及品質
資料集描述	臺北市自來水事業處各水質監測站，所測得水質之濁度、餘氯、酸鹼度等數據15分鐘平均值
主要欄位說明	1.update_date : 日期 2.update_time : 時間 3.qua_id : 監控站ID 4.code_name : 監控站名稱 5.longitude : 經度 6.latitude : 緯度 7.qua_ntu : 濁度(NTU) 8.qua_cl : 餘氯(mg/L) 9.qua_ph : 酸度(pH) 10.偵測值-9係維修中，暫停偵測。
資料集類型	原始資料
最後更新時間	2018-09-03 17:04:02
資料量	91
收錄期間(起)	

Open Data -- 水質監測資訊

欄位說明

- update_date : 日期
- update_time : 時間
- qua_id : 監控站ID
- code_name : 監控站名稱
- longitude : 經度
- latitude: 緯度
- qua_cntu: 濁度(NTU)
- qua_cl: 餘氯(mg/L)
- qua_ph: 酸度(pH)

台北市水質濁度(NTU)狀況圖



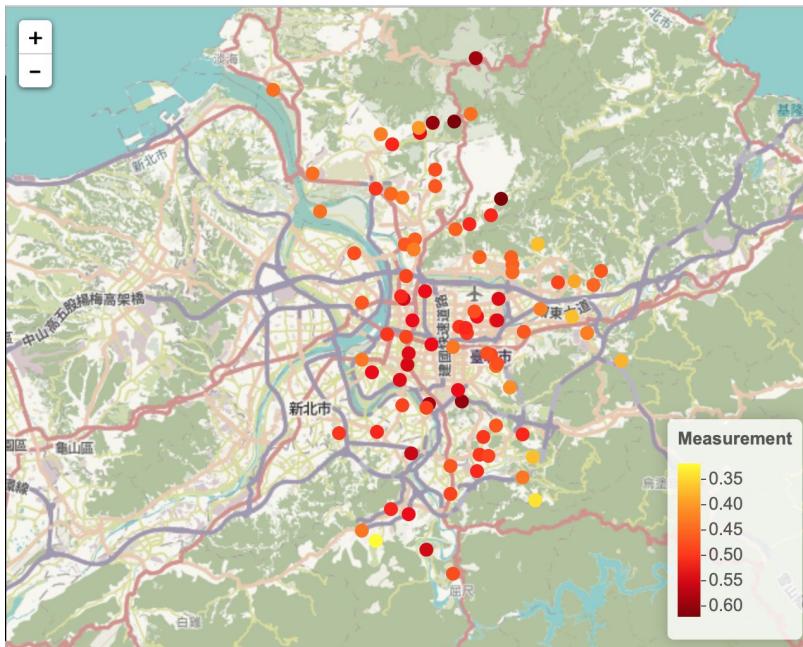
<https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=190796c8-7c56-42e0-8068-39242b8ec927>

```
#載入套件
library(httr)
library(jsonlite)
library(RCurl)
library(dplyr)
library(leaflet)
#讀入資料
data_URL <- "API"
text <- content(GET(data_URL), as = "text", encoding = "UTF-8")
WaterData <- fromJSON(text)
WaterDataFrame <- WaterData$result$results
WaterDataFrame$longitude <- as.numeric(WaterDataFrame$longitude)
WaterDataFrame$latitude <- as.numeric(WaterDataFrame$latitude)
WaterDataFrame$qua_cntu <- as.numeric(WaterDataFrame$qua_cntu)
WaterDataFrame <- WaterDataFrame[which(WaterDataFrame$qua_cntu>0),]
#畫圖
pal <- colorNumeric(palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                     domain = WaterDataFrame$qua_cntu )
TaipeiMap = leaflet(WaterDataFrame) %>%
  addProviderTiles("OpenStreetMap.HOT") %>%
  setView( lng=121.52, lat=25.0449, zoom=11 ) %>%
  addCircleMarkers( lng=~longitude, lat=~latitude ,radius=3, popup=~code_name,
                    color=~pal(qua_cntu), opacity=1,
                    fillColor=~pal(qua_cntu), fillOpacity=1)%>%
  addLegend(title="Measurement", "bottomright",
            pal=pal, values=~qua_cntu ,opacity=1 )
TaipeiMap
```

Kyjen Chan

台北市水質餘氯(mg/L)狀況圖

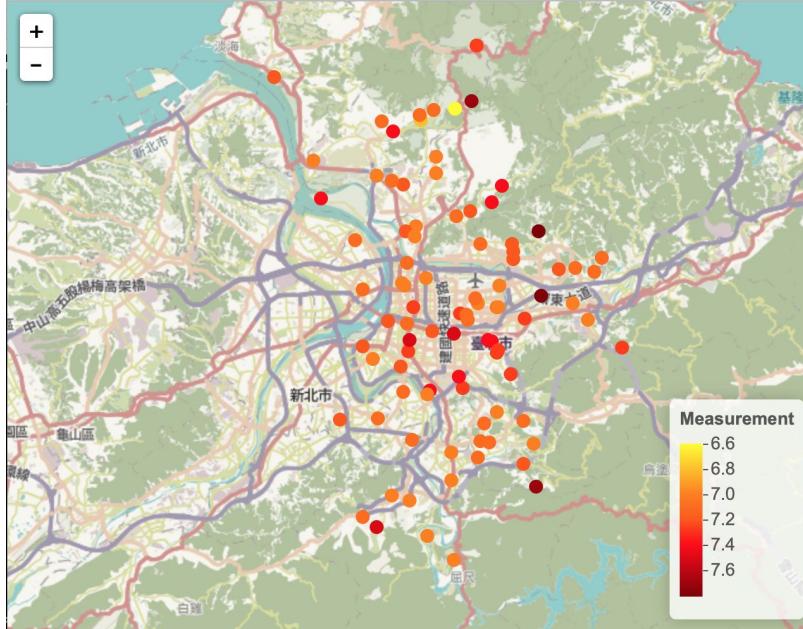
<https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=190796c8-7c56-42e0-8068-39242b8ec927>



```
#載入套件
library(httr)
library(jsonlite)
library(RCurl)
library(dplyr)
library(leaflet)
#讀入資料
data_URL <- "API"
text <- content(GET(data_URL), as = "text", encoding = "UTF-8")
WaterData <- fromJSON(text)
WaterDataFrame <- WaterData$result$results
WaterDataFrame$longitude <- as.numeric(WaterDataFrame$longitude)
WaterDataFrame$latitude <- as.numeric(WaterDataFrame$latitude)
WaterDataFrame$qua_cl <- as.numeric(WaterDataFrame$qua_cl)
WaterDataFrame <- WaterDataFrame[which(WaterDataFrame$qua_cl>0),]
#畫圖
pal <- colorNumeric(palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                     domain = WaterDataFrame$qua_cl )
TaipeiMap = leaflet(WaterDataFrame) %>%
  addProviderTiles("OpenStreetMap.HOT") %>%
  setView( lng=121.52, lat=25.0449, zoom=11 ) %>%
  addCircleMarkers( lng=~longitude, lat=~latitude ,radius=3, popup=~code_name,
                    color=~pal(qua_cl), opacity=1,
                    fillColor=~pal(qua_cl), fillOpacity=1)%>%
  addLegend(title="Measurement", "bottomright",
            pal=pal, values=~qua_cl ,opacity=1 )
TaipeiMap
```

Kyjen Chan

台北市水質酸度(pH)狀況圖

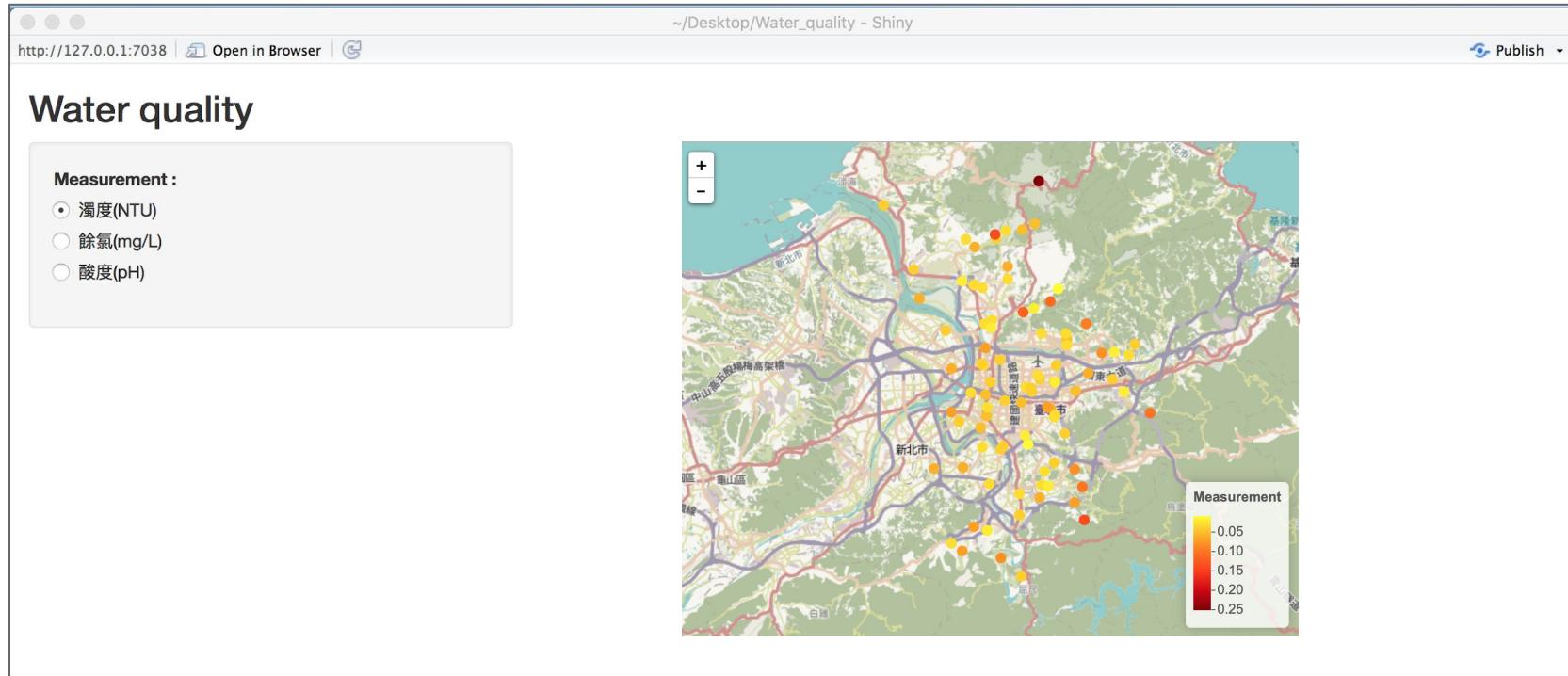


<https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=190796c8-7c56-42e0-8068-39242b8ec927>

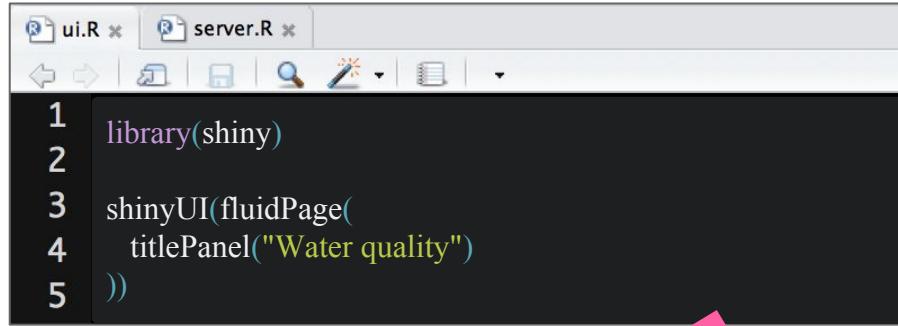
```
#載入套件
library(httr)
library(jsonlite)
library(RCurl)
library(dplyr)
library(leaflet)
#讀入資料
data_URL <- "API"
text <- content(GET(data_URL), as = "text", encoding = "UTF-8")
WaterData <- fromJSON(text)
WaterDataFrame <- WaterData$result$results
WaterDataFrame$longitude <- as.numeric(WaterDataFrame$longitude)
WaterDataFrame$latitude <- as.numeric(WaterDataFrame$latitude)
WaterDataFrame$qua_ph <- as.numeric(WaterDataFrame$qua_ph)
WaterDataFrame <- WaterDataFrame[which(WaterDataFrame$qua_ph>0),]
#畫圖
pal <- colorNumeric( palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                      domain = WaterDataFrame$qua_ph )
TaipeiMap = leaflet(WaterDataFrame) %>%
  addProviderTiles("OpenStreetMap.HOT") %>%
  setView( lng=121.52, lat=25.0449, zoom=11 ) %>%
  addCircleMarkers( lng=~longitude, lat=~latitude ,radius=3, popup=~code_name,
                    color=~pal(qua_ph), opacity=1,
                    fillColor=~pal(qua_ph), fillOpacity=1 )%>%
  addLegend(title="Measurement", "bottomright",
            pal=pal, values=~qua_ph ,opacity=1 )
TaipeiMap
```

Kyjen Chan

[Shiny] Water quality



[Code] Water quality – title



A screenshot of the RStudio interface showing the ui.R file. The code is as follows:

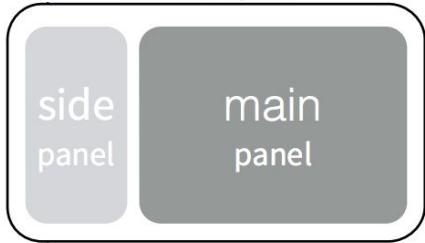
```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Water quality")
5 ))
```



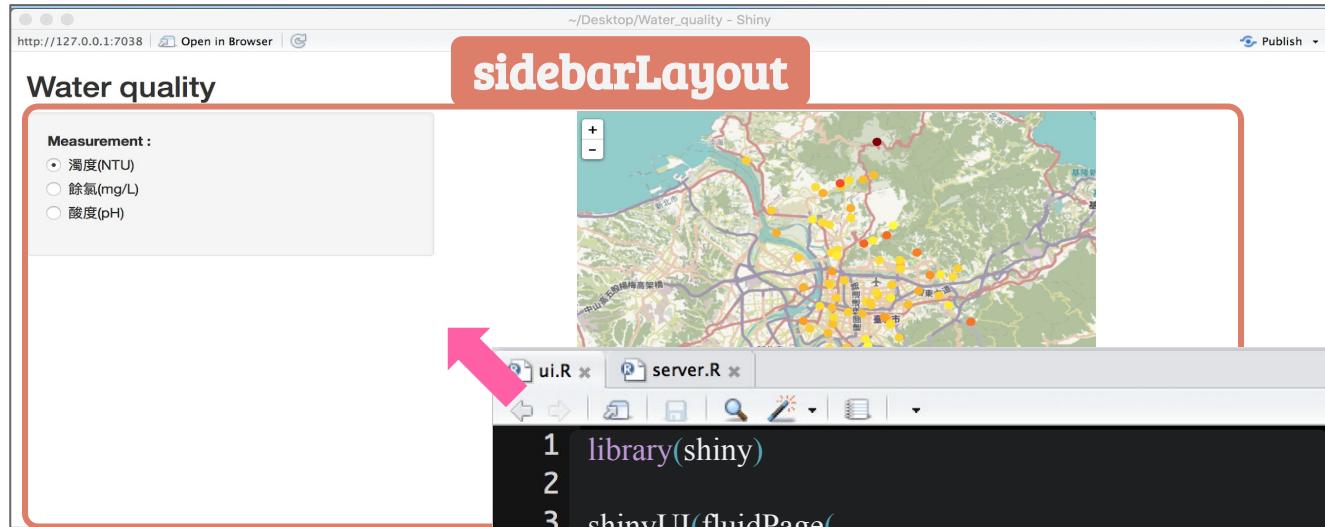
titlePanel → Water quality

[Layout] Water quality

sidebarLayout

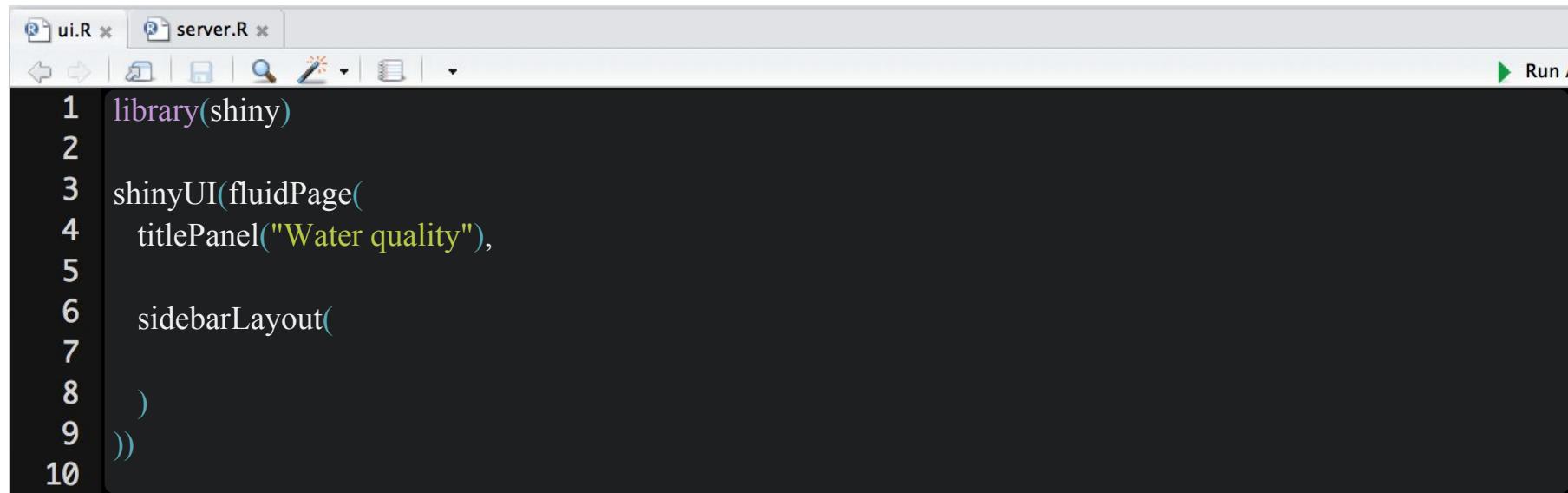


```
shinyUI(fluidPage(  
  sidebarLayout(  
    sidebarPanel(...),  
    mainPanel(...)  
  ))
```



```
1 library(shiny)  
2  
3 shinyUI(fluidPage(  
4   titlePanel("Water quality"),  
5  
6   sidebarLayout(  
7  
8   ))  
9  
10 )
```

[Code] Water quality -- Layout



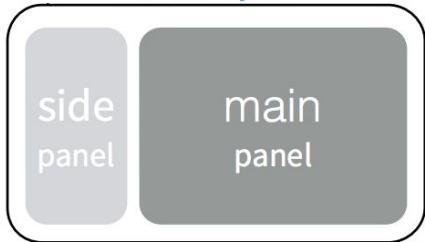
The screenshot shows the RStudio interface with the ui.R file open. The code is as follows:

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Water quality"),
5
6   sidebarLayout(
7
8   )
9 ))
10
```

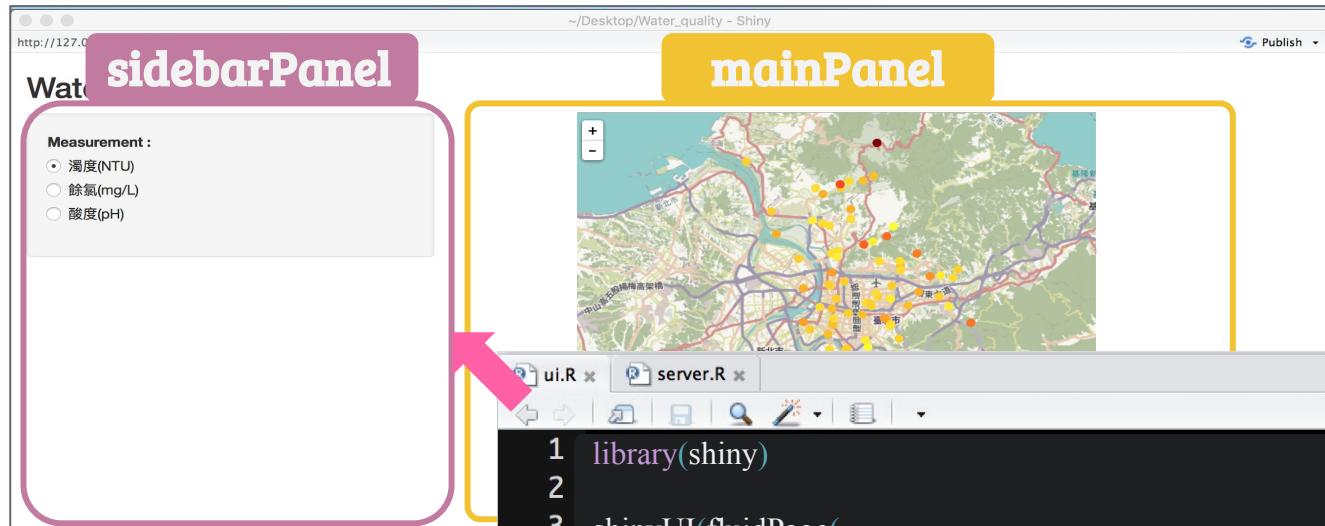
The code defines a shiny application with a title panel titled "Water quality" and a sidebar layout.

[Layout] Water quality

sidebarLayout

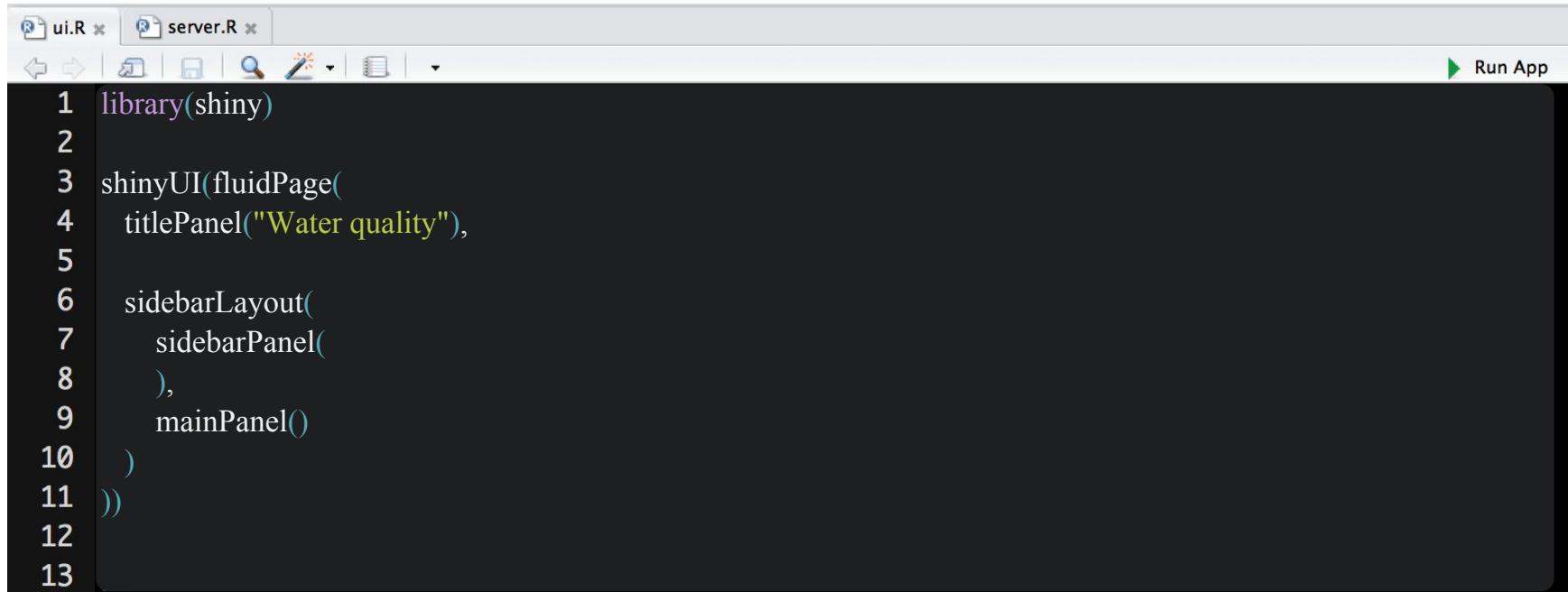


```
shinyUI(fluidPage(  
  sidebarLayout(  
    sidebarPanel(...),  
    mainPanel(...)  
  ))
```



```
1 library(shiny)  
2  
3 shinyUI(fluidPage(  
4   titlePanel("Water quality"),  
5  
6   sidebarLayout(  
7     sidebarPanel(),  
8     mainPanel()  
9   ))  
10 ))
```

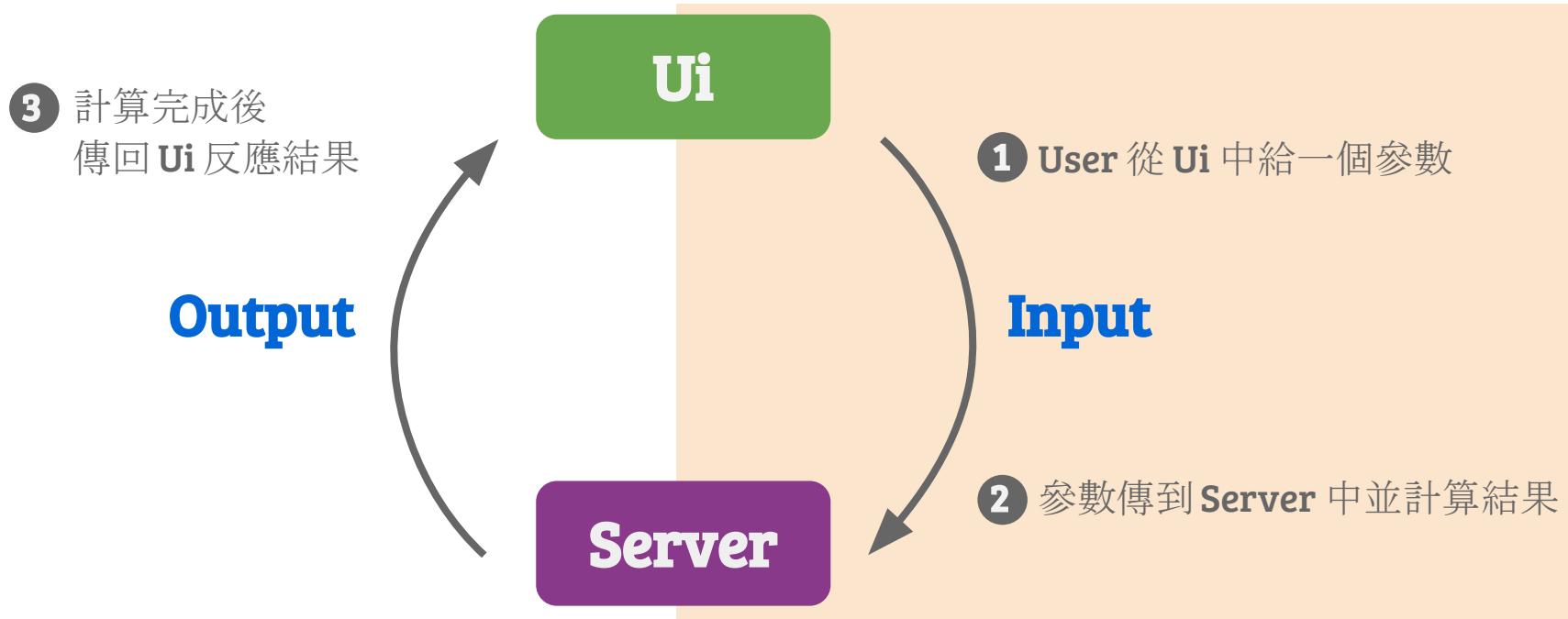
[Code] Water quality -- Layout



The screenshot shows the RStudio interface with the ui.R file open. The code displays the basic structure for a sidebar-based Shiny application.

```
ui.R server.R Run App
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Water quality"),
5
6   sidebarLayout(
7     sidebarPanel(
8       ),
9     mainPanel()
10    )
11  )))
12
13
```

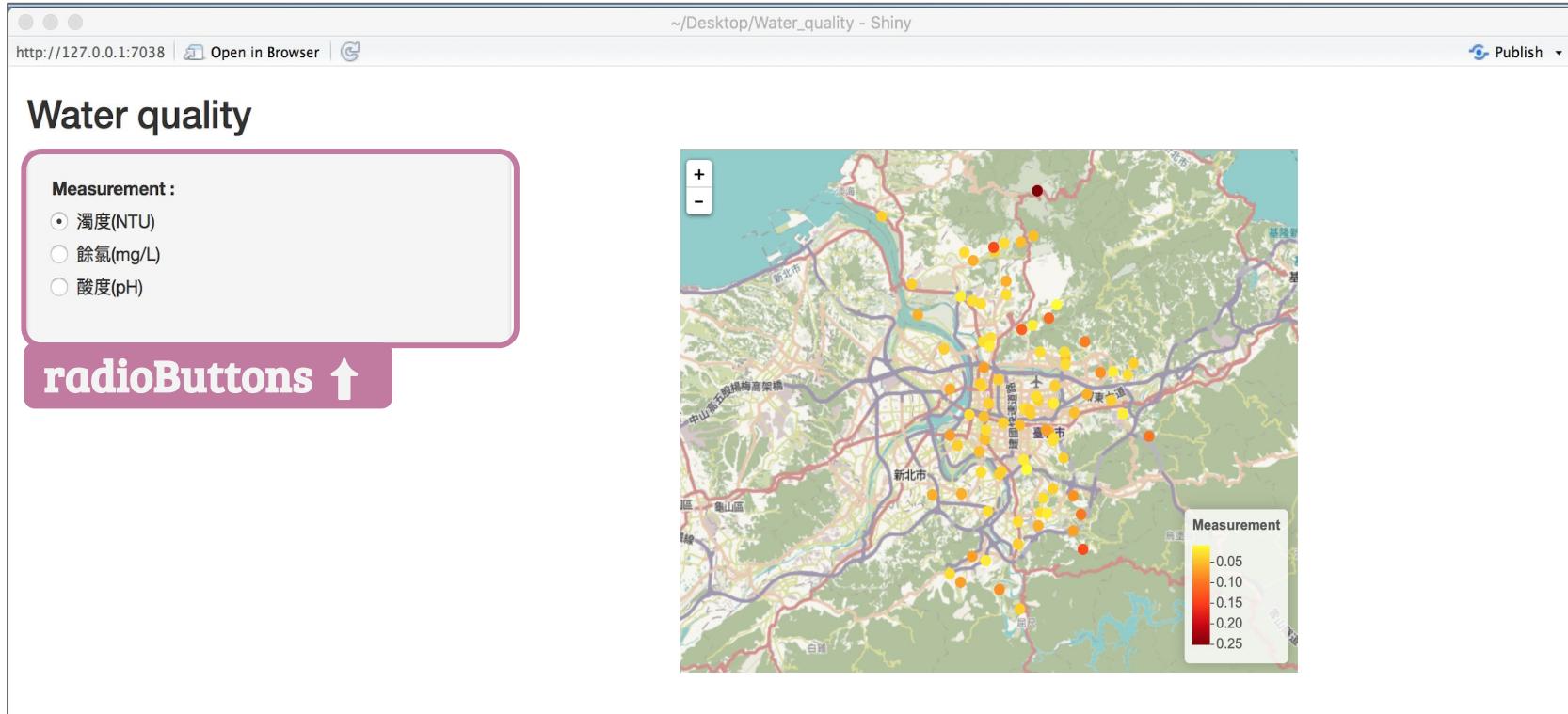
[Review] How does Shiny work



Shiny Input -- sidebarPanel

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

[Layout] Water quality



[Code] Water quality -- input

```
R ui.R x R server.R x
◀ ▶ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌍ Run App ▶
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Water quality"),
5
6   sidebarLayout(
7     sidebarPanel(
8       radioButtons("measurement", "Measurement :",
9                   c( "濁度(NTU)" = "qua_cntu",
10                     "餘氯(mg/L)" = "qua_cl" ,
11                     "酸度(pH)" = "qua_ph" ))
12     ),
13     mainPanel( )
14   )
15 ))
16
```

Water quality -- Example:Measurement

```
radioButtons("measurement", "Measurement : ",
```

```
  c( "濁度(NTU)" = "qua_cntu",  
    "餘氯(mg/L)" = "qua_cl" ,  
    "酸度(pH)" = "qua_ph" ) )
```

inputId

choices

label

Measurement : ← label

濁度(NTU)

餘氯(mg/L)

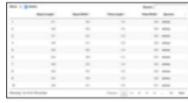
酸度(pH)

← choices



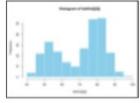
Shiny Output – mainPanel

a



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

b



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

c



`uiOutput(outputId, inline, container, ...)`

d

```
'data.Frame': 3 obs. of  2 variables:
$ Sepal.Length: num  5.1 4.9 4.7
$ Sepal.Width : num  3.5 3.0 3.2
```

`verbatimTextOutput(outputId)`

e

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.10	3.50	1.60	0.10	Iris-setosa
2	4.90	3.00	1.40	0.30	Iris-setosa
3	4.70	3.20	1.30	0.20	Iris-setosa
4	4.50	2.30	1.00	0.20	Iris-setosa
5	4.50	2.30	1.50	0.20	Iris-setosa
6	4.50	2.30	1.50	0.30	Iris-setosa
7	4.60	3.00	1.40	0.20	Iris-setosa
8	5.00	3.40	1.50	0.20	Iris-setosa
9	5.40	3.90	1.70	0.40	Iris-setosa

`tableOutput(outputId)`

f



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

g

foo

`textOutput(outputId, container, inline)`

h



`leafletOutput(outputId,....)`

`leafletOutput("plot_name")`

the type of output
to display

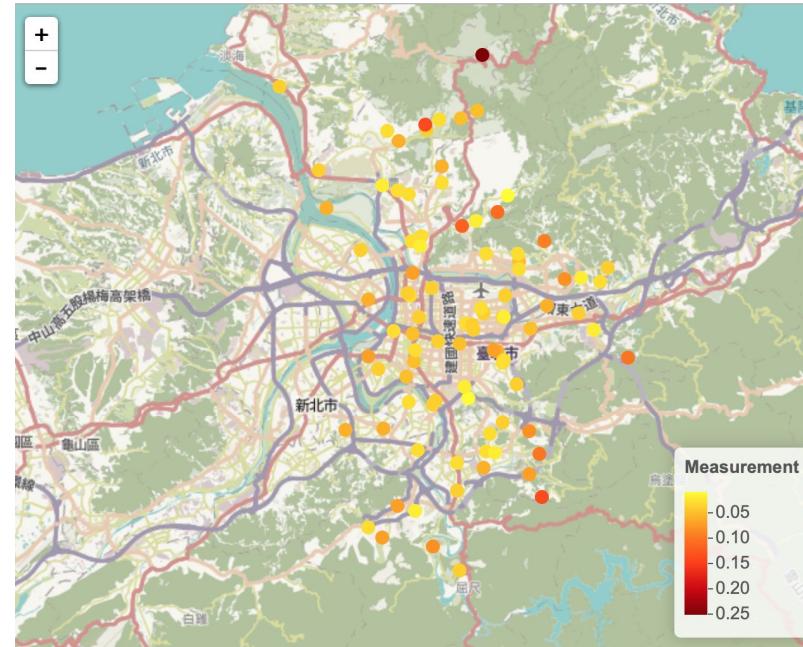
name to give to the
output object

[Code] Water quality

[Code] Water quality – leafletOutput

```
leafletOutput("WaterPlot")
```

↓
outputId



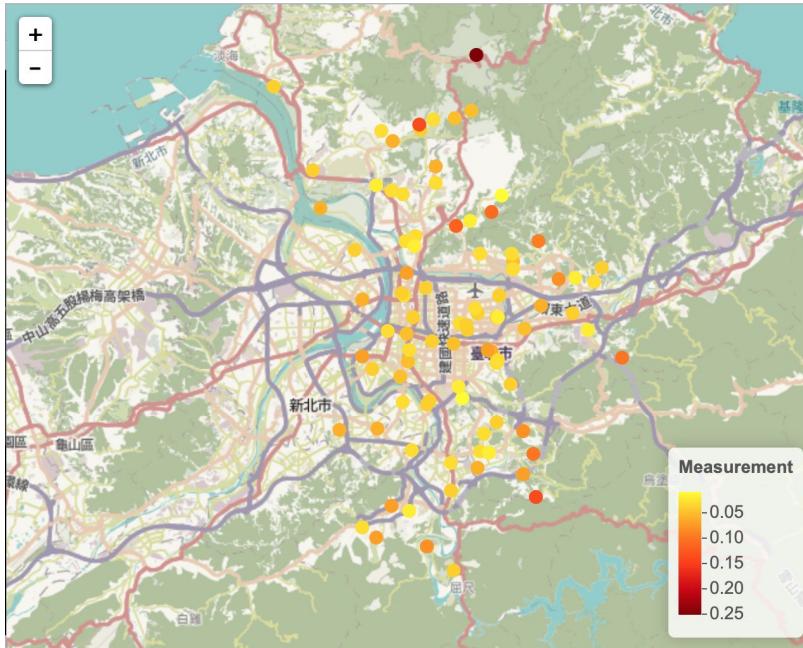
[Code] Water quality -- Original Server.R



The screenshot shows the RStudio interface with the 'server.R' file open. The code in the editor is:

```
1 library(shiny)
2
3 shinyServer( function(input, output) {
4
5
6
7 })
```

[Review] 台北市水質濁度(NTU)狀況圖



```
#載入套件
library(httr)
library(jsonlite)
library(RCurl)
library(dplyr)
library(leaflet)
#讀入資料
data_URL <- "API"
text <- content(GET(data_URL), as = "text", encoding = "UTF-8")
WaterData <- fromJSON(text)
WaterDataFrame <- WaterData$result$results
WaterDataFrame$longitude <- as.numeric(WaterDataFrame$longitude)
WaterDataFrame$latitude <- as.numeric(WaterDataFrame$latitude)
WaterDataFrame$qua_cntu <- as.numeric(WaterDataFrame$qua_cntu)
WaterDataFrame <- WaterDataFrame[which(WaterDataFrame$qua_cntu>0),]
#畫圖
pal <- colorNumeric(palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                     domain = WaterDataFrame$qua_cntu )
TaipeiMap = leaflet(WaterDataFrame) %>%
  addProviderTiles("OpenStreetMap.HOT") %>%
  setView( lng=121.52, lat=25.0449, zoom=11 ) %>%
  addCircleMarkers( lng=~longitude, lat=~latitude ,radius=3, popup=~code_name,
                    color=~pal(qua_cntu), opacity=1,
                    fillColor=~pal(qua_cntu), fillOpacity=1)%>%
  addLegend(title="Measurement", "bottomright",
            pal=pal, values=~qua_cntu ,opacity=1 )
TaipeiMap
```

Kyjen Chan

https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=190796c8-7c56-42e0-8068-39242b8ec927

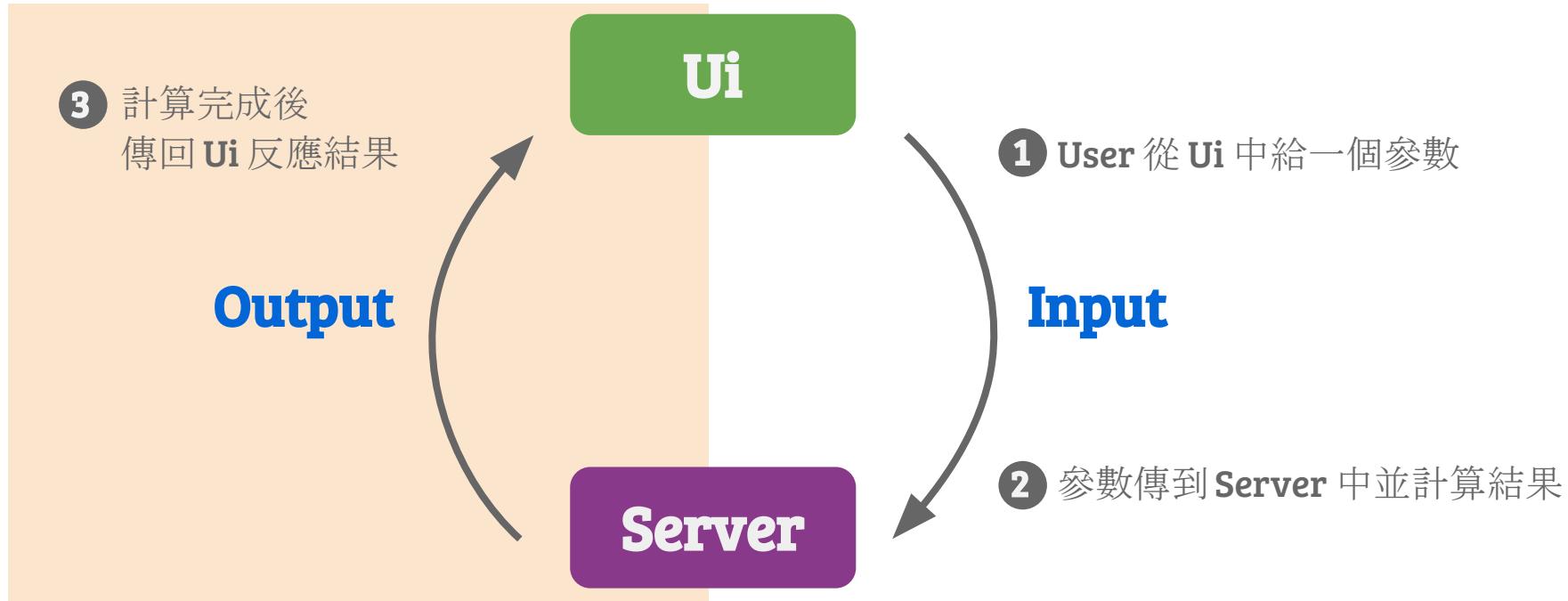
[Code] Water quality

The screenshot shows the RStudio interface with the 'server.R' file open. The code is as follows:

```
1 library(shiny)
2 library(httr)
3 library(jsonlite)
4 library(RCurl)
5 library(leaflet)
6 library(dplyr)
7 shinyServer( function(input, output) {
8   data_URL <- "API"
9   text <- content(GET(data_URL), as = "text", encoding = "UTF-8")
10  WaterData <- fromJSON(text)
11  WaterDataFrame <- WaterData$result$results
12  WaterDataFrame$longitude <- as.numeric(WaterDataFrame$longitude)
13  WaterDataFrame$latitude <- as.numeric(WaterDataFrame$latitude)
14  WaterDataFrame$qua_cntu <- as.numeric(WaterDataFrame$qua_cntu)
15  WaterDataFrame$qua_cl <- as.numeric(WaterDataFrame$qua_cl)
16  WaterDataFrame$qua_ph <- as.numeric(WaterDataFrame$qua_ph)
17 })
18
```

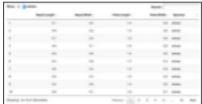
A blue arrow points from the URL in the browser bar to the 'data_URL' variable in the R code, highlighting the connection between the API endpoint and the variable used in the script.

[Review] How does Shiny work



render*() VS. Output()

Server.R

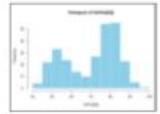


DT::renderDataTable(expr, options, callback, escape, env, quoted)

renderImage(expr, env, quoted, deleteFile)

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

renderLeaflet(expr,...)



Ui.R

works
with

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,...)

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

Build Output Object

Server.R

```
ui.R * server.R *
20  ===== Output =====
21  output$WaterPlot <- renderLeaflet({
22      map_data = WaterDataFrame %>%
23          select(longitude, latitude, code_name, quality_measurement)
24      filter(quality_measurement>0)
25      ## -- Map --
26      pal <- colorNumeric(palette = c("yellow",
27                                  domain = map_data$quality_measurement))
28      TaipeiMap = leaflet(map_data) %>%
29          addProviderTiles("OpenStreetMap.HOT") %
30          setView(lng=121.52, lat=25.0449, zoom=12) %
31          addCircleMarkers(
32              longitude, latitude,
33              color=~pal(quality_measurement),
34              fillColor=~pal(quality_measurement),
35              addLegend(title="Measurement", "bottomright",
36              pal=pal, values=~quality_measurement)
37      )
38  }
39  
```

Ui.R

```
ui.R * server.R *
1 library(shiny)
2
3 shinyUI(fluidPage(
4     titlePanel("Water quality"),
5
6     sidebarLayout(
7         sidebarPanel(
8             radioButtons("measurement", "Measurement : ",
9                         c("濁度(NTU)" = "qua_cntu",
10                           "餘氯(mg/L)" = "qua_cl",
11                           "酸度(pH)" = "qua_ph"))
12         ),
13         mainPanel(leafletOutput("WaterPlot"))
14     )
15 ))
```

Output Object Name

Server.R

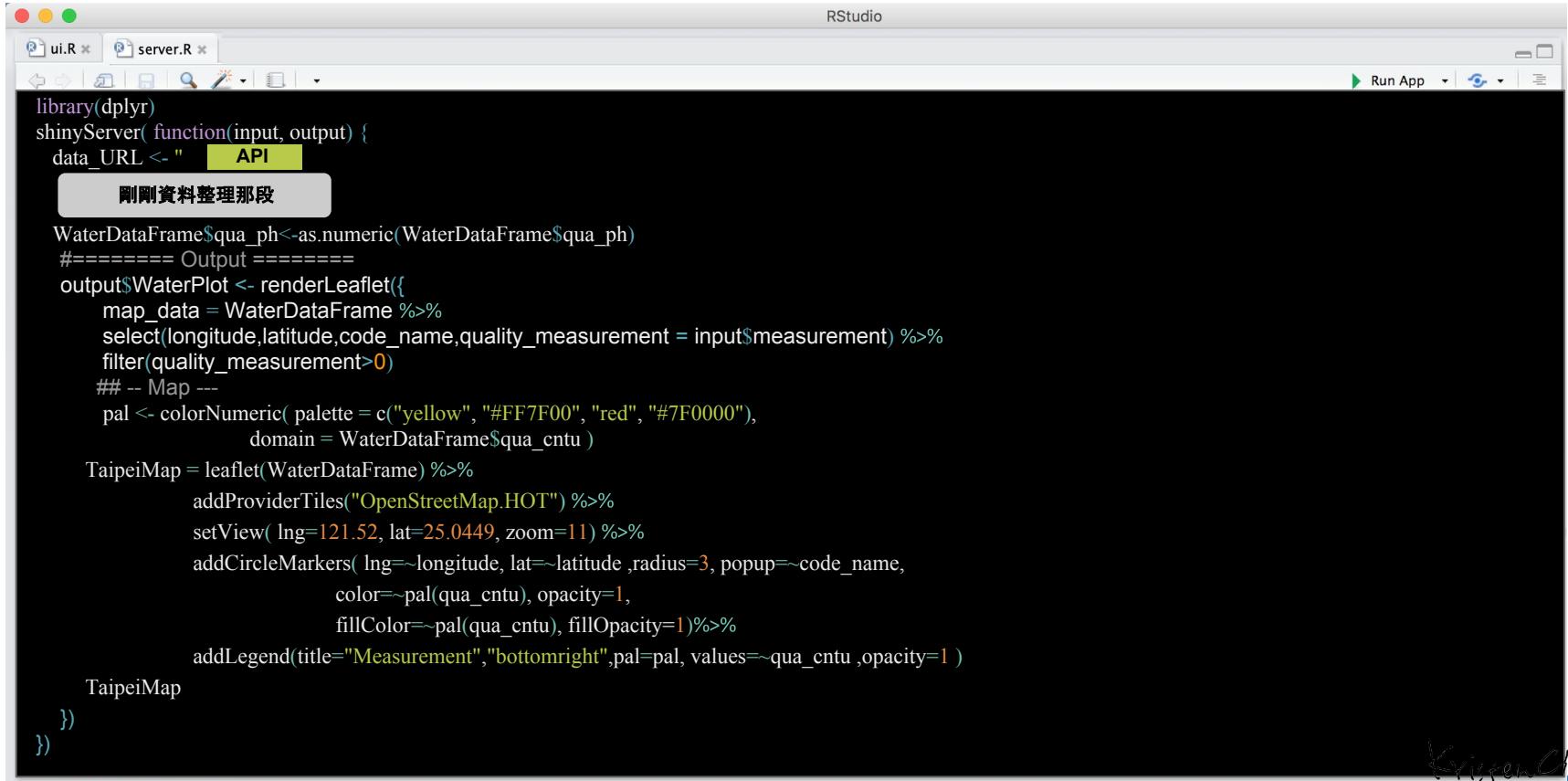
```
ui.R * server.R *
20 #===== Output =====
21 outputs$WaterPlot <- renderLeaflet(
22   map_data = WaterDataFrame,
23   select(longitude, latitude, code_name, quality_measurement),
24   filter(quality_measurement > 0)
25   ## -- Map --
26   pal <- colorNumeric(palette = c("yellow",
27                             domain = map_data$quality_measurement))
28   TaipeiMap = leaflet(map_data) %>%
29     addProviderTiles("OpenStreetMap.HOT") %
30     setView(lng=121.52, lat=25.0449, zoom=12) %
31     addCircleMarkers(
32       longitude, latitude,
33       color=~pal(quality_measurement),
34       fillColor=~pal(quality_measurement),
35       addLegend(title="Measurement", "bottomright",
36                 pal=pal, values=~quality_measurement)
37     )
38 
```

OuputId:
WaterPlot

Ui.R

```
ui.R * server.R *
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Water quality"),
5
6   sidebarLayout(
7     sidebarPanel(
8       radioButtons("measurement", "Measurement : ",
9                   c("濁度(NTU)" = "qua_cntu",
10                     "餘氯(mg/L)" = "qua_cl",
11                     "酸度(pH)" = "qua_ph"))
12     ),
13     mainPanel( leafletOutput("WaterPlot") )
14   )
15 ))
```

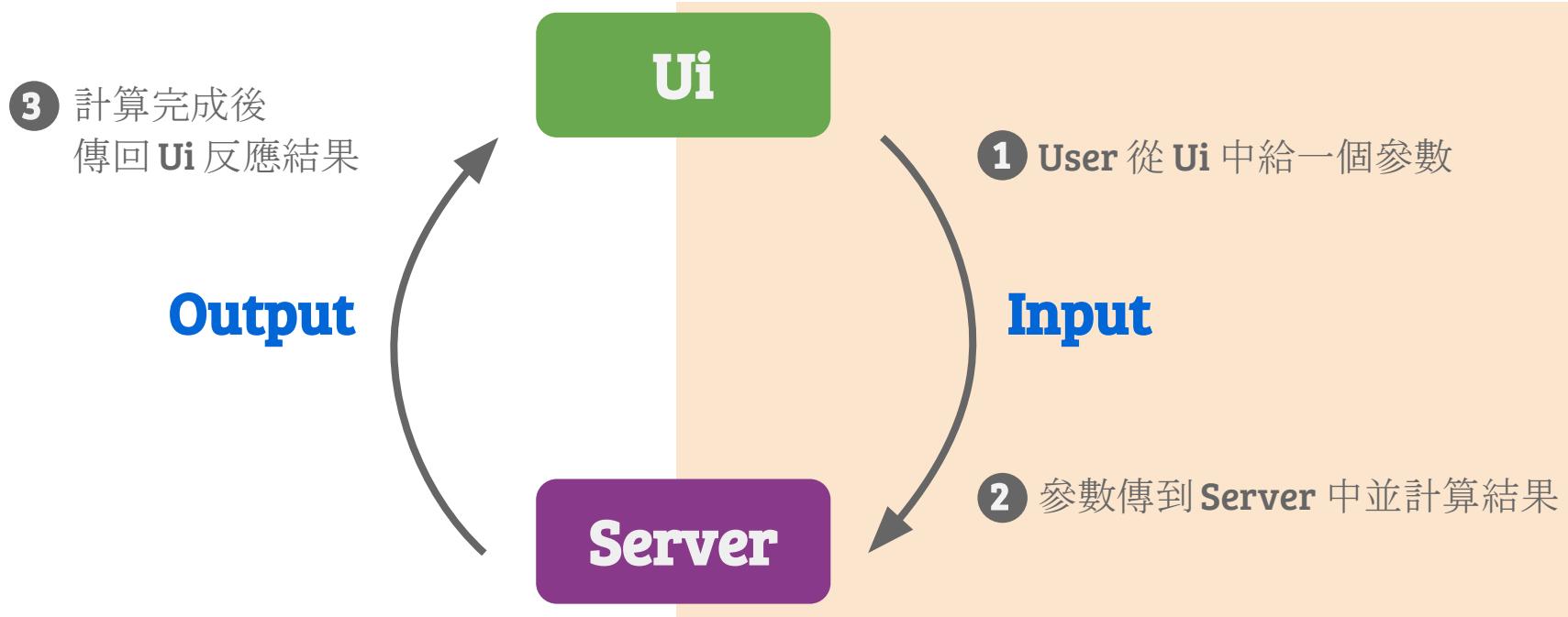
[Code] Shiny



The screenshot shows the RStudio interface with two tabs open: ui.R and server.R. The server.R tab contains the following R code:

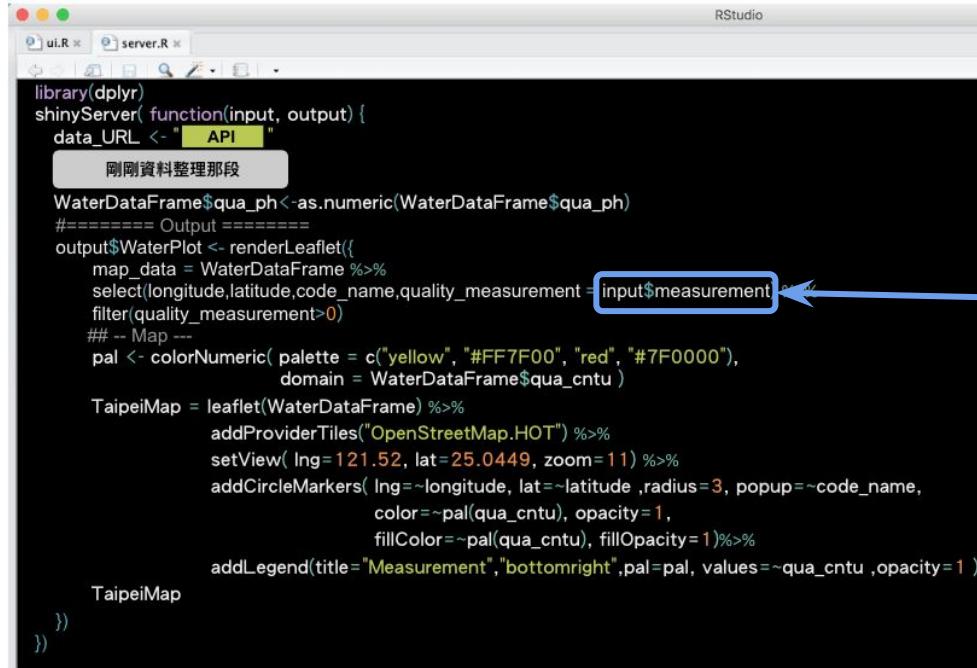
```
library(dplyr)
shinyServer(function(input, output) {
  data_URL <- "API"
  剛剛資料整理那段
  WaterDataFrame$qua_ph <- as.numeric(WaterDataFrame$qua_ph)
  #===== Output ======
  output$WaterPlot <- renderLeaflet({
    map_data = WaterDataFrame %>%
      select(longitude, latitude, code_name, quality_measurement = input$measurement) %>%
      filter(quality_measurement > 0)
    ## -- Map --
    pal <- colorNumeric(palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                        domain = WaterDataFrame$qua_ctnu)
    TaipeiMap = leaflet(WaterDataFrame) %>%
      addProviderTiles("OpenStreetMap.HOT") %>%
      setView(lng=121.52, lat=25.0449, zoom=11) %>%
      addCircleMarkers(lng=~longitude, lat=~latitude, radius=3, popup=~code_name,
                      color=~pal(qua_ctnu), opacity=1,
                      fillColor=~pal(qua_ctnu), fillOpacity=1) %>%
      addLegend(title="Measurement", "bottomright", pal=pal, values=~qua_ctnu, opacity=1)
    TaipeiMap
  })
})
```

[Review] How does Shiny work



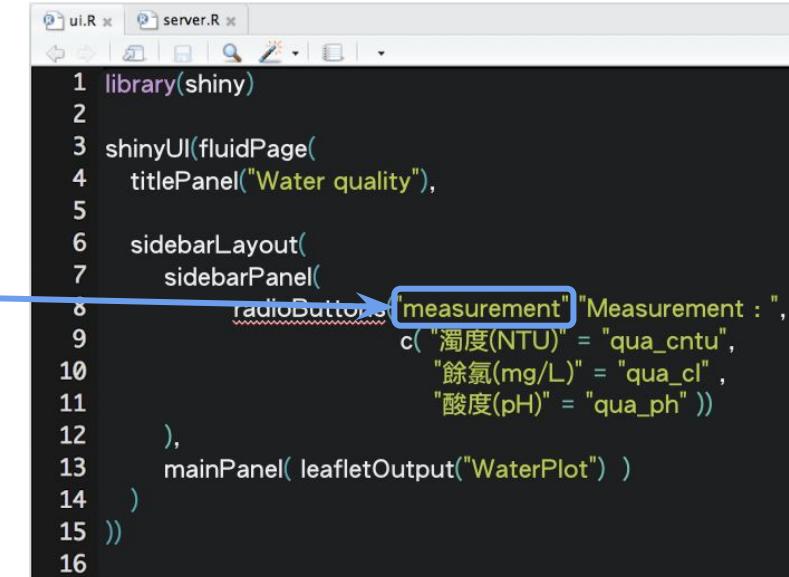
Access Input Value

Server.R



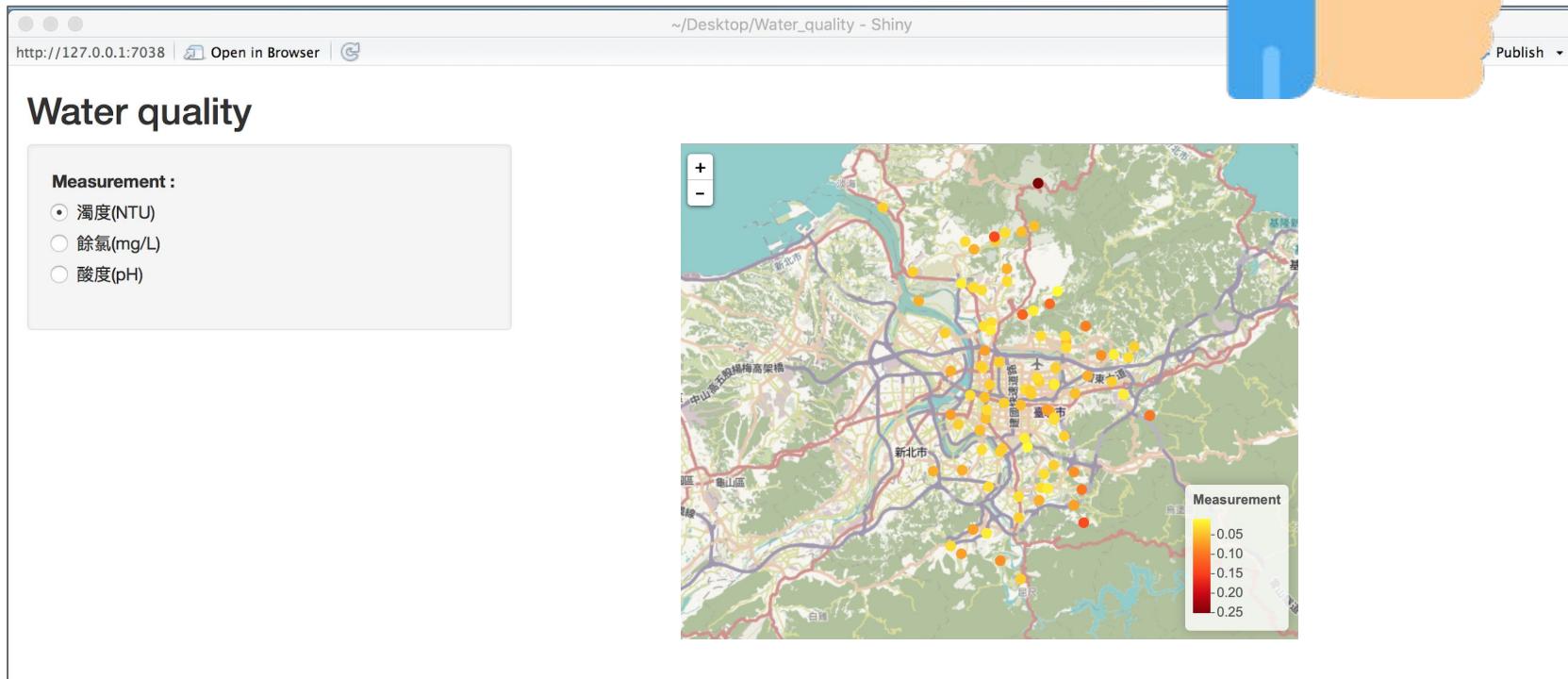
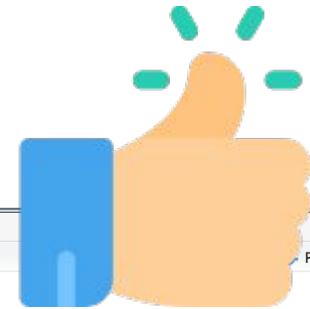
```
library(dplyr)
shinyServer( function(input, output) {
  data_URL <- "API"
  剛剛資料整理那段
  WaterDataFrame$qua_ph<-as.numeric(WaterDataFrame$qua_ph)
  ===== Output =====
  output$WaterPlot <- renderLeaflet({
    map_data = WaterDataFrame %>%
      select(longitude,latitude,code_name,quality_measurement = input$measurement) %>%
      filter(quality_measurement>0)
    ## -- Map --
    pal <- colorNumeric( palette = c("yellow", "#FF7F00", "red", "#7F0000"),
                         domain = WaterDataFrame$qua_cntu )
    TaipeiMap = leaflet(WaterDataFrame) %>%
      addProviderTiles("OpenStreetMap.HOT") %>%
      setView( lng=121.52, lat=25.0449, zoom=11) %>%
      addCircleMarkers( lng=~longitude, lat=~latitude ,radius=3, popup=~code_name,
                        color=~pal(qua_cntu), opacity=1,
                        fillColor=~pal(qua_cntu), fillOpacity=1)%>%
      addLegend(title="Measurement", "bottomright",pal=pal, values=~qua_cntu ,opacity=1 )
    TaipeiMap
  })
})
```

Ui.R



```
library(shiny)
shinyUI(fluidPage(
  titlePanel("Water quality"),
  sidebarLayout(
    sidebarPanel(
      radioButtons("measurement", "Measurement :",
                  c( "濁度(NTU)" = "qua_cntu",
                     "餘氯(mg/L)" = "qua_cl",
                     "酸度(pH)" = "qua_ph" )))
    mainPanel( leafletOutput("WaterPlot") )
  )
))
```

[Shiny] Water quality



Azure ML via Shiny

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish ▾

Titanic Survival Prediction

Passenger Class :

1

Gender :

female

Port Embarkation :

C

Age :

80

Sibling Spouse :

8

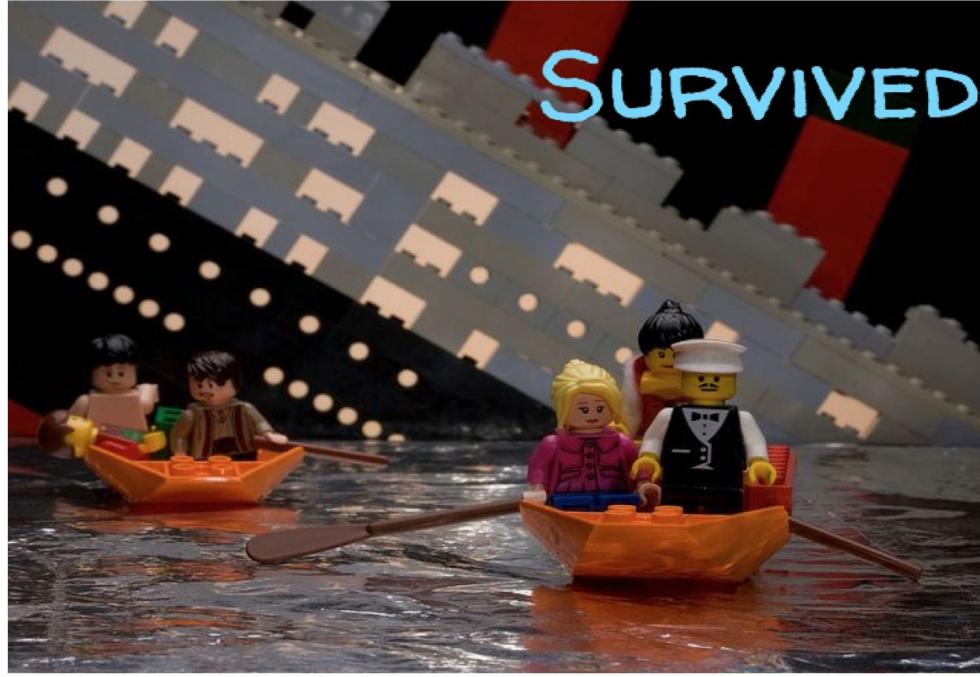
Parent Child :

9

Fare Price :

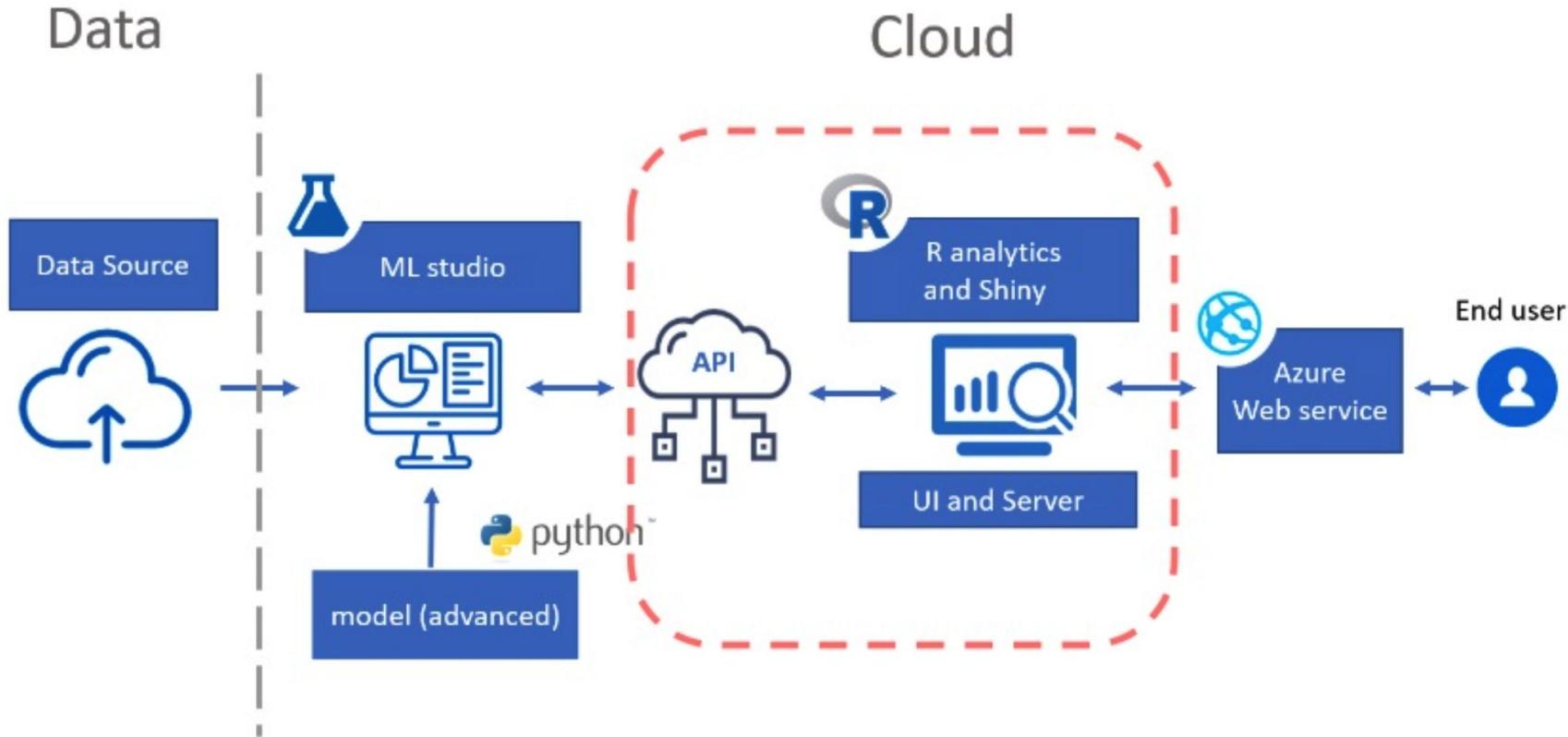
0 512.3292

512.3292



Kristen Chan 51

Data



On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew.

[Note] About Titanic Data

Variable	Definition	Key
survival	Survial	0=No , 1=Yes
pclass	Ticket Class	1=1st , 2=2nd , 3=3rd
embarked	Port of embarkation	C = Cherbourg , Q = Queenstown , S = Southampton
age	Age in years	
sibsp	#of siblings / spouses aboard the Titanic	
parch	#of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
sex	Sex	



[補] Azure ML Step

Azure Machine Learning Studio

Note

<https://studio.azureml.net/>

≡ Microsoft Azure Machine Learning Studio

?



Welcome to Azure
Machine Learning

Try it for free

No Azure subscription? No credit card? No problem! Choose anonymous Guest Access, or sign in with your work or school account, or a Microsoft account.

[Sign In](#)

Not an Azure ML user?

[Sign up here](#)

[Pricing & FAQ](#)

By using this free version, you agree to be bound by the Microsoft Azure Website Terms of Use.

Announcements

Mining Campaign Funds

Aired on August 02, 2017

Play with 2016 Presidential Campaign finance data while learning how to prepare a large dataset for machine learning by processing and engineering features. This sample experiment works on a 2.5 GB dataset and will take about 20 minutes to run in its entirety.

[Learn More](#)

Inside the Data Science VM

Aired on June 21, 2016

Dsvm is a custom Azure Virtual Machine image that is published on the Azure marketplace and available on both Windows and Linux. It contains several popular data science and development tools both from Microsoft and from the open source community all pre-installed and pre-configured and ready to use. We will cover best practices that would show how you can use the Dsvm effectively to run your next data science or analytics project.

[Learn More](#)

Predictive Maintenance Modeling

Aired on July 05, 2016

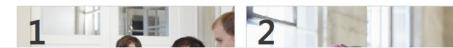
Predictive maintenance is one of the top demanded applications of predictive modelling and is seen as a life-saver in asset-heavy industries such as manufacturing and aerospace due to its potential to provide significant cost savings by reducing downtime due to mechanical problems. We will share what data science techniques are best equipped for this sort of analytics.

[Learn More](#)

More

What's New

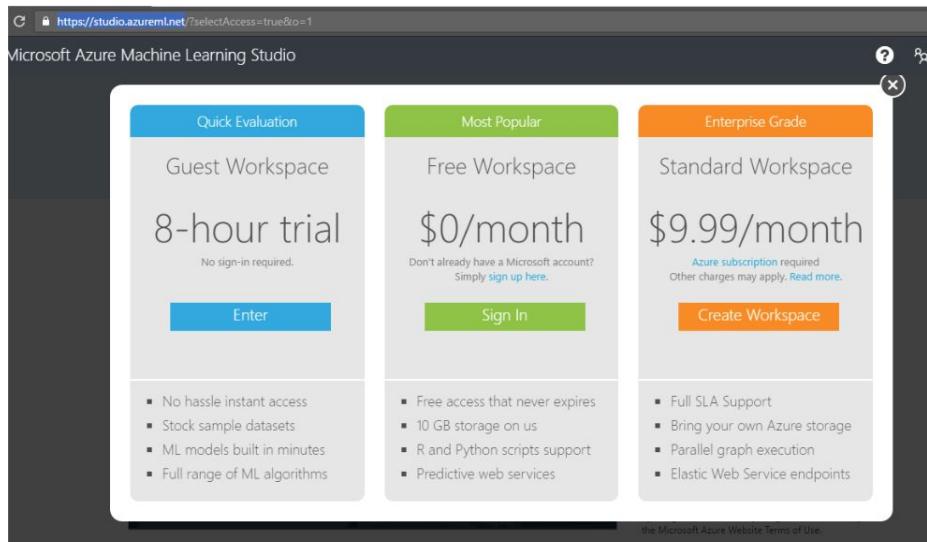
[New text analytics modules](#)



Azure Machine Learning Studio

Note <https://studio.azureml.net/?selectAccess=true&o=2>

- 使用 Microsoft Account 登入, 選擇 Free Workspace
- 不登入使用 Guest Workspace



Create an Experiment

Microsoft Azure Machine Learning Studio

agogomei-Free-Workspace ? 🚧 😊 🌐

PROJECTS

EXPERIMENTS

WEB SERVICES

NOTEBOOKS

DATASETS

TRAINED MODELS

SETTINGS

1

NEW

experiments

MY EXPERIMENTS SAMPLES

NAME	AUTHOR	STATUS	LAST EDITED	PROJECT
Ning_044_UsedCarLinear - C...	agogomei	Draft	1/31/2018 11:50:46 AM	None
3inputExperiment created on ...	agogomei	Finished	12/16/2017 7:08:25 AM	None
3inputExperiment created on ...	agogomei	Finished	12/16/2017 7:07:51 AM	None
102 Linear Regression Stock ...	agogomei	Draft	12/4/2017 7:29:44 PM	None
UsedCarLinear	ningchencontact	Draft	12/3/2017 4:53:32 PM	None
SinputExperiment	agogomei	Finished	11/26/2017 4:00:26 PM	None
UsedCarLinear_DecisionRegr...	agogomei	Finished	11/26/2017 2:19:35 PM	None
Experiment created on 2017/...	agogomei	Finished	11/26/2017 11:47:55 AM	None
UsedCarLinear_SVM - Remov...	agogomei	Draft	11/26/2017 11:39:42 AM	None
Experiment created on 2017/...	agogomei	Failed	11/26/2017 10:48:09 AM	None
Experiment created on 2017/...	agogomei	Finished	11/25/2017 6:06:23 PM	None
Experiment created on 2017/...	agogomei	Finished	11/24/2017 9:27:22 PM	None
4input_titanic [Predictive Exp.]	sinue625	Draft	11/21/2017 6:17:39 PM	None
SinputExperiment [Predictive ...	agogomei	Finished	11/21/2017 10:14:13 AM	None
4input_titanic	agogomei	Finished	11/21/2017 10:05:42 AM	None
CheckhereAzureMLPredictMo...	agogomei	Finished	10/27/2017 8:43:49 AM	None

DELETE **ADD TO PROJECT**

```

graph TD
    A[usedcar1209 - Copy.csv] --> B[Select Columns in Dataset]
    B --> C[Linear Regression]
    C --> D[Split Data]
    D --> E[Train Model]
    E --> F[Score Model]
    F --> G[Evaluate Model]
  
```

Create an Experiment

Microsoft Azure Machine Learning Studio

agogomei-Free-Workspace ? ☺ 🚧

PROJECTS experiments MY EXPERIMENTS SAMPLES EXPERIMENTS NEW

2 Blank Experiment Microsoft Samples

VIEW MORE IN GALLERY

Blank Experiment

Experiment Tutorial

Sample 1: Download dataset from UCI: Adult 2 class dataset

Sample 2: Dataset Processing and Analysis: Auto Imports Regression

Sample 3: Cross Validation for Binary Classification: Adult

Sample 4: Cross Validation for Regression: Auto Imports

Sample 5: Train, Test, Evaluate for Binary Classification: Adult

Sample 6: Train, Test, Evaluate for Regression: Auto Imports Dataset

Sample 7: Train, Test, Evaluate for Multiclass Classification: Letter

Sample 8: Apply SQL transformation

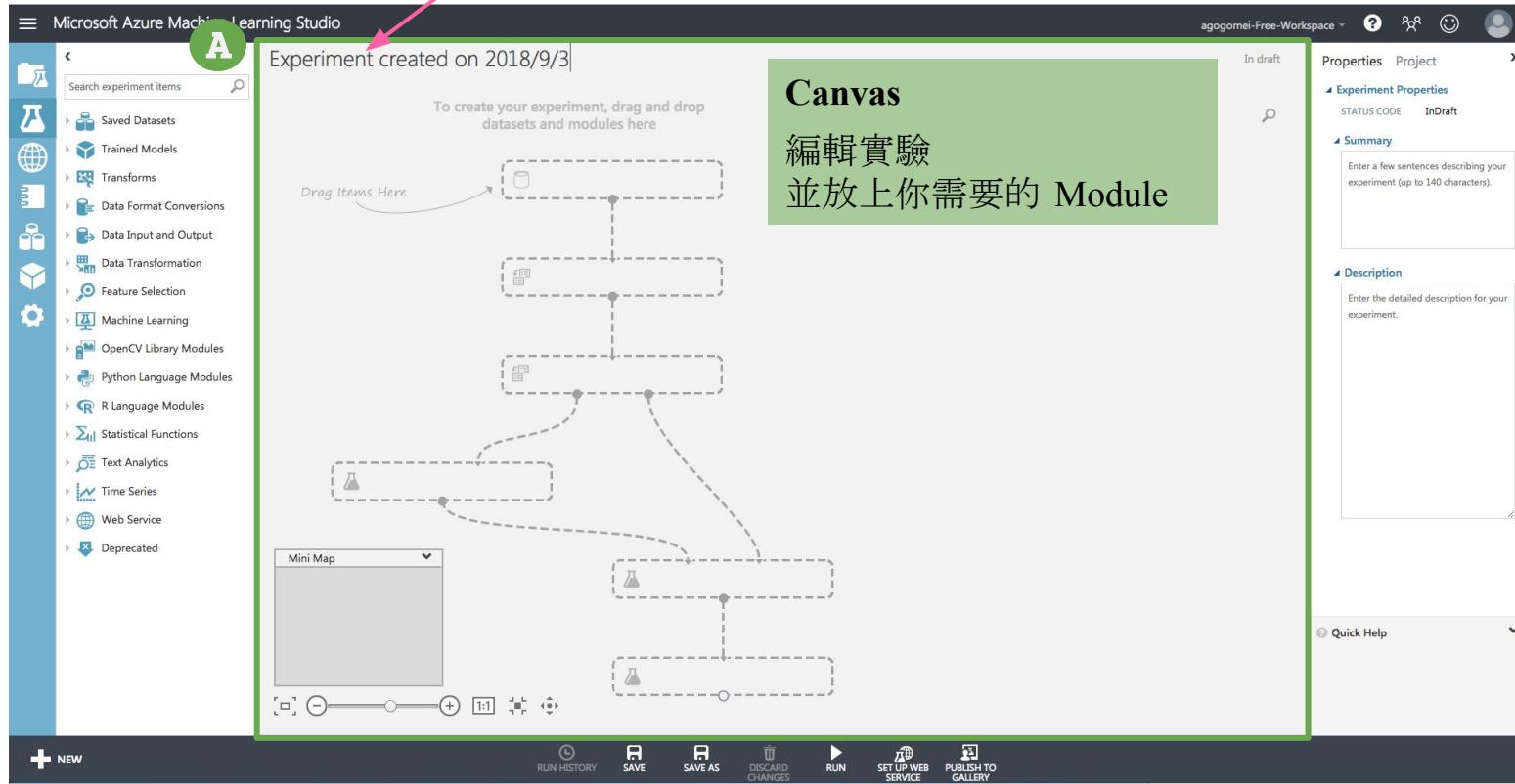
Sample 9: Split, partition and sample system

Anomaly Detection: Credit Risk

Binary Classification: Binary Classification: Binary Classification: Binary Classification: Binary Classification: Binary Classification:

介面介紹

可以改標題名稱



介面介紹

B Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

Module

To create your experiment, drag and drop datasets and modules here

執行各種操作

- 可從上方搜尋欄找 Module
- 可直接拖曳要使用到 Module 到 Canvas 中

In draft

Properties Project

Experiment Properties

Status Code InDraft

Summary

Description

Quick Help

NEW

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

介面介紹

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left is a sidebar with various icons for experiment items like Saved Datasets, Trained Models, Transforms, etc. The main area is titled "Experiment created on 2018/9/3" and contains a canvas where data flows between modules. A tooltip on the canvas explains: "當點選 Canvas 上的 Module 時, Properties 會出現對應的參數供調整 [Note] Quick Help 可以幫助你更深入了解個參數意義". To the right is a "Properties" panel with sections for Project (Experiment Properties, Summary, Description), and a "Quick Help" section.

Experiment created on 2018/9/3

To create your experiment, drag and drop datasets and modules here

當點選 Canvas 上的 Module 時,
Properties 會出現對應的參數供調整
[Note] Quick Help
可以幫助你更深入了解個參數意義

Properties

Project

In draft

Experiment Properties

Status Code InDraft

Summary

Description

Quick Help

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

介面介紹

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with icons for Saved Datasets, Trained Models, Transforms, Data Format Conversions, Data Input and Output, Data Transformation, Feature Selection, Machine Learning, OpenCV Library Modules, Python Language Modules, R Language Modules, Statistical Functions, Text Analytics, Time Series, Web Service, and Deprecated items. A search bar at the top says "Experiment created on 2018/9/3". The main area is titled "To create your experiment, drag and drop datasets and modules here" and features a "Drag Items Here" placeholder. A green callout box labeled "D" points to the "RUN HISTORY" button in the bottom navigation bar. Another green callout box labeled "E" points to the "SAVE AS" button. A large green box in the center contains the text "控制執行 儲存、執行、發布實驗..." (Control Execution, Save, Run, Publish Experiment...). The bottom navigation bar includes buttons for RUN HISTORY, SAVE, SAVE AS, DISCARD CHANGES, RUN, SET UP WEB SERVICE, and PUBLISH TO GALLERY.

Access Data

Titanic 資料

Variavle Name	Description
Survived	Survived (1) or died (0)
Pclass	Passenger's class
Name	Passenger's name
Sex	Passenger's sex
Age	Passenger's age
SibSp	Number of siblings/spouses aboard
Parch	Number of parents/children aboard
Ticket	Ticket number
Fare	Fare
Cabin	Cabin
Embark	Port of embarkation

Access Data

Microsoft Azure Machine Learning Studio
Experiment created on 2018/9/3
In draft Properties Project Import Data

匯入資料(當作 Training / Testing)
→ 從 Module 選擇, Data Input and Output 中的 [Import Data]

The screenshot shows the Azure Machine Learning Studio interface. On the left, a sidebar lists various modules under 'Data Input and Output', with 'Import Data' highlighted with a blue border. A large blue arrow points from this highlighted item towards the center workspace. In the center workspace, there is a single 'Import Data' module icon, which is also highlighted with a blue rounded rectangle. The right side of the screen displays configuration options for the 'Import Data' module, including fields for 'Account key', 'Path to container, director...', 'Blob file format' set to 'CSV', and checkboxes for 'File has header row' and 'Use cached results'. At the bottom, there are standard studio navigation buttons: RUN HISTORY, SAVE, SAVE AS, DISCARD CHANGES, RUN, SET UP WEB SERVICE, and PUBLISH TO GALLERY.

Access Data

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with various icons and a list of experiment items. The main area is titled "Experiment created on 2018/9/3". In the center, there's a "Import Data" step, which is highlighted with a blue rounded rectangle and has a small number "1" next to it. A large blue arrow points from the text below towards this step. On the right side, there's a properties panel for the "Import Data" step, showing settings like "Data source: Azure Blob Storage", "Authentication type: Storage Account", and "Account name", "Account key", "Path to container, director...". At the bottom, there's a toolbar with buttons for NEW, RUN HISTORY, SAVE, SAVE AS, DISCARD CHANGES, RUN, SET UP WEB SERVICE, and PUBLISH TO GALLERY.

Properties 設定

1. Data Source 選擇 : [Web URL via Http]
2. Data Source URL 輸入
3. CSV or TSV has header : 若資料含有 Header 就要打勾
4. Use cached results : 打勾表示把資料 cached 起來就不用每次執行實驗都重抓資料

Access Data

https://raw.githubusercontent.com/rладистайpei/Azureml-shiny-app/master/Shiny_Titanic/Titanic_train.csv

The screenshot illustrates the steps to import data into Azure Machine Learning Studio:

- Step 1:** In the main workspace, click "Import Data" (highlighted by a pink circle).
- Step 2:** Select "Web URL via HTTP" (highlighted by a pink circle).
- Step 3:** Enter the URL https://raw.githubusercontent.com/rладистайpei/Azureml-shiny-app/master/Shiny_Titanic/Titanic_train.csv, choose "CSV" as the data format, and check the box "CSV or TSV has header row" (highlighted by a pink circle). A note "資料有 Header 就要打勾" (Check if the data has a header) is overlaid on this step.

The "Properties" panel on the right shows the configuration details:

- Data source: Web URL via HTTP
- Data source URL: https://raw.githubusercontent.com/rладистайpei/Azureml-shiny-app/master/Shiny_Titanic/Titanic_train.csv
- Data format: CSV
- File has header row: checked
- Use cached results: checked

Access Data

資料設定完成後，檢查資料輸入狀況

1. [Run]
2. Results dataset 選擇 [Visualize]

The screenshot shows the Azure Machine Learning Studio interface. On the left, there's a sidebar with various modules like Data Format Conversions, Data Input and Output (selected), Feature Selection, Machine Learning, etc. In the center, there's a flowchart with a single step labeled 'Import Data'. Below the flowchart, a 'Mini Map' shows the step's position. At the bottom, there's a toolbar with buttons for RUN HISTORY, SAVE, SAVE AS, DISCARD CHANGES, RUN (highlighted with a pink circle and the number 1), SET UP WEB SERVICE, and PUBLISH TO GALLERY. To the right, there's a 'Quick Help' panel for the 'Import Data' step.

Import Data

1

Run

Run selected

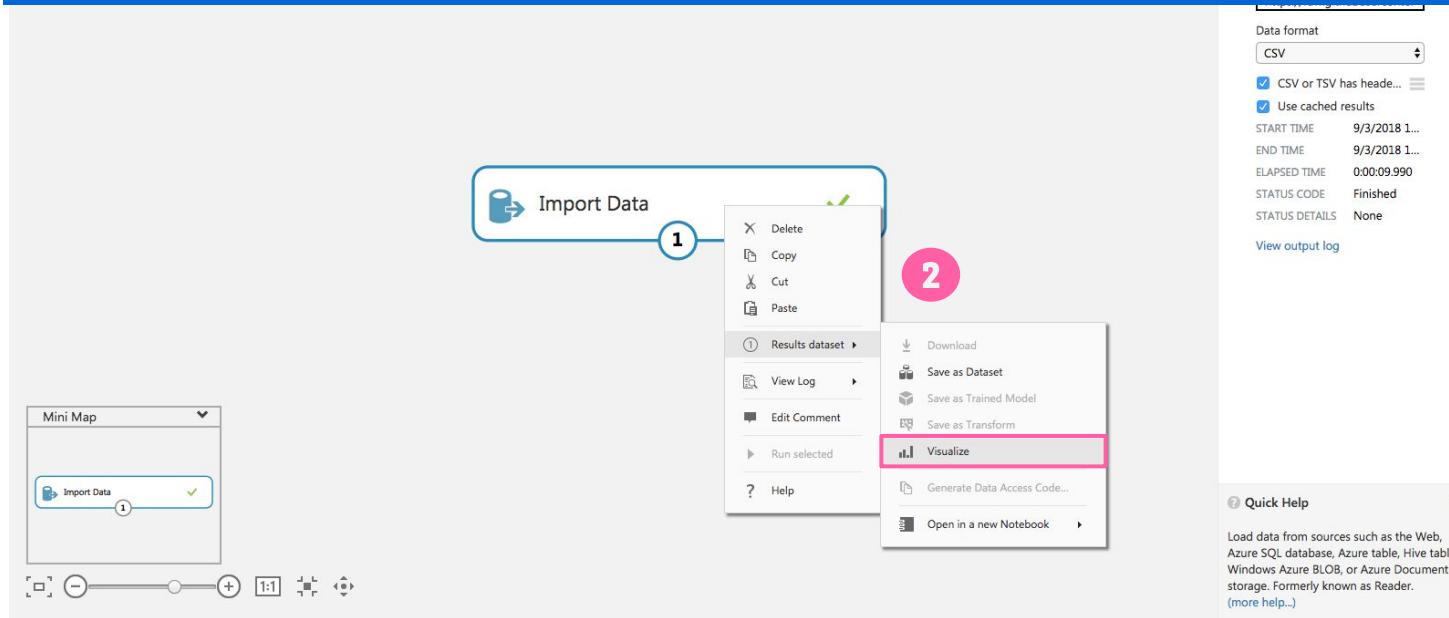
Quick Help

Load data from sources such as the Web, Azure SQL database, Azure table, Hive table, Windows Azure BLOB, or Azure DocumentDB storage. Formerly known as Reader. (more help...)

Access Data

資料設定完成後，檢查資料輸入狀況

1. [Run]
2. Import Data 右鍵 → Results dataset 選擇 [Visualize]



Access Data

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

experiment created on 2018/9/3 > Import Data > Results dataset

rows 891 columns 8

Survived PassengerClass Gender Age SiblingSpouse ParentChild FarePrice PortEmbarkation

Note 總共有 891 筆資料

檢查 Import 進來的資料有沒有問題

To view, select a column in the table.

RUN HISTORY SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

storage. Formerly known as Reader.
(more help...)

Data Select

Microsoft Azure Machine Learning Studio
Experiment created on 2018/9/3
In draft Properties Project

篩選欄位

→ 從 Module 選擇, Data Transformation 中 Manipulation 的 [Select Columns in Dataset]

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with various icons and a search bar. The 'Data Transformation' section is open, and under 'Manipulation', the 'Select Columns in Dataset...' option is highlighted with a blue border. A large blue arrow points from this highlighted option to the same module in the main workspace. The workspace itself shows a flow starting with 'Import Data' followed by 'Select Columns in Dataset'. A 'Quick Help' panel is open on the right, providing details about the 'Select Columns in Dataset' module.

Import Data

Select Columns in Dataset

Mini Map

Select Columns in Dataset

Import Data

Select Columns in Dataset

Quick Help

Selects columns to include or exclude from a dataset in an operation. Formerly known as Project Columns.
(more help...)

Run History Save Save As Discard Changes Run Set Up Web Service Publish to Gallery

Krisjen Chan 70

Data Select

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

In draft

Draft saved at 午後12:48:03

Properties Project

Select Columns in Dataset

Selected columns:
Launch the selector tool to make a selection

Launch column selector

Import Data

Select Columns in Dataset

按著連接點拖到連接處

Note 因為還沒設定 Properties

Mini Map

Import Data

Select Columns in Dataset

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Krispen Chan 71

```
graph TD; ImportData[Import Data] --> SelectColumns[Select Columns in Dataset];
```

Data Select

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

In draft Draft saved at: 下午12:48:03

Properties Project

1 Selected columns: Launch the selector tool to make a selection

2 Available columns: PassengerClass, Gender, Age, SiblingSpouse, ParentChild, FarePrice, PortEmbarkation

3 Selected column: Survived

要的選過來

選完記得打勾

Mini Map

Import Data

Select Columns in Dataset

RUN HISTORY SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

NEW

KiragenChian 72

Data Clean

Microsoft Azure Machine Learning Studio
Experiment created on 2018/9/3
In draft Properties Project

Search experiment items

Saved Datasets
Trained Models
Transforms
Data Format Conversions
Data Input and Output

Data Transformation

- Filter
- Learning with Counts
- Manipulation**

- Add Columns
- Add Rows
- Apply SQL Transformati...
- Clean Missing Data**
- Convert to Indicator Val...
- Edit Metadata
- Group Categorical Values
- Join Data
- Remove Duplicate Rows
- Select Columns in Datas...
- Select Columns Transfor...
- SMOTE

- Sample and Split
- Scale and Reduce
- Feature Selection

篩選欄位
→ 從 Module 選擇, Data Transformation 中 Manipulation 的 [Clean Missing Data]

```
graph TD; Import[Import Data] --> Select[Select Columns in Dataset]; Select --> Clean[Clean Missing Data];
```

1
Cleaning mode
Custom substitution value
Replacement value
0
Generate missing val...
Quick Help
Specifies how to handle the values missing from a dataset
(more help...)

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Krisjen Chan 73

Data Clean

Microsoft Azure Machine Learning Studio

agogomei-Free-Workspace ➔



Search experiment items

- Saved Datasets
- Trained Models
- Transforms
- Data Format Conversions
- Data Input and Output
- Data Transformation**
 - Filter
 - Learning with Counts
- Manipulation**
 - Add Columns
 - Add Rows
 - Apply SQL Transformati...
 - Clean Missing Data
 - Convert to Indicator Val...
 - Edit Metadata
 - Group Categorical Values
 - Join Data
 - Remove Duplicate Rows
 - Select Columns in Datas...
 - Select Columns Transfor...
 - SMOTE
- Sample and Split
- Scale and Reduce
- Feature Selection

Experiment created on 2018/9/3

In draft

Properties Project

Clean Missing Data

Columns to be cleaned

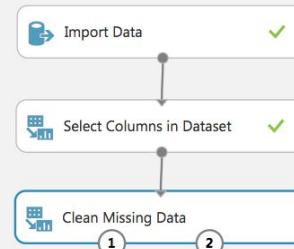
Selected columns:
All columns

Launch column selector

Minimum missing value ra...

0

- Replace using MICE
- Custom substitution value
- Replace with mean
- Replace with median
- Replace with mode
- Remove entire row** (selected)
- Remove entire column
- Replace using Probabilistic PCA



處理 Missing 方法

在 Cleaning mode 中選
[Remove entire row] : 當其中一個欄位出現
 遺失值時，整筆(row)刪掉

Quick Help

Specifies how to handle the values missing from a dataset
[\(more help...\)](#)



NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

Data Clean

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

712 Note 剩下有 712 筆資料

Gender Age SiblingSpouse ParentChild FarePrice PortEmbarkation

0 3 male 22 1 0 7.25 S

Statistics Visualizations

檢查資料處理狀況

1. Clean Missing Data 右鍵 → Cleaned dataset 選擇 [Visualize]

Apply	1	3	female	27	0	2	11.1333	S
Clean	1	2	female	14	1	0	30.0708	C
Conv	1	3	female	4	1	1	16.7	S
Edit N	1	1	female	58	0	0	26.55	S
Group	0	3	male	20	0	0	8.05	S
Join D	0	3	male	39	1	5	31.275	S
Remo	0	3	female	14	0	0	7.8542	S
Select	1	2	female	55	0	0	16	S
Select	0	3	male	2	4	1	29.125	Q
Select	0	3	female	31	1	0	18	S
SMO	0	2	male	35	0	0	26	S
Sampling	1	2	male	34	0	0	13	S
Scale and Reduce	1	2	male	15	1	1	0.0000	Q

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

KristenChen 75

Split Data

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft Draft saved at 下午4:55:29

Properties Project

Split Data

Splitting mode: Split Rows

Fraction of rows in the first set: 0.5

Randomized split

Random seed: 0

Stratified split: False

```

graph TD
    A[Import Data] --> B[Select Columns in Dataset]
    B --> C[Clean Missing Data]
    C --> D[Split Data]
    D -- 1 --> E
    D -- 2 --> F
  
```

切割 Training / Teating

→ 從 Module 選擇, Data Transformation 中 Sample and Split 的 [Split Data]

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

KristenChen 76

Split Data

Properties Project >

Split Data

Splitting mode: Split Rows

Fraction of rows in the first dataset: 0.7

Randomized split

Random seed: 1

Stratified split: True

Stratification key column: Selected columns: Column names: Survived

Launch column selector

1

選擇 Split Rows 的模式, 依照給定的比例切割資料

2

Training Data:70% Testing Data:30%

3

Random Seed : 為了讓每次執行切的 Dataset 一致

4

Stratified Split : 指對某個 Column 的值平均分配

Note

這裡使用的原因是避免 Training Data 中的 Survived 都是 1 或 0 , 盡量使其平均分配在 Training / Testing 中

Model

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Properties Project

Two-Class Decision Forest

Resampling method: Bagging

Create trainer mode: Single Parameter

Number of decision trees: 8

Maximum depth of the de...: 32

Number of random splits ...: 128

Minimum number of sam...: 1

Allow unknown values...

建立 Model
→ 從 Module 選擇, Machine Learning 中 Initialize Model 的 [Classification]
→ [Two-Class Decision Forest]

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

KristenChen 78

Model

Microsoft Azure Machine Learning Studio agogomei-Free-Workspace

建立 Model
→ 從 Module 選擇, Machine Learning 中 Train 的 [Train Model]

Column names: Survived

Launch column selector

Properties Project

Train Model

Label column

Selected columns:
Column names: Survived

Launch column selector

1

2

Select a single column

BY NAME WITH RULES

AVAILABLE COLUMNS

All Types | search columns

PassengerClass
FarePrice
ParentChild
Age
SibSpSpouse

SELECTED COLUMNS

All Types | search columns

Survived

選 Target 變數

1 columns selected

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

KristenChen 79

Score Model

Microsoft Azure Machine Learning Studio

建立 Model
→ 從 Module 選擇, Machine Learning 中 Score 的 [Score Model]

```
graph TD; Import[Import Data] --> Select[Select Columns in Dataset]; Select --> Clean[Clean Missing Data]; Clean --> FeatureSelection[Filter Based Feature Selection]; FeatureSelection --> DecisionJungle[Two-Class Decision Jungle]; DecisionJungle --> Split[Split Data]; Split --> Train[Train Model]; Train --> ScoreModel[Score Model];
```

Quick Help

Score a trained classification or regression model
(more help...)

NEW

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Krispen Chan 80

Evaluate Model

Microsoft Azure Machine Learning Studio

建立 Model
→ 從 Module 選擇, Machine Learning 中 Evaluate 的 [Evaluate Model]

END TIME: 11/30/2018...
STATUS CODE: InDraft
STATUS DETAILS: None
Prior Run

Summary
Enter a few sentences describing your experiment (up to 140 characters).

Description
Enter the detailed description for your experiment.

Quick Help

```
graph TD; Import[Import Data] --> Select[Select Columns in Dataset]; Select --> Clean[Clean Missing Data]; Clean --> TwoClass[Two-Class Decision Forest]; Clean --> Split[Split Data]; TwoClass --> Train[Train Model]; Split --> Train; Train --> Score[Score Model]; Score --> Evaluate[Evaluate Model];
```

NEW RUN HISTORY SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

KristenChen 81

Select Features

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

In draft

Draft saved at 下午 2:05:50

Properties Project

Experiment Properties

- START TIME 9/3/2018 2:05:50
- END TIME 9/3/2018 2:05:50
- STATUS CODE InDraft
- STATUS DETAILS None

Prior Run

Summary

Enter a few sentences describing your experiment (up to 140 characters).

Description

Enter the detailed description for your experiment.

Import Data → Select Columns in Dataset → Clean Missing Data → Filter Based Feature Selection

選擇 Y 較有影響力欄位
→ 從 Module 選擇, Feature Selection 中的 [Filter Based Feature Selection]

Note 可以使 Training Data 更適合一點

Quick Help

NEW

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

```
graph TD; A[Import Data] --> B[Select Columns in Dataset]; B --> C[Clean Missing Data]; C --> D[Filter Based Feature Selection];
```

Select Features

Microsoft Azure Machine Learning Studio
Experiment created on 2018/9/3
agogomei-Free-Workspace Properties Project
Draft saved at 11:42 AM

Properties Project
Filter Based Feature Selection
Feature scoring method
 Pearson Correlation
Mutual Information
Kendall Correlation
Spearman Correlation
Chi Squared
Fisher Score
Count Based
Launch column selector
Number of desired features
5
選擇選取 Feature 的方法
Text Analytics
Time Series
Web Service
Deprecated

1

2

3

選擇 Target Column

Import Data

Select a single column

BY NAME WITH RULES

AVAILABLE COLUMNS
All Types search columns
PassengerClass
Gender
Age
SiblingSpouse
ParentChild
FarePrice
PortEmbarkation

SELECTED COLUMNS
All Types search columns
Survived

7 columns available
1 columns selected

Launch column selector
Selected columns:
Column names: Survived
Number of desired features
5

欲挑選 Column 數

RUN HISTORY SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Kristen Chan 83

Select Features

看哪些對 Y 較有影響力欄位
→ Filter Based Feature Selection 右鍵 → Feature 選擇 [Visualize]

Microsoft Azure Machine Learning Studio

agogomei-Free-Workspace

Search experiment items

Saved Datasets

Trained Models

Transforms

Data Format Conversions

Data Input and Output

Data Transformation

Feature Selection

- Filter Based Feature Select...
- Fisher Linear Discriminant ...
- Permutation Feature Import...

Machine Learning

OpenCV Library Modules

Python Language Modules

R Language Modules

Statistical Functions

Text Analytics

Time Series

Web Service

Deprecated

+

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

NEW

Target column: Survived

Selected columns: Column names: Survived

Launch column selector

Number of desired features: 5

START TIME: 9/3/2018 2:15:00 PM

END TIME: 9/3/2018 2:15:00 PM

ELAPSED TIME: 0:00:02.629

STATUS CODE: Finished

STATUS DETAILS: None

View output log

Quick Help

Identifies the features in a dataset with the greatest predictive power (more help...)

Chia 84

Select Features

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

Experiment created on 2018/9/3 > Filter Based Feature Selection > Features

rows: 1 columns: 8

	Survived	PassengerClass	FarePrice	ParentChild	Age	SiblingSpouse	Gender	PortEmbarkation
view as	1	0.356462	0.2661	0.095265	0.082446	0.015523	0	0

Filter Based Feature Selection

Statistics

Visualizations

To view, select a column in the table.

+

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

agomei-Free-Workspace

Properties Project

Krisjen Chan 85

Split Data

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

In draft Draft saved at 午后2:26:38

Properties Project

Split Data

Splitting mode: Split Rows

Fraction of rows in the first set: 0.5

Randomized split: checked

Random seed: 0

Stratified split: False

```
graph TD; Import[Import Data] --> Select[Select Columns in Dataset]; Select --> Clean[Clean Missing Data]; Clean --> Filter[Filter Based Feature Selection]; Filter --> Split[Split Data]; Split -- 1 --> Out1; Split -- 2 --> Out2;
```

切割 Training / Teating
→ 從 Module 選擇, Data Transformation 中 Sample and Split 的 [Split Data]

Split the rows of a dataset into two distinct sets
(more help..)

NEW RUN HISTORY SAVE DISCARD CHANGES SET UP WEB SERVICE PUBLISH TO GALLERY

Kristen Chan 86

Split Data

Properties Project >

Split Data

Splitting mode: Split Rows

Fraction of rows in the first dataset: 0.7

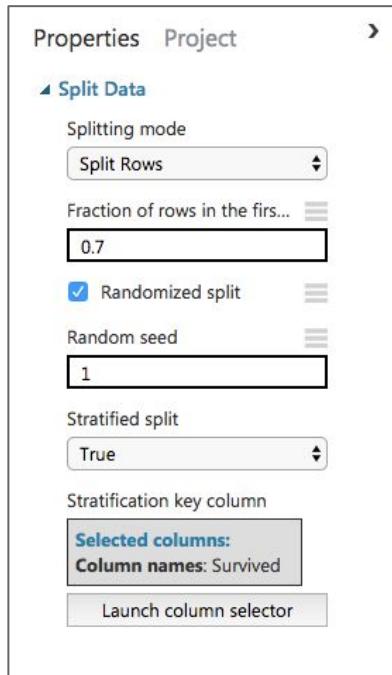
Randomized split

Random seed: 1

Stratified split: True

Stratification key column: Selected columns: Column names: Survived

Launch column selector



1

選擇 Split Rows 的模式, 依照給定的比例切割資料

2

Training Data:70% Testing Data:30%

3

Random Seed : 為了讓每次執行切的 Dataset 一致

4

Stratified Split : 指對某個 Column 的值平均分配

Note

這裡使用的原因是避免 Training Data 中的 Survived 都是 1 或 0 , 盡量使其平均分配在 Training / Testing 中

Model

Microsoft Azure Machine Learning Studio

Experiment created on 2018/9/3

In draft

Properties Project

Two-Class Decision Jungle

- Resampling method: Bagging
- Create trainer mode: Single Parameter
- Number of decision DAGs: 8
- Maximum depth of the de...: 32
- Maximum width of the de...: 128
- Number of optimization st...: 2048
- Allow unknown values...

Import Data → Select Columns in Dataset → Clean Missing Data → Filter Based Feature Selection → Split Data

Two-Class Decision Jungle

建立 Model
→ 從 Module 選擇, Machine Learning 中 Initialize Model 的 [Classification]
→ [Two-Class Decision Forest]

NEW

RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Kristen Chan 88

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with various icons for Saved Datasets, Trained Models, Transforms, Data Format Conversions, Data Input and Output, Data Transformation, Feature Selection, and Machine Learning. Under Machine Learning, there are sub-options for Evaluate, Initialize Model (with Anomaly Detection and Classification), and several Multiclass and Multicategory models like Multiclass Decision Forest, Multiclass Decision Tree, Multiclass Logistic Regression, etc. The main workspace displays a workflow starting with 'Import Data', followed by 'Select Columns in Dataset', 'Clean Missing Data', 'Filter Based Feature Selection', and finally 'Split Data'. A module named 'Two-Class Decision Jungle' is highlighted with a blue border and a circled '1' below it. The 'Properties' pane on the right is open for 'Two-Class Decision Jungle', showing settings for resampling (Bagging), trainer mode (Single Parameter), number of decision DAGs (8), maximum depth (32), maximum width (128), number of optimization steps (2048), and allowing unknown values. The status bar at the bottom shows navigation options: RUN HISTORY, SAVE, SAVE AS, DISCARD CHANGES, RUN, SET UP WEB SERVICE, and PUBLISH TO GALLERY.

Model

Microsoft Azure Machine Learning Studio
Experiment created on 2018/9/3
agogomei-Free-Workspace ↗ Properties Project

建立 Model → 從 Module 選擇, Machine Learning 中 Train 的 [Train Model]

The screenshot shows a machine learning experiment in Azure ML Studio. The workflow consists of the following steps:

- Input: Two-Class Decision Jungle → Split Data
- Split Data → Select Columns in Dataset
- Select Columns in Dataset → Clean Missing Data
- Clean Missing Data → Filter Based Feature Selection
- Filter Based Feature Selection → Train Model

A pink circle labeled "1" highlights the "Train Model" step. A pink arrow labeled "2" points from the "Properties" panel to the "Selected columns" section of the "Train Model" properties.

Properties Panel (Train Model):

- Label column: Selected columns: Column names: Survived
- Launch column selector

Column Selector Dialog:

- Available columns: PassengerClass, FarePrice, ParentChild, Age, SiblingSpouse
- Selected columns: Survived

Description:

Enter a few sentences describing your experiment (up to 140 characters).

Summary:

Enter the detailed description for your experiment.

Bottom Navigation:

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Kristen Chan 89

Score Model

Microsoft Azure Machine Learning Studio

建立 Model
→ 從 Module 選擇, Machine Learning 中 Score 的 [Score Model]

```
graph TD; Import[Import Data] --> Select[Select Columns in Dataset]; Select --> Clean[Clean Missing Data]; Clean --> FeatureSelection[Filter Based Feature Selection]; FeatureSelection --> DecisionJungle[Two-Class Decision Jungle]; DecisionJungle --> Split[Split Data]; Split --> Train[Train Model]; Train --> ScoreModel[Score Model];
```

Quick Help
Score a trained classification or regression model
(more help...)

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Evaluate Model

Microsoft Azure Machine Learning Studio

建立 Model
→ 從 Module 選擇, Machine Learning 中 Evaluate 的 [Evaluate Model]

```
graph TD; A[Import Data] --> B[Select Columns in Dataset]; B --> C[Clean Missing Data]; C --> D[Filter Based Feature Selection]; D --> E[Two-Class Decision Jungle]; E --> F[Split Data]; F --> G[Train Model]; G --> H[Score Model]; H --> I[Evaluate Model]
```

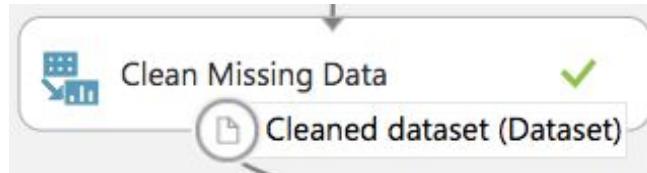
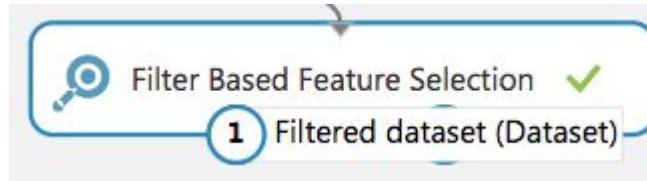
Quick Help

Evaluates a scored classification or regression model with standard metrics
(more help...)

NEW RUN HISTORY SAVE SAVE AS DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Kristen Chan 91

Note 為什麼有兩個節點



Note

滑鼠移到連接點上可以查詢

Azure Machine Learning Studio

Microsoft Azure Machine Learning Studio

Titanic

Finished running ✓ Draft saved at 午後3:08:01

Properties Project

Experiment Properties

- START TIME 9/3/2018 3:...
- END TIME 9/3/2018 3:...
- STATUS CODE Finished
- STATUS DETAILS None

Prior Run

Summary

Enter a few sentences describing your experiment (up to 140 characters).

Description

Enter the detailed description for your experiment.

Import Data → Select Columns in Dataset → Clean Missing Data → Filter Based Feature Selection → Two-Class Decision Jungle → Split Data → Train Model → Score Model → Evaluate Model

檢查模型狀況

Evaluate Model 右鍵 Evaluation results 選擇 [Visualize]

Blood donation data Book Reviews from Am... NEW RUN HISTORY SAVE DISCARD CHANGES SET UP WEB SERVICE PUBLISH TO GALLERY

Azure Machine Learning Studio

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

ROC PRECISION/RECALL LIFT

True Positive Rate

False Positive Rate

Scored dataset

Score Model

Evaluate Model

Airport Codes Dataset

Run History

Save

Discard Changes

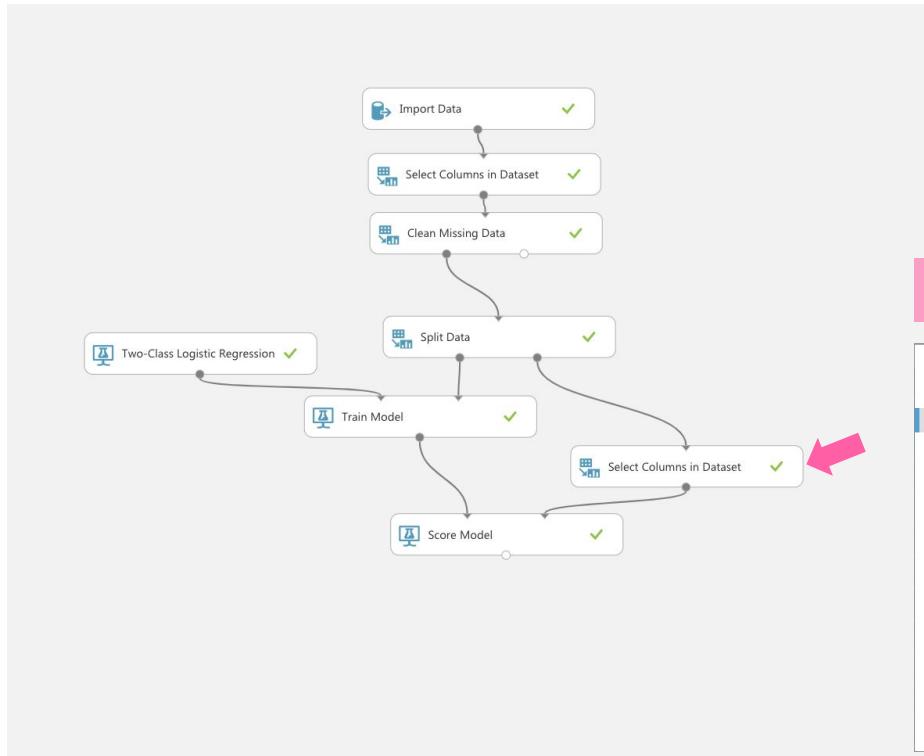
Run

Set Up Web Service

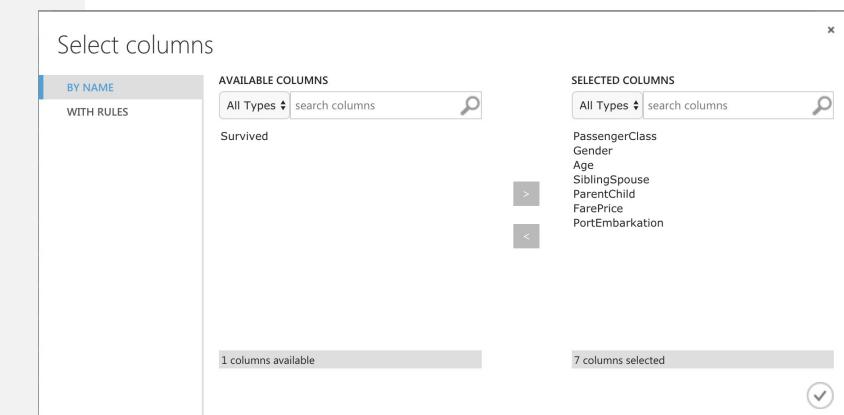
Publish To Gallery

The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there's a sidebar with various icons for data management and model creation. The main area displays an ROC curve titled 'ROC PRECISION/RECALL LIFT' with 'True Positive Rate' on the y-axis and 'False Positive Rate' on the x-axis. A legend indicates that the blue line represents the 'Scored dataset'. Below the plot, it says 'Experiment created on 2018/11/30 > Evaluate Model > Evaluation results'. To the right, a workflow diagram shows a sequence of steps: 'Score Model' followed by 'Evaluate Model'. Both steps are marked with green checkmarks. At the bottom of the workflow, there are connection points and a 1:1 ratio indicator. The status bar at the bottom shows '1 E'.

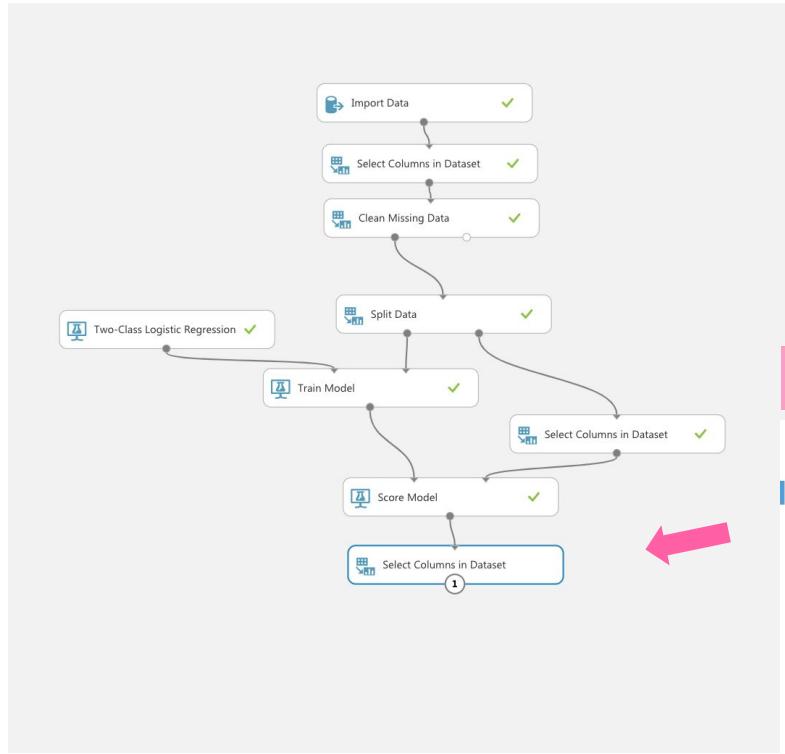
Azure Machine Learning Studio



只選擇 [解釋變數]



Azure Machine Learning Studio



選擇 [Predict變數]: Scored Labels

Select columns

BY NAME

WITH RULES

AVAILABLE COLUMNS

All Types

PassengerClass
Gender
Age
SiblingSpouse
ParentChild
FarePrice
PortEmbarkation
Scored Probabilities

SELECTED COLUMNS

All Types

Scored Labels

8 columns available

1 columns selected



How to connect with Shiny -- Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Draft saved at 下午6:27:04

Properties Project

Select Columns in Dataset

Selected columns:
Column names: Scored Labels

Launch column selector

Quick Help

Selects columns to include or exclude from a dataset in an operation. Formerly known as Project Columns.
(more help...)

```
graph TD; Import[Import Data] --> Select1[Select Columns in Dataset]; Select1 --> Clean[Clean Missing Data]; Clean --> Split[Split Data]; Split --> Train[Train Model]; Split --> Score[Score Model]; Train --> Select2[Select Columns in Dataset]; Score --> Select2;
```

1

Predictive Web Service [Recommended]

Retraining Web Service

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

Kristen Chan 97

How to connect with Shiny -- Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Draft saved at 下午6:27:04

Properties Project

Web service input

STEP 1

INPUT & OUTPUT NODES

You now have the ability to add, change or remove inputs and outputs in your experiment

NEXT

Train Model

Select Columns in Dataset

Score Model

Select Columns in Dataset

Web service output

Web service output

Quick Help

Selects columns to include or exclude from a dataset in an operation. Formerly known as Project Columns.

(more help...)

+

-

1:1

Input

Output

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

NEW

KristenChen 98

```
graph TD; TrainModel[Train Model] --> ScoreModel[Score Model]; ScoreModel --> Output1[Web service output]; SelectC1[Select Columns in Dataset] --> ScoreModel; SelectC2[Select Columns in Dataset] --> Output2[Web service output]
```

How to connect with Shiny -- Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Draft saved at 下午6:29:45

Properties Project

◀ Select Columns in Dataset

Select columns

Selected columns:
Column names: Scored Labels

Launch column selector

Quick Help

Selects columns to include or exclude from a dataset in an operation. Formerly known as Project Columns.
(more help...)

```
graph TD; Import[Import Data] --> Select1[Select Columns in Dataset]; Import --> Clean[Clean Missing Data]; Select1 --> Split[Split Data]; Clean --> Split; Split --> Train[Train Model]; Train --> Score[Score Model]; Score --> Select2[Select Columns in Dataset]; Score --> Output1[Web service output]; Train --> Select2; Select2 --> Output2[Web service output];
```

+

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

1 E

KristenChen 99

How to connect with Shiny – Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Draft saved at 下午6:29:45

Properties Project

◀ Select Columns in Dataset

Select columns

Selected columns:
Column names: Scored Labels

Launch column selector

Quick Help

Selects columns to include or exclude from a dataset in an operation. Formerly known as Project Columns.
(more help...)

```
graph TD; Import[Import Data] --> SC1[Select Columns in Dataset]; SC1 --> Clean[Clean Missing Data]; Clean --> TLR[Two-Class Logistic Regression]; Clean --> Split[Split Data]; TLR --> Train[Train Model]; Split --> Train; Train --> Score[Score Model]; Score --> SC2[Select Columns in Dataset]; SC2 --> Output1[Web service output];
```

3

刪掉

1

NEW

RUN HISTORY

SAVE

SAVE AS

DISCARD CHANGES

RUN

SET UP WEB SERVICE

PUBLISH TO GALLERY

Kristen Chan 100

How to connect with Shiny – Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30 搬到 Score Model 前

4

In draft Draft saved at 下午6:31:52

Properties Project

Experiment Properties

- START TIME 11/30/2018...
- END TIME 11/30/2018...
- STATUS CODE InDraft
- STATUS DETAILS None

Prior Run

Summary

Enter a few sentences describing your experiment (up to 140 characters).

Description

Enter the detailed description for your experiment.

Quick Help

Web service output

```
graph TD; Import[Import Data] --> Select1[Select Columns in Dataset]; Import --> Clean[Clean Missing Data]; Select1 --> Reg[Two-Class Logistic Regression]; Clean --> Split[Split Data]; Split --> Train[Train Model]; Split --> Score[Score Model]; Train --> Select2[Select Columns in Dataset]; Score --> Select2; Select2 --> Output[Web service output]; Select2 --> Input[Web service input]
```

Input

Output

NEW RUN HISTORY SAVE AS DISCARD CHANGES SET UP WEB SERVICE PUBLISH TO GALLERY

How to connect with Shiny -- Setup Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

In draft

Draft saved at 下午6:35:31

Properties Project

Experiment Properties

- START TIME 11/30/2018...
- END TIME 11/30/2018...
- STATUS CODE InDraft
- STATUS DETAILS None

Prior Run

Summary

Description

Quick Help

Import Data → Select Columns in Dataset → Clean Missing Data → Two-Class Logistic Regression → Split Data → Train Model → Score Model → Select Columns in Dataset → Web service output

Web service input

Input Output

NEW

RUN HISTORY SAVE SAVE AS DISCARD CHANGES SET UP WEB SERVICE PUBLISH TO GALLERY

5

Kristen Chan 102

How to connect with Shiny – Deploy Web Service

Microsoft Azure Machine Learning Studio

Experiment created on 2018/11/30

Finished running ✓

Properties Project

Experiment Properties

- START TIME 11/30/2018...
- END TIME 11/30/2018...
- STATUS CODE Finished
- STATUS DETAILS None

Prior Run

Summary

Description

Quick Help

Import Data → Select Columns in Dataset → Clean Missing Data → Split Data → Two-Class Logistic Regression → Train Model → Score Model → Select Columns in Dataset → output

Input → - → + → 1:1 → ⚡ → Predictive Web Service [Recommended] → Deploy Web Service

Search experiment items

- Saved Datasets
- Trained Models
- Transforms
- Data Format Conversions
- Data Input and Output
- Data Transformation
- Feature Selection
- Machine Learning
- OpenCV Library Modules
- Python Language Modules
- R Language Modules
- Statistical Functions
- Text Analytics
- Time Series
- Web Service

Kristen Chan 103

Go to Azure ML Studio Web Service

Microsoft Azure Machine Learning Studio

titanic

DASHBOARD CONFIGURATION

New Web Services Experience [preview](#)

General Published experiment:

View snapshot View latest

Description

No description provided for this web service.

API key

Cf+IT8XsiGAFZD/ `CUoRZ2RF3A9d+Hlw==`

Default Endpoint

API HELP PAGE TEST LAST UPDATED

REQUEST/RESPONSE Test [Test preview](#) 9/3/2018 3:18:28 PM

BATCH EXECUTION [Test preview](#) 9/3/2018 3:18:28 PM

+

NEW

DELETE

agogomei-Free-Workspace ▾ ? 🔍 😊 🚫

Krisjen Chan 104

Go to Azure ML Studio Web Service

Microsoft Azure Machine Learning Studio

PROJECTS EXPERIMENTS WEB SERVICES NOTEBOOKS DATASETS TRAINED MODELS SETTINGS

web services

NAME	CREATED ON
<input type="checkbox"/> UsedCarWebService	11/21/2017 11:00:42 AM
<input type="checkbox"/> 5inputExperiment [Predictive Exp.]	11/21/2017 10:13:24 AM
<input type="checkbox"/> 4input_titanic [Predictive Exp.]	11/21/2017 10:12:45 AM
<input type="checkbox"/> 3inputtitanic	11/20/2017 7:54:46 PM
<input checked="" type="checkbox"/> TitanicWS	9/22/2017 9:01:51 PM

Note If You already have web services check here

Azure ML Studio Web Services

≡ Microsoft Azure Machine Learning Web Services

Quickstart

Dashboard

Batch Request Log

Configure

Consume

Test

Swagger API

← TitanicWS

default

View in Studio ↗

Web service consumption options



Excel 2013 or later



Excel 2010 or earlier



Request-Response
Web App Template

Basic consumption info

Want to see how to consume this information? [Check out this easy tutorial.](#)

Primary Key

/D04m7PMuV

rw==



Secondary Key

alcP/Tw3ZJEj

lg==



Azure ML Studio Web Services

≡ Microsoft Azure Machine Learning Web Services

Quickstart

Dashboard

Batch Request Log

Configure

Consume

Test

Swagger API

Sample Code

Request–Response

1

Select R Code

C# Python Python 3+

R

2

Copy



```
library("RCurl")
library("rjson")

# Accept SSL certificates issued by public Certificate Authorities
options(RCurlOptions = list(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl")))

h = basicTextGatherer()
hdr = basicHeaderGatherer()

req = list(
  Inputs = list(
    "input1" = list(
      list(
        'PassengerClass' = "3",
        'Gender' = "male",
        'Age' = "22",
        'SiblingSpouse' = "1",
        'ParentChild' = "0",
        'FarePrice' = "7.25",
        'PortEmbarkation' = "S"
      )
    )
  )
)
```

Open R Studio



RStudio

Untitled1

Source on Save

Addins

Run

Source

Console

Project: (None)

1

3 Paste the R Code here

Environment

History

Import Dataset

Global Environment

Environment is empty

R Script 2

Files Plots Packages Help Viewer

R: Split Data into Test and Train Set [Find in Topic]

sample.split (caTools)

R Documentation

Split Data into Test and Train Set

Description

Split data from vector Y into two sets in predefined ratio while preserving relative ratios of different labels in Y. Used to split the data used during classification into train and test subsets.

Usage

```
sample.split( Y, SplitRatio = 2/3, group = NULL )
```

Arguments

Y Vector of data labels. If there are only a few labels (as is expected) than relative ratio of data in both subsets will be the same.

SplitRatio Splitting ratio:

- if $(0 <= \text{SplitRatio} < 1)$ then **SplitRatio** fraction of points from Y will be set to TRUE;
- if $(\text{SplitRatio} == 1)$ then one random point from Y will be set to TRUE;
- if $(\text{SplitRatio} > 1)$ then **SplitRatio** number of points from Y will be set to TRUE;

group Optional vector/list used when multiple copies of each sample are present. In such a case group contains unique sample labels, marking all copies of the same sample with the same label, and the function tries to place all copies in either train or test subset. If provided then it has to have the same length as Y.

Kristen Chai 108

Azure ML API in R

```
1 library("RCurl")
2 library("rjson")
3
4 # Accept SSL certificates issued by public Certificate Authorities
5 options(RCurlOptions = list(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl")))
6
7 h = basicTextGatherer()
8 hdr = basicHeaderGatherer()
9
10 req = list(
11   Inputs = list(
12     "input1" = list(
13       list(
14         'PassengerClass' = "3",
15         'Gender' = "male",
16         'Age' = "22",
17         'SiblingSpouse' = "1",
18         'ParentChild' = "0",
19         'FarePrice' = "7.25",
20         'PortEmbarkation' = "S"
21       )
22     ),
23   GlobalParameters = setNames(fromJSON('{}'), character(0))
24 )
25
26
27 body = enc2utf8(toJSON(req))
28 api_key = "abc123" # Replace this with the API key for the web service
29 authz_hdr = paste('Bearer', api_key, sep=' ')
30
31 h$reset()
```

"PassengerClass" = "3",
"Gender" = "female",
"Age" = "22",
"SiblingSpouse" = "1",
"ParentChild" = "0",
"FarePrice" = "7.25",
"PortEmbarkation" = "S"

Azure ML Studio Web Services -- Get API Key

≡ Microsoft Azure Machine Learning Web Services

Quickstart

Dashboard

Batch Request Log

Configure

Consume

Test

Swagger API

← TitanicWS

default

View in Studio ↗

Web service consumption options



Excel 2013 or later



Excel 2010 or earlier



Request-Response
Web App Template

Basic consumption info

Want to see how to consume this information? [Check out this easy tutorial.](#)

Primary Key

/D04m7PMuV

Copy Your API Key

2W==



Secondary Key

alcP/Tw3ZJEj

lg==



Open R Studio



The screenshot shows the R Studio interface with the following details:

- Code Editor:** An R script titled "Untitled1.R" is open. The code uses the RCurl and rjson packages to make a POST request to an API. It includes setting SSL certificates, creating basic text and header gatherers, defining a request list with passenger details, and setting global parameters. A line of code to set the API key is highlighted with a blue box and a large number "4".

```
library("RCurl")
library("rjson")
# Accept SSL certificates issued by public Certificate Authorities
options(RCurlOptions = list(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl")))
h = basicTextGatherer()
hdr = basicHeaderGatherer()
req = list(
  Inputs = list(
    "input:" = list(
      list(
        'PassengerClass' = "3",
        'Gender' = "male",
        'Age' = "22",
        'SiblingSpouse' = "1",
        'ParentChild' = "0",
        'FarePrice' = "7.25",
        'PortEmbarkation' = "S"
      )
    )
  ),
  GlobalParameters = setNames(fromJSON('{}'), character(0))
)
body = enc2utf8 toJSON(req)
api_key = "abc123" # Replace this with the API key for the web service
authz_header = paste0(header, api_key, sep = ' ')
h$reset()
```
- Console:** The console window is empty.
- File Browser:** The sidebar shows a file tree with a .History file and various system folders.
- Environment:** The environment pane shows the Global Environment tab.
- Message Bar:** A blue bar at the bottom says "Modify Your API Key".
- Callout:** A blue arrow points from the "api_key" line in the code editor to the message bar.
- Text Overlay:** Below the message bar, the text "Like: api_key = "Your API Key"" is displayed.

Open R Studio



The screenshot shows an R Studio interface with the following components:

- Code Editor:** Contains R code for performing a curl request to an Azure ML endpoint. A blue circle labeled "5" is positioned next to the "Run all" button.
- Console:** Displays the output of the R code. A blue circle labeled "6" is positioned next to the result. The output includes:

```
> if (httpStatus >= 400)
+   print(paste("The request failed with status code:", httpStatus, sep=""))
+   # Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
+   print(headers)
+
> print("Result:")
[1] "Result:"
> result = h$value()
> print(fromJSON(result))
$Results
$Results$output2
$Results$output2[[1]]
$Results$output2[[1]]$PredictedSurvived
[1] "0"
```
- Environment:** Shows variables defined in the session, including `api_key`, `outhz_hdr`, `body`, `h`, `hdr`, `headers`, `httpStatus`, `req`, and `result`.
- Note:** A pink callout box contains the text "Just make sure your azure ml can run in R."
- Remember the result format:** A large blue arrow points from the "6" in the console to a black bar containing the text "\$Results\$output2[[1]]\$`Scored Labels`".

Titanic Survival Prediction

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish

Titanic Survival Prediction

Passenger Class :

1

Gender :

female

Port Embarkation :

C

Age :

80

Sibling Spouse :

8

Parent Child :

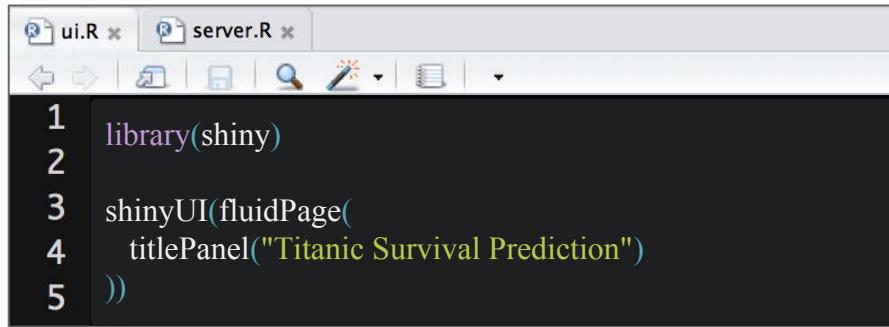
9

Fare Price :

0 51.23 802.466 204.932 307.398 409.864 512.3292

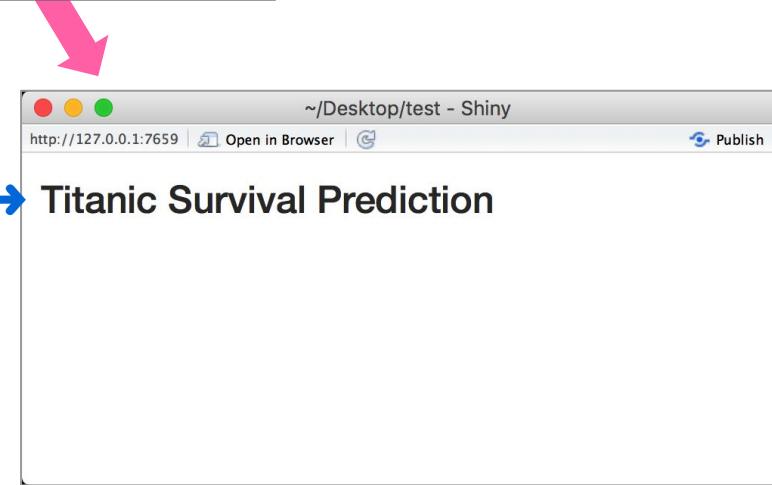


[Code] Titanic Survival Prediction -- title



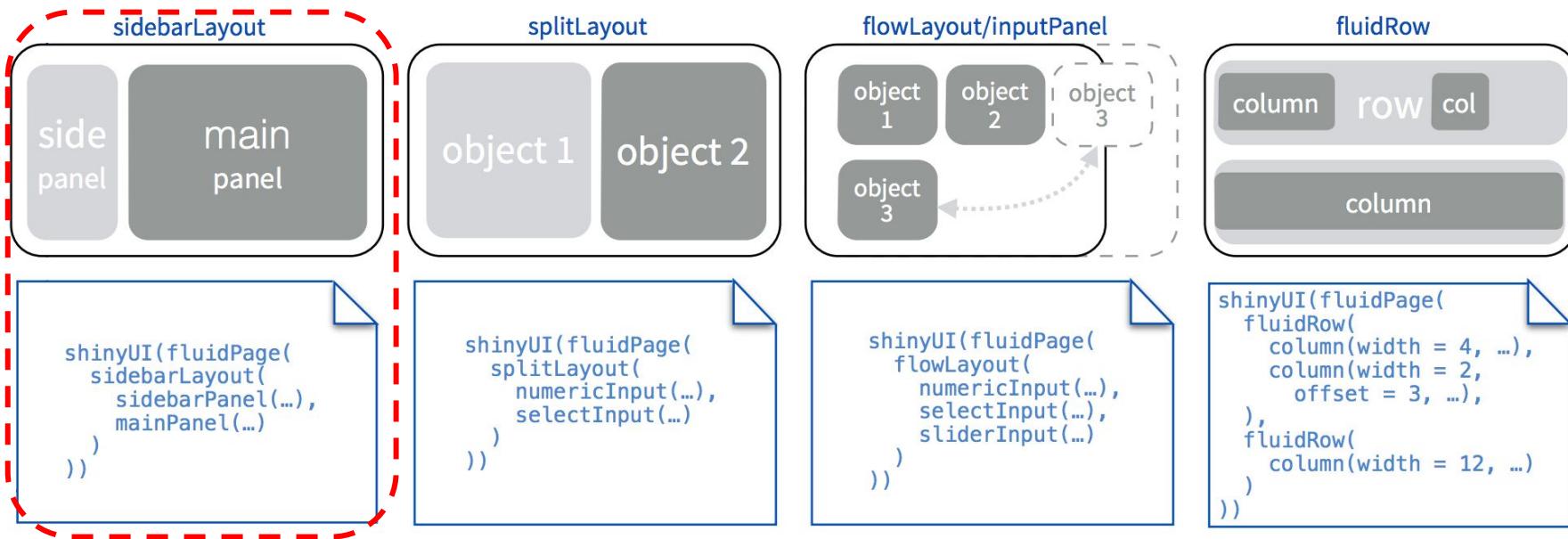
A screenshot of the RStudio interface showing the `ui.R` file. The code is as follows:

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction")
5 ))
```



`titlePanel` → **Titanic Survival Prediction**

Layout



[Layout] Titanic Survival Prediction

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish

Titanic Survival Prediction sidebarLayout

Passenger Class : 1

Gender : female

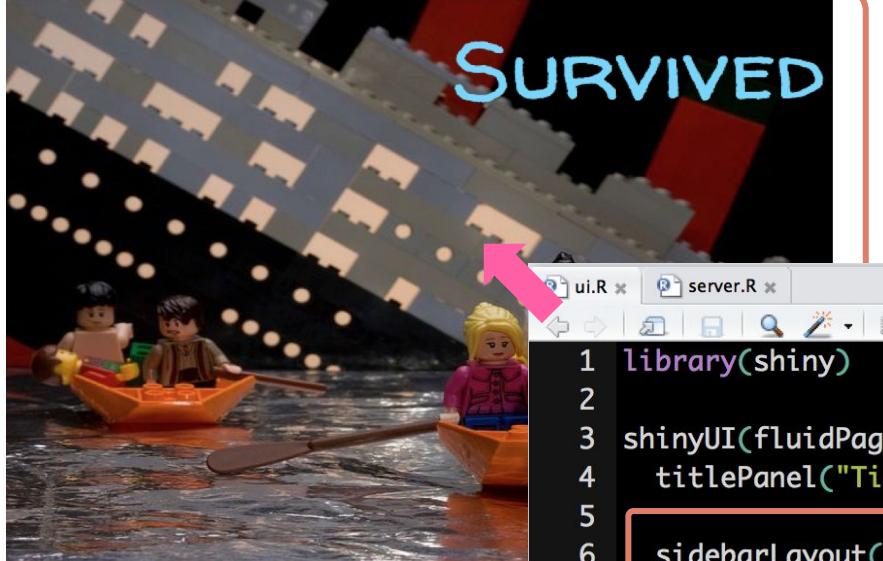
Port Embarkation : C

Age : 80

Sibling Spouse : 8

Parent Child : 9

Fare Price : 0 to 512.3292

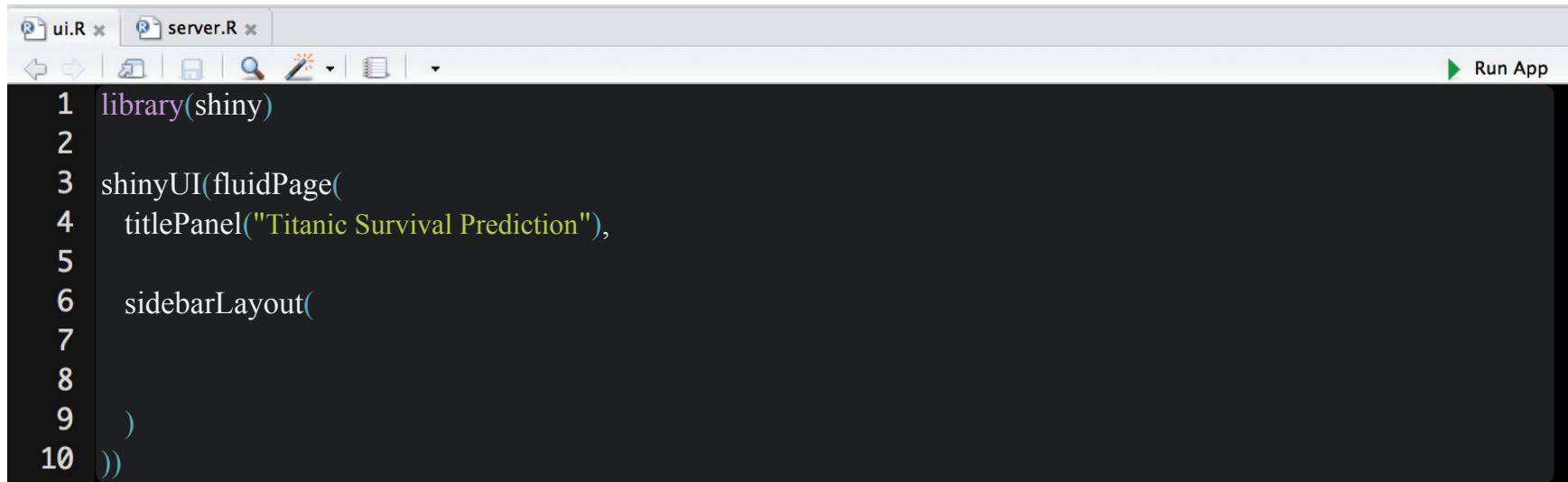


A pink arrow points from the word 'sidebarLayout' in the title bar to the corresponding line of code in the R script.

```
library(shiny)
shinyUI(fluidPage(
  titlePanel("Titanic Survival Prediction"),
  sidebarLayout(
```

1 library(shiny)
2
3 shinyUI(fluidPage(
4 titlePanel("Titanic Survival Prediction"),
5
6 sidebarLayout(
7
8)
9
10))

[Code] Titanic Survival Prediction -- Layout



The screenshot shows the RStudio interface with the ui.R file open. The code defines a simple Shiny application layout:

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5
6   sidebarLayout(
7
8
9   )
10 ))
```

The code includes imports, a title panel, and a sidebar layout.

Titanic Survival Prediction

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish

sidebarPanel

Passenger Class : 1

Gender : female

Port Embarkation : C

Age : 80

Sibling Spouse : 8

Parent Child : 9

Fare Price : 0 to 512.3292

mainPanel

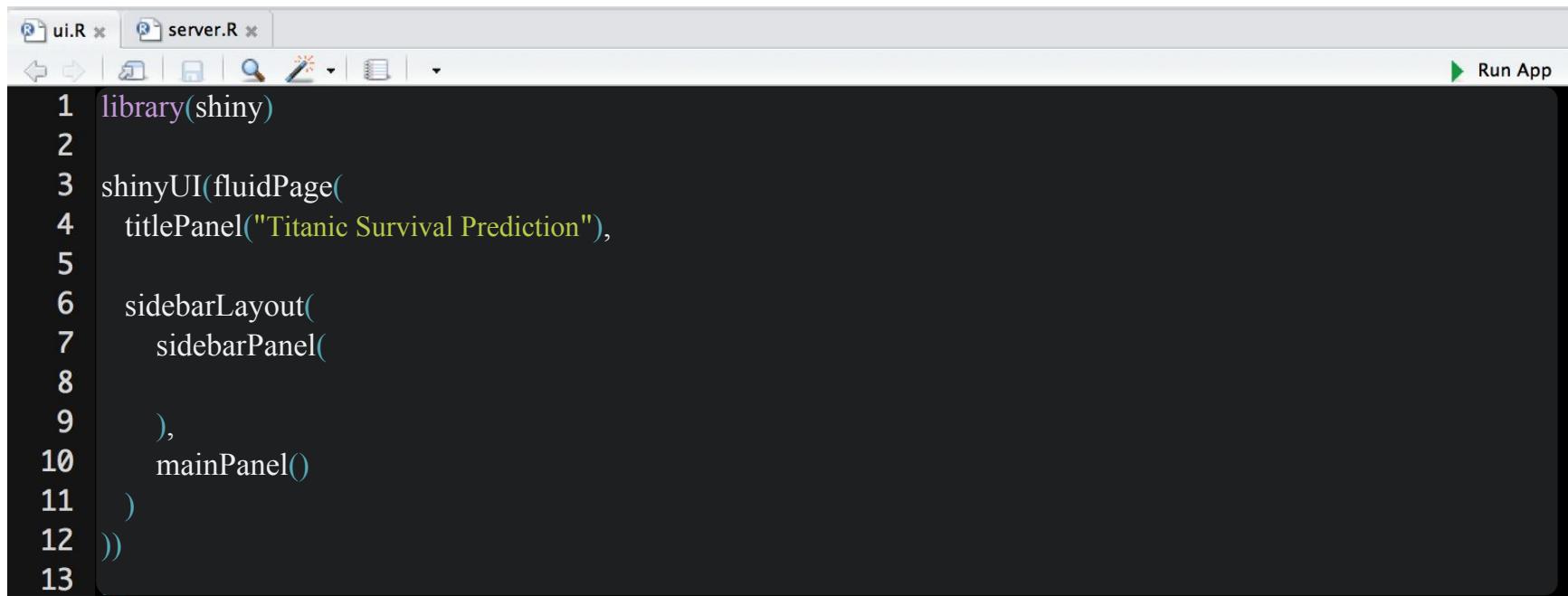


ui.R

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction") ,
5
6   sidebarLayout(
7     sidebarPanel(
8       ),
9       mainPanel(
10      )
11    )
12  )
13 ))
```

A pink arrow points from the "mainPanel" label to the "mainPanel(" line in the R code.

[Code] Titanic Survival Prediction -- Layout



The screenshot shows the RStudio interface with the ui.R file open. The code defines a Shiny application with a title panel and a sidebar layout.

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5
6   sidebarLayout(
7     sidebarPanel(
8
9     ),
10    mainPanel()
11  )
12))
13
```

Shiny Input -- sidePanel

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

Shiny Input -- sidePanel

K	<code>submitButton(text, icon)</code> (Prevents reactions across entire app)	M	<code>textInput(inputId, label, value)</code>
L	<code>selectInput(inputId, label, choices, selected, multiple, selectize, width, size) (also selectizeInput())</code>	N	<code>sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)</code>

Titanic Survival Prediction

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish

Titanic Survival Prediction

Passenger Class :

Gender :

Port Embarkation :

Age :
selectInput ↑

Sibling Spouse :

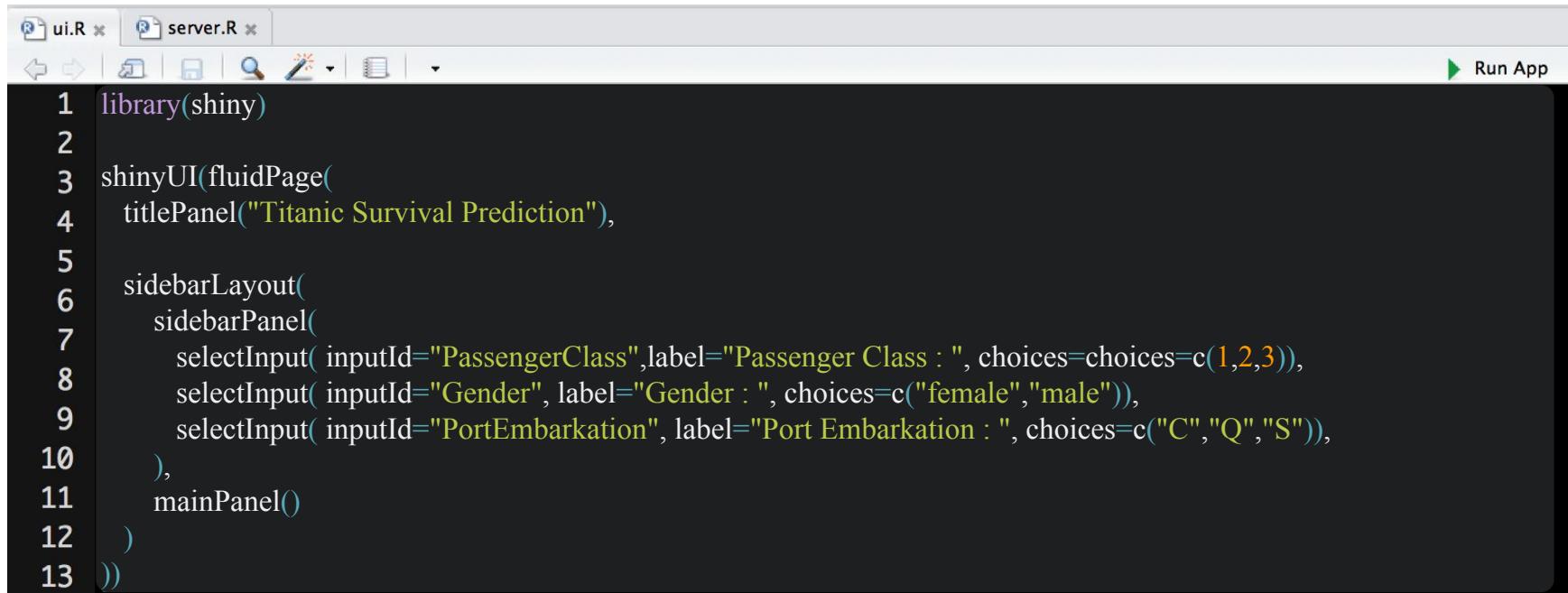
Parent Child :

Fare Price :



The image shows a creative visualization of the Titanic survival story using LEGO. Two minifigures are in small orange boats on a dark, textured surface representing water. In the background, a large LEGO model of the Titanic is depicted as it sinks, with its hull partially submerged and several red and white vertical supports visible. The word "SURVIVED" is prominently displayed in large blue capital letters on the side of the sinking ship. The overall scene is a playful yet somber take on the historical event.

[Code] Titanic Survival Prediction -- input



The screenshot shows the RStudio interface with the ui.R file open. The code defines a sidebar layout for a Shiny application. It includes a title panel, a sidebar panel with three dropdown inputs for Passenger Class, Gender, and Port Embarkation, and a main panel. The sidebar panel contains three selectInput statements. The first selectInput has choices 1, 2, and 3. The second has choices "female" and "male". The third has choices "C", "Q", and "S". The main panel is currently empty.

```
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5
6   sidebarLayout(
7     sidebarPanel(
8       selectInput( inputId="PassengerClass",label="Passenger Class : ", choices=choices=c(1,2,3)),
9       selectInput( inputId="Gender", label="Gender : ", choices=c("female","male")),
10      selectInput( inputId="PortEmbarkation", label="Port Embarkation : ", choices=c("C","Q","S")),
11    ),
12  )
13 ))
```

Titanic Survival Prediction-- Example:Gender

```
selectInput( "Gender", "Gender : ", choices = gender )
```

inputId label choices

```
gender = c( "female","male" )
```

Gender : ← label

female

female

male

← choices

Titanic Survival Prediction

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | Open in Browser | Publish

Titanic Survival Prediction

Passenger Class :

Gender :

Port Embarkation :

Age :

Sibling Spouse :

Parent Child :

Fare Price :

← numericInput

← sliderInput

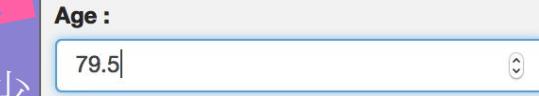
The screenshot shows a Shiny application titled "Titanic Survival Prediction". On the left, there are several input fields: "Passenger Class" (dropdown, value 1), "Gender" (dropdown, value "female"), "Port Embarkation" (dropdown, value "C"), "Age" (text input, value "80"), "Sibling Spouse" (text input, value "8"), "Parent Child" (text input, value "9"), and "Fare Price" (slider input, range 0 to 512.3292, current value 0). A purple button labeled "← numericInput" is positioned next to the "Age" field, and another purple button labeled "← sliderInput" is next to the "Fare Price" slider. The background of the app features a photograph of two LEGO minifigures in a lifeboat on dark water, with the word "SURVIVED" written in blue capital letters on a brick wall behind them. The URL "http://127.0.0.1:3857" and the title "~/Desktop/Shiny/Shiny_Titanic - Shiny" are visible at the top of the browser window.

[Code] Titanic Survival Prediction

```
numericInput( "Age", "Age : ", min = 0, max=100, value=10, step=0.5 )
```

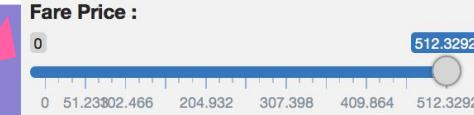
inputId label

min,max: 最大,最小值
value: 第一次啟動時值是多少
step: 間隔多少



```
sliderInput( "FacePrice", "Fare Price : ", min = 0, max=100, value=10, step=0.0001, sep="" )
```

min,max: 最大,最小值
value: 第一次啟動時值是多少
step: 間隔多少
sep: 千位數逗點符號; “” 表示不要有任何符號

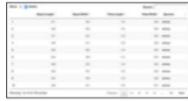


[Code] Titanic Survival Prediction -- input

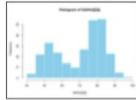
```
ui.R * server.R *
Run App
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5   sidebarLayout(
6     sidebarPanel(
7       selectInput( inputId="PassengerClass",label="Passenger Class : ", choices=c(1,2,3)),
8       selectInput( inputId="Gender", label="Gender : ", choices=c("female","male")),
9       selectInput( inputId="PortEmbarkation", label="Port Embarkation : ", choices=c("C","Q","S")),
10      numericInput( inputId="Age", label="Age : ", min=0, max=80, value=80, step=0.5),
11      numericInput( inputId="SiblingSpouse", label="Sibling Spouse : ",min=0, max=8, value=8, step=1),
12      numericInput( inputId="ParentChild", label="Parent Child : ", min=0, max=9, value=9, step=1),
13      sliderInput( inputId="FarePrice", label="Fare Price : ", min=0, max=512.35, value=512.35, step=0.0001, sep=""")
14    ),
15    mainPanel()
16  )
17 ))
```

Shiny Output – mainPanel

a

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

b

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

c

`uiOutput(outputId, inline, container, ...)`

d

```
'data.Frame': 3 obs. of  2 variables:  
$ Sepal.Length : num  5.1 4.9 4.7  
$ Sepal.Width  : num  3.5 3.3 3.2
```

`verbatimTextOutput(outputId)`

e

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.10	3.50	1.60	0.10	Iris-setosa
2	4.90	3.00	1.40	0.30	Iris-setosa
3	4.70	3.20	1.30	0.20	Iris-setosa
4	4.50	2.30	1.00	0.20	Iris-setosa
5	4.50	2.30	1.40	0.30	Iris-setosa
6	4.60	3.10	1.50	0.20	Iris-setosa
7	5.00	3.60	1.40	0.20	Iris-setosa
8	5.40	3.90	1.70	0.40	Iris-versicolor
9	4.90	3.00	1.50	0.30	Iris-versicolor
10	4.90	3.40	1.40	0.30	Iris-versicolor
11	4.80	3.40	1.30	0.20	Iris-versicolor
12	4.30	3.00	1.00	0.20	Iris-versicolor
13	5.80	4.00	1.20	0.20	Iris-virginica
14	5.10	3.80	1.60	0.20	Iris-virginica
15	5.90	3.80	4.80	1.80	Iris-virginica

`tableOutput(outputId)`

f

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

g

`foo``textOutput(outputId, container, inline)`

h

`htmlOutput(outputId, inline, container, ...)``imageOutput("plot_name")`the type of output
to displayname to give to the
output object

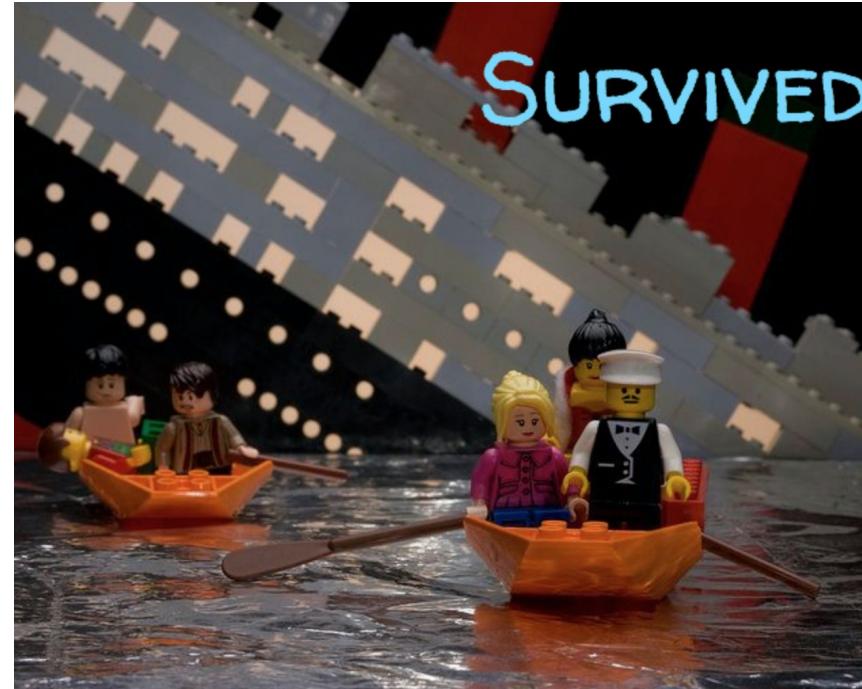
[Code] Titanic Survival Prediction

```
ui.R * server.R *
Run App
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5   sidebarLayout(
6     sidebarPanel(
7       selectInput( inputId="PassengerClass",label="Passenger Class : ", choices=c(1,2,3)),
8       selectInput( inputId="Gender", label="Gender : ", choices=c("female","male")),
9       selectInput( inputId="PortEmbarkation", label="Port Embarkation : ", choices=c("C","Q","S")),
10      numericInput( inputId="Age", label="Age : ", min=0, max=80, value=80, step=0.5),
11      numericInput( inputId="SiblingSpouse", label="Sibling Spouse : ",min=0, max=8, value=8, step=1),
12      numericInput( inputId="ParentChild", label="Parent Child : ", min=0, max=9, value=9, step=1),
13      sliderInput( inputId="FarePrice", label="Fare Price : ", min=0, max=512.35, value=512.35, step=0.0001, sep=""")
14    ),
15    mainPanel( imageOutput("result_plot") )
16  )
17 ))
18
```

Titanic Survival Prediction -- imageOutput

```
imageOutput("result_plot")
```

↓
outputId



[Code] Titanic Survival Prediction -- Original Server.R



The screenshot shows the RStudio interface with the 'server.R' tab active. The code in the editor is:

```
1 library(shiny)
2
3 shinyServer( function(input, output) {
4
5
6
7 })
```

[Review] Azure ML API in R

```
1 library("RCurl")
2 library("rjson")
3
4 # Accept SSL certificates issued by public Certificate Authorities
5 options(RCurlOptions = list(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl")))
6
7 h = basicTextGatherer()
8 hdr = basicHeaderGatherer()
9
10 req = list(
11   Inputs = list(
12     "input1" = list(
13       list(
14         'PassengerClass' = "3",
15         'Gender' = "male",
16         'Age' = "22",
17         'SiblingSpouse' = "1",
18         'ParentChild' = "0",
19         'FarePrice' = "7.25",
20         'PortEmbarkation' = "S"
21       )
22     ),
23     GlobalParameters = setNames(fromJSON('{}'), character(0))
24   )
25 )
26
27 body = enc2utf8(toJSON(req))
28 api_key = "abc123" # Replace this with the API key for the web service
29 authz_hdr = paste('Bearer', api_key, sep=' ')
30
31 h$reset()
```

"PassengerClass" = "3",
"Gender" = "female",
"Age" = "22",
"SiblingSpouse" = "1",
"ParentChild" = "0",
"FarePrice" = "7.25",
"PortEmbarkation" = "S"

Access Input Value

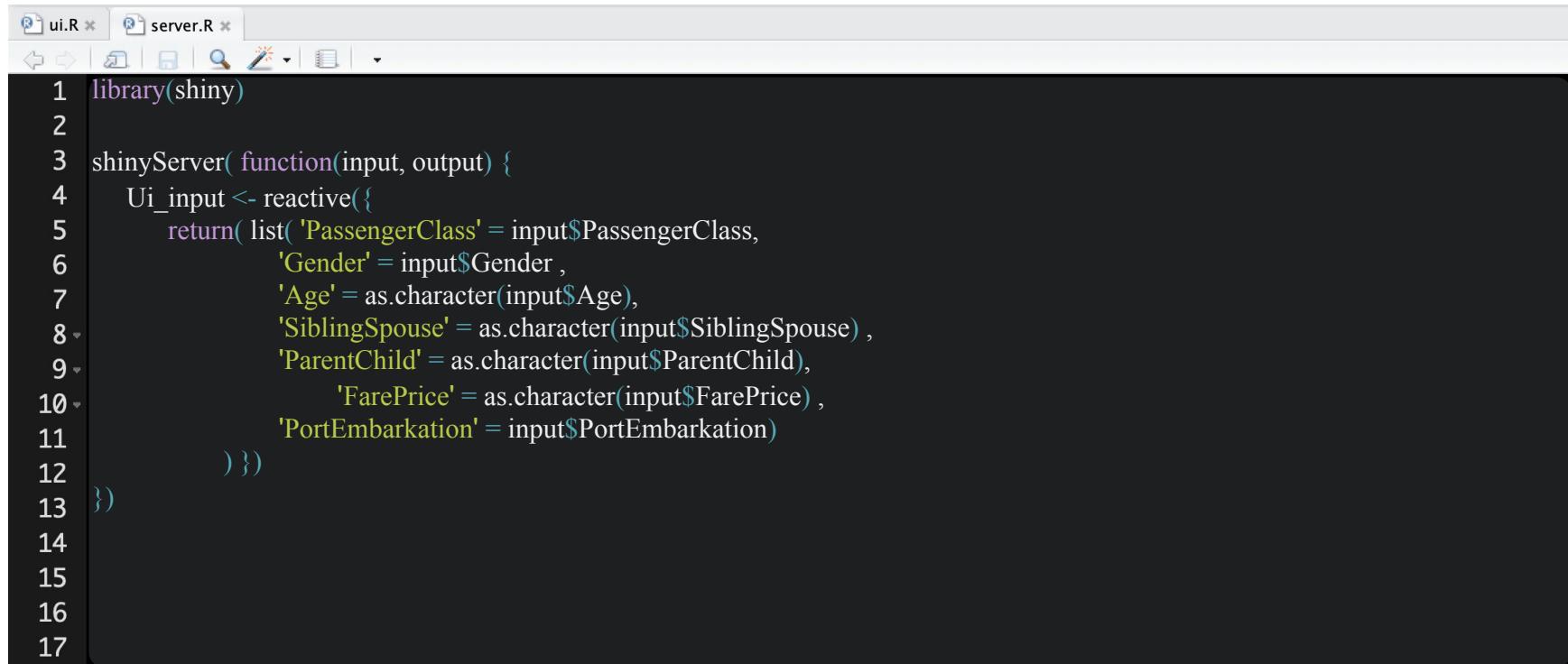
Server.R

```
8
9 #=====
10 # Server
11 #=====
12 function(input, output) {
13   #==== Get UI.R's input ====
14   Ui_input <- reactive({ ##### Check 1 #####
15     return( list( 'PassengerClass' <- input$PassengerClass
16                 'Gender' = input$Gender ,
17                 'Age' = as.character(input$Age),
18                 'SiblingSpouse' = as.character(input$Sib
19                 'ParentChild' = as.character(input$Pare
20                 'FarePrice' = as.character(input$FarePr
21                 'PortEmbarkation' = input$PortEmbarkati
22   })
23 }
```

Ui.R

```
ui.R x server.R x
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5   sidebarLayout(
6     sidebarPanel(
7       selectInput( inputId="PassengerClass", lab
8       selectInput( inputId="Gender", label="Gend
9       selectInput( inputId="PortEmbarkation", la
10      numericInput( inputId="Age", label="Age :
11      numericInput( inputId="SiblingSpouse", lab
12      numericInput( inputId="ParentChild", label
13      sliderInput( inputId="FarePrice", label="F
14    ),
15    mainPanel( imageOutput("result_plot") )
16  )
17
18 ))
```

[Code] Titanic Survival Prediction -- Access Input Value

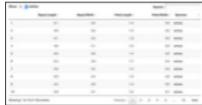


```
1 library(shiny)
2
3 shinyServer( function(input, output) {
4   Ui_input <- reactive({
5     return( list( 'PassengerClass' = input$PassengerClass,
6                  'Gender' = input$Gender ,
7                  'Age' = as.character(input$Age),
8                  'SiblingSpouse' = as.character(input$SiblingSpouse) ,
9                  'ParentChild' = as.character(input$ParentChild),
10                 'FarePrice' = as.character(input$FarePrice) ,
11                 'PortEmbarkation' = input$PortEmbarkation)
12   })
13 }
14
15
16
17 )}
```

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

render*() VS. Output()

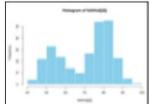
Server.R



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



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



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

`'data.Frame': 3 obs. of 2 variables:`
`$ Sepal.Length: num 5.1 4.9 4.7`
`$ Sepal.Width : num 3.5 3 3.2`

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-versicolor
7	7.0	3.2	1.0	0.2	Iris-versicolor
8	6.4	3.0	1.8	0.3	Iris-versicolor
9	5.7	3.8	1.3	0.2	Iris-versicolor
10	6.9	3.3	1.9	0.2	Iris-versicolor

foo



`renderPrint(expr, env, quoted, func, width)`

`renderTable(expr,..., env, quoted, func)`

`renderText(expr, env, quoted, func)`

`renderUI(expr, env, quoted, func)`

works
with

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)`

`verbatimTextOutput(outputId)`

`tableOutput(outputId)`

`textOutput(outputId, container, inline)`

`uiOutput(outputId, inline, container, ...)`

`htmlOutput(outputId, inline, container, ...)`

&

Build Output Object

Server.R

```
ui.R * server.R *
[File] [New] [Open] [Save] [Print] [Find] [Edit]
24  #==== Output : Prediction ====
25  output$result_plot <- renderImage{ 
26  #---- Connect to Azure ML workspace ----
27  | options(RCurlOptions = list(caIntra = system.fi
28  # Accept SSL certificates issued by public Cer
29
30  h = basicTextGatherer()
31  hdr = basicHeaderGatherer()
32
33  #---- Put input_data to Azure ML workspace ---
34  req = list(
35    Inputs = list(
36      "input1" = list(
37        Ui_input()
38    )
39  )
40  )
41  )
42  )
43  )
44  )
45  )
46  )
47  )
48  )
49  )
50  )
51  )
52  )
53  )
54  )
55  )
56  )
57  )
58  )
59  )
60  )
61  )
62  )
63  )
64  )
65  )
66  )
67  )
68  )
69  )
70  )
71  )
72  )
73  )
74  )
75  )
76  )
77  )
78  )
79  )
80  )
81  )
82  )
83  )
84  )
85  )
86  )
87  )
88  )
89  )
90  )
91  )
92  )
93  )
94  )
95  )
96  )
97  )
98  )
99  )
100 )
101 )
102 )
103 )
104 )
105 )
106 )
107 )
108 )
109 )
110 )
111 )
112 )
113 )
114 )
115 )
116 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
159 )
```

Ui.R

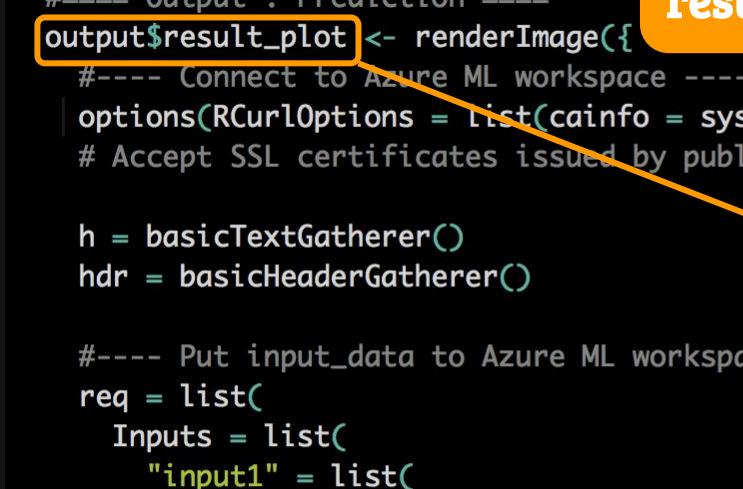
```
ui.R * server.R *
[File] [New] [Open] [Save] [Print] [Find] [Edit]
1 library(shiny)
2
3 shinyUI(fluidPage(
4   titlePanel("Titanic Survival Prediction"),
5   sidebarLayout(
6     sidebarPanel(
7       selectInput( inputId="PassengerClass", lab
8       selectInput( inputId="Gender", label="Gend
9       selectInput( inputId="PortEmbarkation", la
10      numericInput( inputId="Age", label="Age :
11      numericInput( inputId="SiblingSpouse", lab
12      numericInput( inputId="ParentChild", label
13      sliderInput( inputId="FarePrice", label="F
14    ),
15    mainPanel( imageOutput("result_plot") )
16  )
17
18 ))
```

Output Object Name

Server.R

OuputId :
result_plot

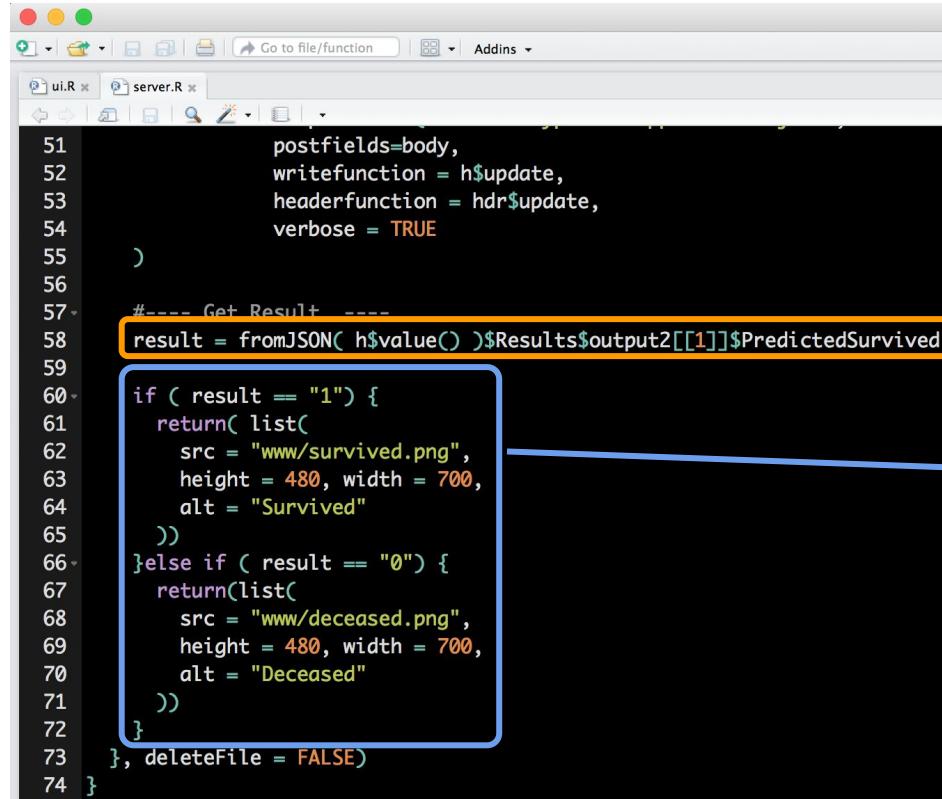
Ui.R



```
ui.R x server.R x
[File] [Edit] [View] [Project] [Help]
24  #==== Output : Prediction ====
25  output$result_plot <- renderImage({
26  #---- Connect to Azure ML workspace ----
27  | options(RCurlOptions = list(cainfo = system.
28  # Accept SSL certificates issued by public Ce
29
30  h = basicTextGatherer()
31  hdr = basicHeaderGatherer()
32
33  #---- Put input_data to Azure ML workspace --
34  req = list(
35    Inputs = list(
36      "input1" = list(
37        Ui_input()
38      )

```

Get AzureML Result & Setting Output Plot



```
51         postfields=body,
52         writefunction = h$update,
53         headerfunction = hdr$update,
54         verbose = TRUE
55     )
56
57     #---- Get Result ----
58     result = fromJSON( h$value() )$Results$output2[[1]]$PredictedSurvived
59
60     if ( result == "1" ) {
61         return( list(
62             src = "www/survived.png",
63             height = 480, width = 700,
64             alt = "Survived"
65         ))
66     } else if ( result == "0" ) {
67         return( list(
68             src = "www/deceased.png",
69             height = 480, width = 700,
70             alt = "Deceased"
71         ))
72     }
73 }, deleteFile = FALSE)
74 }
```

←AzureML Result

```
if ( result == "1" ) {
    Survived Plot
} else if ( result == "0" ) {
    Deceased Plot
}
```

~/Desktop/Shiny/Shiny_Titanic - Shiny

http://127.0.0.1:3857 | [Open in Browser](#) | 

Titanic Survival Prediction

Passenger Class :

1

Gender :

female

Port Embarkation :

C

Age :

80

Sibling Spouse :

8

Parent Child :

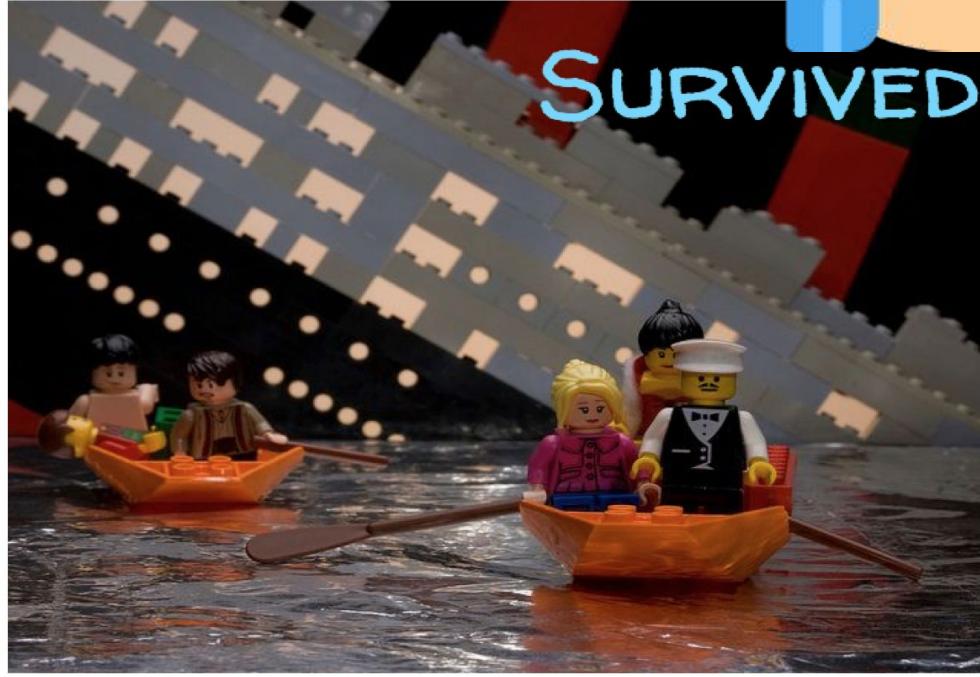
9

Fare Price :

0 512.3292

512.3292

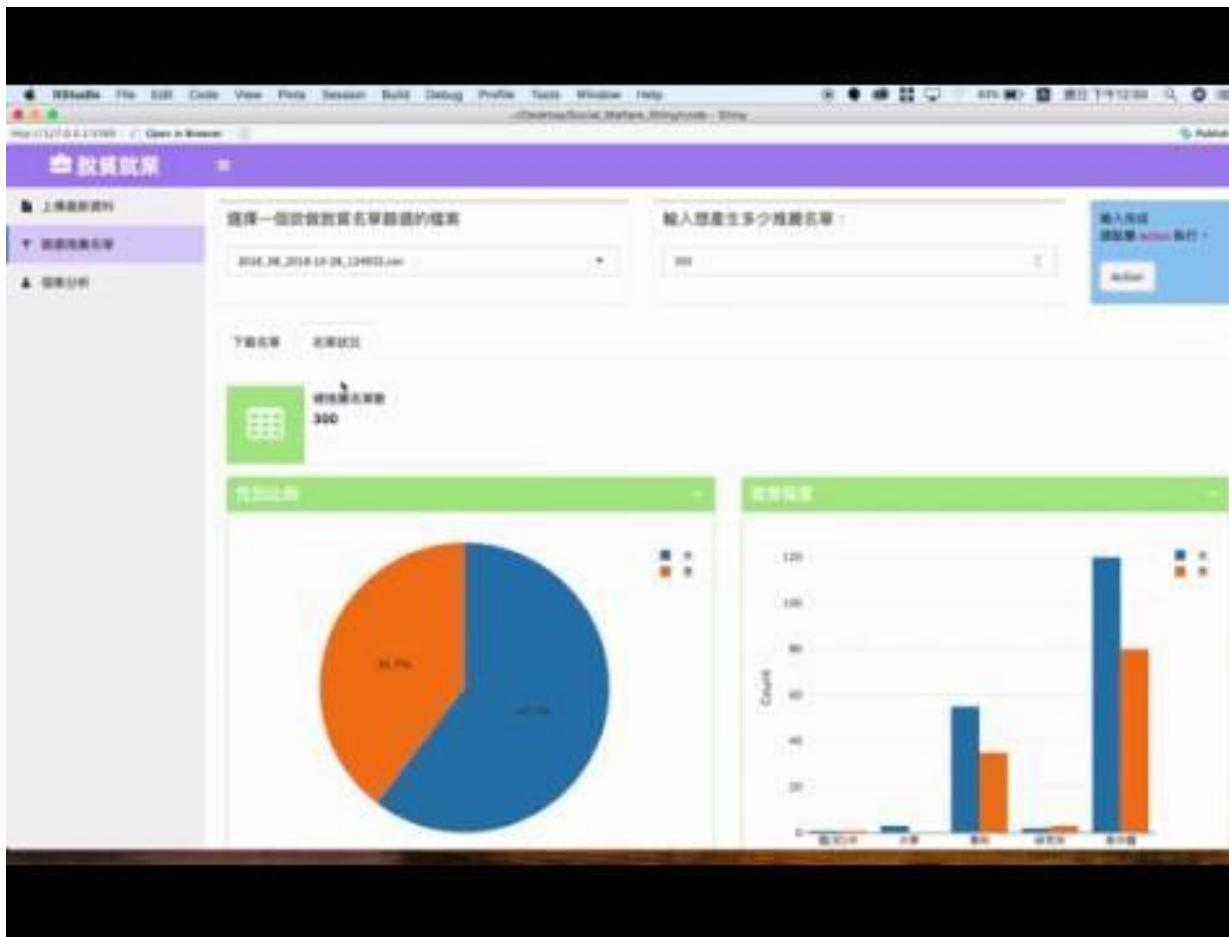
0 51.23802.466 204.932 307.398 409.864 512.3292



Kristen Chan 139

產生脫貧推薦名單

產生脫貧推薦名單



Reference

✓ Shiny Cheat Sheet

<https://shiny.rstudio.com/articles/cheatsheet.html>

✓ More Shiny Example

<https://shiny.rstudio.com/gallery/>