

Data Visualization

Getting Fancy: Interactive Data Visualization

Ciara Zogheib

Data Sciences Institute, University of Toronto

Exit Survey Time!



Assignment 4 Tip – Importing Data

- Toronto Open Data only (sorry!)
- Load datasets directly into R (without downloading as CSV/XLSX) using the related R package

```
library(opendatatoronto)
list_package_resources("https://open.toronto.ca/dataset/nei-
ghbourhoods/") |>
  head(1) |>
  get_resource()
```

In today's class, we will...

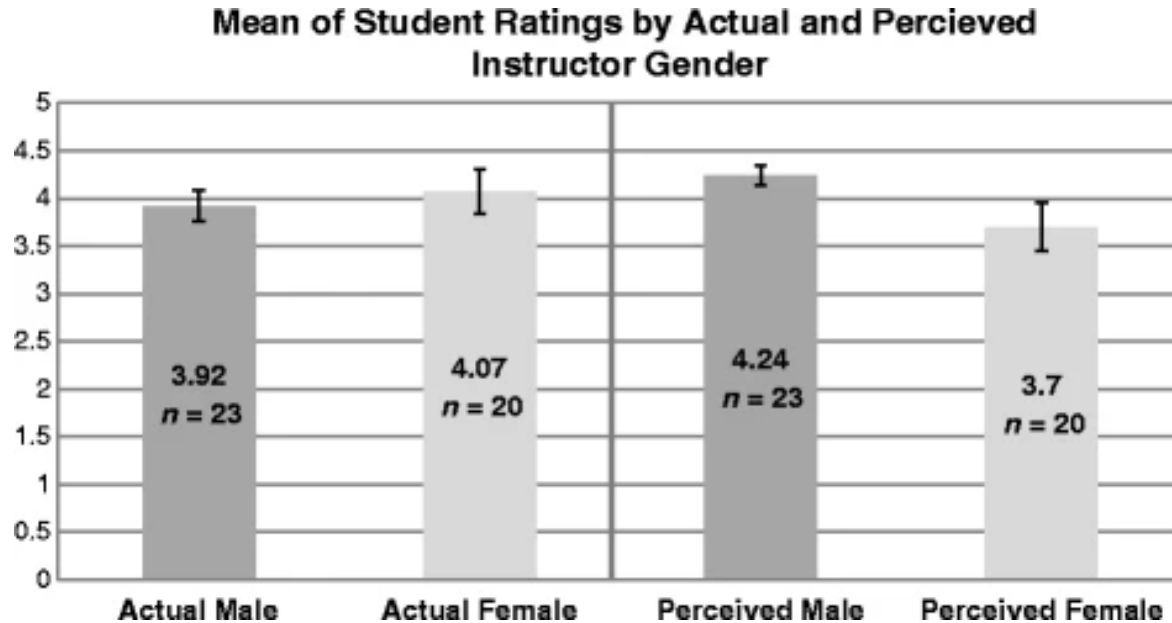
- Explore examples of dynamic data visualization
- Discuss the costs, benefits, and design considerations for creating dynamic data visualizations
- Develop basic understanding of rAmCharts, gganimate, and Shiny packages for creating dynamic data visualizations in R
- Review course content
- Work on (/ask questions about) Assignment 4

Case Study: Gender bias in teaching evaluations

How does faculty gender impact course evaluations?

- The gender of instructors ([perceived](#) or [actual](#)) influences how they are scored by students on teaching and course evaluations, with student evaluations tending to be significantly biased in favour of men and against women, even when all else is equal
- The bias in these student evaluations can adversely impact the ability of female scholars to be “full-time tenure-track, to hold tenured positions, to attain higher leadership roles in academia, and to earn the same salary as males in the same positions”

Mean of Student Ratings by Actual and Perceived Instructor Gender (MacNell et al., 2015)



Gendered Language in Teacher Reviews (Schmidt, 2015)

Gendered Language in Teacher Reviews

I've had [trouble keeping this site up continuously](#) during COVID. As of March 2021, I'm now trying a [new strategy](#) to cache common queries on the server even when the underlying database is down. If you find that many searches don't change the results, that's why.

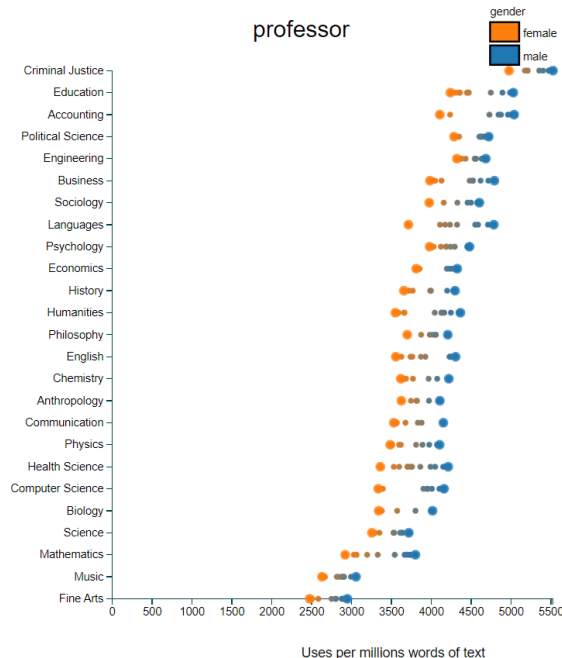
This interactive chart lets you explore the words used to describe male and female teachers in about 14 million reviews from RateMyProfessor.com.

Not all words have gender splits, but a surprising number do. Even things like pronouns are used quite differently by gender.

Search term(s) (case-insensitive):
use commas to aggregate multiple terms

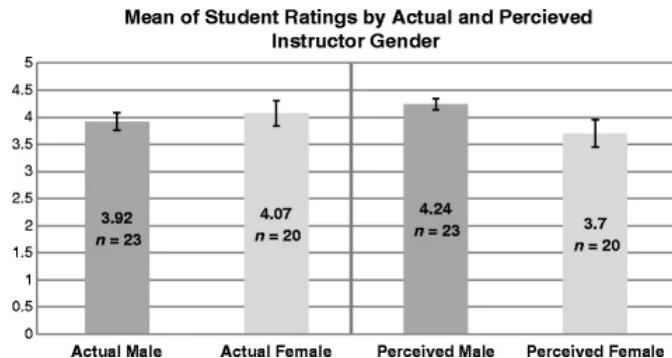
[All ratings](#) [Only positive](#) [Only negative](#)

You can enter any other word (or two-word phrase) into the box above to see how it is split across gender and



(Click to go to dynamic visualization)

Activity: Comparing data visualizations



Gendered Language in Teacher Reviews

I've had trouble keeping this site up continuously during COVID. As of March 2021, I'm now trying a new strategy to cache common queries on the server even when the underlying database is down. If you find that many searches don't change the results, that's why.

This interactive chart lets you explore the words used to describe male and female teachers in about 14 million reviews from RateMyProfessor.com.

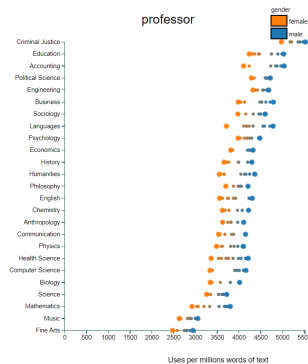
Not all words have gender splits, but a surprising number do. Even things like pronouns are used quite differently by gender.

Search term(s) (case-insensitive):
use commas to aggregate multiple terms

professor

All ratings Only positive Only negative

You can enter any other word (or two-word phrase) into the box above to see how it is split across gender and



- Let's discuss the two examples.
 - How are they different? What does each visualization 'do'?
 - What are the pros and cons of each?

Static vs. dynamic data visualization

Defining static vs. dynamic data visualization

- Static data visualization

- An image-based chart or infographic (think PDF, PNG, JPG)
- A snapshot of data
- Most of what we have seen so far in this course

- Dynamic data visualization

- Interactive applications or web pages that allow users to modify or filter a data visualization
- Multiple data stories in one

Benefits of dynamic data visualizations

- Dynamic data visualizations
 - Can provide information that cannot be obtained from static charts
 - Are useful for viewing individual-level data
 - Allow audiences to explore the data in-depth, supporting transparency and reproducibility
 - Can increase interest and engagement in research outputs

Costs of dynamic data visualizations

- Dynamic data visualizations
 - Make it more challenging to tell a single clear story or communicate a clear message
 - Can be confusing or overwhelming for audiences
 - Present access and sharing challenges (e.g. an image can be viewed in print, online, on mobile, but a dynamic viz might need to be hosted on a particular platform or software)

Designing dynamic data visualizations

Considerations

- Best practices for data visualization and accessibility still apply, but we also need to consider unique elements of dynamic data visualizations. For example:
 - Are the interactive elements easy to navigate?
 - Are the interactive elements accessible?
 - How much time will your audience have to interact with the data visualization?
 - **Important: Do interactive features actually help your data visualization serve your purpose?**

Elements of dynamic data visualizations

- We can conceptualize changes to our dynamic data visualizations in terms of how they affect two visual elements of our plots:
 - **Spatial elements** → Position and quantity (e.g. number of data points on a plot, scales of axes)
 - **Retinal elements** → Size, brightness, rotation, patterning, shape, and colour
- **We need to consider:** in our dynamic data visualizations, will spatial and retinal elements be fixed or mutable? Can they be created and deleted based on user interaction? Can their meanings change?

Types of changes

Identity-preserving	<ul style="list-style-type: none">• Maintain associations between visual elements and underlying data• Some part of a representation stays constant (eg. keeping the relative position of data points constant)• Important for comparing across time/snapshots
Transitional	<ul style="list-style-type: none">• Maintain some associations, and limit changes to known values• New elements may be added or changed• Balance flexibility with the ability to compare over short period of time/small changes between snapshots
Immediate	<ul style="list-style-type: none">• Generally do not preserve associations• Create and delete elements, alter scales, remaps variables every time a new 'snapshot' is created• Does not allow for comparison across time/snapshots

Activity: Types of changes

- Return to our earlier example of gendered language from RateMyProfessor.com
- What types of changes occur when we interact with this dynamic data visualization?
- Which spatial and retinal elements stay constant? Which change?
- If you were redesigning this viz, would you choose to make different changes? Why or why not?

Gendered Language in Teacher Reviews

I've had trouble keeping this site up continuously during COVID. As of March 2021, I'm now trying a new strategy to cache common queries on the server even when the underlying database is down. If you find that many searches don't change the results, that's why.

This interactive chart lets you explore the words used to describe male and female teachers in about 14 million reviews from RateMyProfessor.com.

Not all words have gender splits, but a surprising number do. Even things like pronouns are used quite differently by gender.

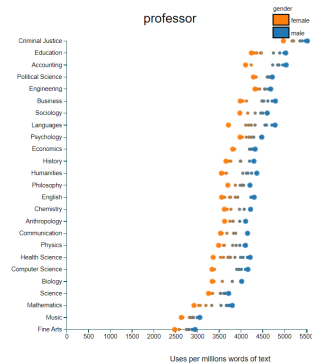
Search term(s) (case-insensitive):

use comma to aggregate multiple terms

professor

All ratings Only positive Only negative

You can enter any other word (or two-word phrase) into the box above to see how it is split across gender and



Tools for dynamic data visualization

Tools for dynamic data visualizations



Power BI



rAmCharts

- One of the simplest ways to create a dynamic data visualization in R is to use the `rAmCharts` library
- We can experiment using the iris sample dataset:

```
library(rAmCharts)
amPlot(iris, col = c('Sepal.Length', 'Sepal.Width'),
       type = c("line", "step"),
       zoom = TRUE, legend = TRUE)
```

rAmCharts

Your interactive plot will appear in the 'viewer' pane - try hovering over parts of the plot with your cursor!

```
library(rAmCharts)
amPlot(iris, col = colnames(iris)[1:2], type = c("l", "st"),
       zoom = TRUE, legend = TRUE)
```

Environment: Empty

Files | Plots | Packages | Help | Viewer

iris

as shown by amCharts

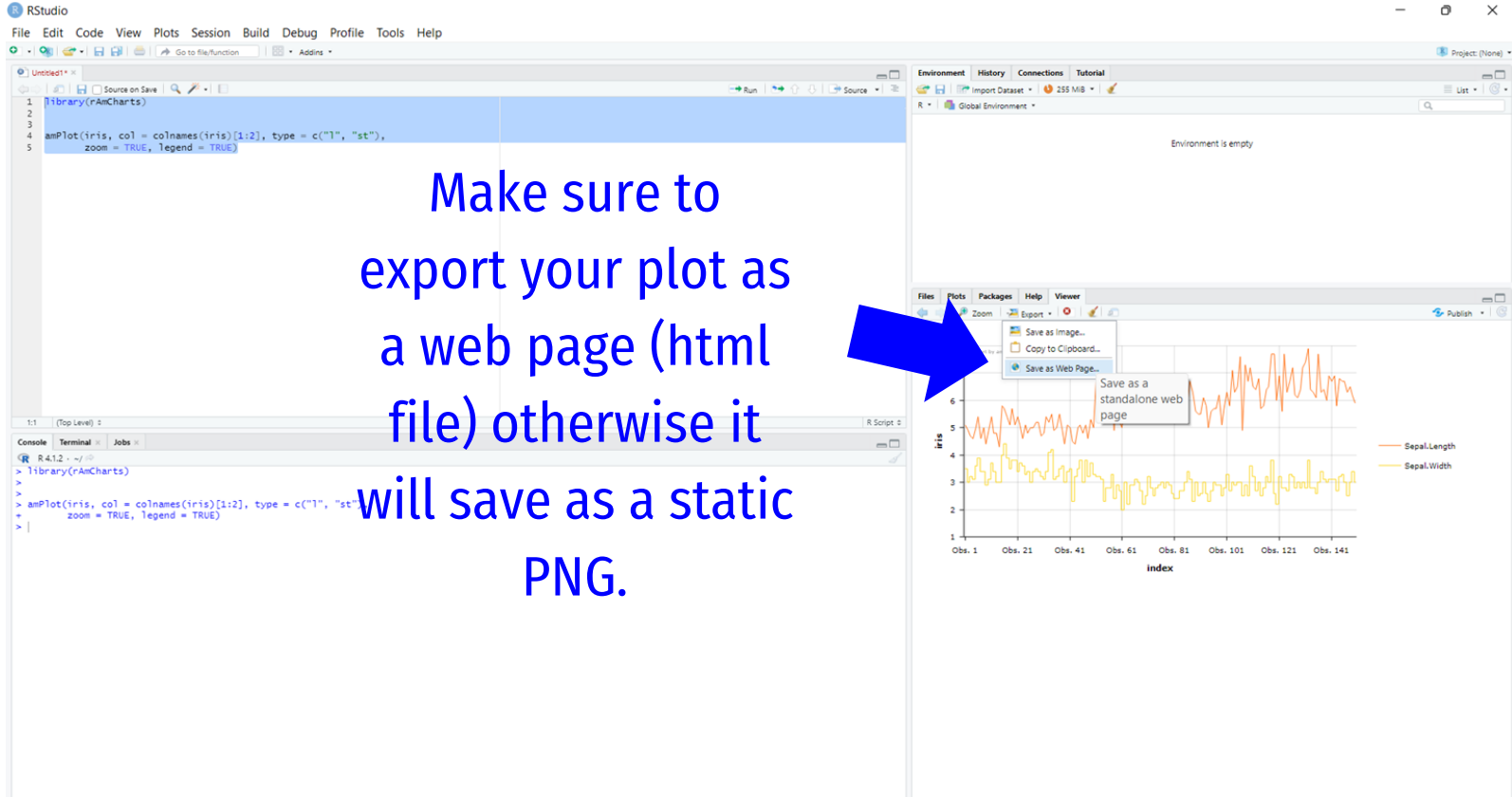
Sepal.Length

Sepal.Width

index

rAmCharts

Make sure to export your plot as a web page (html file) otherwise it will save as a static PNG.



The screenshot displays the RStudio interface. The top-left pane shows the R script editor with the following code:

```
1 library(rAmCharts)
2
3
4 amPlot(iris, col = colnames(iris)[1:2], type = c("l", "st"),
5        zoom = TRUE, legend = TRUE)
```

The bottom-left pane shows the console output:

```
> library(rAmCharts)
> amPlot(iris, col = colnames(iris)[1:2], type = c("l", "st"),
+        zoom = TRUE, legend = TRUE)
> |
```

The top-right pane shows the Environment window, which is empty. The bottom-right pane shows the Viewer window displaying a line plot. The plot has 'index' on the x-axis (ranging from 1 to 141) and 'Iris' on the y-axis (ranging from 1 to 6). Two lines are plotted: 'Sepal.Length' (orange line) and 'Sepal.Width' (yellow line). The 'Export' menu is open, showing options: 'Zoom', 'Save as Image...', 'Copy to Clipboard...', and 'Save as Web Page...'. A blue arrow points from the text to the 'Save as Web Page...' option.

gganimate

- `rAmCharts` uses base R graphing functionality, but if we want to use `ggplot`, we can play with the `gganimate` add-on and `gifski` renderer
- To demonstrate, we will use our `gapminder` sample dataset

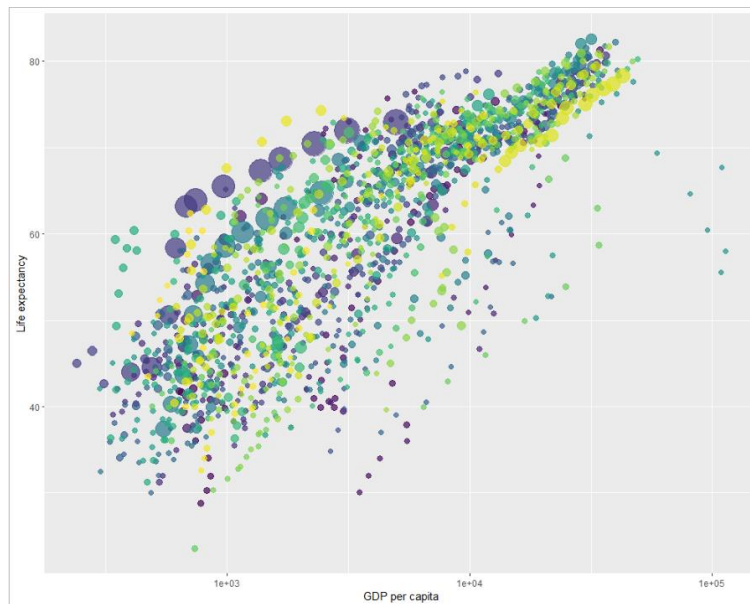
```
library(ggplot2)
library(gganimate)
library(gifski)
library(gapminder)
```


gganimate

- First, we make a static ggplot: a scatterplot showing the relationship between GDP and life expectancy

```
p <- ggplot(gapminder, aes(x = gdpPercap, y=lifeExp, size = pop,
  colour = country)) +
  geom_point(show.legend = FALSE, alpha = 0.7) +
  scale_color_viridis_d() +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  labs(x = "GDP per capita", y = "Life expectancy")
p
```

gganimate



```
p <- ggplot(gapminder, aes(x = gdpPercap, y=lifeExp, size = pop, colour =  
country)) + geom_point(show.legend = FALSE, alpha = 0.7) +  
scale_color_viridis_d() + scale_size(range = c(2, 12)) + scale_x_log10()  
+ labs(x = "GDP per capita", y = "Life expectancy")
```

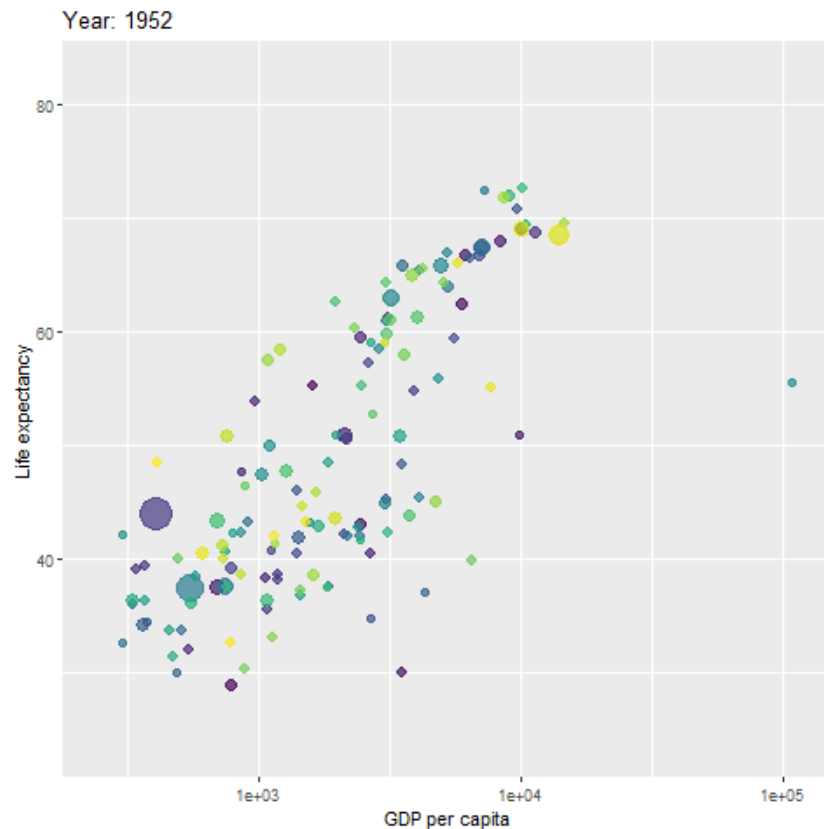
gganimate

- If we want to show how our plot, `p`, changes over time, we use the `transition_time()` function, with our time variable as the argument
- The `frame_time` variable in our label argument ensures that our label will change along with the plot

```
p2 <- p + transition_time(year) +  
  labs(title = "Year: {frame_time}")  
animate(p2, duration = 10, renderer = gifski_renderer())
```

gganimate

```
p2 <- p + transition_time(year) +  
  labs(title = "Year:  
  {frame_time}")  
  
animate(p2,duration = 10,renderer  
= gifski_renderer())
```



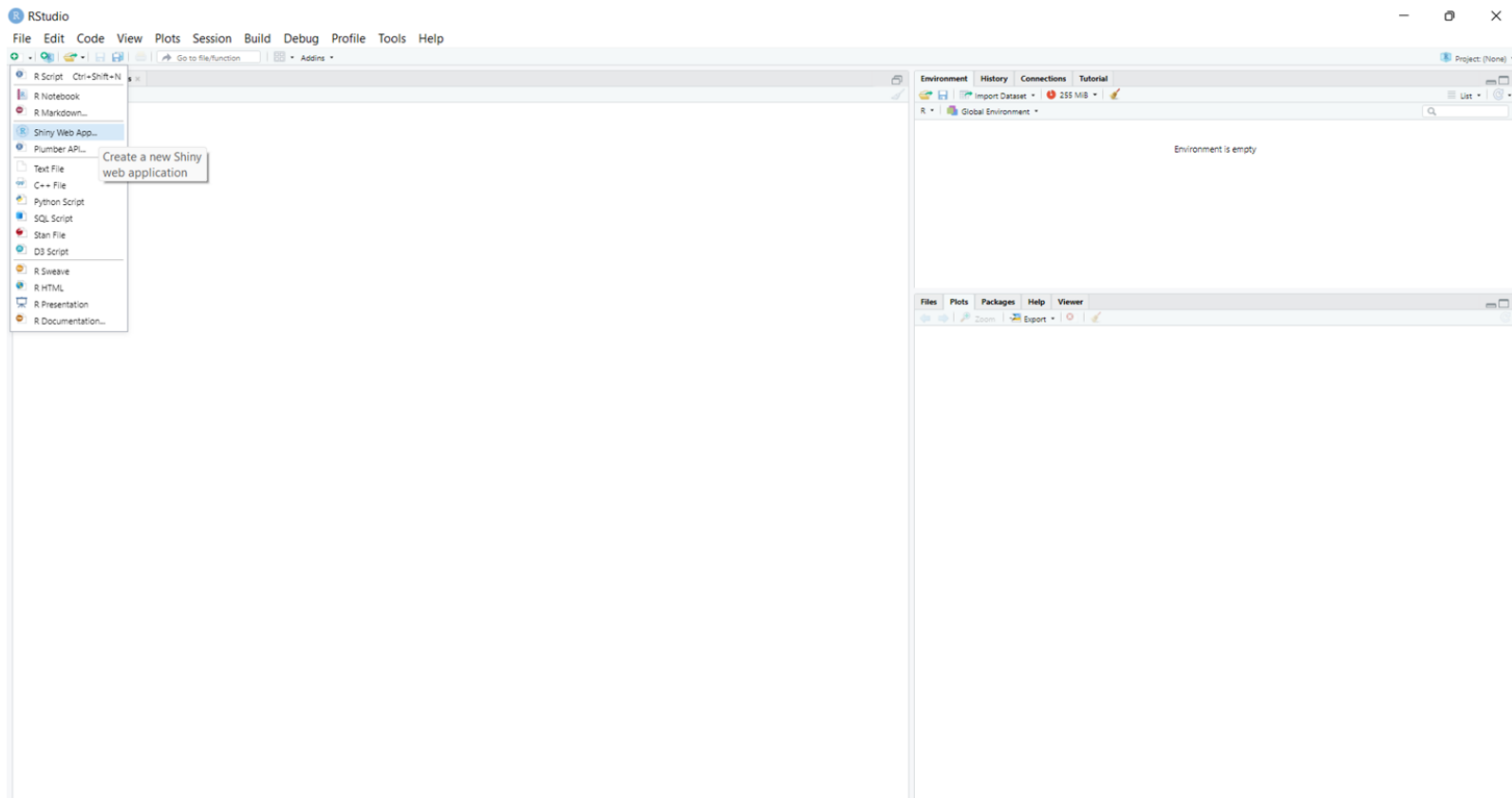
Shiny

- Shiny is a package that allows us to use R language to code web applications
- We can use Shiny to create dynamic data visualizations
- A Shiny app has three components:
 - user interface object
 - server function
 - call to the shinyApp function

First: Install shiny

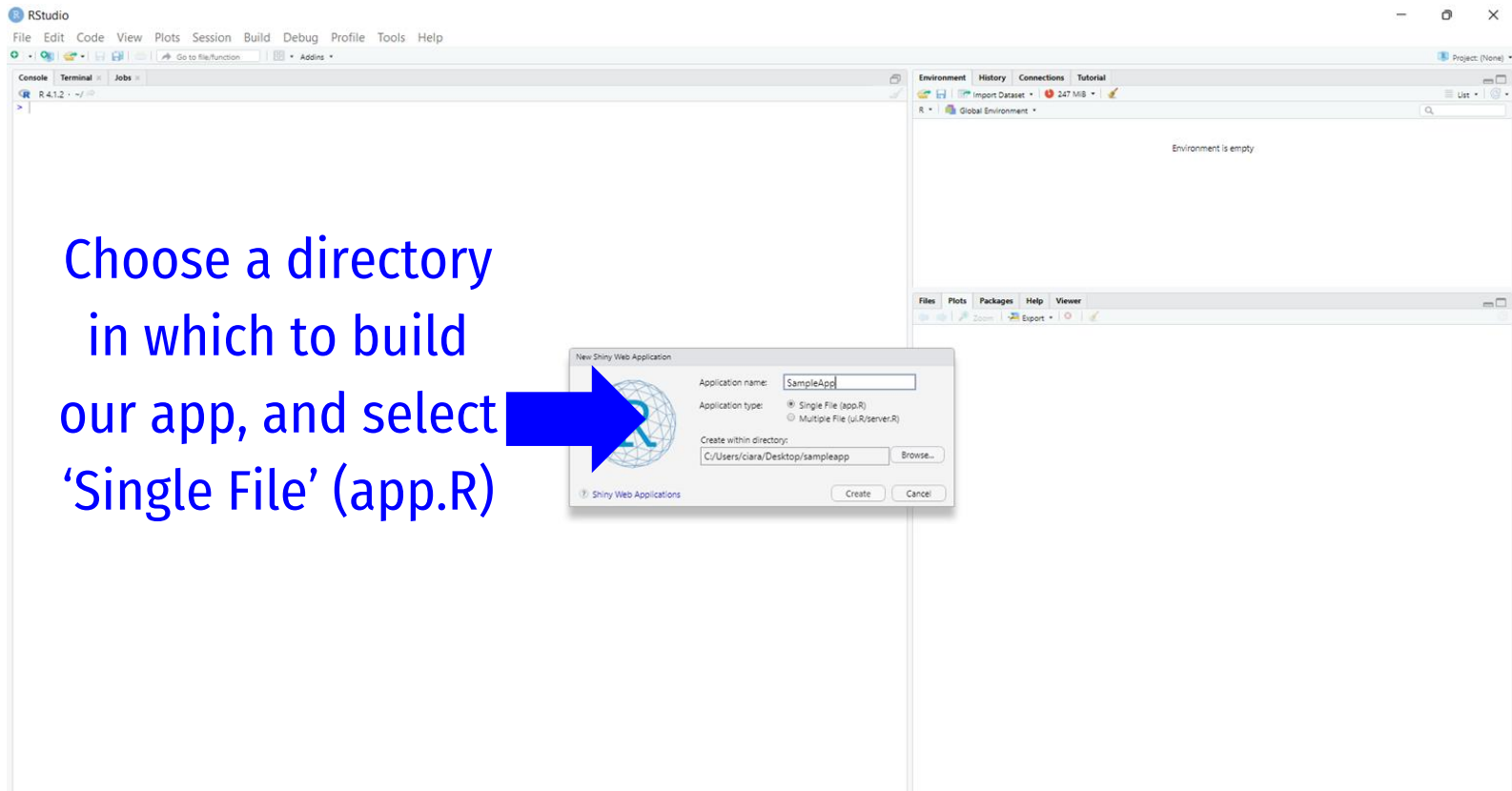
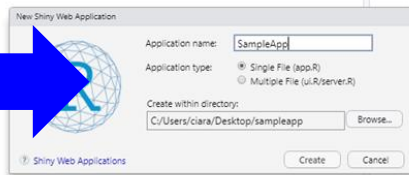
```
install.packages("shiny")
```

Shiny

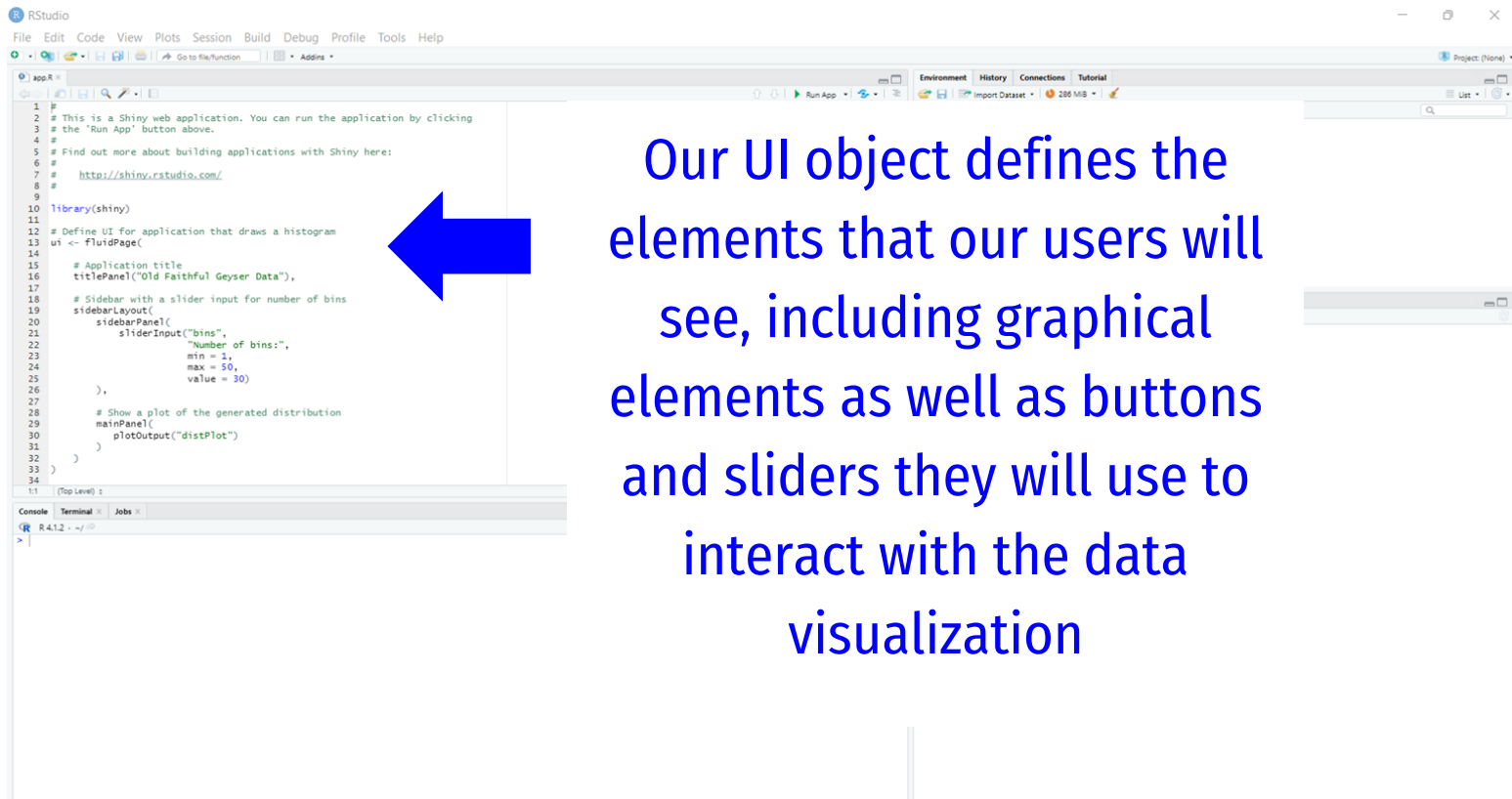


Shiny

Choose a directory
in which to build
our app, and select
'Single File' (app.R)



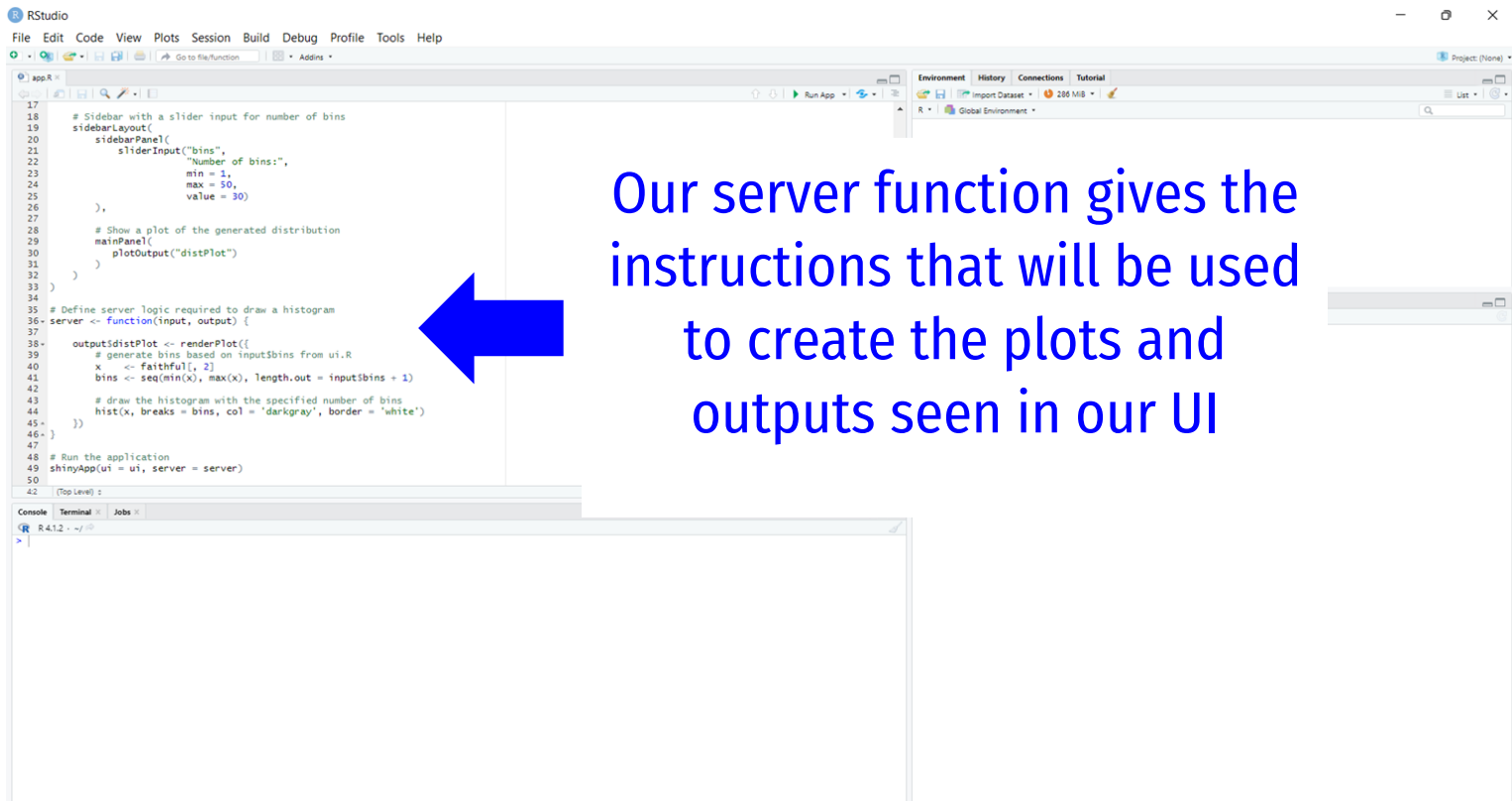
Shiny



```
1 #
2 # This is a Shiny web application. You can run the application by clicking
3 # the 'Run App' button 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 ui <- fluidPage(
14
15   # Application title
16   titlePanel("Old Faithful Geyser Data"),
17
18   # Sidebar with a slider input for number of bins
19   sidebarLayout(
20     sidebarPanel(
21       sliderInput("bins",
22         "Number of bins:",
23         min = 1,
24         max = 50,
25         value = 30)
26     ),
27
28     # Show a plot of the generated distribution
29     mainPanel(
30       plotOutput("distPlot")
31     )
32   )
33 )
34
```

Our UI object defines the elements that our users will see, including graphical elements as well as buttons and sliders they will use to interact with the data visualization

Shiny



The screenshot shows the RStudio interface with a Shiny application code file open. The code defines a user interface (UI) and a server function. A large blue arrow points from the text on the right to the server function in the code.

```
17 # Sidebar with a slider input for number of bins
18 sidebarLayout(
19   sidebarPanel(
20     sliderInput("bins",
21               "Number of bins:",
22               min = 1,
23               max = 50,
24               value = 30)
25   ),
26   # Show a plot of the generated distribution
27   mainPanel(
28     plotOutput("distPlot")
29   )
30 )
31
32 # Define server logic required to draw a histogram
33 server <- function(input, output) {
34   output$distPlot <- renderPlot({
35     # generate bins based on input$bins from ui.R
36     x <- faithful[, 2]
37     bins <- seq(min(x), max(x), length.out = input$bins + 1)
38     # draw the histogram with the specified number of bins
39     hist(x, breaks = bins, col = 'darkgray', border = 'white')
40   })
41 }
42
43 # Run the application
44 shinyApp(ui = ui, server = server)
```

Our server function gives the instructions that will be used to create the plots and outputs seen in our UI

Shiny

The screenshot shows the RStudio interface with a Shiny application code file open. The code defines a sidebar with a slider input for the number of bins and a main panel that displays a histogram. The server logic generates bins based on the input and draws the histogram. The application is run using the `shinyApp` function.

```
17  
18 # Sidebar with a slider input for number of bins  
19 sidebarLayout(  
20   sidebarPanel(  
21     sliderInput("bins",  
22       "Number of bins:",  
23       min = 1,  
24       max = 50,  
25       value = 30)  
26   ),  
27   # Show a plot of the generated distribution  
28   mainPanel(  
29     plotOutput("distPlot")  
30   )  
31 )  
32  
33  
34  
35 # Define server logic required to draw a histogram  
36 server <- function(input, output) {  
37  
38   output$distPlot <- renderPlot({  
39     # generate bins based on input$bins from ui.R  
40     x <- faithful[, 2]  
41     bins <- seq(min(x), max(x), length.out = input$bins + 1)  
42  
43     # draw the histogram with the specified number of bins  
44     hist(x, breaks = bins, col = 'darkgray', border = 'white')  
45   })  
46 }  
47  
48 # Run the application  
49 shinyApp(ui = ui, server = server)  
50  
51  
52 [Top Level] :  
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  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

Our shinyApp function calls our UI and server to run our app

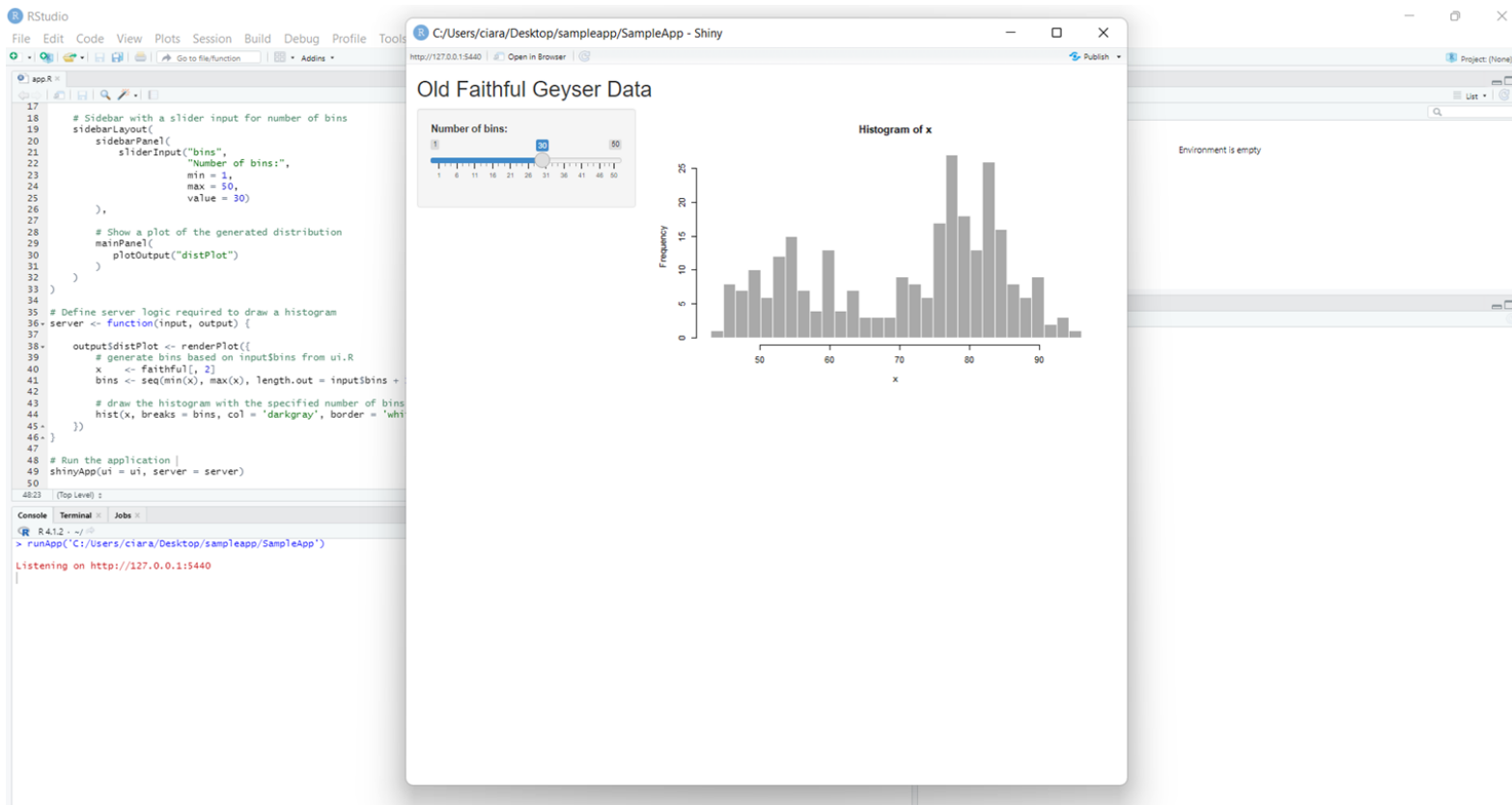
Shiny

The screenshot displays the RStudio environment with a Shiny application script open in the editor. The script defines a sidebar with a slider input for the number of bins and a main panel that renders a histogram. A blue arrow points to the 'Run App' button in the top right toolbar. The Environment pane on the right shows an empty global environment, and the Console at the bottom is ready for input.

```
17  
18 # Sidebar with a slider input for number of bins  
19 sidebarLayout(  
20   sidebarPanel(  
21     sliderInput("bins",  
22       "Number of bins:",  
23       min = 1,  
24       max = 50,  
25       value = 30)  
26   ),  
27   # Show a plot of the generated distribution  
28   mainPanel(  
29     plotOutput("distPlot")  
30   )  
31 )  
32 )  
33  
34  
35 # Define server logic required to draw a histogram  
36 server <- function(input, output) {  
37  
38   output$distPlot <- renderPlot({  
39     # generate bins based on input$bins from ui.R  
40     x <- faithful[, 2]  
41     bins <- seq(min(x), max(x), length.out = input$bins + 1)  
42  
43     # draw the histogram with the specified number of bins  
44     hist(x, breaks = bins, col = 'darkgray', border = 'white')  
45   })  
46 }  
47  
48 # Run the application  
49 shinyApp(ui = ui, server = server)  
50  
51  
52 [Top Level]
```

Click 'Run App' in top right to view and interact with our app

Shiny



Sharing Shiny apps

- Shiny apps can be published at shinyapps.io, where they are accessible to anyone with the URL
- Once hosted at shinyapps.io, our interactive data visualizations can be embedded in websites, used as presentation aids, or shared with the public!

Review Time!

Course Topics

- **First Steps**

- Get started
- Make a plot (substantive qualities)
- Thinking about reproducibility

- **Graphing Our Data**

- Show the right numbers (subsets, changing axes)
- Graph tables, add labels
- Choosing the right visualization (perceptual qualities)

Course Topics

- **Visualization with Purpose**
 - Refine our plots (aesthetic qualities)
 - Colour theory and accessible design
 - Data visualization as advocacy
- **Getting Fancy**
 - Working with models
 - Drawing maps and flow charts
 - Interactive data visualizations



Assignment 4

Congratulations!

You have finished your intro to Data Visualization!

Learning Objectives of this Course

1. Develop ability to **create and customize data visualizations** start to finish in R
2. Build an understanding of general design principles for creating **accessible and equitable** data visualizations in R and other software
3. Build an understanding of **data visualization as purposeful/telling a story** (and the ethical/professional implications thereof)