

Interactive data: Introduction to Shiny

Sophia Gibson
Computational Genomics
December 2, 2023



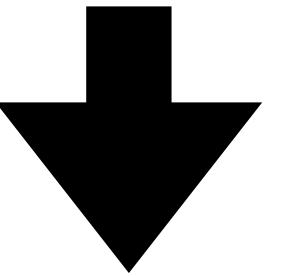
There will be an interactive portion
Run: install.packages("shiny")

Me everyday

What can I do
with my LRS data?

Me everyday

What can I do
with my LRS data?

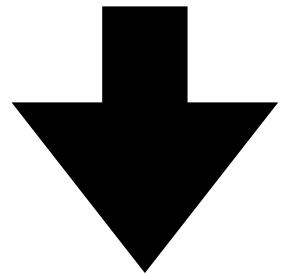


Make/apply an analysis pipeline

Me everyday

What can I do
with my LRS data?

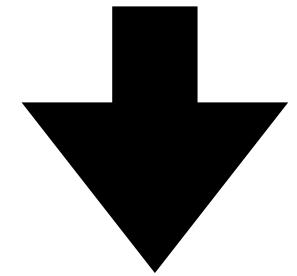
How do I visualize
my results?



Make/apply an analysis pipeline

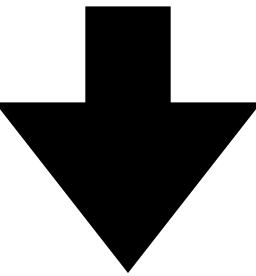
Me everyday

What can I do
with my LRS data?



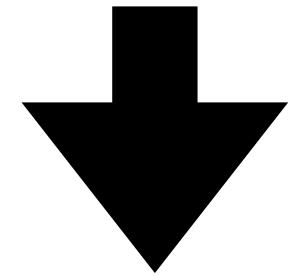
Make/apply an analysis pipeline

How do I visualize
my results?



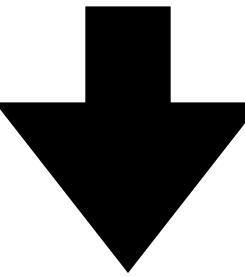
Me everyday

What can I do
with my LRS data?



Make/apply an analysis pipeline

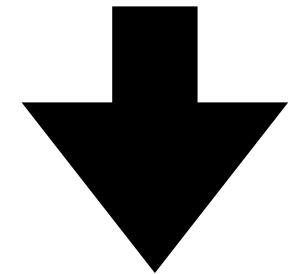
How do I visualize
my results?



How do I make the
results broadly
accessible?

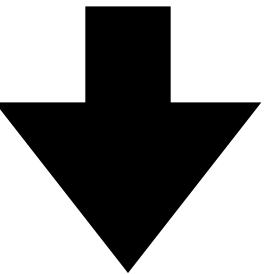
Me everyday

What can I do
with my LRS data?

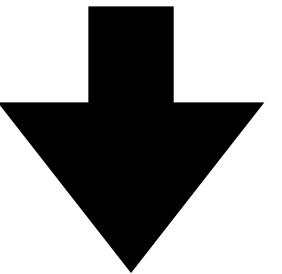


Make/apply an analysis pipeline

How do I visualize
my results?

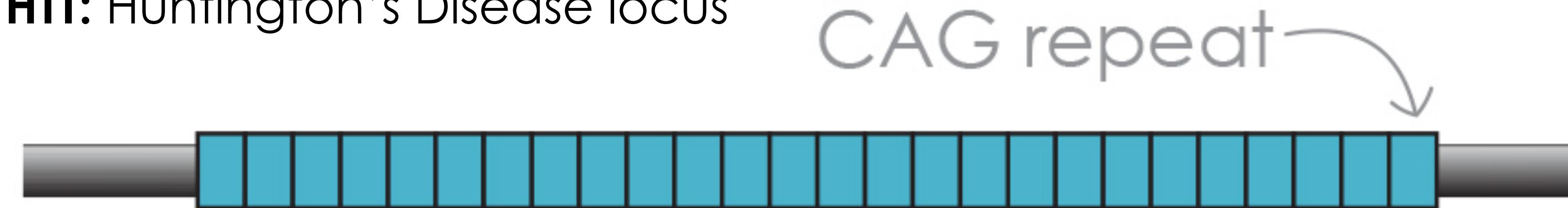


How do I make the
results broadly
accessible?



Genotype repeat size at STR loci using long reads

HTT: Huntington's Disease locus



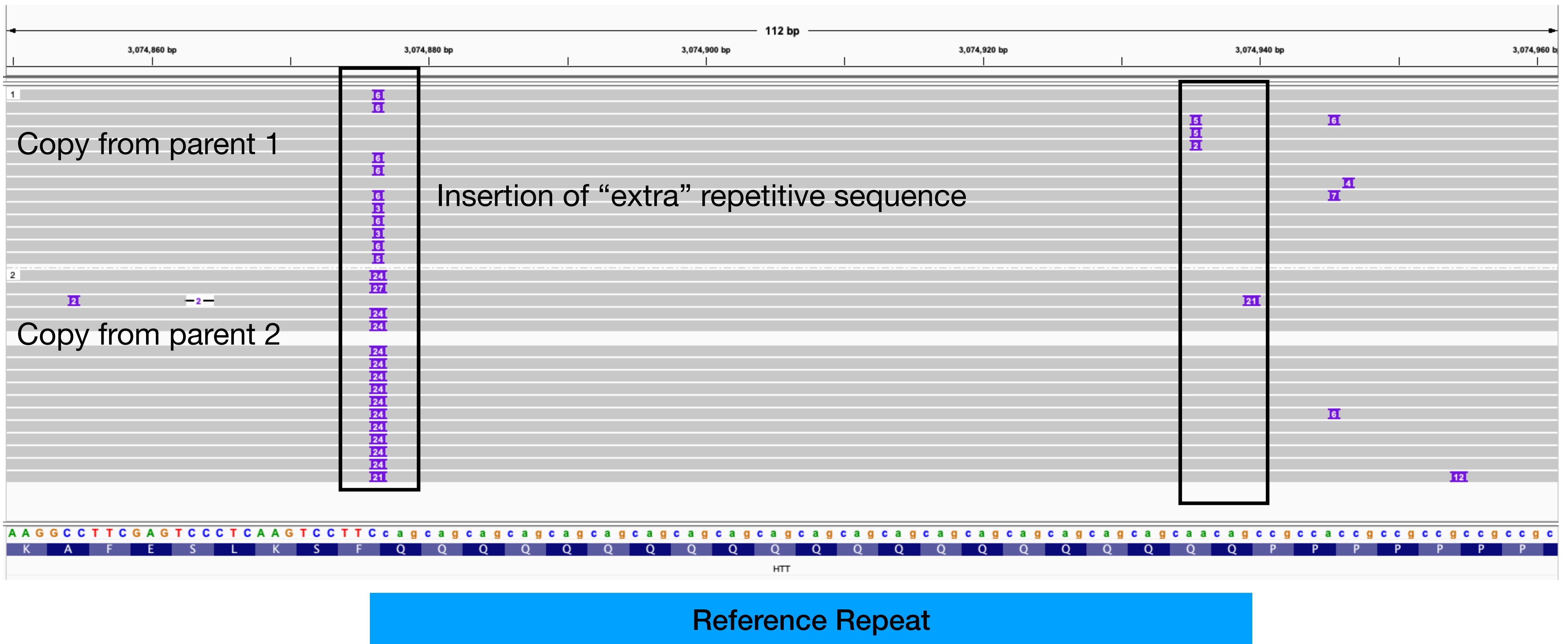
Short reads



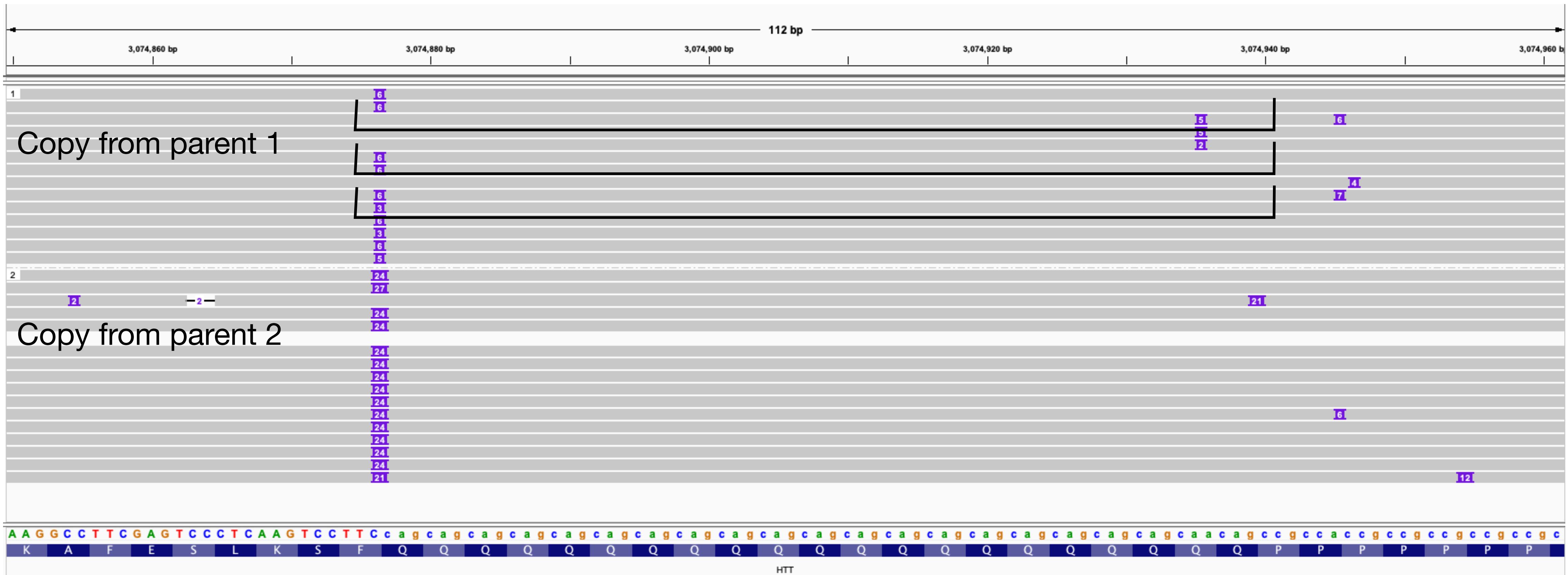
Long reads



What this looks like in IGV



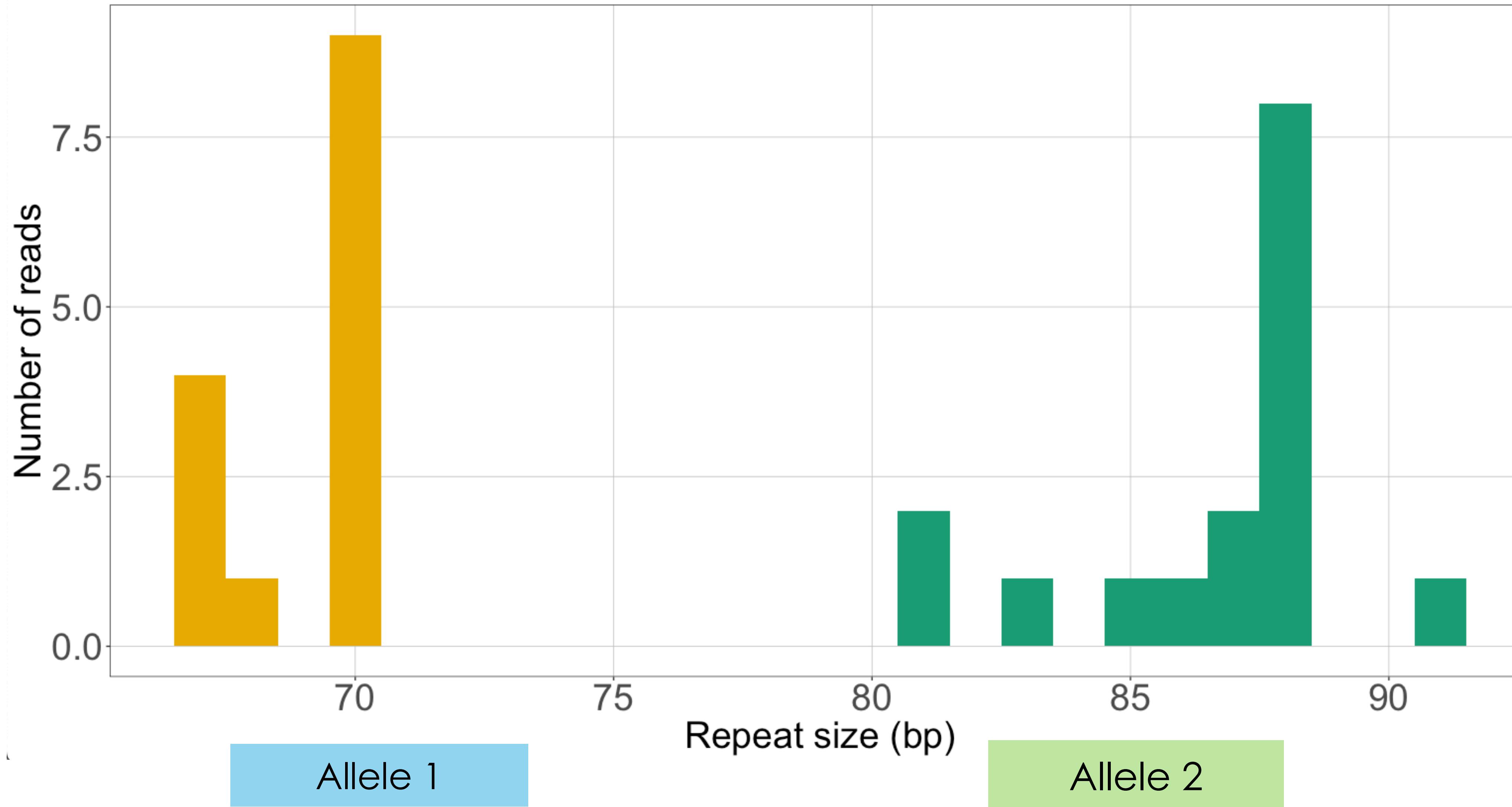
Measure the sequence length for each read



Reference Repeat

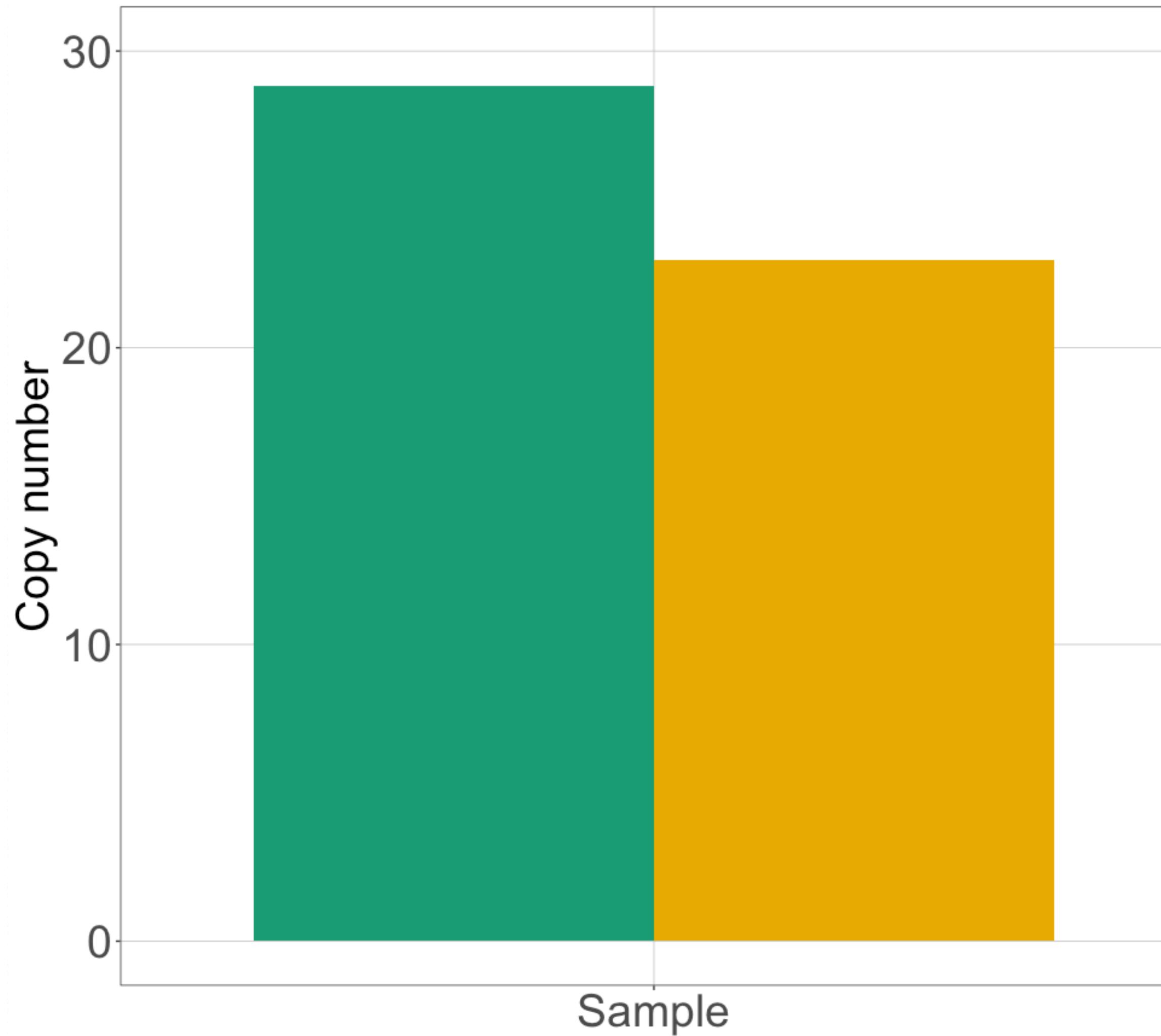
Bimodal distribution of repeat sizes for reads for one sample

HTT: Huntington's Disease locus

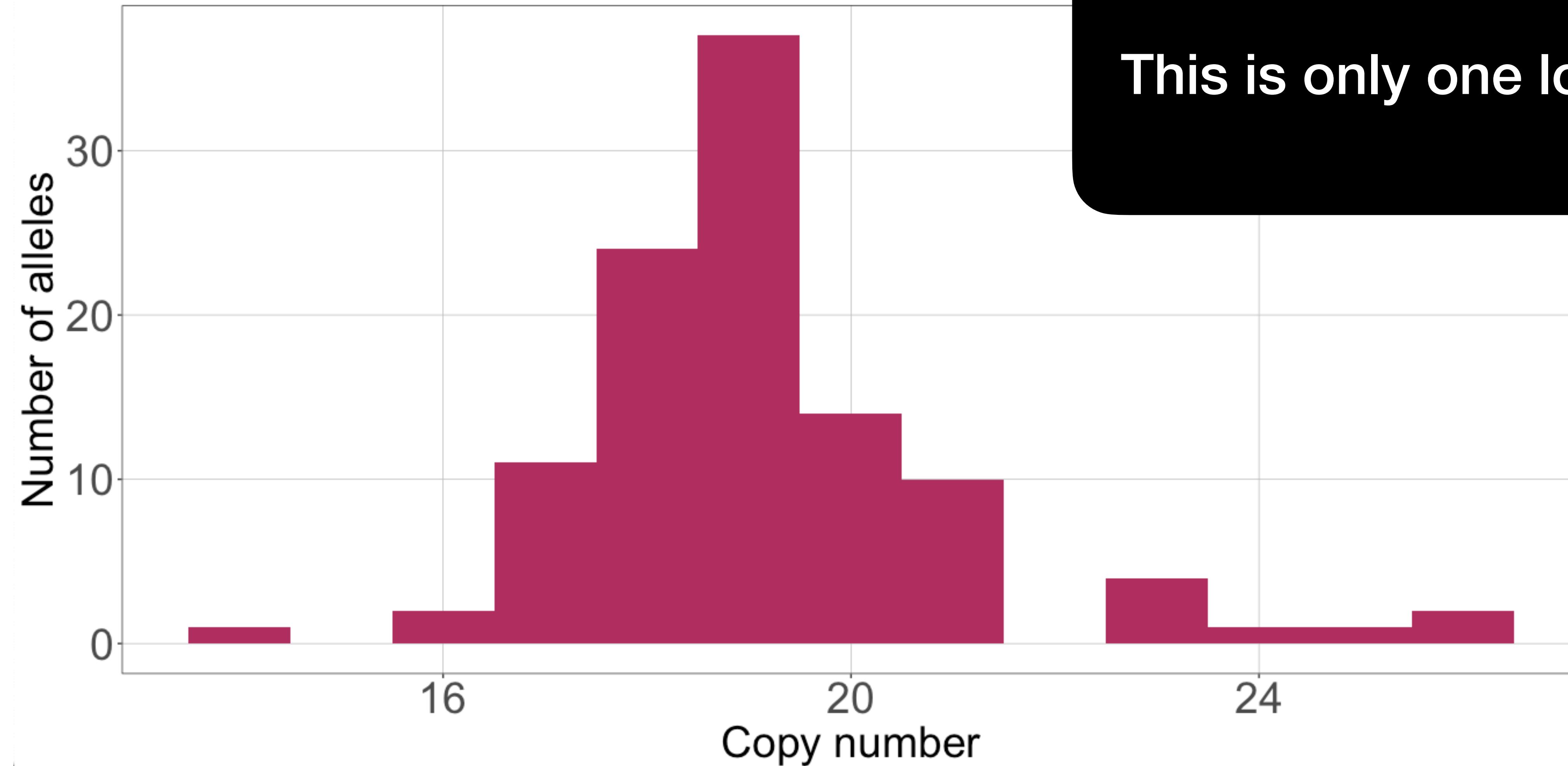


Calculate copy number estimate for each allele

HTT: Huntington's Disease locus

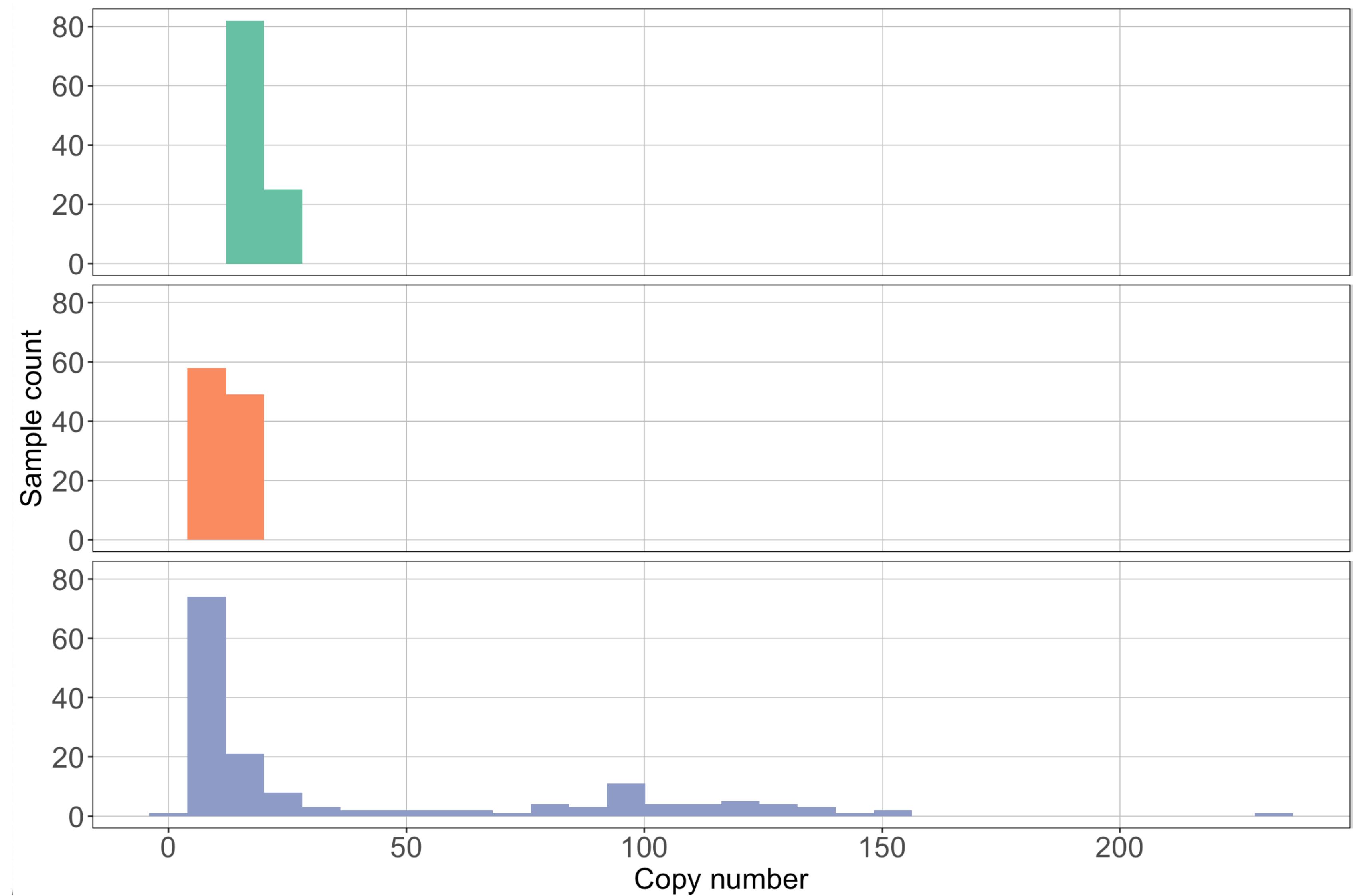


All copy number alleles at HTT locus

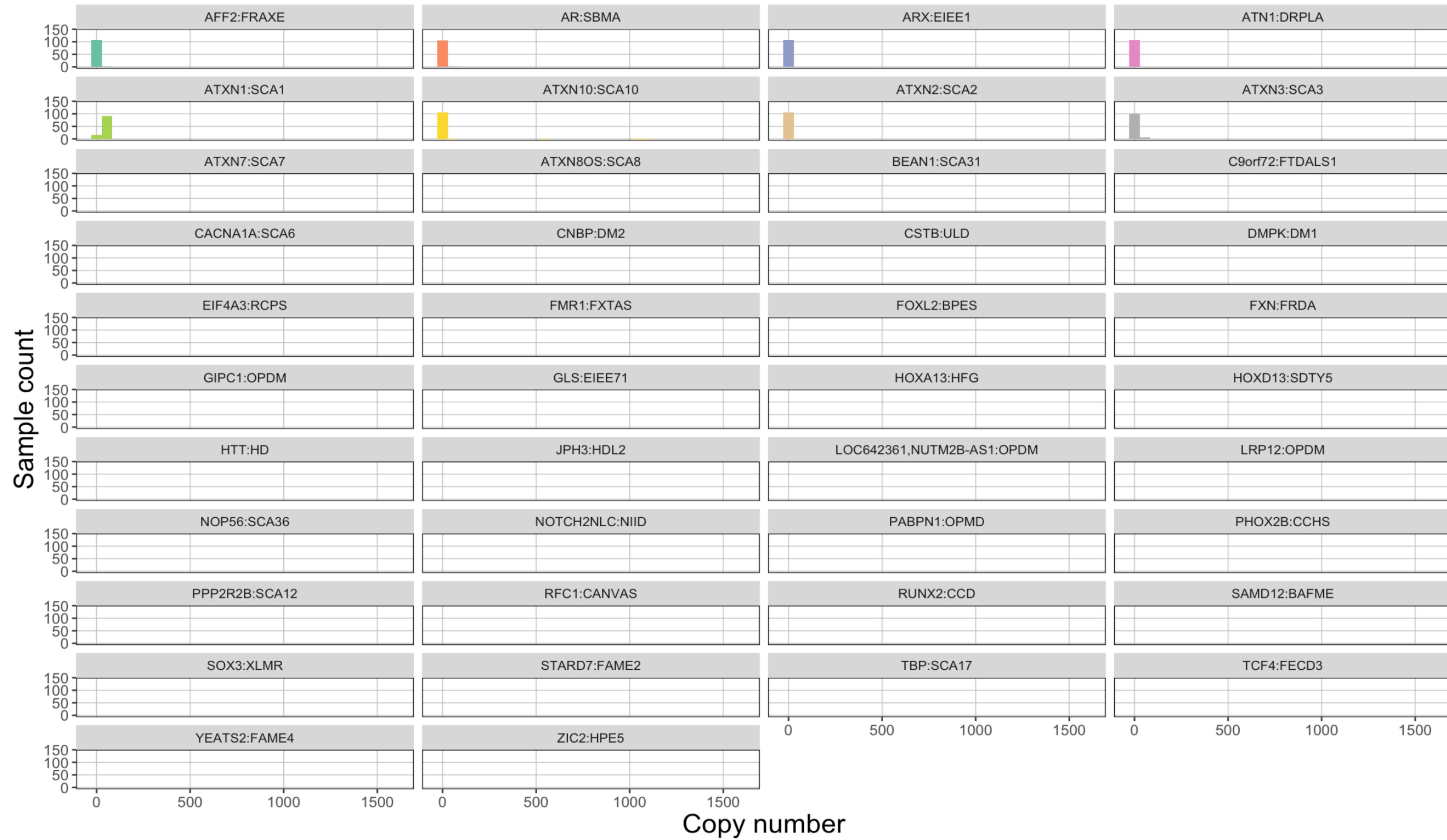


This is only one locus

Look at multiple sites with faceting



I have thousands of sites in my dataset



**Shiny: A package for building
interactive web apps
from your data**

1000G ONT Data Explorer (Beta)

Tandem Repeat Copy Numbers (Beta)

Methylation Plots (Beta) Acknowledgements

Nothing is final, adjustments ongoing

Tandem Repeat Copy Numbers

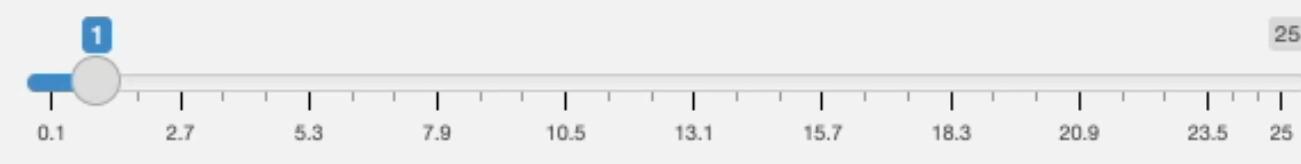
This is the Tandem Repeat Copy Number App. There is the Copy Number Distribution tab and the Two Alleles tab. Currently over 40 disease-causing tandem repeat loci can be viewed. Soon it will be over 500,000 simple repeat loci. On the Copy Number Distribution tab, select the desired repeat according to chromosome and adjust the binwidth to your liking. Click on a histogram bin to view the samples included. The Two Alleles tab shows the copy numbers for both alleles from samples with 2. If you navigate away from the Copy Number Distribution Tab and return, please re-select your locus of interest to re-configure the app. Copy numbers are estimated with straglr (Chiu et al., 2021).

Select Chromosome:

Select Known Tandem Repeat:

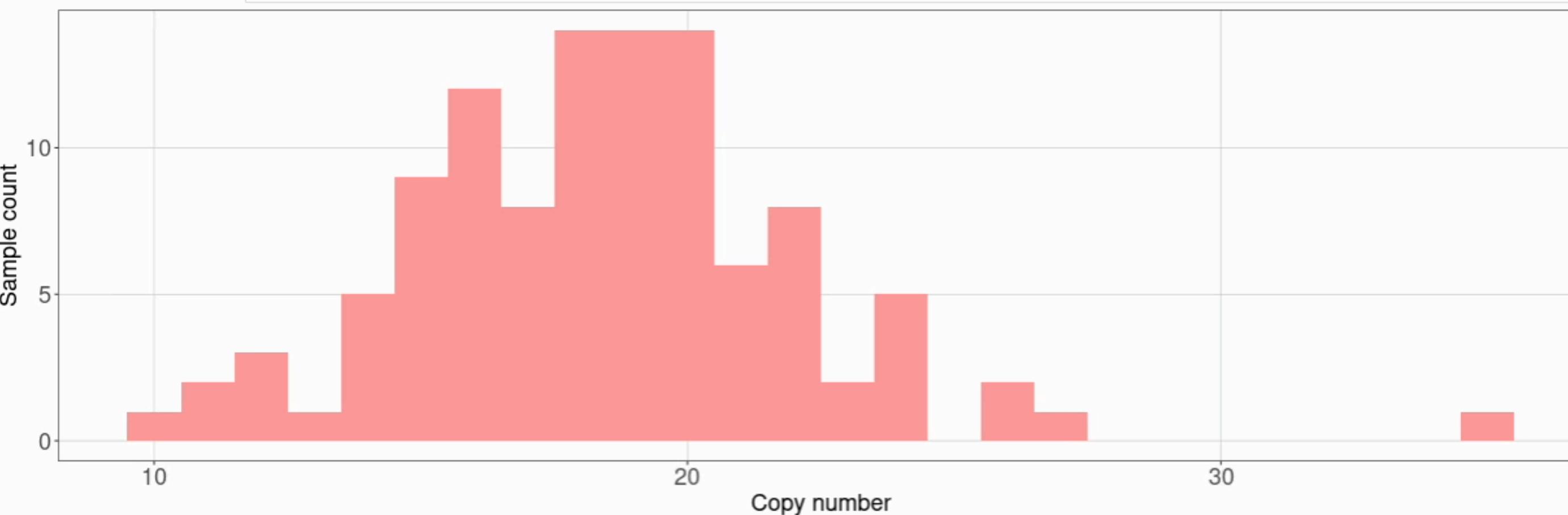
Select Repeat Motif

Binwidth:



Copy Number Distribution

Two Alleles



Under the hood

chr	start	end	repeat_unit	size	copy_number	support	allele	sample	allele_count	motif	gene	component
chr4	3074876	3074939	CAG	54.0	18.0	55	A1	GM18501	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	60.0	20.0	51	A1	GM18519	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	49.5	16.5	52	A1	GM18856	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	54.9	18.3	70	A1	GM18861	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	48.0	16.0	50	A1	GM18864	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.0	17.0	68	A1	GM18865	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	55.5	18.5	63	A1	GM18871	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.6	17.2	64	A1	GM18876	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	57.0	19.0	37	A1	GM18924	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.6	17.2	40	A1	GM19035	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	54.9	18.3	53	A1	GM19041	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	63.3	21.1	53	A1	GM19308	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	60.9	20.3	70	A1	GM19312	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	58.5	19.5	43	A1	GM19462	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	53.7	17.9	49	A1	GM19471	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	61.2	20.4	54	A1	GM19473	1	CAG	HTT:HD	coding

Under the hood

chr	start	end	repeat_unit	size	copy_number	support	allele	sample	allele_count	motif	gene	component
chr4	3074876	3074939	CAG	54.0	18.0	55	A1	GM18501	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	60.0	20.0	51	A1	GM18519	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	49.5	16.5	52	A1	GM18856	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	54.9	18.3	70	A1	GM18861	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	48.0	16.0	50	A1	GM18864	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.0	17.0	68	A1	GM18865	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	55.5	18.5	63	A1	GM18871	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.6	17.2	64	A1	GM18876	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	57.0	19.0	37	A1	GM18924	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	51.6	17.2	40	A1	GM19035	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	54.9	18.3	53	A1	GM19041	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	63.3	21.1	53	A1	GM19308	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	60.9	20.3	70	A1	GM19312	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	58.5	19.5	43	A1	GM19462	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	53.7	17.9	49	A1	GM19471	1	CAG	HTT:HD	coding
chr4	3074876	3074939	CAG	61.2	20.4	54	A1	GM19473	1	CAG	HTT:HD	coding

Browsable variant information

Simple ClinVar

Home Results About

Simple ClinVar

The content on this website is based on [ClinVar](#) database version July 14, 2021

Type 'clinvar', a 'disease term', a 'gene name' or a 'variant' in HGVS format

Submit

Simple ClinVar was developed to provide gene- and disease-wise summary statistic based on all available genetic variants from ClinVar. How many missense variants are associated to heart disease? What are the top 10 genes mutated in Alzheimer? Does CDKL5 have pathogenic mutations? If so, where? Simple ClinVar is able to answer these questions and more, in a matter of seconds.

 powered by
NCBI

Browsable RNA-Seq data

Discovery and analysis of the C. elegans Neuronal Gene Expression Network -- CeNGEN

CeNGENApp Help and Documentation

Gene expression by cell type

Find markers based on percentage of expression

Enriched Genes by cell type

Find Differential Expression between Cell Types

Heatmaps of gene expression

Find all genes expressed in a given cell type, or all cell types expressing a given gene (or group of genes).

Select one of four thresholds for expression:

1 (least stringent) to 4 (most stringent) or select unfiltered data

Choose All Cells Unfiltered to query the entire unfiltered dataset, including non-neuronal cells

Select cell type

ADA

Select threshold

2

Expressed genes

Type gene name

zig-4

Select threshold

2

Which cell types

Query multiple genes for download

flp*

WBGene00001447

WBGene00001448,zig-4

Select threshold

2

Download batch

WARNING: Expression values for dpy-20, cho-1, C30A5.16, eat-4, saeg-2, unc-47, unc-119, pha-1, F38B6.2, lin-15B, lin-15A, C30F8.3, gey-35, unc-54, cex-1, rol-6, unc-53 are unreliable as they have been overexpressed to generate transgenic strains.

Experimental design tools

HTA Dilutions App

Setup

HTA Version:

v2 (96h) v3 (48h)

Food type:

Lysate HB101 (OD 100)

Original dose response?

Free serial dose response?

Dilution factor dose response?

Minimum volume to pipette:

0.5

Number of drugs:

1

Setup Complete

setup dilutions

Not sure what concentration your drug is at? Check out this [Google Sheet](#)

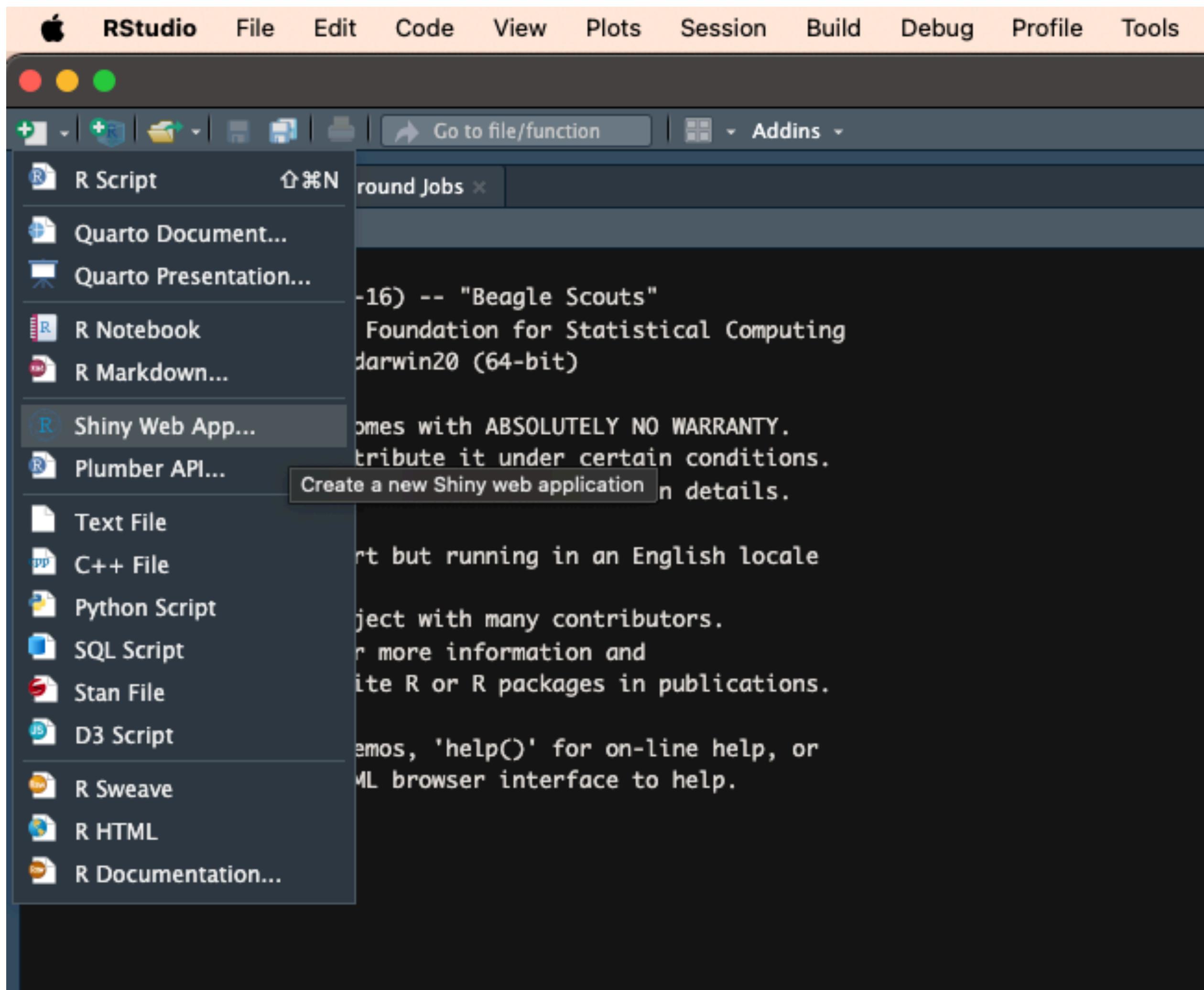
Calculate

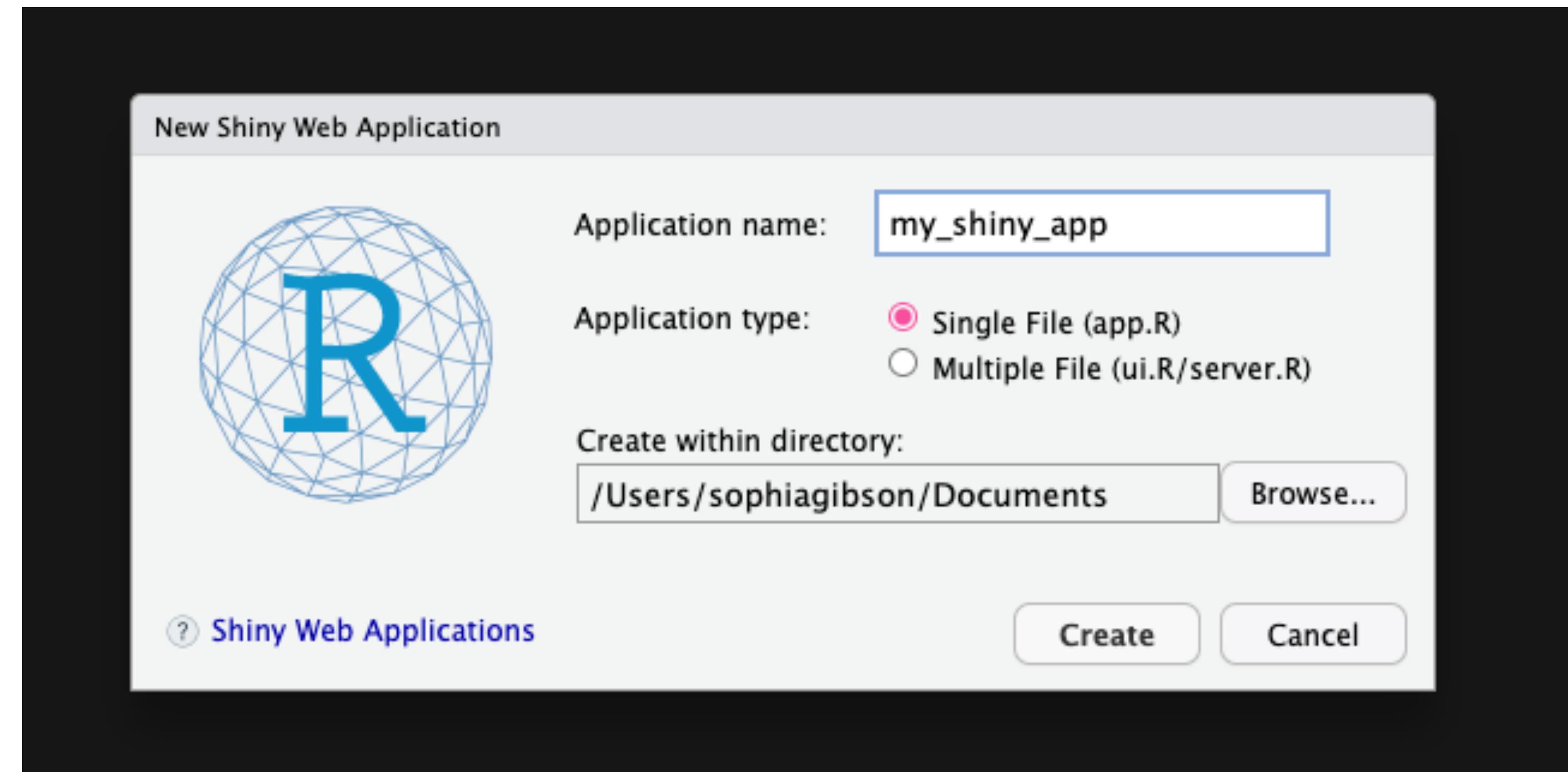


Dr. Kathryn Evans

How to build a shiny app

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





The screenshot shows the RStudio interface with the file `app.R` open. The code defines a Shiny web application for visualizing the Old Faithful Geyser Data.

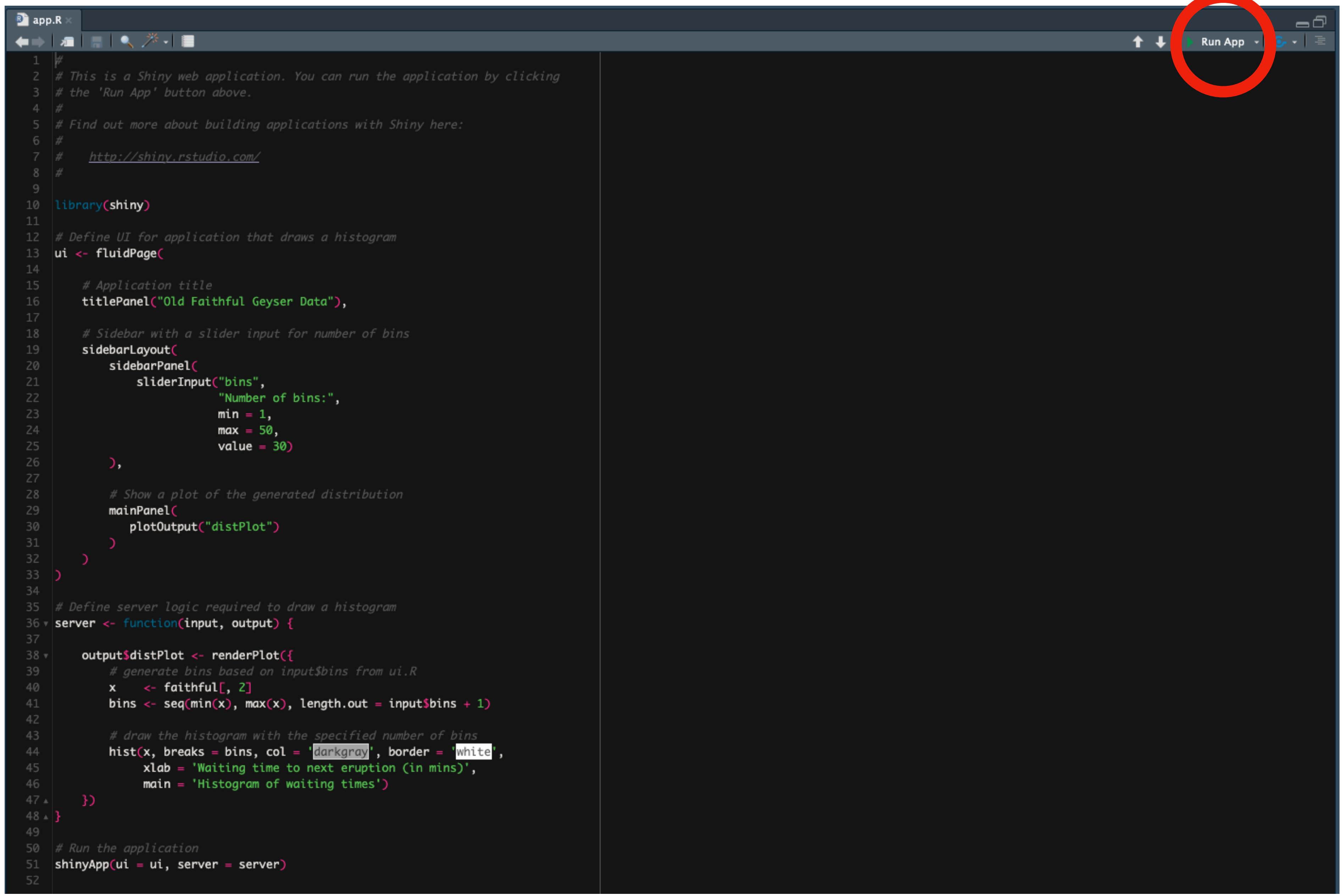
```
1 # This is a Shiny web application. You can run the application by clicking
2 # the 'Run App' button above.
3 #
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 #     http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
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
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               xlab = 'Waiting time to next eruption (in mins)',
46               main = 'Histogram of waiting times')
47     })
48 }
49
50 # Run the application
51 shinyApp(ui = ui, server = server)
```

```
app.R
```

```
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  
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               xlab = 'Waiting time to next eruption (in mins)',  
46               main = 'Histogram of waiting times')  
47     })  
48 }  
49  
50 # Run the application  
51 shinyApp(ui = ui, server = server)
```

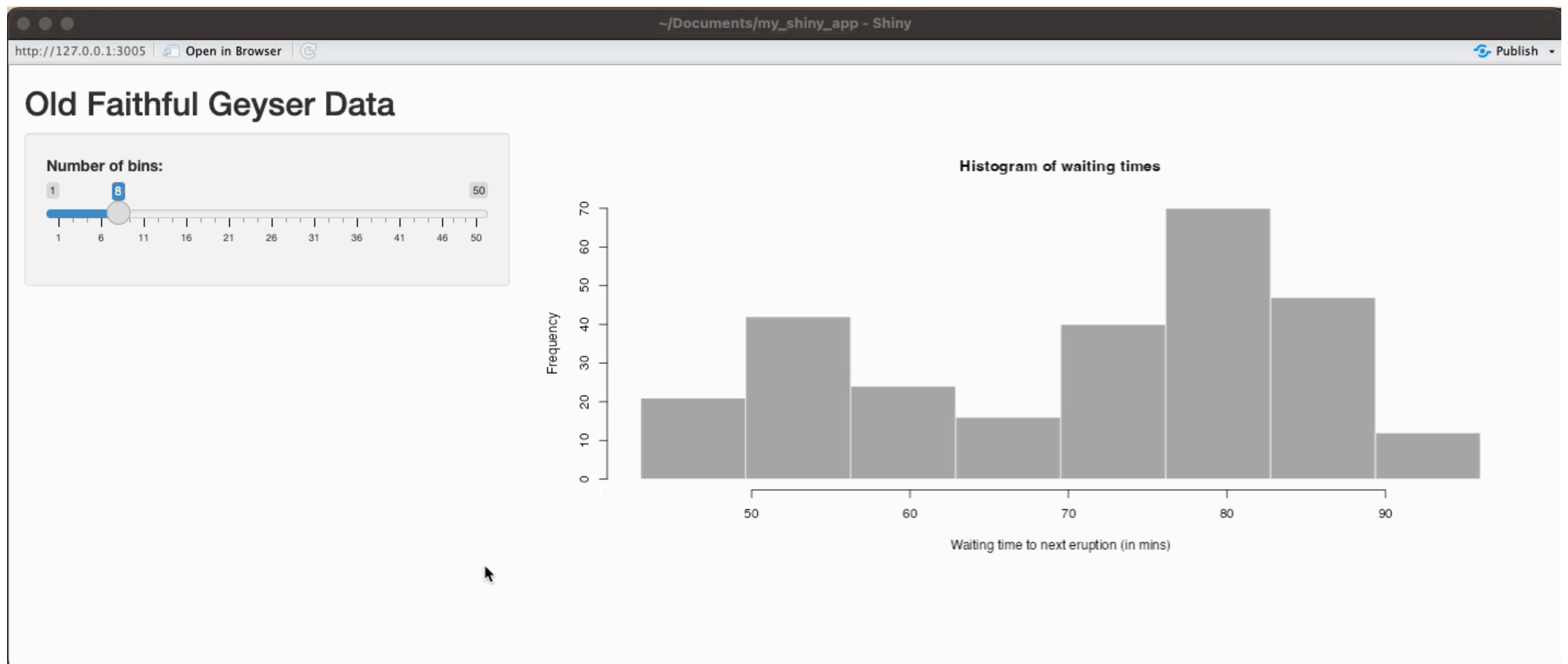
UI

Server



The screenshot shows the RStudio interface with the code editor open to an R script named 'app.R'. The script contains code for building a Shiny web application. A red circle highlights the 'Run App' button in the top right corner of the toolbar.

```
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  
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             xlab = 'Waiting time to next eruption (in mins)',  
46             main = 'Histogram of waiting times')  
47     })  
48 }  
49  
50 # Run the application  
51 shinyApp(ui = ui, server = server)
```



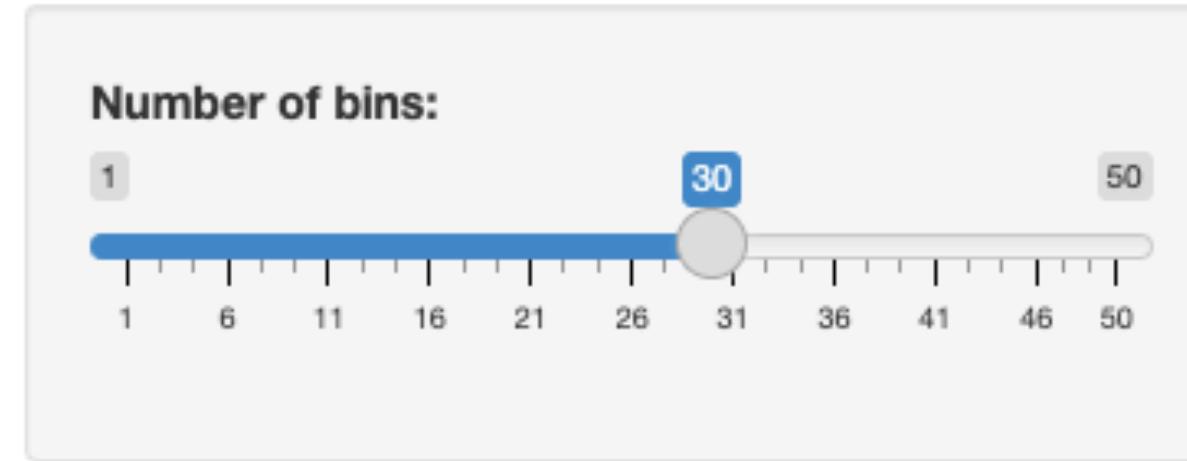
Basic shiny app layout

Title → Old Faithful Geyser Data

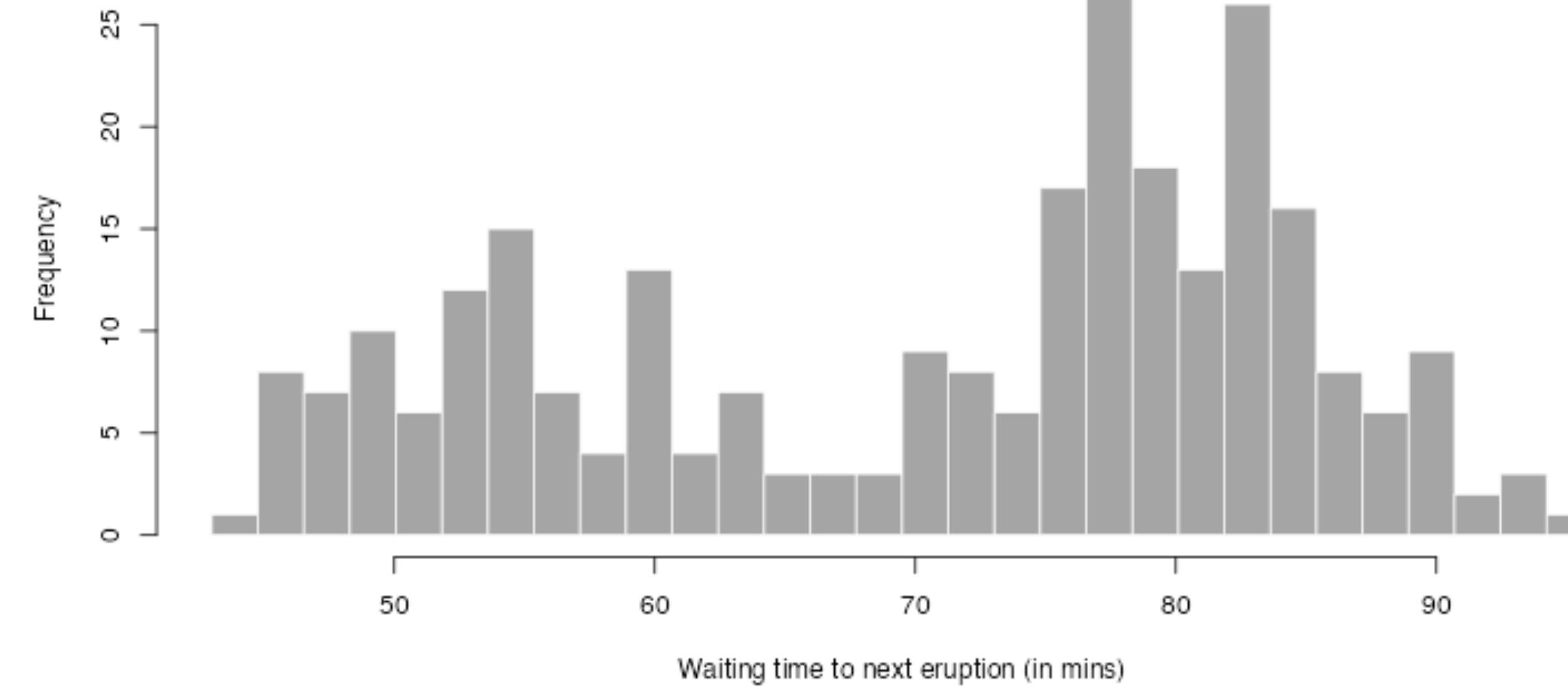
Sidebar

Number of bins:

1 30 50



Histogram of waiting times



Main panel

UI - What does my app look like?

```
fluidPage(..., title = NULL, theme = NULL, lang =  
NULL)
```

Create a page with fluid layout

Functions for creating fluid page layouts. A fluid page layout consists of rows which in turn include columns. Rows exist for the purpose of making sure their elements appear on the same line (if the browser has adequate width). Columns exist for the purpose of defining how much horizontal space within a 12-unit wide grid it's elements should occupy. Fluid pages scale their components in realtime to fill all available browser width.

```
9  
10 library(shiny)  
11  
12 # Define UI for application that draws a histogram  
13 ui <- fluidPage(  
14     # Application title  
15     titlePanel("Old Faithful Geyser Data"),  
16  
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     ),  
27  
28     # Show a plot of the generated distribution  
29     mainPanel(  
30         plotOutput("distPlot")  
31     )  
32 )  
33 )  
34 )
```

UI - What does my app look like?

```
titlePanel(title, windowTitle = title)
Create a panel containing an application title.

Create a panel containing an application title.
```

```
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
```

Basic shiny app layout

Fluid page

Title → Old Faithful Geyser Data

UI - What does my app look like?

```
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         mainPanel(
29             plotOutput("distPlot")
30         )
31     )
32 )
33 )
```

Layout a sidebar and main area

Create a layout (`sidebarLayout()`) with a sidebar (`sidebarPanel()`) and main area (`mainPanel()`). The sidebar is displayed with a distinct background color and typically contains input controls. The main area occupies 2/3 of the horizontal width and typically contains outputs.

```
sliderInput(inputId, label, min, max, value, step =
  NULL, round = FALSE, ticks = TRUE, animate = FALSE,
  width = NULL, sep = ",", pre = NULL, post = NULL,
  timeFormat = NULL, timezone = NULL, dragRange =
  TRUE)
```

Slider Input Widget

Constructs a slider widget to select a number, date, or date-time from a range.

```
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         mainPanel(
29             plotOutput("distPlot")
30         )
31     )
32 )
33 )
```

Basic shiny app layout

Old Faithful Geyser Data

Number of bins:

1 30 50

1 6 11 16 21 26 31 36 41 46 50

Sidebar

The image shows a basic Shiny application layout. On the left side, there is a sidebar labeled "Sidebar". Inside the sidebar, there is a title "Old Faithful Geyser Data" and a slider input labeled "Number of bins". The slider has a range from 1 to 50, with major ticks at 1, 6, 11, 16, 21, 26, 31, 36, 41, 46, and 50. The value is currently set to 30. The main content area is currently empty.

UI - What does my app look like?

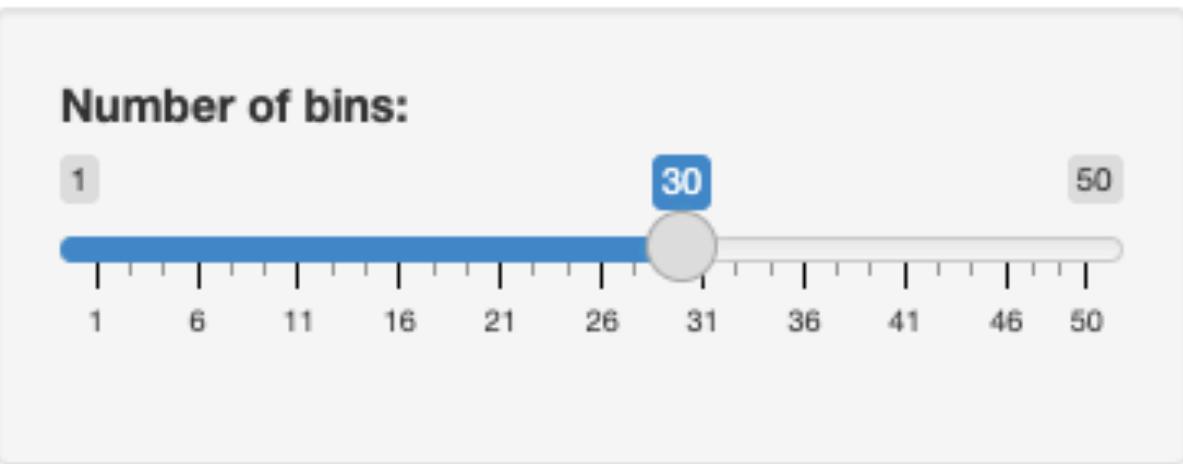
```
mainPanel(..., width = 8)  
Layout a sidebar and main area
```

Create a layout (`sidebarLayout()`) with a sidebar (`sidebarPanel()`) and main area (`mainPanel()`). The sidebar is displayed with a distinct background color and typically contains input controls. The main area occupies 2/3 of the horizontal width and typically contains outputs.

```
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 )
```

Basic shiny app layout

Old Faithful Geyser Data



“distPlot”

Main panel

Server - what am I plotting from my data?

```
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           xlab = 'Waiting time to next eruption (in mins)',
46           main = 'Histogram of waiting times')
47   })
48 }
49
50 # Run the application
51 shinyApp(ui = ui, server = server)
52
```

Server - what am I plotting from my data?

```
renderPlot(expr, width = "auto", height = "auto", res = 72, ..., alt = NA, env = parent.frame(), quoted = FALSE, execOnResize = FALSE, outputArgs = list())
```

Plot Output

Renders a reactive plot that is suitable for assigning to an output slot.

	eruptions	waiting
1	3.600	79
2	1.800	54
3	3.333	74
4	2.283	62
5	4.533	85
6	2.883	55
7	4.700	88
8	3.600	85
9	1.950	51
10	4.350	85
11	1.833	54
12	3.917	84
13	4.200	78
14	1.750	47
15	4.700	83

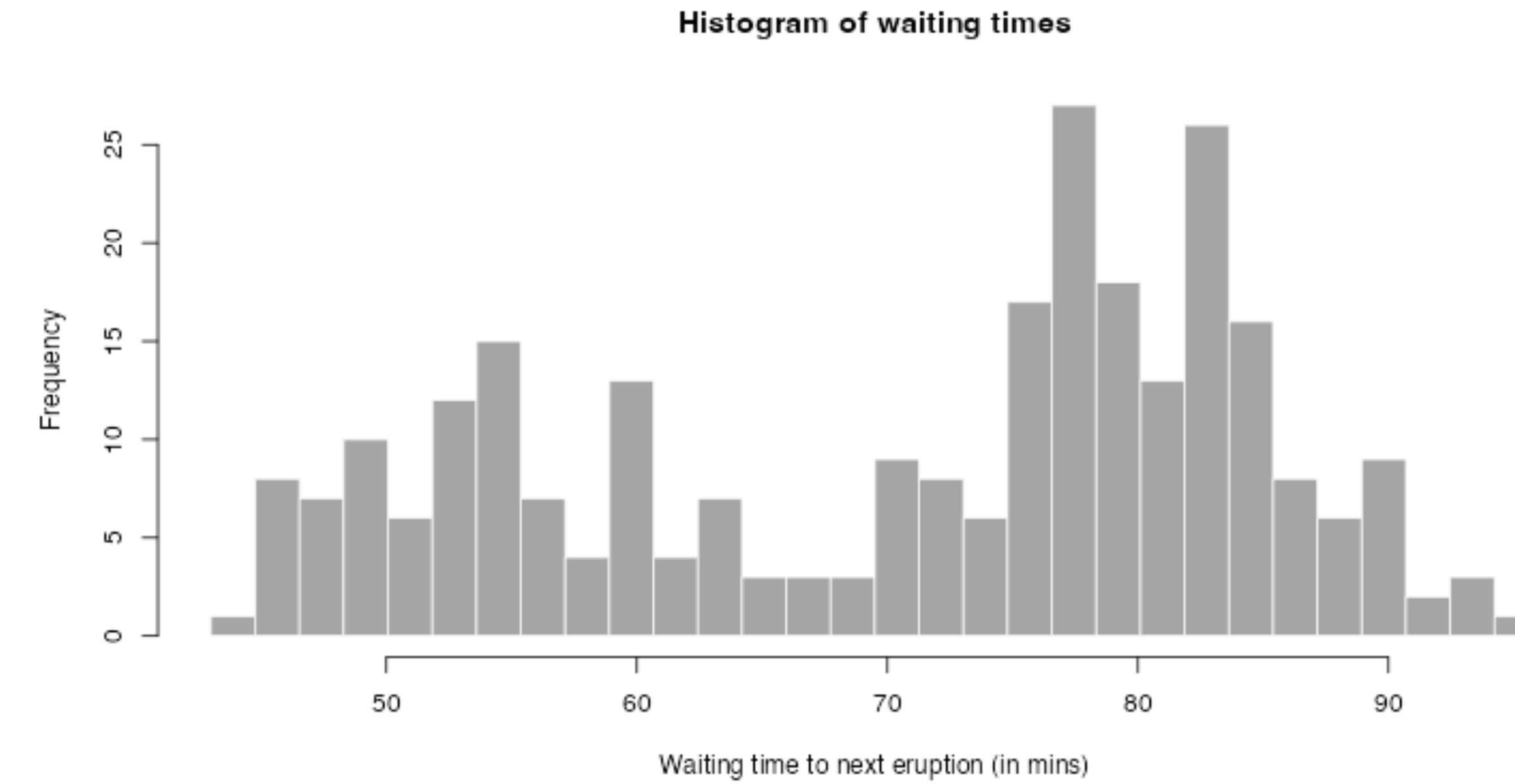
```
35 # Define server logic required to draw a histogram
36 server <- function(input, output) {
37
38   output$distPlot <- renderPlot({
39     # generate bins based on input$bins from ui.R
40     x      <- faithful[, 2]
41     bins   <- seq(min(x), max(x), length.out = input$bins + 1)
42
43     # draw the histogram with the specified number of bins
44     hist(x, breaks = bins, col = 'darkgray', border = 'white',
45           xlab = 'Waiting time to next eruption (in mins)',
46           main = 'Histogram of waiting times')
47   })
48 }
49
50 # Run the application
51 shinyApp(ui = ui, server = server)
52 }
```

Basic shiny app layout

Old Faithful Geyser Data

Number of bins:

1 6 11 16 21 26 31 36 41 46 50



Main panel

Run the application

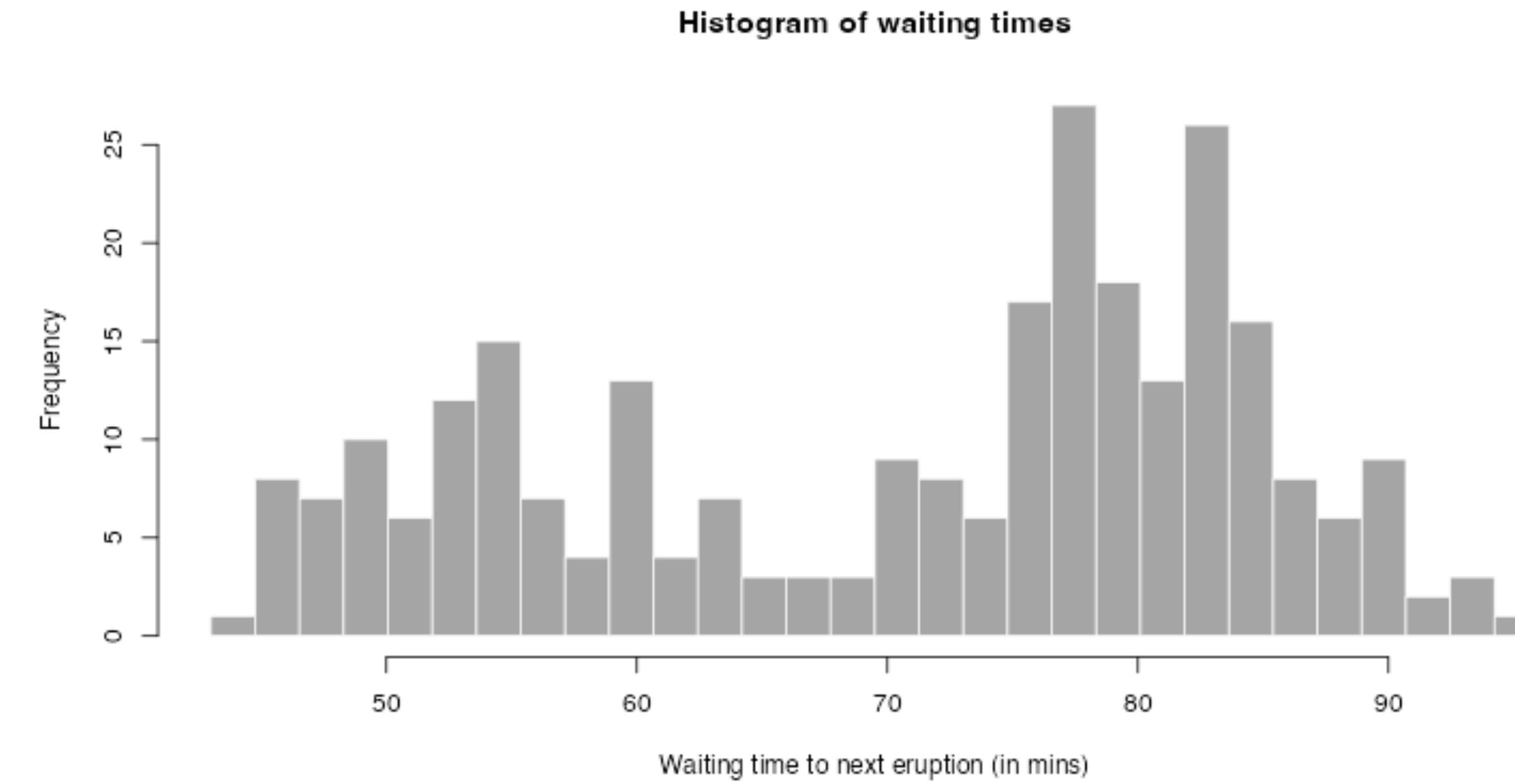
```
35 # Define server logic required to draw a histogram
36 server <- function(input, output) {
37
38   output$distPlot <- renderPlot({
39     # generate bins based on input$bins from ui.R
40     x    <- faithful[, 2]
41     bins <- seq(min(x), max(x), length.out = input$bins + 1)
42
43     # draw the histogram with the specified number of bins
44     hist(x, breaks = bins, col = 'darkgray', border = 'white',
45           xlab = 'Waiting time to next eruption (in mins)',
46           main = 'Histogram of waiting times')
47   })
48 }
49
50 # Run the application
51 shinyApp(ui = ui, server = server)
52
```

Basic shiny app layout

Old Faithful Geyser Data

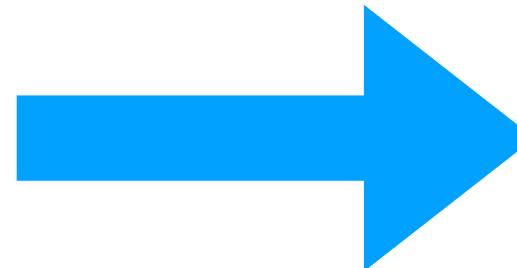
Number of bins:

1 6 11 16 21 26 31 36 41 46 50



Base R plot

Rewrite the app for tidyverse



```
library(shiny)
library(ggplot2)

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

	eruptions	waiting
1	3.600	79
2	1.800	54
3	3.333	74
4	2.283	62
5	4.533	85
6	2.883	55
7	4.700	88
8	3.600	85
9	1.950	51
10	4.350	85
11	1.833	54
12	3.917	84
13	4.200	78
14	1.750	47
15	4.700	83

```
# Define server logic required to draw a histogram
server <- function(input, output) {

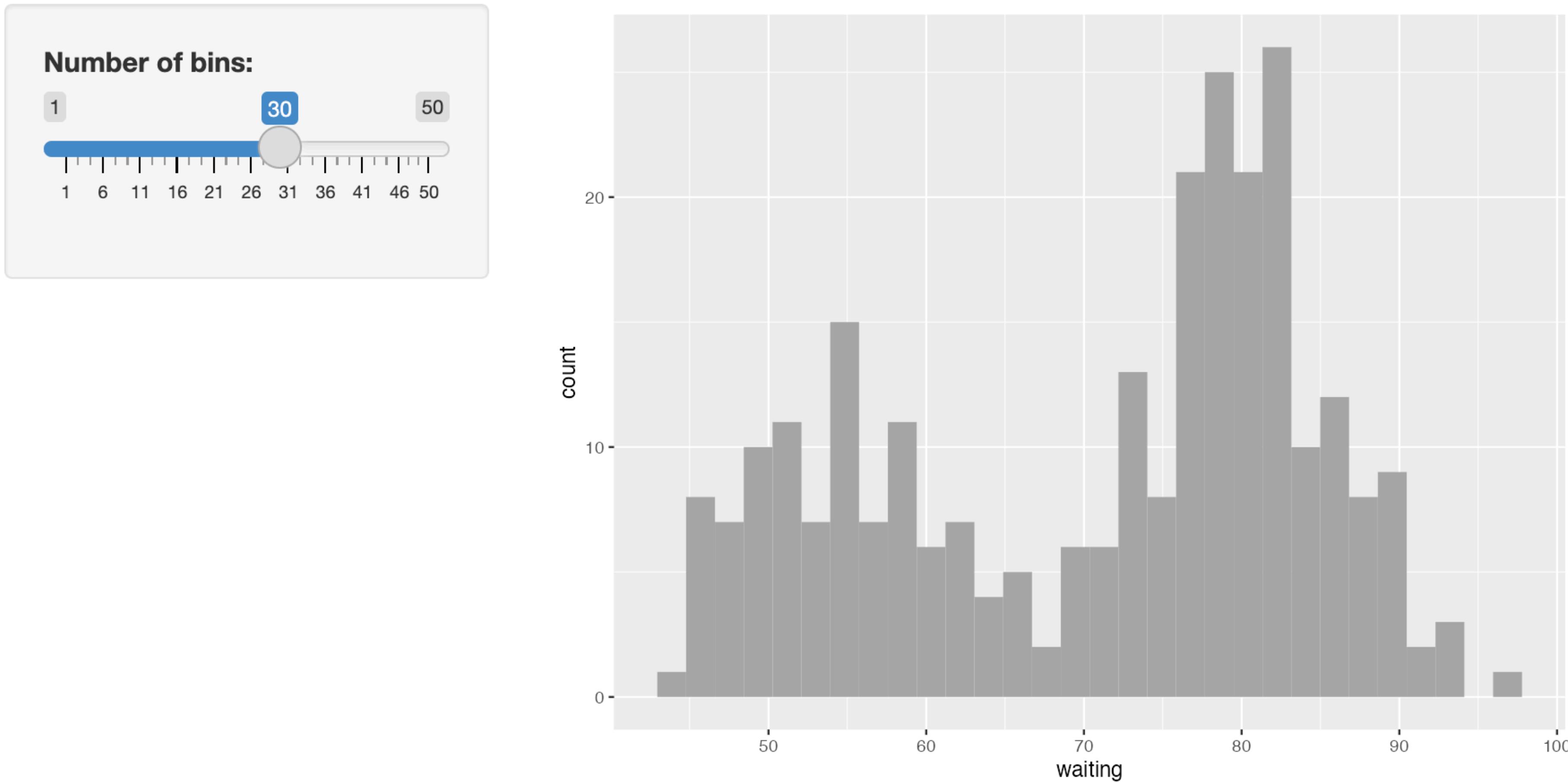
  output$distPlot <- renderPlot({

    #GGPLOT instead of base R
    ggplot(data=faithful, aes(x=waiting, stat="identity")) +
      geom_histogram(bins = input$bins, fill="darkgrey")

  })
}
```

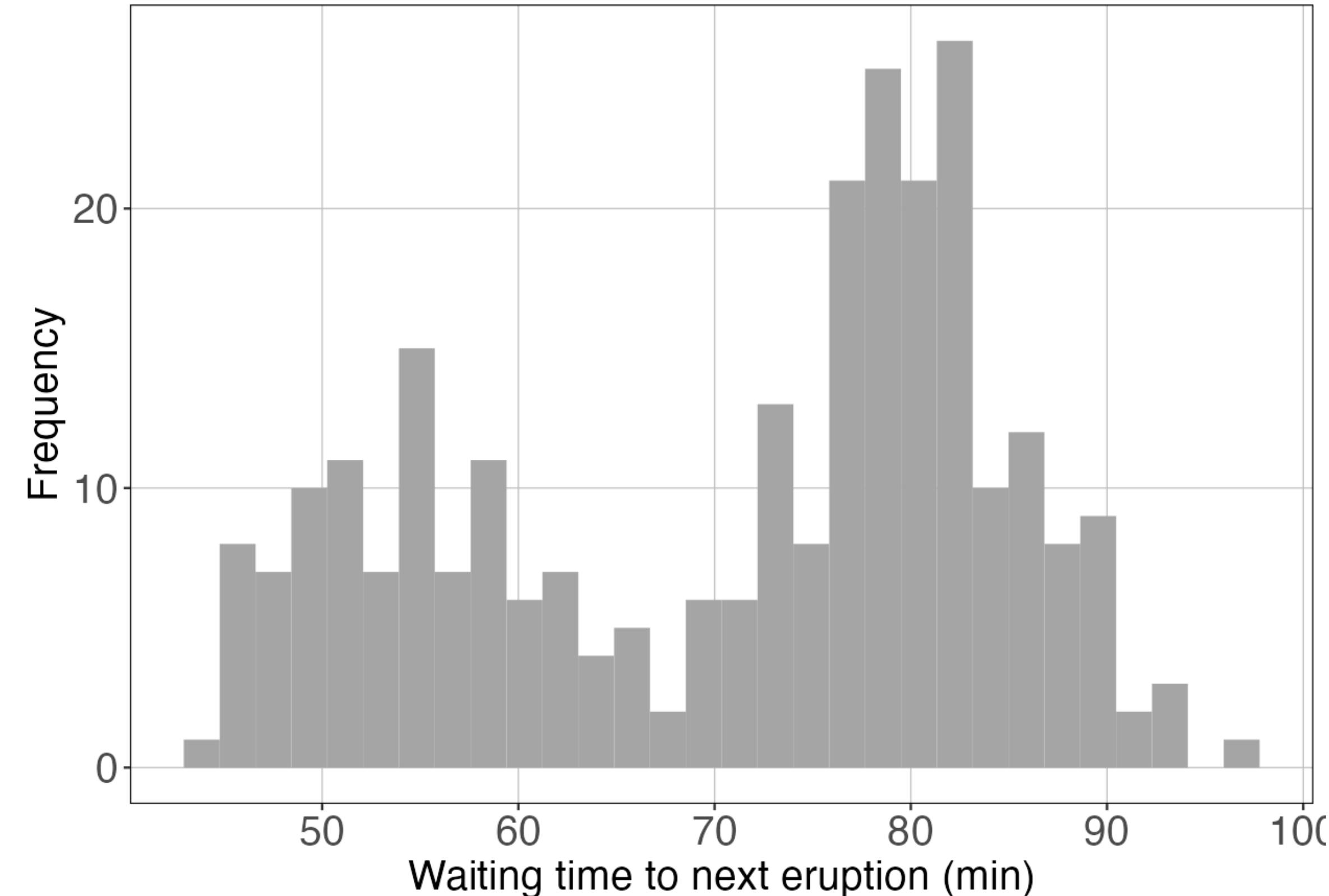
Rewrite the app for tidyverse

Old Faithful Geyser Data



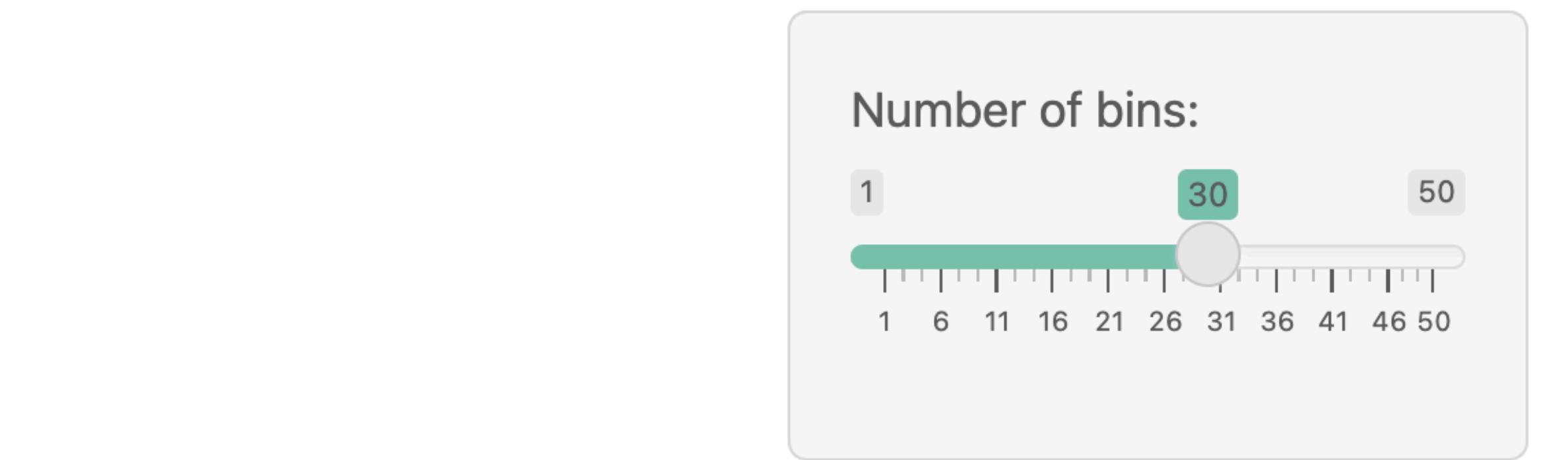
Customize the plot

```
#GGPLOT instead of base R
ggplot(data=faithful, aes(x=waiting, stat="identity")) +
  geom_histogram(bins = input$bins, fill="darkgrey") +
  xlab("Waiting time to next eruption (min)") +
  ylab("Frequency") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size=18),
        panel.background = element_rect(fill = "white"),
        panel.border = element_rect(colour = "black", fill=NA),
        panel.grid.major = element_line(size = 0.25, linetype = 'solid',
                                         colour = "grey"))
```



Add a theme

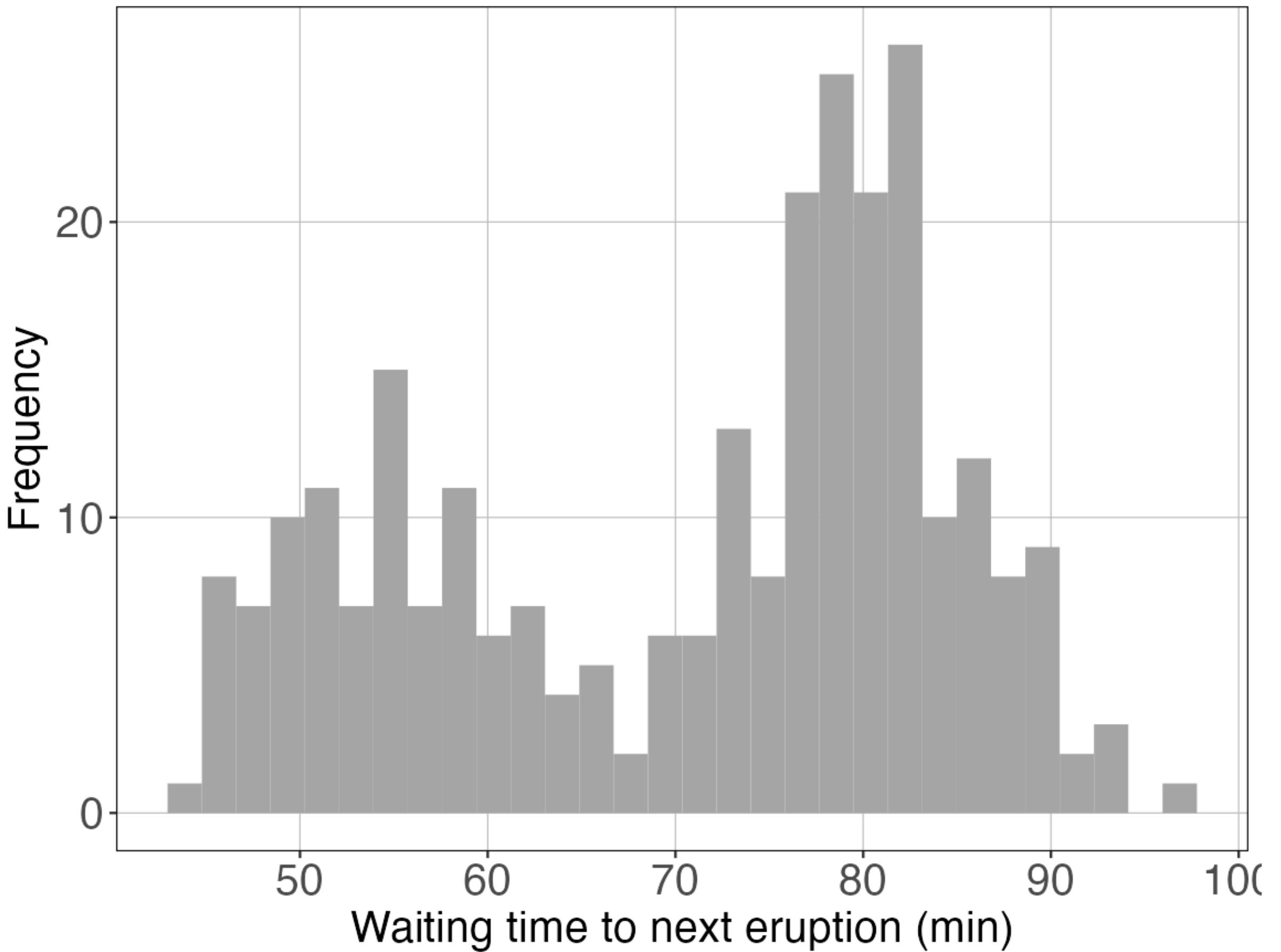
Old Faithful Geyser Data



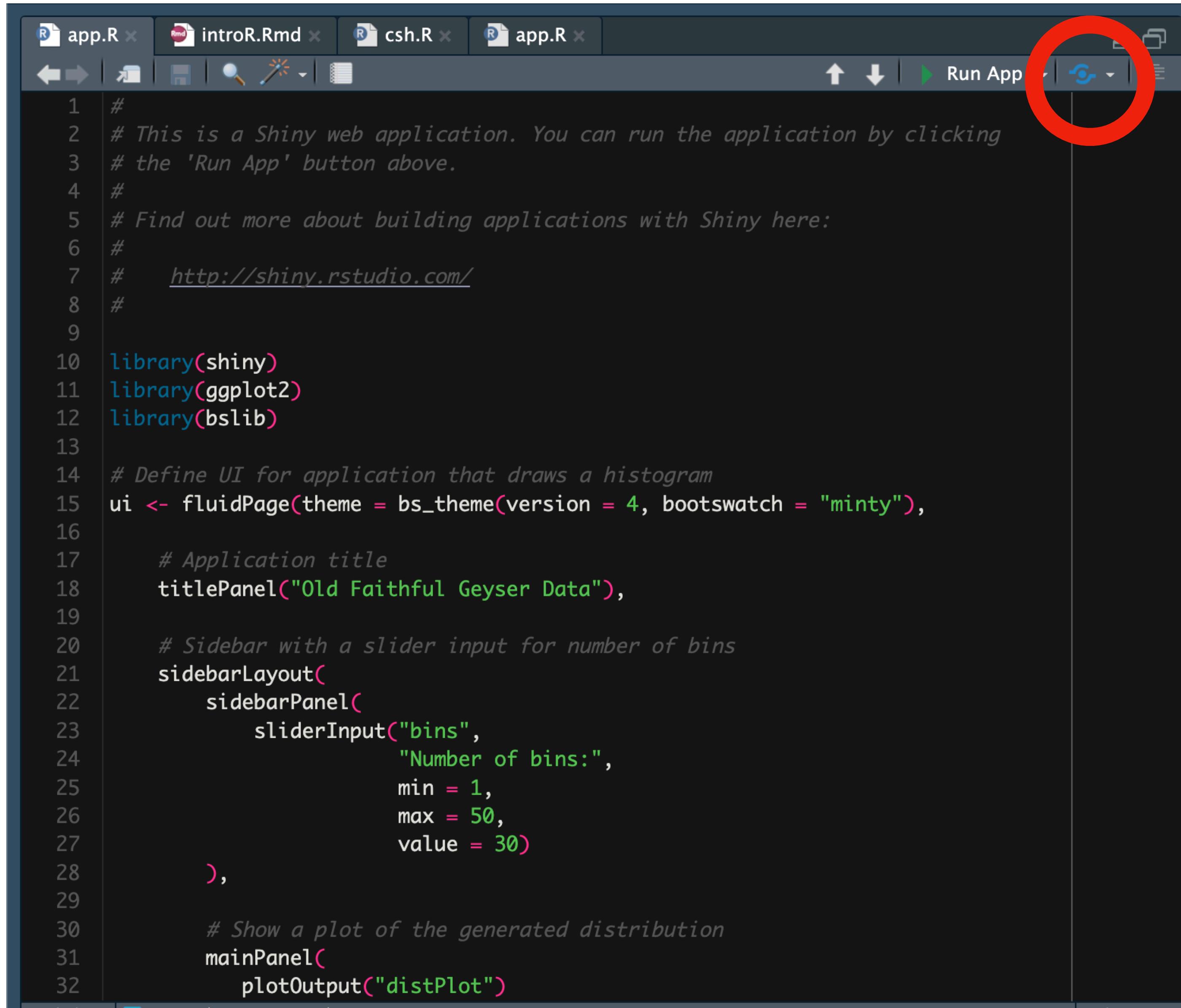
```
library(shiny)
library(ggplot2)
library(bslib)

# Define UI for application that draws a histogram
ui <- fluidPage(theme = bs_theme(version = 4, bootswatch = "minty"),

  # Application title
  titlePanel("Old Faithful Geyser Data"),
```



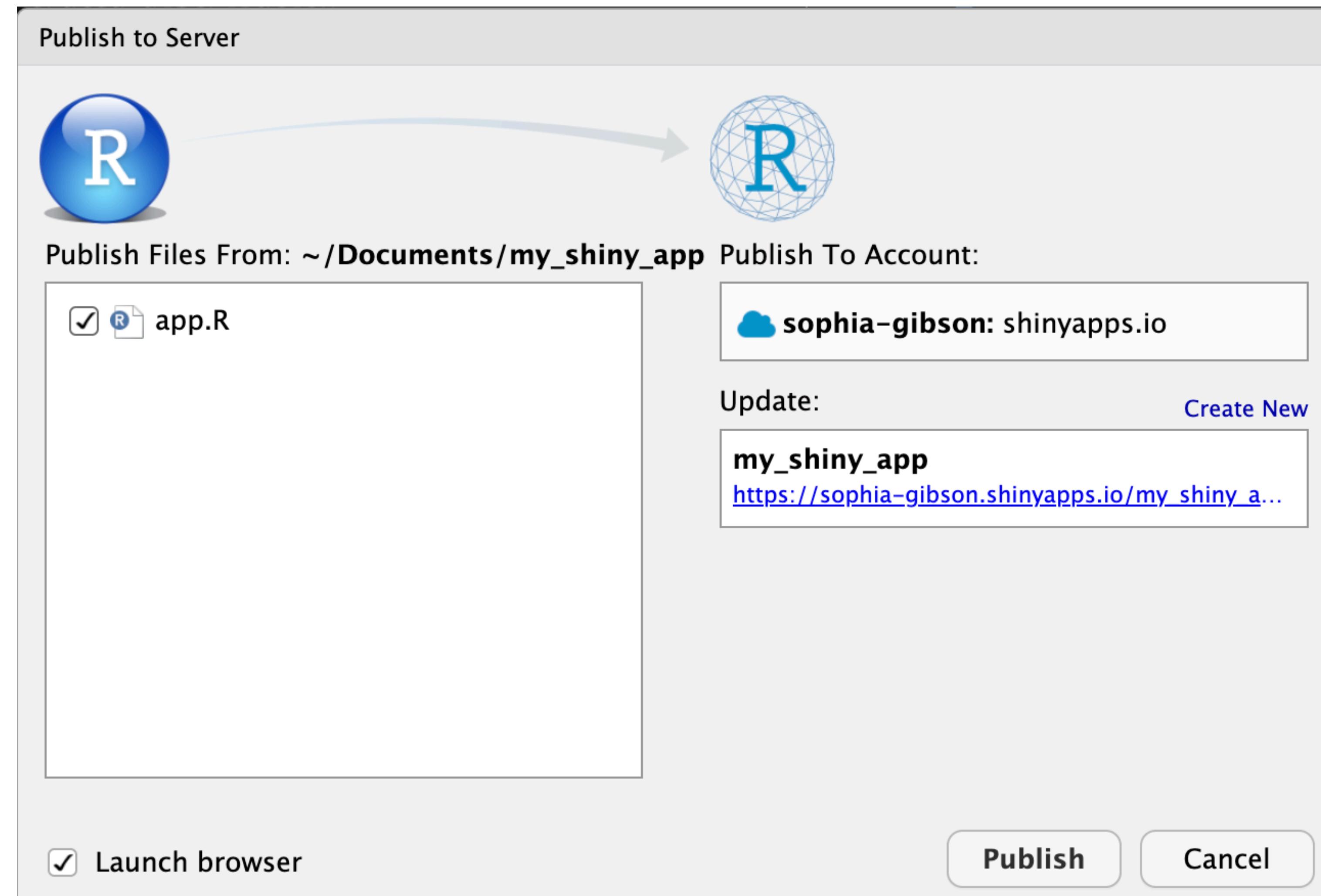
Publish my shiny app



The screenshot shows the RStudio interface with several tabs open at the top: 'app.R x', 'introR.Rmd x', 'csh.R x', and 'app.R x'. Below the tabs is a toolbar with various icons. A red circle highlights the 'Run App' button, which is the third icon from the left in the toolbar. The main area of the window displays an R script titled 'app.R'. The script contains code for building a Shiny web application. The code includes comments explaining the purpose of the application and how to run it. It also imports libraries like shiny, ggplot2, and bslib, defines a UI for a histogram, and a main panel with a plot output.

```
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 library(ggplot2)
12 library(bslib)
13
14 # Define UI for application that draws a histogram
15 ui <- fluidPage(theme = bs_theme(version = 4, bootswatch = "minty"),
16
17     # Application title
18     titlePanel("Old Faithful Geyser Data"),
19
20     # Sidebar with a slider input for number of bins
21     sidebarLayout(
22         sidebarPanel(
23             sliderInput("bins",
24                         "Number of bins:",
25                         min = 1,
26                         max = 50,
27                         value = 30)
28         ),
29
30         # Show a plot of the generated distribution
31         mainPanel(
32             plotOutput("distPlot")
```

Make a Posit account and publish to a server



My dashboard

The screenshot shows the shinyapps.io admin dashboard. The top navigation bar includes the shinyapps.io logo, a search bar, and account information for Sophia Gibson. The left sidebar has links for Dashboard, Applications, and Account. The main content area features a "WHAT'S NEW?" section and a "RECENT APPLICATIONS" table.

WHAT'S NEW?

RECENT APPLICATIONS

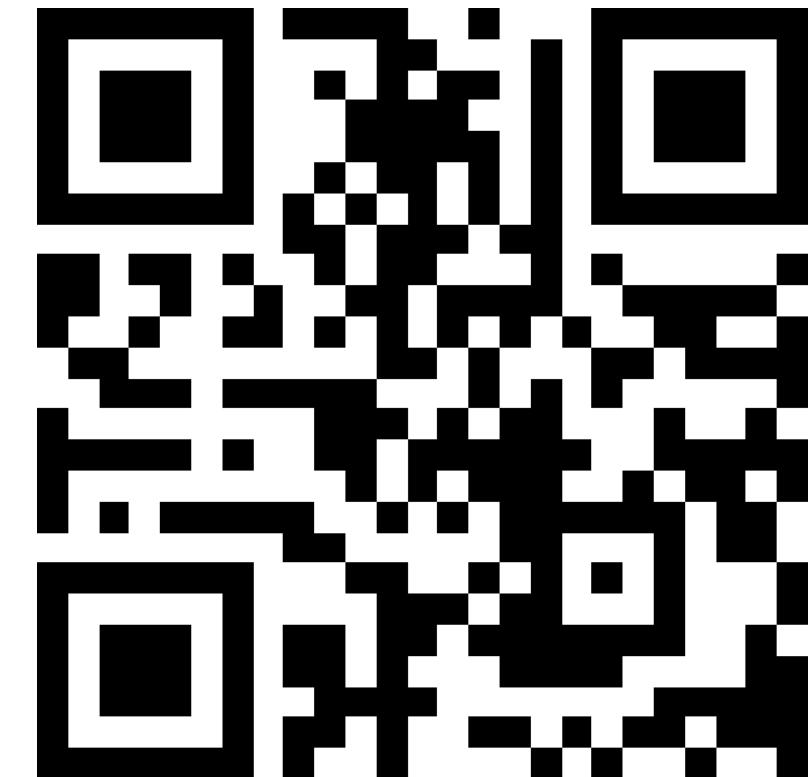
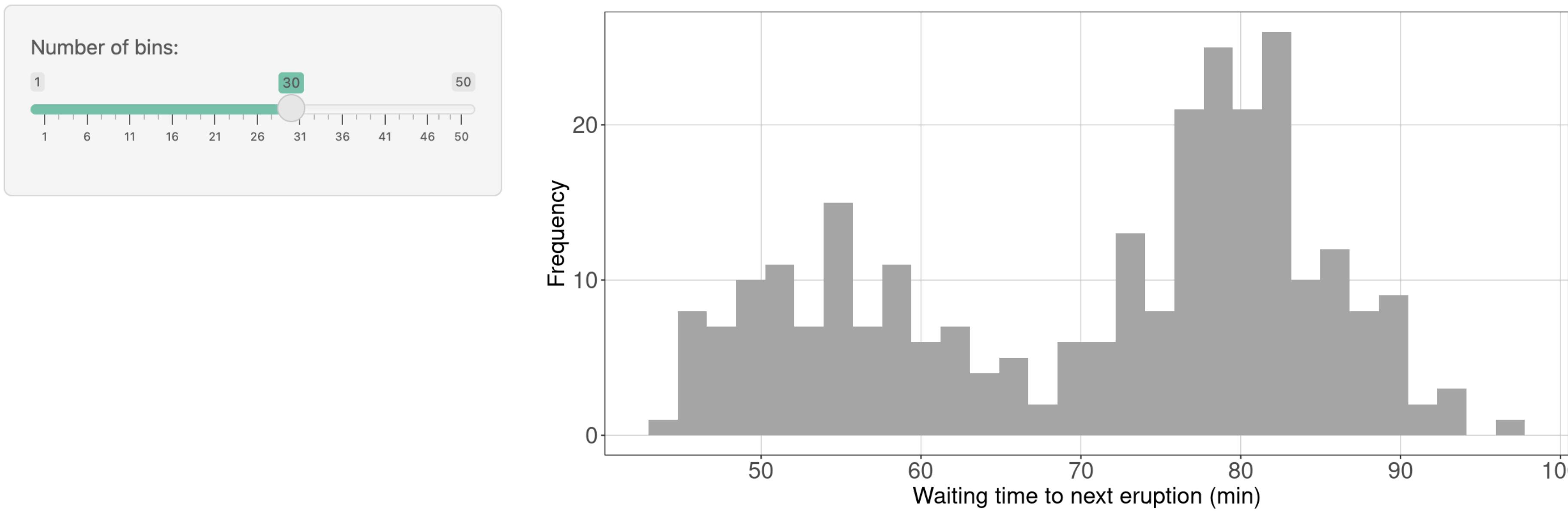
ID	Name	Status
10690474	my_shiny_app	Running
10477749	vamos_app	Sleeping

© 2022 Posit Software, PBC | All Rights Reserved | Terms Of Use

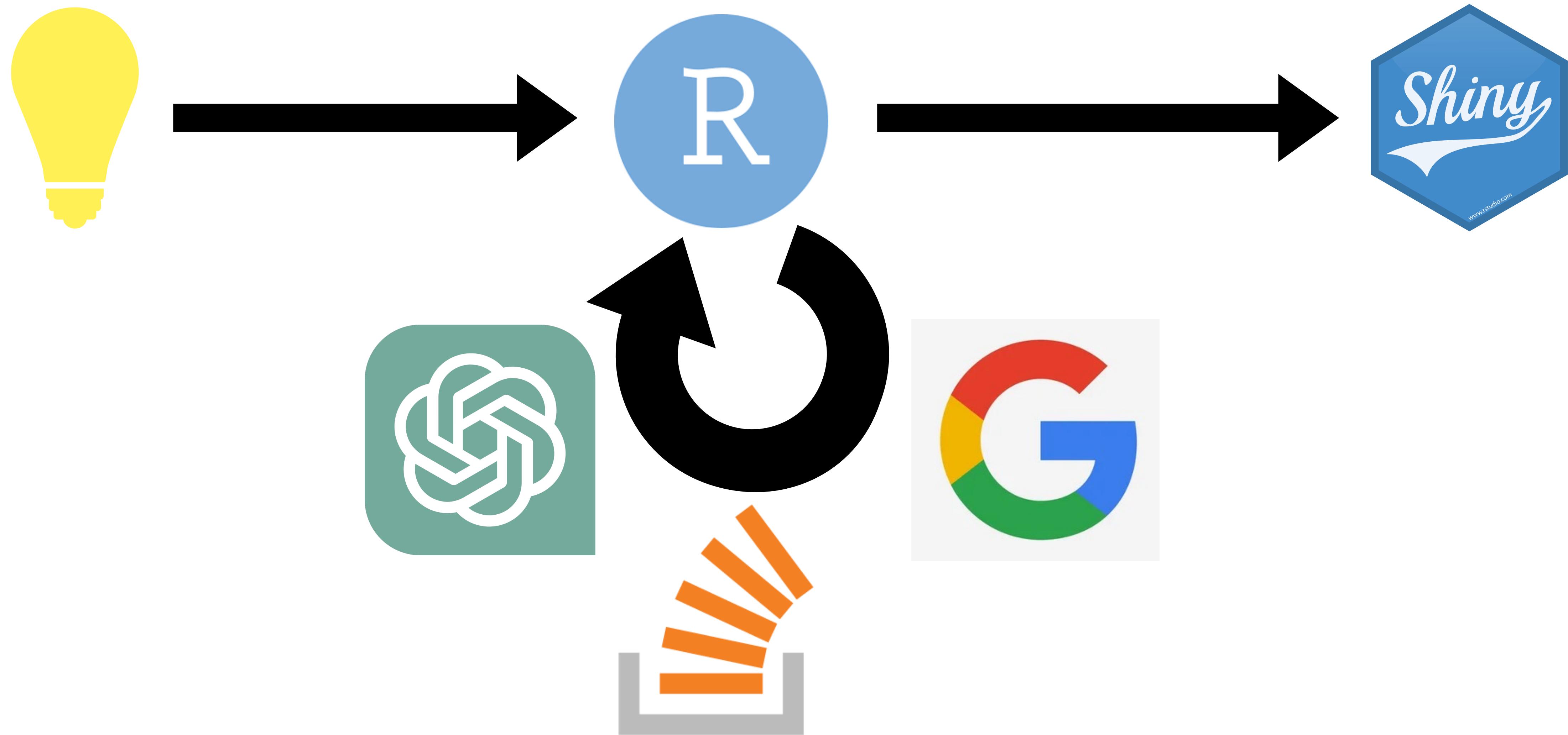
My app is online!



Old Faithful Geyser Data



How to *really* build a shiny app



Resources

- Shiny introduction: <https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>
- Shiny Gallery: <https://shiny.posit.co/r/gallery/>
- GitHub Repo for this app: https://github.com/sophia-gibson/shiny_intro_cshl2023

Now with
X-chromosome methylation!



1000G ONT



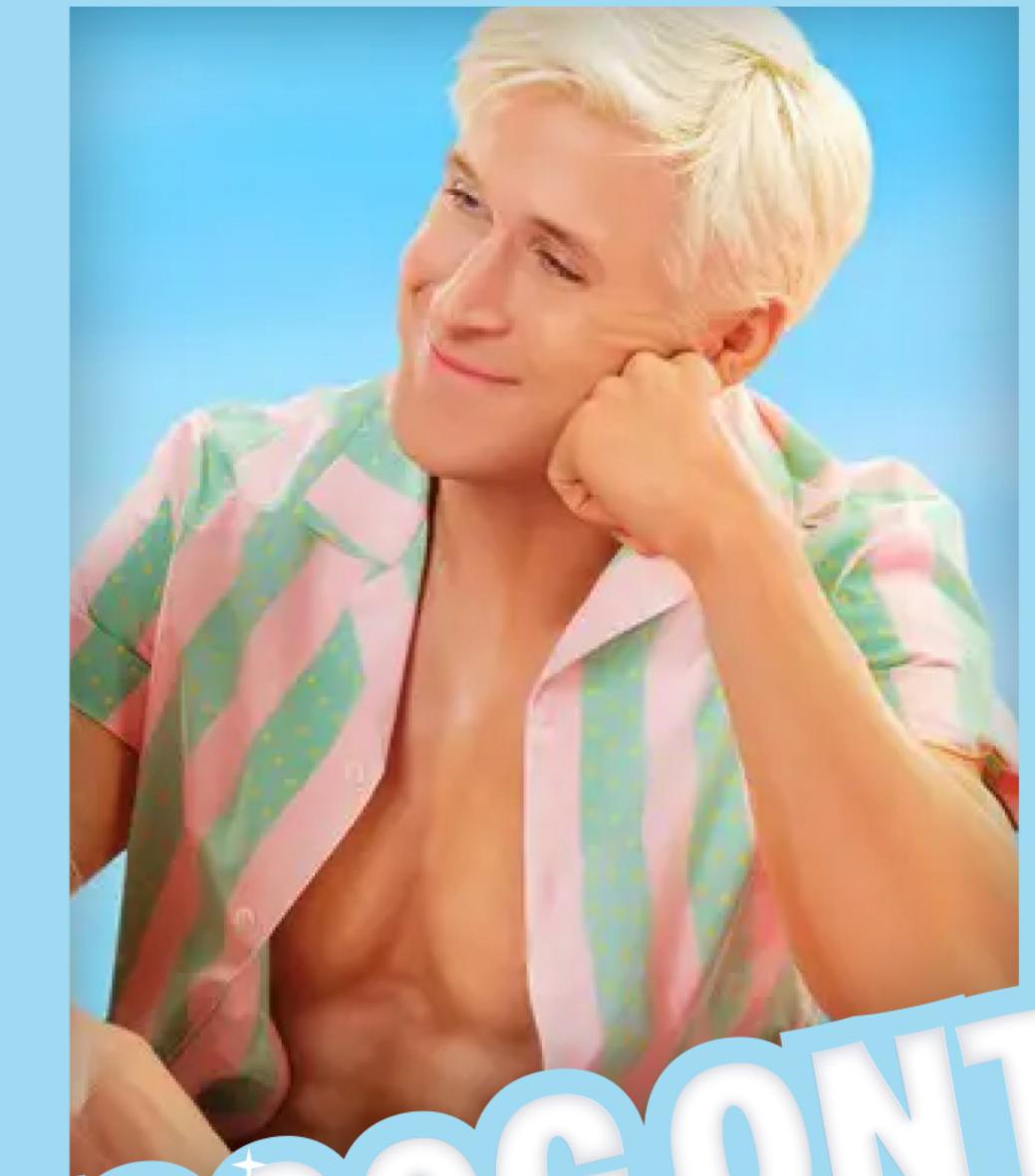
Haven't you always
wanted to easily browse
1000G ONT
Consortium data?

Well, now you
Ken!

But wait! Visit now and we'll add
structural variants
FREE!

Sophia Gibson
Monday, September 11

Now with
repeat expansions!



1000G ONT

