Widgets

Publishing a Widget

Any HTML format (document, notebook, presentation, or dashboard) can contain interactive components.

htmlwidgets

HTML is an interactive format, and you can take advantage of that interactivity with **htmlwidgets**, R functions that produce interactive HTML visualisations. For example, take the **leaflet** map below. If you're viewing this page on the web, you can drag the map around, zoom in and out, etc. You obviously can't do that in a book, so rmarkdown automatically inserts a static screenshot for you.

```
library(leaflet)
leaflet() %>%
  setView(174.764, -36.877, zoom = 16) %>%
  addTiles() %>%
  addMarkers(174.764, -36.877, popup = "Maungawhau")
```

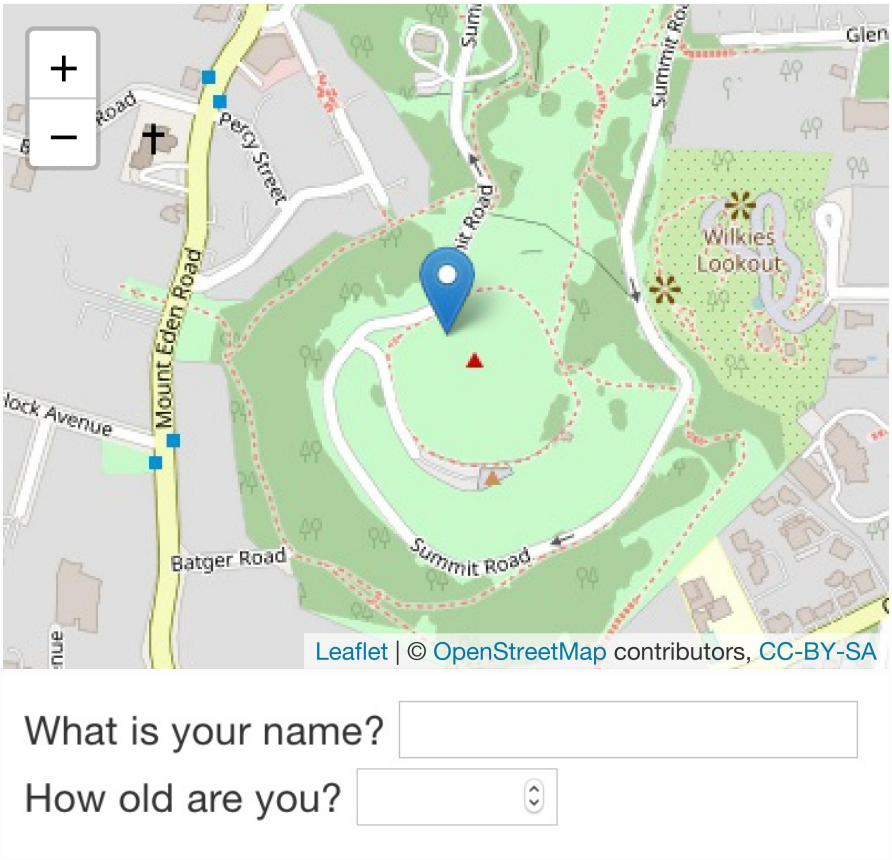


Figure 36.1
You can then refer to the values with `input$name` and `input$age`, and the code that uses them will be automatically re-run whenever they change.

I can't show you a live shiny app here because shiny interactions occur on the **server-side**. This means that you can write interactive apps without knowing JavaScript, but you need a server to run them on. This introduces a logistical issue: Shiny apps need a Shiny server to be run online. When you run shiny apps on your own computer, shiny automatically sets up a shiny server for you, but you need a public facing shiny server if you want to publish this sort of interactivity online. That's the fundamental trade-off of shiny: you can do anything in a shiny document that you can do in R, but it requires someone to be running R.

Learn more about Shiny at http://shiny.rstudio.com/.

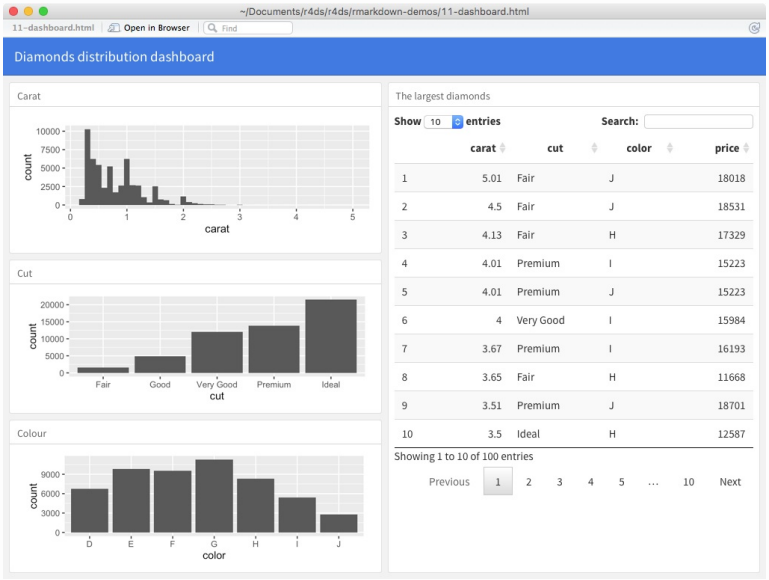
Dashboards

Publishing a Dashboard

Dashboards are a useful way to communicate large amounts of information visually and quickly. Flexdashboard makes it particularly easy to create dashboards using R Markdown and a convention for how the headers affect the layout:

- Each level 1 header (#) begins a new page in the dashboard.
- Each level 2 header (##) begins a new column.
- Each level 3 header (###) begins a new row.

For example, you can produce this dashboard:



.guides/img/rmarkdown-flexdashboard

Figure 36.1

Using this code:

````
---
title: "Diamonds distribution dashboard"
output: flexdashboard::flex_dashboard
---

```{r setup, include = FALSE}
library(ggplot2)
library(dplyr)
knitr::opts_chunk$set(fig.width = 5, fig.asp = 1/3)
```

## Column 1

### Carat

```{r}
ggplot(diamonds, aes(carat)) + geom_histogram(binwidth = 0.1)
```

### Cut

```{r}
ggplot(diamonds, aes(cut)) + geom_bar()
```

### Colour

```{r}
ggplot(diamonds, aes(color)) + geom_bar()
```

## Column 2

### The largest diamonds

```{r}
diamonds %>%
  arrange(desc(carat)) %>%
  head(100) %>%
  select(carat, cut, color, price) %>%
  DT::datatable()
```
````

Flexdashboard also provides simple tools for creating sidebars, tabsets, value boxes, and gauges. To learn more about flexdashboard visit http://rmarkdown.rstudio.com/flexdashboard/.