The Challenger Space Shuttle Disaster

AUTHOR
Prof. Bell (worked with Finn)

Introduction to Quarto

Quarto is a tool to easily generate reports using code. Quarto is already installed in Posit Cloud projects. To open a new Quarto file, go to the "New File" button and select "Quarto Document."

Quarto is able to generate reports as HTML, PDF, Word, and other file types. Because we are working in Posit Cloud, we will only render our reports as HTML and PDF documents, which we can preview in the "Viewer" pane.

There are three basic principles of Quarto that you should know before we get started:

- Quarto is a separate tool from R and RStudio. Quarto itself uses it's own language, as well as another language called Markdown. In fact, Quarto can be used with many different coding languages. For that reason, we will now be working with `.qmd` scripts rather than `.R` scripts.
- When you render a Quarto document, it is as if you are running your code in a **brand new** R session. This means that none of the objects in your Environment pane will be known to your `.qmd` file. Any objects that you want to use in your Quarto document need to be created within the `.qmd` script.

  However, Quarto will be able to access all of your installed packages. **You should never include `install.packages()` in a `.qmd` file. All packages should be installed in your Posit Cloud project before attempt to render the document.** You will need to load those packages with `library()` in your `.qmd` file since the document is rendering in a new R session.

  - The working directory is the folder where the `.qmd` file lives. This is where R looks for files that you ask it to load, and where it will put any files that you ask it to save.

    For example, if you are loading a `csv` file from a folder called `data` and your `.qmd` file is in your main project folder, your command will be `read_csv("data/filename.csv")`.

  - A `.qmd` file with an error in the code will not render.
- The R code you write in a `.qmd` script is just like the code that you write in an `.R` script, except that it will go inside of a "code chunk." You can create a new code chunk by clicking on the green "C" icon in the upper right of the Scripts pane. Code chunks will run in the order that the are written in the script (top to bottom), *except* for a special chunk called the `setup` chunk. The `setup` chunk is always run first. It should contain all of the introductory commands that we usually include at the beginning of our `.R` scripts. Here is the `setup` chunk for this document:

```
library(ggplot2)
library(openintro)
library(dplyr)
```

The `#| message: false` option in my code chunk header means that Quarto should not print any warning messages from that code.

The Challenger Space Shuttle Disaster

In 1986, the NASA space shuttle Challenger exploded just over a minute after launch due to a faulty part called an O-ring. Unfortunately, NASA engineers had enough information to know the the O-rings were likely to fail, but did not display the information in a way that would lead mission control make the decision to cancel the launch. Let's load the `orings` data from the `openintro` package to see what went wrong.

```
data(orings)
```

We can now `glimpse()` our data and use the help file (`?orings`) to learn more about our data. However, we should do this *outside* of Quarto, since we probably do not want to show our `glimpse()` output or the help file in our document. Alternatively, we can use the `include` code chunk option to hide the code and it's output from our document.

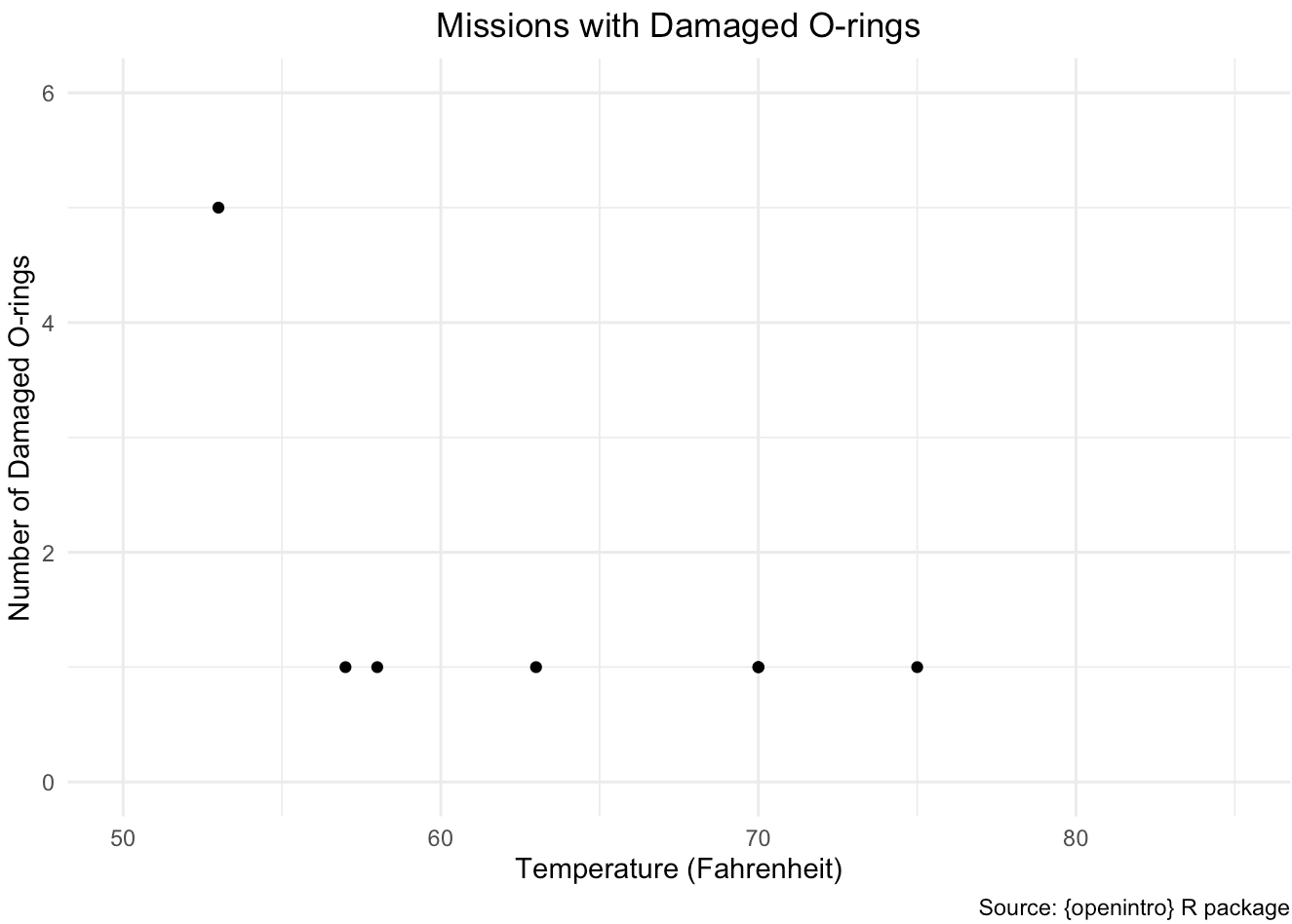
Generally speaking, we do not want to include "raw" R output in our Quarto documents.

The Morning of the Launch

NASA engineers convened an emergency conference on the morning of the launch to consider whether the O-rings might fail due to some particularly cold temperatures at the Kennedy Space Center that day. Someone scribbled down a chart to show all the O-ring failures that NASA had experienced in the past:

▼ Code

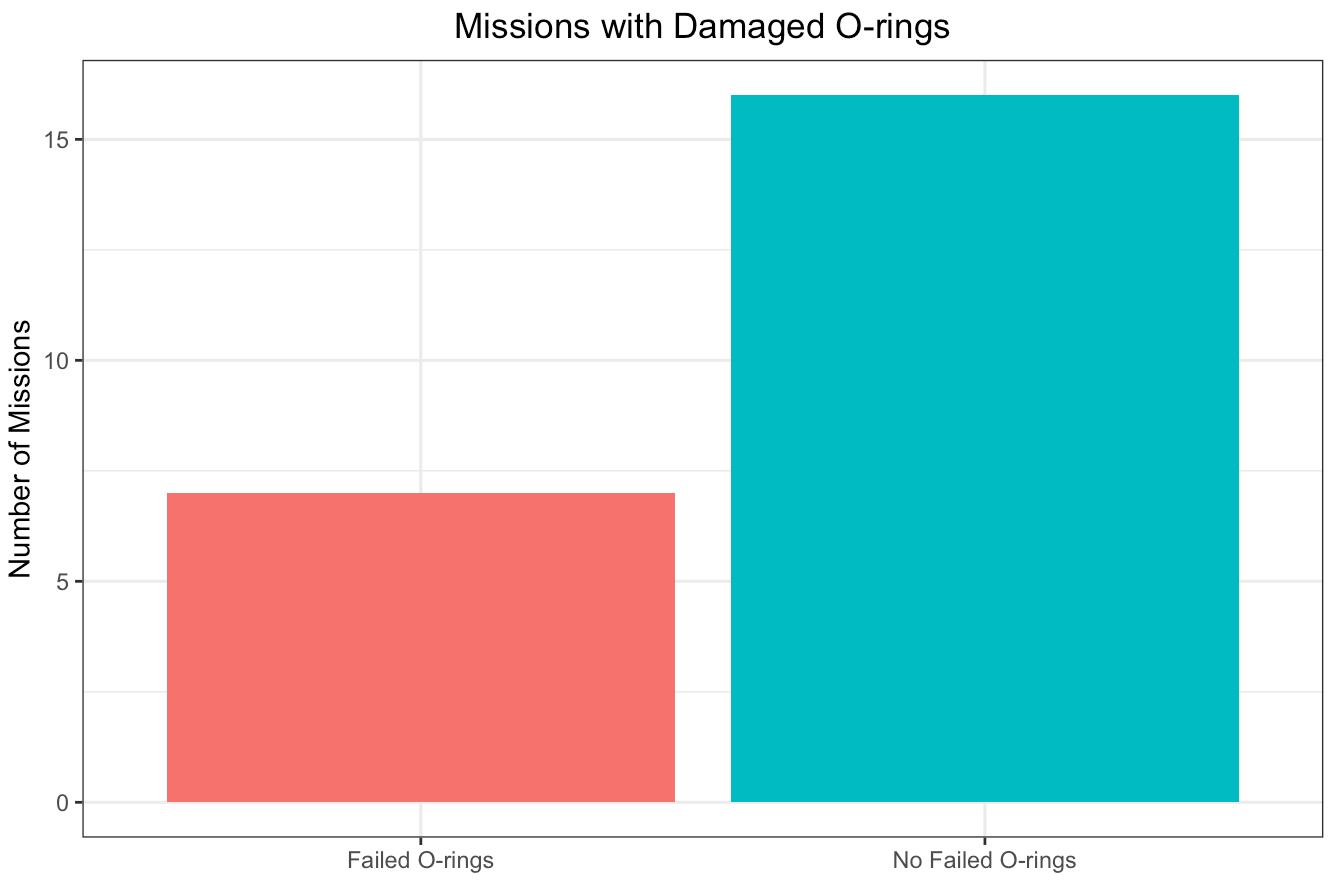
```
past_failures <- orings[orings$damaged > 0,]

ggplot(past_failures) +
    geom_point(aes(x = temperature, y = damaged)) +
  scale_x_continuous(limits = c(50,85)) +
  scale_y_continuous(limits = c(0, 6)) +
  labs(x = "Temperature (Fahrenheit)",
       y = "Number of Damaged O-rings",
       title = "Missions with Damaged O-rings",
       caption = "Source: {openintro} R package") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5))
```



From this data, there does not appear to be a clear correlation between temperature and O-ring failures. But here is the problem: we only know half of the story. With only these data points, we aren't seeing most of the data. How many missions did not have any failed O-rings?

▼ Code

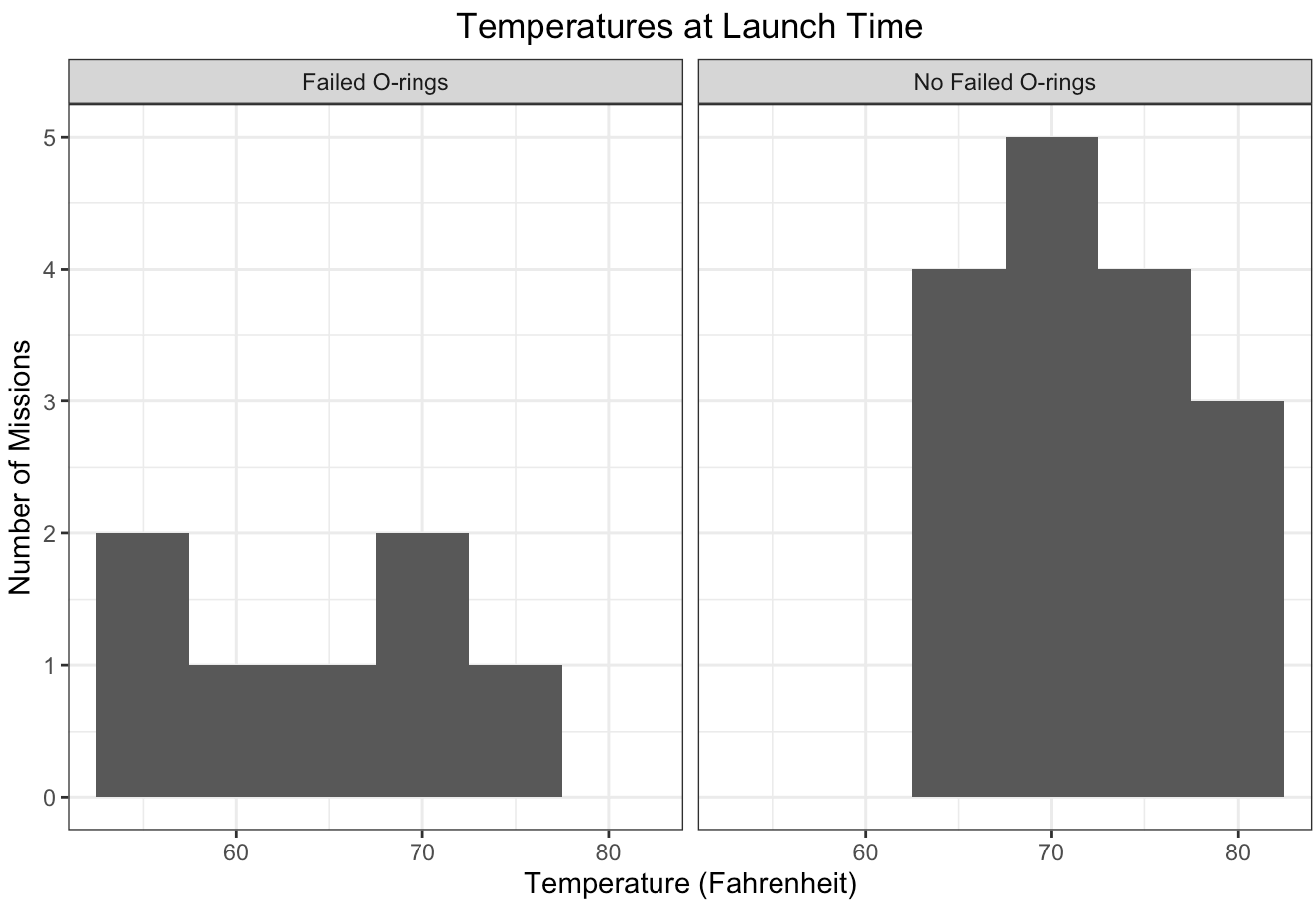
```
orings$fail <- ifelse(orings$damaged > 0, "Failed O-rings", "No Failed O-rings")

ggplot(orings) +
  geom_bar(aes(x = fail, fill = fail)) +
  labs(x = "",
       y = "Number of Missions",
       title = "Missions with Damaged O-rings",
       caption = "Source: {openintro} R package") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5),
        legend.position = "none")
```



In fact, 16 missions had no failed O-rings. How do the distributions of temperatures compare in these two groups?

▼ Code

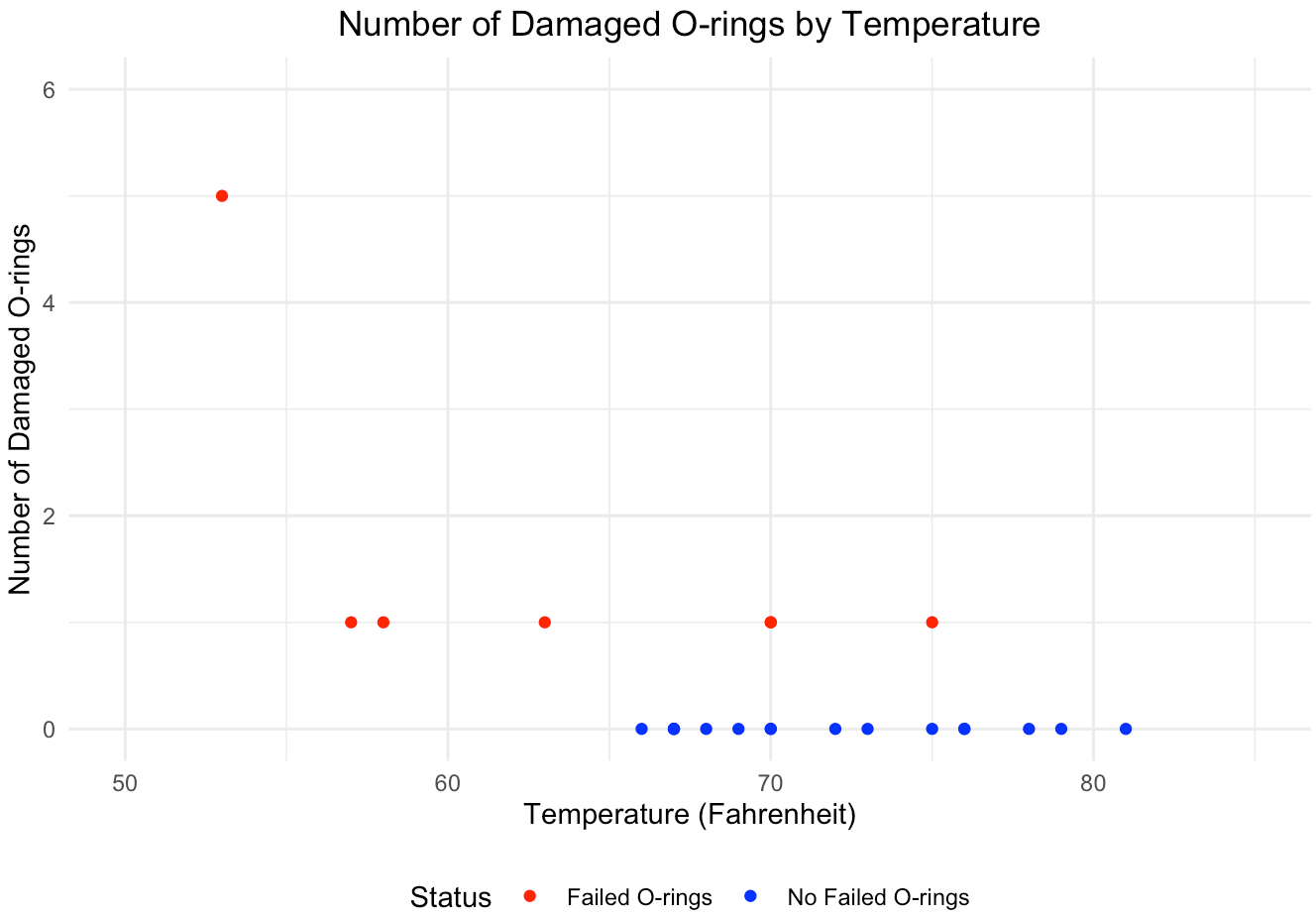
```
ggplot(orings) +
  geom_histogram(aes(x = temperature), binwidth = 5) +
  facet_wrap(~ fail) +
  labs(x = "Temperature (Fahrenheit)",
       y = "Number of Missions",
       title = "Temperatures at Launch Time",
       caption = "Source: {openintro} package") +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5))
```



Now it seems much more likely that temperature is correlated with O-ring failure. Let's imagine that the NASA engineers had designed their scatterplot to show this difference:

▼ Code

```
ggplot(orings) +
    geom_point(aes(x = temperature, y = damaged, color = fail)) +
  scale_x_continuous(limits = c(50,85)) +
  scale_y_continuous(limits = c(0, 6)) +
  scale_color_manual(values = c("red", "blue")) +
  labs(x = "Temperature (Fahrenheit)",
       y = "Number of Damaged O-rings",
       title = "Number of Damaged O-rings by Temperature",
       color = "Status") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = .5),
        legend.position = "bottom")
```



If NASA engineers had presented the data in this way, it seems much more likely that mission control would have cancelled the launch.