LEARN MORE 365

Quarto Lesson

Quarto

What is Quarto?

Quarto enables you to weave together content and executable code into a finished document. Quarto will output (or "render") that document in a format of your choosing (PDF, Word, HTML etc.).

In this example, the text is about to be interrupted with some R code in which we'll run 1 + 1.

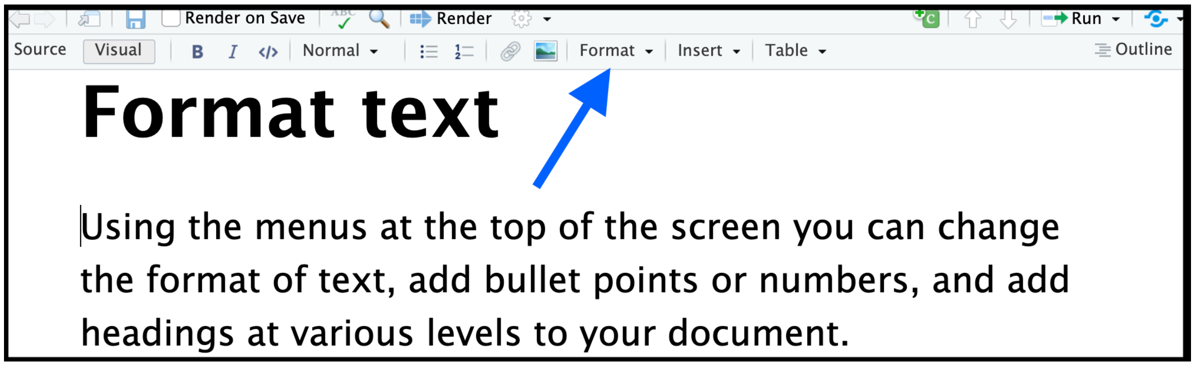
You'll notice that both the code and the code output are included in the rendered document. We can choose to hide the code and only include the output (but more about that later). Here is a simple example of both code and the output being inserted into our document.

```
1 + 1
```

```
[1]2
```

Editing and formatting

Most editing and formatting in Quarto work exactly as it would in any other word processor.



Format text

Using the menus at the top of the screen you can change the format of text, add bullet points or numbers, and add headings at various levels to your document.

Tip

You can even insert a callout note like this. In this note I'll remind you that you can insert pictures, hyperlinks etc. all by using either the menus or the upward slash "/" key that will provide you with a shortcut menu of insertion options

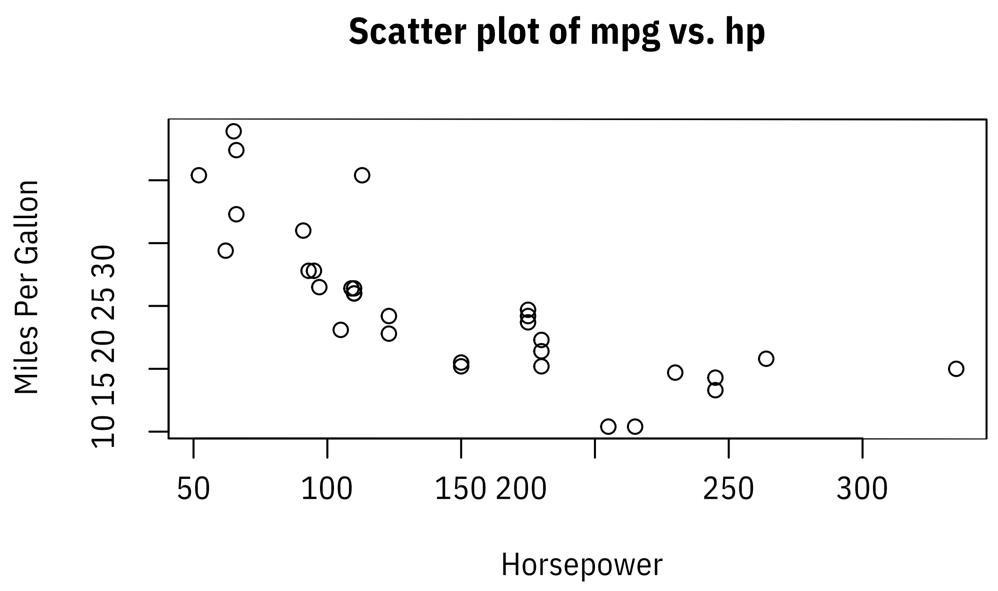
Text can also be emphasized like this by surrounding the text with a backtick '

Chunks of code

The basics

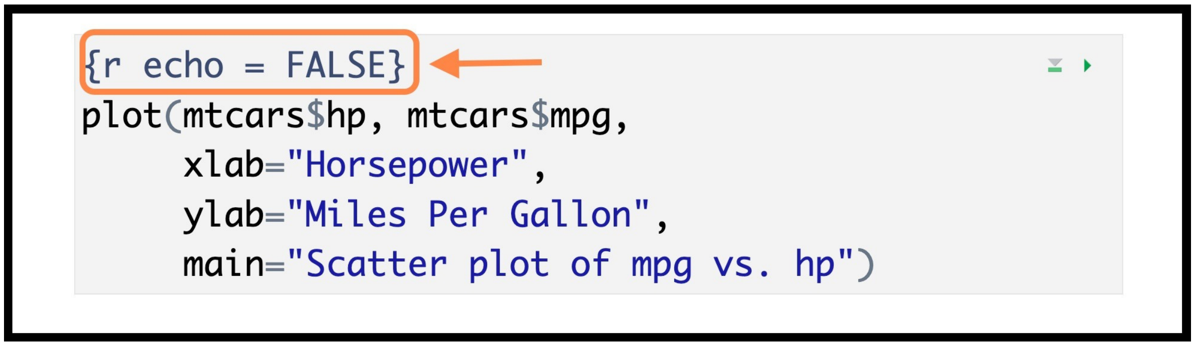
Once again, either by using menus or the upward slash "/" shortcut, you can insert code into your document. In the example below, we'll create a simple plot.

```
plot(mtcars$hp, mtcars$mpg,
xlab="Horsepower",
ylab="Miles Per Gallon",
        main="Scatter plot of mpg vs. hp")
```



If we wanted the output plot but didn't want to display the code we'd simple apply an `option` in the chunk heading. In this case we'd apply echo = FALSE

Here is a screenshot of that code (because by definition, do it isn't going to render into this cument.



```
{r echo = FALSE}
plot(mtcars$hp, mtcars$mpg,
     xlab="Horsepower",
     ylab="Miles Per Gallon",
     main="Scatter plot of mpg vs. hp")
```

The most commonly used controls are:

1. **include**: Determines if both the code and output should be included in the final document. Default is **TRUE**.

• **include=FALSE** will hide both the code and its output.

2. **echo**: Controls whether the code is displayed. Default is **TRUE**.

• **echo=FALSE** hides the code but shows the output.

3. **results**: Controls how the output is formatted.

• **results='hide'** hides printed output but not plots.

• **results='asis'** treats the output as if it were part of the markdown document (useful for generating tables or raw HTML).

4. **warning**: Controls the display of warnings. Default is **TRUE**.

• **warning=FALSE** hides all warnings generated by that chunk.

5. **message**: Controls the display of messages. Default is **TRUE**.

• **message=FALSE** hides all messages generated by the chunk.

6. **error**: Controls the behavior upon encountering errors. Default is **TRUE**.

• **error=FALSE** will continue rendering the rest of the document even if there are errors in the chunk.

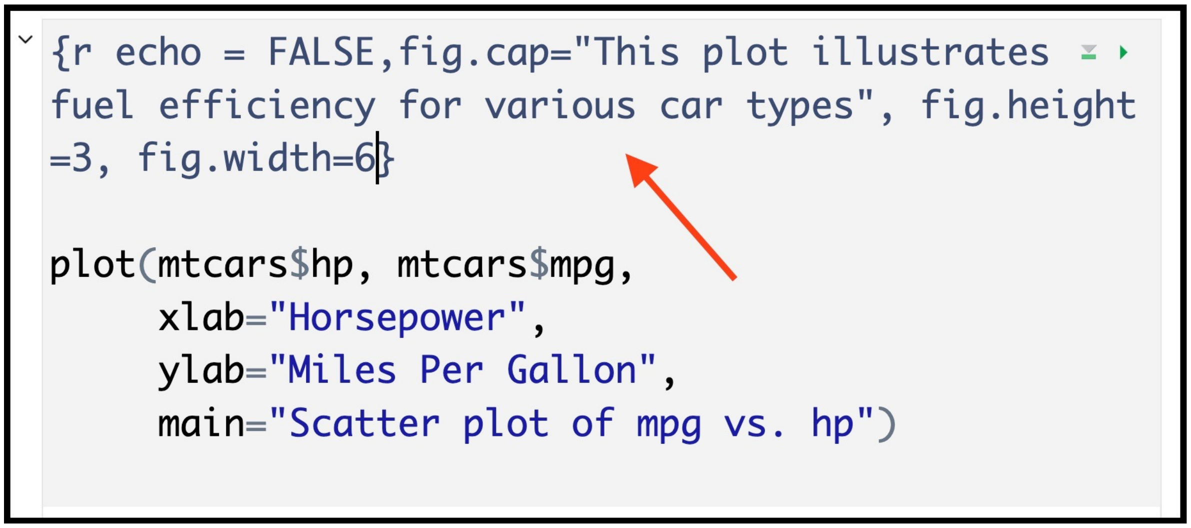
7. **fig.cap**: Provides captions for figures.

• **fig.cap="A caption for the plot."** will label the plot with the provided caption.

8. **fig.height** and **fig.width**: Control the height and width of the plot, respectively.

• **fig.height=5** and **fig.width=7** will adjust the size of the plot.

Let's apply some of these to the plot that we created above. In the plot below I've asked Quarto to hide the code (which we've seen before). If we were to include the code we wouldn't see the controls (they are always hidden) so I'm going to insert a screenshot of them so that you can see how they relate to the output.



```
{r echo = FALSE,fig.cap="This plot illustrates fuel efficiency for various car types", fig.height =3, fig.width=6}

plot(mtcars$hp, mtcars$mpg,
     xlab="Horsepower",
     ylab="Miles Per Gallon",
     main="Scatter plot of mpg vs. hp")
```

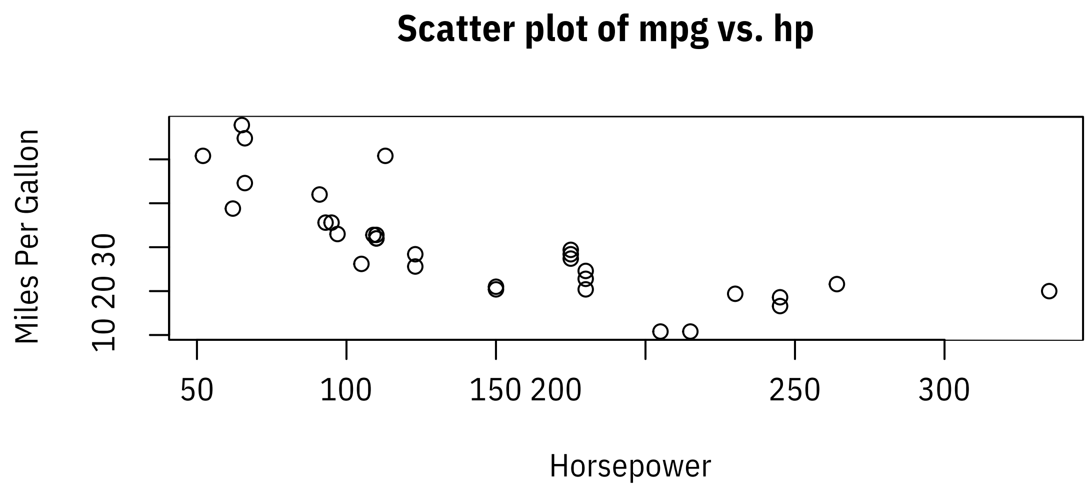


Figure 1: This plot illustrates fuel efficiency for various car types

To create the plot below I used the following controls at the beginning of the code chunk.

{r echo = FALSE, message=FALSE, label = fig-starwars, fig.cap="This figure illustrates the relationship between height and mass accross characters in starwars movies"}

Let me walk you through them one at a time. echo = FALSE and fig.cap we've talked about (so I won't go over those again). message = FALSE means that the messages that appear when I use the code library(tidyverse) are excluded too. I call the tidyverse package in order to utilize the both ggplot and the starwars dataset. label = fig-starwars allows me to make reference to the plot with a hyperlinked figure reference. To do this we can insert the text @fig-starwars anywhere in our text and we'll get this " Figure 2 " in the body of our text. Neat hey?

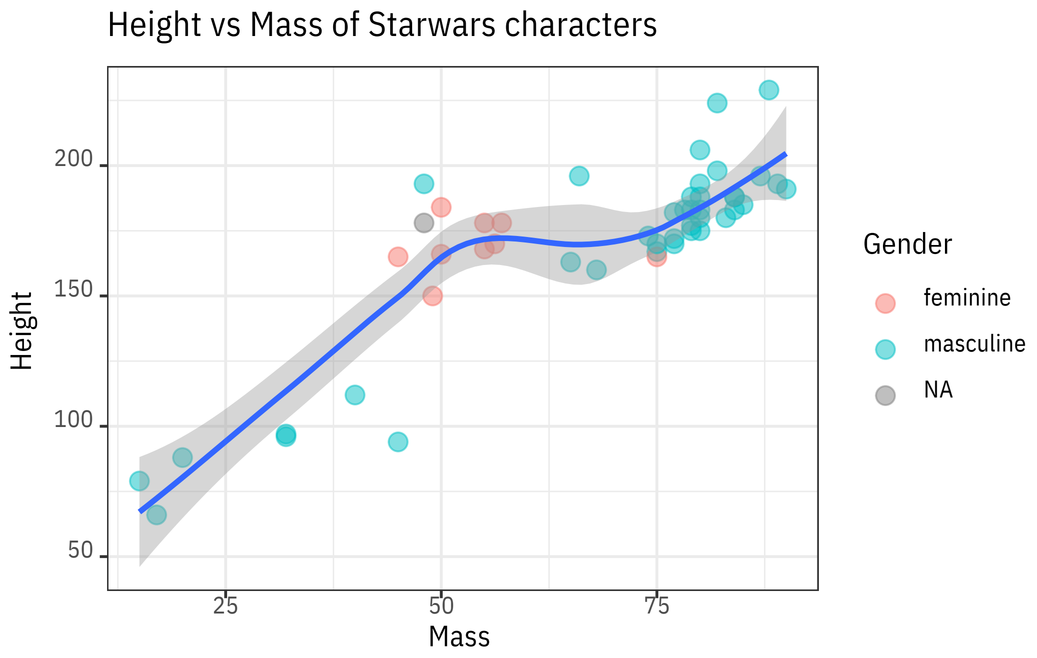


Figure 2: This figure illustrates the relationship between height and mass accross characters in starwars movies

The figures will automatically number themselves according to the order in which they are found in the document. And of course, by using @fig- you can make reference to any of the figures at any point in the document and you'll get a hyperlink with the correct figure number that will take you to the point in the document where that figure is, like this Figure 3 . Note that the caption below the figure also contains the appropriate figure number.

Code inserted into your text

You might not want to interrupt your text with a chunk of code but instead might want to make reference to a data within the text of your document itself. For example, I might want to say that the average height of starwars characters is 1.74 meters. Here is a screenshot of how I inserted code into my text. Remember that the highlighted text below has backticks ‘ ‘ wrapped around it.

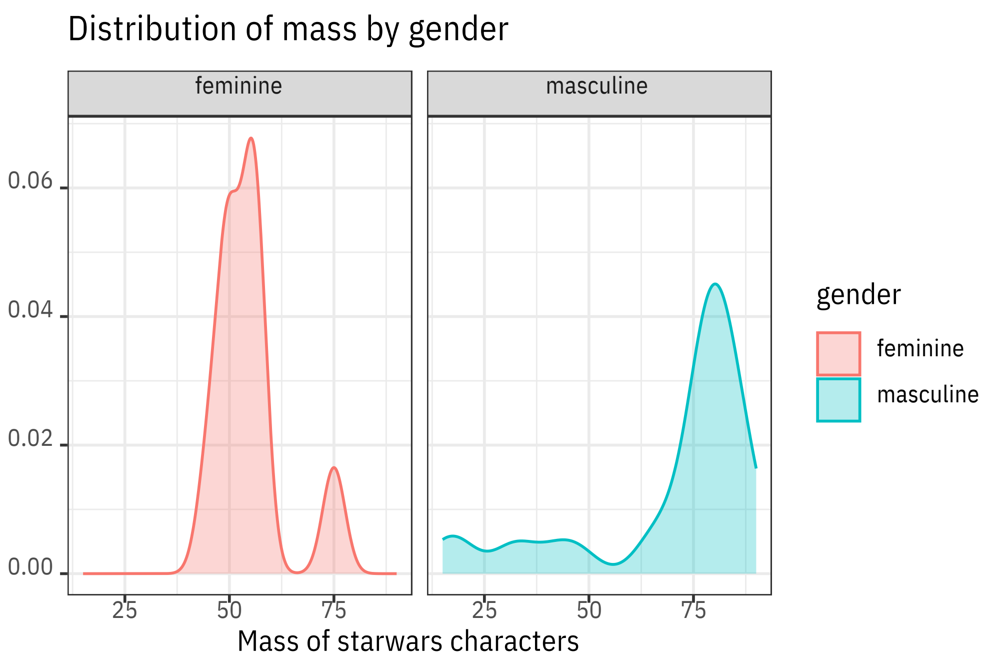
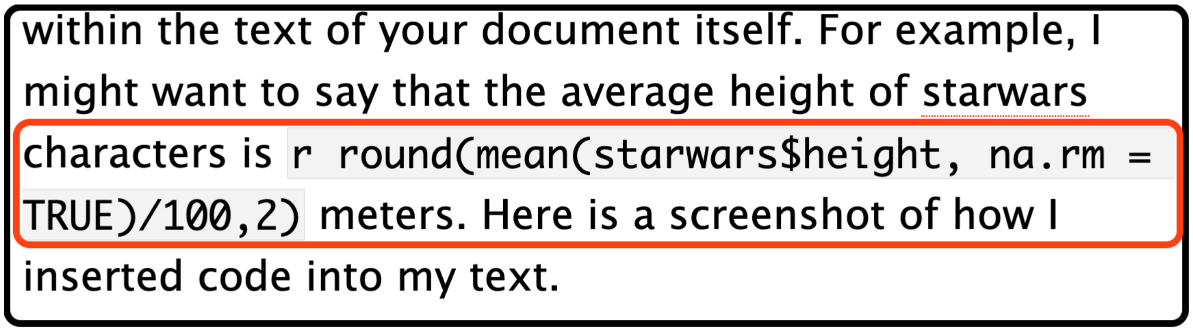


Figure 3: This figure is a density plot providing the relative distribution of the masses of starwars characters



within the text of your document itself. For example, I might want to say that the average height of starwars characters is `r round(mean(starwars$height, na.rm = TRUE)/100,2)` meters. Here is a screenshot of how I inserted code into my text.

Inserting a table

You can of course insert a table directly into the text (as you might with any word processor) or have a table that is created from data and code (like the one below). A shout-out to Little Miss Data who created this example table using the formattable package. See details here.

Indicator Name	2011	2012	2013	2014	2015	2016	Average	Improvement
Diabetes	8.0	7.2	9.3	7.2	7.5	10.4	8.27	-30
Tobacco Use	17.4	15.0	15.3	12.2	16.6	16.7	15.53	4.02
Obesity	19.1	23.6	23.3	20.5	24.0	23.2	22.28	-21.47
Cardiovascular Disease	5.0	4.9	1.5	4.4	4.9	6.2	4.48	-24

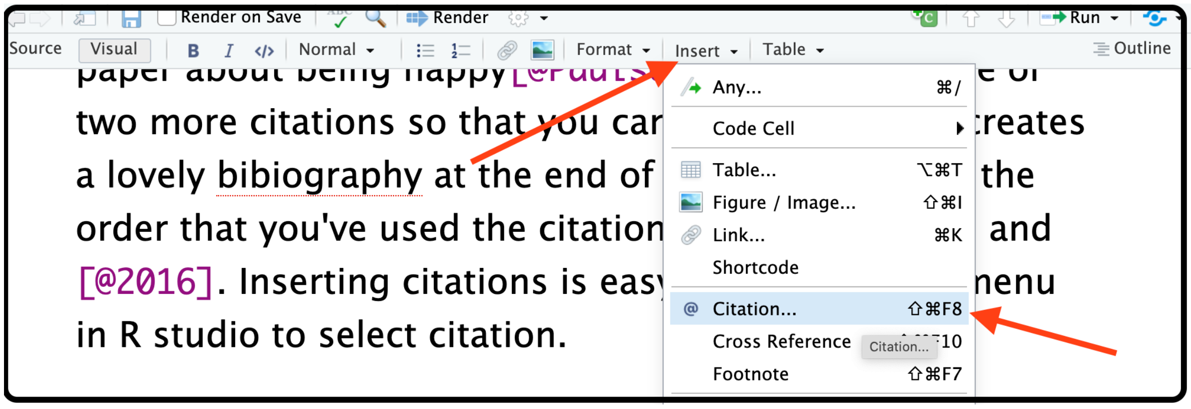
Warning

Some packages that are used to create tables do not work across all output formats. In this case, the formattable package creates and output that works well in HTHM but not PDFs and Word docs. There are workarounds but they are beyond the scope of this do cument.

Citations and references

You can easily cite peer reviewed literature and have Quarto create a bibliography at the end of your document.

Here is a paper from the PubMed database about using R programming for data analysis(Chan 2018). And here is a paper about being happy(Paulson et al. 2016). I'll use one or two more citations so that you can see how Quarto creates a lovely bibiography at the end of your document in the order that you've used the citations (Paulson et al. 2016) and (The Lancet 2016). Inserting citations is easy. Simply use the menu in R studio to select citation.


two more citations so that you car   creates
a lovely bibiography at the end of   the
order that you've used the citation   and
[@2016]. Inserting citations is easy   menu
in R studio to select citation.

Bibliography

Chan, Bertram K. C. 2018. “Data Analysis Using r Programming.” In, 47–122. Springer International Publishing. https://doi.org/10.1007/978-3-319-93791-5_2.

Paulson, Steve, Kim K. Azzarelli, Darrin M. McMahon, and Barry Schwartz. 2016. “A New Science of Happiness: The Paradox of Pleasure.” *Annals of the New York Academy of Sciences* 1384 (1): 12–31. https://doi.org/10.1111/nyas.13068.

The Lancet. 2016. “Health and Happiness.” *The Lancet* 387 (10025): 1251. https://doi.org/10.1016/s0140- 6736(16)30062- 9.

Learn More 365

Share this cheat sheet with others

Unlock a wealth of FREE resources TODAY!

Click on the links below to enrich your knowledge in these topics:

- 📚 Statistics & Research Methods Resource Library
- 📚 R Programming & Data Visualization Resource Library
- 📚 Discover the Public Health & Epidemiology Resource Library

Explore video lessons on **Statistics & Research Methods, R Programming & Data Visualization Public Health & Epidemiology,** and more at Learn More 365!

Click **HERE** or scan the QR code below to start learning!