

demo-R

February 2, 2022

An example Jupyter file, illustrating the use of R code chunks.

1 How to generate a document from this file

You can create an HTML or PDF output file by simply selecting

`File -> Download as -> HTML (.html)` or `File -> Download as -> PDF via LaTeX (.pdf)`

To generate a document from the command line, you can do the following:

```
jupyter nbconvert demo-R.ipynb --to html # for html
jupyter nbconvert demo-R.ipynb --to pdf  # for pdf
```

Generating PDF involves converting the Notebook to LaTeX, and that seems to fail for notebooks using R.

2 Some basic Markdown formatting

Here's an *introduction* to our **critical** discovery. We have some code to display but not evaluate: `exp(7)`, and we can embed the code in a static code block as follows:

```
a = 7 %% 5
b = exp(a)
```

This document will focus on embedding math and code and not on standard Markdown formatting. There are lots of sources of information on Markdown. [RStudio has good information on R Markdown](#) (including Markdown formatting).

3 Embedding equations using LaTeX

This can be done with the following syntax.

Here is an inline equation $f(x) = \int f(y, x)dy$. (You may need to make sure that no space appears immediately after the first “\$”.

Here's a displayed equation

$$f_{\theta}(x) = \int f_{\theta}(y, x)dy.$$

4 Embedding R code

Here's an R code chunk

```
[12]: a <- c(7, 3)
      mean(a)
      b <- a + 3
      mean(b)
```

5

8

Here's another chunk

```
[13]: mean(b)
```

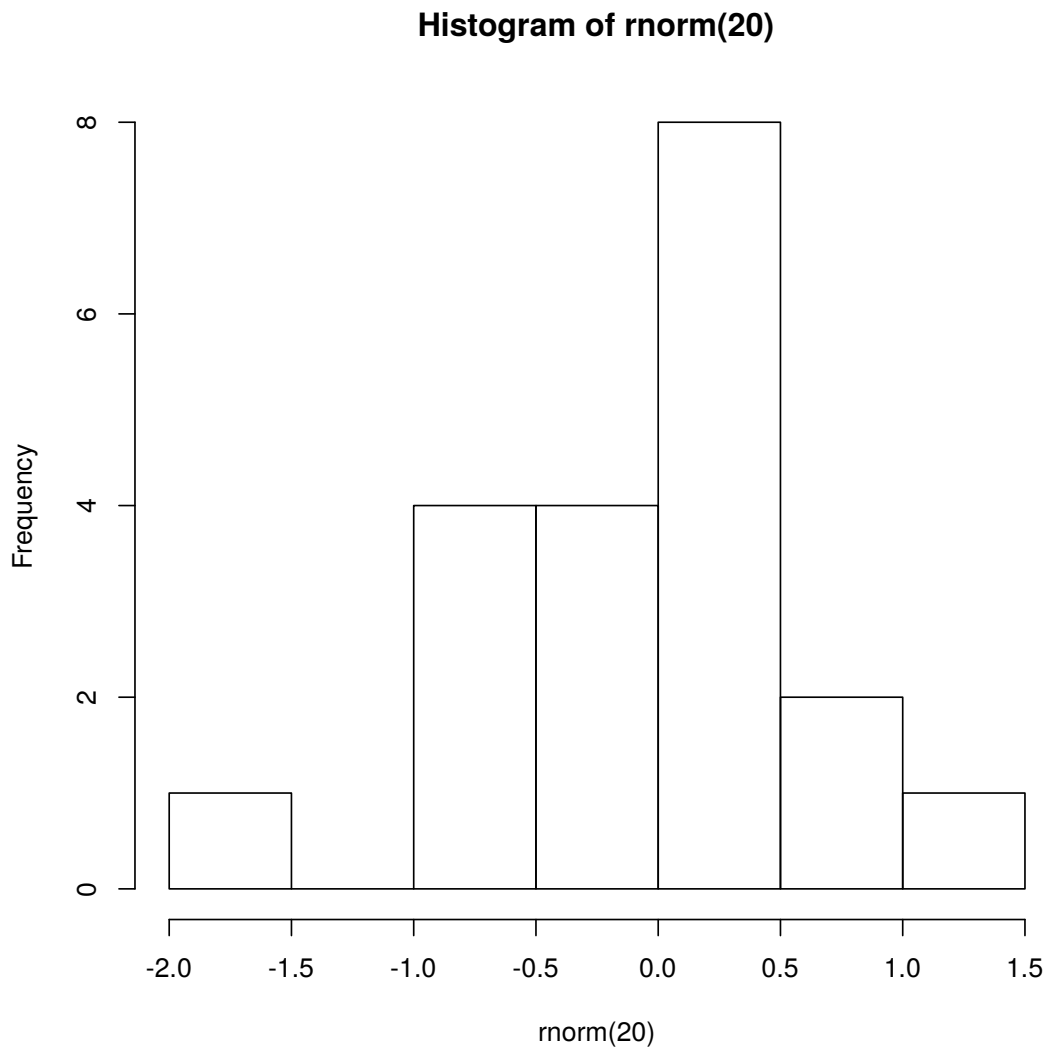
8

Note above that output is not printed interspersed with the code, as one would generally want. So you need to break up your code into chunks such that the output appears immediately following the code.

Also note above that later chunks have access to result from earlier chunks (i.e., state is preserved between chunks).

Let's make a plot as follows.

```
[2]: hist(rnorm(20))
```



You can't insert inline code to be evaluated.

To control whether code is evaluated, simply choose whether or not to run the chunk (i.e., using the evaluation button in the toolbar). I don't believe you have control over whether the code in a chunk is printed when creating an output file.

```
[ ]: a <- c(7, 3)
      mean(a)
      b <- a + 3
      mean(b)
```

I'm not aware of a direct way to cache the results of a calculation to save time when running the code in a notebook multiple times.

5 Embedding Python and bash code

Unlike in a notebook with Python chunks, I haven't been able to use the %% magic to embed Python or bash code chunks.

6 Reading code from an external file

I'm not aware of a way to directly source code from an external file as the code in a code chunk.

7 Formatting of long lines of code and of output

Here we'll see code and output of a code chunk do not overflow the page but the line breaks may be awkward in HTML output and long lines in the notebook document will cause a horizontal scrollbar to be inserted.

```
[1]: b <- "Statistics at UC Berkeley: We are a community engaged in research and_
      ↪education in probability and statistics. In addition to developing_
      ↪fundamental theory and methodology, we are actively"
      # Statistics at UC Berkeley: We are a community engaged in research and_
      ↪education in probability and statistics. In addition to developing_
      ↪fundamental theory and methodology, we are actively

      vecLongName = rnorm(100)
      a = length(mean(5 * vecLongName + vecLongName - exp(vecLongName) + vecLongName_
      ↪* vecLongName, na.rm = TRUE))
      a = length(mean(5 * vecLongName + vecLongName)) # this is a comment that goes_
      ↪over the line by a good long ways
      a = length(mean(5 * vecLongName + vecLongName - exp(vecLongName) + vecLongName,
      ↪na.rm = TRUE)) # this is a comment that goes over the line by a good long_
      ↪long long long long long long long ways

      # long output usually is fine
      rnorm(30)
```

Statistics at UC Berkeley: We are a community engaged in research and education in probability and statistics. In addition to developing fundamental theory and methodology, we are actively

```
1. 0.120844352144518 2. 0.530590817231262 3. 0.0774870494719949 4. -0.832056761924332
5. 1.4026610580005 6. -0.288780491480682 7. -0.878128990347846 8. -2.32536594919303
9. 0.321131046124207 10. 1.27552909256496 11. -0.469432568639959 12. 0.114622189377705
13. -0.0512167979060532 14. -0.557890165604866 15. -0.71000249210743 16. 0.529386133701903
17. 0.262432847501987 18. 0.0186038647392888 19. 1.30155867387957 20. 2.28504665129921
21. 0.279729434738486 22. -1.60923627270044 23. -0.303750332004864 24. 1.2171462564197
25. -1.21134380592999 26. 1.25132025275561 27. 0.458455270108449 28. -0.403808425001392
29. 0.89952464026832 30. -0.629759869330632
```

We can tidy up the code manually to improve where the lines break.

```
[2]: # breaking up a string
b <- "Statistics at UC Berkeley: We are a community engaged in research
and education in probability and statistics. In addition to developing
fundamental theory and methodology, we are actively"

# breaking up a comment
# Statistics at UC Berkeley: We are a community engaged in research and
# education in probability and statistics. In addition to developing
# fundamental theory and methodology, we are actively

# breaking up code lines
vecLongName = rnorm(100)
a <- length(mean(5 * vecLongName + vecLongName - exp(vecLongName) +
vecLongName * vecLongName, na.rm = TRUE))
a <- length(mean(5 * vecLongName + vecLongName)) # this is a comment
# that goes over the line by a good long ways
a <- length(mean(5 * vecLongName + vecLongName - exp(vecLongName) +
vecLongName, na.rm = TRUE)) # this is a comment that goes over the
# line by a good long long long long long long long long long ways
```

8 References

It's possible to insert references and a bibliography into a Jupyter document but it's slightly involved so we won't go into it here.