My Example Computed Manuscript

Created in Rmarkdown

Jeffrey M. Perkel, Technology Editor, Nature

01 Feb 2022 17:18:55 MST

A mock computed manuscript created in RStudio using {Rmarkdown}. The {Bookdown} and {Rticles} packages were used to output the text in Springer Nature’s desired manuscript format.

# 1 Introduction

“Literate programming” is a style of programming that uses computational notebooks to weave together code, explanatory text, data and results into a single document, enhancing scientific communication and computational reproducibility.1–3 (These references were added into the document using RStudio’s integration with the open-source Zotero reference manager4 plus the [Better BibTeX](https://retorque.re/zotero-better-bibtex/) Zotero plugin.)

Several platforms for creating such documents exist.5 Typically, these documents interleave code and text ‘blocks’ to build a computational narrative. But some, including [R Markdown](https://rmarkdown.rstudio.com/), [Observable](https://www.observablehq.com), and the [Jupyter Book](https://jupyterbook.org/intro.html) extension to the Jupyter ecosystem, also allow authors to include and execute code “inline” – that is, within the text itself.

This makes it possible to create fully executable manuscripts in which the document itself computes and inserts values and figures into the text rather than requiring authors to input them manually. This is in many ways the ‘killer feature’ of computed manuscripts: it circumvents the possibility that the author will enter an incorrect number, or forget to update a figure or value should new data arise. Among other uses, that allows authors to automatically time-stamp their documents, or insert the current version number of the software they use into their methods. For instance, this document was built at **01 Feb 2022 17:18:56 MST** and calls the following R packages: {tidyverse} ver. **1.3.0**, {ggbeeswarm} ver. **0.6.0** and {bookdown} ver. **0.24**.

In this manuscript, created in RStudio using the R Markdown language, we will demonstrate a more practical example. (A JavaScript version is available at <https://observablehq.com/d/7a08166446057936>.)

# 2 Results

## 2.1 Inline computation

Imagine we are analyzing data from a clinical trial. We have grouped subjects in three bins and measured the concentration of some metabolite. (These data are simulated.)

Rather than analyzing those data and then copying the results into our manuscript, we can use the programming language R to do that in the manuscript itself. Simply enclose the code inside backticks, with the letter r. For instance, we could calculate the circumference and area of a circle:

You could write “A = `r pi \* r^2` and C = `r 2 \* pi \* r`”. Plugging in the radius *r* = **10**, that evaluates to “A = **314.16** and C = **62.83**”.

Returning to our dataset, we can count the rows in our table to determine the number of samples, and insert that into the text. Thus, we have **99** (simulated) subjects in our study (see Table 3.1; see [R/mock\_data.R](https://github.com/jperkel/computed_manuscript/blob/main/R/mock_data.R) in the GitHub repository for code to generate the mock dataset). Note that the tables, figures and sections in this document are numbered automatically thanks to the {bookdown} package.

The average metabolite concentration in our dataset is **185.36** (range: **78 to 298**). We have **32** subjects in Group 1, **43** subjects in Group 2, and **24** in Group 3. (The numbers in **bold face type** throughout this document are computed values.)

## 2.2 Incorporating new data

Now suppose we get another tranche of data (Table 3.2). There are **60** subjects in this new dataset. Their average concentration is **185.13** (range: **77 to 299**).

Combining the two datasets, we have a total of **159** subjects. The revised average metabolite concentration is **185.28** (range: **77 to 299**). We now have **55** subjects in Group 1, **60** subjects in Group 2, and **44** in Group 3. The concentration distribution for each group is shown graphically in Figure 2.1.

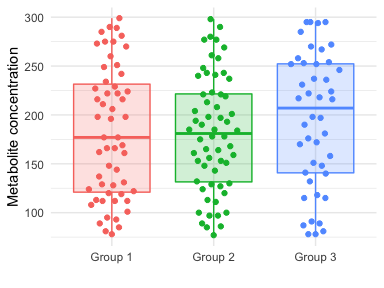


Figure 2.1: Metabolite concentration of clinical trial subjects

# 3 Code

The following code was used to load, merge, and plot the (simulated) clinical trial data:

# load libraries  
library(tidyverse)  
library(ggbeeswarm)  
library(bookdown)

# read in some initial data  
df1 <- read\_csv('data/example-data-1.csv')

# read new dataset  
df2 <- read\_csv('data/example-data-2.csv')

# merge datasets  
final\_data <- rbind(df1, df2)

# plot the data  
p <- final\_data %>%   
 ggplot(aes(x = class, y = conc, fill = class, color = class)) +  
 geom\_boxplot(outlier.shape = NA, alpha = 0.2) +  
 ggbeeswarm::geom\_quasirandom(width = 0.25) +   
 xlab("") +  
 ylab("Metabolite concentration") +   
 theme\_minimal() +  
 theme(legend.position = "none")  
p

Table 3.1: Initial subject data

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Class | Conc | | | ID | Class | Conc | | | ID | Class | Conc |
| ID\_1 | Group 2 | 153 | | | ID\_34 | Group 2 | 221 | | | ID\_67 | Group 3 | 148 |
| ID\_2 | Group 1 | 224 | | | ID\_35 | Group 1 | 112 | | | ID\_68 | Group 1 | 281 |
| ID\_3 | Group 2 | 127 | | | ID\_36 | Group 3 | 246 | | | ID\_69 | Group 3 | 295 |
| ID\_4 | Group 2 | 194 | | | ID\_37 | Group 2 | 190 | | | ID\_70 | Group 2 | 111 |
| ID\_5 | Group 1 | 251 | | | ID\_38 | Group 1 | 177 | | | ID\_71 | Group 2 | 132 |
| ID\_6 | Group 1 | 81 | | | ID\_39 | Group 1 | 148 | | | ID\_72 | Group 2 | 261 |
| ID\_7 | Group 2 | 100 | | | ID\_40 | Group 2 | 290 | | | ID\_73 | Group 1 | 122 |
| ID\_8 | Group 1 | 270 | | | ID\_41 | Group 2 | 151 | | | ID\_74 | Group 2 | 124 |
| ID\_9 | Group 2 | 100 | | | ID\_42 | Group 2 | 159 | | | ID\_75 | Group 1 | 234 |
| ID\_10 | Group 1 | 161 | | | ID\_43 | Group 2 | 113 | | | ID\_76 | Group 2 | 184 |
| ID\_11 | Group 3 | 158 | | | ID\_44 | Group 1 | 249 | | | ID\_77 | Group 3 | 272 |
| ID\_12 | Group 3 | 118 | | | ID\_45 | Group 1 | 124 | | | ID\_78 | Group 1 | 242 |
| ID\_13 | Group 2 | 143 | | | ID\_46 | Group 3 | 87 | | | ID\_79 | Group 2 | 277 |
| ID\_14 | Group 2 | 258 | | | ID\_47 | Group 1 | 166 | | | ID\_80 | Group 3 | 236 |
| ID\_15 | Group 3 | 224 | | | ID\_48 | Group 1 | 196 | | | ID\_81 | Group 1 | 101 |
| ID\_16 | Group 3 | 254 | | | ID\_49 | Group 1 | 112 | | | ID\_82 | Group 3 | 218 |
| ID\_17 | Group 3 | 190 | | | ID\_50 | Group 1 | 289 | | | ID\_83 | Group 2 | 130 |
| ID\_18 | Group 2 | 148 | | | ID\_51 | Group 2 | 161 | | | ID\_84 | Group 1 | 128 |
| ID\_19 | Group 1 | 89 | | | ID\_52 | Group 3 | 270 | | | ID\_85 | Group 3 | 252 |
| ID\_20 | Group 2 | 89 | | | ID\_53 | Group 2 | 237 | | | ID\_86 | Group 1 | 198 |
| ID\_21 | Group 3 | 253 | | | ID\_54 | Group 2 | 280 | | | ID\_87 | Group 1 | 169 |
| ID\_22 | Group 3 | 231 | | | ID\_55 | Group 2 | 175 | | | ID\_88 | Group 2 | 185 |
| ID\_23 | Group 1 | 112 | | | ID\_56 | Group 2 | 223 | | | ID\_89 | Group 1 | 216 |
| ID\_24 | Group 2 | 277 | | | ID\_57 | Group 3 | 295 | | | ID\_90 | Group 2 | 185 |
| ID\_25 | Group 2 | 197 | | | ID\_58 | Group 1 | 275 | | | ID\_91 | Group 2 | 97 |
| ID\_26 | Group 2 | 208 | | | ID\_59 | Group 2 | 120 | | | ID\_92 | Group 2 | 165 |
| ID\_27 | Group 2 | 193 | | | ID\_60 | Group 1 | 78 | | | ID\_93 | Group 3 | 89 |
| ID\_28 | Group 3 | 141 | | | ID\_61 | Group 3 | 78 | | | ID\_94 | Group 2 | 221 |
| ID\_29 | Group 1 | 206 | | | ID\_62 | Group 3 | 140 | | | ID\_95 | Group 1 | 162 |
| ID\_30 | Group 2 | 168 | | | ID\_63 | Group 3 | 294 | | | ID\_96 | Group 1 | 131 |
| ID\_31 | Group 2 | 298 | | | ID\_64 | Group 3 | 295 | | | ID\_97 | Group 1 | 93 |
| ID\_32 | Group 1 | 144 | | | ID\_65 | Group 3 | 285 | | | ID\_98 | Group 2 | 240 |
| ID\_33 | Group 2 | 241 | | | ID\_66 | Group 2 | 129 | | | ID\_99 | Group 2 | 86 |

Table 3.2: New subject data

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Class | Conc | | | ID | Class | Conc | | | ID | Class | Conc |
| ID\_100 | Group 2 | 219 | | | ID\_120 | Group 2 | 85 | | | ID\_140 | Group 2 | 77 |
| ID\_101 | Group 2 | 243 | | | ID\_121 | Group 3 | 181 | | | ID\_141 | Group 1 | 299 |
| ID\_102 | Group 2 | 213 | | | ID\_122 | Group 3 | 216 | | | ID\_142 | Group 3 | 222 |
| ID\_103 | Group 1 | 177 | | | ID\_123 | Group 1 | 222 | | | ID\_143 | Group 1 | 85 |
| ID\_104 | Group 3 | 197 | | | ID\_124 | Group 3 | 252 | | | ID\_144 | Group 1 | 273 |
| ID\_105 | Group 2 | 198 | | | ID\_125 | Group 1 | 166 | | | ID\_145 | Group 3 | 115 |
| ID\_106 | Group 1 | 120 | | | ID\_126 | Group 2 | 204 | | | ID\_146 | Group 1 | 290 |
| ID\_107 | Group 3 | 170 | | | ID\_127 | Group 2 | 243 | | | ID\_147 | Group 2 | 269 |
| ID\_108 | Group 3 | 78 | | | ID\_128 | Group 3 | 198 | | | ID\_148 | Group 2 | 97 |
| ID\_109 | Group 1 | 129 | | | ID\_129 | Group 1 | 119 | | | ID\_149 | Group 1 | 229 |
| ID\_110 | Group 1 | 137 | | | ID\_130 | Group 1 | 198 | | | ID\_150 | Group 3 | 176 |
| ID\_111 | Group 3 | 217 | | | ID\_131 | Group 3 | 151 | | | ID\_151 | Group 2 | 164 |
| ID\_112 | Group 1 | 227 | | | ID\_132 | Group 3 | 115 | | | ID\_152 | Group 3 | 172 |
| ID\_113 | Group 3 | 81 | | | ID\_133 | Group 3 | 237 | | | ID\_153 | Group 1 | 222 |
| ID\_114 | Group 2 | 248 | | | ID\_134 | Group 2 | 178 | | | ID\_154 | Group 1 | 285 |
| ID\_115 | Group 1 | 211 | | | ID\_135 | Group 1 | 275 | | | ID\_155 | Group 2 | 153 |
| ID\_116 | Group 1 | 113 | | | ID\_136 | Group 2 | 178 | | | ID\_156 | Group 3 | 132 |
| ID\_117 | Group 1 | 216 | | | ID\_137 | Group 3 | 267 | | | ID\_157 | Group 2 | 156 |
| ID\_118 | Group 3 | 91 | | | ID\_138 | Group 1 | 95 | | | ID\_158 | Group 1 | 260 |
| ID\_119 | Group 3 | 258 | | | ID\_139 | Group 1 | 108 | | | ID\_159 | Group 2 | 201 |

# 4 Colophon

This manuscript was built at **01 Feb 2022 17:18:57 MST** using the following computational environment and dependencies:

## R version 4.0.4 (2021-02-15)  
## Platform: x86\_64-apple-darwin17.0 (64-bit)  
## Running under: macOS Big Sur 10.16  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] bookdown\_0.24 ggbeeswarm\_0.6.0 forcats\_0.5.1 stringr\_1.4.0   
## [5] dplyr\_1.0.5 purrr\_0.3.4 readr\_2.1.1 tidyr\_1.1.3   
## [9] tibble\_3.1.6 ggplot2\_3.3.3 tidyverse\_1.3.0   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_1.0.7 lubridate\_1.8.0 assertthat\_0.2.1 digest\_0.6.29   
## [5] utf8\_1.2.2 R6\_2.5.1 cellranger\_1.1.0 backports\_1.2.1   
## [9] reprex\_2.0.0 evaluate\_0.14 highr\_0.9 httr\_1.4.2   
## [13] pillar\_1.6.4 rlang\_0.4.12 readxl\_1.3.1 rstudioapi\_0.13   
## [17] rmarkdown\_2.11 labeling\_0.4.2 bit\_4.0.4 munsell\_0.5.0   
## [21] broom\_0.7.6 compiler\_4.0.4 vipor\_0.4.5 modelr\_0.1.8   
## [25] xfun\_0.29 pkgconfig\_2.0.3 htmltools\_0.5.2 tidyselect\_1.1.1  
## [29] fansi\_1.0.0 crayon\_1.4.2 tzdb\_0.2.0 dbplyr\_2.1.0   
## [33] withr\_2.4.3 grid\_4.0.4 jsonlite\_1.7.2 gtable\_0.3.0   
## [37] lifecycle\_1.0.1 DBI\_1.1.1 magrittr\_2.0.1 scales\_1.1.1   
## [41] cli\_3.1.0 stringi\_1.7.6 vroom\_1.5.7 farver\_2.1.0   
## [45] fs\_1.5.2 xml2\_1.3.3 ellipsis\_0.3.2 generics\_0.1.1   
## [49] vctrs\_0.3.8 tools\_4.0.4 bit64\_4.0.5 glue\_1.6.0   
## [53] beeswarm\_0.4.0 hms\_1.1.1 parallel\_4.0.4 fastmap\_1.1.0   
## [57] yaml\_2.2.1 colorspace\_2.0-0 rvest\_1.0.2 knitr\_1.37   
## [61] haven\_2.3.1

The current Git commit details are:

## [34f14f2] 2022-02-02: Add a generic function to print a table 3cols wide

# References

1. Shen, H. Interactive notebooks: Sharing the code. *Nature* **515**, 151–152 (2014).

2. Perkel, J. M. A toolkit for data transparency takes shape. *Nature* **560**, 513–515 (2018).

3. Perkel, J. M. Why Jupyter is data scientists’ computational notebook of choice. *Nature* **563**, 145–146 (2018).

4. Perkel, J. M. Streamline your writing and collaborations with these reference managers. *Nature* **585**, 149–150 (2020).

5. Perkel, J. M. Reactive, reproducible, collaborative: computational notebooks evolve. *Nature* **593**, 156–157 (2021).