

# My Example Computed Manuscript

## Created in Rmarkdown

Jeffrey M. Perkel ·

Received: date / Accepted: date

**Abstract** 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.

**Keywords**

## 1 Introduction

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Here we’ll add some references from Zotero (Perkel 2020): (Fisch et al. 2015; Argelaguet et al. 2021; Lê Cao et al. 2021).

Markdown documents can include inline equations written in L<sup>A</sup>T<sub>E</sub>X, such as  $F = ma$ . Here is an equation on its own line:

$$a^2 + b^2 = c^2$$

## 2 Results

### 2.1 Inline computation

One ‘killer feature’ of computed manuscripts is the ability to compute and insert values into the text rather than requiring authors to input them man-

---

Jeffrey M. Perkel  
Springer Nature, 1 New York Plaza, New York, NY  
E-mail: [jeffrey.perkel@nature.com](mailto:jeffrey.perkel@nature.com)

**Table 1** Original subject data

ID	class	wt	—	ID	class	wt	—	ID	class	wt
1	Group 1	218	—	34	Group 2	251	—	67	Group 2	134
2	Group 2	239	—	35	Group 2	240	—	68	Group 1	169
3	Group 3	292	—	36	Group 2	252	—	69	Group 2	152
4	Group 3	220	—	37	Group 2	165	—	70	Group 2	218
5	Group 3	287	—	38	Group 3	278	—	71	Group 2	297
6	Group 1	75	—	39	Group 2	176	—	72	Group 3	167
7	Group 1	167	—	40	Group 1	119	—	73	Group 3	136
8	Group 2	252	—	41	Group 1	159	—	74	Group 2	178
9	Group 1	88	—	42	Group 2	243	—	75	Group 1	266
10	Group 1	269	—	43	Group 2	111	—	76	Group 3	164
11	Group 1	90	—	44	Group 1	209	—	77	Group 1	170
12	Group 3	209	—	45	Group 2	139	—	78	Group 3	214
13	Group 2	91	—	46	Group 1	235	—	79	Group 3	105
14	Group 2	107	—	47	Group 1	170	—	80	Group 3	201
15	Group 1	78	—	48	Group 1	299	—	81	Group 2	124
16	Group 2	253	—	49	Group 1	244	—	82	Group 2	119
17	Group 3	130	—	50	Group 1	261	—	83	Group 2	255
18	Group 1	99	—	51	Group 3	251	—	84	Group 2	75
19	Group 1	283	—	52	Group 1	280	—	85	Group 3	296
20	Group 2	222	—	53	Group 1	295	—	86	Group 2	101
21	Group 3	113	—	54	Group 2	271	—	87	Group 1	77
22	Group 3	237	—	55	Group 1	189	—	88	Group 2	184
23	Group 3	237	—	56	Group 3	149	—	89	Group 2	246
24	Group 2	216	—	57	Group 3	284	—	90	Group 3	176
25	Group 1	200	—	58	Group 1	182	—	91	Group 2	123
26	Group 1	250	—	59	Group 2	80	—	92	Group 2	150
27	Group 1	136	—	60	Group 1	138	—	93	Group 3	142
28	Group 1	196	—	61	Group 1	144	—	94	Group 3	161
29	Group 1	105	—	62	Group 1	91	—	95	Group 3	292
30	Group 2	163	—	63	Group 3	189	—	96	Group 3	168
31	Group 1	121	—	64	Group 3	79	—	97	Group 1	232
32	Group 1	243	—	65	Group 1	84	—	98	Group 3	144
33	Group 3	95	—	66	Group 2	170	—	99	Group 1	204

ually. That circumvents the possibility that the author will enter an incorrect number, or forget to update them should new data arise.

For instance, imagine we are analyzing data from a clinical trial. We have grouped subjects in three bins and measured their weights.

Rather than analyzing those data and then copying them into our manuscript, we can use the programming language R to do that in the manuscript itself. Simply enclose the code inside backticks: ``r <code to evaluate>``. For instance, to calculate the circumference and area of a circle with radius  $r = 10$ , you could write “`A = `r pi * r^2``” and “`C = `r 2 * pi * r``”. Those give  $A = 314.159$  and  $C = 62.832$ .

**Table 2** New subject data

ID	class	wt	—	ID	class	wt	—	ID	class	wt
1	Group 3	201	—	21	Group 2	290	—	41	Group 1	280
2	Group 2	299	—	22	Group 3	139	—	42	Group 1	146
3	Group 1	272	—	23	Group 1	257	—	43	Group 2	278
4	Group 1	294	—	24	Group 1	264	—	44	Group 1	142
5	Group 2	125	—	25	Group 2	99	—	45	Group 1	92
6	Group 3	232	—	26	Group 2	212	—	46	Group 1	273
7	Group 1	115	—	27	Group 2	103	—	47	Group 1	104
8	Group 3	144	—	28	Group 3	253	—	48	Group 1	137
9	Group 2	289	—	29	Group 1	171	—	49	Group 3	97
10	Group 3	163	—	30	Group 3	196	—	50	Group 1	132
11	Group 1	200	—	31	Group 3	160	—	51	Group 1	279
12	Group 3	104	—	32	Group 2	228	—	52	Group 3	162
13	Group 1	253	—	33	Group 2	205	—	53	Group 1	291
14	Group 1	142	—	34	Group 3	78	—	54	Group 1	196
15	Group 1	174	—	35	Group 1	202	—	55	Group 1	188
16	Group 1	150	—	36	Group 3	210	—	56	Group 1	108
17	Group 3	177	—	37	Group 1	208	—	57	Group 3	119
18	Group 1	199	—	38	Group 3	273	—	58	Group 3	148
19	Group 3	216	—	39	Group 3	244	—	59	Group 3	103
20	Group 2	278	—	40	Group 2	243	—	60	Group 1	85

We have **99** subjects in our study (Table 1). The average weight is **184.32** (range: **75-299**). We have **38** subjects in Group 1, **33** subjects in Group 2, and **28** in Group 3. (The numbers in **bold face type** are computed values.)

Now suppose we get another tranche of data:

There are **60** subjects in this new dataset (Table 2). Their average weight is **190.37** (range: **78-299**).

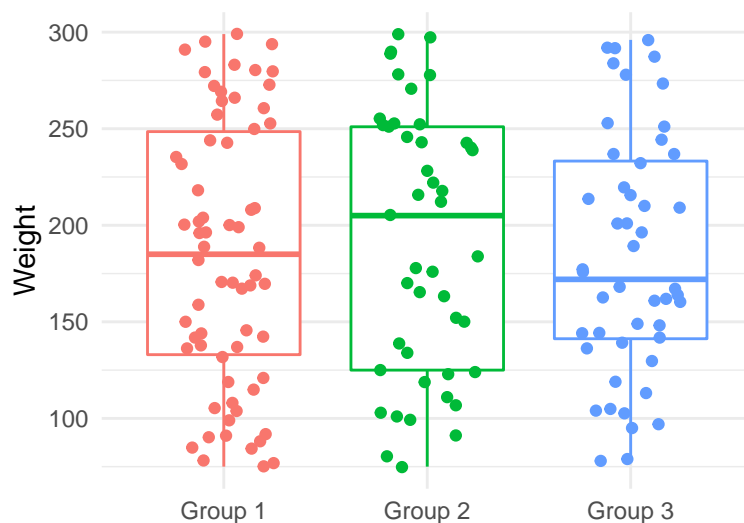
Combining the two datasets, we have a total of **159** subjects. The revised average weight is **186.6** (range: **75-299**). We now have **66** subjects in Group 1, **45** subjects in Group 2, and **48** in Group 3.

## 2.2 Plotting the data

As Rmarkdown documents can do anything R can do, we can also create and include figures. Here we plot boxplots of our clinical trial data. The data are shown in Figure 1. Note that this figure number (as well as the table numbers above) are automatically generated.

## 3 Methods

The following code was used in section 2.1 to create the original clinical trial data:



**Fig. 1** Weight distribution of clinical trial subjects

```
# create a mock dataset
n_subjs <- 99
subjID <- 1:n_subjs

# generate 99 random #s between 1 and 10
tmp <- floor(runif(n_subjs, min = 0, max = 9))
# assign those numbers to any of 3 subject groups
fn <- function(x) {
  if (x >= 6) 'Group 1'
  else if (x >= 3) 'Group 2'
  else 'Group 3'
}
subj_class <- sapply(tmp, fn)

# pick random weights between 75 and 300
wts <- floor(runif(n_subjs, min = 75, max = 300))
# combine the data into a table
df <- data.frame(ID = subjID, class = subj_class, wt = wts)

# display the table, splitting the 99 rows into 3 cols wide
tmp <- cbind(df[1:33,], rep('|', 33),
             df[34:66,], rep('|', 33),
             df[67:99,])
names(tmp) <- c('ID', 'class', 'wt', '|', 'ID', 'class', 'wt',
```

```
      '|', 'ID', 'class', 'wt')
knitr::kable(tmp, format = 'latex', booktabs = TRUE,
             caption = "Original subject data")
```

The following code was used in section 2.2 to plot the data:

```
final_data %>%
  ggplot(aes(x = class, y = wt, color = class)) +
  geom_boxplot() +
  geom_jitter(width = 0.25) +
  xlab("") +
  ylab("Weight") +
  theme_minimal() +
  theme(legend.position = "none")
```

## References

- Argelaguet, Ricard, Anna S. E. Cuomo, Oliver Stegle, and John C. Marioni. 2021. "Computational Principles and Challenges in Single-Cell Data Integration." *Nature Biotechnology*, May, 1–14. <https://doi.org/10.1038/s41587-021-00895-7>.
- Fisch, K. M., T. Meissner, L. Gioia, J.-C. Ducom, T. M. Carland, S. Loguercio, and A. I. Su. 2015. "Omics Pipe: A Community-Based Framework for Reproducible Multi-Omics Data Analysis." *Bioinformatics* 31 (11): 1724–28. <https://doi.org/10.1093/bioinformatics/btv061>.
- Lê Cao, Kim-Anh, Al J. Abadi, Emily F. Davis-Marcisak, Lauren Hsu, Arshi Arora, Alexis Coullomb, Atul Deshpande, et al. 2021. "Community-Wide Hackathons to Identify Central Themes in Single-Cell Multi-Omics." *Genome Biology* 22 (1): 220. <https://doi.org/10.1186/s13059-021-02433-9>.
- Perkel, Jeffrey M. 2020. "Streamline Your Writing — and Collaborations — with These Reference Managers." *Nature* 585 (7823): 149–50. <https://doi.org/10.1038/d41586-020-02491-2>.