

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^AT_EX, 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

Table 1 Original subject data

ID	class	wt	—	ID	class	wt	—	ID	class	wt
1	Group 2	134	—	34	Group 1	183	—	67	Group 3	226
2	Group 1	95	—	35	Group 1	234	—	68	Group 1	124
3	Group 3	160	—	36	Group 3	194	—	69	Group 2	75
4	Group 2	269	—	37	Group 2	106	—	70	Group 2	178
5	Group 1	204	—	38	Group 2	247	—	71	Group 1	212
6	Group 2	145	—	39	Group 2	120	—	72	Group 3	77
7	Group 1	167	—	40	Group 2	90	—	73	Group 3	146
8	Group 3	271	—	41	Group 1	206	—	74	Group 1	263
9	Group 1	107	—	42	Group 2	263	—	75	Group 2	147
10	Group 1	115	—	43	Group 2	208	—	76	Group 3	236
11	Group 2	236	—	44	Group 3	134	—	77	Group 1	79
12	Group 1	80	—	45	Group 1	154	—	78	Group 2	293
13	Group 3	208	—	46	Group 1	143	—	79	Group 1	211
14	Group 2	189	—	47	Group 2	87	—	80	Group 2	161
15	Group 3	215	—	48	Group 1	112	—	81	Group 2	296
16	Group 2	206	—	49	Group 1	84	—	82	Group 2	89
17	Group 3	273	—	50	Group 1	132	—	83	Group 3	204
18	Group 3	240	—	51	Group 1	172	—	84	Group 2	224
19	Group 1	171	—	52	Group 3	241	—	85	Group 1	259
20	Group 1	104	—	53	Group 3	152	—	86	Group 2	165
21	Group 1	241	—	54	Group 3	244	—	87	Group 1	284
22	Group 2	260	—	55	Group 1	189	—	88	Group 2	103
23	Group 3	134	—	56	Group 1	89	—	89	Group 3	218
24	Group 3	237	—	57	Group 3	198	—	90	Group 2	207
25	Group 1	214	—	58	Group 1	178	—	91	Group 1	252
26	Group 1	114	—	59	Group 1	110	—	92	Group 1	206
27	Group 1	102	—	60	Group 2	127	—	93	Group 2	79
28	Group 2	272	—	61	Group 2	160	—	94	Group 1	290
29	Group 3	201	—	62	Group 3	210	—	95	Group 2	199
30	Group 1	282	—	63	Group 1	116	—	96	Group 1	181
31	Group 2	165	—	64	Group 3	158	—	97	Group 1	202
32	Group 2	238	—	65	Group 1	158	—	98	Group 1	213
33	Group 3	252	—	66	Group 1	120	—	99	Group 3	176

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 programmatically and then copying them into our manuscript, we can use the programming language R to do that in the manuscript itself. 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 $C = 62.832$.

We have **99** subjects in our study (Table 1). The average weight is **180.76** (range: **75-296**). We have **42** subjects in Group 1, **32** subjects in Group 2, and **25** in Group 3. (The numbers in **bold face type** are computed values.)

Table 2 New subject data

ID	class	wt	—	ID	class	wt	—	ID	class	wt
1	Group 3	256	—	21	Group 3	104	—	41	Group 3	144
2	Group 3	296	—	22	Group 2	227	—	42	Group 3	105
3	Group 3	291	—	23	Group 2	144	—	43	Group 3	89
4	Group 2	258	—	24	Group 3	250	—	44	Group 3	112
5	Group 1	78	—	25	Group 2	194	—	45	Group 3	88
6	Group 1	238	—	26	Group 2	90	—	46	Group 1	299
7	Group 1	121	—	27	Group 1	75	—	47	Group 2	85
8	Group 3	96	—	28	Group 1	213	—	48	Group 1	292
9	Group 2	120	—	29	Group 1	122	—	49	Group 3	267
10	Group 1	273	—	30	Group 2	178	—	50	Group 3	159
11	Group 3	100	—	31	Group 2	155	—	51	Group 1	147
12	Group 1	106	—	32	Group 3	249	—	52	Group 1	96
13	Group 3	121	—	33	Group 2	292	—	53	Group 3	172
14	Group 3	185	—	34	Group 1	294	—	54	Group 1	299
15	Group 3	223	—	35	Group 3	264	—	55	Group 3	189
16	Group 3	205	—	36	Group 3	133	—	56	Group 3	215
17	Group 1	129	—	37	Group 2	183	—	57	Group 2	154
18	Group 3	258	—	38	Group 1	91	—	58	Group 3	291
19	Group 2	130	—	39	Group 3	183	—	59	Group 1	204
20	Group 1	272	—	40	Group 1	141	—	60	Group 2	293

Now suppose we get another tranche of data:

There are **60** subjects in this new dataset (Table 2). Their average weight is **183.97** (range: **75-299**).

Combining the two datasets, we have a total of **159** subjects. The revised average weight is **181.97** (range: **75-299**). We now have **61** subjects in Group 1, **46** subjects in Group 2, and **52** 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:

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

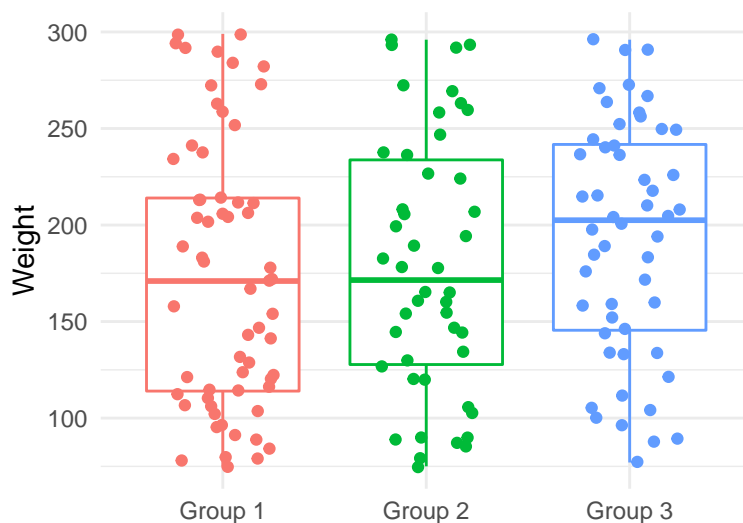


Fig. 1 Weight distribution of clinical trial subjects

```
# generate 99 random #s between 1 and 10
tmp <- floor(runif(n_subjs, min = 1, max = 10))
# assign those numbers to any of 3 subject groups
fn <- function(x) {
  if (x >= 7) 'Group 1'
  else if (x >= 4) '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>.