Noname manuscript No.
(will be inserted by the editor)

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 LaTeX, 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	GRP_3	175	—	34	GRP_2	115	—	67	GRP_1	214
2	GRP_3	125	—	35	GRP_2	121	—	68	GRP_3	251
3	GRP_3	120	—	36	GRP_1	131	—	69	GRP_3	227
4	GRP_3	255	—	37	GRP_3	112	—	70	GRP_1	277
5	GRP_2	287	—	38	GRP_1	115	—	71	GRP_3	210
6	GRP_1	184	—	39	GRP_3	82	—	72	GRP_2	142
7	GRP_2	209	—	40	GRP_2	248	—	73	GRP_3	153
8	GRP_1	227	—	41	GRP_3	91	—	74	GRP_1	275
9	GRP_2	272	—	42	GRP_1	136	—	75	GRP_2	123
10	GRP_2	204	—	43	GRP_3	229	—	76	GRP_2	295
11	GRP_1	109	—	44	GRP_3	96	—	77	GRP_1	256
12	GRP_3	295	—	45	GRP_2	274	—	78	GRP_1	259
13	GRP_2	80	—	46	GRP_1	260	—	79	GRP_2	199
14	GRP_2	157	—	47	GRP_2	94	—	80	GRP_3	161
15	GRP_3	159	—	48	GRP_1	269	—	81	GRP_3	80
16	GRP_1	180	—	49	GRP_2	113	—	82	GRP_1	131
17	GRP_2	297	—	50	GRP_2	104	—	83	GRP_3	189
18	GRP_1	258	—	51	GRP_1	190	—	84	GRP_1	126
19	GRP_2	105	—	52	GRP_1	131	—	85	GRP_1	170
20	GRP_1	207	—	53	GRP_1	182	—	86	GRP_1	108
21	GRP_1	187	—	54	GRP_1	85	—	87	GRP_1	279
22	GRP_1	88	—	55	GRP_3	239	—	88	GRP_1	147
23	GRP_3	236	—	56	GRP_2	205	—	89	GRP_3	187
24	GRP_2	293	—	57	GRP_1	157	—	90	GRP_2	93
25	GRP_1	121	—	58	GRP_3	262	—	91	GRP_2	182
26	GRP_3	97	—	59	GRP_2	227	—	92	GRP_1	91
27	GRP_2	192	—	60	GRP_2	255	—	93	GRP_2	95
28	GRP_2	204	—	61	GRP_2	110	—	94	GRP_3	288
29	GRP_1	287	—	62	GRP_2	130	—	95	GRP_3	191
30	GRP_2	270	—	63	GRP_1	238	—	96	GRP_1	116
31	GRP_3	179	—	64	GRP_2	245	—	97	GRP_3	196
32	GRP_3	121	—	65	GRP_1	123	—	98	GRP_2	187
33	GRP_3	237	—	66	GRP_1	152	—	99	GRP_1	289

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 **183.08** (range: **80-297**). We have **37** subjects in Group 1, **33** subjects in Group 2, and **29** 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	GRP_3	250	—	21	GRP_1	254	—	41	GRP_2	93
2	GRP_1	108	—	22	GRP_1	205	—	42	GRP_1	122
3	GRP_1	121	—	23	GRP_1	197	—	43	GRP_3	207
4	GRP_3	284	—	24	GRP_3	90	—	44	GRP_3	256
5	GRP_2	243	—	25	GRP_3	154	—	45	GRP_2	292
6	GRP_1	260	—	26	GRP_2	254	—	46	GRP_1	120
7	GRP_2	297	—	27	GRP_3	251	—	47	GRP_1	144
8	GRP_1	293	—	28	GRP_2	191	—	48	GRP_2	190
9	GRP_3	146	—	29	GRP_3	182	—	49	GRP_3	156
10	GRP_3	244	—	30	GRP_3	103	—	50	GRP_2	114
11	GRP_3	246	—	31	GRP_3	221	—	51	GRP_3	87
12	GRP_2	163	—	32	GRP_3	236	—	52	GRP_1	99
13	GRP_3	101	—	33	GRP_3	210	—	53	GRP_3	288
14	GRP_1	168	—	34	GRP_1	143	—	54	GRP_2	85
15	GRP_2	289	—	35	GRP_2	176	—	55	GRP_1	124
16	GRP_2	166	—	36	GRP_2	243	—	56	GRP_2	89
17	GRP_2	81	—	37	GRP_3	256	—	57	GRP_1	124
18	GRP_3	239	—	38	GRP_2	276	—	58	GRP_1	241
19	GRP_2	149	—	39	GRP_3	155	—	59	GRP_2	221
20	GRP_3	252	—	40	GRP_1	173	—	60	GRP_2	104

Now suppose we get another tranche of data:

There are **60** subjects in this new dataset (Table 2). Their average weight is **187.1** (range: **81-297**).

Combining the two datasets, we have a total of **159** subjects. The revised average weight is **184.6** (range: **80-297**). We now have **54** subjects in Group 1, **53** 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
```

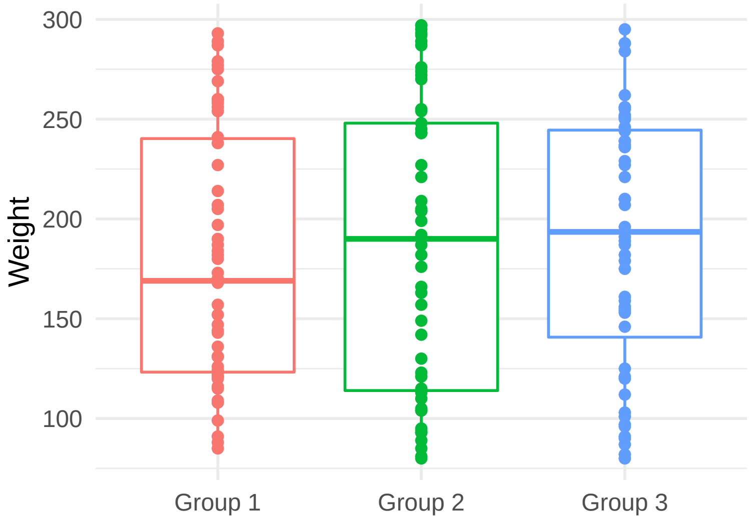

Fig. 1 Weight distribution of clinical trial subjects

```r
# 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) 'GRP_1'
  else if (x >= 4) 'GRP_2'
  else 'GRP_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 %>%
  mutate(class = str_replace(class, '_', ' '),
         class = str_replace(class, 'GRP', 'Group')) %>%
  ggplot(aes(x = class, y = wt, color = class)) +
  geom_boxplot() +
  geom_point() +
  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.