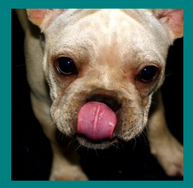
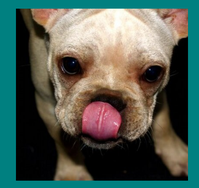


escaping batpigday

Charles T. Gray[1]

[1]Department of Mathematics and Statistics, La Trobe University, Victoria, Australia



batpigday

batpigday *noun* The coding equivalent of groundhogday.

the problem

Simulating data is a bitch.

Debugging frequently dominates the time of students in mathematical science. These students know how to solve equations, and next to nothing about code.

New tools are emerging daily to enable researchers to avoid these timesink pitfalls.
These tools have lowered the programmatic barrier for researchers, but it still a learning curve.

We consider a case study in meta-analysis.

meta-analysis Statistical methodology for combining the results of several studies.

meta-analysis of medians

Conventional meta-analytic tools, such as `metafor::rma`, require an **effect** and a **variance** of that effect.

But what if the reported statistics are **median** and **interquartile** range?

To test our proposed estimator for the variance of the sample median, I found myself repeating tasks and checks in the algorithms.

I tried to find a better way of debugging and writing simulations.
This lead to:

1. a packaged analysis, `varameta::*`, which is built on
2. the simulation package for meta-analysis data, `metasim::*`.

(*in development)

escaping batpigday

coding is the easiest part of coding

- Modular code, break the code into chunks.
- Reproducibility is more than `set.seed()`: accessibility, refactoring, integratability, versioning,

```
library(tidyverse)
library(metasim)
```

Generate sample sizes for k studies.

```
# simulate 2 studies where most have at most 25
sim_n(k = 2, min_n = 10, max_n = 25) %>% output_table()
```

Table 1:

study	group	n
study_1	control	20
study_2	control	15
study_1	intervention	36
study_2	intervention	14

```
# generate simulation dataframe
sim_df() %>% head(2) %>% select(-n) %>% output_table()
```

Table 2:

k	tau2_true	median_ratio	prop	rdist	parameters	id	true_effect
3	0	1	0.3	norm	list(mean = 67, sd = 0.3)	sim_1	67.0
3	0	1	0.3	exp	list(rate = 2)	sim_2	0.3

Each **row** of this dataframe represents a set of **simulation** parameters.

Each simulation runs a **trial** function.

```
metatrial() %>% output_table()
```

Table 3:

conf_low	conf_high	tau_sq	k	effect	measure	true_effect	coverage	bias
23.5	65.0	69.9	3	44.2	m	50.0	TRUE	-5.8
-0.5	1.4	0.2	3	0.5	lr	0.2	TRUE	0.3

Each **simulation** runs the trial function |trials| times.

```
metasim() %>% pluck("results") %>% select(-coverage_count) %>%  output_table()
```

Table 4:

measure	tau_sq	ci_width	bias	successful_trials	coverage	id
lr	0.7	3.7	0.0	4	1	simulation1
m	668.4	110.0	4.7	4	1	simulation1

the maths

references

This poster was created with `posterdown::`.