Code availability is not enough:
A schema-based representation for comprehensible a extensible data analysis

The University of Texas at Austin
Department of Statistics and Data Sciences
College of Natural Sciences

huizezhang-sherry/tines

H. Sherry Zhang & Roger D. Peng | Department of Statistics and Data Sciences, University of Texas at Austin, USA

Introduction

Does reproducibility practices make it easier to understand a data analysis??

An independent researcher should be able to, without much difficulty,

1) track decisions made by the authors,
2) understand the rationale behind those decisions, and
3) comfortably modify the code to explore alternative decision choices.

Contribution

1) an intermediate representation to describe data analysis, and

2) an accompanying workflow of working with the schema using LLM to generate code and explore alternatives

Application

Many analyst football study (Silberzahn et al, 2018):

29 research teams | same research question: whether soccer referees are more likely to give red cards to dark-skin-toned player than to light-skin-toned players.

Group 5: Binomial GLMM with random effect of player and referee, random effect of skin tone rating (`rating`) across referee countries:

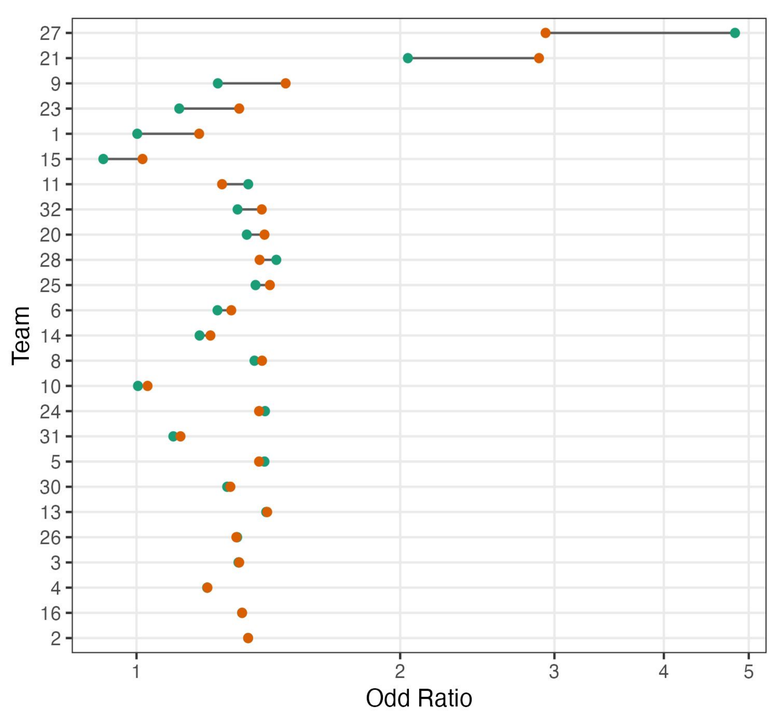
```
glmer(redCards ~ 1 + rating + (1 |
playerShort) + (1| refNum) + (1+rating |
refCountry), family = binomial, …)
```

YAML schema:

```
meta:
  type: schema
nodes:
  - id: select_error_distribution
    objective: which error distribution to
use for modeling the response variable
    decision: use binomial error distribution
    rationale: …
  - id: fit_glmm
    objective: which modelling framework to
use for this analysis
    decision: estimate generalized linear
mixed models
    rationale: …
```

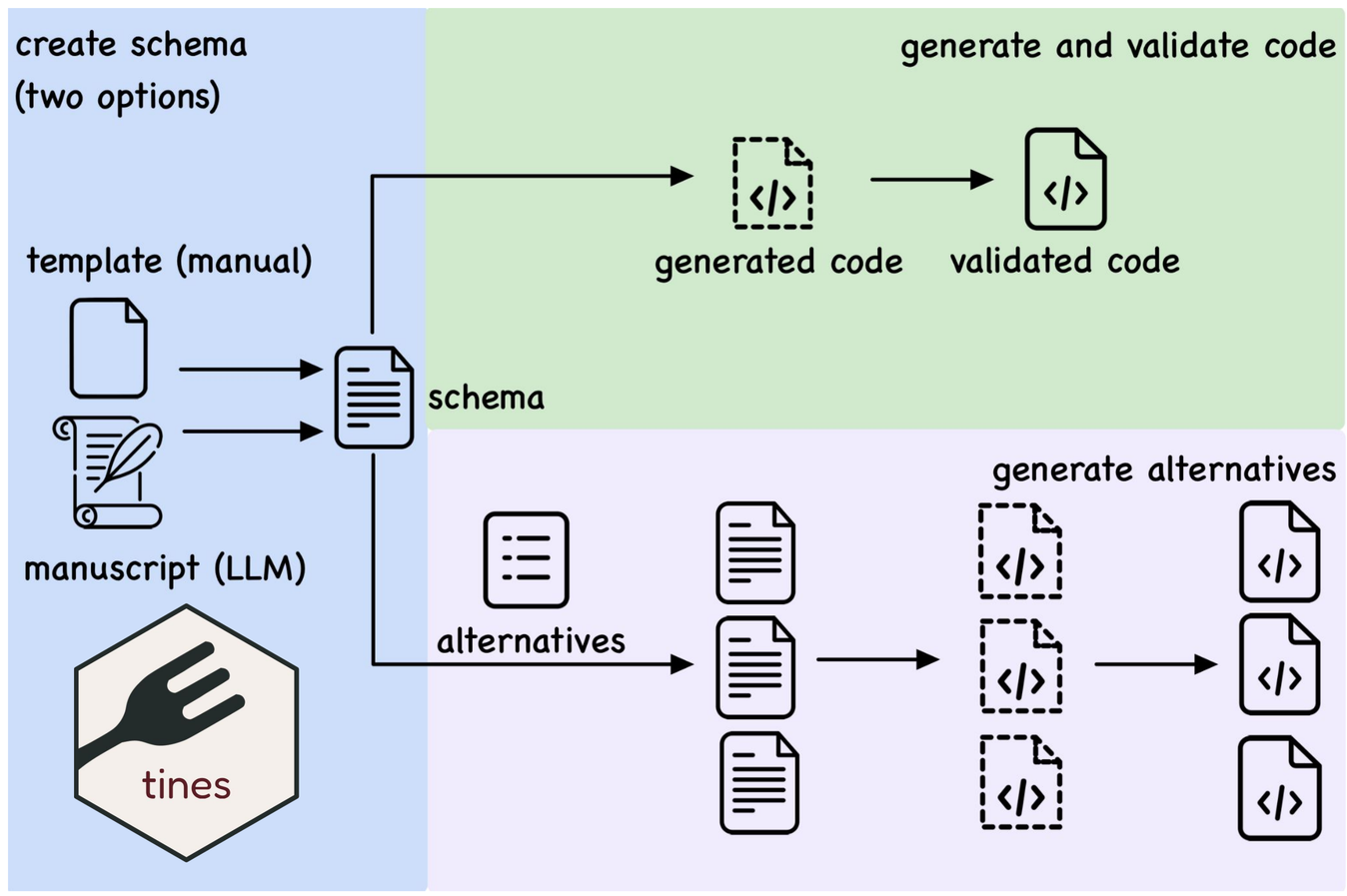
script-mod_iter2.R

```r
# Load required libraries
library(readr)
library(dplyr)
library(glmmTMB)

# Load the data
df <- read_csv("data/football.csv")

# Step 1: create_skin_tone_variable
# Average the two ratings of skin-tone and rescale to crea
# Input: rater1, rater2
# Output: avgrate01
df <- df %>%
  mutate(avgrate01 = (rater1 + rater2) / 2)

# Rescale to 0-1 range
avgrate_min <- min(df$avgrate01, na.rm = TRUE)
avgrate_max <- max(df$avgrate01, na.rm = TRUE)
df <- df %>%
  mutate(avgrate01 = (avgrate01 - avgrate_min) / (avgrate_
```



Workflow

This schema, coupled with LLMs, can be used to generate code scripts from natural-language descriptions in manuscripts, and explore alternative decision choices.



Application

YAML alternatives

```
meta:
  type: alternatives
  id: specify_random_effects_structure
  objective: how to account for ..
alternatives:
  - id: gm1
    decision: a random effect of player and referee
    rationale: …
  - id: gm2
    decision: a random effect of player and referee, and a random
effect of skin-tone across referees"
    rationale: …
  - id: "gm3"
    decision: "a random effect of player, referee, and referees'
countries of origin, and a random effect of skin tone across
referees' countries of origin"
    rationale: …
```

gm1 gm2 gm3

group	term	gm1	gm2	gm3
playerShort	var__(Intercept)	0.3737	0.3738	0.2899
refNum	var__(Intercept)	0.1528	0.1932	0.0748
refNum	var__avgrate01	NA	0.0286	NA
refCountry	var__(Intercept)	NA	NA	0.3990
refCountry	var__avgrate01	NA	NA	0.0870