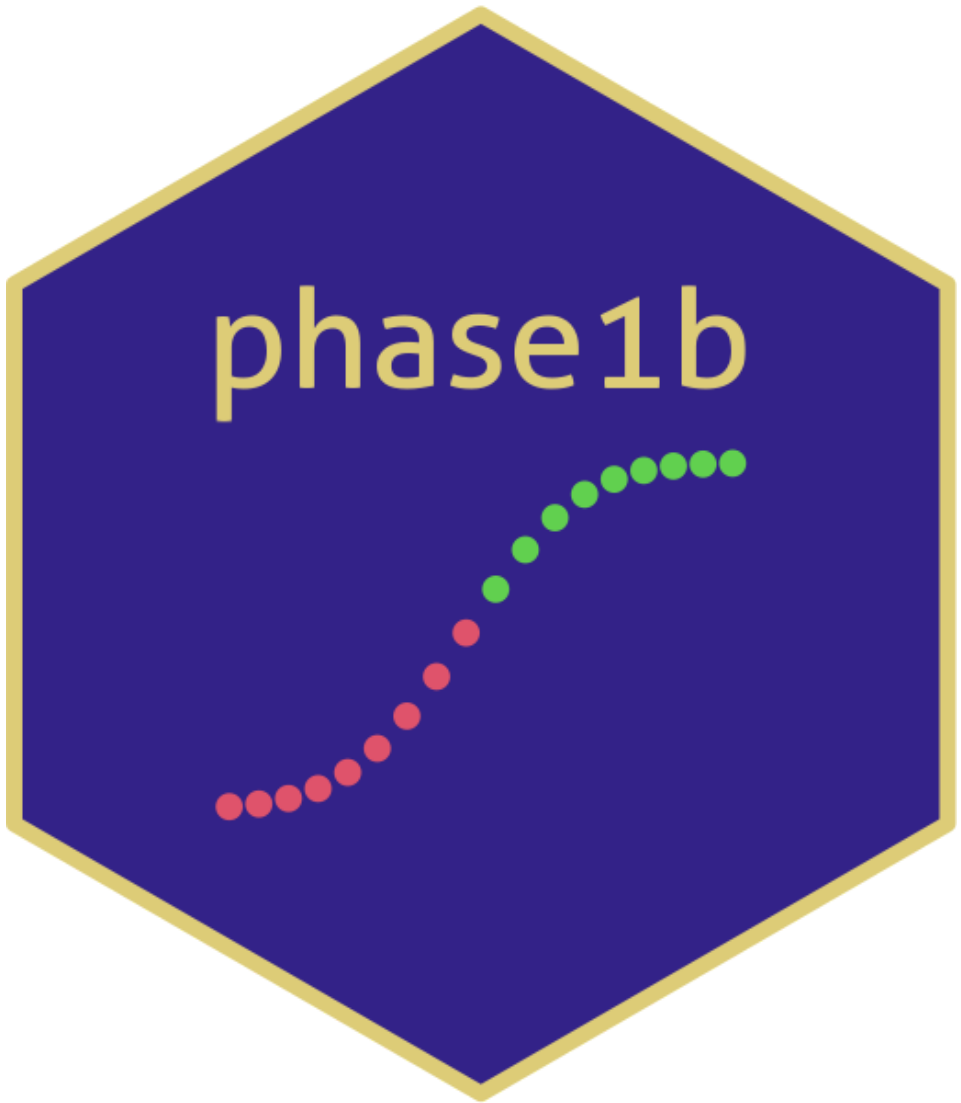
Gearing our industry Statisticians up for Software success : A `phase1b` journey.

Audrey Yeo

1. Opinions do not reflect employer
2. This presentation has ALT text and as much as possible, uses colour-blind friendly palettes
3. Code for this Quarto-rendered .html will also be shared

About phase1b



Audrey Yeo

`phase1b` package history

- 2015 : Started as a need in Roche's early development group, package development led by Daniel Sabanés Bové in 2015.
- 2023 : Refactoring, Renaming, adding Unit and Integration tests as current State-of-Art Software Engineering practice.
- 100% written in R and Open Source.
- website : genentech.github.io/phase1b/

```
1 library(devtools)
2 devtools::install_github("https://github.com/Genentech/phase1b")
3 library(phase1b)
```

Why Should Statisticians make Software ?

"Modern statistical inference is unthinkable without the use of a computer"

Likelihood and Bayesian Inference, Held & Sabanés Bové (2020)

Audrey Yeo

Clinical Trials Statisticians understand :

…the combined Scientific and Business development of a therapy.

- Roadblocks to end points e.g. recruitment rates and failures, business assumptions and strategies
- Drug pharmacokinetics and pharmacodynamics e.g. latent drug effect
- Patient population e.g. rare diseases

My Journey at F. Hoffmann La-Roche

- graduated MSc Biostatistics in 2020
- started at RWD at Roche in mid 2021
- joined R&D at Roche in mid 2022

```
- Project Lead Statistician in early Oncology Trials
- Study Statistician for phase 1-2, phase 3 trials
```

- first internal statistics presentation end 2022 on decision gating
  - skills : R programming, writing functions, version control (GitLab, GitHub), presentation and communication
- started `phase1b` in July 2023
- `phase1b`'s first external tour at PSI (pic later) and useR! in 2024

Why Software Engineering ?

Why find solutions when we can also build them ?

- Mathematics is Elegant
- Building software can be inclusive and collaborative
- I can create delightful experiences users and bring everyone along : *values of inclusion, building great products, having an impact*

Why Software Engineering : A response to dynamic decision making in early development

Audrey Yeo

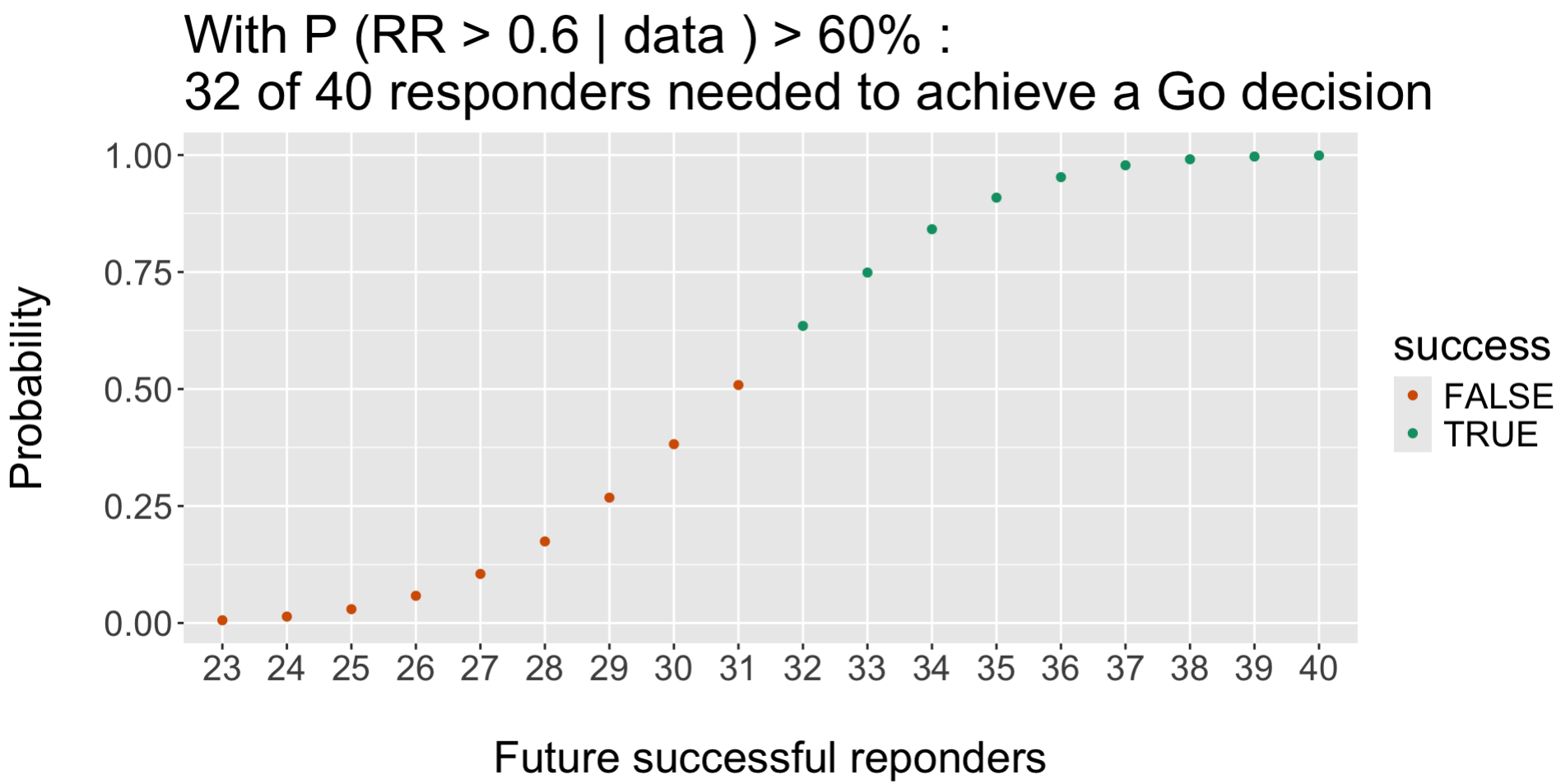


Figure 1: Predictive Posterior CDF for different Efficacy Rules

Skills needed : Covering the basics

- Onboarding June + July 2023
  - R basics: coding and style
  - Good function writing in R
  - R debugging
  - R advanced: package development
    - Basics of Package Development in R
    - Roxygen2 introduction another resource
    - Good Software Engineering Practice for R Packages

in summary, gaps were : Git merging, Pull Requests, Styling, Debugging, Writing Tests

New beginnings :

Knowledge and skills for Statistical Software Engineering!

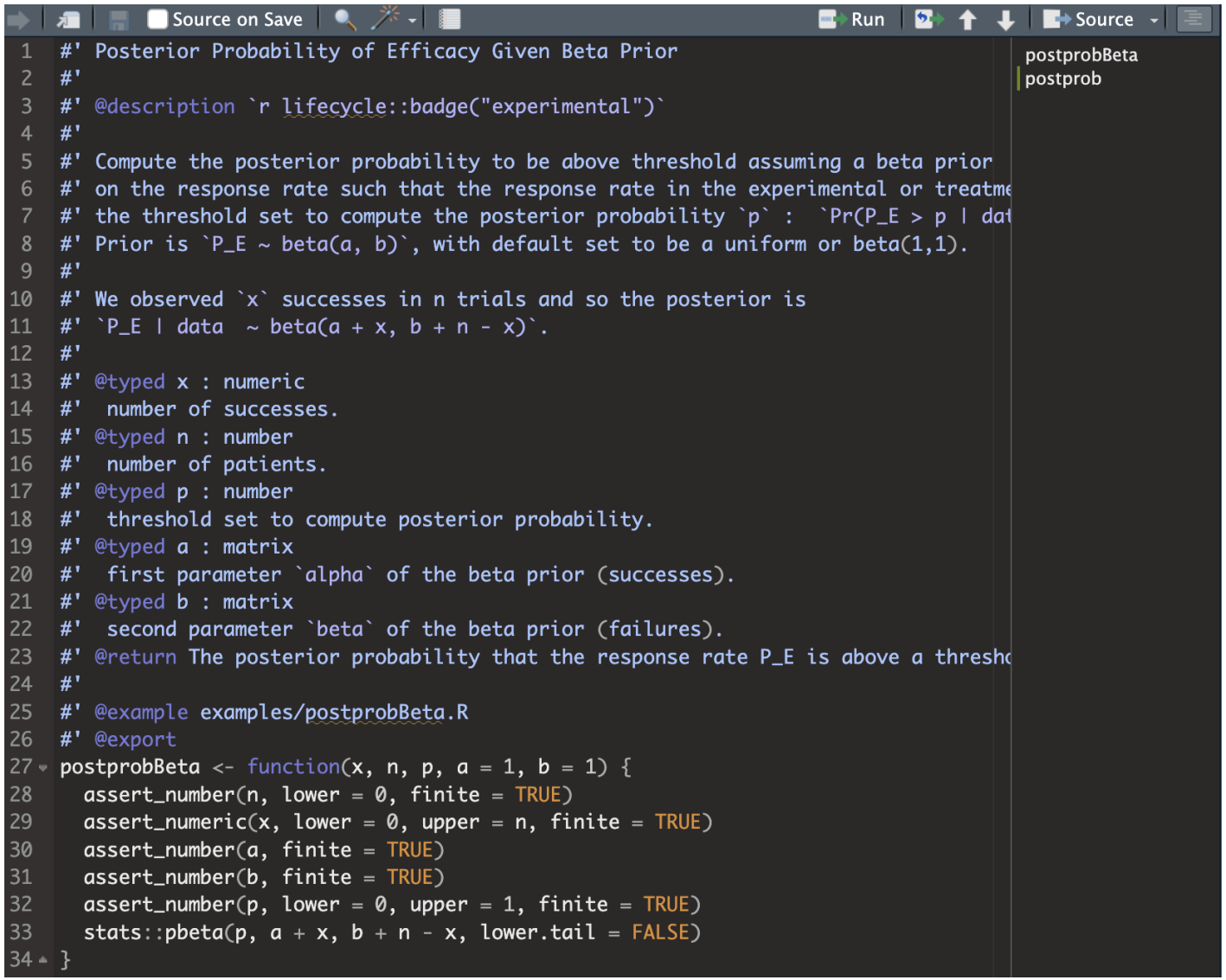
- Visualing branches : `Git kraken`, `VS Code Git Graph Extension`
- Checking : `pre-commit`, `Git hub checks`, `R CMD`
- Documentation : Writing (`roxygen`), building, reviewing documentation
- Testing : Taking a reviewer perspective (`testthat` and `checkmate`)
- Styling : `styler`, `prettier` … building your own style

What is refactoring ?

- “Polishing an antique cup”, making new of “vintage”.

Peranakan bowl from Late 19-20th Century, Kuala Lumpur, British Malaya

Tools for refactoring



Roxygen skeleton example for one user-facing function in phase1b

Tools for refactoring

- Variables are `typed` defined from package `roxygen2` and it's type asserted from package `checkmate`
- Unit and integration testing takes on the perspective of a reviewer and to ensure the function calls calculates what it says it calculates (they are my favourite). Package `testthat` and `checkmate` are used here
- Making `phase1b` a State of Art Software[1] - reproducible, robust, testable, intuitive and open to collaboration

1. Subject to a change in definition with better tools and practices

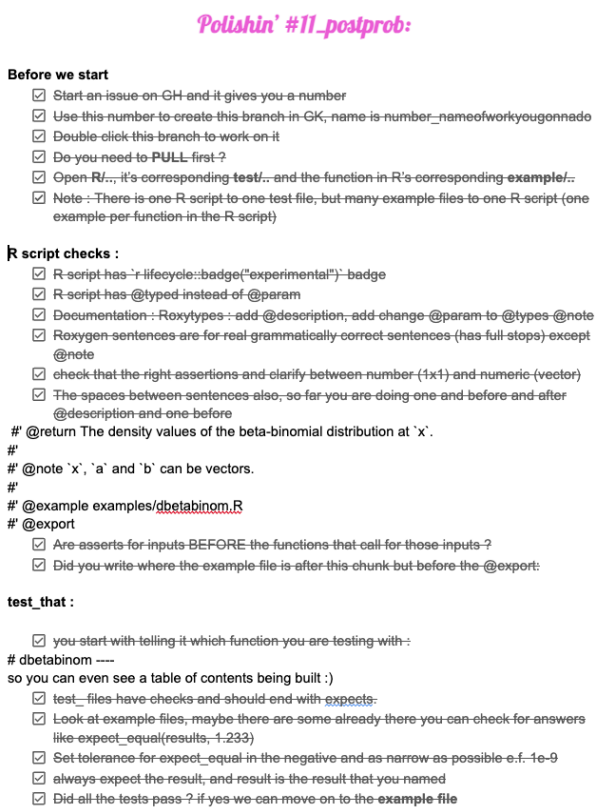
What is debugging ?

To find a needle in the haystack systematically

- Takes the most time and is the most difficult
- Needs a lot more courage than creativity
- Maintains humility

embrace small steps as smaller PRs can be sizeABLE.

Polishin' the vintage function - A checklist approach



Polishin' #11_postprob:

Before we start

- ☑ ~~Start an issue on GH and it gives you a number~~
- ☑ ~~Use this number to create this branch in GK, name is number_nameofworkyougonnado~~
- ☑ ~~Double click this branch to work on it~~
- ☑ ~~Do you need to **PULL** first ?~~
- ☑ ~~Open **R/**.., it's corresponding **test/**.. and the function in R's corresponding **example/**..~~
- ☑ ~~Note : There is one R script to one test file, but many example files to one R script (one example per function in the R script)~~

R script checks :

- ☑ ~~R script has `r lifecycle::badge("experimental")` badge~~
- ☑ ~~R script has @typed instead of @param~~
- ☑ ~~Documentation : Roxytypes : add @description, add change @param to @types @note~~
- ☑ ~~Roxygen sentences are for real grammatically correct sentences (has full stops) except @note~~
- ☑ ~~check that the right assertions and clarify between number (1x1) and numeric (vector)~~
- ☑ ~~The spaces between sentences also, so far you are doing one and before and after @description and one before~~

#' @return The density values of the beta-binomial distribution at `x`.
#'
#' @note `x`, `a` and `b` can be vectors.
#'
#' @example examples/dbetabinom.R
#' @export

- ☑ ~~Are asserts for inputs BEFORE the functions that call for those inputs ?~~
- ☑ ~~Did you write where the example file is after this chunk but before the @export:~~

test_that :

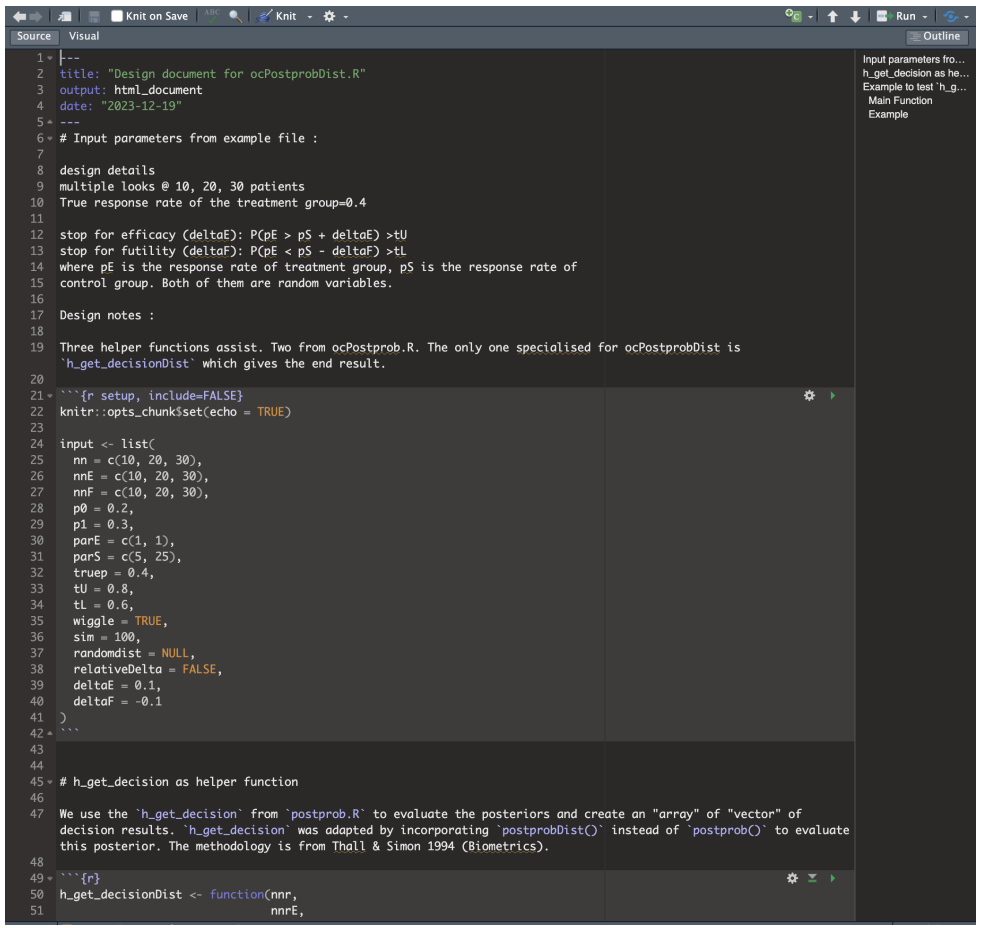
- ☑ ~~you start with telling it which function you are testing with :~~

dbetabinom ----
so you can even see a table of contents being built :)

- ☑ ~~test_ files have checks and should end with expects.~~
- ☑ ~~Look at example files, maybe there are some already there you can check for answers like expect_equal(results, 1.233)~~
- ☑ ~~Set tolerance for expect_equal in the negative and as narrow as possible e.f. 1e-9~~
- ☑ ~~always expect the result, and result is the result that you named~~
- ☑ ~~Did all the tests pass ? if yes we can move on to the **example file**~~

1. One issue to one branch
2. One issue to one Pull Request (PR)
3. One Design Document for entire function
4. One issue is the smallest task from (3)
5. Create Test files for helper functions
6. Create Test files and example files for main function calls

The Design Document



User-facing functions start with Design-document (DD)

1. DD helps achieve Clarity on the form and purpose of the user-facing function
2. Use the DD to test regular and edge cases
3. Makes the rest of the work “easier” when the goals are clear
4. Most of the “skeleton” and “flesh” of the work can already be done in the DD

First successes

- First Pull Request (PR) merged on Aug 29 2023 🥳
- submitted an abstract by November 2023 at **PSI**

A Bayesian approach to decision making in early development clinical trials : An R solution.

Early clinical trials play a critical role in Oncology drug development. The main purpose of early trials is to determine whether a novel treatment demonstrates sufficient safety and efficacy signals to warrant further investment (Lee & Liu, 2008). The new R package *phase1b* (Yeo et al, 2024) is a flexible toolkit that calculates many properties to this end, especially in the oncology therapeutic area. The primary focus of this package is on binary endpoints. The benefit of a Bayesian approach is the possibility to account for prior data (Thall & Simon, 1994) in that a new drug may have shown some signals of efficacy owing to its proposed mode of action, or similar activity based on prior data. The concept of the *phase1b* package is to evaluate the posterior probability that the response rate with a novel drug is better than with the current standard of care treatment in early phase trials such as Phase I. The *phase1b* package provides a facility for early development study teams to decide on further development of a drug either through designing for phase 2 or 3, or expanding current cohorts. The prior distribution can incorporate any previous data via mixtures of beta distributions. Furthermore, based on an assumed true response rate if the novel drug was administered in the wider population, the package calculates the frequentist probability that a current clinical trial would be stopped for efficacy or futility conditional on true values of the response, otherwise known as operating characteristics. The intended user is the early clinical trial statistician in the design and interim stage of their study and offers a flexible approach to setting priors and weighting.

References

Thall P F, Simon R (1994), Practical Guidelines for Phase IIB Clinical Trials, Biometrics, 50, 337-349

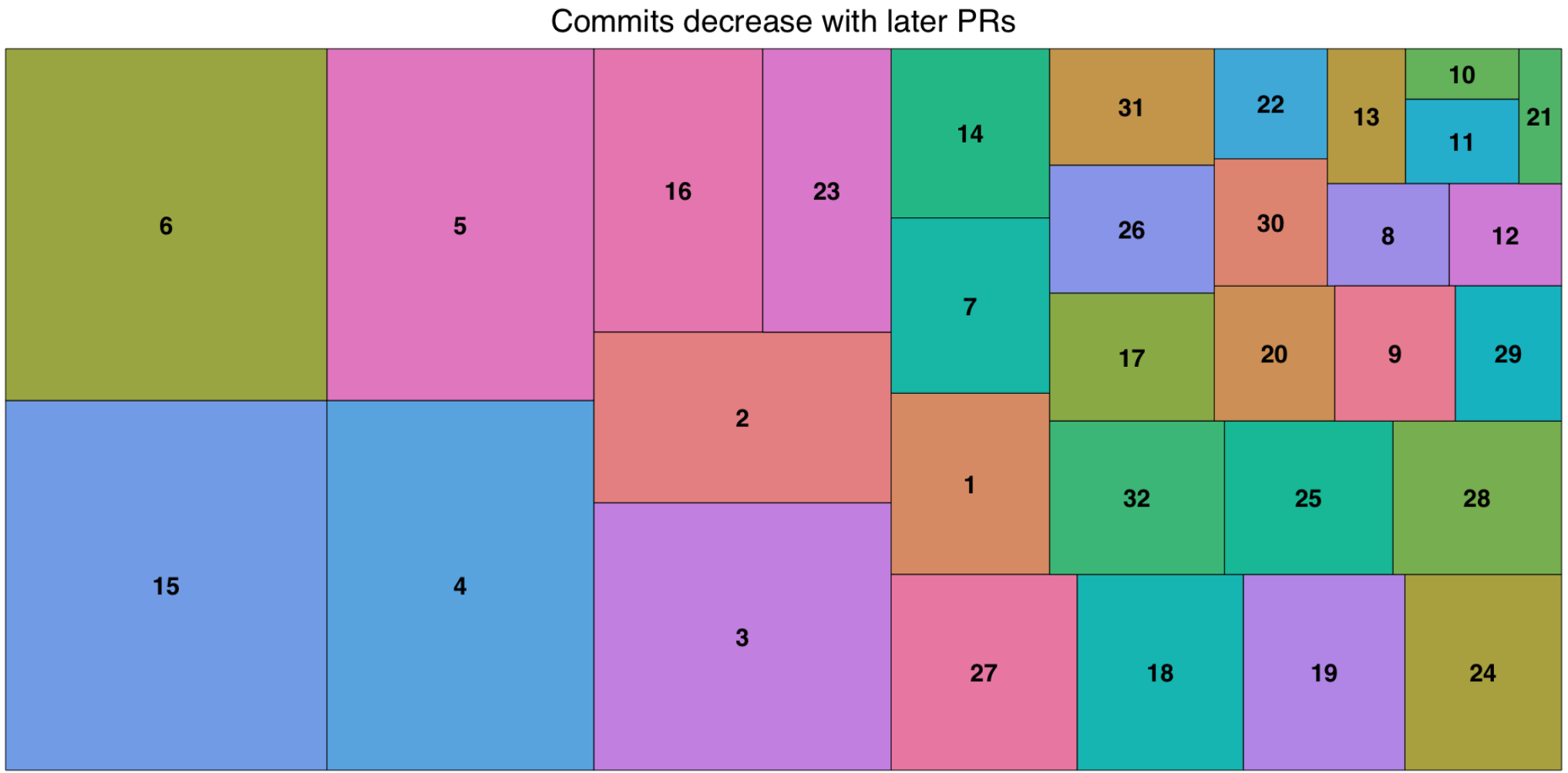
Lee J J, Liu D D (2008), A Predictive probability design for phase II cancer clinical trials, 5(2), 93-106, Clinical Trials

Yeo, A T, Sabanés Bové D, Elze M, Pourmohamad T, Zhu J, Lymp J, Teterina A (2024). Phase1b : Calculations for decisions on Phase 1b clinical trials. R package version 1.0.0, < https://genentech.github.io/phase1b>
(To be published prior to conference)

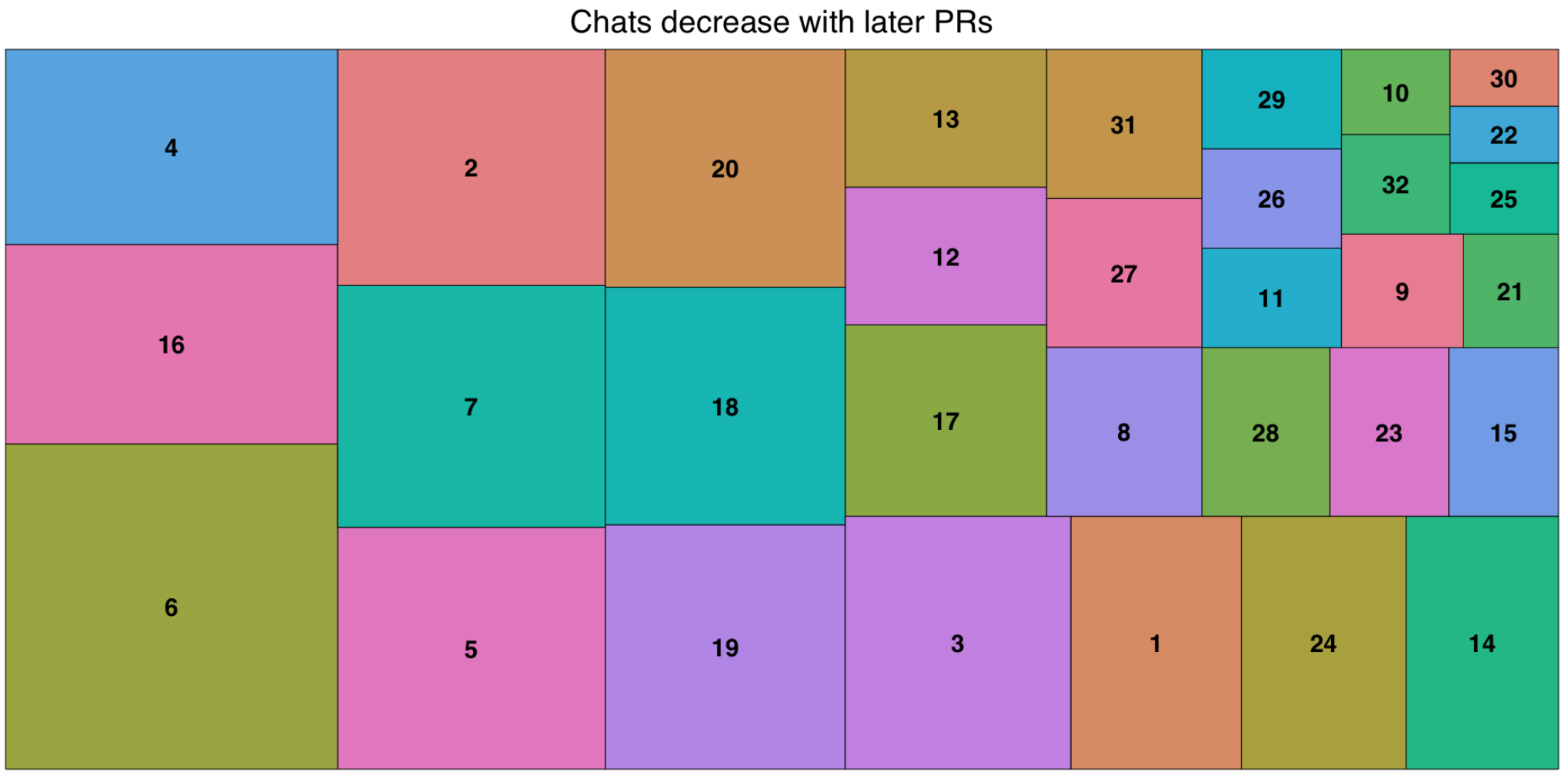
The `phase1b` journey in Pull Requests

Pull Request	Order of work
test for dbetabinom	1
pbetaMix and qbetaMix	2
postprob	3
ocPostprob	4
betadiff	5
postprobDist	6
getBetaMix and dbetaMix	7
predprob	8
Design doc for predprobDist	9
h_predprobdist_single_arm	10
h_predprobdist	11
predprobDist	12
h_get_decisionDist	13
h_getbetaMixpost	14
Design-doc_ocPredprob	15
h_get_decision_one_Predprob	16
h_get_decision_two_Predprob	17
h_get_oc	18
ocPredprob	19
Design-doc_ocPredprobDist	20

The `phase1b` journey in Commits



The `phase1b` journey in Chats



Features of Statistical Software Engineering

- Moving parts, collaboration is must
- Alignment of Forest vs Trees perspective
- Requires focused and deep thinking
- Small steps are good starts e.g. small PRs
- Creative work comes after foundation
- Systematic deduction is key to debugging
- Lots of theory and reading literature about new methodology
- Good engineering practices are habits, needs automation which needs practice which needs time

Good practices for new starter

Goal : introduce good practices and improve muscle memory and collaborate

- Mindset : Being dependent on others is normal. Asking questions are key because there are also many moving parts. Group chats are valuable in a safe space to ask questions.
- Regular stand ups
- Taking Small steps : are good starts
- Taking Systematic steps : no matter how slow
  - e.g. `debug()`, `undebug()`, `options(recover = error)`, `options(recover = NULL)`
- older PRs are helpful to learn and reflect

Good practices for the Engineering Lead

Goal : introduce good practices and improve muscle memory and collaborate

- Patience
- Quick turnover for feedback
- Positive feedback but no over positiveness, always based on facts
- Use chats in PR can be helpful to precise the feedback
- Demonstrate small and systematic deduction during debugging
- Acknowledge learning styles can be different
- Allow developers to do the work, make mistakes and keep sharing rationale

Conditions for success from the `phase1b` work

“Consistency is key to building, and trust is easily broken in the absence of consistency. Open and constant communication is essential.”

- Positive bias & Trust are win-win situations
- Learning through mistakes is key
- Safe space to be ask any questions, and iterate, even for a seasoned engineer

Kinds of feedback

- “Great motivation to learn new skills”
- “… it was always fun and easy to work with Audrey”

How I want to be a better Software Engineer

- Develop the skills of slow and systematic deduction (efficiency gains)
- Create more conditions for focused, systematic, deep and creative work
- Perceive small steps as good starts
- Continue having an open mind on what good practices are
- Be more patient with the process
- Fall on the engineering principles (testing, debugging, iteration, reliability, correctness…)

Sharing the Success story: Touring `phase1b`

- PSI June 2024 conference in Netherlands link
- useR!2024 Salzburg link
- PHUSE September 2024 in Basel link
- University of Basel November 2024 link

Audrey Yeo

Future outlook

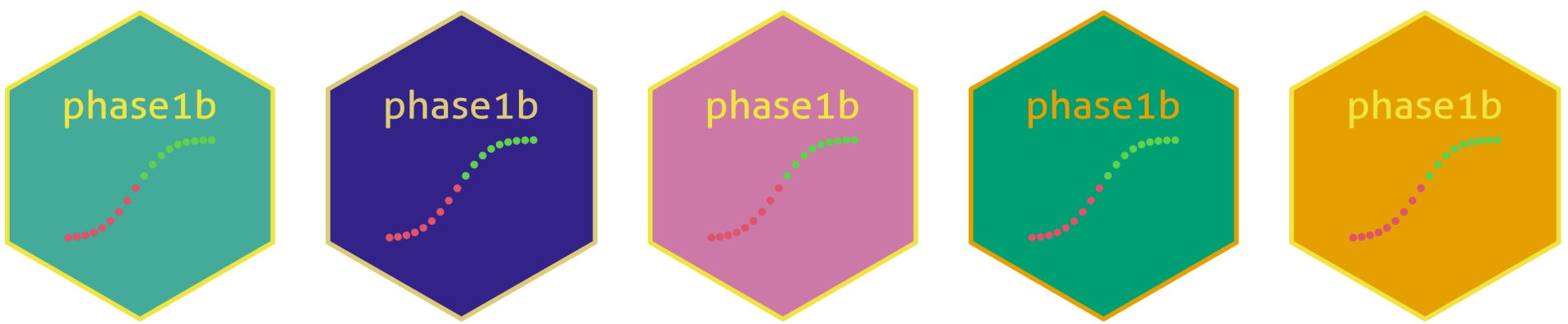
- Go through several more issues to complete and make it more delightful
- submission to CRAN
- Collaborate by contacting me

More software work for statisticians with values of : Inclusion, Diversity, Impact and Delightful user experience

Final Reflections

- Doing something I wasn't prepared to do paid off
- Confidence through Competence : I got more confident about the R Language
- Can use this as a reference to build other packages and in other languages e.g. Julia, see my blog post
- Less likely to doubt myself when I learn new things

No matter how much you've improved it's not permanent



Thank you

- Daniel Sabanés Bové
- …and other colleagues at Data Science Acceleration at Roche
- Open Source community and R community that do great work and share their knowledge

I’d love to know how this presentation relates to you or does not !

Some references

- Thall P F, Simon R (1994), Practical Guidelines for Phase IIB Clinical Trials, Biometrics, 50, 337-349
- Lee J J, Liu D D (2008), A Predictive probability design for phase II cancer clinical trials, 5(2), 93-106, Clinical Trials
- Yeo, A T, Sabanés Bové D, Elze M, Pourmohamad T, Zhu J, Lymp J, Teterina A (2024). Phase1b : Calculations for decisions on Phase 1b clinical trials. R package version 1.0.0, https://genentech.github.io/phase1b
- Code for this presentation

Some more references

- Code with engineering. Microsoft
- How to do a code review. Google
- Pachecho, C, A Technical Journey into API Design-First: Best Practices and Lessons Learned link
- Why We need to Improve Software Engineering in Biostatistics (October 26 2023 R/Pharma) link
- Inclusive Speaker Course by Linux Foundation link
- license

Audrey Yeo