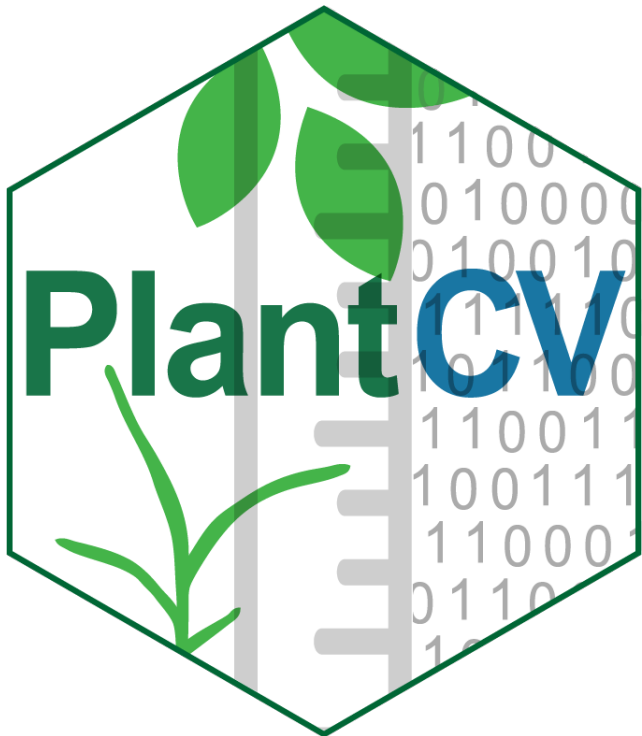


# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- Resources

# PlantCV

- PlantCV is an open-source python-based image analysis software developed initially by Malia Gehan and Noah Fahlgren and maintained by their labs and others.



## PlantCV: Plant Computer Vision

PlantCV is an open-source image analysis software package, specifically for plant science. Use PlantCV to measure plant traits (aka phenotypes) from images. The project is made possible by the effort of many generous contributors, collaborators, and users, and is managed by Malia Gehan and Noah Fahlgren.

## Usage Statistics

Publications **139** Latest Version **v4.5** GitHub contributors **57**

conda downloads **28k** PyPI downloads **180/day**

Stars **648** Follow @plantcv

# PlantCV

- PlantCV extracts numeric phenotypes from images, with a variety of shape/size/color data available.
- Sometimes the data-science core would get feedback requesting support for statistical analysis of those phenotypes.
  - To address that request we started working on pcvr

# pcvr

- The main goal of pcvr is to lower the barrier to entry for several kinds of longitudinal modeling options and selected Bayesian statistics.
- Secondary goals are to support common analysis based on what is often done at the Danforth Plant Science Center based on the data-science core's experience.

# Status of pcvr

- In September of 2024 v1.0.0 of pcvr was posted to CRAN
- Development versions are available from the danforth center's github page:
  - [danforthcenter/pcvr](https://github.com/danforthcenter/pcvr)

# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- Resources

# Bayesian Probability

- Data is observed and therefore known, the parameters in our models are random (probability distributions).

# Bayesian Probability

- Data is observed and therefore known, the parameters in our models are random (probability distributions).
- Things that can happen in more ways are more likely to happen, so we count the ways
  - Given **what happened** what is the **most plausible explanation**?
  - $P(\text{model} \mid \text{data}) \sim \text{Posterior Probability}$
  - High probability makes us trust the proposed **model**



# Probability Distributions

- A probability distribution is a mathematical function giving probabilities of different outcomes for an experiment.
  - It must sum to 1 (100%)
  - There are very many probability distributions such as the Normal, Student T, Chi-Square, Beta, etc.
  - Some have special properties that make them very appealing for statistics as a whole or Bayesian statistics in particular.

# Frequentist Probability

- Frequentist probability uses the frequency of events in very large samples/complete populations.

# Frequentist Probability

- Frequentist probability uses the frequency of events in very large samples/complete populations.
- Parameters in models have set values, data has a “sampling distribution” driven by randomness.
  - Given **what happened** what is the **most plausible explanation**?
  - $P(\text{data} \mid \text{model}) \sim P \text{ value}$
  - **Model** is the Null  $H_0$ , so a low p-value rejects it.

	Frequentist	Bayesian
Fixed	True Effect	Observed Data
Random	Observed Data	True Effect
Interpretation	If True Effect is 0 then there is an $\alpha$ * 100% chance of estimating an effect of X or more.	Given the effect size in our data there is a P probability of the True effect being at least X.

Where X is an effect size magnitude

	Frequentist	Bayesian
Fixed	True Effect	Observed Data
Random	Observed Data	True Effect
Interpretation	$P[\text{Data} \mid \text{No True Population Effect}]$	$P[\text{hypothesis} \mid \text{Prior} + \text{Observed Data}]$

Where X is an effect size magnitude

	Frequentist	Bayesian
Fixed	True Effect	Observed Data
Random	Observed Data	True Effect
Interpretation	$P[\text{Data} \mid \text{No True Population Effect}]$	$P[\text{hypothesis} \mid \text{Prior} + \text{Observed Data}]$

I like Bayesian probability and pcvr has some focus on Bayesian statistics.


Where X is an effect size magnitude

# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- Resources

We talked about probability and I'm sorry.  
We can take a break if you want.

# Conjugate

**conjugate** 

kɒn'jə-gāt"

**intransitive verb**

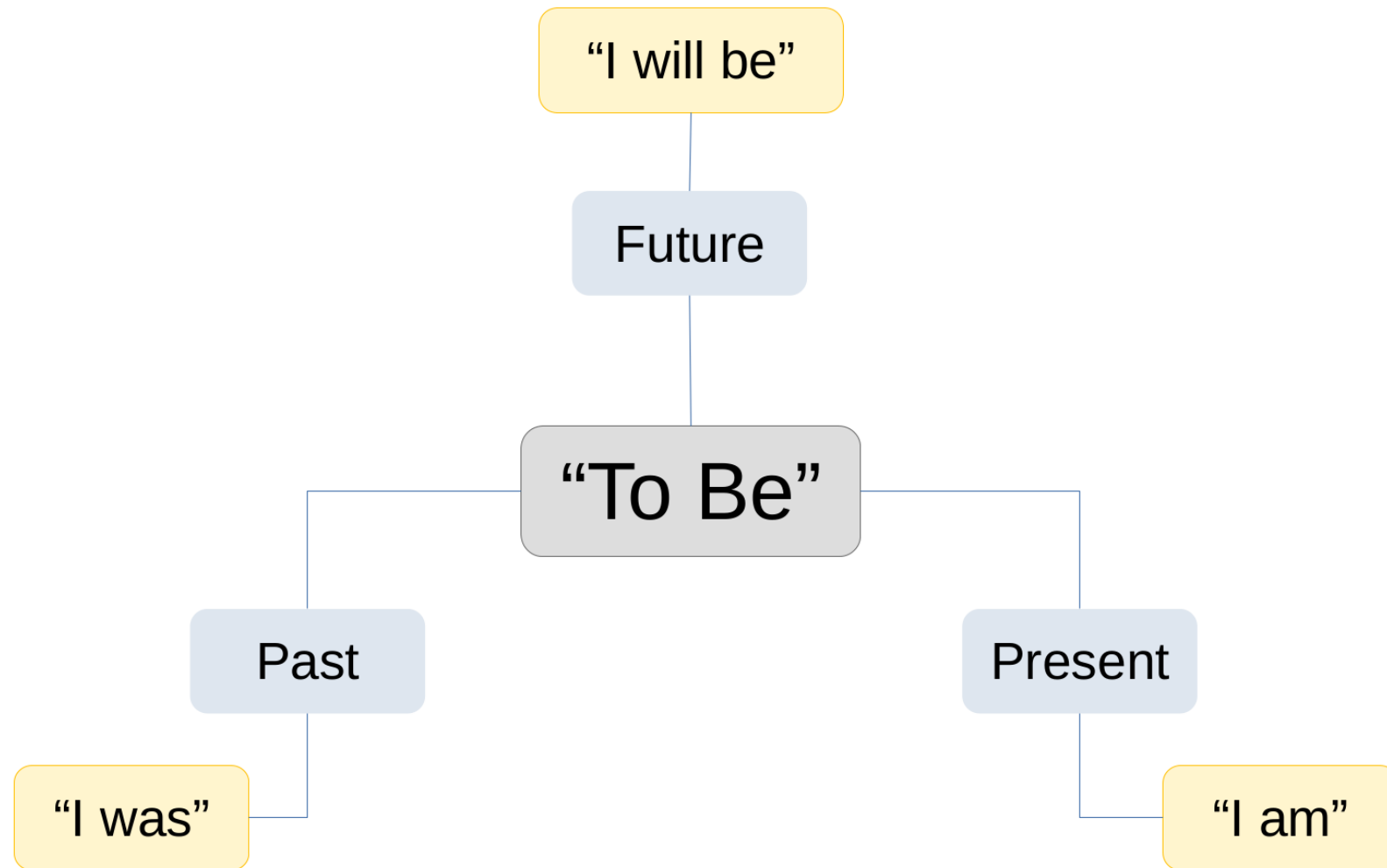
1. To inflect (a verb) in its forms for distinctions such as number, person, voice, mood, and tense.
2. To join together.
3. To undergo conjugation.



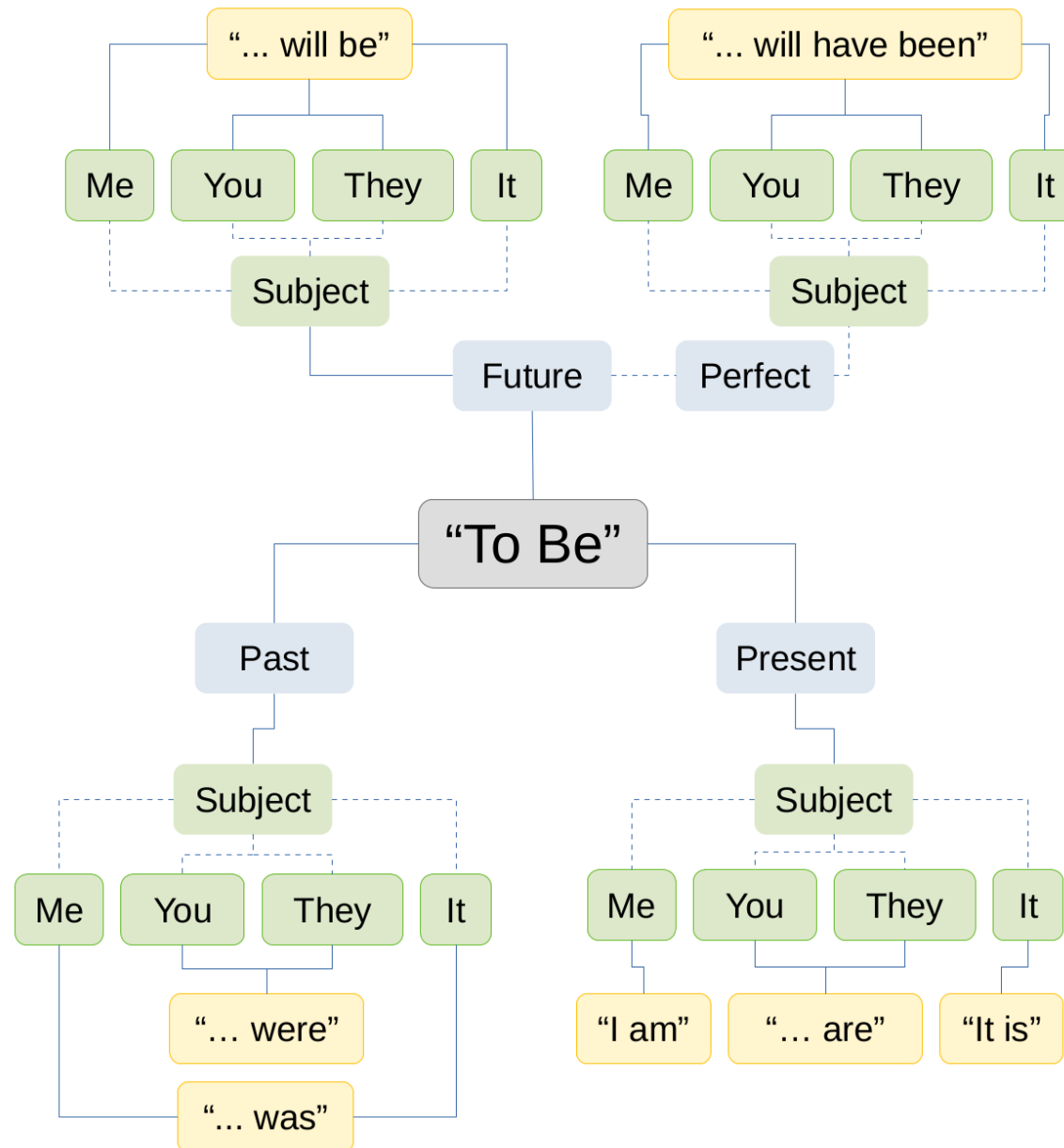
# A Distribution is a Fundamental Verb

“To Be”

# Tense is Data Added to the Verb



# More data makes it more specific

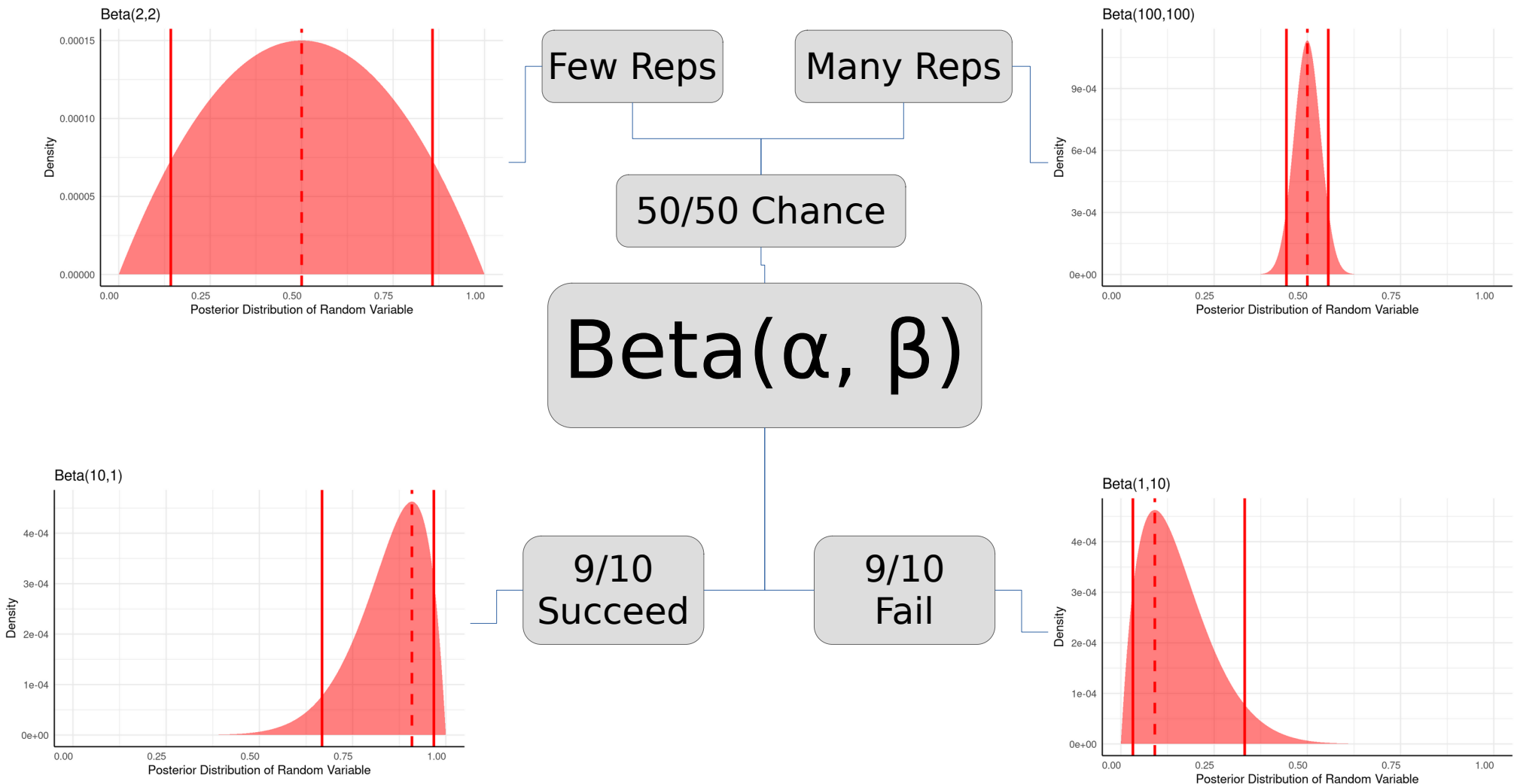


There are probability distributions  
that work the same way

Beta( $\alpha$ ,  $\beta$ )

$$\frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

# For Conjugate distributions we can add data to them very easily



# Getting a little more formal

$$\text{Beta}(\alpha, \beta) + (\alpha, \beta) \sim \text{Beta}(\alpha', \beta')$$

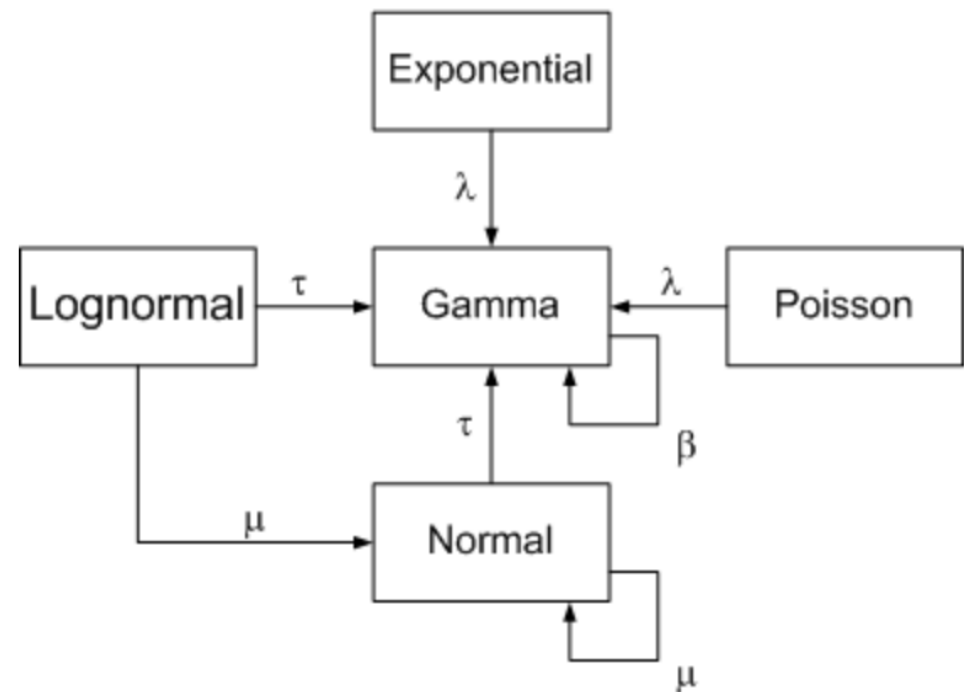
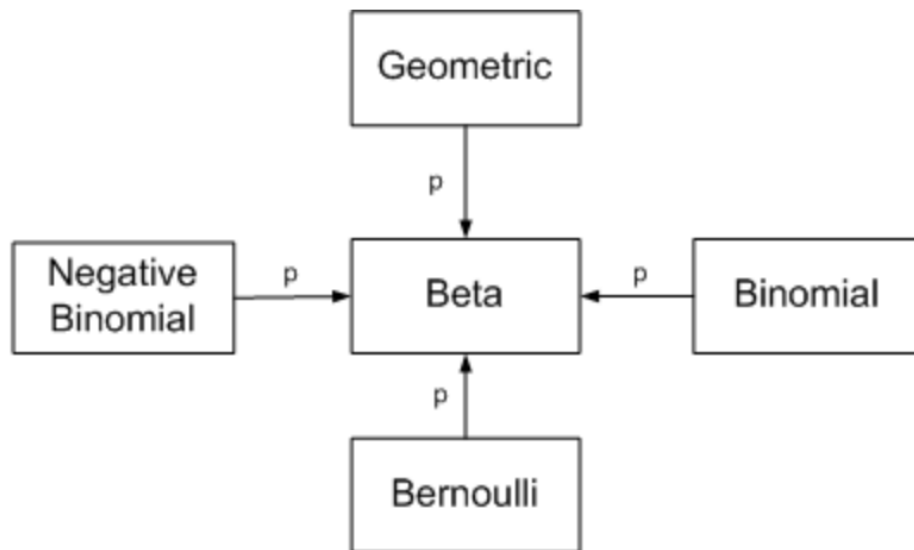
# Getting a little more formal

$$\text{Beta}(\alpha, \beta) + (\alpha, \beta) \sim \text{Beta}(\alpha', \beta')$$

$$\text{Prior} + \text{Data} \sim \text{Posterior}$$

We know that the prior and posterior  
Will be the same kind of distribution(verb)

# There are lots of conjugate distributions.





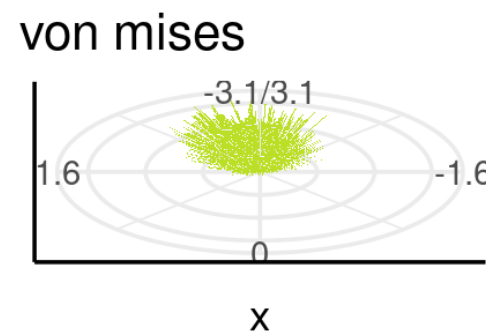
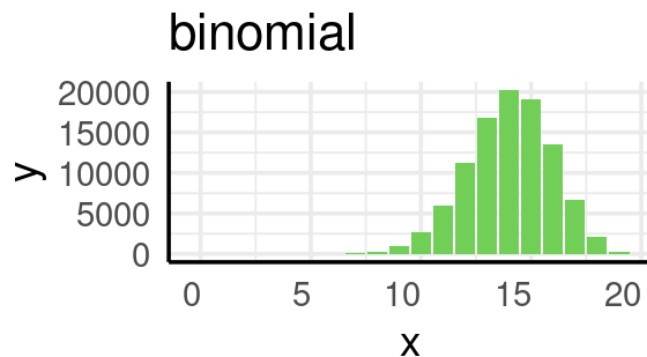
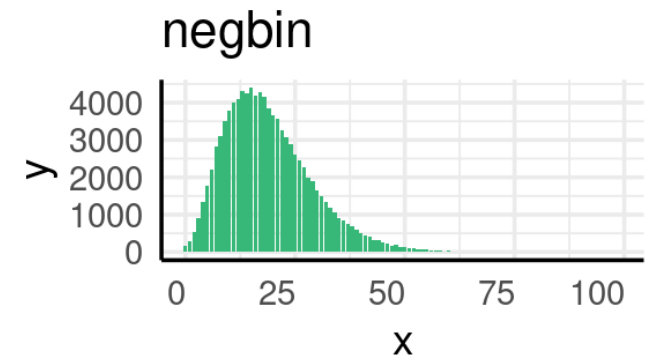
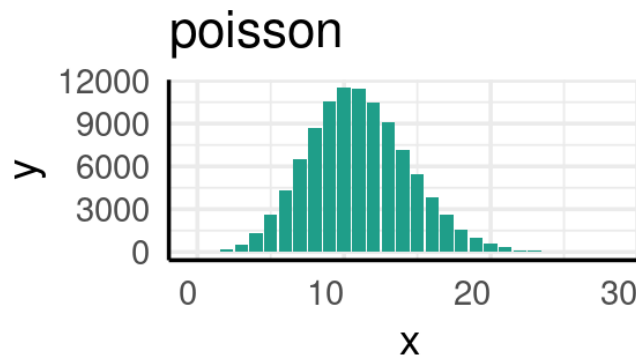
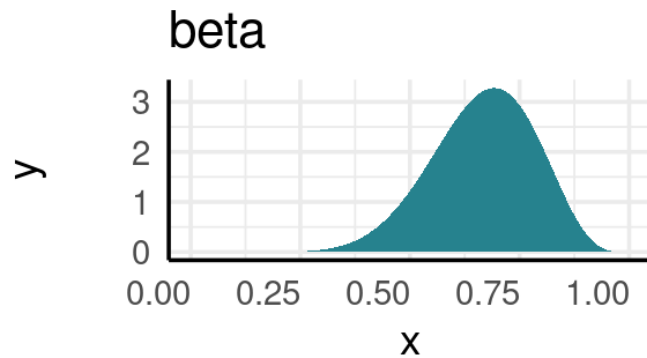
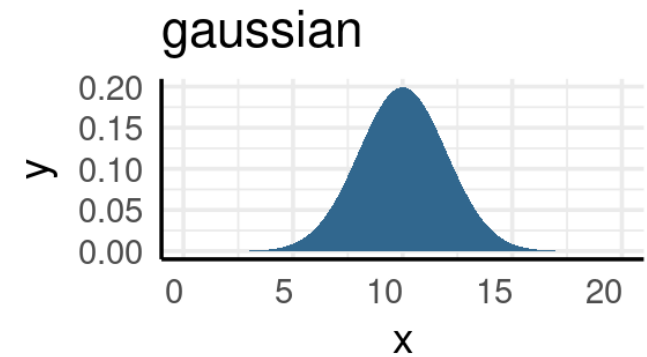
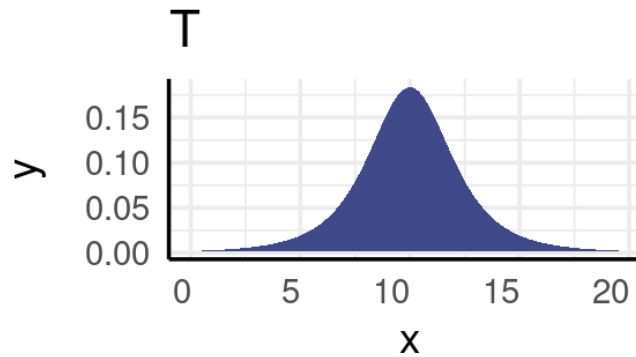
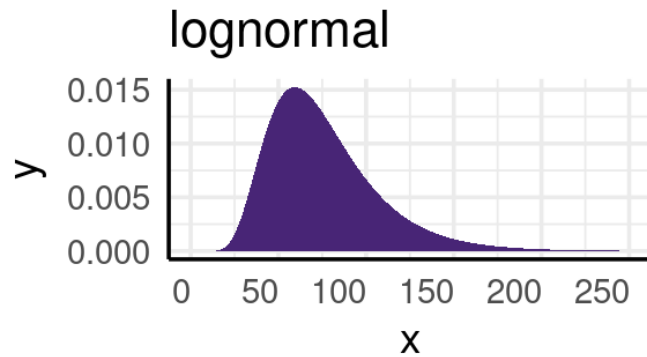
# pcvr::conjugate

- In pcvr the `conjugate` function can be used to make these simple distribution comparisons for a variety of data.

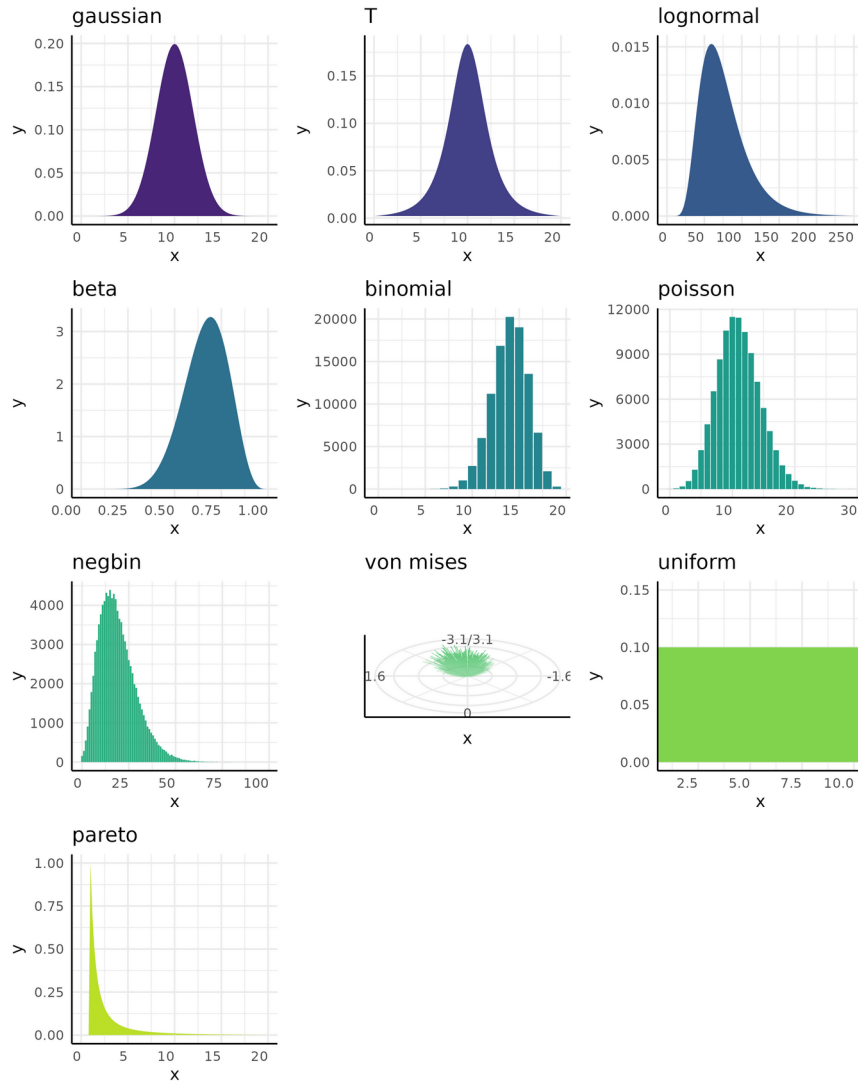
# pcvr::conjugate

- In pcvr the `conjugate` function can be used to make these simple distribution comparisons for a variety of data.
- Currently there are 8 supported distributions.

# pcvr::conjugate distributions



# pcvr::conjugate UPDATED



# pcvr::conjugate distributions

Distribution	Data	Flow Chart
<b>T</b>	Gaussian Means	T test
<b>Gaussian</b>	Gaussian Distributions	Z test*
<b>Lognormal</b>	Right Skewed Continuous	Wilcox
<b>Von Mises</b>	Symmetric Circular data	Watson*
<b>Poisson</b>	Counts	Wilcox
<b>Negative Binomial</b>	Counts	Wilcox
<b>Beta</b>	Percentages	Wilcox
<b>(Beta) Binomial</b>	Success/Failure counts	Wilcox

# pcvr::conjugate ROPE

- ROPE (Region of Practical Equivalence) testing is also implemented in `pcvr::conjugate`, where `rope_range` and `rope_ci` can be specified.

# ROPE tests help decision-making

- We often talk about how statistical significance is not biological significance.
- ROPE helps bridge that gap.

# An Example of conjugate+ROPE

- Say we want to know if two percentages are different, but if the difference is less than 7% then it is not very interesting biologically.



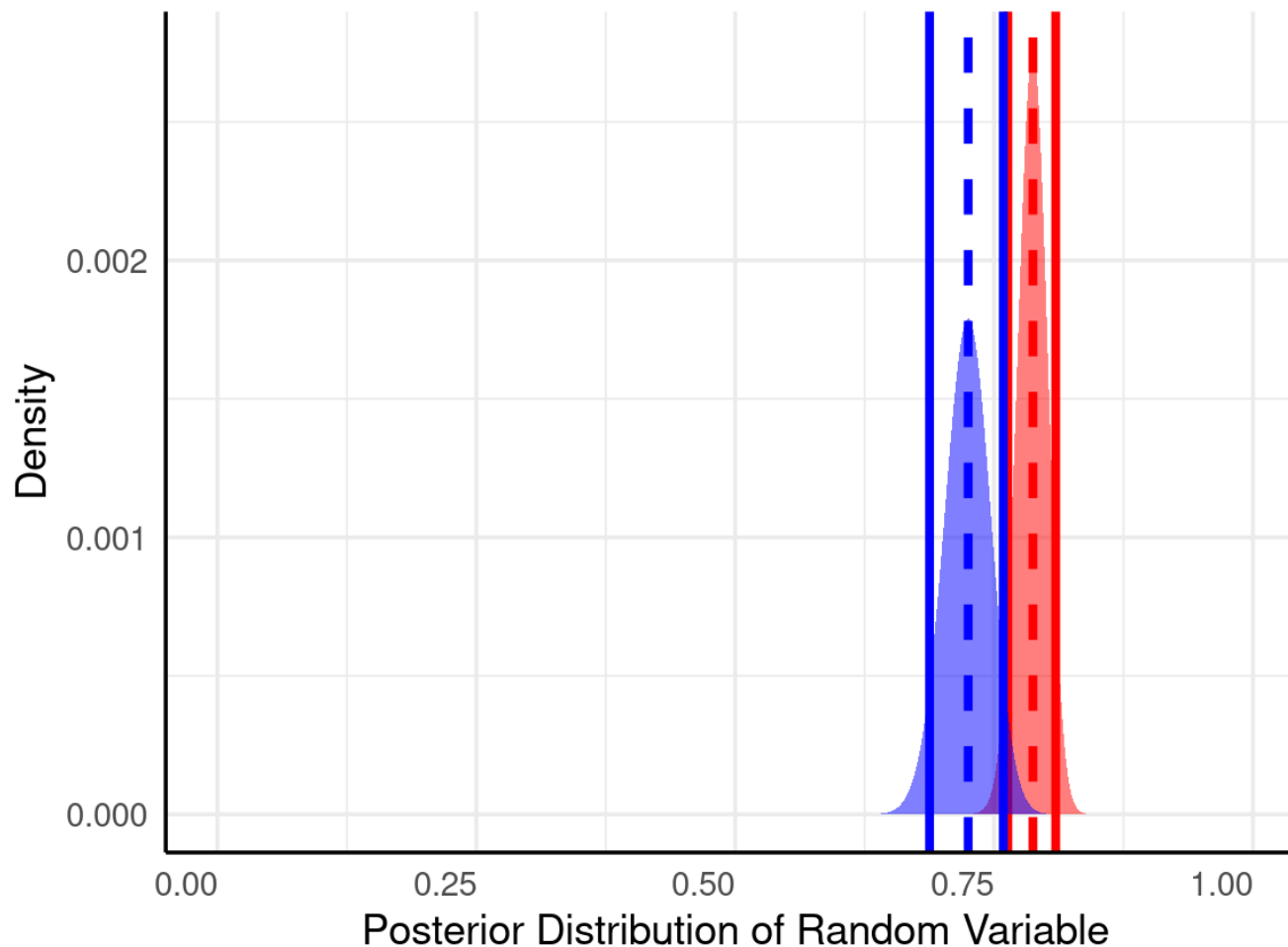
# An Example of conjugate+ROPE

## Distribution of Samples

Sample 1: 0.79 [0.76, 0.81]

Sample 2: 0.72 [0.69, 0.76]

$P[p_1=p_2] = 0.08408$

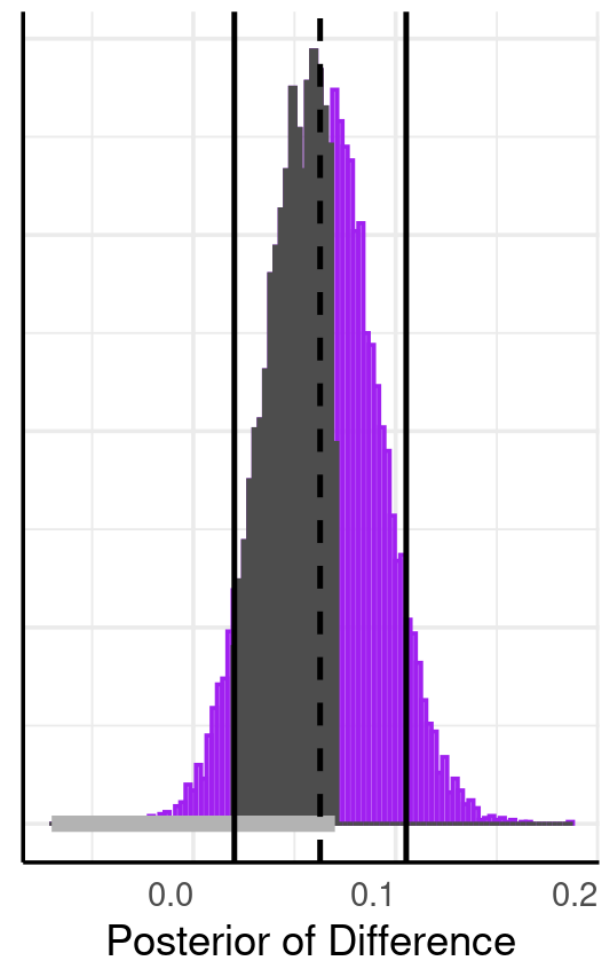


## Distribution of Difference

Median Difference of 0.06

89% CI [0.02, 0.11]

0.89% HDI in [-0.07, 0.07]: 0.62



# An Example of conjugate+ROPE

## Distribution of Samples

Sample 1: 0.79 [0.76, 0.81]

Sample 2: 0.72 [0.69, 0.76]

$P[p_1=p_2] = 0.08408$

## Distribution of Difference

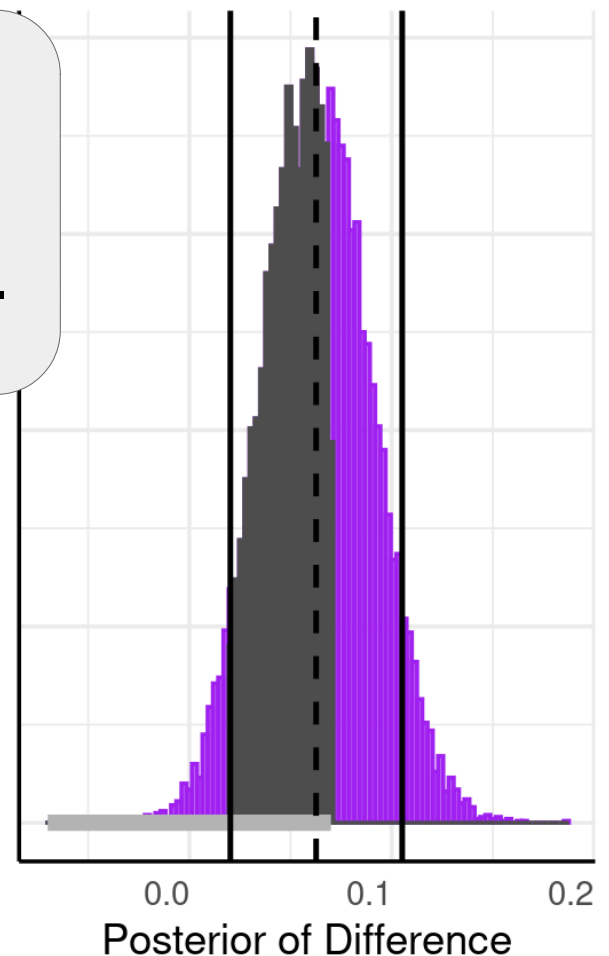
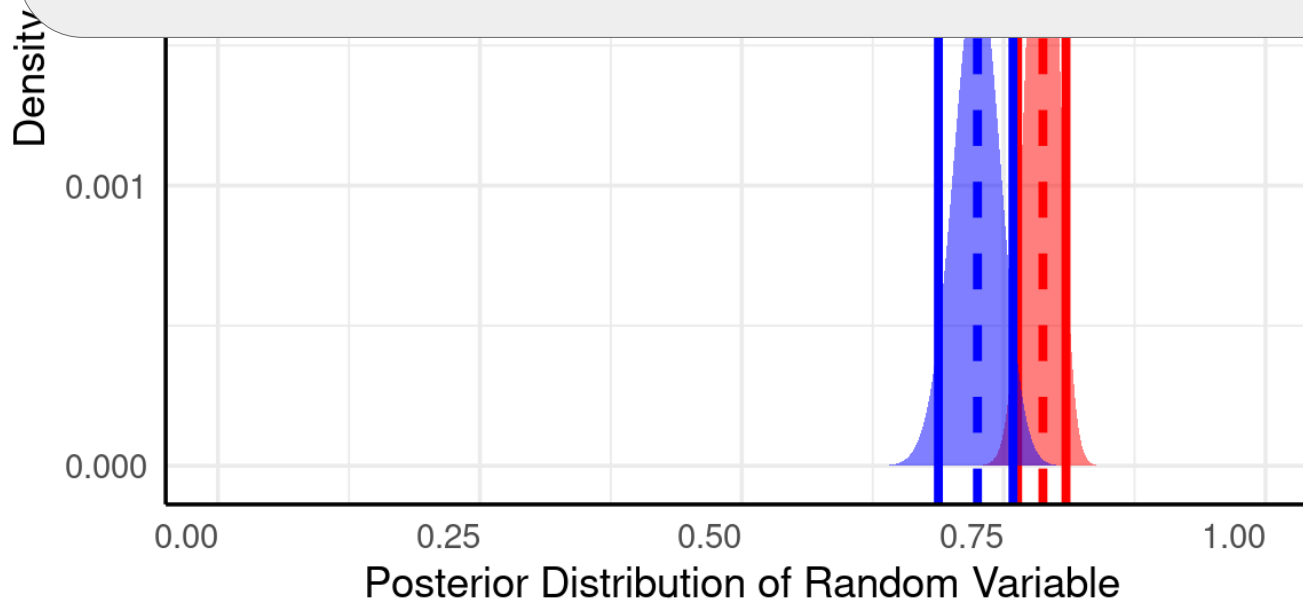
Median Difference of 0.06

89% CI [0.02, 0.11]

0.89% HDI in [-0.07, 0.07]: 0.62

## Interpretation:

There is about an 8.5% chance that the Rate is the same between these groups.



# An Example of conjugate+ROPE

## Distribution of Samples

Sample 1: 0.79 [0.76, 0.81]

Sample 2: 0.72 [0.69, 0.76]

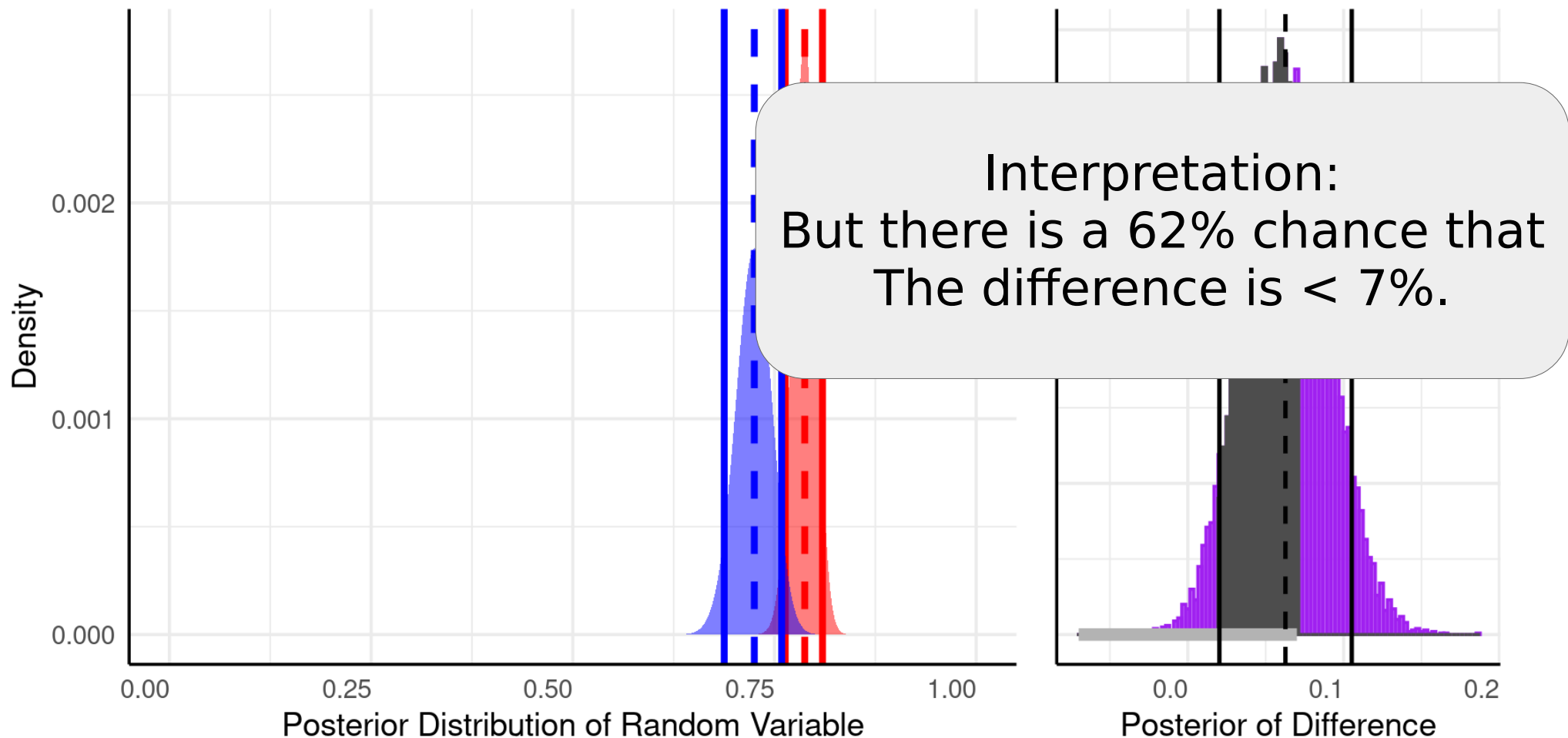
$P[p_1=p_2] = 0.08408$

## Distribution of Difference

Median Difference of 0.06

89% CI [0.02, 0.11]

0.89% HDI in [-0.07, 0.07]: 0.62



# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- Resources

That was a quick introduction to Bayes and Conjugacy. Take a break then we'll talk a little about non-linear modeling.

# Non-Linear Modeling

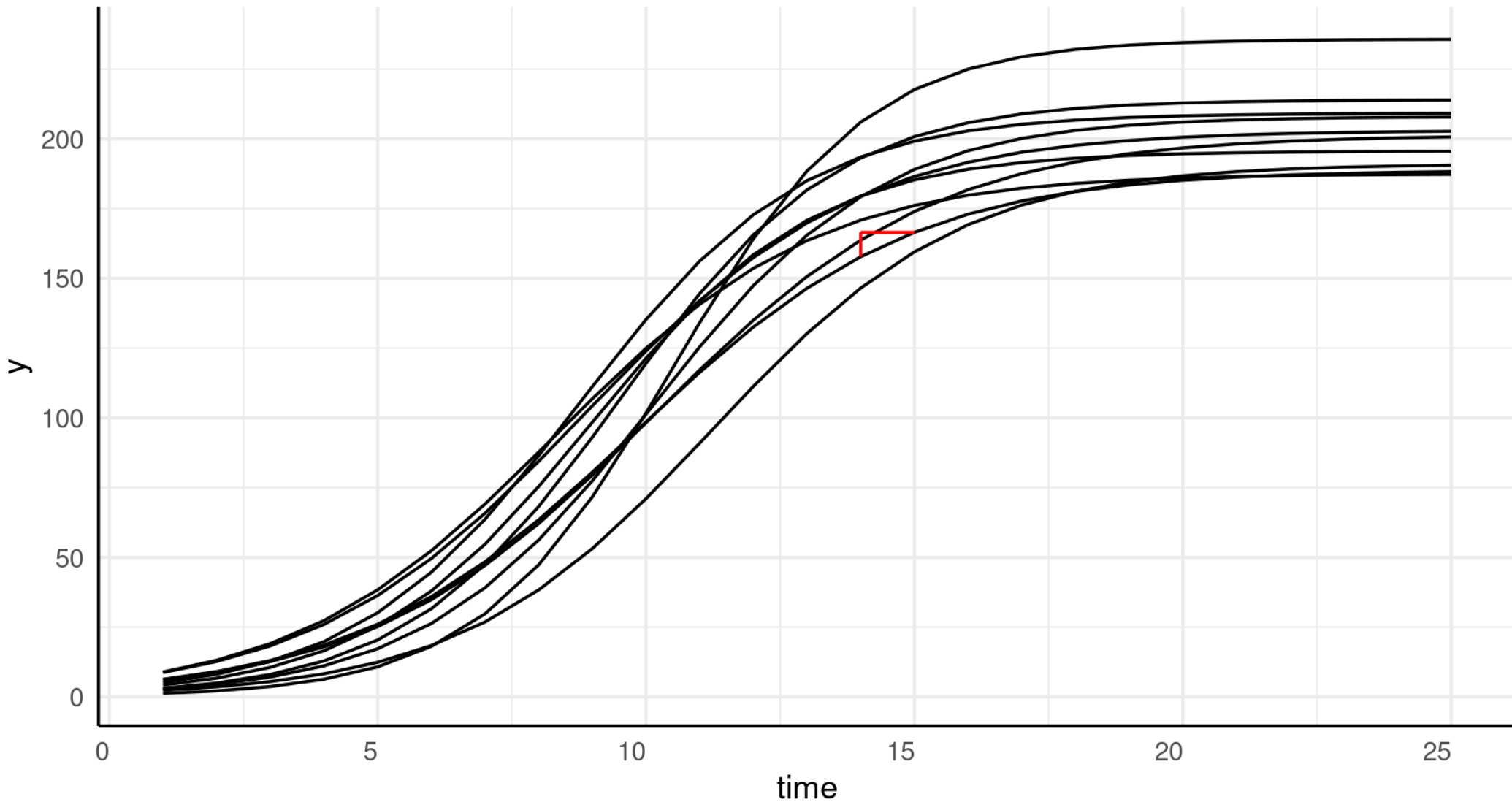
- pcvr has several functions that try to simplify common non-linear modeling needs.

# Non-Linear Modeling

- pcvr has several functions that try to simplify common non-linear modeling needs.
- One of the main settings where non-linear modeling comes up in plant science is for longitudinal modeling.
  - Other options include dose-response curves and time-to-event analysis.

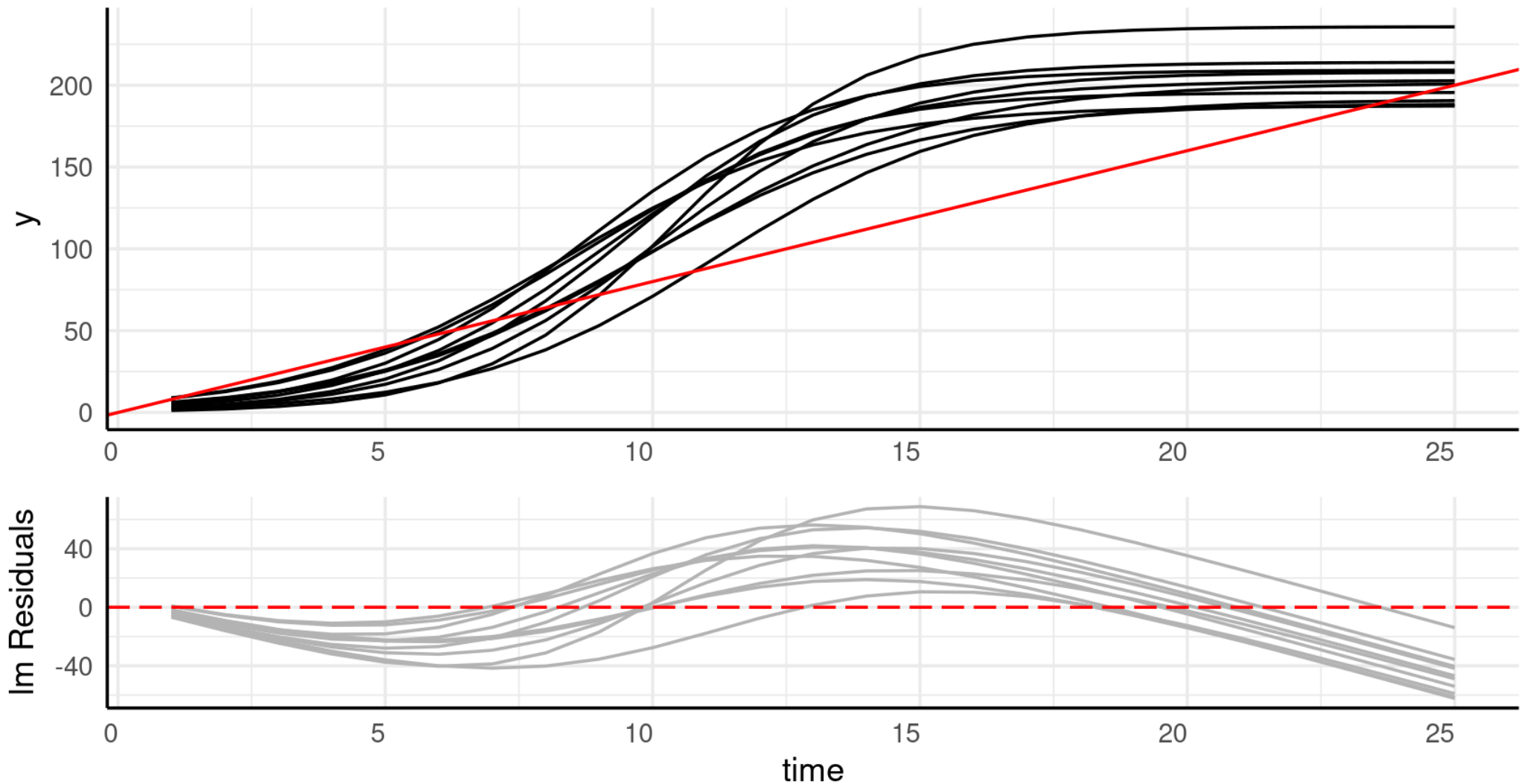
# Longitudinal Modeling

Autocorrelation



# Longitudinal Modeling

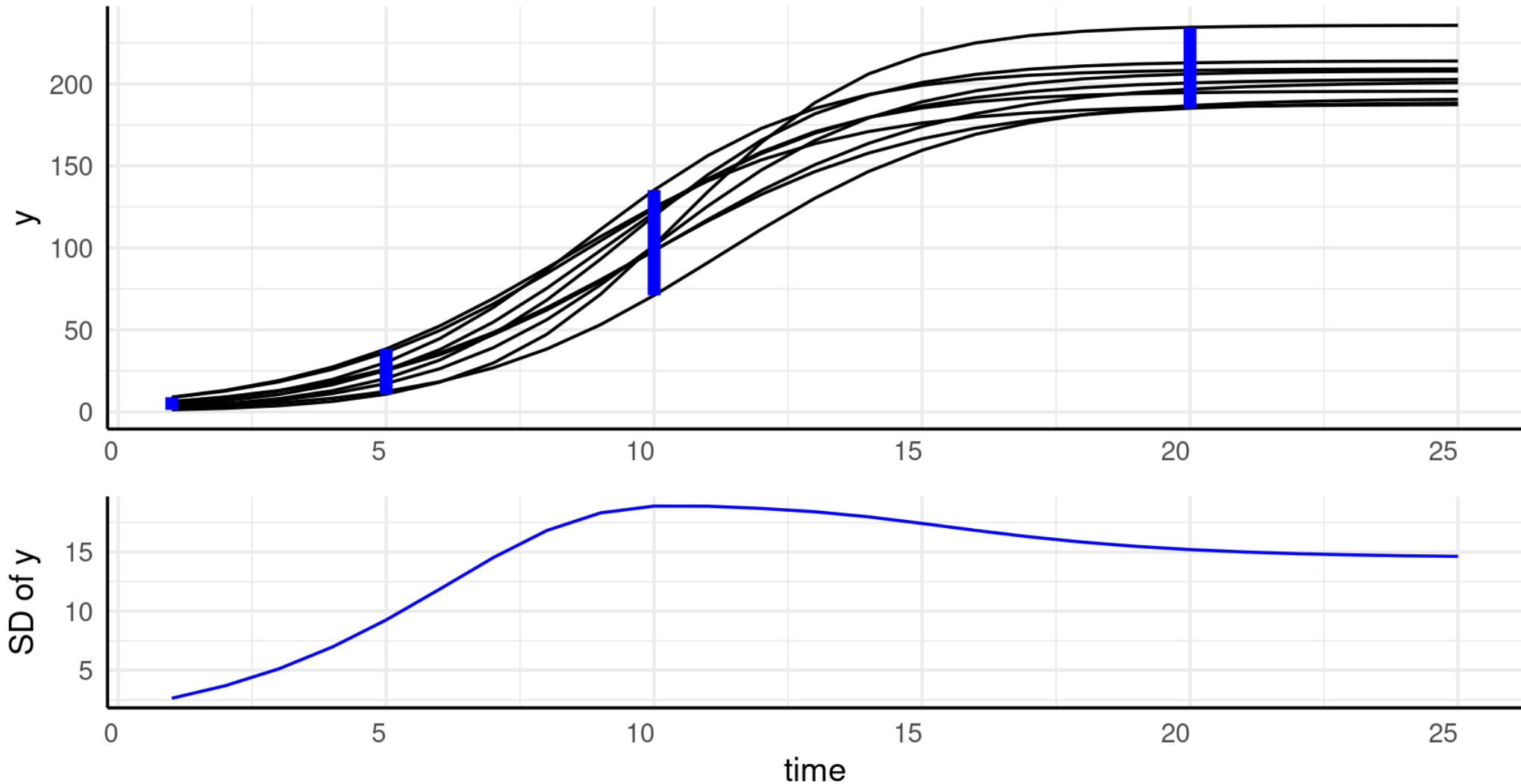
Non-Linearity





# Longitudinal Modeling

Heteroskedasticity

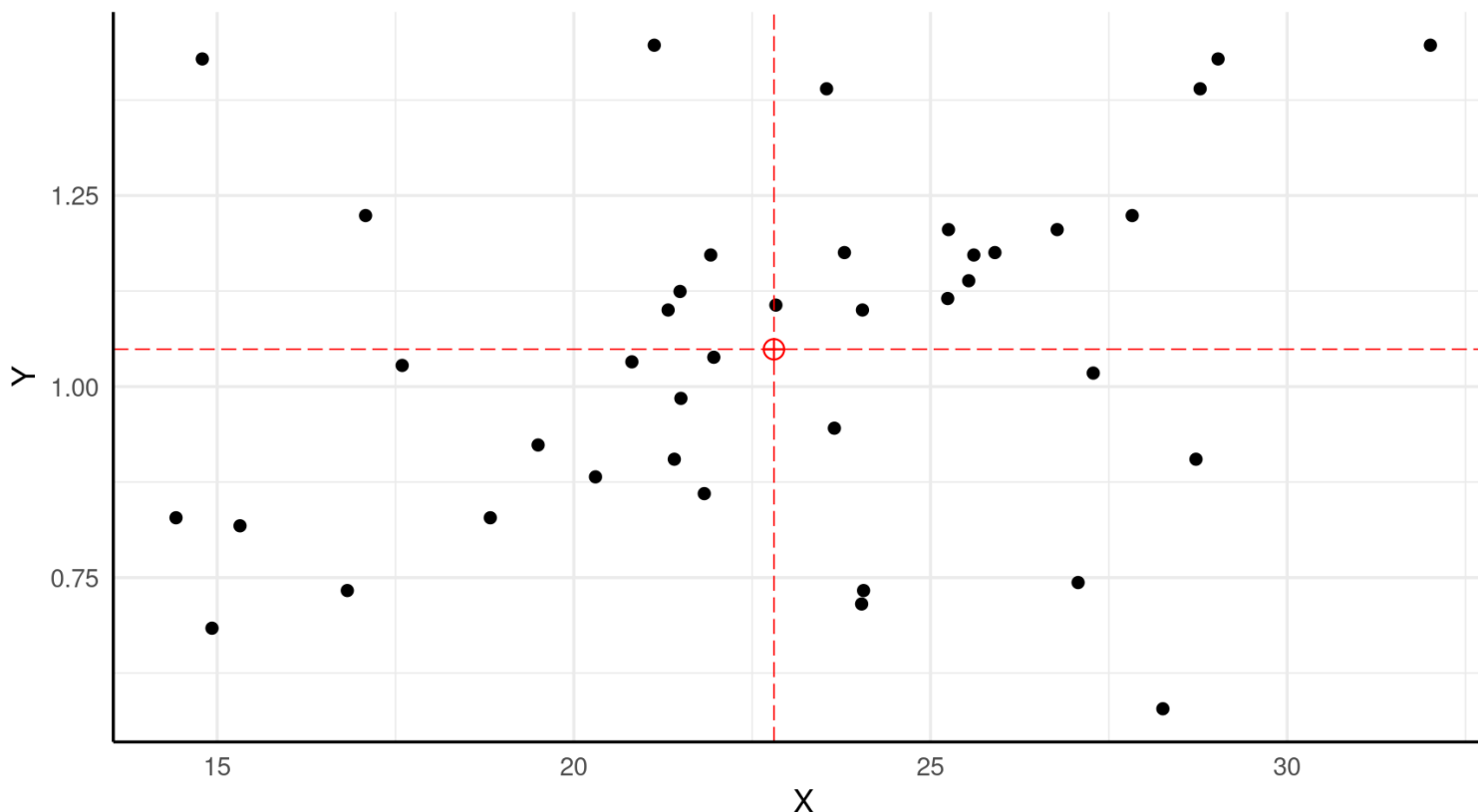


# Non-linear Modeling can be complex

- In linear models we can easily work directly from the data.

We label the midpoint of X and Y

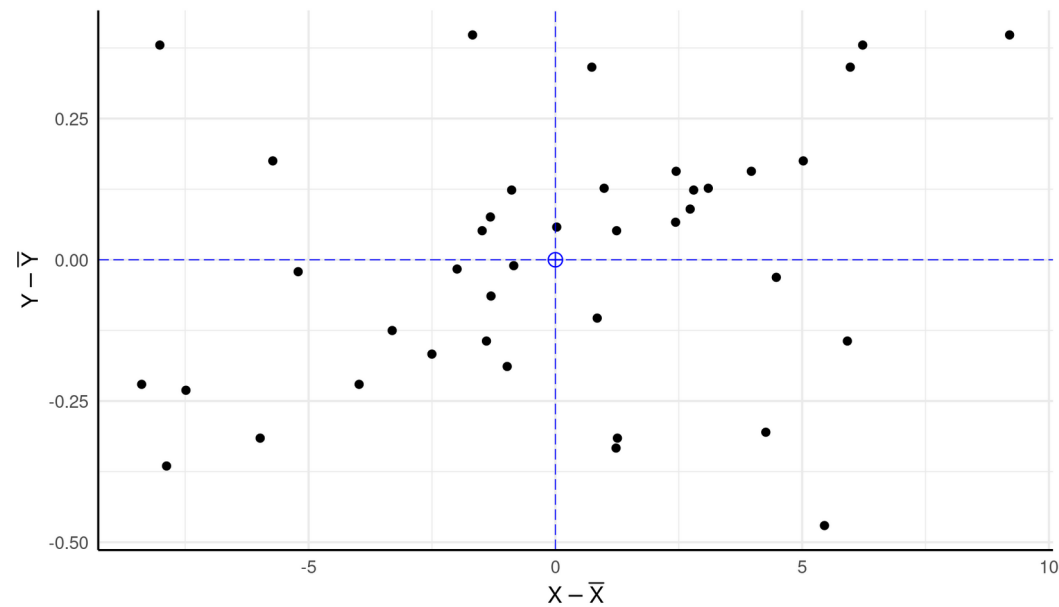
We know our mean trend line will pass through here.



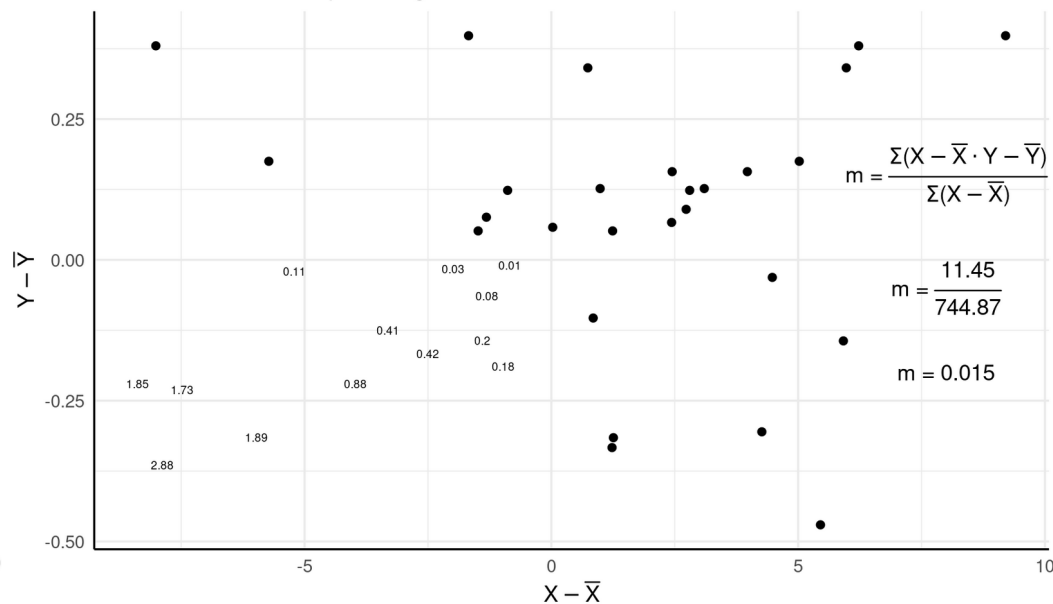
# Non-linear Modeling can be complex

- In linear models we can easily work directly from the data.

We subtract the mean of X and Y out of the data

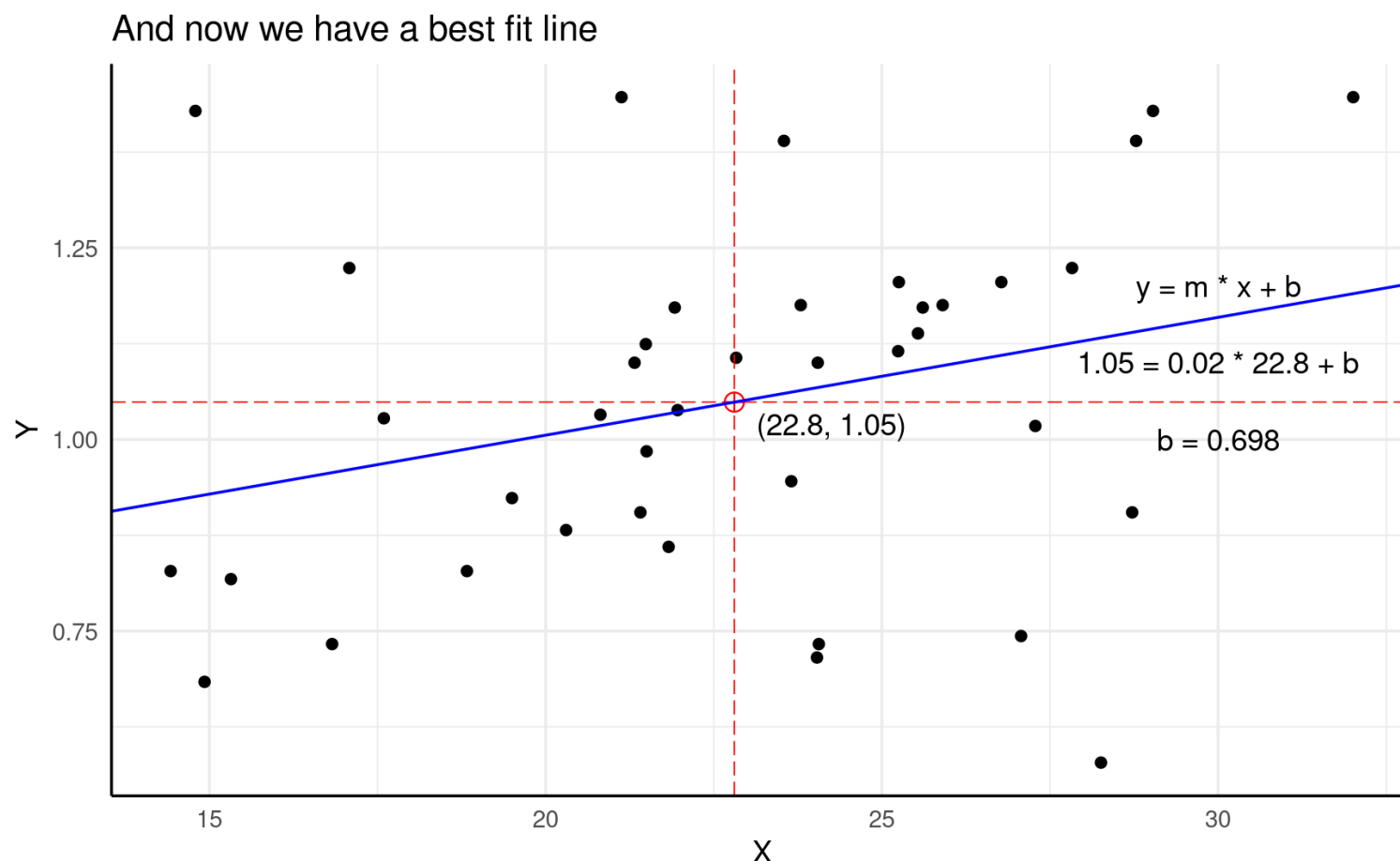


We calculate the slope using these coordinates



# Non-linear Modeling can be complex

- In linear models we can easily work directly from the data.



# Non-linear Modeling can be complex

- In Non-linear models we need
  - A more explicit formula
  - starting values

# Non-linear formulas

- We have to explicitly name the parameters we want to estimate.
- This allows for great flexibility, but it is more involved.

```
> y <- rnorm(10)
> x <- rnorm(10)
> lm(y~x)
```

Call:

```
lm(formula = y ~ x)
```

Coefficients:

(Intercept)	x
0.32236	0.03631

```
> nls(y ~ I + A * x, start = list("I" = 0, "A" = 0))
```

Nonlinear regression model

model:  $y \sim I + A * x$

data: parent.frame()

I	A
0.32236	0.03631

residual sum-of-squares: 2.489

Number of iterations to convergence: 1

Achieved convergence tolerance: 1.412e-08

# Starting Values

- A non-linear model needs starting values to optimize from.

```
> nls(y ~ (A / (1 + exp( (B-x) / C ) ) , data = simdf,
+      start = list(A = 1, B = 1, C = 1), trace = TRUE)
5600494.    (2.62e+00): par = (1 1 1)
2358449.    (4.62e+00): par = (156.5259 149.0843 662.4917)
Error in nls(y ~ (A/(1 + exp((B - x)/C))), data = simdf, start = list(A = 1, :
  singular gradient
> ss <- growthSS("logistic", y ~ x, df = simdf, type = "nls")
> fitGrowth(ss, trace = TRUE)
49868.30    (4.00e-10): par = (203.4609 9.635022 2.306136)
Nonlinear regression model
  model: y ~ A/(1 + exp((B - x)/C))
  data: ss[["df"]]
      A      B      C
203.461  9.635  2.306
residual sum-of-squares: 49868
```

- These can be tricky, so “self-starting” models are preferred.

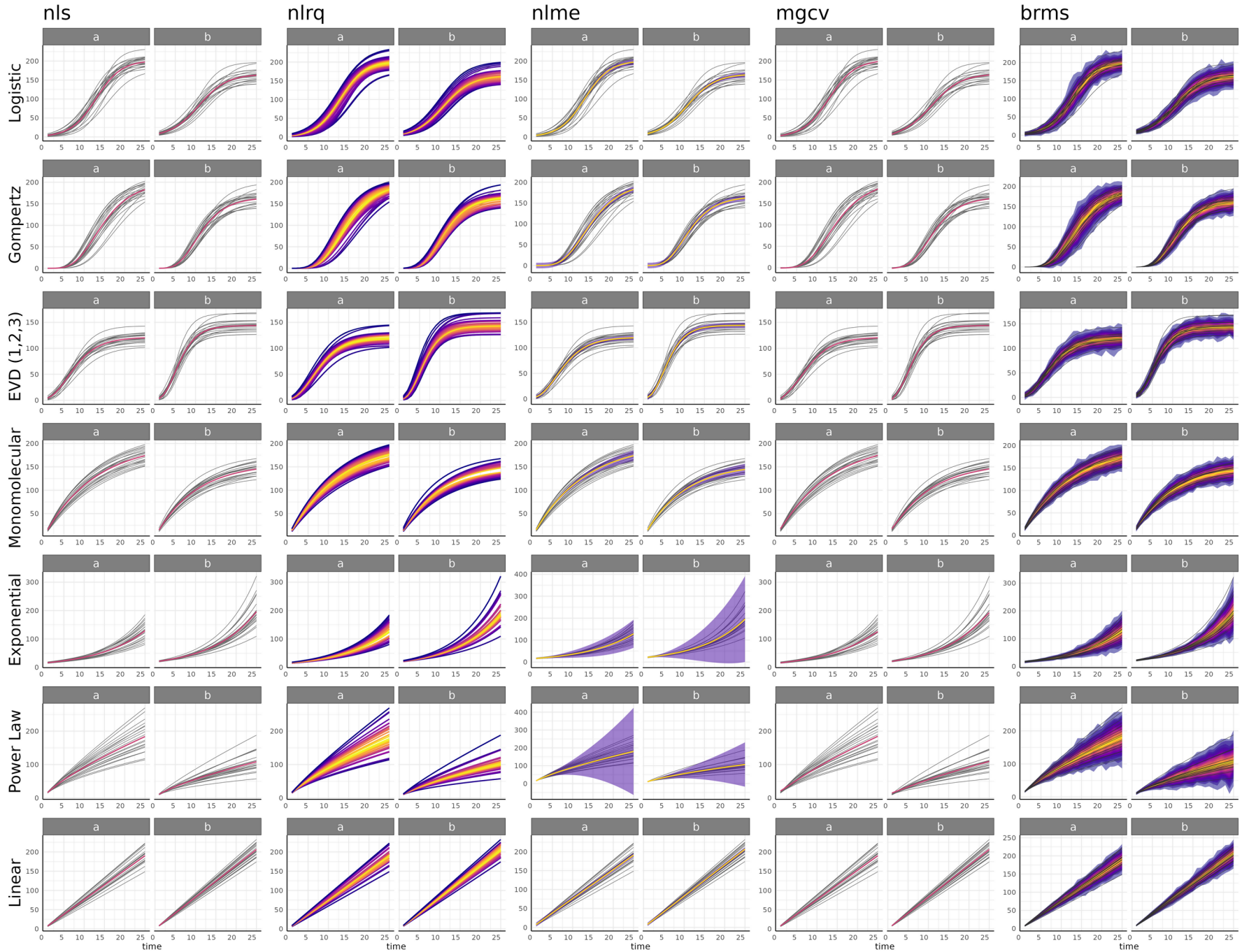
```
Number of iterations to convergence: 0
Achieved convergence tolerance: 4.003e-10
> rbind(ss$start)
      A      B      C
[1,] 203.4609 9.635022 2.306136
```

# pcvr::growthSS

- The `growthSS` function specifies self-starting non-linear models using common growth models across several model backends.



# pcvr::growthSS Main Options



# pcvr::growthSS Additional Terms

- There are several keywords that can be used in the `growthSS` model argument to:
  - add intercept terms
  - switch to modeling decay
  - Model time-to-event data
  - Change Distributions
  - Specify changepoint models

# pcvr::fitGrowth

- `growthSS` specifies a list that can be used by other downstream functions, but it does not actually fit the model, that is done by `fitGrowth`.

# pcvr::fitGrowth

- `growthSS` specifies a list that can be used by other downstream functions, but it does not actually fit the model, that is done by `fitGrowth`.
- In general only the output of `growthSS` needs to be passed to `fitGrowth`, but you can pass other arguments on to the model backend as well.

# pcvr::testGrowth

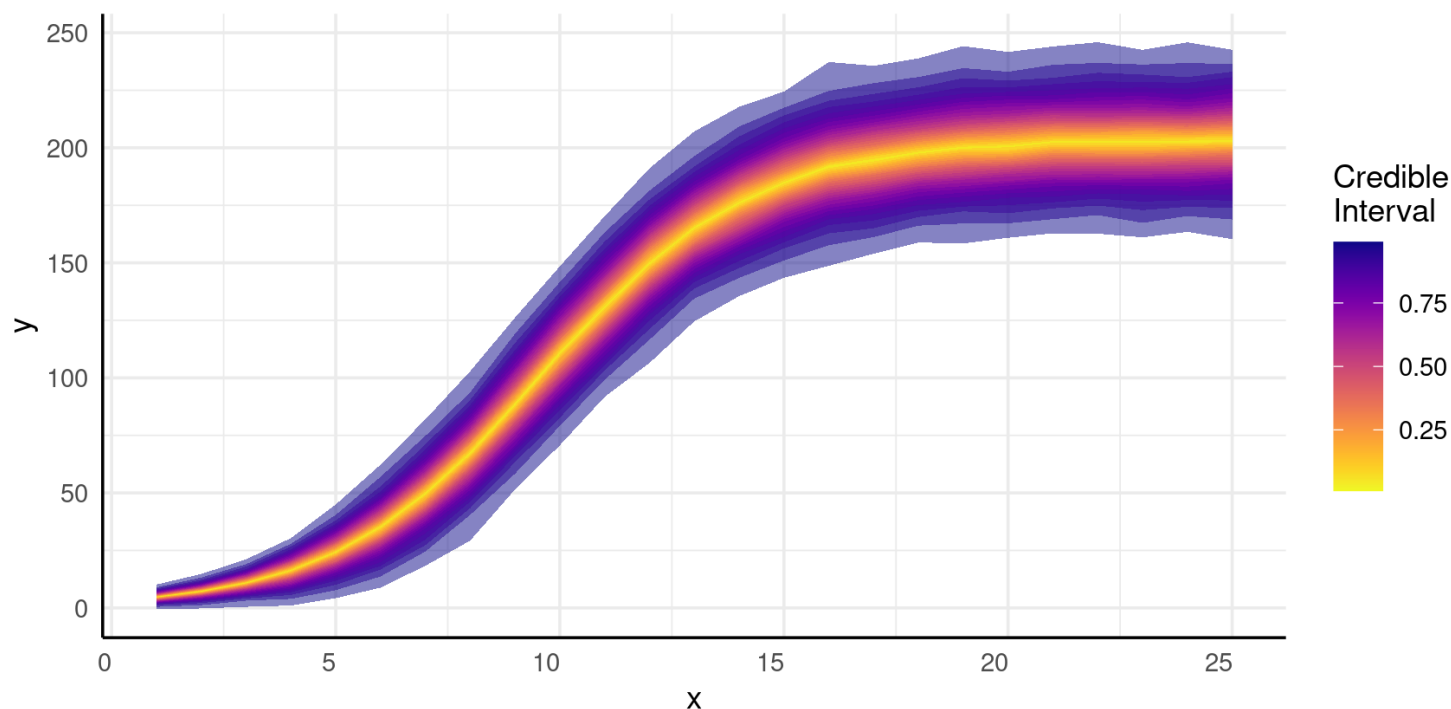
- **fitGrowth** will return a model using the specified backend. Any of those models can have a variety of hypotheses tested using **testGrowth**.
  - The details of what kinds of linear and non-linear hypotheses can be tested is shown in the **testGrowth** documentation.

# pcvr::growthPlot

- The models made from `fitGrowth` can also be visualized using `growthPlot`.
  - See previous figure of main `growthSS` options.
  - The details of the plot depend on the model backend that was used.

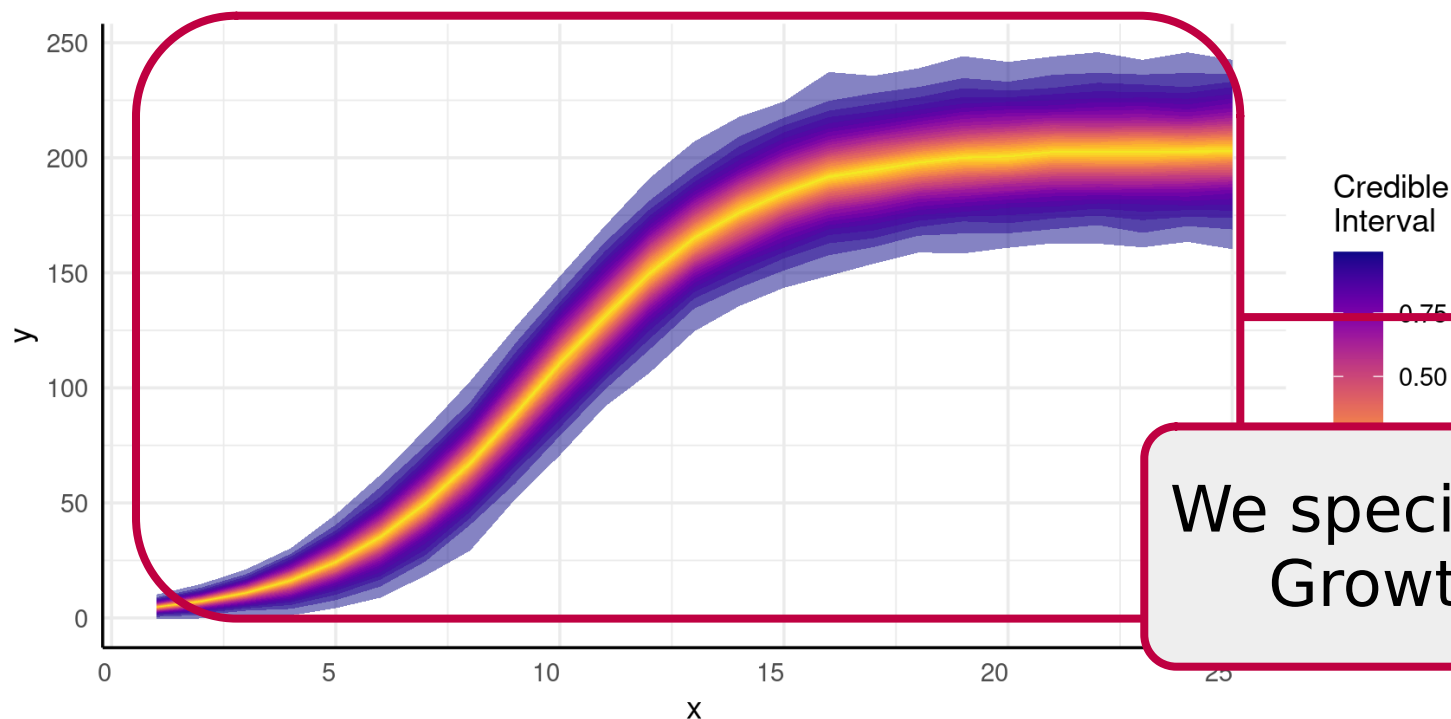
# Example Longitudinal Model

```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
              sigma = "logistic",  
              start = list("A" = 130, "B" = 10, "C" = 3,  
                           "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



# Example Longitudinal Model

```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
               sigma = "logistic",  
               start = list("A" = 130, "B" = 10, "C" = 3,  
                             "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```

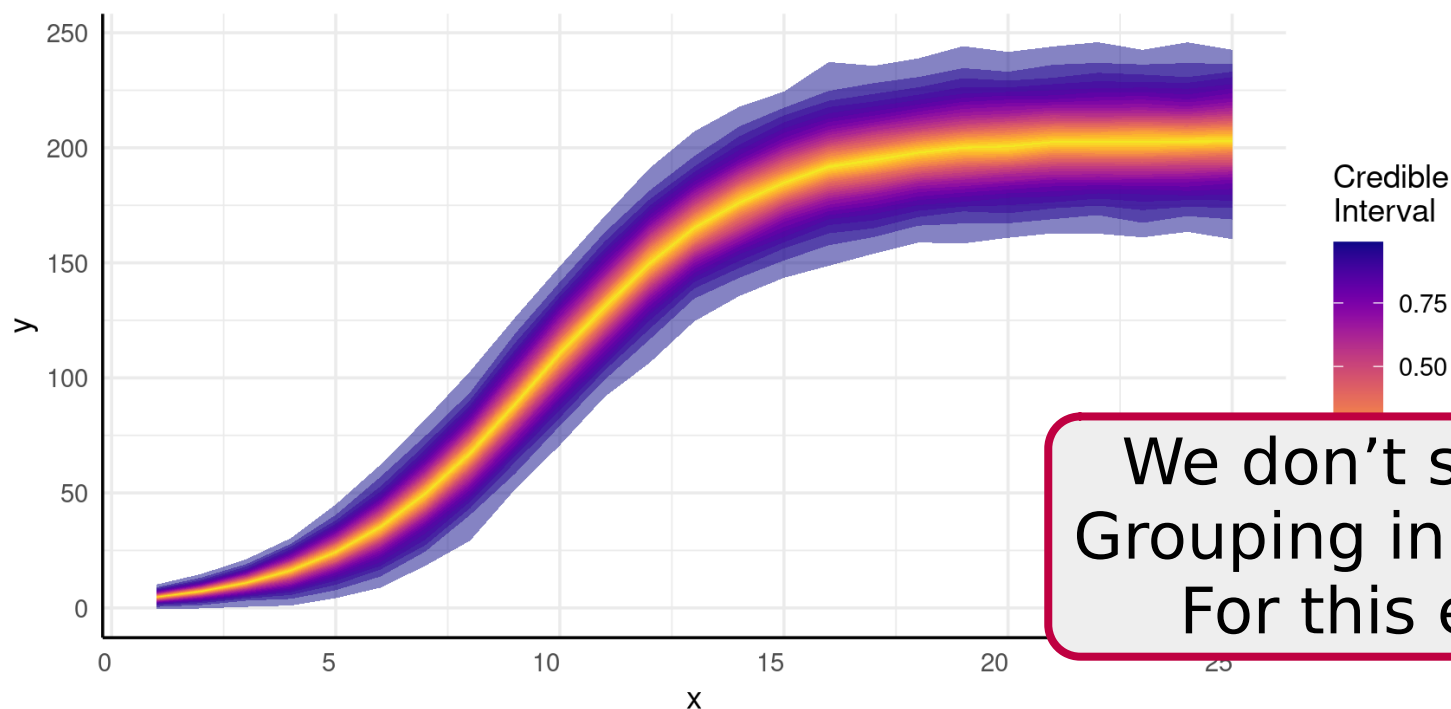


We specify a Logistic Growth model.



# Example Longitudinal Model

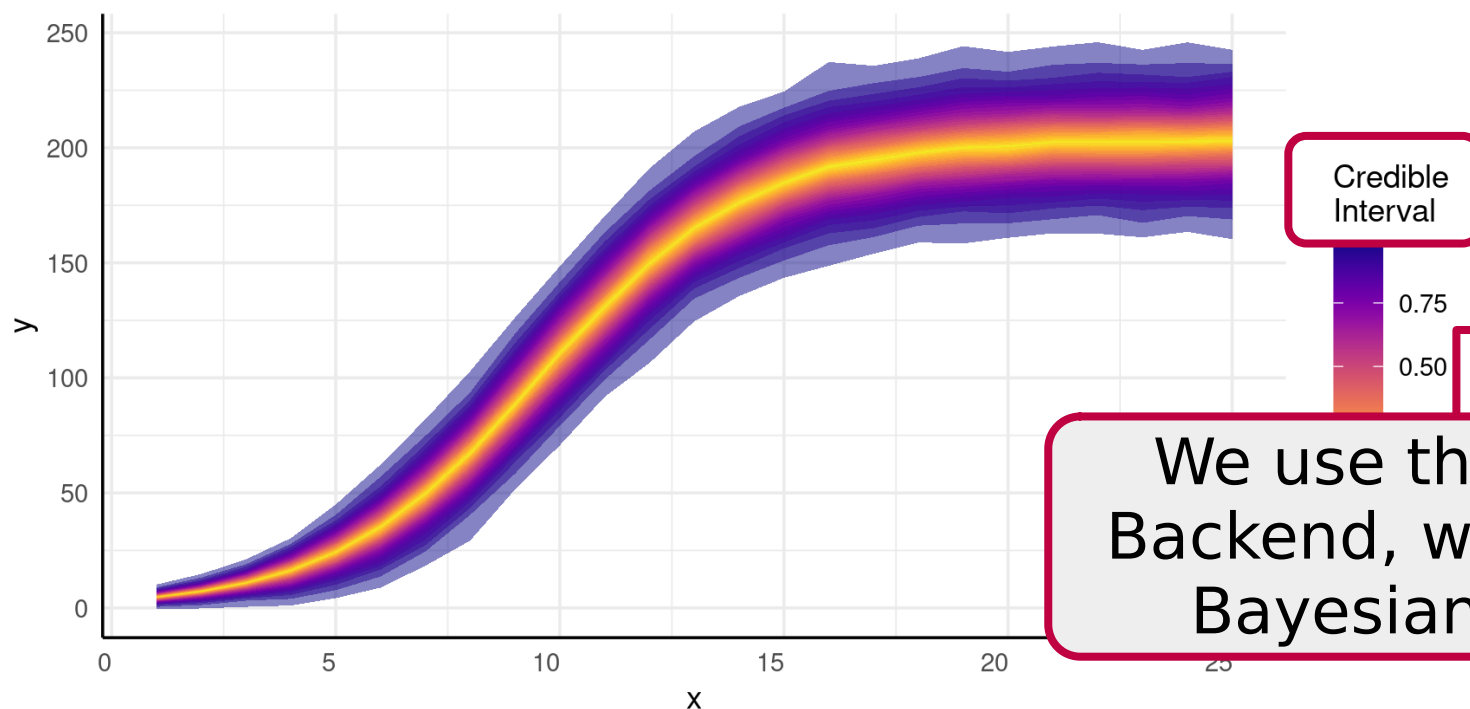
```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
              sigma = "logistic",  
              start = list("A" = 130, "B" = 10, "C" = 3,  
                           "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



We don't specify any Grouping in the formula For this example.

# Example Longitudinal Model

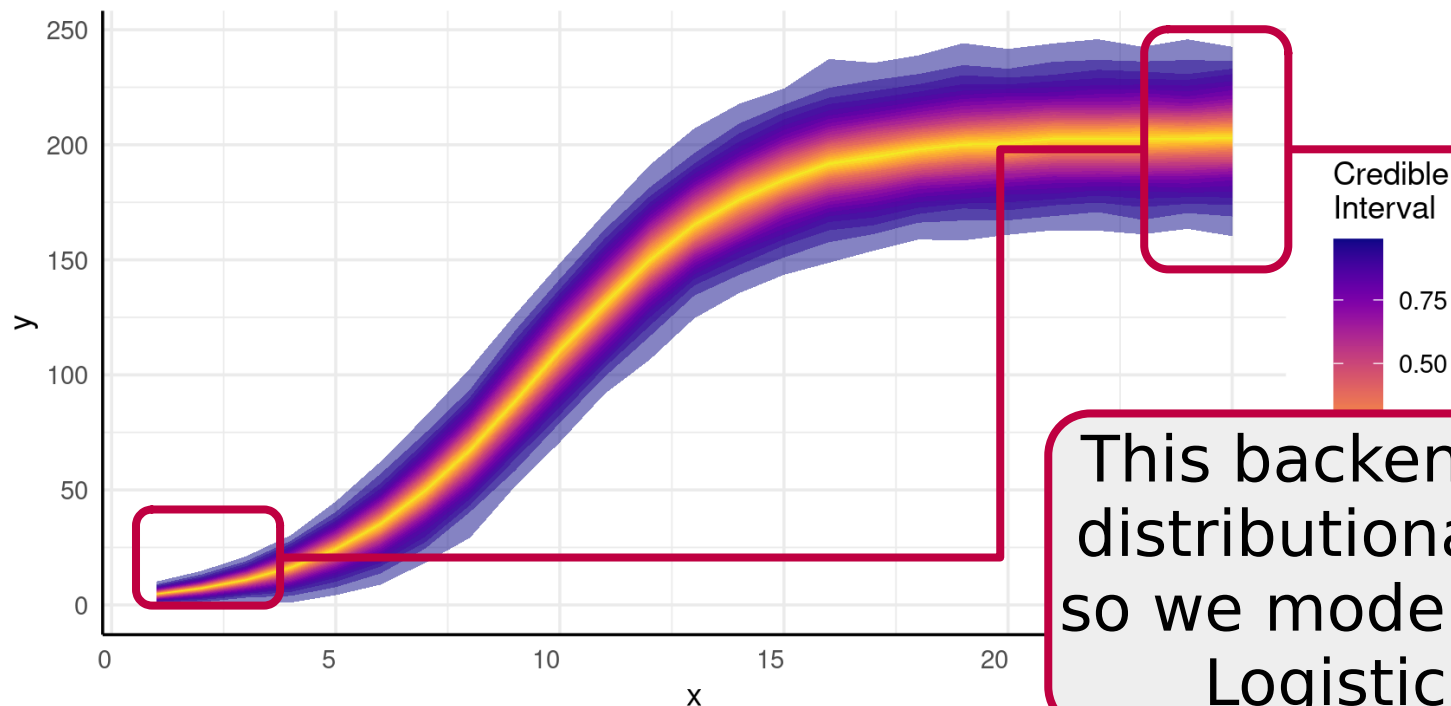
```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
              sigma = "logistic",  
              start = list("A" = 130, "B" = 10, "C" = 3,  
                           "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



We use the “brms” Backend, which fits a Bayesian model.

# Example Longitudinal Model

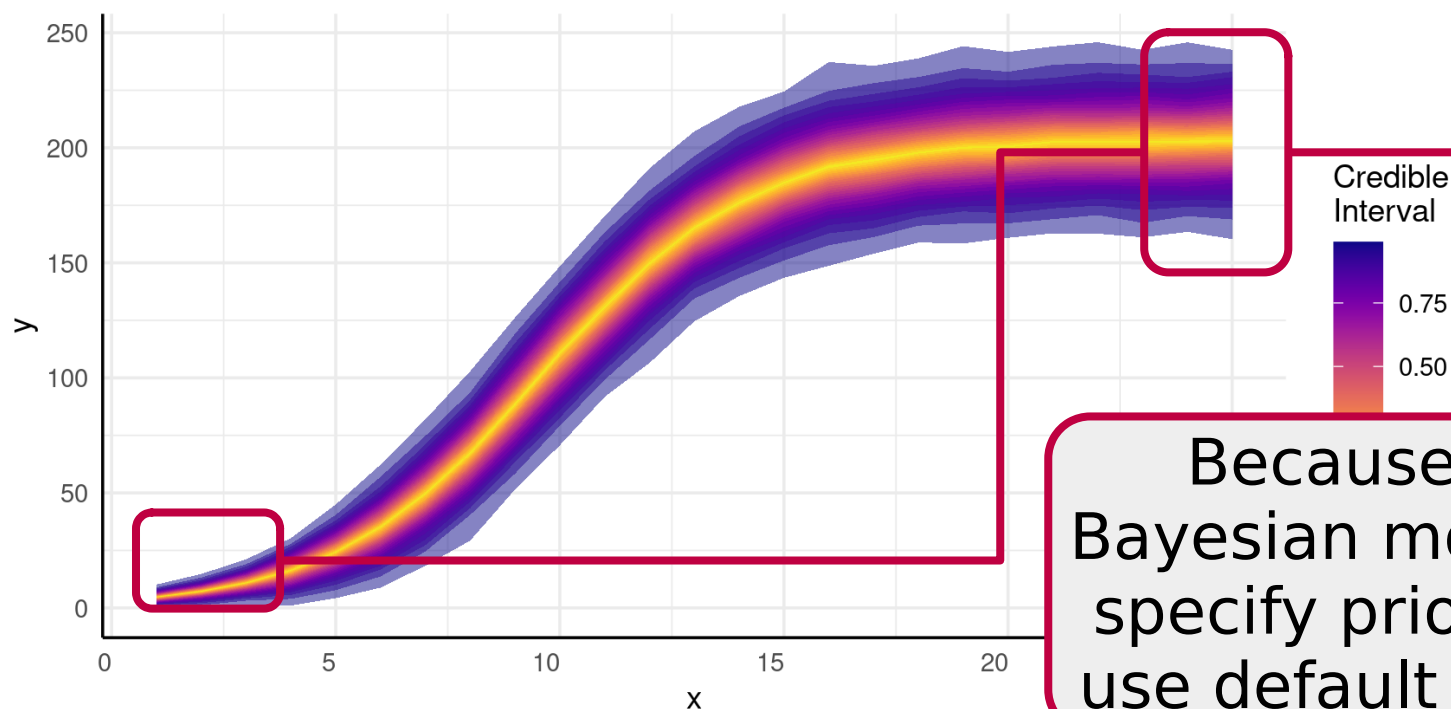
```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
  sigma = "logistic",  
  start = list("A" = 130, "B" = 10, "C" = 3,  
    "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



This backend allows for distributional modeling, so we model variance as Logistic as well.

# Example Longitudinal Model

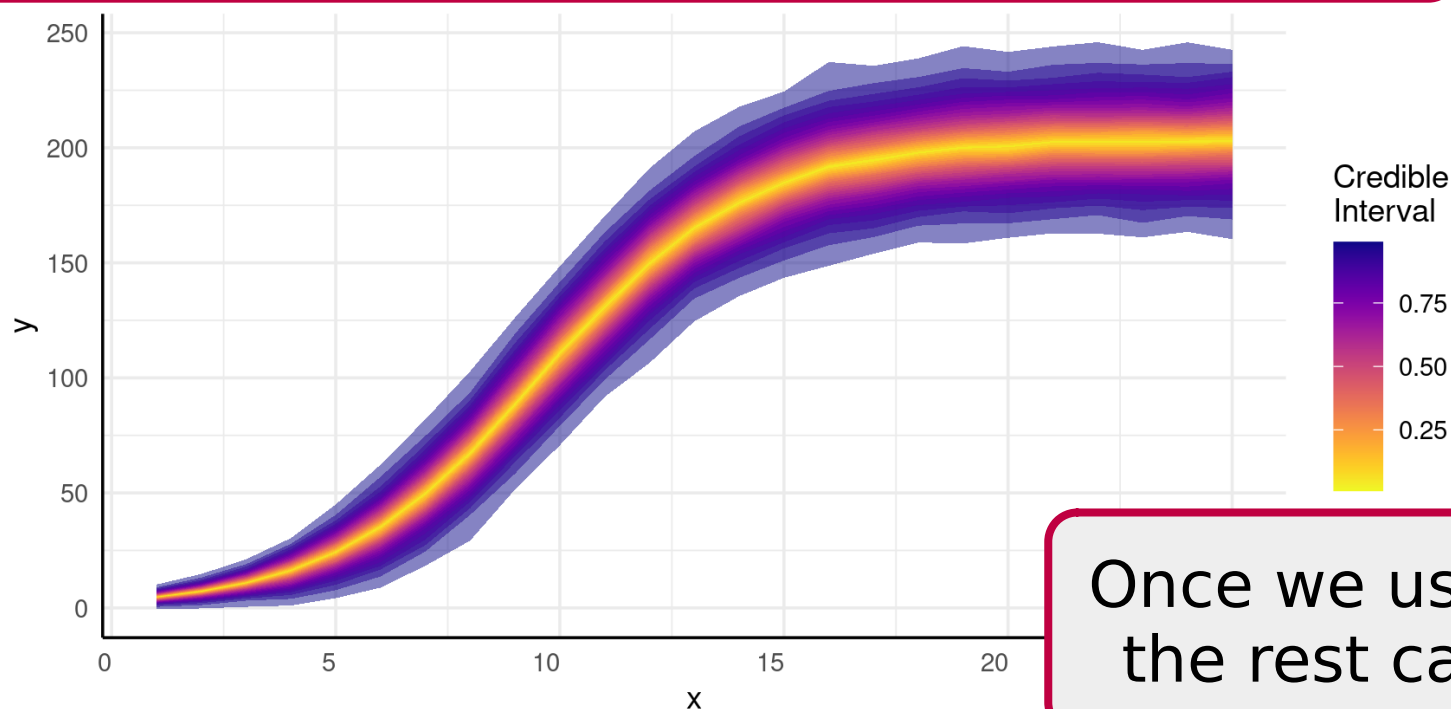
```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
              sigma = "logistic",  
              start = list("A" = 130, "B" = 10, "C" = 3,  
                           "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



Because this is a Bayesian model we also specify priors, here we use default lognormals.

# Example Longitudinal Model

```
ss <- growthSS("logistic", y ~ x, df = simdf, type = "brms",  
  sigma = "logistic",  
  start = list("A" = 130, "B" = 10, "C" = 3,  
    "sigmaA" = 10, "sigmaB" = 10, "sigmaC" = 2))  
fit <- fitGrowth(ss, iter = 1000, chains = 4, cores = 4)  
growthPlot(fit, form = ss$pcvrForm)
```



Once we use `growthSS`  
the rest can just run.

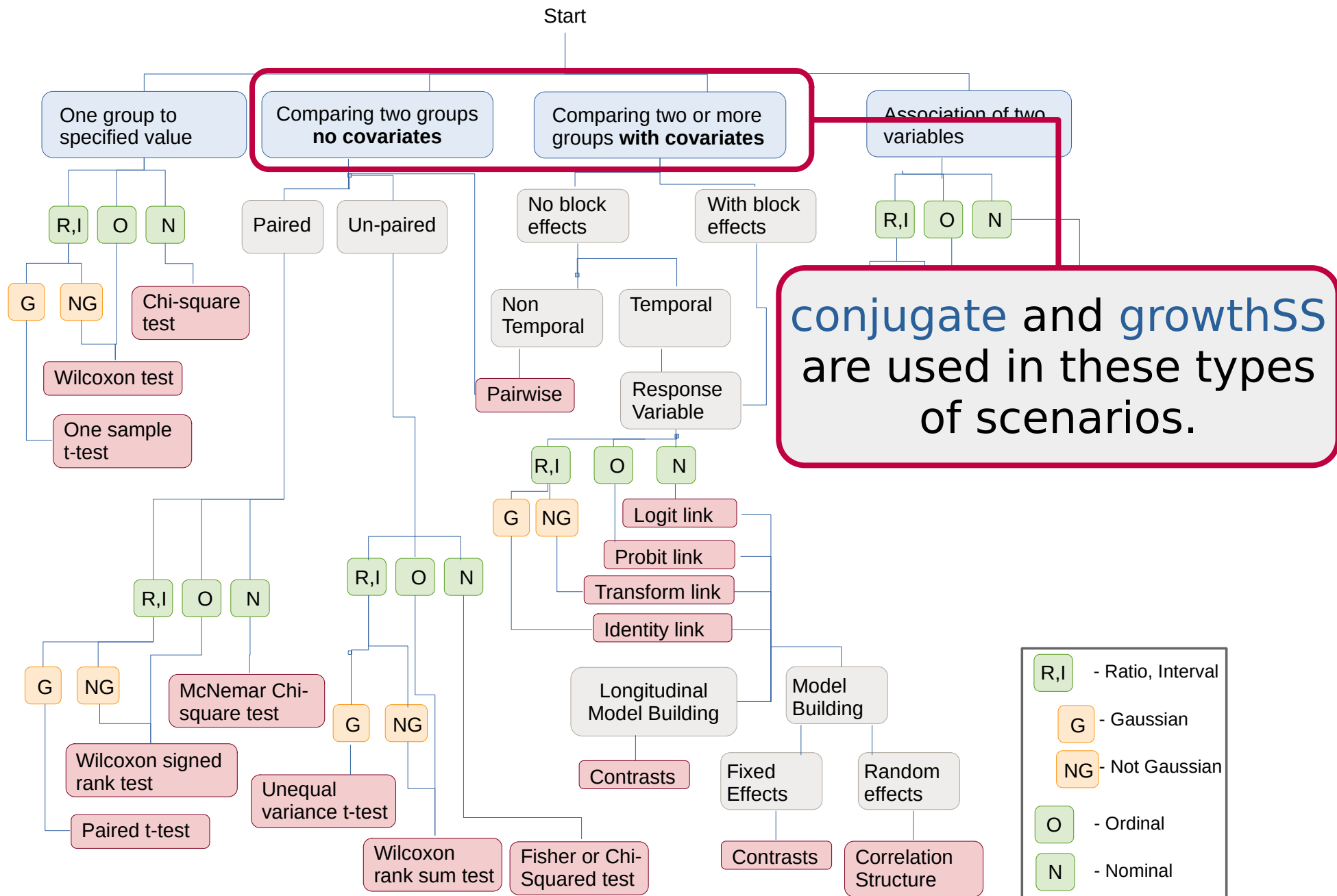
# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- Resources

Those were the two pcvr options we are using today, take a break and we'll end by applying them to some examples.

# Example Scenarios

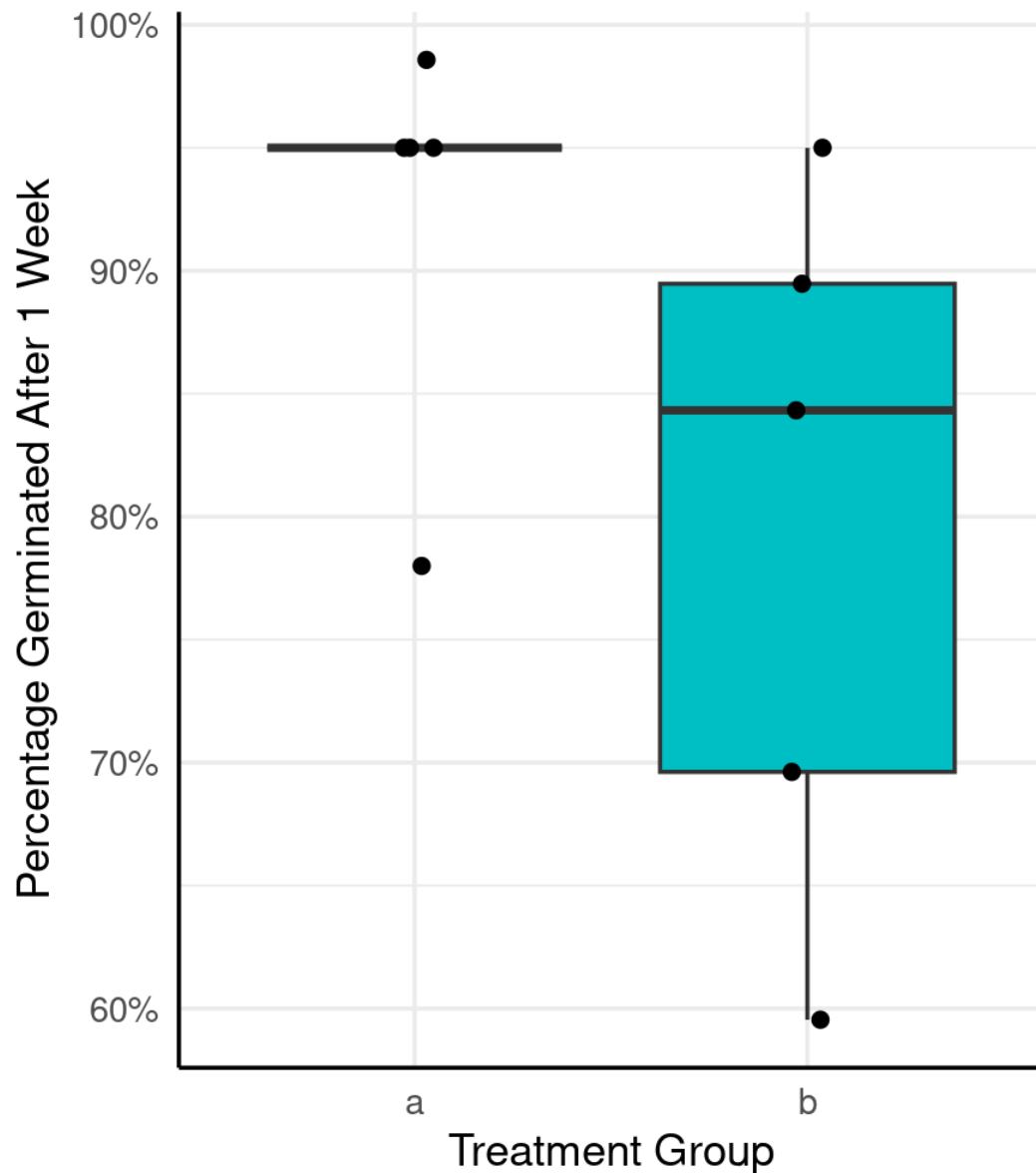
- The rest of today will be similar to the *Stats in R* workshop, but we'll use `pcvr` functions and highlight any differences between those and the standard endpoints.



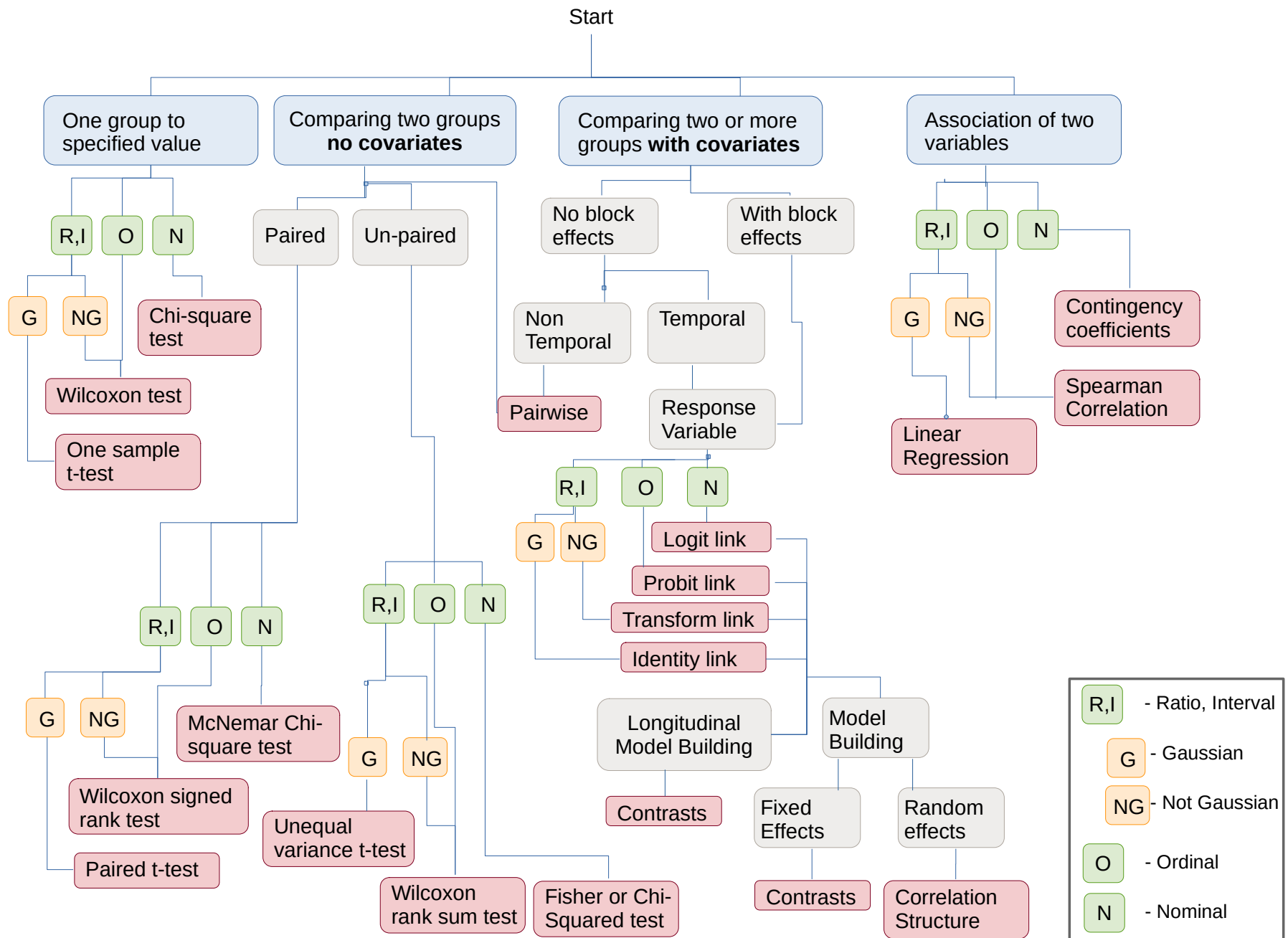
\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better



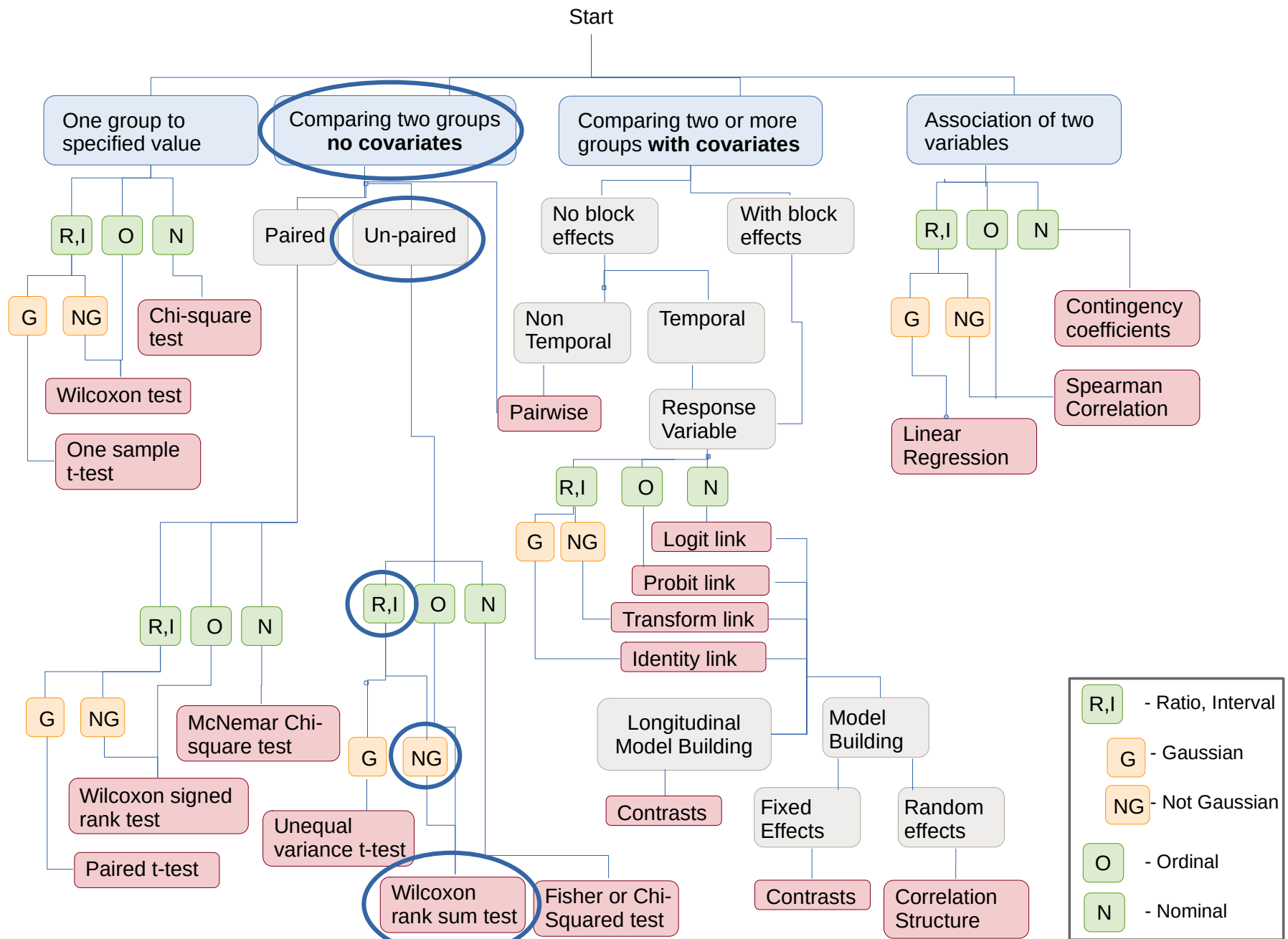
# Scenario 1



- You are curious if the germination between Heat treated and Control seeds is different after 1 week. You collect 5 reps from each group and are all set to compare them. What test do you use?

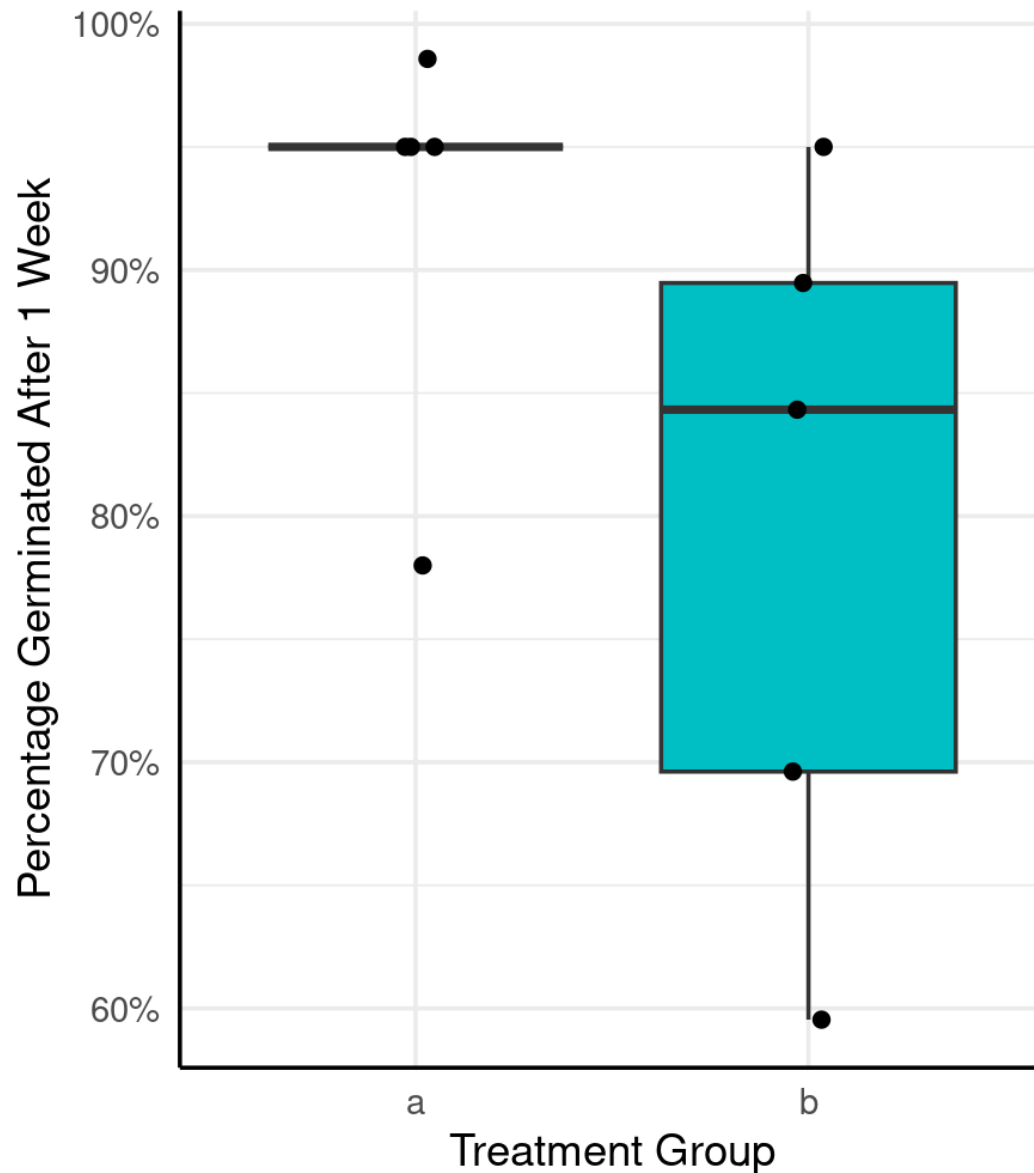


\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better



\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better

# Scenario 1 – Wilcoxon Rank Sum Test



- Wilcoxon Rank Sum is a fine choice, but it does require us to break ties in the ranking and it has low power with only 5 reps per group.

# Scenario 1 – Wilcoxon Rank Sum Test

```
> wilcox.test(values ~ group, df)
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: values by group
```

```
W = 20.5, p-value = 0.106
```

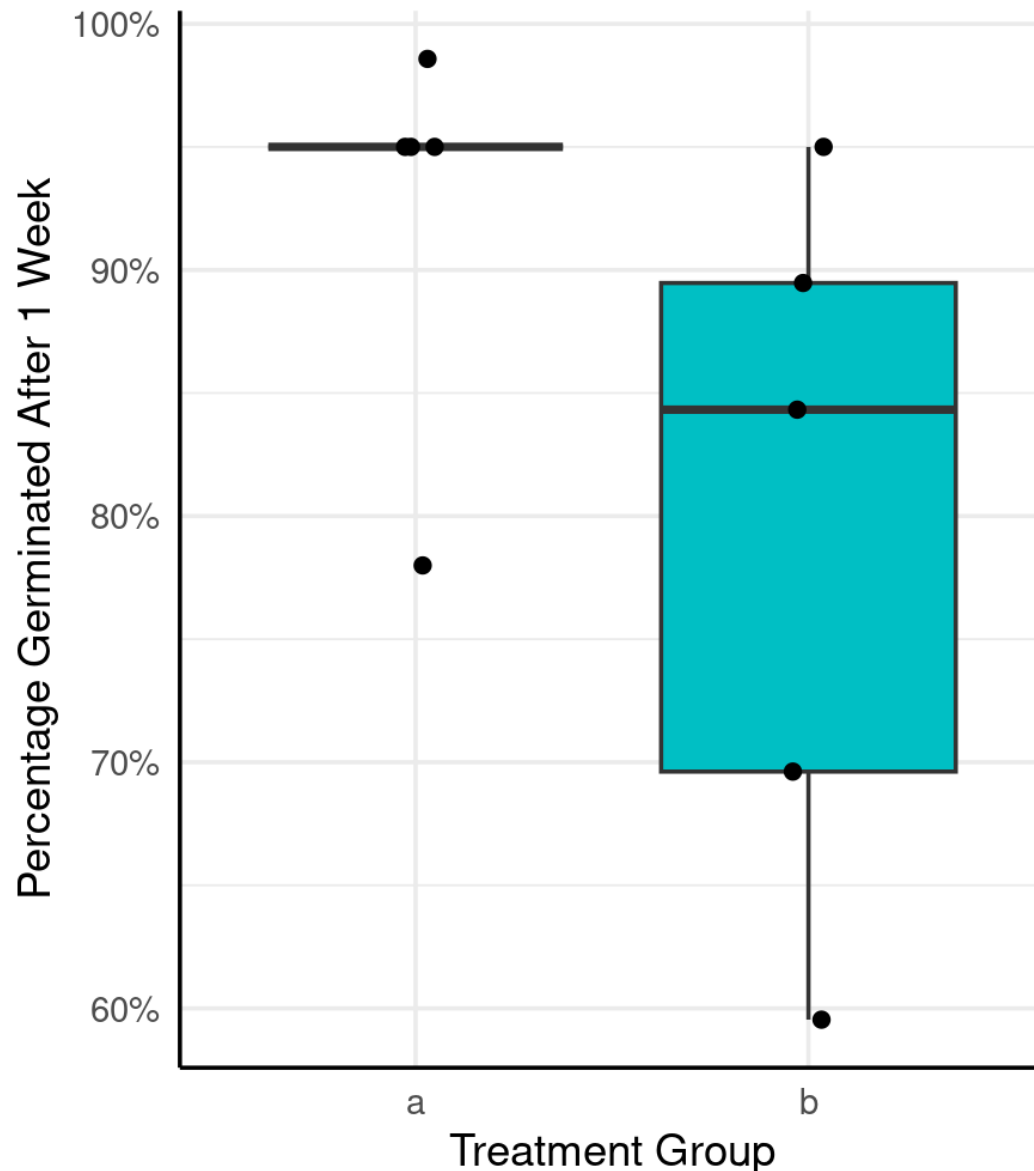
```
alternative hypothesis: true location shift is not equal to 0
```

```
Warning message:
```

```
In wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...) :  
cannot compute exact p-value with ties
```

- We get a P-value, but we don't have an estimate of what the difference is.

# Scenario 1 – `pcvr::conjugate`



- Germination is a percentage.
- Depending on the data format we might use the binomial (for counts/total) or the beta (for pure percentages).

# Scenario 1 – `pcvr::conjugate`

```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

We give the data as two numeric samples.

# Scenario 1 – `pcvr::conjugate`

```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

We specify the distribution  
and a (weak) prior.



# Scenario 1 – `pcvr::conjugate`

```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

We return a plot  
of the Posterior.

# Scenario 1 – `pcvr::conjugate`

```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,   
                      hypothesis = "unequal")
```

We don't run ROPE comparisons here.

# Scenario 1 – `pcvr::conjugate`

```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

We return the Probability that the groups are unequal.

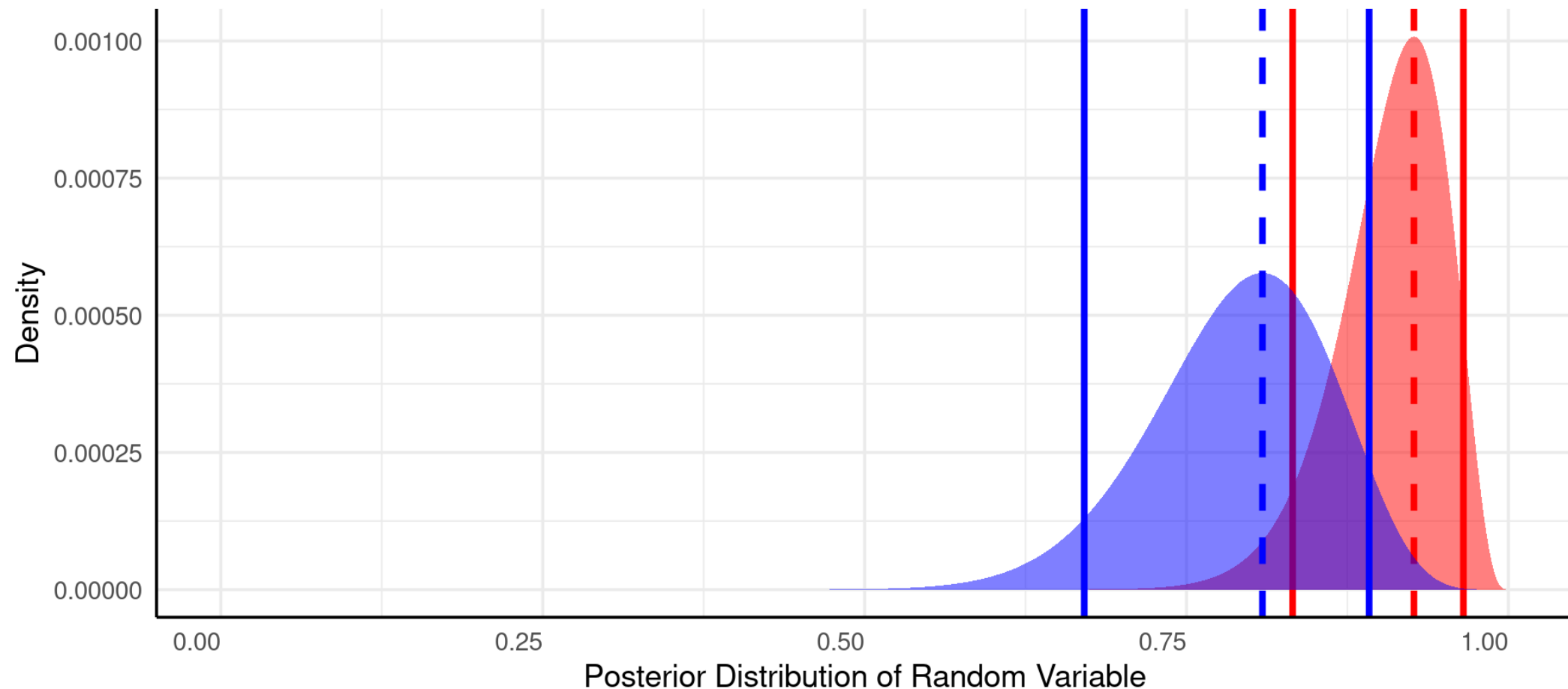
# Scenario 1 – Beta Distribution

## Distribution of Samples

Sample 1: 0.93 [0.83, 0.97]

Sample 2: 0.81 [0.67, 0.89]

$P[p_1 \neq p_2] = 0.70952$



# Scenario 1 – `pcvr::conjugate`

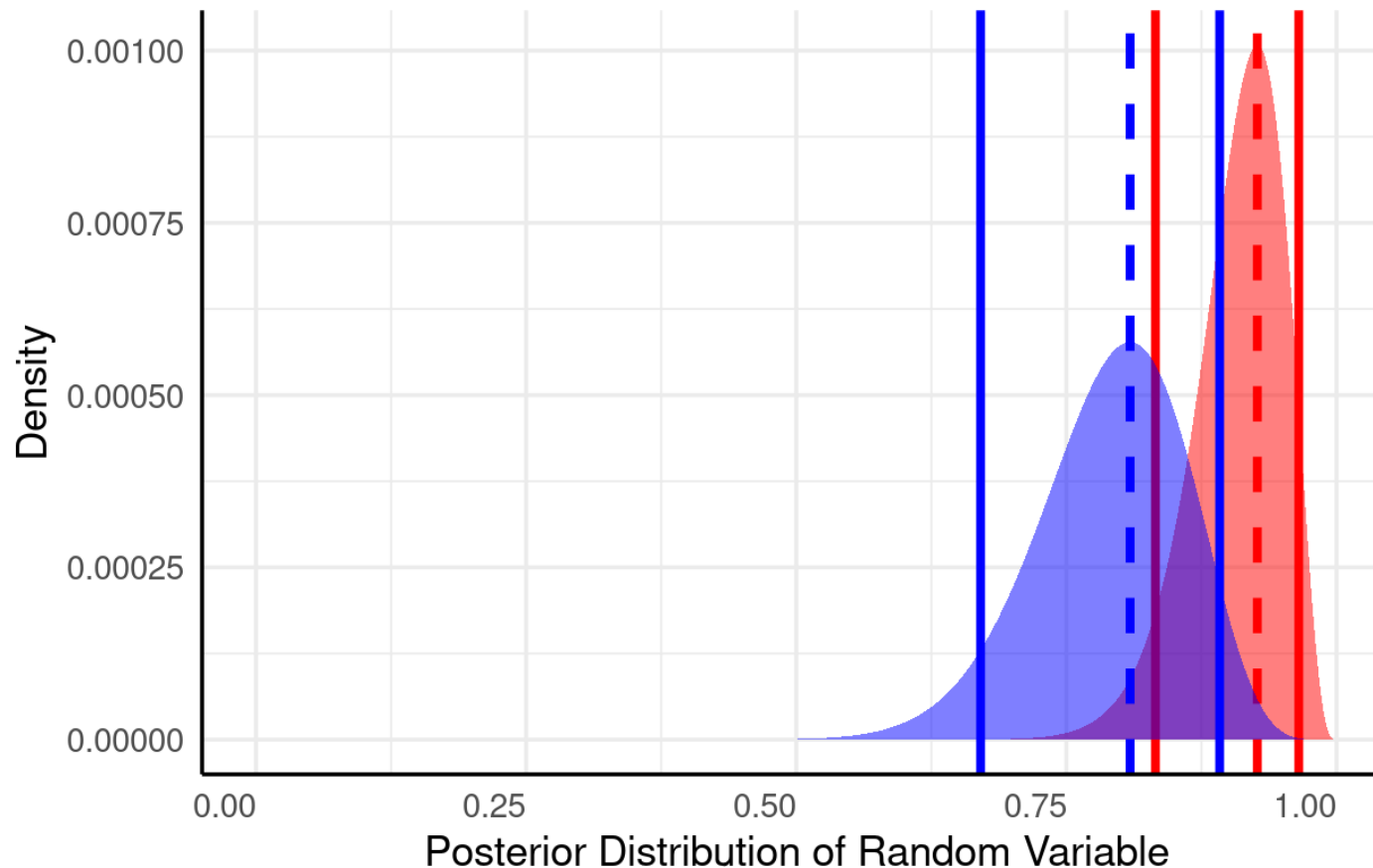
```
res <- pcvr::conjugate(df[df$group == "a", "values"],  
                      df[df$group == "b", "values"],  
                      method = "beta",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE, rope_range = c(-0.025, 0.025),  
                      hypothesis = "unequal")
```

Now we also ask for the Probability that the difference In groups is within  $[+/-2.5]$ .

# Scenario 1 – pcvr::conjugate

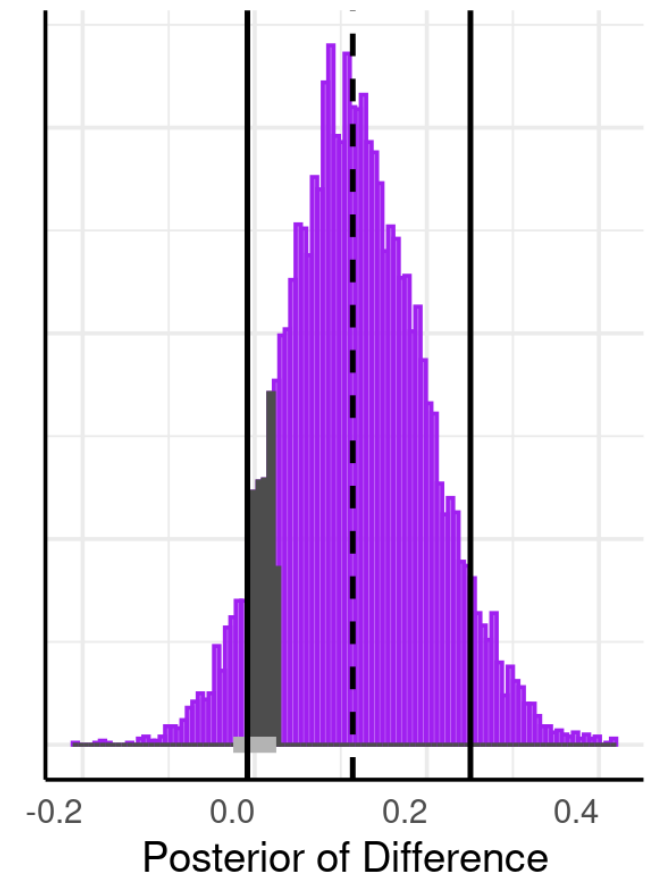
## Distribution of Samples

Sample 1: 0.93 [0.83, 0.97]  
Sample 2: 0.81 [0.67, 0.89]  
 $P[p1 \neq p2] = 0.70952$



## Distribution of Difference

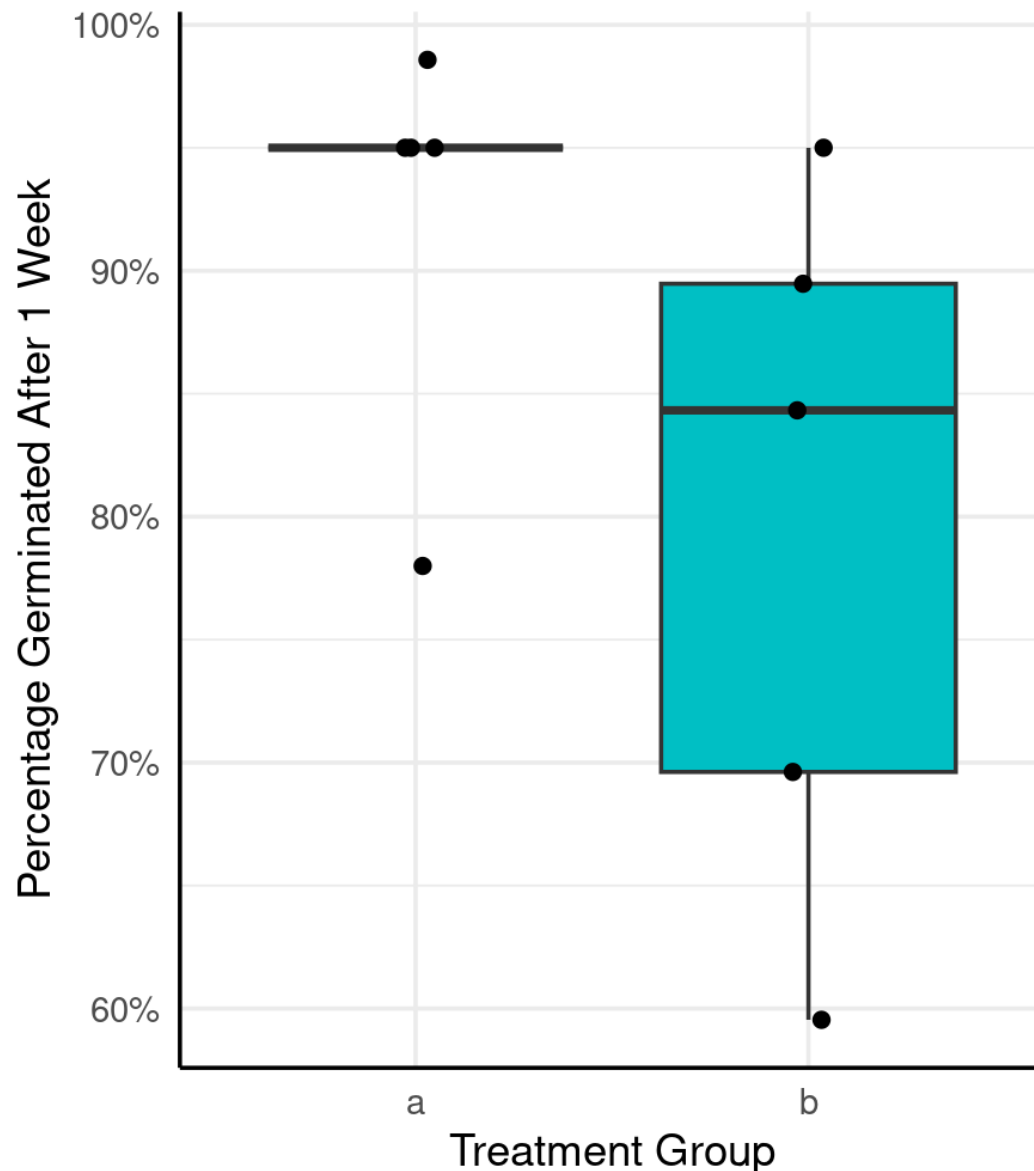
Median Difference of 0.11  
89% CI [-0.01, 0.25]  
0.89% HDI in [-0.025, 0.025]: 0.08



# Scenario 1 – Takeaway Points

- `conjugate` provides another option when non-parametrics may lack power or not return some of the information you are interested in.
- The cost is that you have to tell it some distribution to use.
  - There are examples and explanations for these distributions in the documentation.

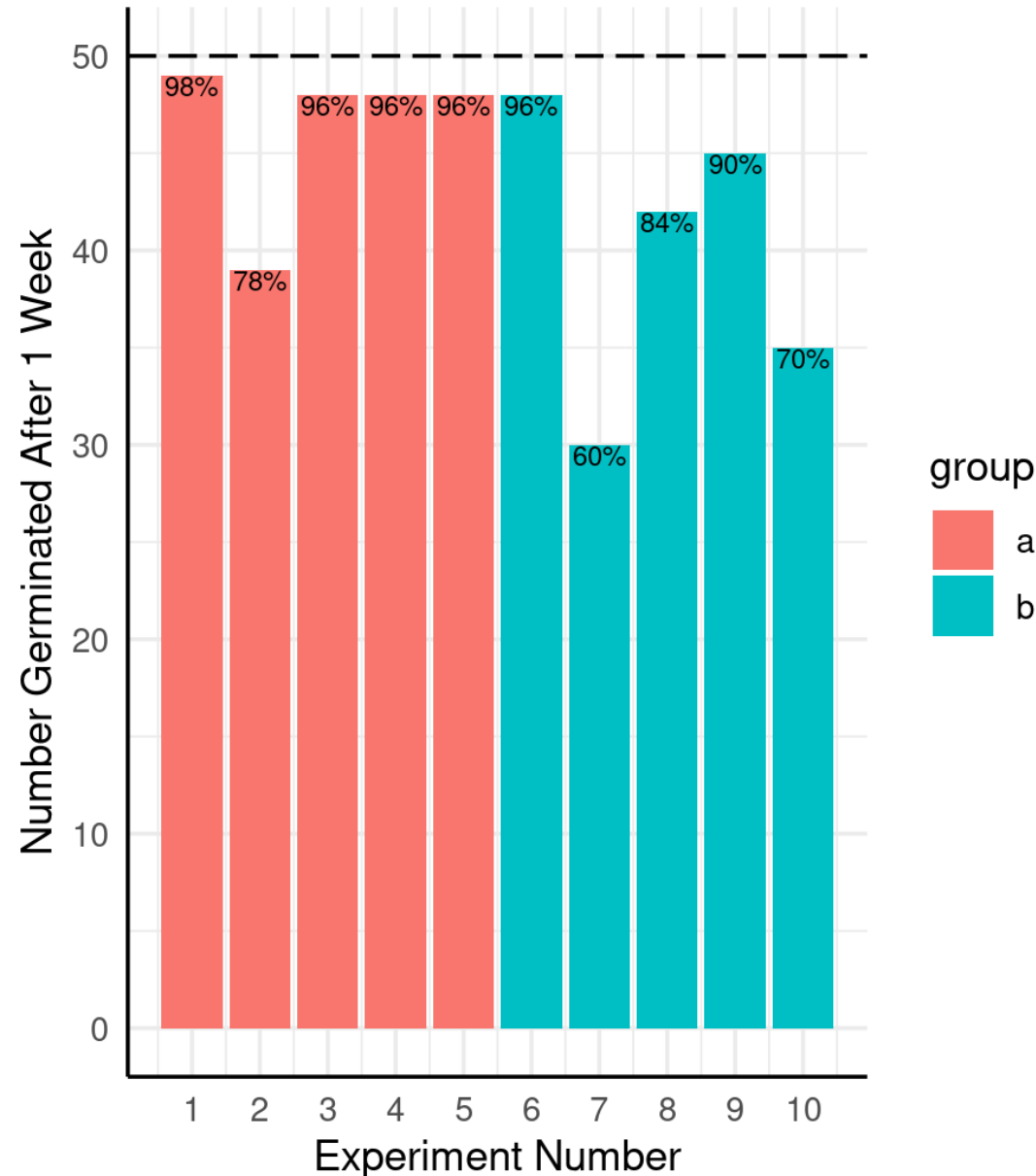
# Scenario 1.5



- You are curious if the germination between Heat treated and Control seeds is different after 1 week. You collect 5 reps from each group and are all set to compare them. What test do you use?
- This isn't really how our data generally works, we get percentages given a number of trials.

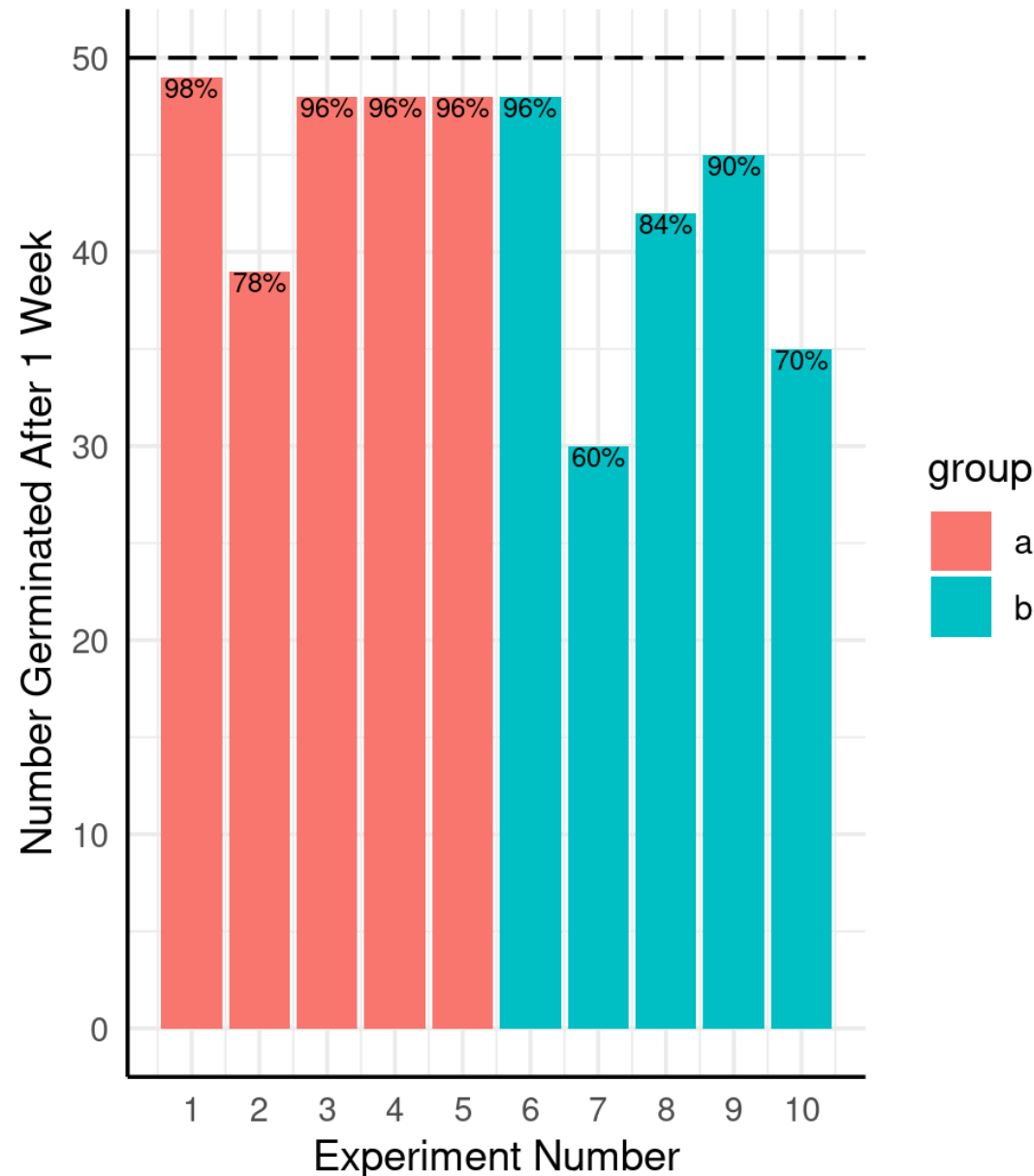


# Scenario 1.5



- You are curious if the germination between Heat treated and Control seeds is different after 1 week. You collect 5 reps from each group and are all set to compare them. What test do you use?
- Our data actually looks more like this

# Scenario 1.5



- You are curious if the germination between Heat treated and Control seeds is different after 1 week. You collect 5 reps from each group and are all set to compare them. What test do you use?
- Our data actually looks more like this

# Scenario 1.5 – `pcvr::conjugate`

```
dfa <- df[df$group == "a", ]  
dfb <- df[df$group == "b", ]  
res <- pcvr::conjugate(s1 = list(successes = dfa$successes, trials = dfa$trials),  
                      s2 = list(successes = dfb$successes, trials = dfb$trials),  
                      method = "binomial",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

The beta-binomial method is Unique in that it takes a list of Data with success and trial Counts.

# Scenario 1.5 – `pcvr::conjugate`

```
dfa <- df[df$group == "a", ]  
dfb <- df[df$group == "b", ]  
res <- pcvr::conjugate(s1 = list(successes = dfa$successes, trials = dfa$trials),  
                      s2 = list(successes = dfb$successes, trials = dfb$trials),  
                      method = "binomial",  
                      priors = list(a = 3, b = 1),  
                      plot = TRUE,  
                      hypothesis = "unequal")
```

The prior is still a Beta, like we  
Looked at in the first example  
Of conjugacy.

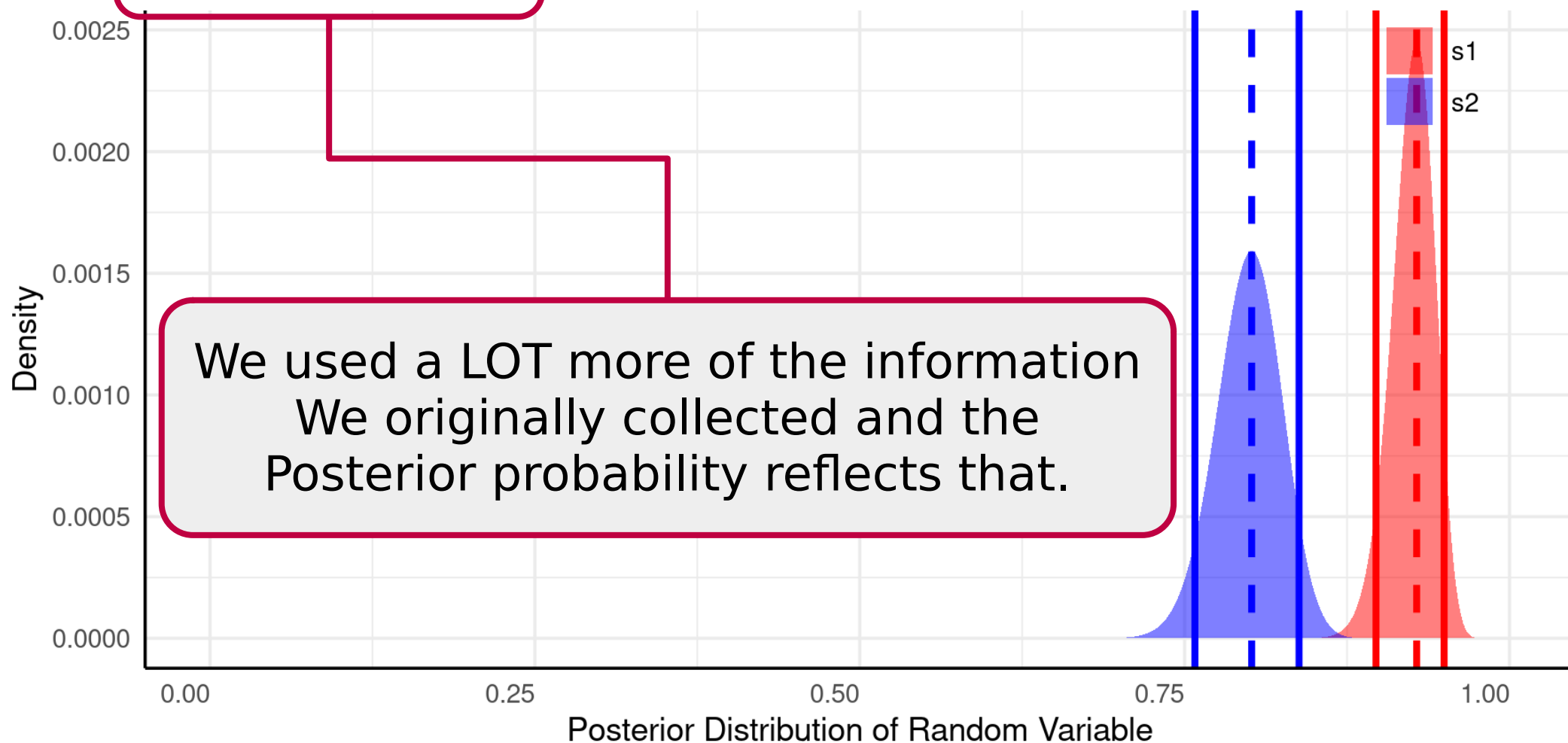
# Scenario 1.5 – `pcvr::conjugate`

## Distribution of Samples

Sample 1: 0.93 [0.9, 0.95]

Sample 2: 0.8 [0.76, 0.84]

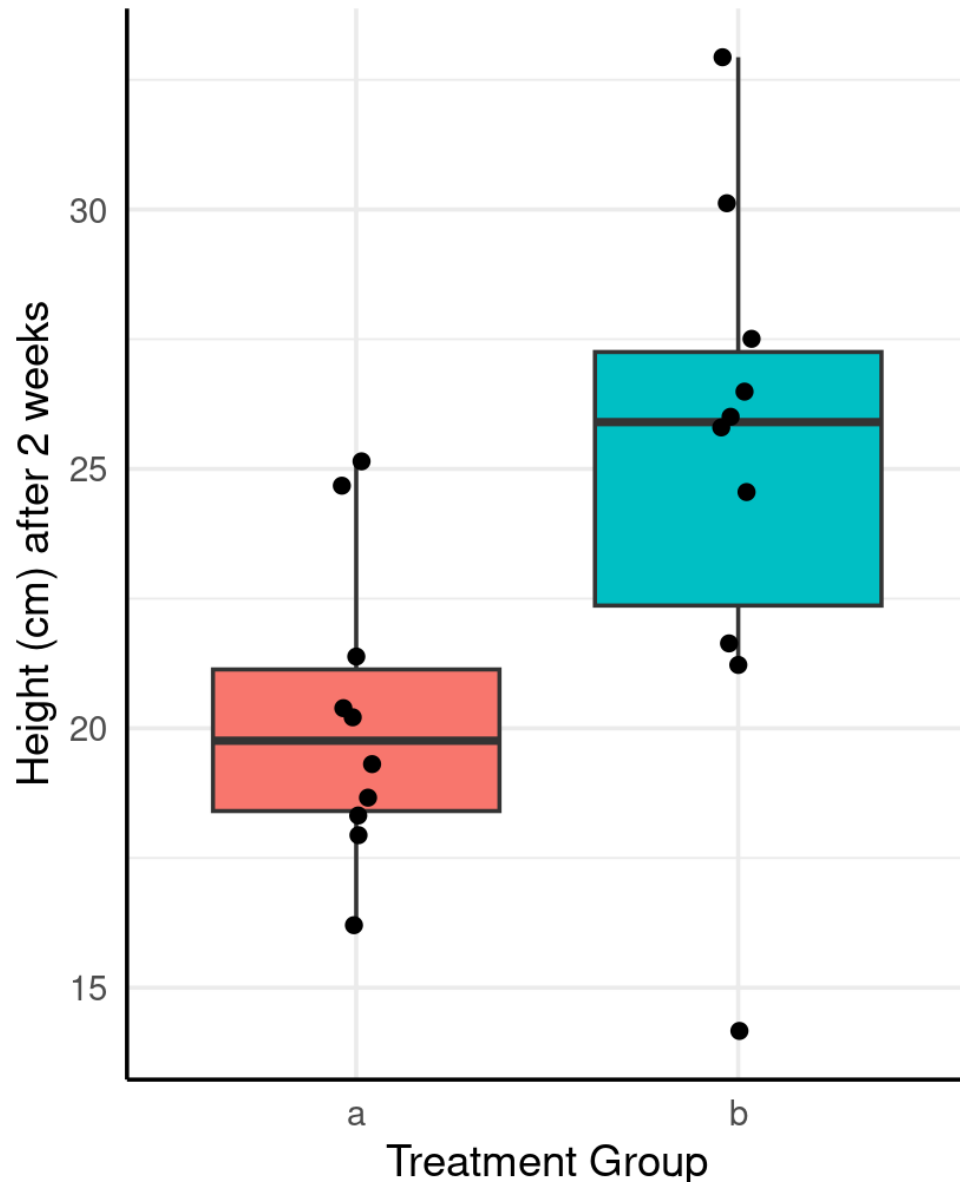
$P[p1 \neq p2] = 0.99716$



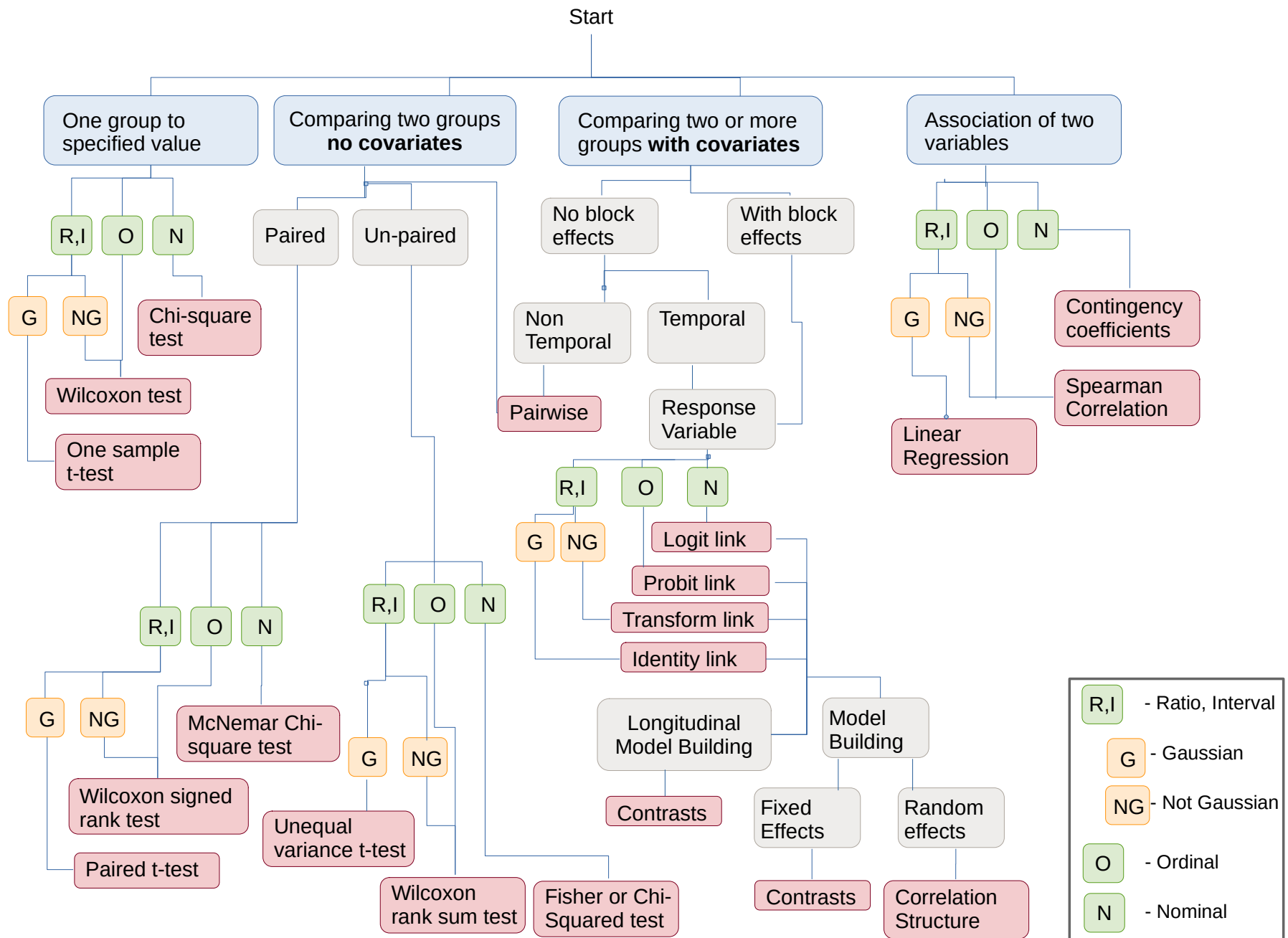
# Scenario 1.5 – Takeaway Points

- The more of your data you can use the better conclusions you can draw.
- Including the “context” of the percentages from example 1 in effect gave us 50x more information to fuel our decisionmaking.
- *This is something to keep in mind throughout experimental design, data collection, and analysis.*

# Scenario 2

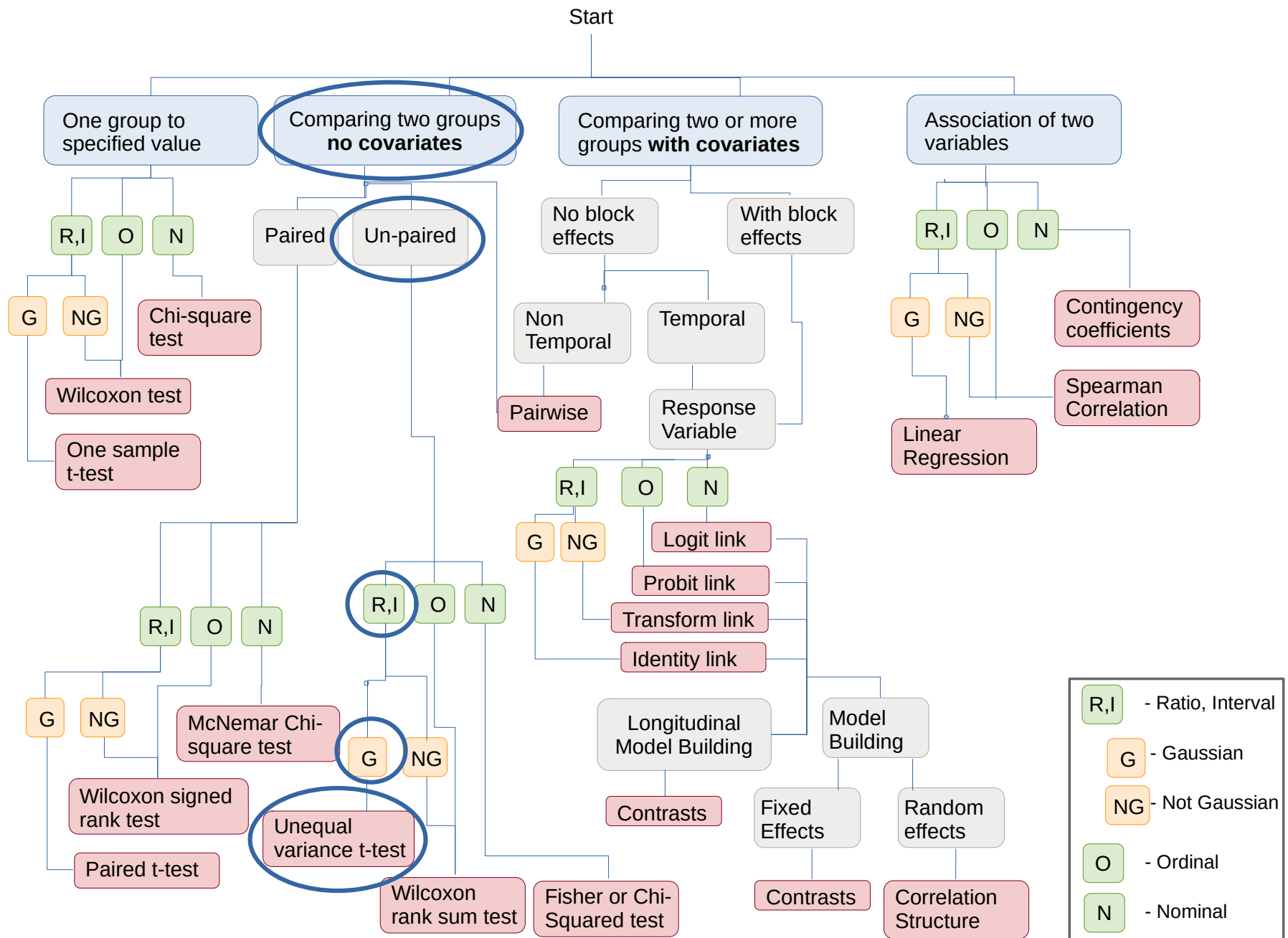


- You are comparing plant heights between two treatment groups (say, control vs cold stress). Given this data how do you do that?



\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better

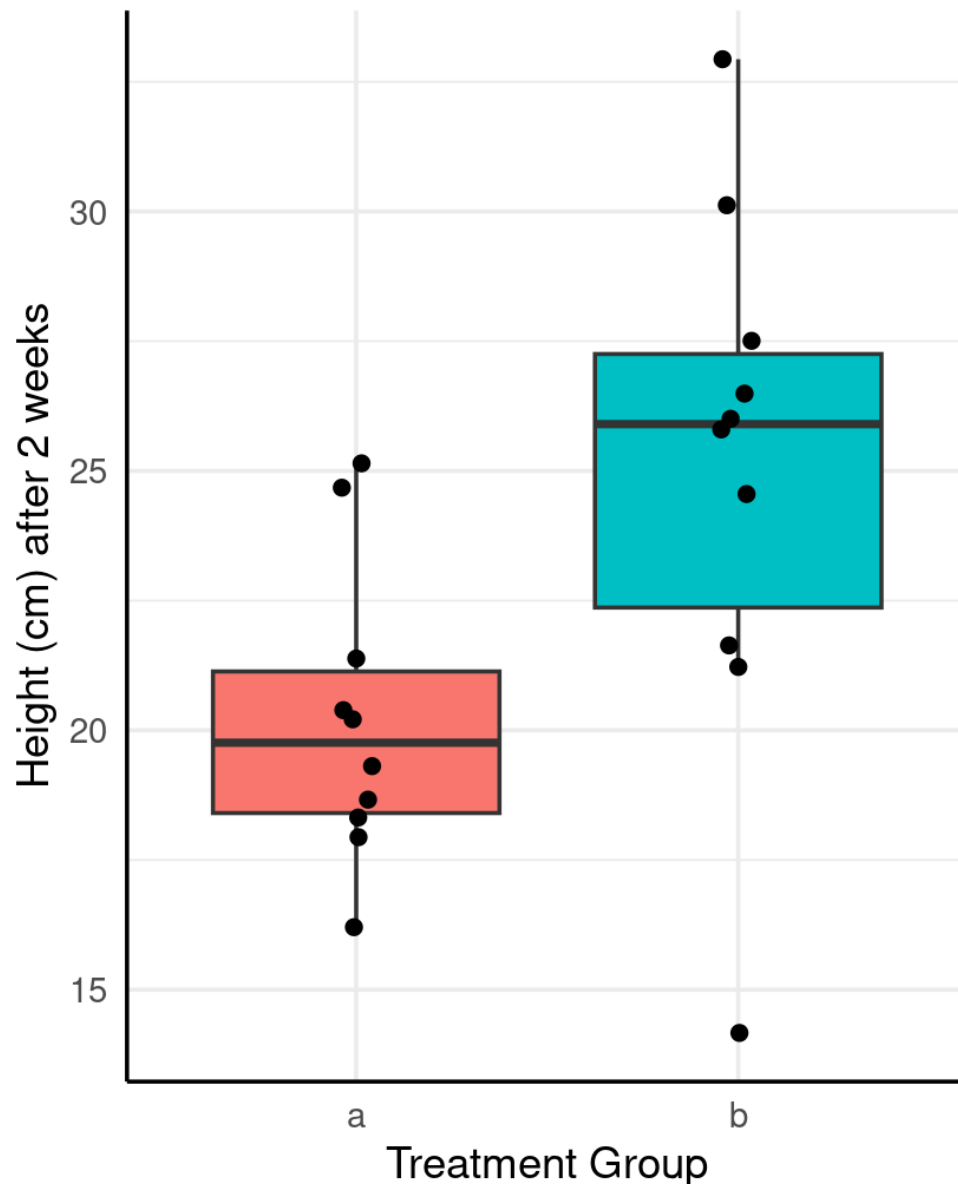




\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better

# Scenario 2

- You are comparing plant heights between two treatment groups (say, control vs cold stress). Given this data how do you do that?



```
> t.test(values ~ group, df)
```

Welch Two Sample t-test

data: values by group

t = -2.5713, df = 14.008, p-value = 0.02217

alternative hypothesis: true difference in m

95 percent confidence interval:

-8.8388370 -0.7996288

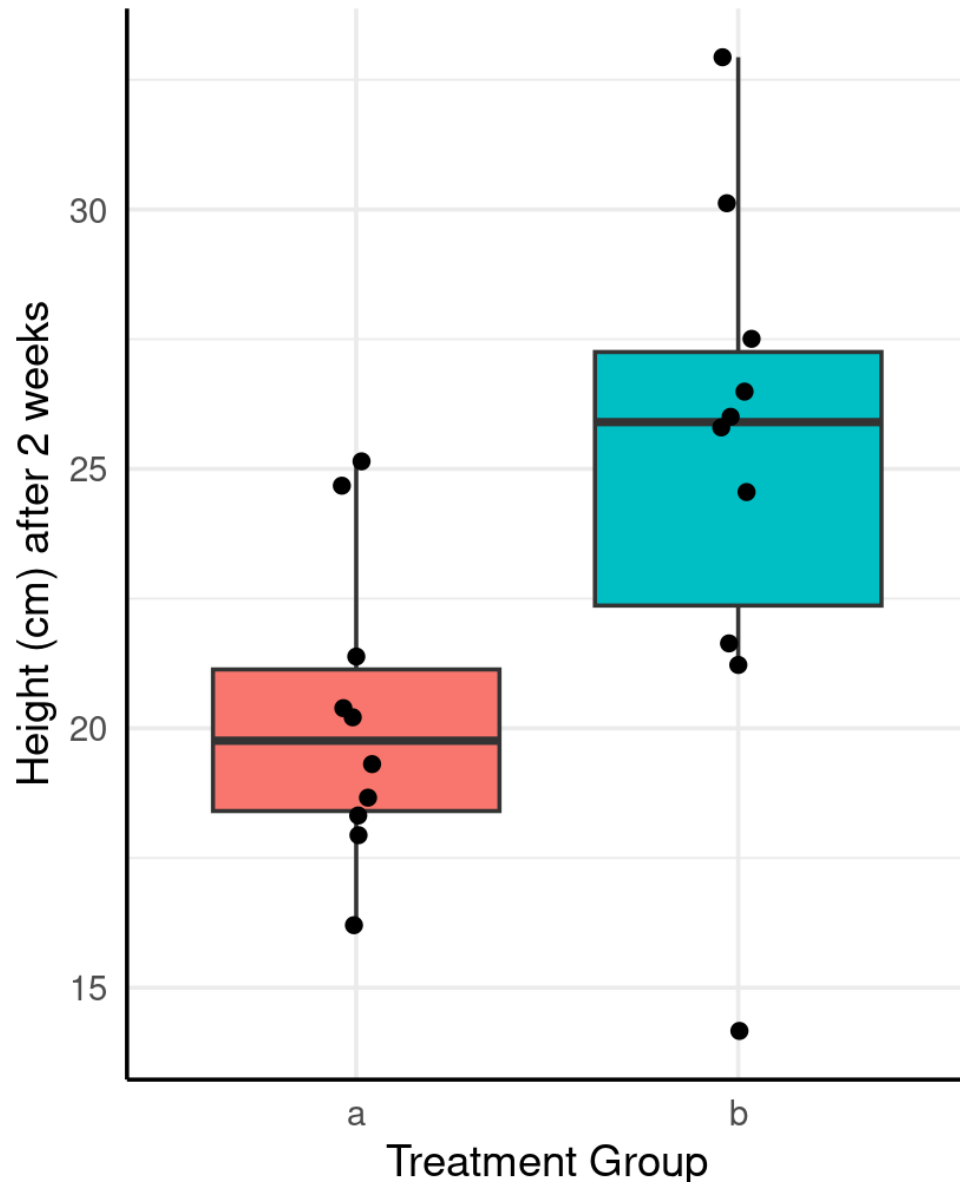
sample estimates:

mean in group a mean in group b

20.22388

25.04311

# Scenario 2 – `pcvr::conjugate`



- There are “t” and “gaussian” distributions we can use.
  - “t” is a comparison of gaussian means.
    - Think T test
  - “gaussian” is a comparison of gaussian distributions.
    - Think Z test

## Scenario 2 – `pcvr::conjugate`

```
res <- pcvr::conjugate(s1, s2, method = "t",  
  priors = list(mu=c(25,25),n=c(1,1),s2=c(20,20) ),  
  rope_range = c(-2, 2),  
  plot = TRUE, hypothesis = "unequal")  
  
res2 <- pcvr::conjugate(s1, s2, method = "gaussian",  
  priors = list(mu=c(25,25),n=c(1,1),s2=c(30,30) ),  
  rope_range = c(-2, 2),  
  plot = TRUE, hypothesis = "unequal")
```

These will yield different results and the choice between them should be made based on your question.

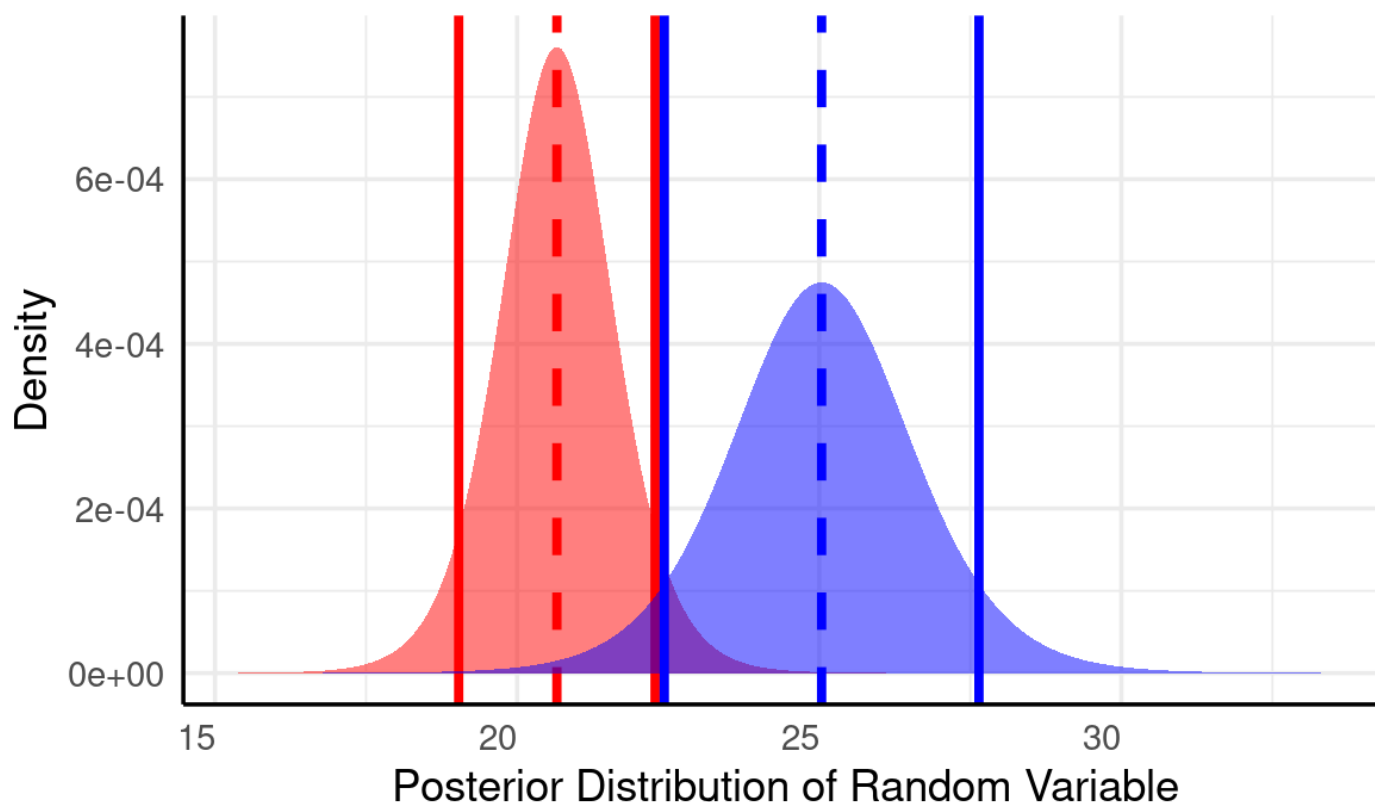
# Scenario 2 – pcvr::conjugate t

## Distribution of Samples

Sample 1: 20.66 [19.03, 22.28]

Sample 2: 25.04 [22.44, 27.64]

$P[p1 \neq p2] = 0.90362$

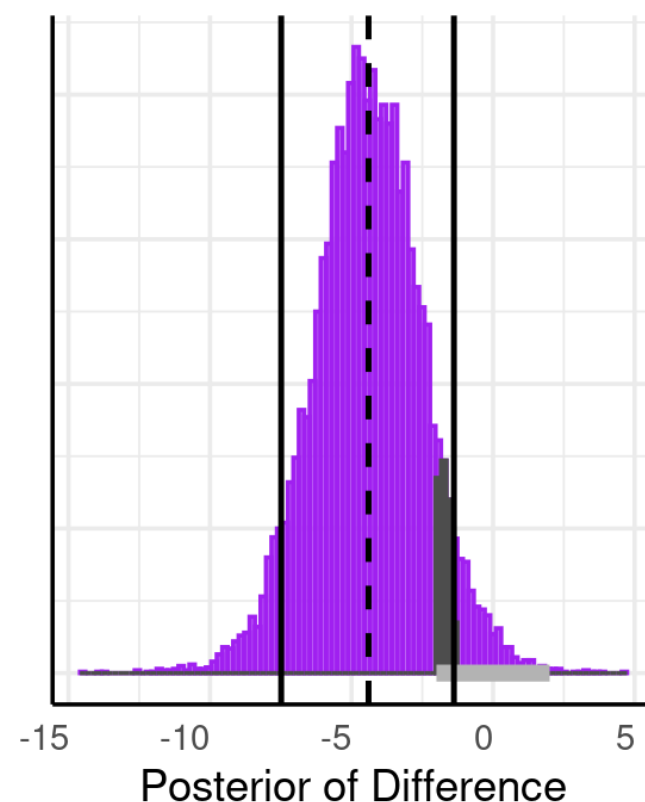


## Distribution of Difference

Median Difference of -4.4

89% CI [-7.49, -1.38]

0.89% HDI in [-2, 2]: 0.05



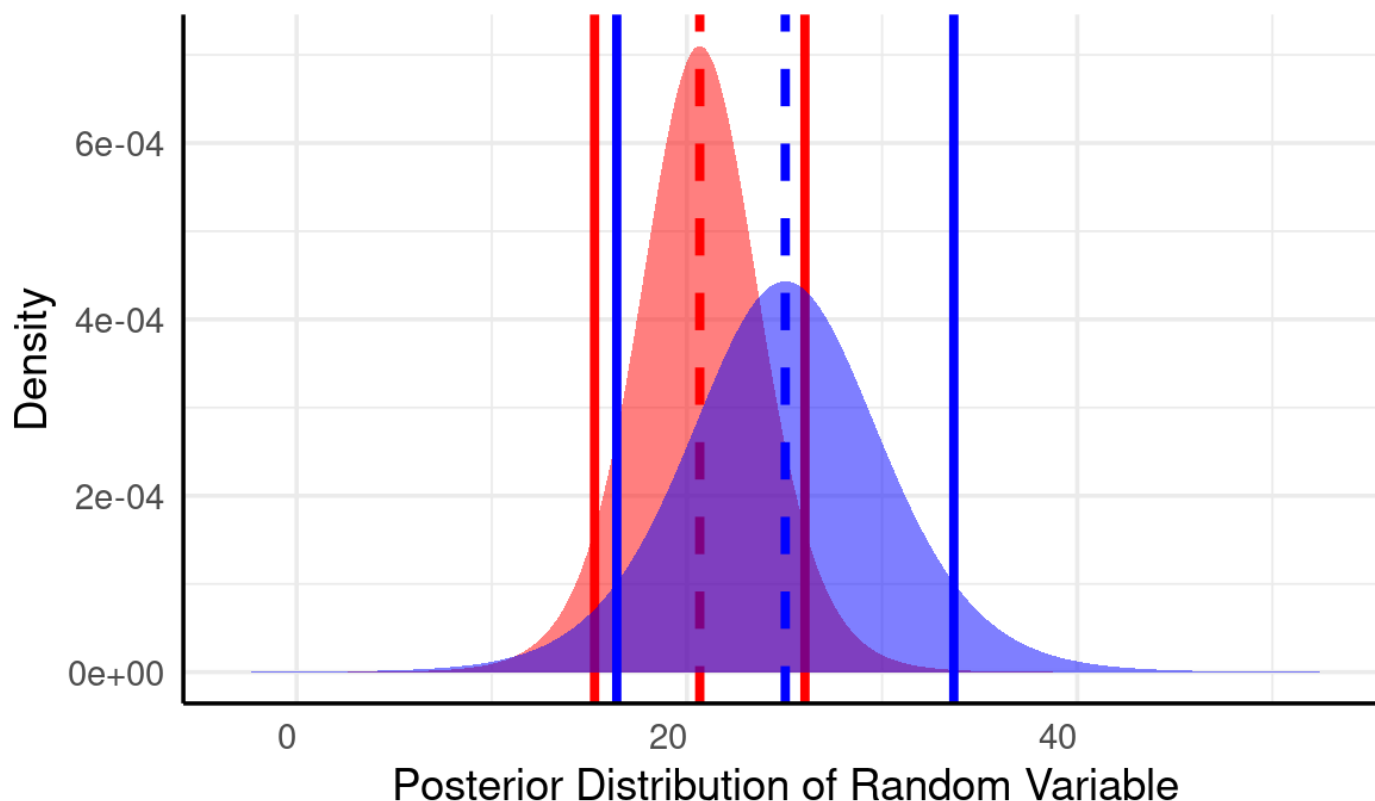
# Scenario 2 – `pcvr::conjugate gaussian`

## Distribution of Samples

Sample 1: 20.66 [15.27, 26.05]

Sample 2: 25.04 [16.4, 33.68]

$P[p1 \neq p2] = 0.43495$

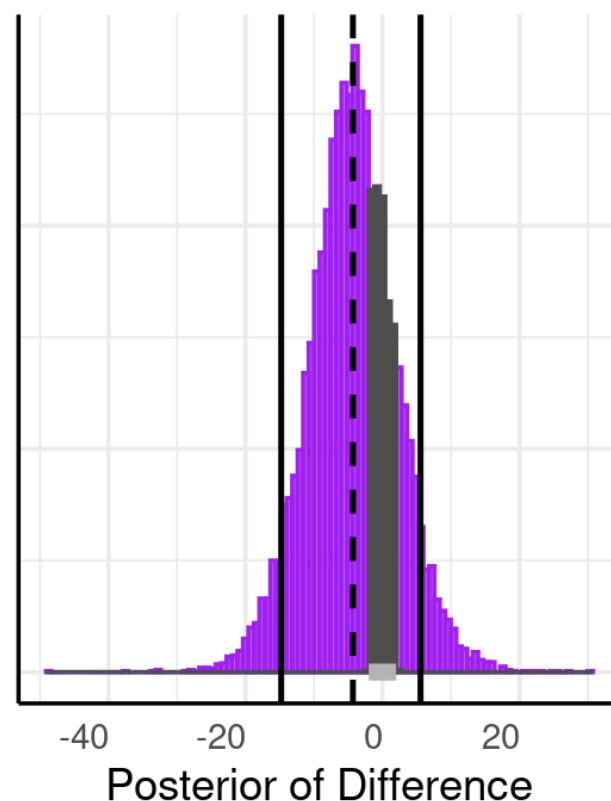


## Distribution of Difference

Median Difference of -4.31

89% CI [-14.79, 5.56]

0.89% HDI in [-2, 2]: 0.22



# Scenario 2

- Our P-value looked “stronger” than the posterior probability. Why would we use the Bayesian option?
  - 1: This is just an example, generally you shouldn't be trying several tests of the same hypothesis and picking based on the output.

# Scenario 2

- Our P-value looked “stronger” than the posterior probability. Why would we use the Bayesian option?
  - 1: This is just an example, generally you shouldn't be trying several tests of the same hypothesis and picking based on the output.
  - 2: The interpretation piece may matter to you.
  - 3: You may have more informative prior information that should be included.



# Scenario 2 - Interpretation

	T Test	Bayesian Test
Probability	If the population means are the same we would see data with this difference or more ~2.2% of the time.	There is a 90.4% chance that the population means are different.
Estimate	$\mu_1 - \mu_2$	$T(\mu_1, \sigma_1, v_1) - T(\mu_2, \sigma_2, v_2)$
Interval	95% Conf. Interval will include the true effect size 95 out of 100 times.	We are 95% sure that the 95% Cred. Interval contains the true effect size.

# Scenario 2 – Prior Information

- Priors are an often criticized part of Bayesian statistics.

# Scenario 2 – Prior Information

- Priors are an often criticized part of Bayesian statistics.
- Priors parameterize information that is not in your data but which is relevant to your decision-making/analysis.
  - Include: Previous research, Biological boundaries, etc.
  - Exclude: Suspicions, Goals, Hunches, etc.

# Scenario 2 – Prior Information

- Priors are an often criticized part of Bayesian statistics.
- Priors parameterize information that is not in your data but which is relevant to your decision-making/analysis.
  - Include: Previous research, Biological boundaries, etc.
  - Exclude: Suspicions, Goals, Hunches, etc.
- Broadly, priors come as weak, strong, and flat.

# Scenario 2 – Prior Information

- Priors are an often criticized part of Bayesian statistics.
- Priors parameterize information that is not in your data but which is relevant to your decision-making/analysis.
  - Include: Previous research, Biological boundaries, etc.
  - Exclude: Suspicions, Goals, Hunches, etc.
- Broadly, priors come as strong, weak, and flat.

# Strong Priors

- **Negative:** This guy has OPINIONS and they are not going to change based on your paltry “evidence”
- **Positive:** This guy is far from gullible, he will not exaggerate and aggrandize bad information.

# Mild (weak) Priors

- **Negative:** This guy is not a domain expert, he is not very sure about what to expect. If you don't have much (data) to contribute then your conclusions will be limited.
- **Positive:** This guy knows he is not a domain expert, he is able to contribute to a conversation and collaborate without talking over the evidence you present.

# Flat Priors

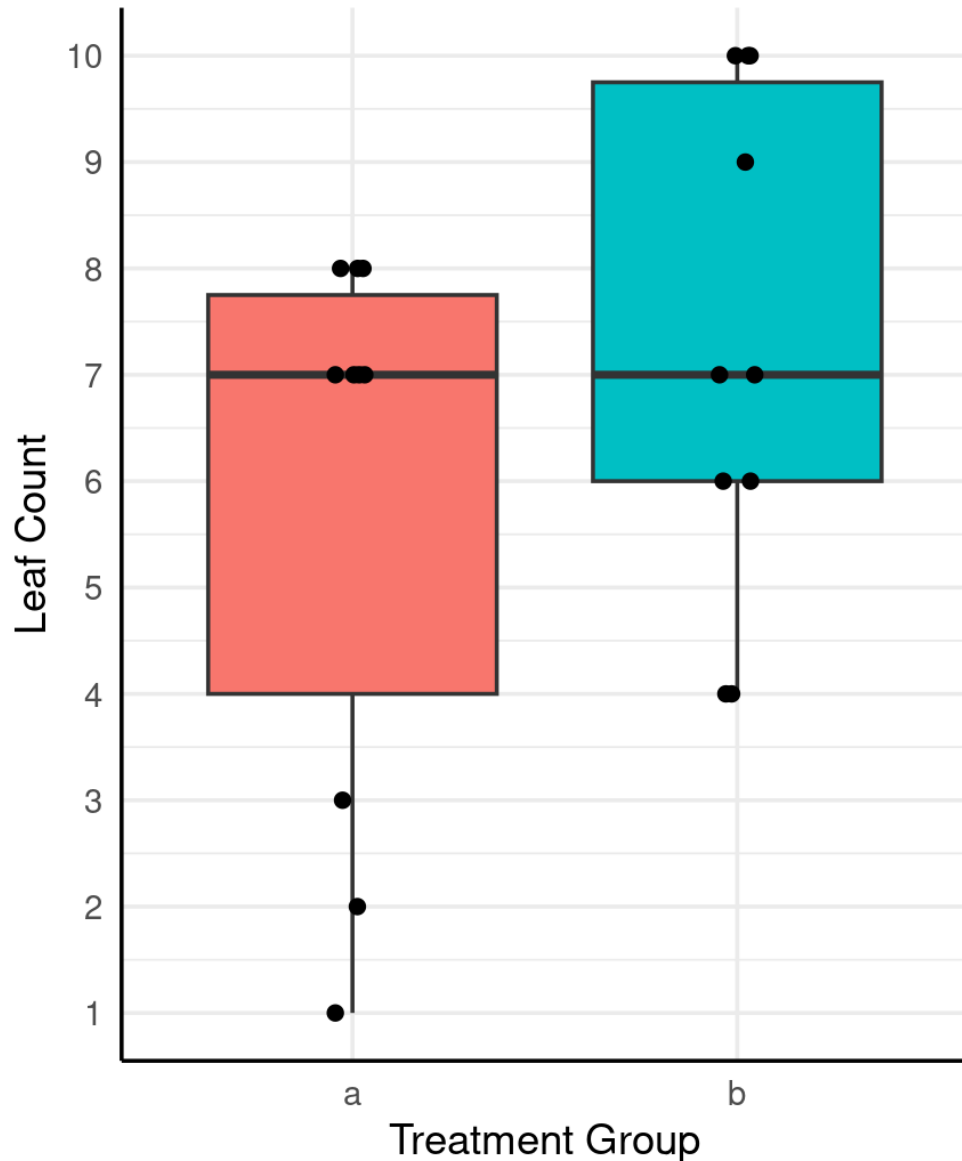
- **Negative:** This guy doesn't understand the world at all. He contributes nothing to the conversation and was only invited to round out the numbers.
- **Positive:** “Unbiased” in the eyes of many people, but those people are confusing “unbiased” with “ignorant”.



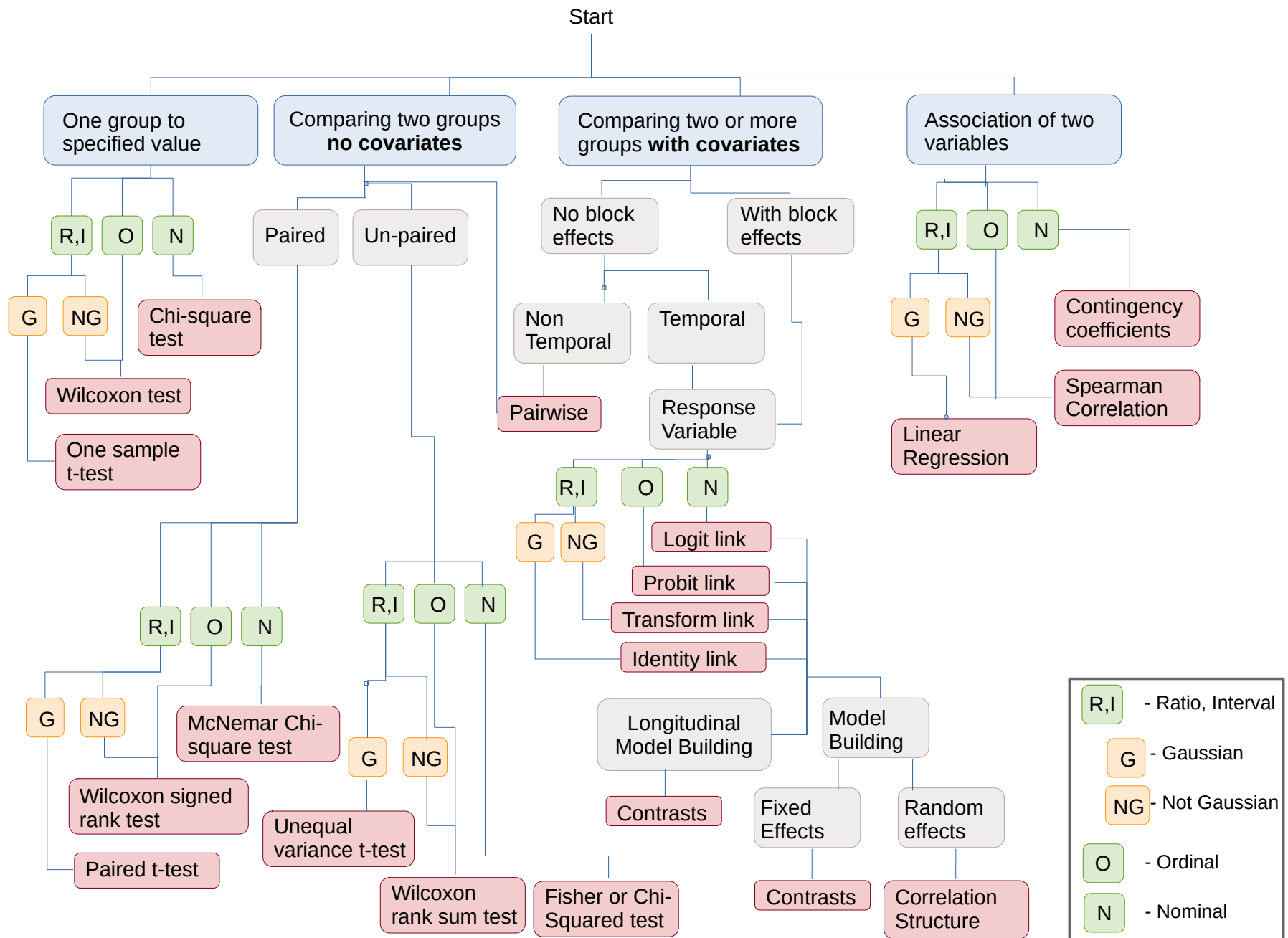
# Scenario 2 – Takeaway Points

- Bayesian testing provides an alternative to frequentist parametric testing as well, with benefits in interpretation.
- The ability to include prior information should be viewed as an opportunity to be more accurate rather than a concession that your results must be unfairly biased.

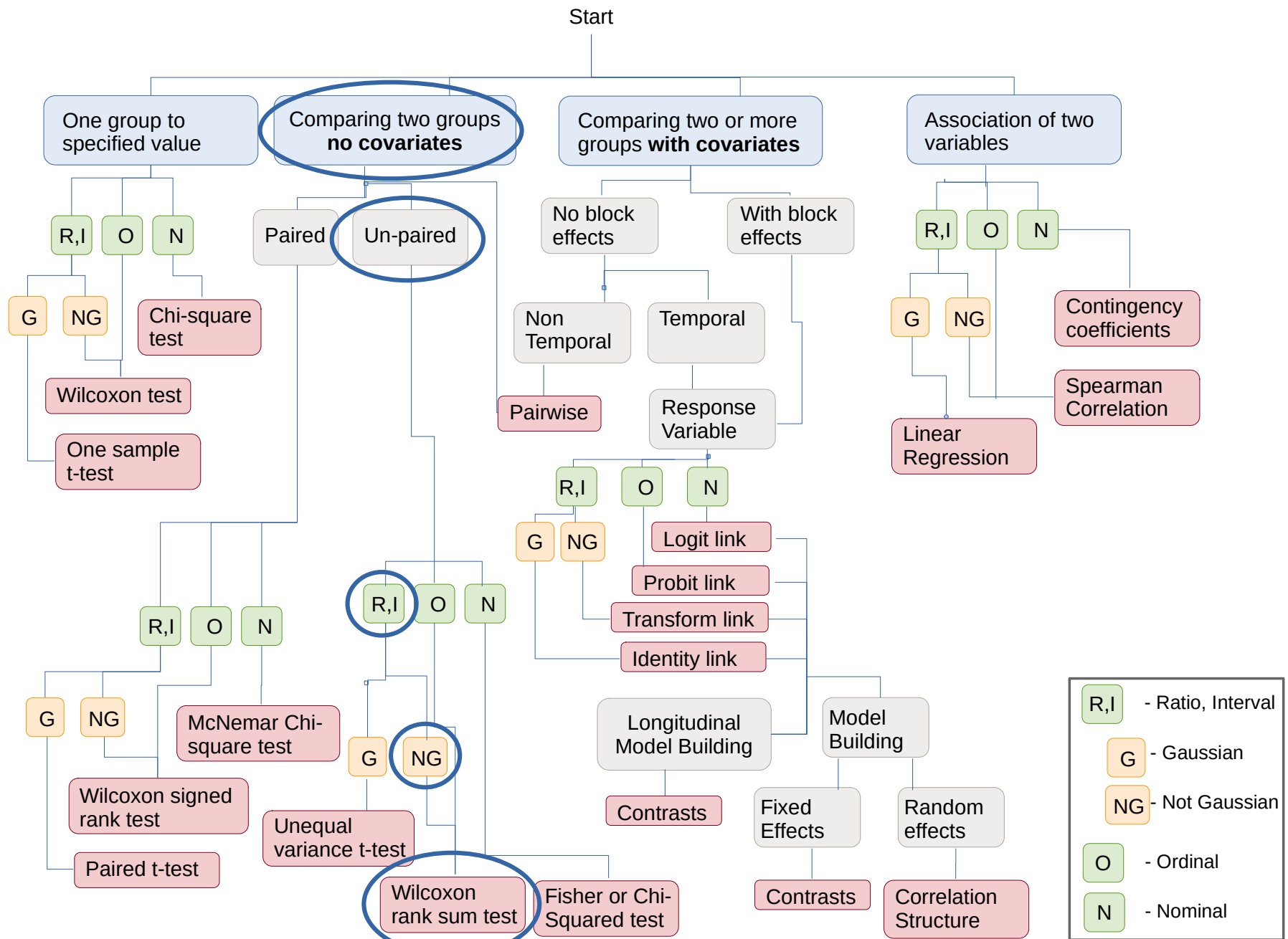
# Scenario 3



- You are interested in the difference in number of leaves between two genotypes. How do you analyze the data?

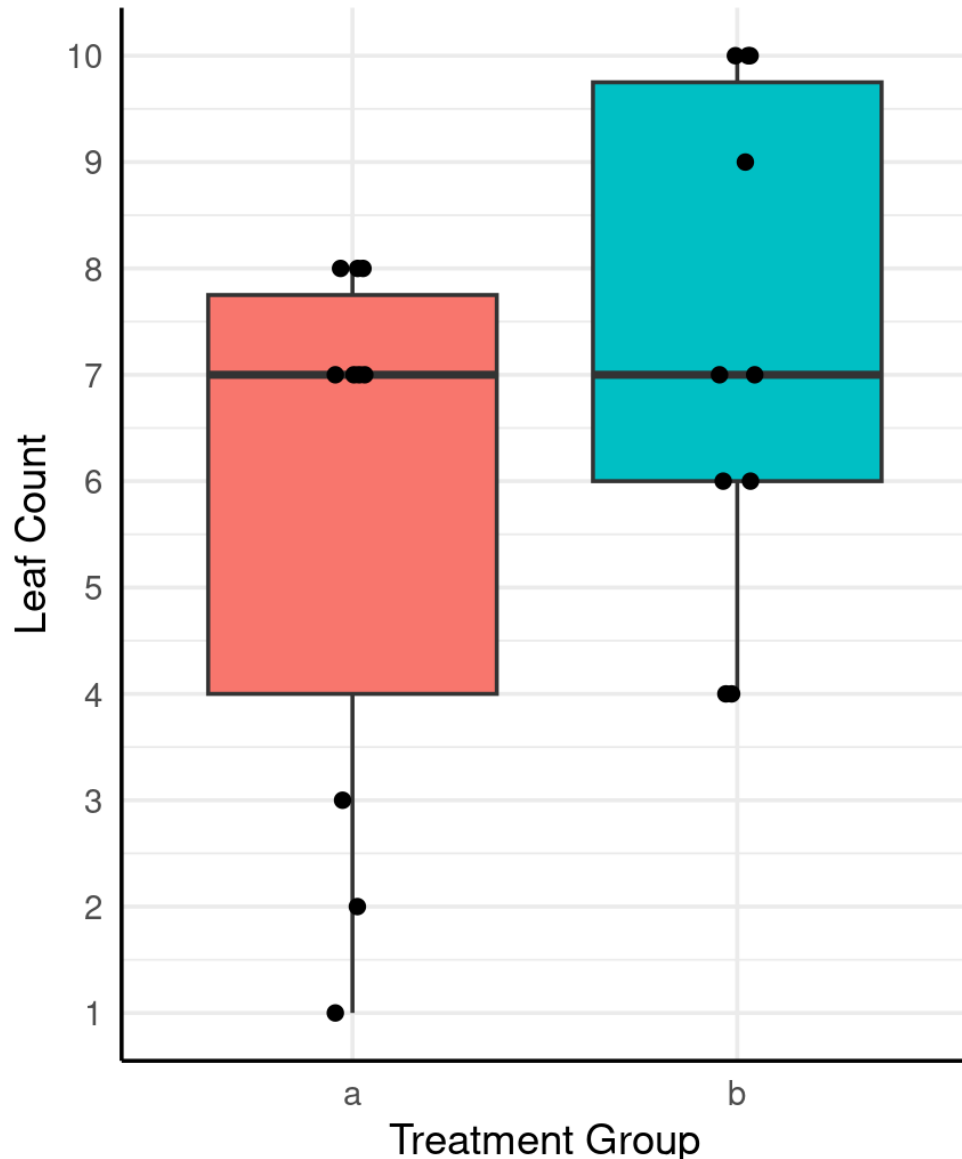


\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better



\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better

# Scenario 3



- Wilcox is a good option here, but again we have ties and again we will not get an estimate of the effect size.

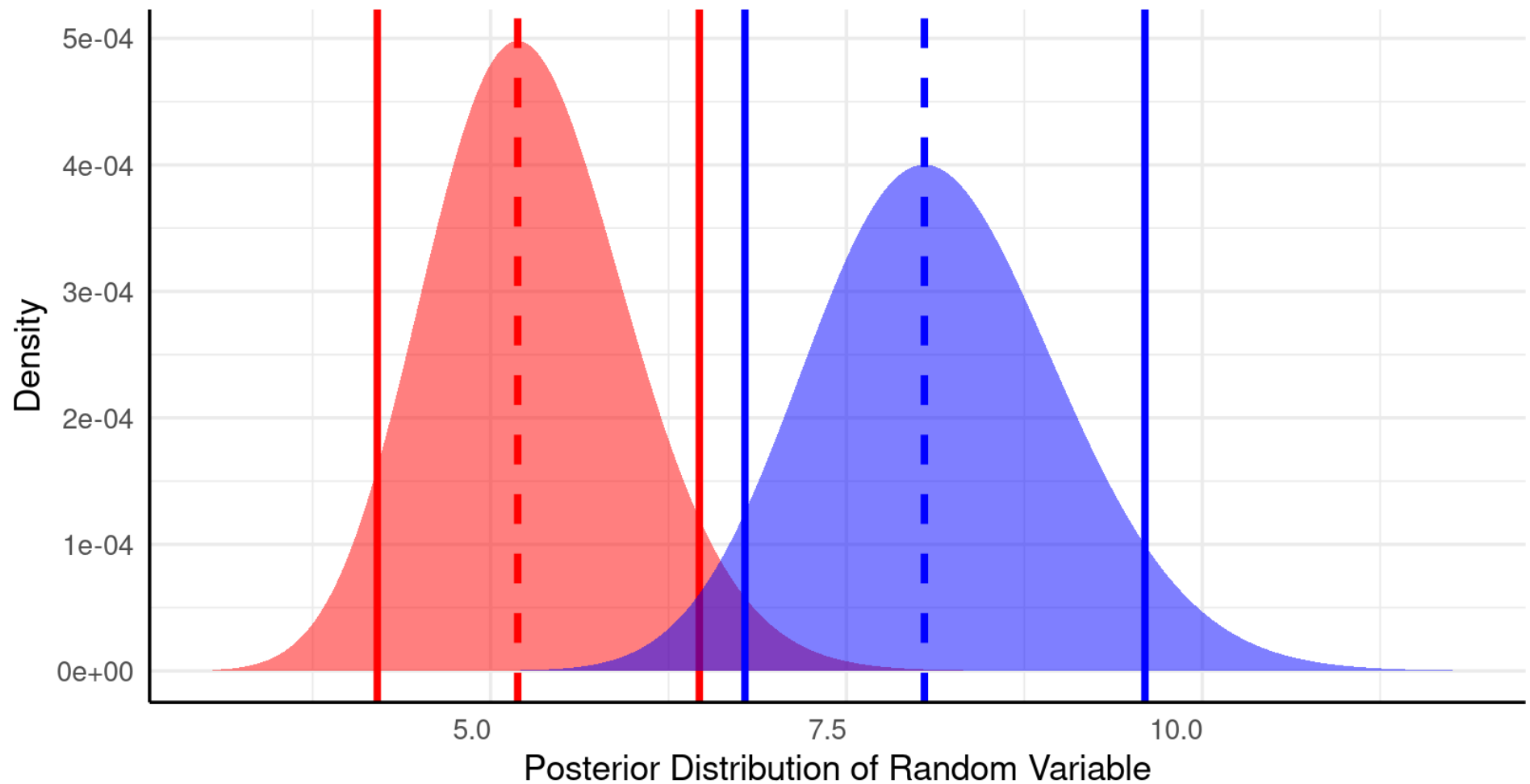
# Scenario 3

## Distribution of Samples

Sample 1: 5.19 [4.2, 6.47]

Sample 2: 8.05 [6.79, 9.6]

$P[p1 \neq p2] = 0.92812$



# Scenario 3

- Why the continuous curves? That was count data?

# Scenario 3

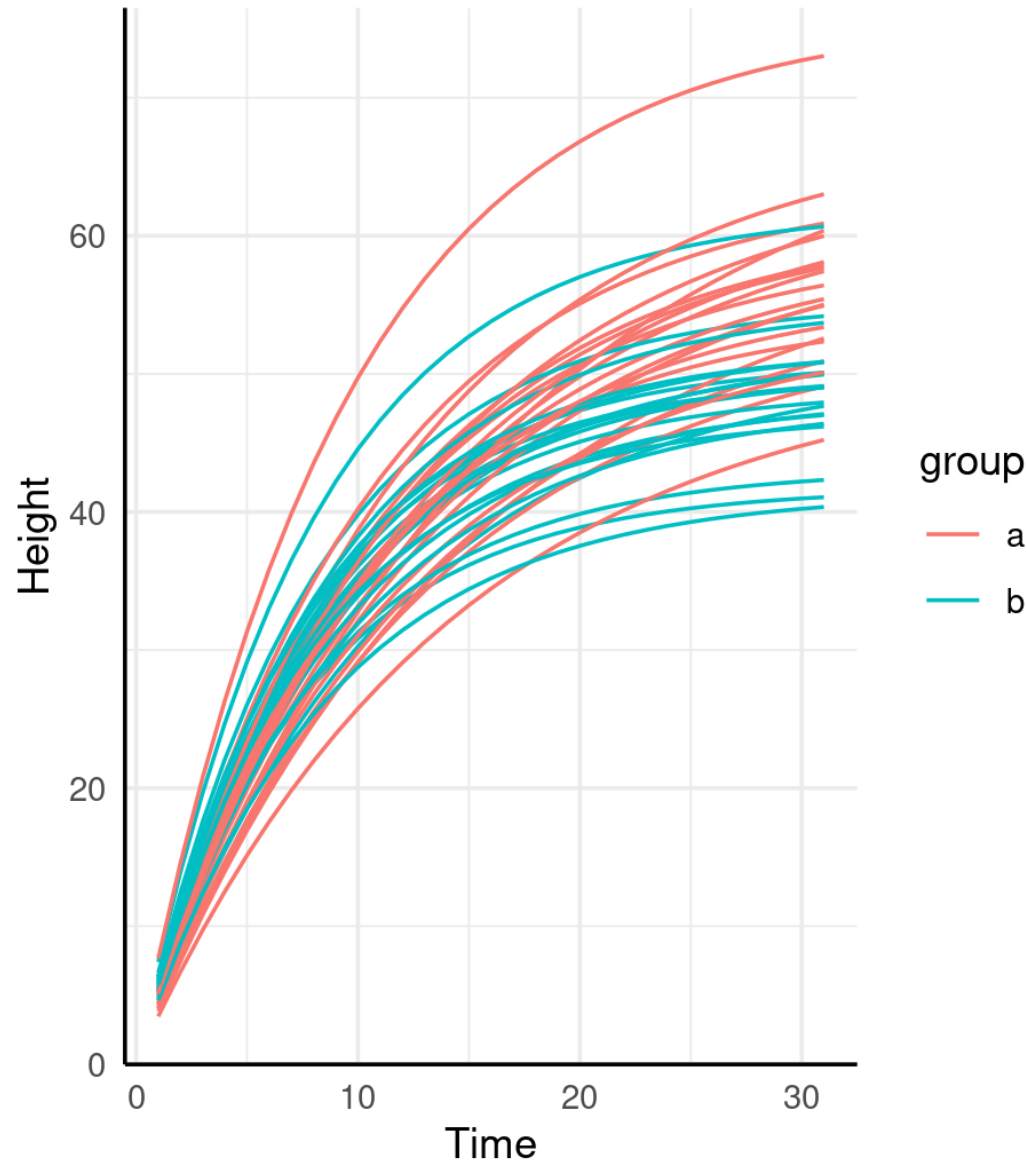
- Why the continuous curves? That was count data?
  - In the previous examples we estimated the mean of a Gaussian or the Gaussian itself, so the parameter space and posterior distribution's support were the same.
  - Here we are estimating  $\lambda$  from  $\text{Poisson}(\lambda)$ , which is conjugate to the Gamma distribution (continuous positive).



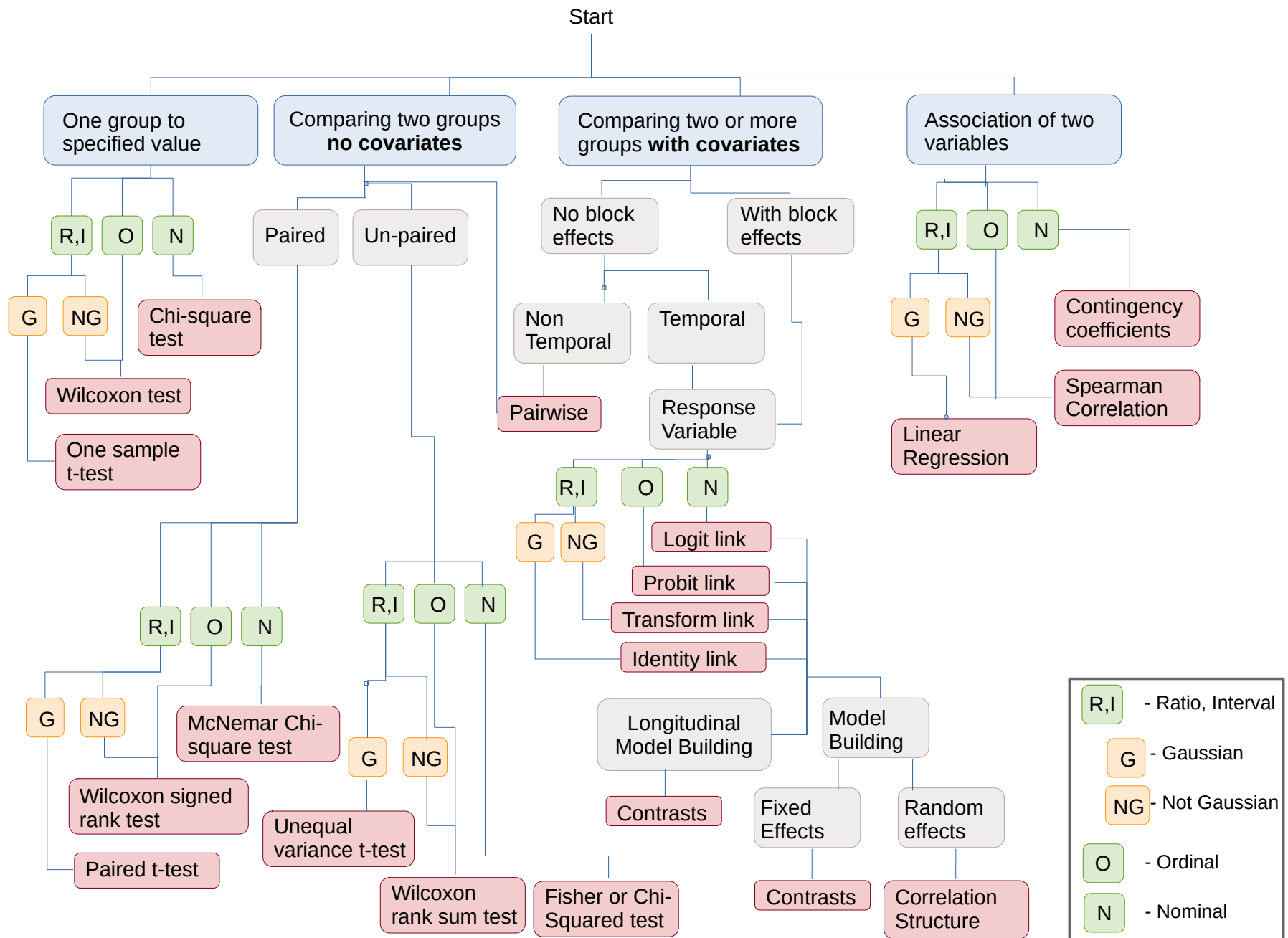
# Scenario 3 - Takeaway

- We are working with several distributions so **conjugate** is a little less plug-and-play than T tests and Wilcoxon tests.
- Some conjugate priors are in the same space as the observed data, some like this one have one degree of abstraction.

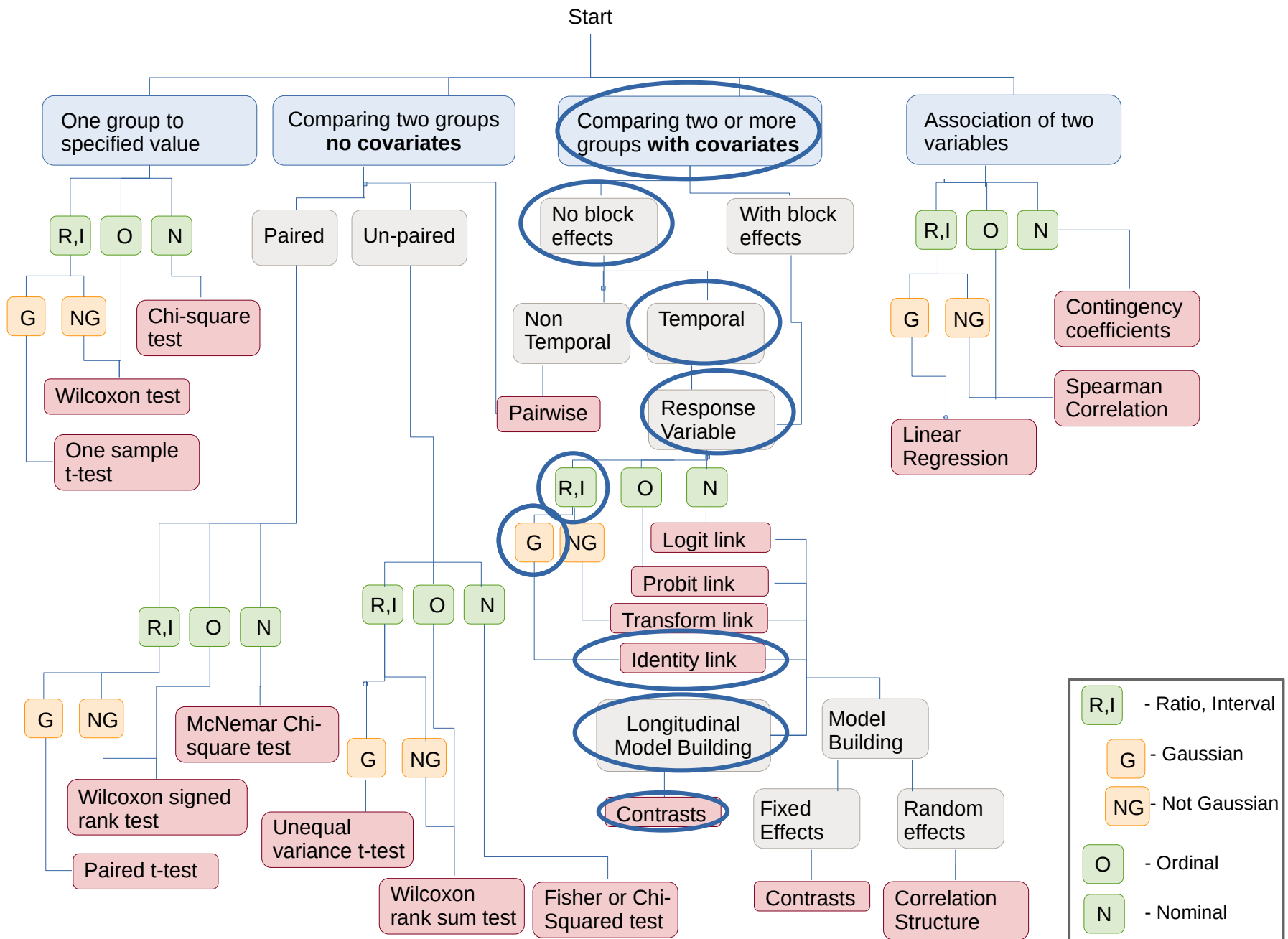
# Scenario 4



- You image plants in a growth chamber for a month and want to compare the growth rate between two soil treatments. How do you analyze your data?

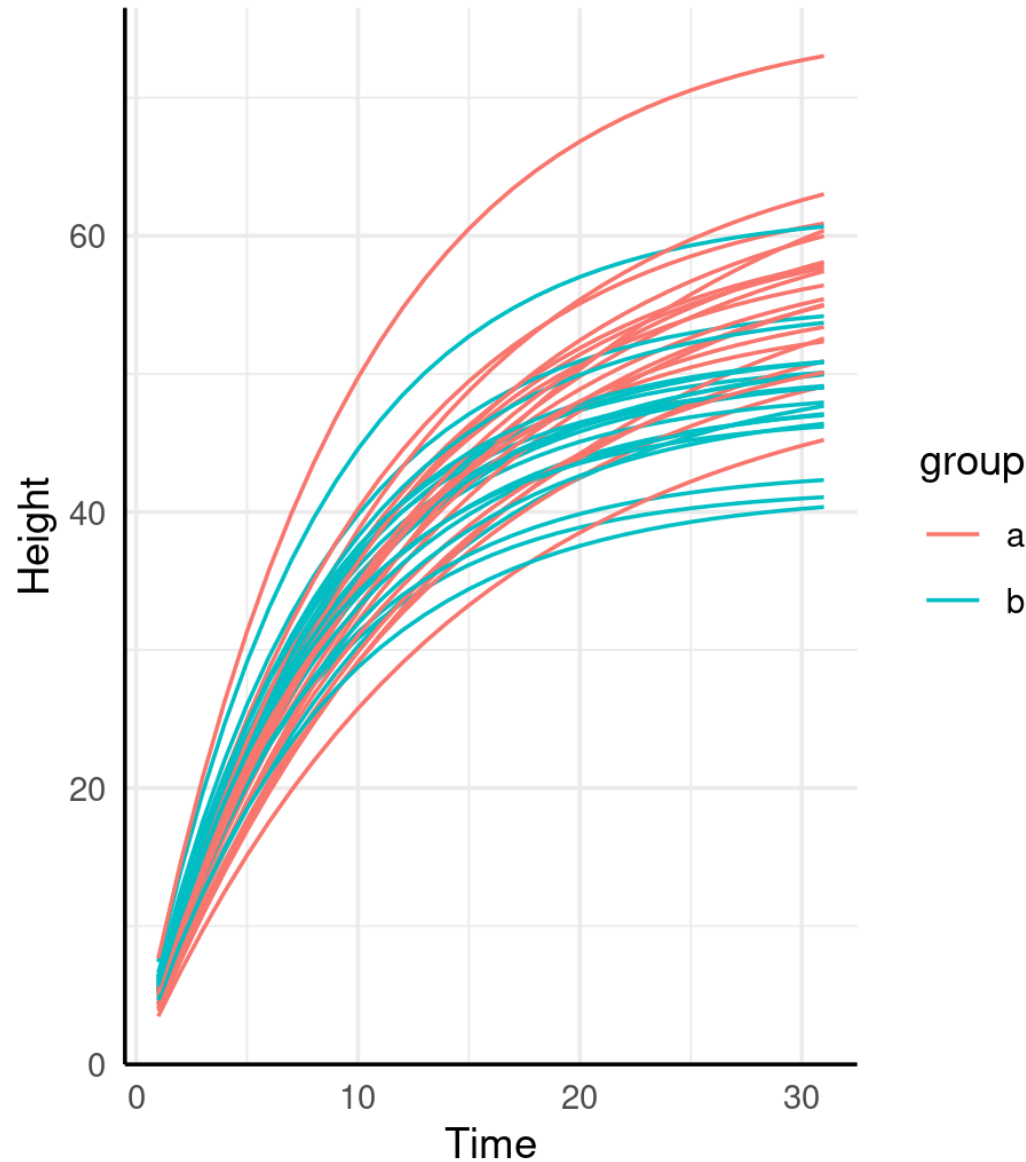


\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better



\* If the outcome and main predictor are both two-level factors, Breslow-Day and Cochren-Mantel-Hanzel tests are better

# Scenario 4



- We're going to have to make a model, we'll try two options before showing how this works in [pcvr](#).

# Scenario 4 - nls

```
> m1 <- nls(y ~ A - A * exp(-B * time),  
+          data = simdf, start = list("A" = 20, "B" = 0.1 ))  
> m1
```

Nonlinear regression model

model:  $y \sim A - A * \exp(-B * \text{time})$

data: simdf

A	B
---	---

54.6522	0.1016
---------	--------

We specify a monomolecular  
Growth formula, potentially  
A point where we'd get hung up.

residual sum-of-squares: 29658

Number of iterations to convergence: 3

Achieved convergence tolerance: 6.1e-07

# Scenario 4 - nls

```
> m1 <- nls(y ~ A-A * exp(-B * time),  
+          data = simdf, start = list("A" = 20, "B" = 0.1 ))  
> m1
```

Nonlinear regression model

model:  $y \sim A - A * \exp(-B * \text{time})$

data: simdf

A	B
---	---

54.6522	0.1016
---------	--------

We specify starting values.  
here without modeling groups  
this is not terribly difficult.

residual sum-of-squares: 29658

Number of iterations to convergence: 3

Achieved convergence tolerance: 6.1e-07

# Scenario 4 - nls

```
> nls(y ~ A[group]-A[group] * exp(-B[group] * time),  
+     data = simdf, start = list("A" = 55, "B" = 0.1 ))  
Error in numericDeriv(form[[3L]], names(ind), env, central = nDcentral) :  
  Missing value or an infinity produced when evaluating the model
```

Modeling our groups makes  
this much more difficult to  
Initialize.



# Scenario 4 - lm

```
> lm(y ~ log(time)*group, simdf)
```

Call:

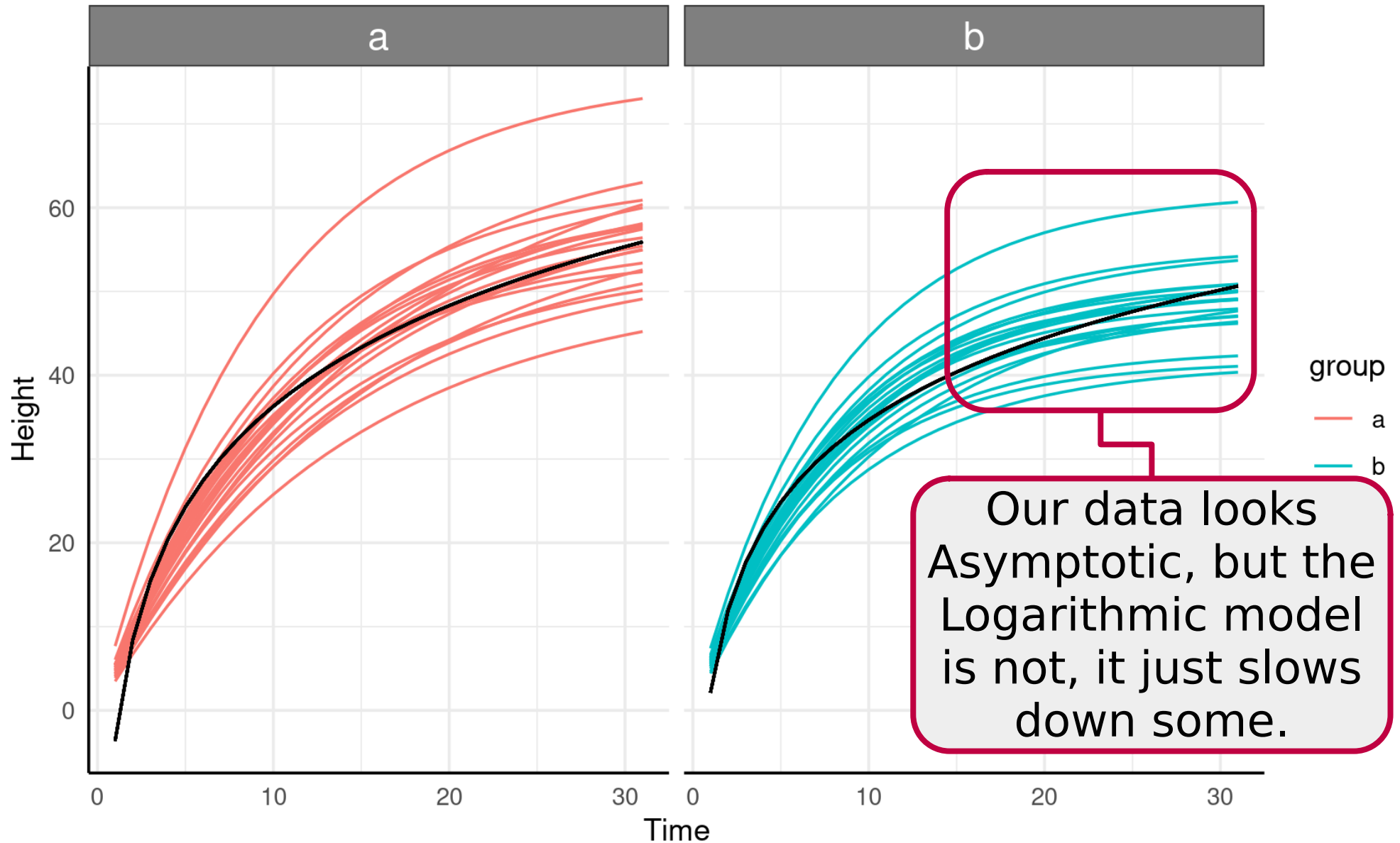
```
lm(formula = y ~ log(time) * group, data = simdf)
```

Coefficients:

(Intercept)	log(time)	groupb	log(time):groupb
-3.642	17.343	5.754	-3.211

Here we use a linear logarithmic Growth model. This is easy, why not use this option?

# Scenario 4 - Im



## Scenario 4 - pcvr

```
> ss <- growthSS("monomolecular",  
+               y ~ time|id/group,  
+               df = simdf, type = "nls")
```

Individual is not used with type = 'nls'.

```
> fit <- fitGrowth(ss)  
> fit
```

Nonlinear regression model

model:  $y \sim A[\text{group}] - A[\text{group}] * \exp(-B[\text{group}] * \text{time})$

data: ss[["df"]]

A1	A2	B1	B2
----	----	----	----

60.51876	49.77994	0.08475	0.12225
----------	----------	---------	---------

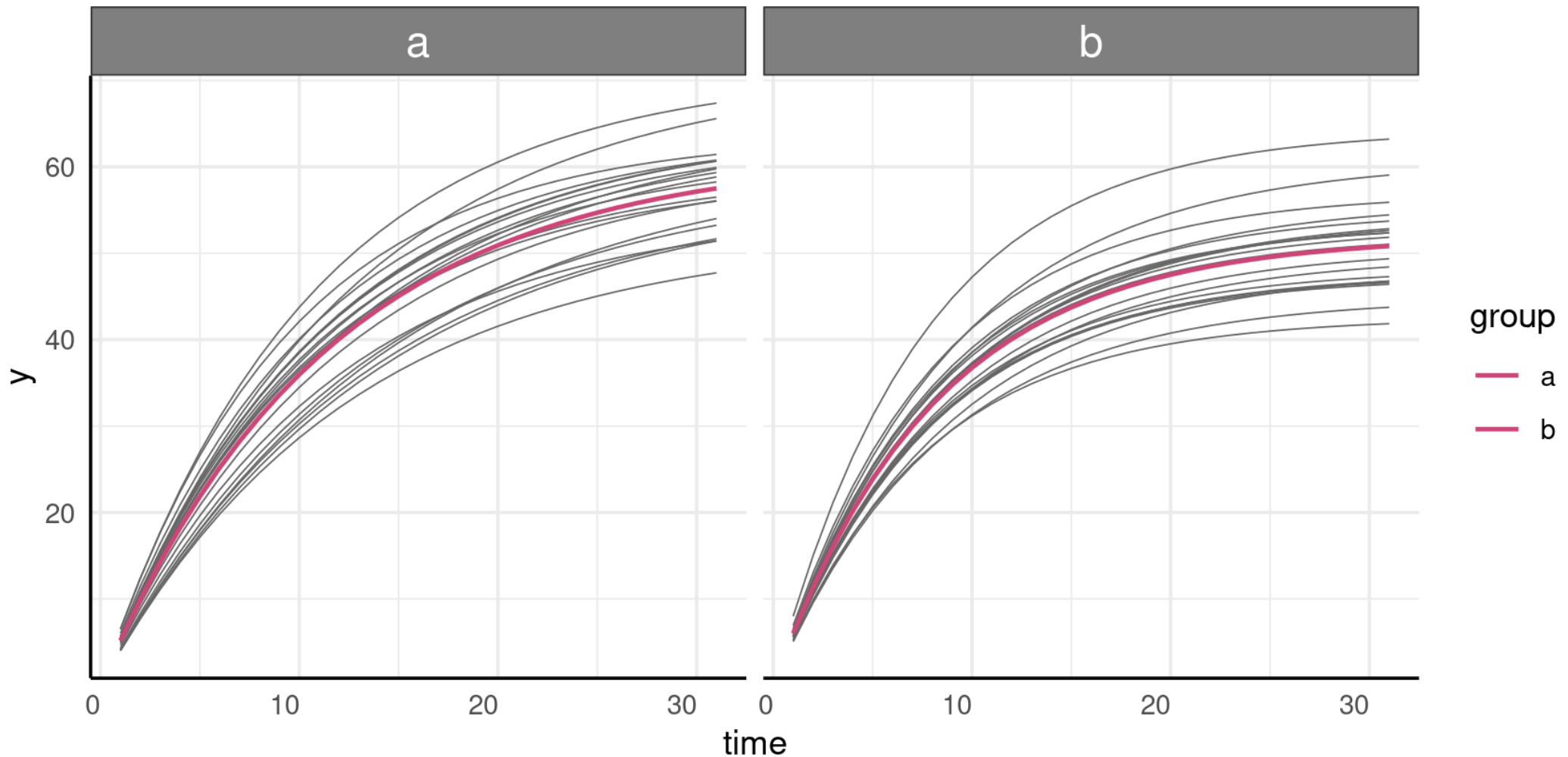
residual sum-of-squares: 24777

Number of iterations to convergence: 6

Achieved convergence tolerance: 5.759e-09

# Scenario 4 - pcvr

```
growthPlot(fit, form = ss$pcvrForm, df = ss$df)
```



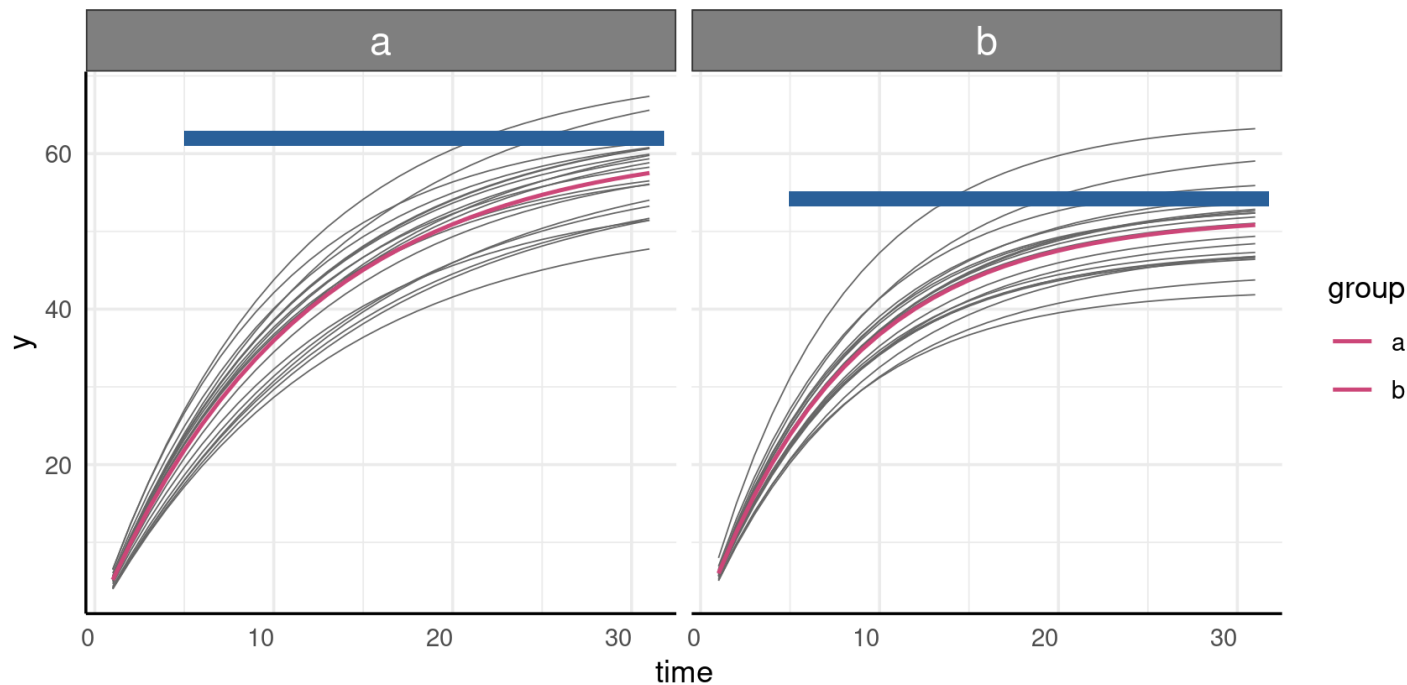
# Scenario 4 - pcvr

```
> coef(fit)
```

	A1	A2	B1	B2
	61.56988178	51.98329496	0.08773047	0.12275720

```
> testGrowth(ss, fit, test = "A1 - A2*1.1")
```

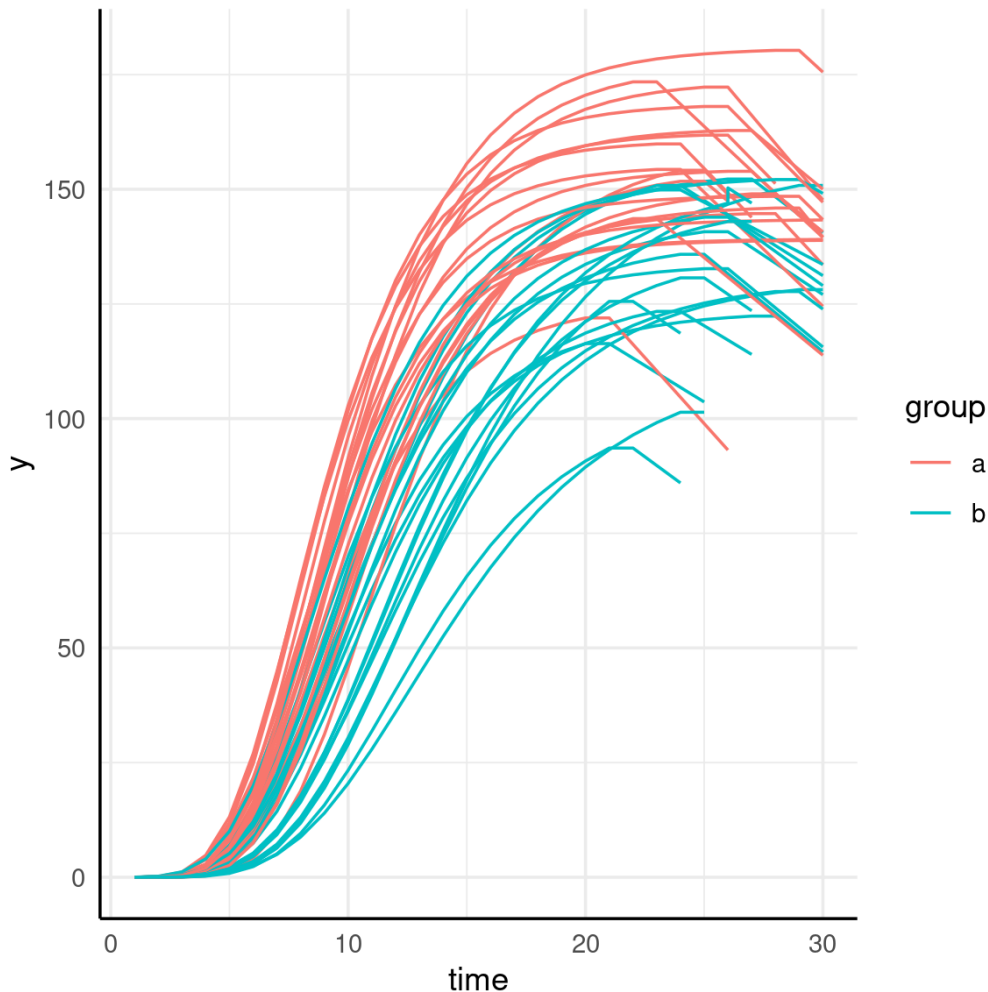
	Form	Estimate	SE	t-value	p-value
1	A1 - A2*1.1	4.388257	0.7512528	5.841252	6.61723e-09



# Scenario 4 - Takeaway

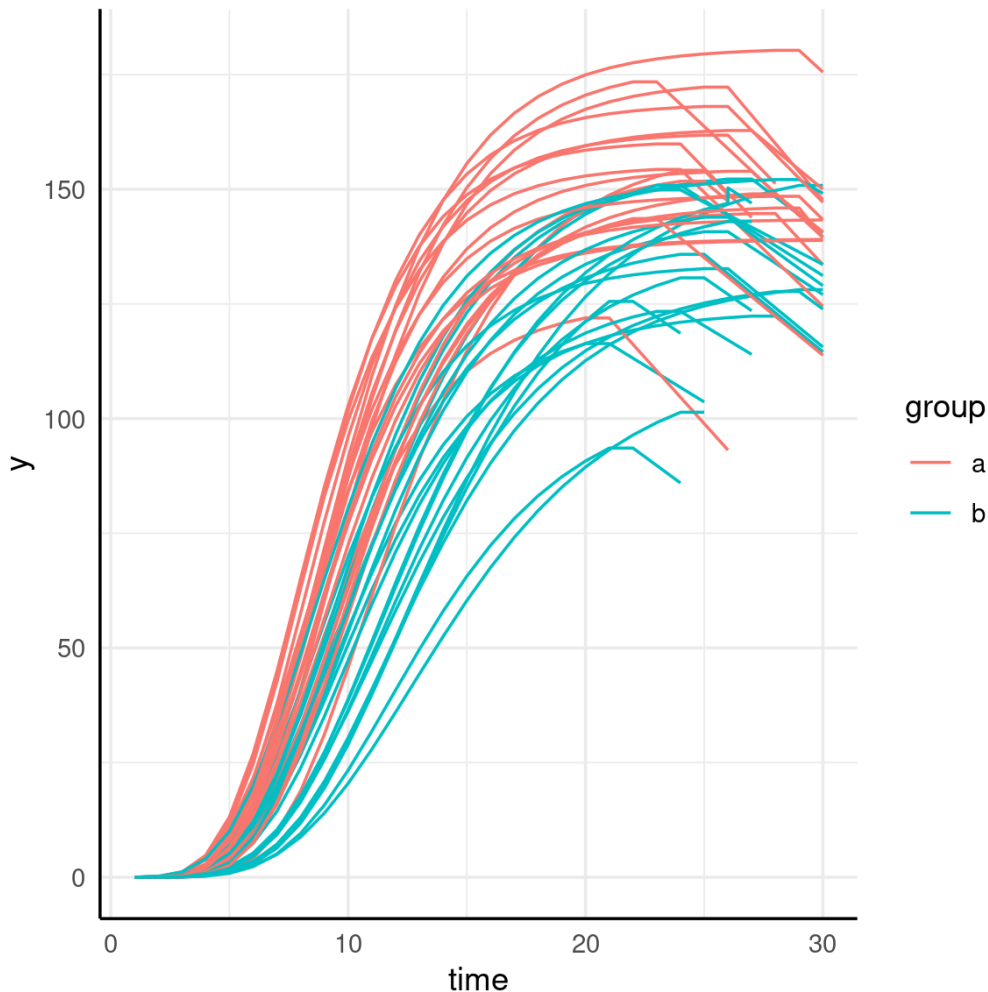
- Even simple non-linear models are easier to use if they are among those included in `growthSS`.
- There is support for visualization and testing of those models through `growthPlot` and `testGrowth`.

# Scenario 5



- You have collected a month of growth data across 20 plants in each of 2 groups. You want to model the growth rate and final size. How do you start?

# Scenario 5

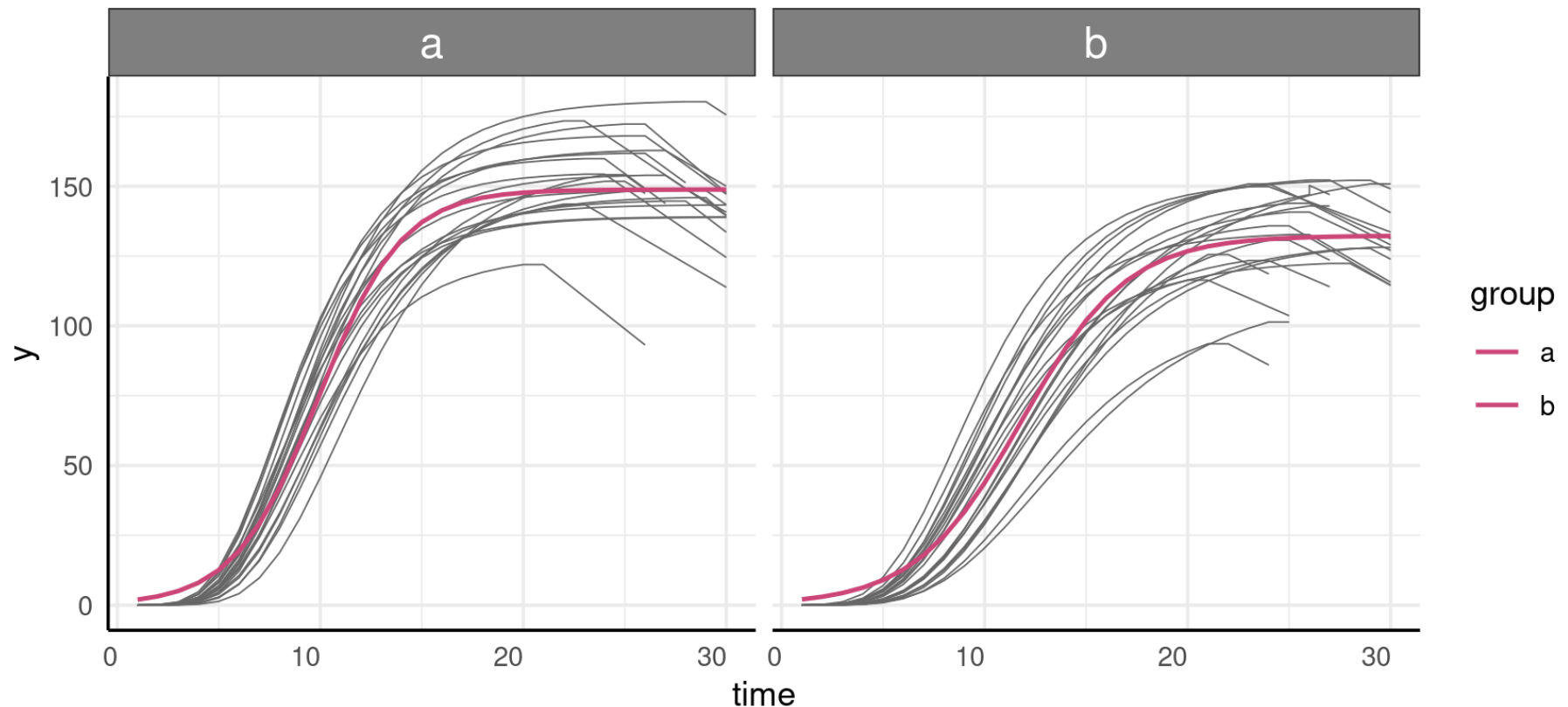


- Let's start with the simplest model that makes sense, maybe a logistic growth model.



# Scenario 5

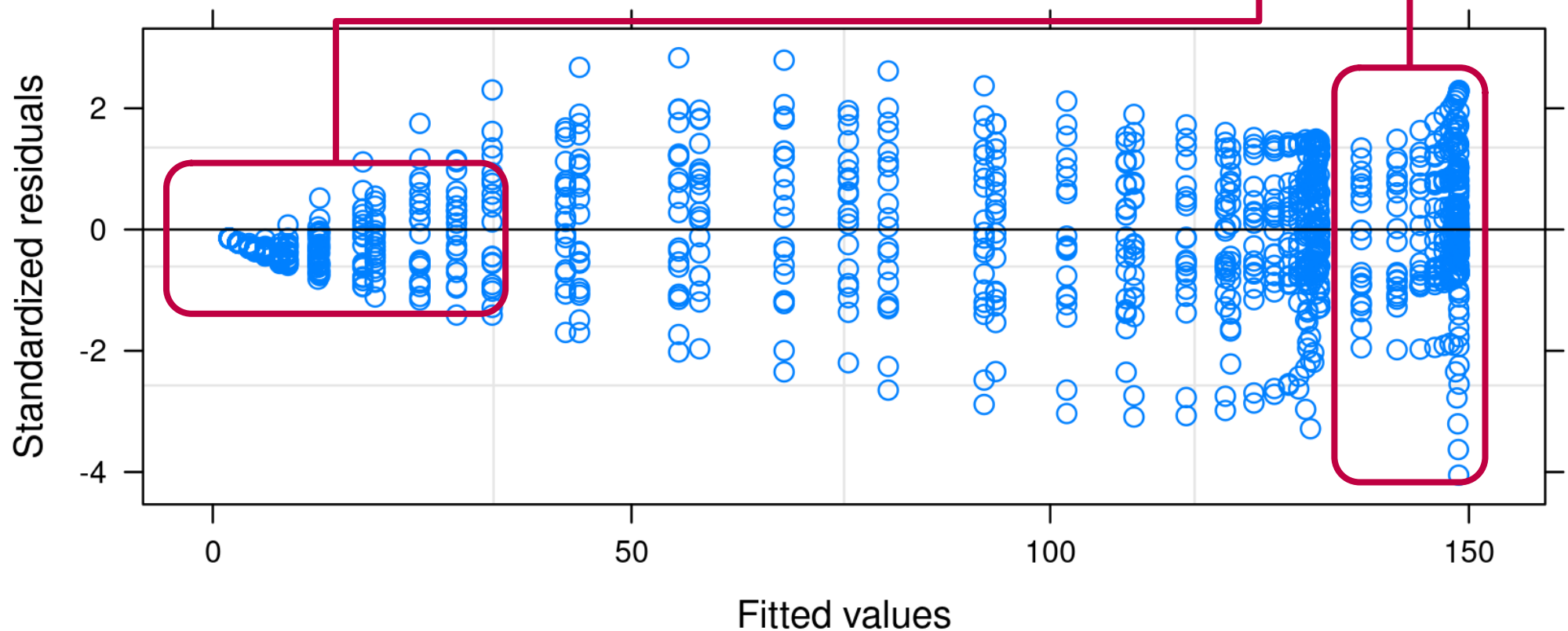
```
ss <- growthSS("logistic", form = y ~ time|id/group,  
               df = simdf, type = "nls")  
fit <- fitGrowth(ss)  
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```



# Scenario 5

```
> plot(fit)
```

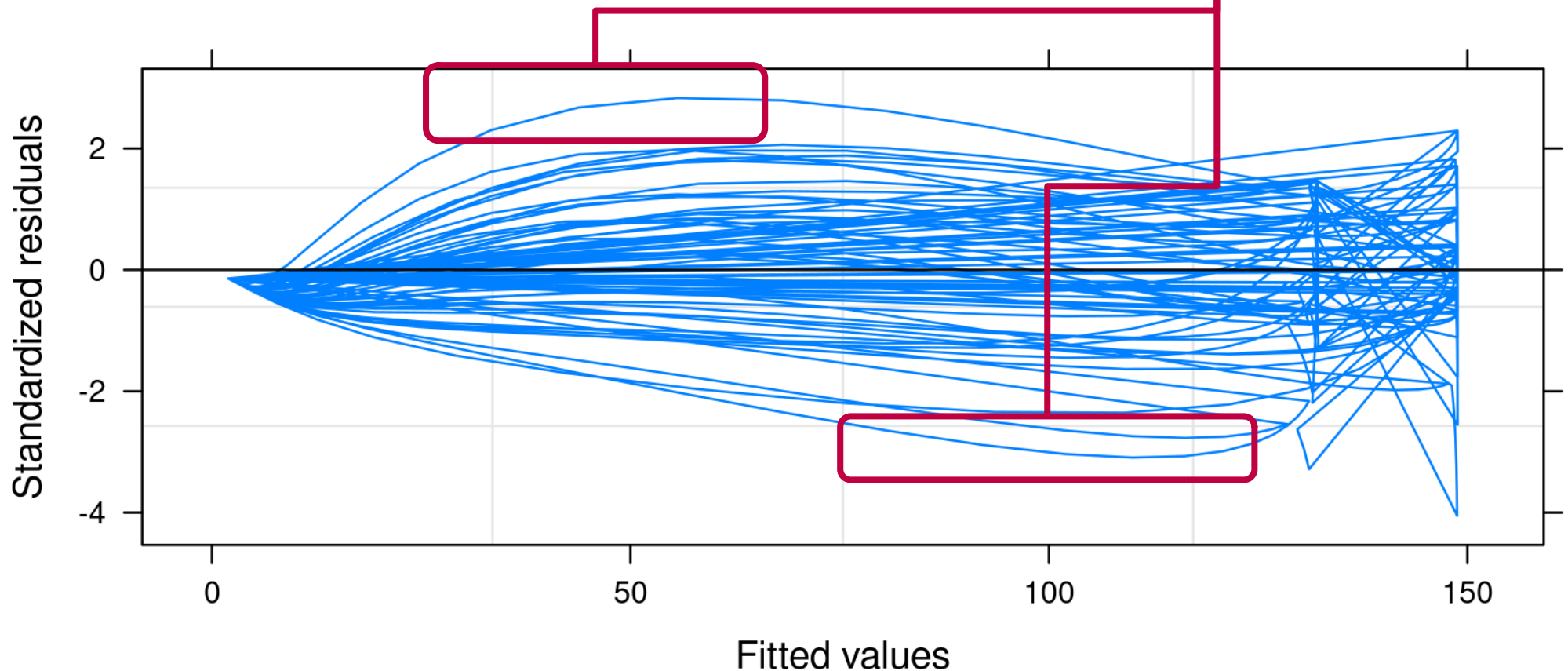
We have a clear  
Pattern in the residuals



# Scenario 5

```
> plot(fit)
```

We also have strong autocorrelation that is not taken into account yet.



# Scenario 5

- Okay so we have some shortcomings, we are going to want a better model.

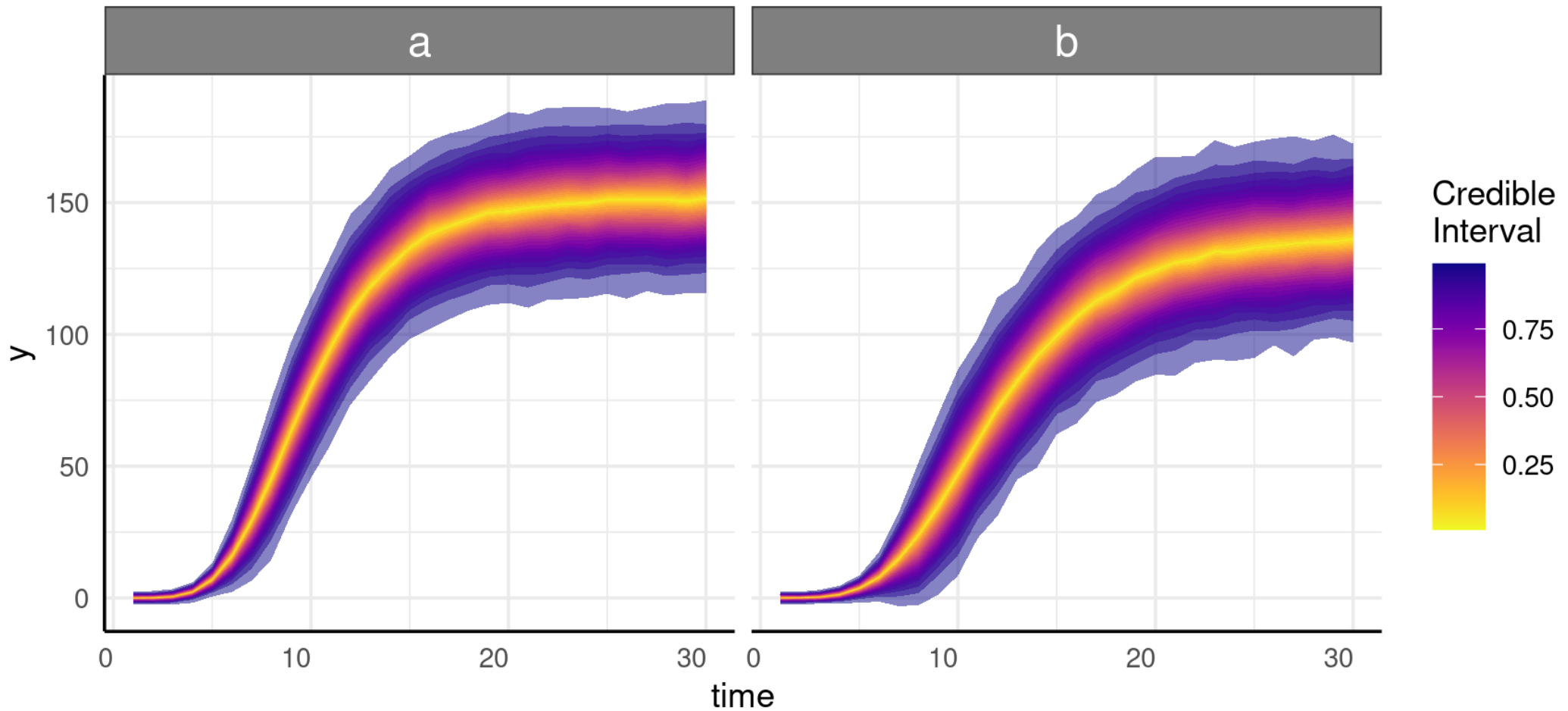
# Scenario 5

- Okay so we have some shortcomings, we are going to want a better model.

```
ss <- growthSS("gompertz", form = y ~ time|id/group, sigma = "logistic",  
               df = simdf, type = "brms",  
               start = list("A" = 125, "B" = 10, "C" = 0.2,  
                             "sigmaA" = 20, "sigmaB" = 15, "sigmaC" = 3))  
fit <- fitGrowth(ss, cores = 4, chains = 4, iter = 1000)
```

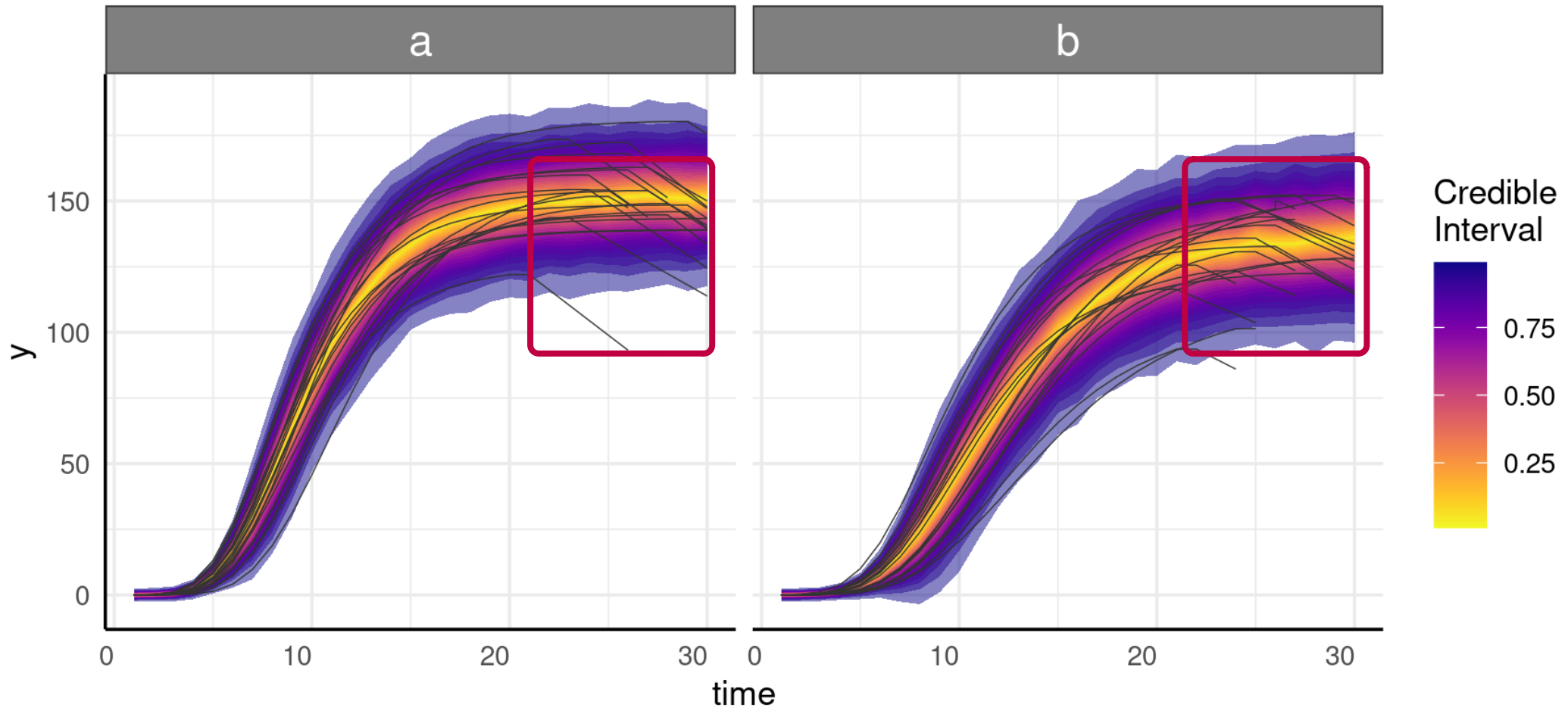
# Scenario 5

```
growthPlot(fit, form = ss$pcvrForm)
```



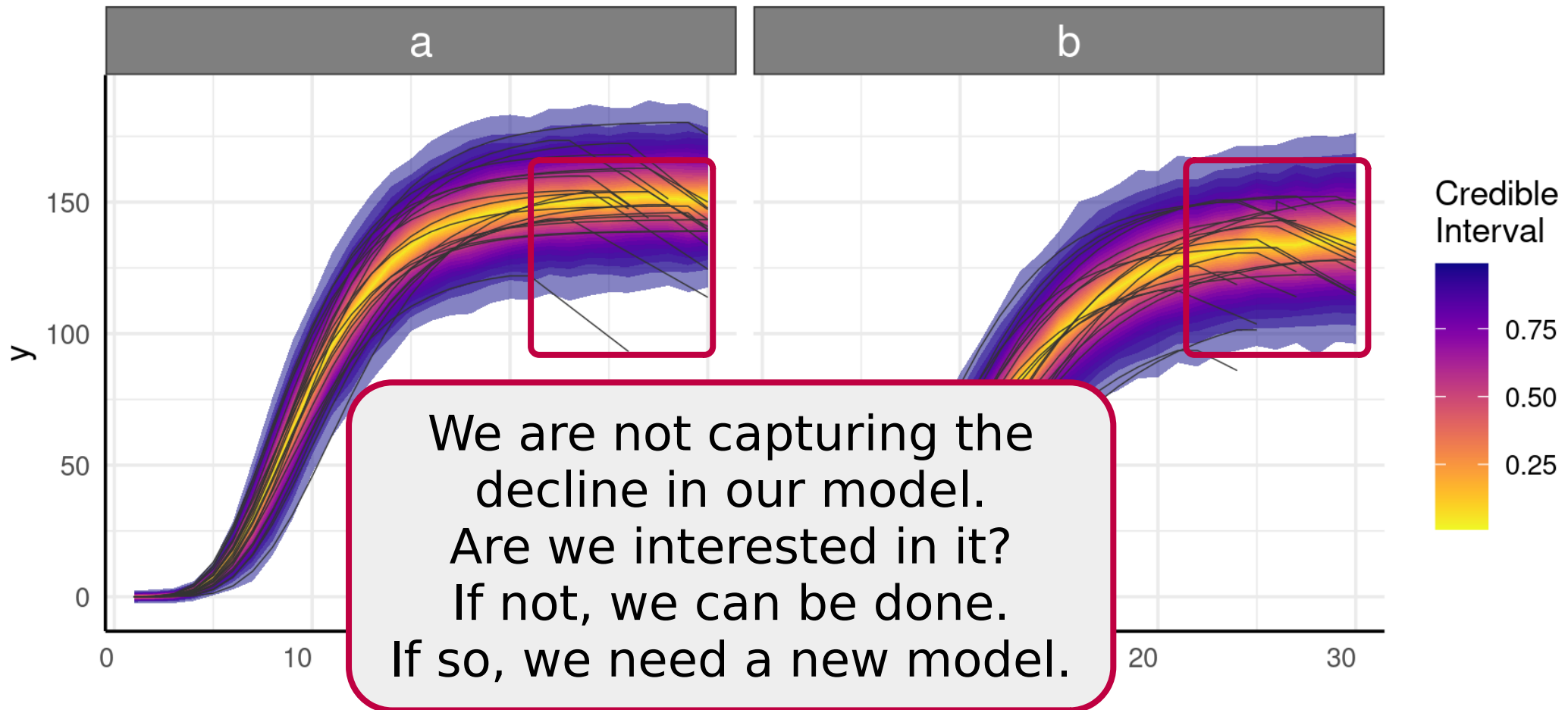
# Scenario 5

```
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```



# Scenario 5

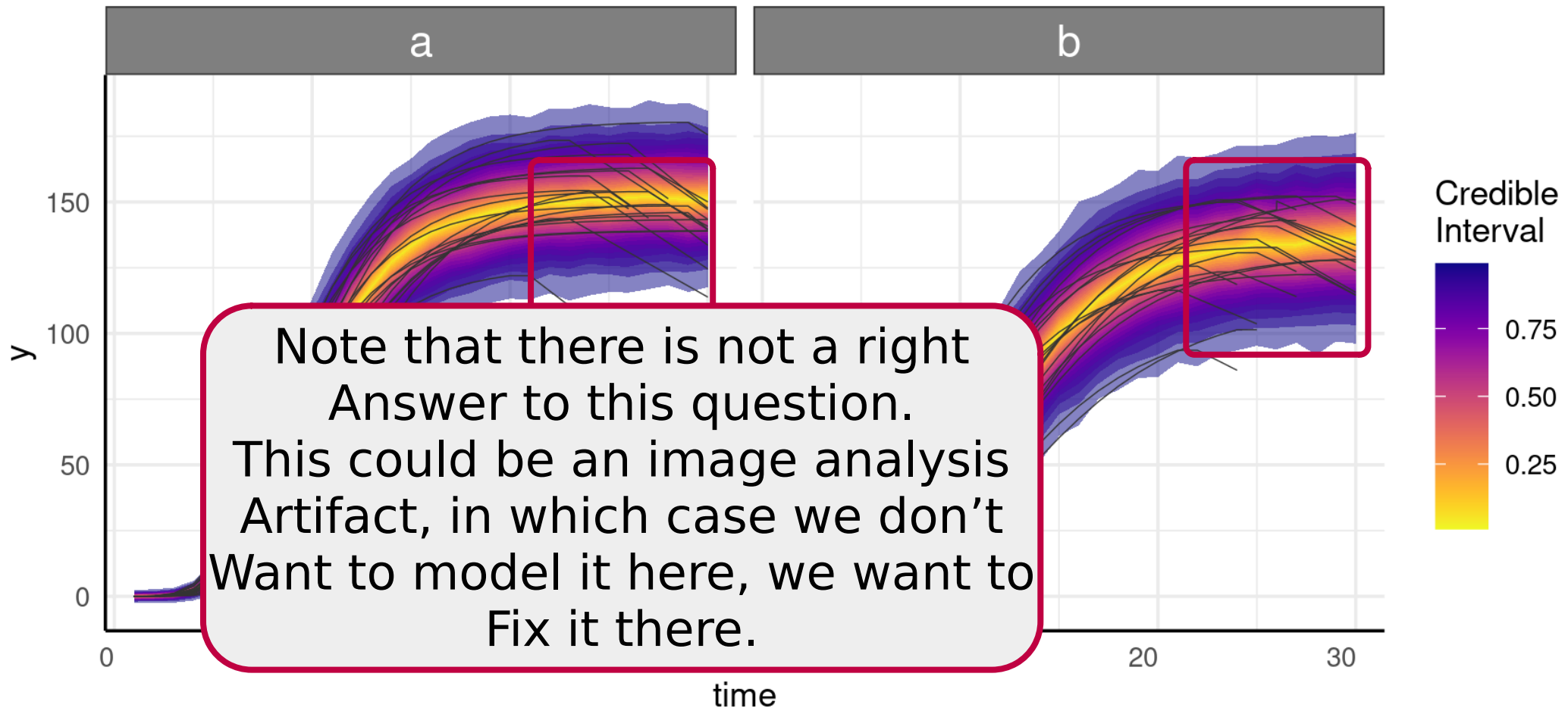
```
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```





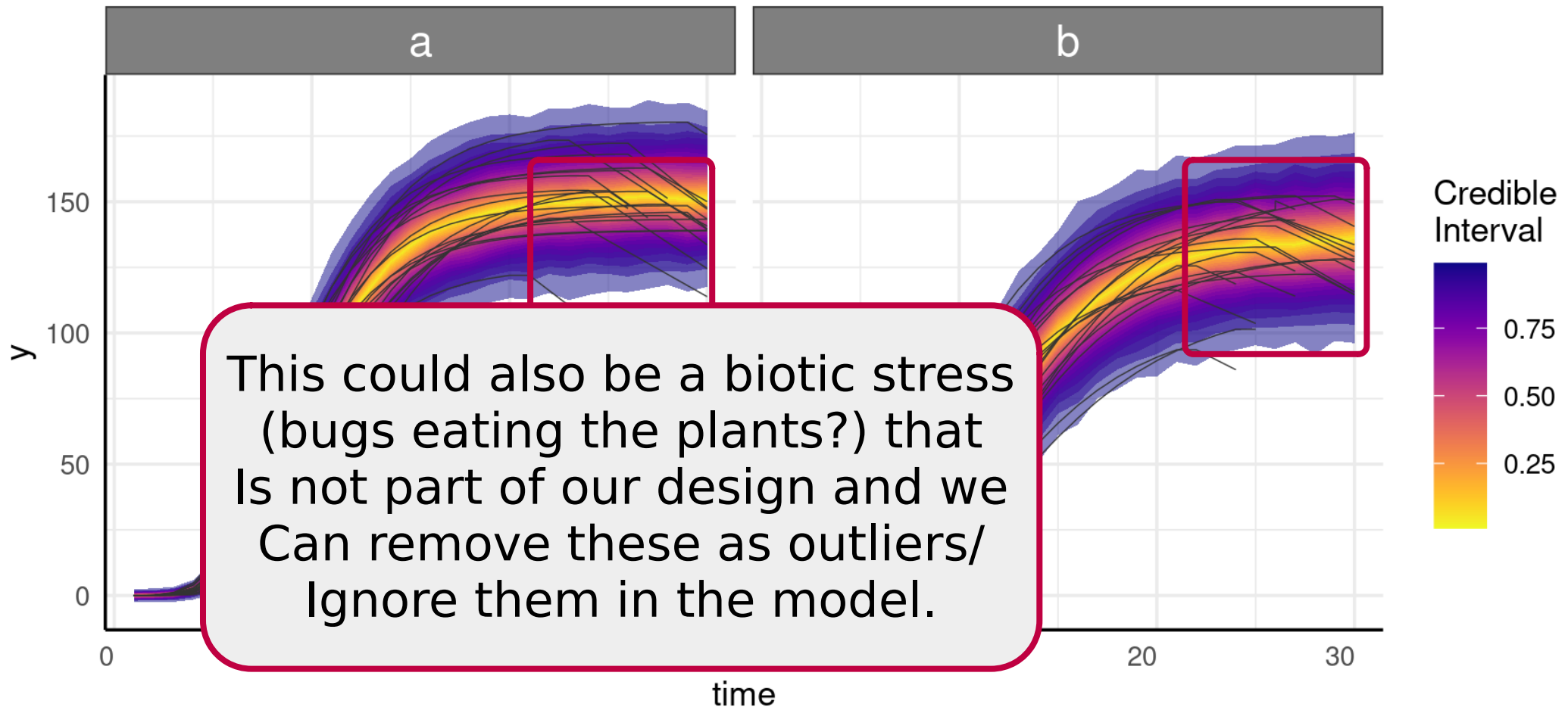
# Scenario 5

```
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```



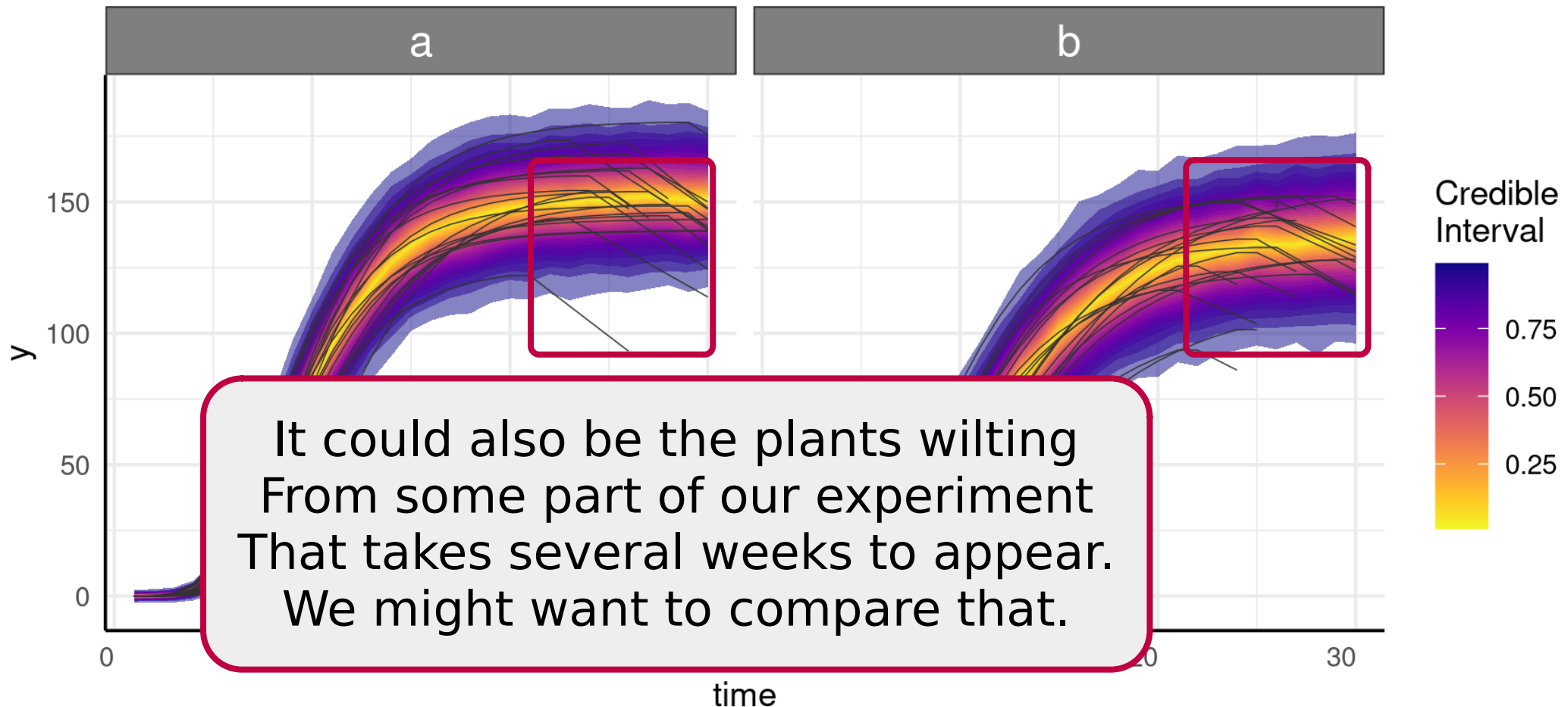
# Scenario 5

```
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```



# Scenario 5

```
growthPlot(fit, df = ss$df, form = ss$pcvrForm)
```



# Scenario 5

- We'll assume we want to model this trend.

```
ss <- growthSS("gompertz + linear decay", form = y ~ time|id/group,  
  sigma = "logistic", df = simdf,  
  start = list("gompertz1A" = 125,  
    "gompertz1B" = 10, "gompertz1C" = 0.2,  
    "changePoint1" = 25, "linear2A" = 5,  
    "sigmaA" = 20, "sigmaB" = 15, "sigmaC" = 3),  
  type = "brms")  
fit <- fitGrowth(ss, cores = 4, chains = 4, iter = 1000)
```

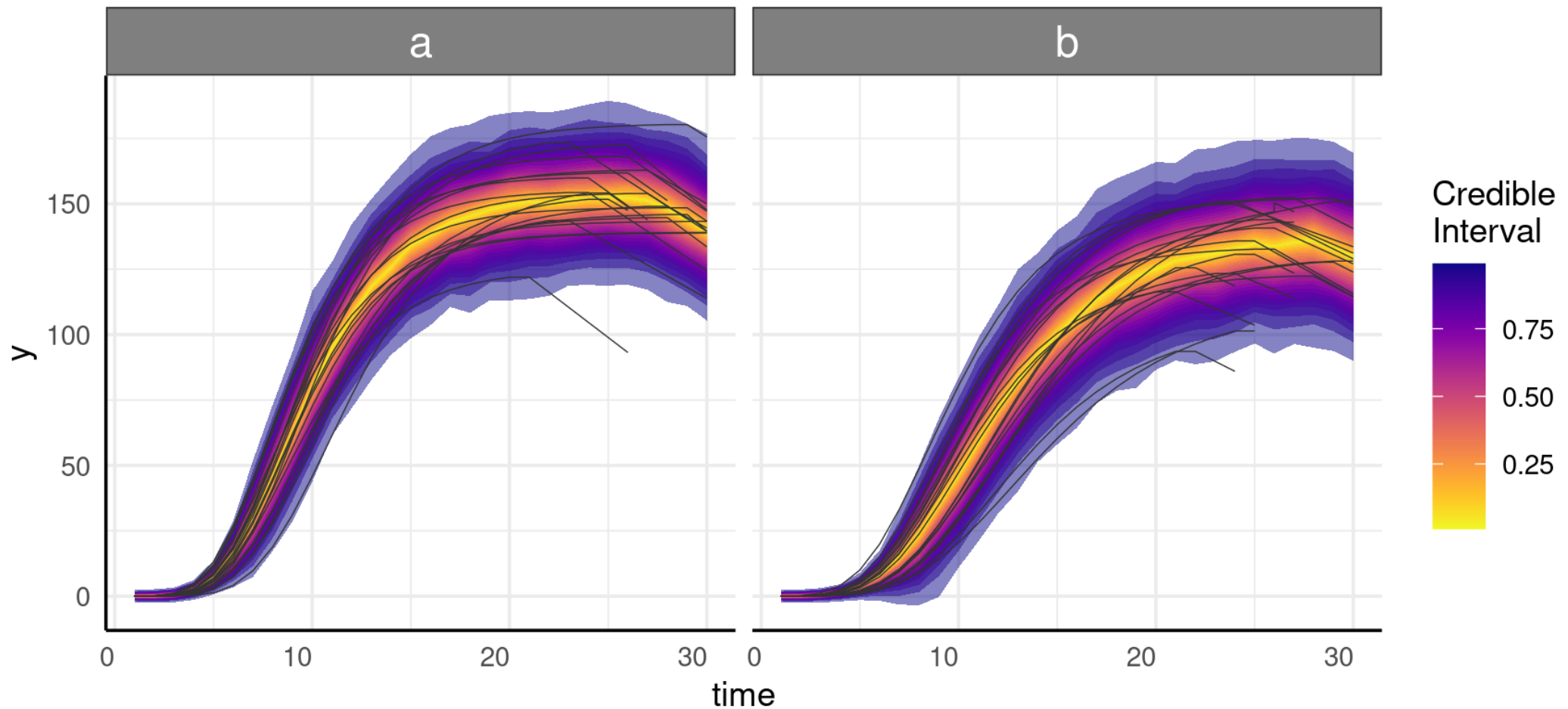
# Scenario 5

- We'll assume we want to model this trend.

```
ss <- growthSS("gompertz + linear decay", form = y ~ time|id/group,  
  sigma = "logistic", df = simdf,  
  start = list("gompertz1A" = 125,  
    "gompertz1B" = 10, "gompertz1C" = 0.2,  
    "changePoint1" = 25, "linear2A" = 5,  
    "sigmaA" = 20, "sigmaB" = 15, "sigmaC" = 3),  
  type = "brms")  
fit <- fitGrowth(ss, cores = 4, chains = 4, iter = 1000)
```

Now we have a changepoint model.  
We are estimating a changepoint and  
A trend of linear decay after that time.

# Scenario 5



# Scenario 5

```
> hyp1 <- brms::hypothesis(fit,"linear2A_groupa > linear2A_groupb")
> hyp2 <- testGrowth(fit = fit, test = "linear2A_groupa > linear2A_groupb")
> identical(hyp1, hyp2)
```

```
[1] TRUE
```

```
> hyp1
```

Hypothesis Tests for class b:

	Hypothesis	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob	Star
1	(linear2A_groupa)... > 0	-0.5	1.63	-3.25	2.19	0.63	0.38	

---

'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.

'\*': For one-sided hypotheses, the posterior probability exceeds 95%;

for two-sided hypotheses, the value tested against lies outside the 95%-CI.

Posterior probabilities of point hypotheses assume equal prior probabilities.

# Statistics in pcvr

- Introduction
- Frequentist and Bayesian statistics
- Conjugate
- Non-linear modeling
- Example scenarios
- **Resources**



# Resources and Conclusion

- This is a limited introduction to the **conjugate** comparisons that are possible and the modeling supported by **growthSS**, for more examples see documentation or vignettes/articles online:
  - <https://danforthcenter.github.io/pcvr/>
- Note that many conjugate options were used in place of the wilcoxon rank sum test, that is because in the normal flow chart we don't worry about distributions other than the Gaussian, but that is a large focus in **conjugate**.

# Resources and Conclusion

- This is a limited introduction to the **conjugate** comparisons that are possible and the modeling supported by **growthSS**, for more examples see documentation or vignettes/articles online:
  - <https://danforthcenter.github.io/pcvr/>
- Feel free to ask questions in slack or to raise issues in github.