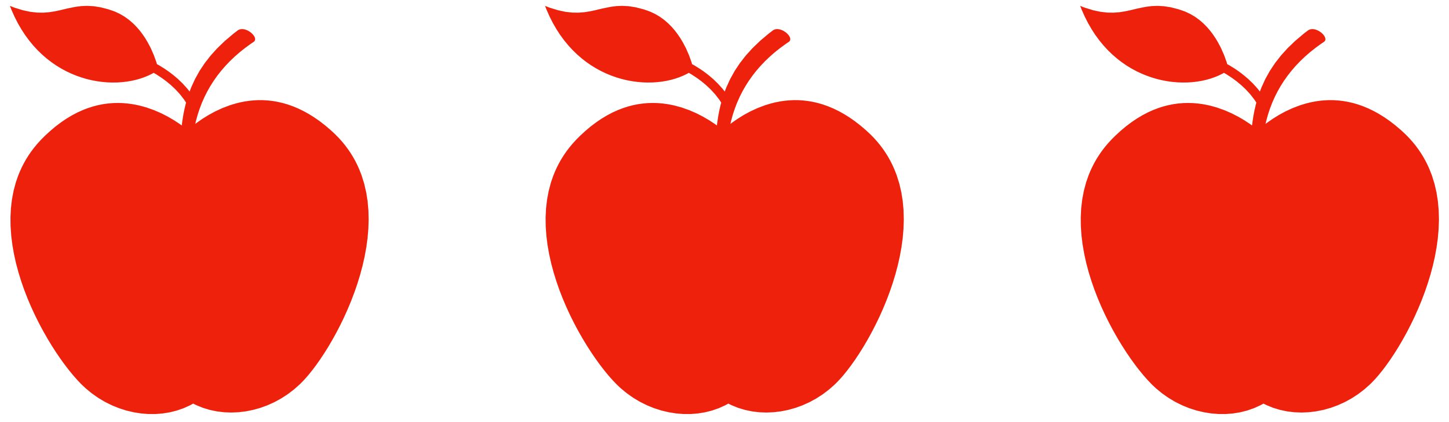
Fitting models to different types of data in R

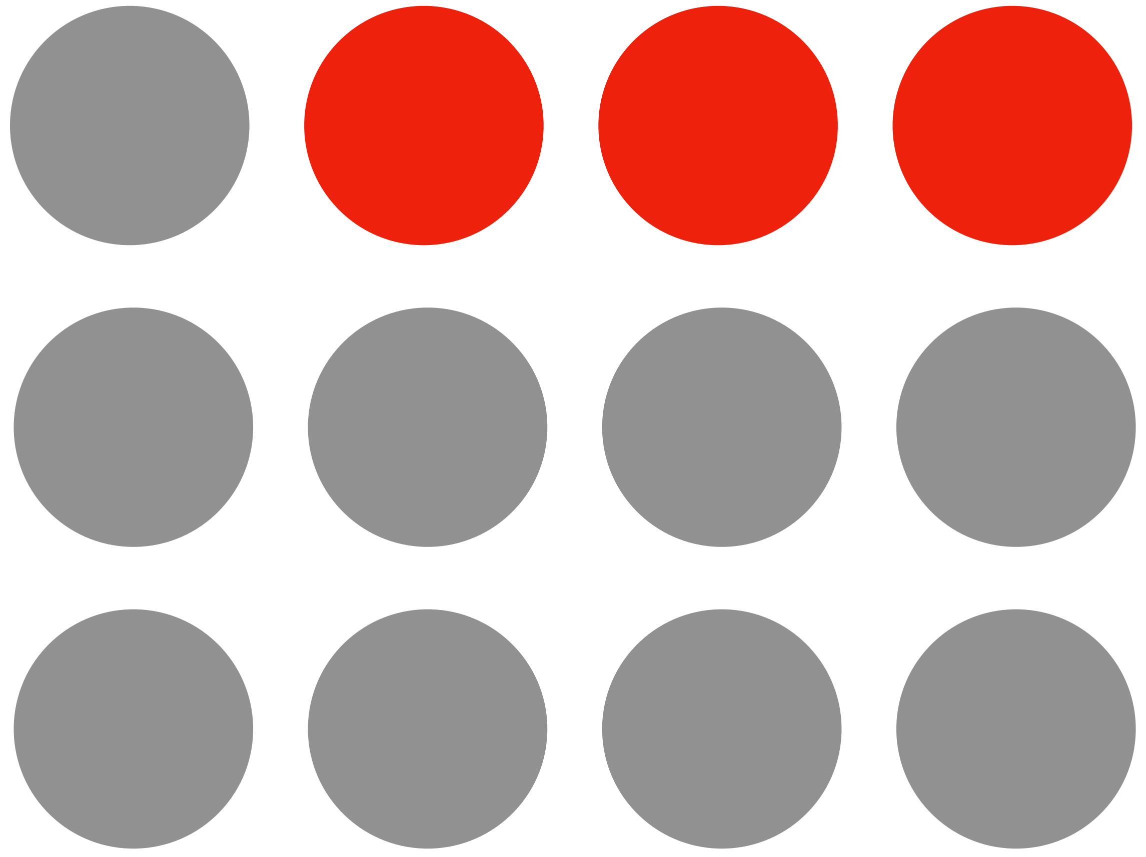
2020/6/12

ECRC Data Science Seminar

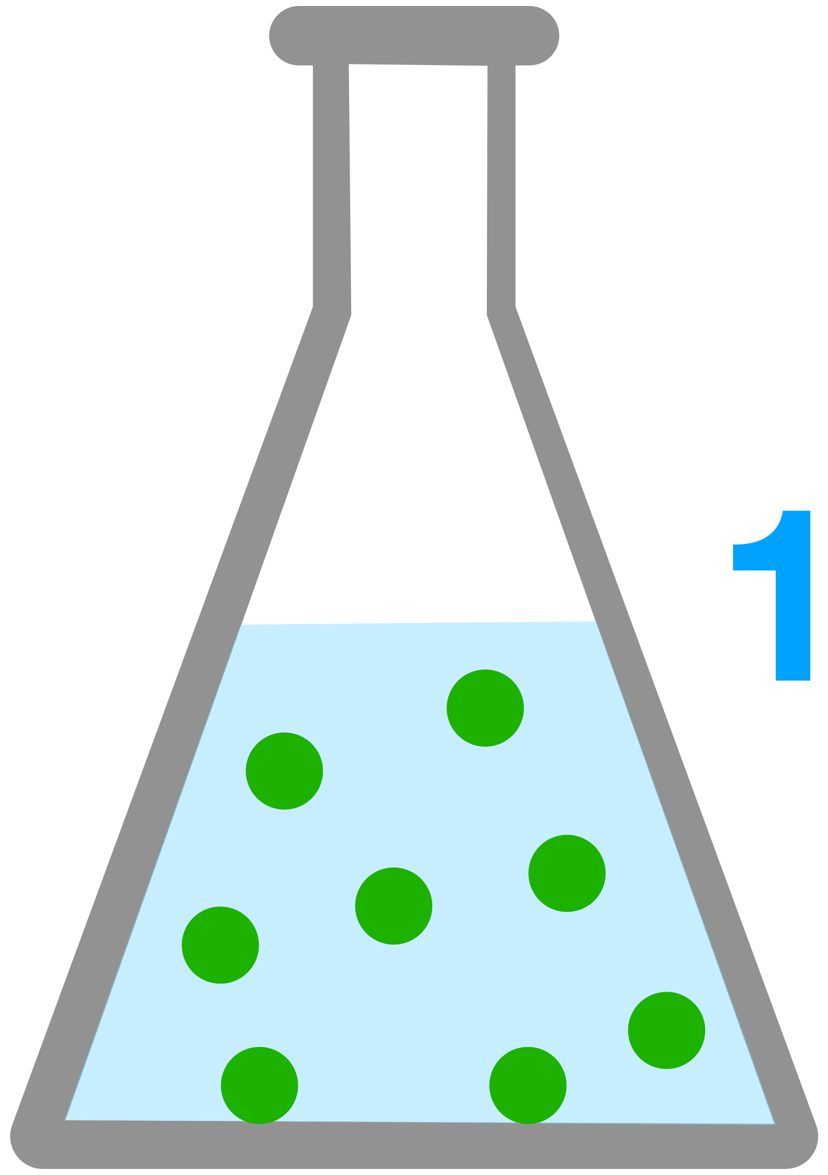
Chia-Yu Chen, AG Forslund



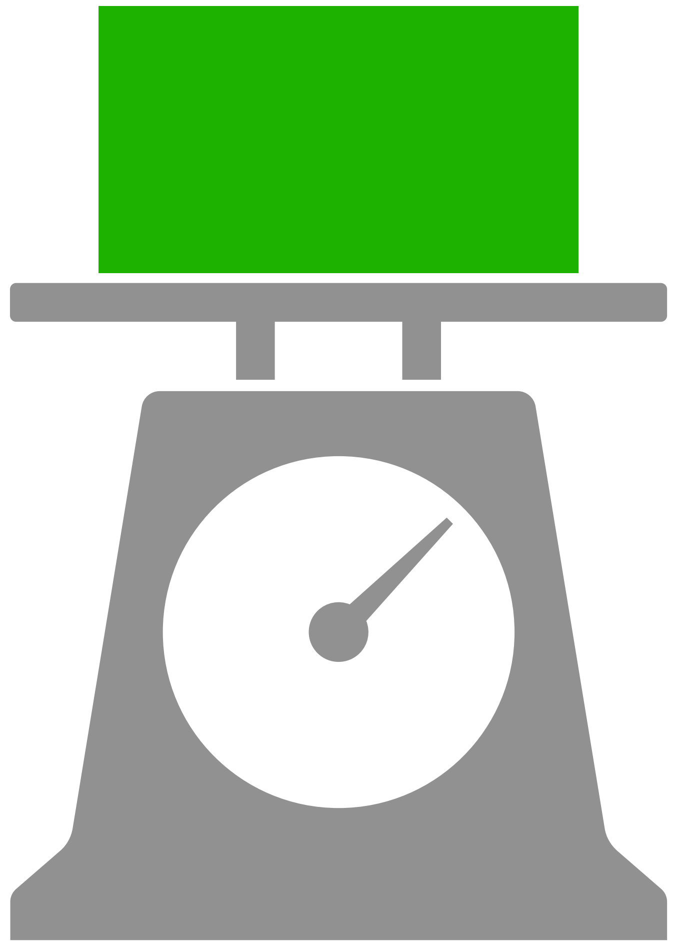
1 2 3...



25%



1.6 g/ml



12.9 kg

What kind of model should we fit
to each of the different types of data?

Data types

Quantitative	Count	• Non-negative integers resulted from counting • Discrete	• 10 apples • 80 dogs
	Measurement	• Can be measured at finer and finer scale • Continuous	• 1.6 g/ml • 9.5 cm
	Proportion	• Ranges from 0 to 1	• 25% classified as A • 10% classified as B
Qualitative	Binary	• Sort things into one of two mutually exclusive categories	• True/False • Reject/Accept
	Ordinal	• Ranked • The distance between two categories is not known	• Small/Medium/Large • Dislike/Neutral/Like

Simple linear model (LM)

$$y = a + bx + e$$

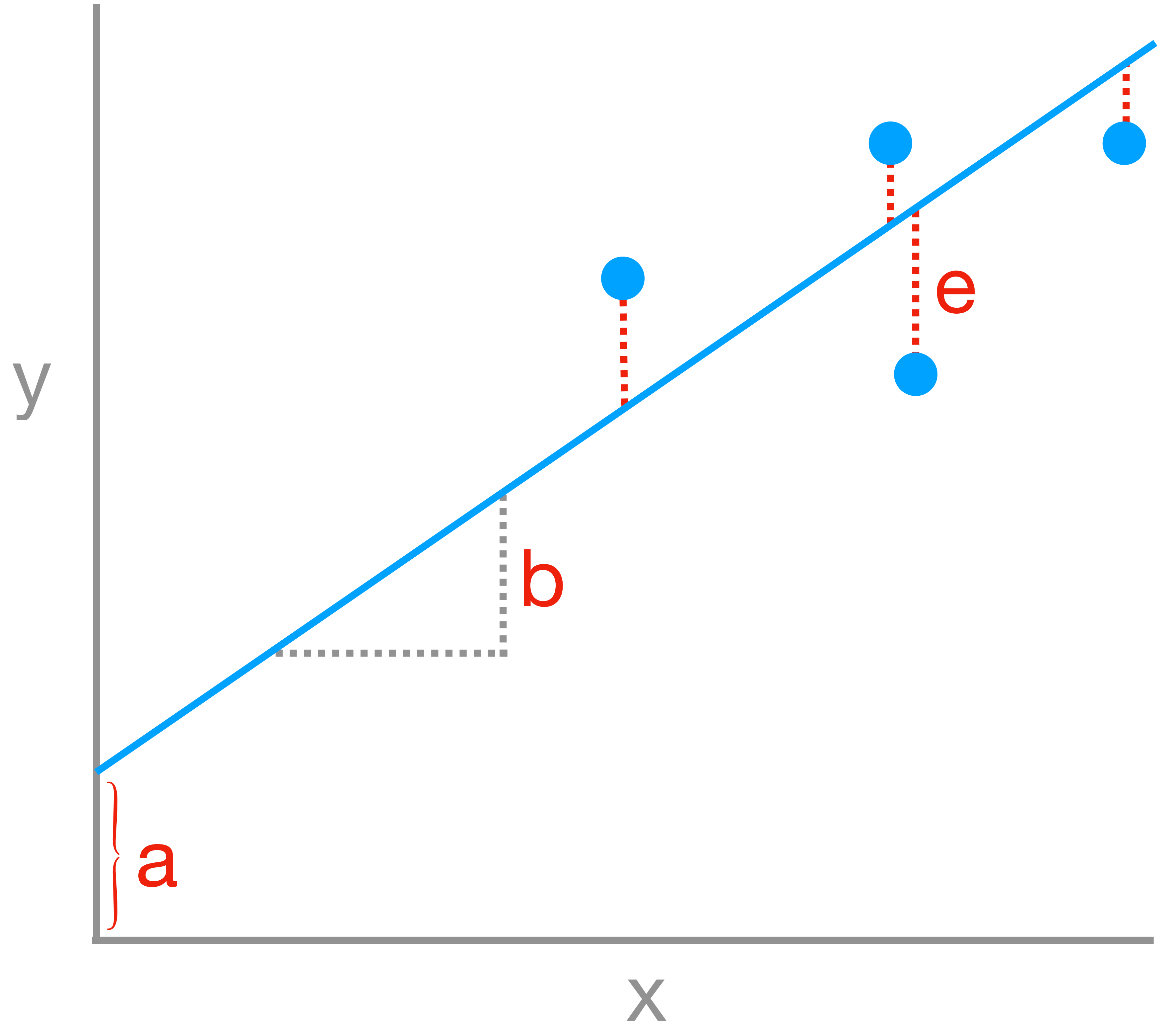
x: explanatory variable

y: dependent variable

a: intercept of regression line

b: slope of regression line

e: error term



Linear models in R

Math equation:

$$y = a + bx + e$$

R syntax:

$$y \sim x$$

```
model <- lm(formula = y~x, data = your_data)
```

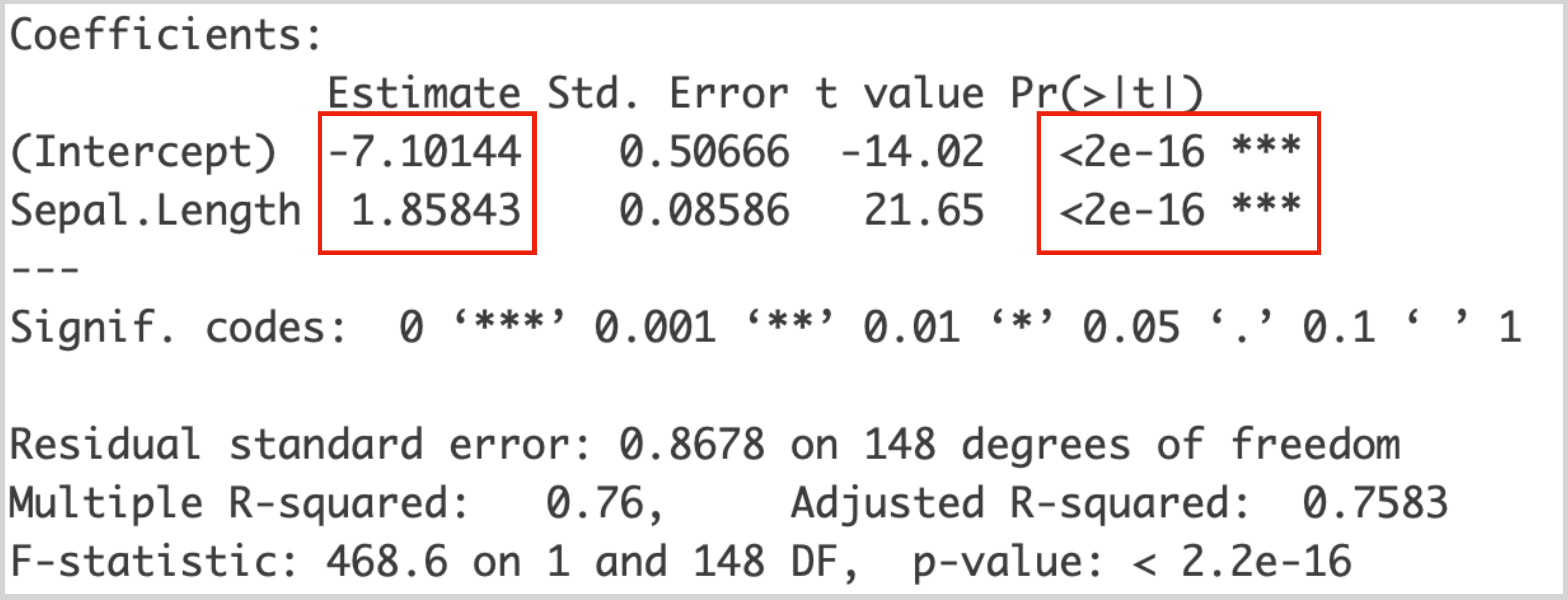
Linear models in R

Iris dataset

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5.0	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1

Linear models in R

```
model = lm(formula = Petal.Length ~ Sepal.Length, data = iris)
summary(model)
```



```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -7.10144    0.50666  -14.02   <2e-16 ***
Sepal.Length  1.85843    0.08586   21.65   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8678 on 148 degrees of freedom
Multiple R-squared:   0.76,	Adjusted R-squared:  0.7583 
F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

Petal.Length = -7.1 + 1.9 *Sepal.Length

Simple linear model requirements

$$y_i = a + bx_i + e_i$$

1. y is a continuous variable
2. y is normally distributed
3. A linear relationship between the y and x
4. Homogeneity of variance: the variance of y for each value of x is constant
5. Errors are normally distributed

However, there are many scenarios where these assumptions are not met.
In these cases, fitting data with simple linear model isn’t appropriate.

Generalized linear model (GLM)

GLM is a flexible generalization of linear model

1. y can be either continuous or discrete
2. y doesn’t need to be normally distributed
3. Doesn’t assume a linear relationship between the y and x
4. The homogeneity of variance does NOT need to be satisfied.
5. Errors doesn’t need to be normally distributed

GLM generalizes linear regression by allowing the linear model to be related to the response variable (y) via a link function.

Generalized linear model (GLM)

GLM is made up of a linear predictor and two functions:

1. **Linear predictor** η_i **:** linear sum of the effects of one or more explanatory variables

$$\eta_i = a + b_1 x_{1i} + \ldots + b_p x_{pi}$$

2. **Link function:** describes how the mean of the response (expected value) depends on the linear predictor η_i:

$$g(\mu_i) = \eta_i$$

3. **Variance function:** describes how the variance of the response depends on the mean (dispersion parameter θ is a constant)

$$var(y_i) = \theta V(\mu)$$

LM is a special case of GLMs

$$y_i = a + b_1x_{1i} + b_2x_{2i} + e_i$$

1. **Linear predictor :**

$$\eta_i = a + b_1x_{1i} + b_2x_{2i}$$

2. **Link function (identity link, the simplest link function):**

$$g(\mu_i) = \mu_i = \eta_i$$

3. **Variance function (variance is independent of mean and is a constant)**

$$var(y_i) = \theta V(\mu)$$

$$V(\mu_i) = 1$$

Generalized linear model (GLM)

Model
Linear
Logistic
Poisson
Beta

Generalized linear model (GLM) in R

Similar to the lm function, we can fit GLMs with glm function:

```
model <- glm(formula = y~x, family = “poisson”, data = your_data)
```

The choice of family is dependent on the property of y.

It can be binomial, gaussian, poisson, quasi, quasibinomial, Gamma, quasipoisson……

Data types

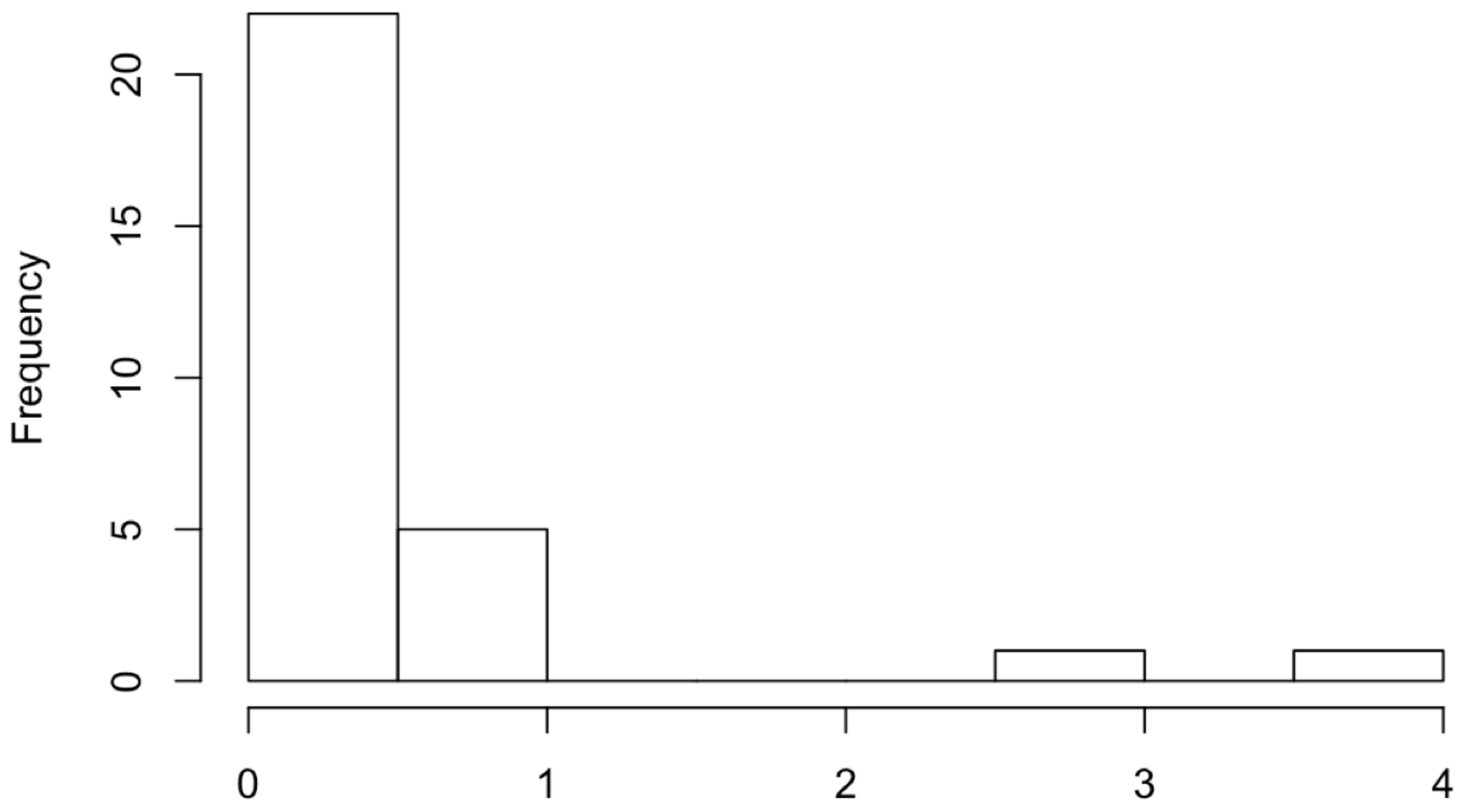
Quantitative	Count	- Non-negative integers resulted from counting - Discrete	- 10 apples - 80 dogs
	Measurement	- Can be measured at finer and finer scale - Continuous	- 1.6 g/ml - 9.5 cm
	Proportion	- Ranges from 0 to 1	- 25% classified as A - 10% classified as B
Qualitative	Binary	- Sort things into one of two mutually exclusive categories	- True/False - Reject/Accept
	Ordinal	- Ranked - The distance between two categories is not known	- Small/Medium/Large - Dislike/Neutral/Like

Data types

Quantitative	Count	• Non-negative integers resulted from counting • Discrete	• 10 apples • 80 dogs

Count data

- Observations can take only the non-negative integer values (0, 1, 2, …..)
- These integers arise from counting, not ranking or binary signal
- Skewed distribution: Contain a large number of data points for just a few values, making the frequency distribution skewed
- Sparsity: Many data points are zero



Model for count data: Poisson regression

- Poisson regression assumes the response variable y has a Poisson distribution
- Poisson distribution formula:

$$Pr\{Y = y\} = \frac{e^{-\mu}\mu^{y}}{y!} \qquad y = \{0,1,2,\ldots,n\}$$

$\mu > 0$

- Variance equals to mean

$$var(Y) = \mu$$

- Overdispersion: $var(Y) > \mu$ (variance > mean)
- Ignoring overdispersion causes confidence intervals to be too narrow and inflates the rate of false positives

Model for count data: negative binomial regression

- Generalization of Poisson regression
- Negative binomial distribution formula: y = number of failures before r^th^ success

$$Pr\{Y = y\} = \binom{r+y-1}{y} p^r (1-p)^y \qquad y = \{0,1,2,\ldots,n\}$$

- Doesn't assume variance equals to mean, allows overdispersion

$$var(Y) = \mu + \frac{\mu^2}{\theta} \qquad \theta = \text{dispersion parameter}$$

- Better than Poisson when there's overdispersion

Count data

Individual	Time_point	Microbial_abundance
a	1	0
a	2	3
a	3	5
b	1	8
b	2	14
b	3	29
c	1	0
c	2	35
c	3	6

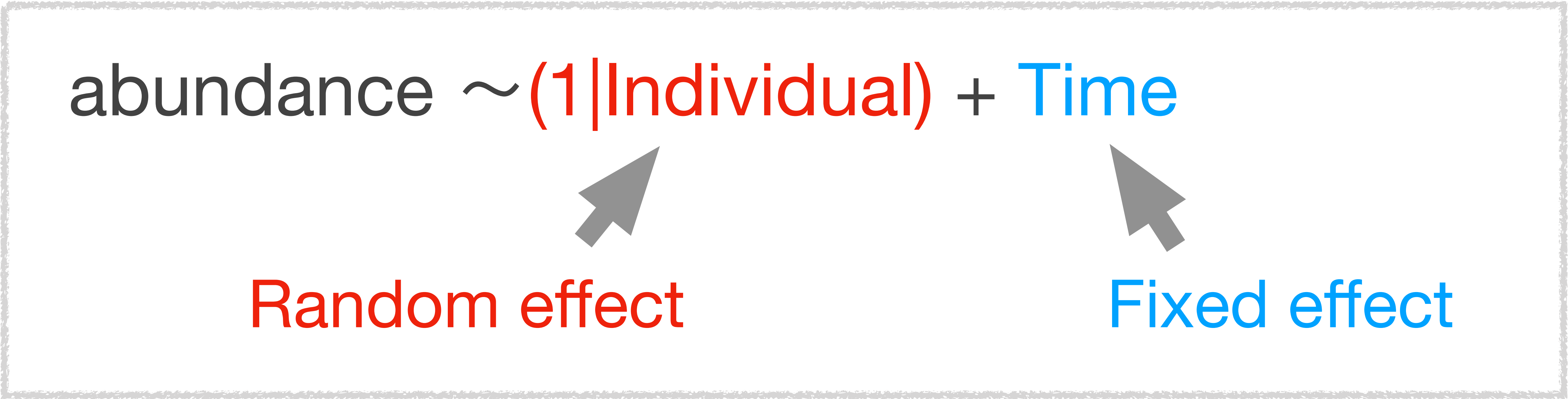
Negative binomial model

To find out if Time_point is a significant predictor of Microbial_abundance:

```
glmmTMB( Microbial_abundance ~ (1|Individual) + Time_point, family = nbinom2)
```

glmmTMB: A function from glmmTMB package capable of fitting linear and generalized linear mixed models

Mixed effect models



abundance ~(1|Individual) + Time

Random effect

Fixed effect

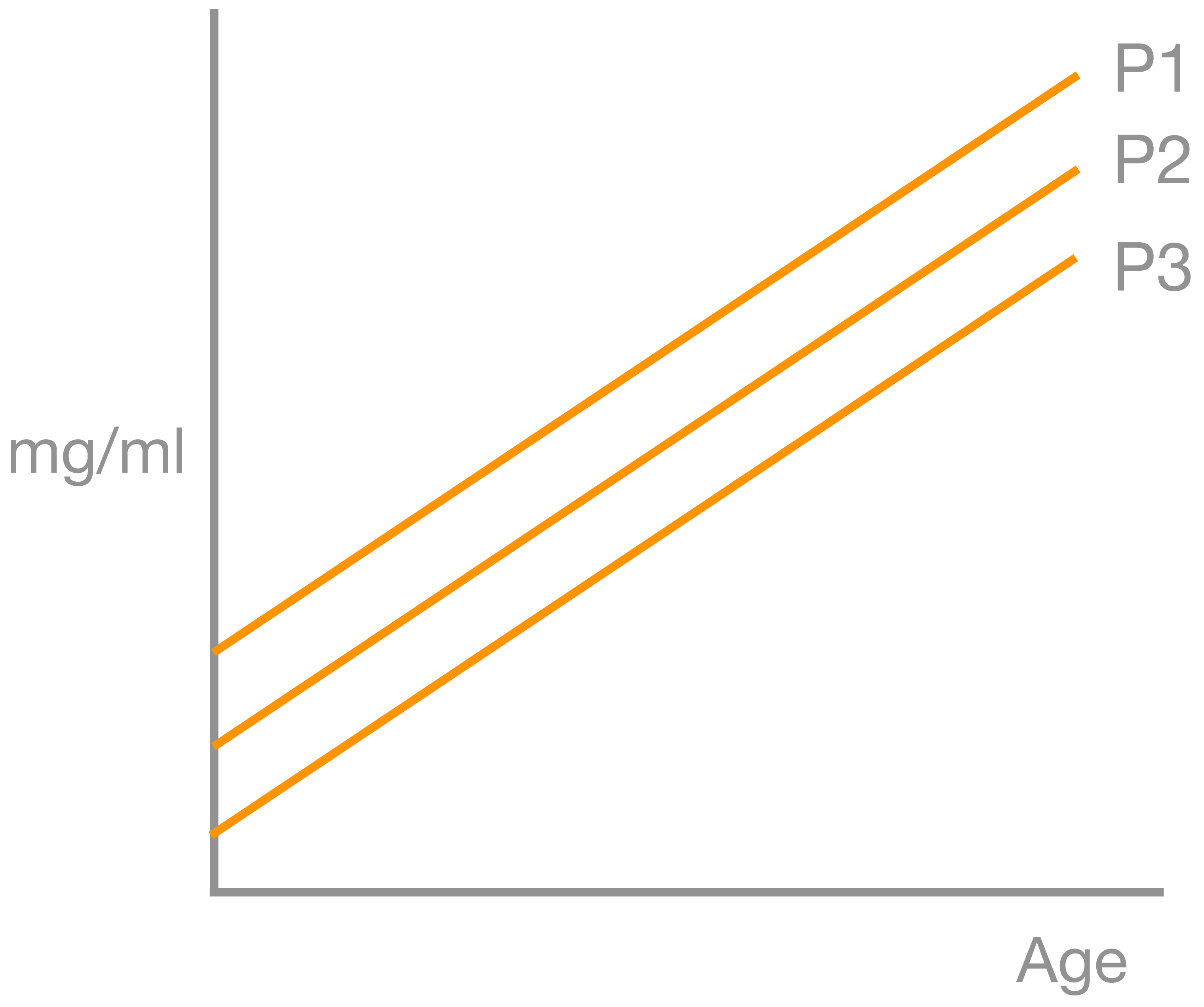
- Mixed effect model: Having both fixed effect and random effect in a model
- Random effect: takes the differences between individual study effects into account
- Used when there is non-independence in the data
- Hierarchical structure in data: Each classroom sample 10 students and compare
- Repeated measure: multiple measurement from same patient

Random intercept and random slope

“Patient” as random variable

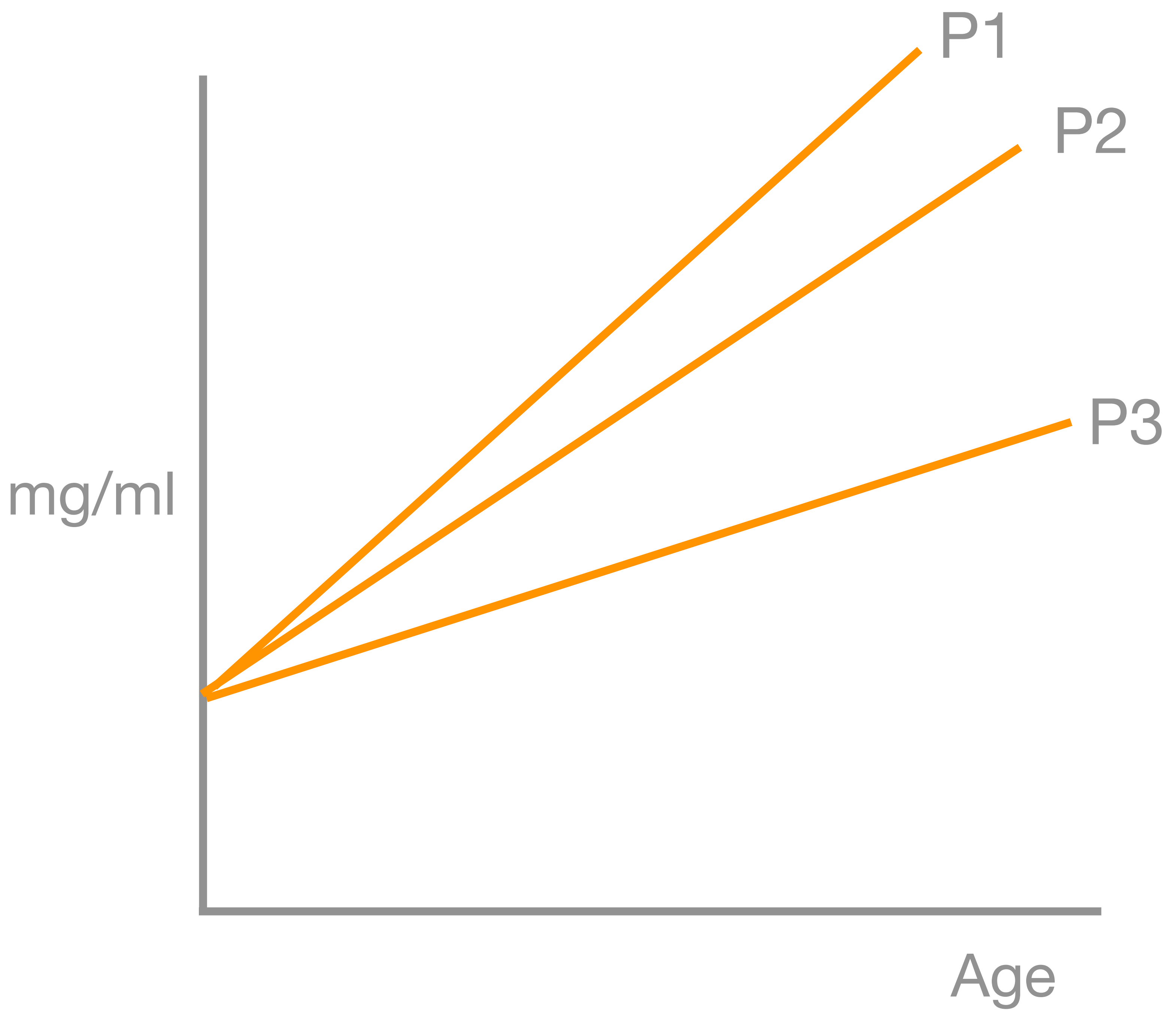
Random intercept

(1 | Patient)



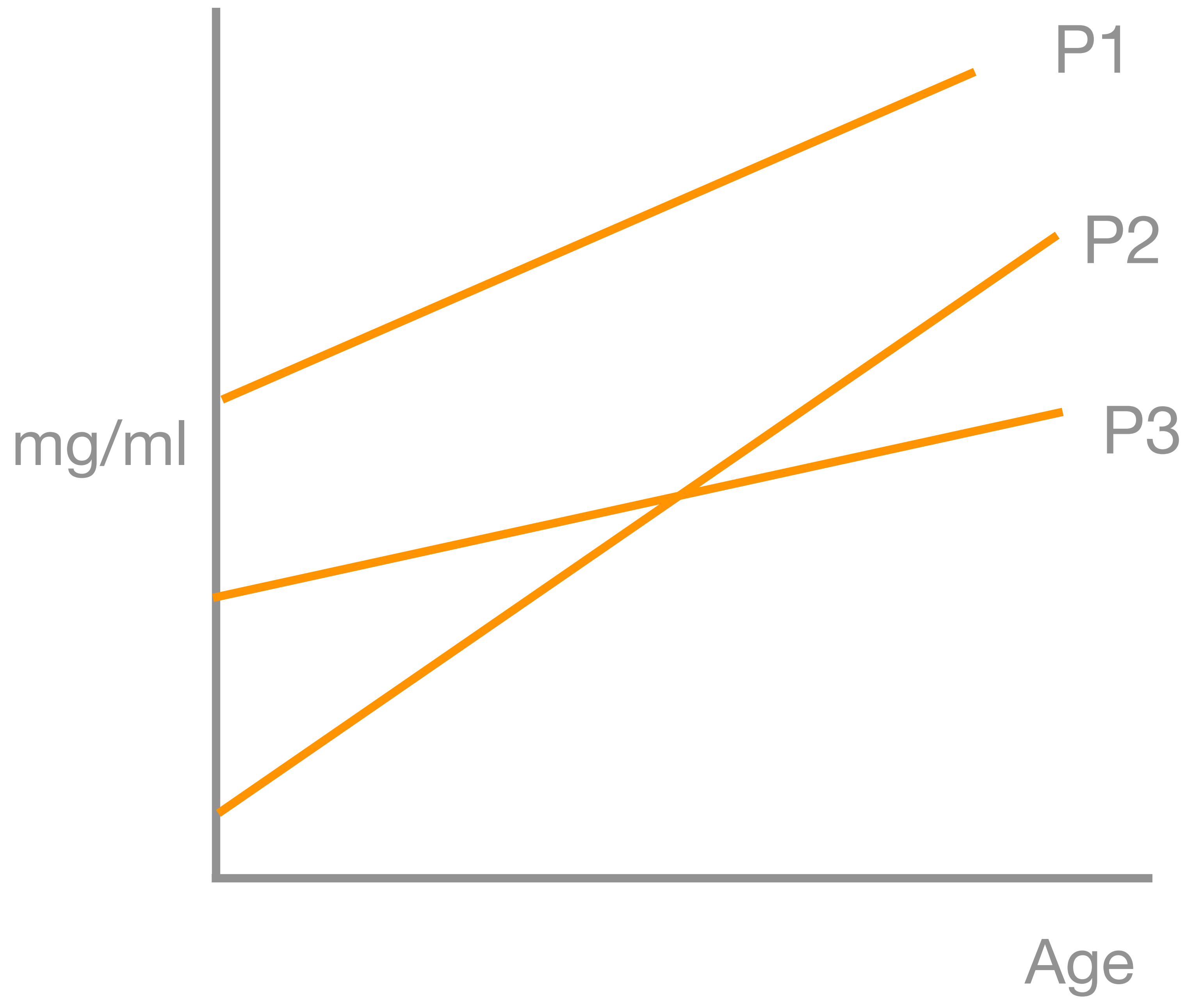
Random slope

(Age | Patient)



Random slope and intercept

(1 + Age | Patient)



Definitions of fixed and random effects

Fixed effect

Fixed effects don’t change across individuals

When samples exhaust the population, the variable is fixed
Example: gender: male/female, dosage: low/high

Fixed effects are those you are interested in

Random effect

Random effects vary across individuals

When the sample only covers a small part of all the possible levels, it’s random
Example: patients

Random effects are the ones you’re not interested in

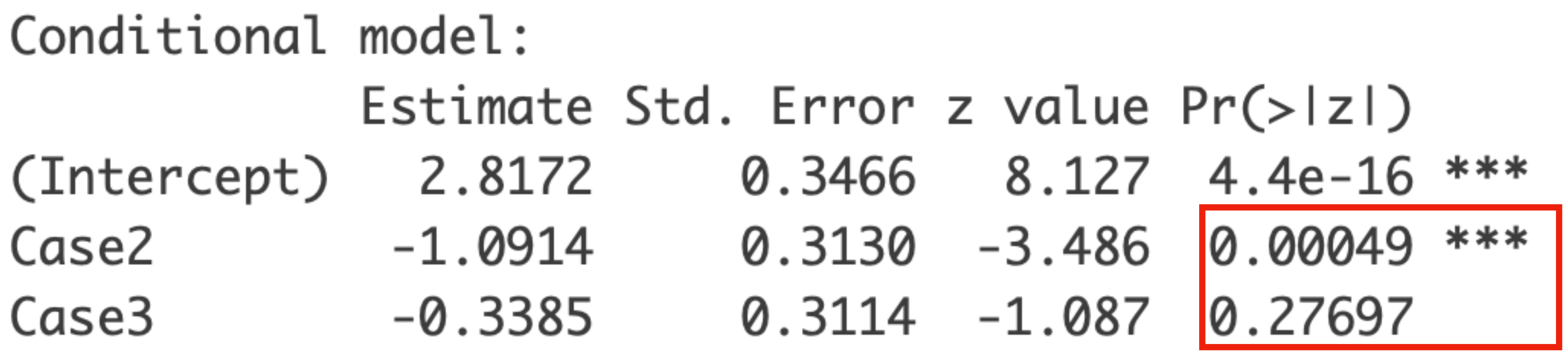
Random effects are most useful when the grouping variable has more than 5 levels. A binary variable shouldn’t be treated as a random effect.

Mixed effect models in R

	Simple linear model	Generalized linear model
Fixed effect model	lm( )	glm( )
Mixed effect model	lmer( )	glmer( ), glmmTMB( )…

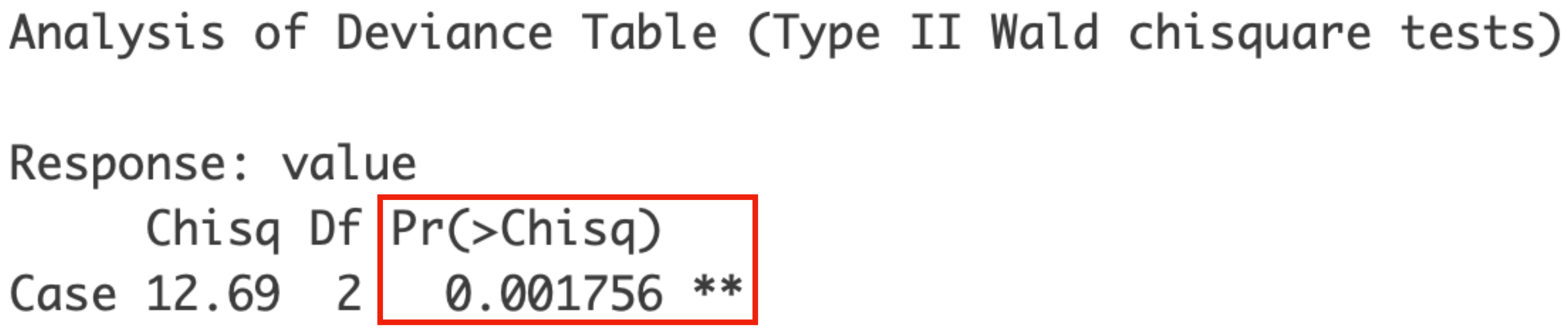
Negative binomial model

model <- glmmTMB(abundance ~ (1|Individual) + Time_point, family = nbinom2)



```
Conditional model:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.8172     0.3466   8.127  4.4e-16 ***
Case2        -1.0914     0.3130  -3.486  0.00049 ***
Case3        -0.3385     0.3114  -1.087  0.27697
```

Anova(model, test.statistic=c("Chisq"))

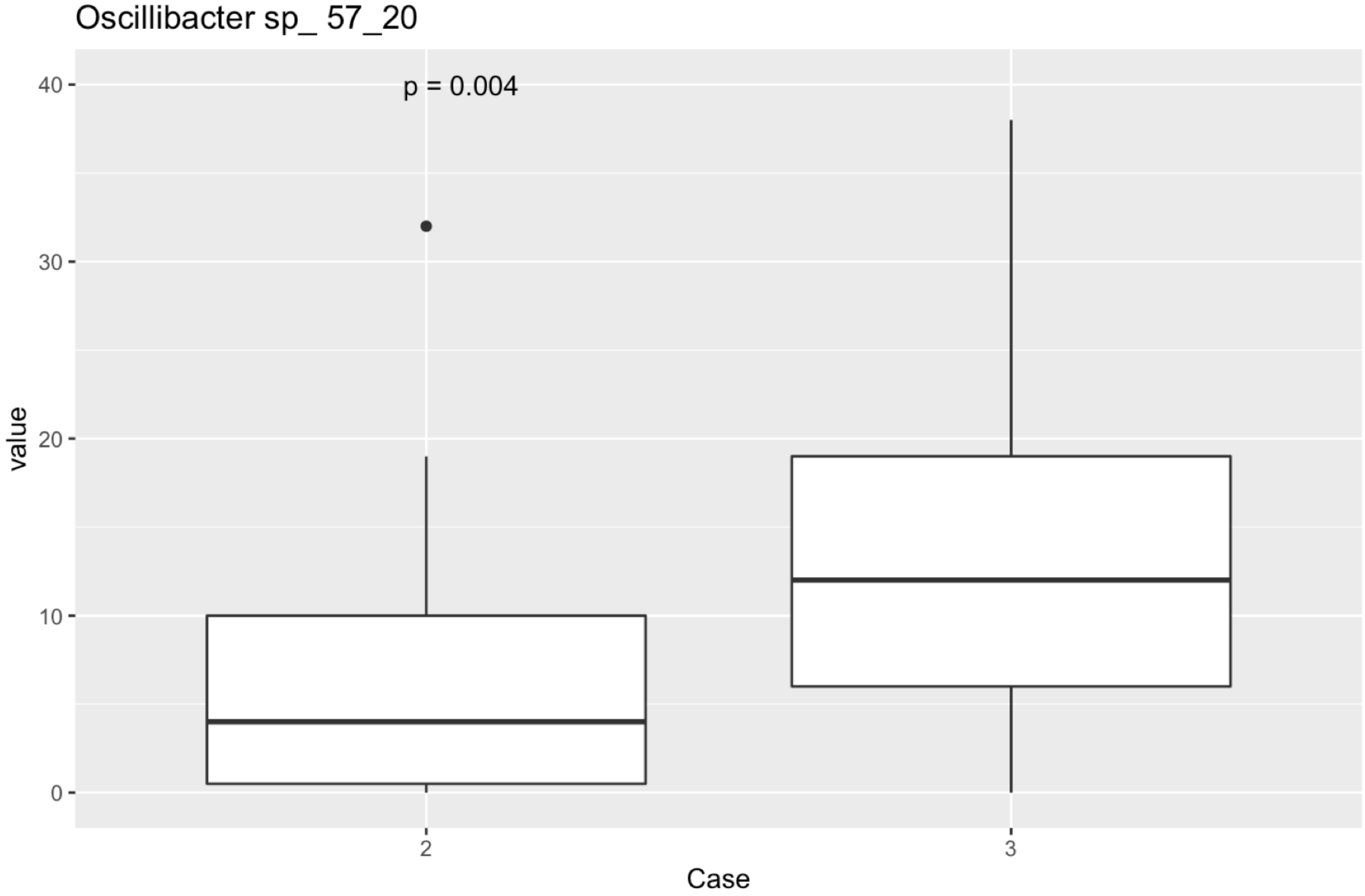


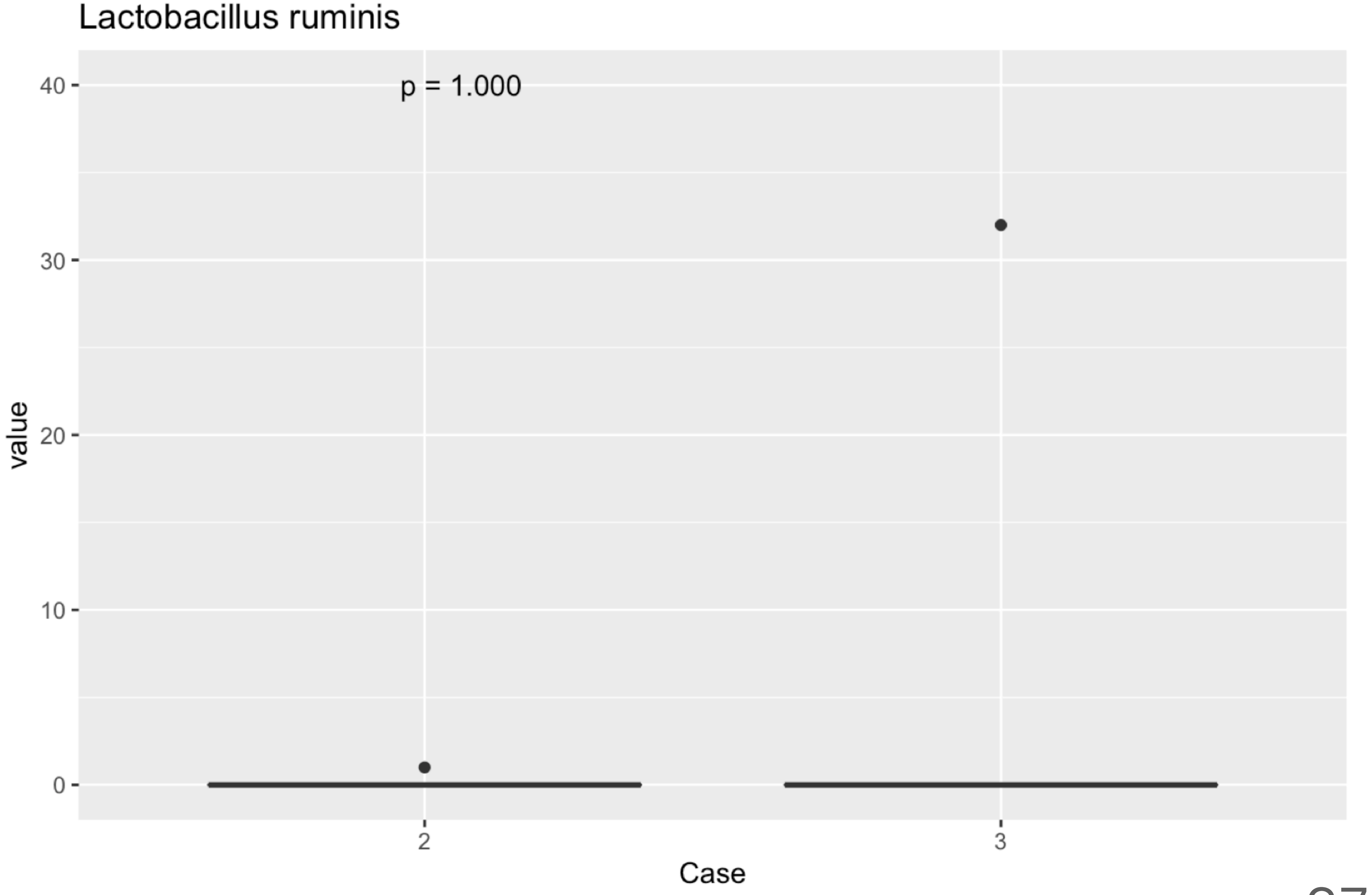
```
Analysis of Deviance Table (Type II Wald chisquare tests)

Response: value
     Chisq Df Pr(>Chisq)
Case 12.69  2   0.001756 **
```

Theta threshold

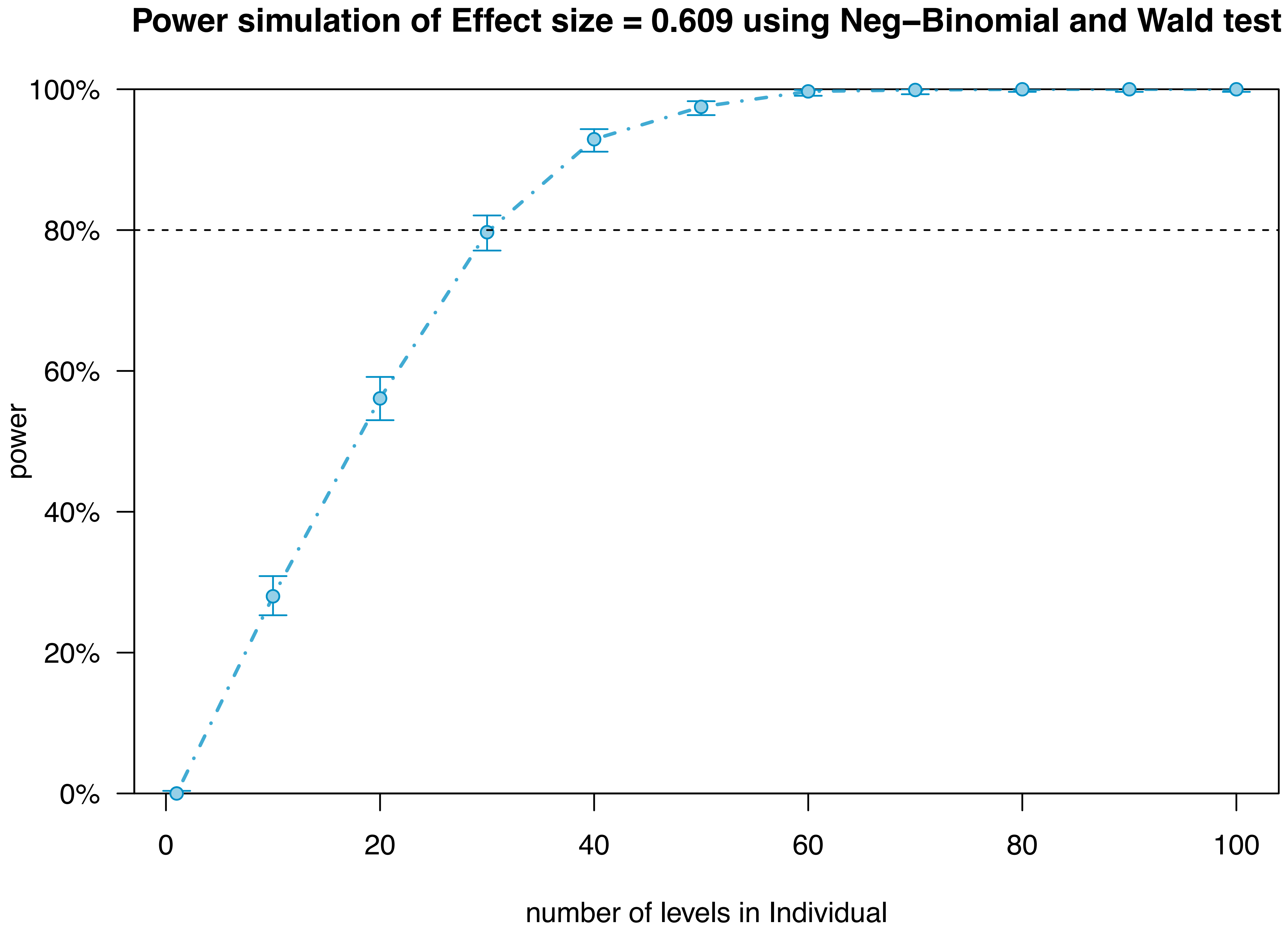
	Effect size between C2 & C3	Absolute sparsity (n = 46)
Oscillibacter sp. 57_20	0.609	11
Lactobacillus ruminis	0.0435	44

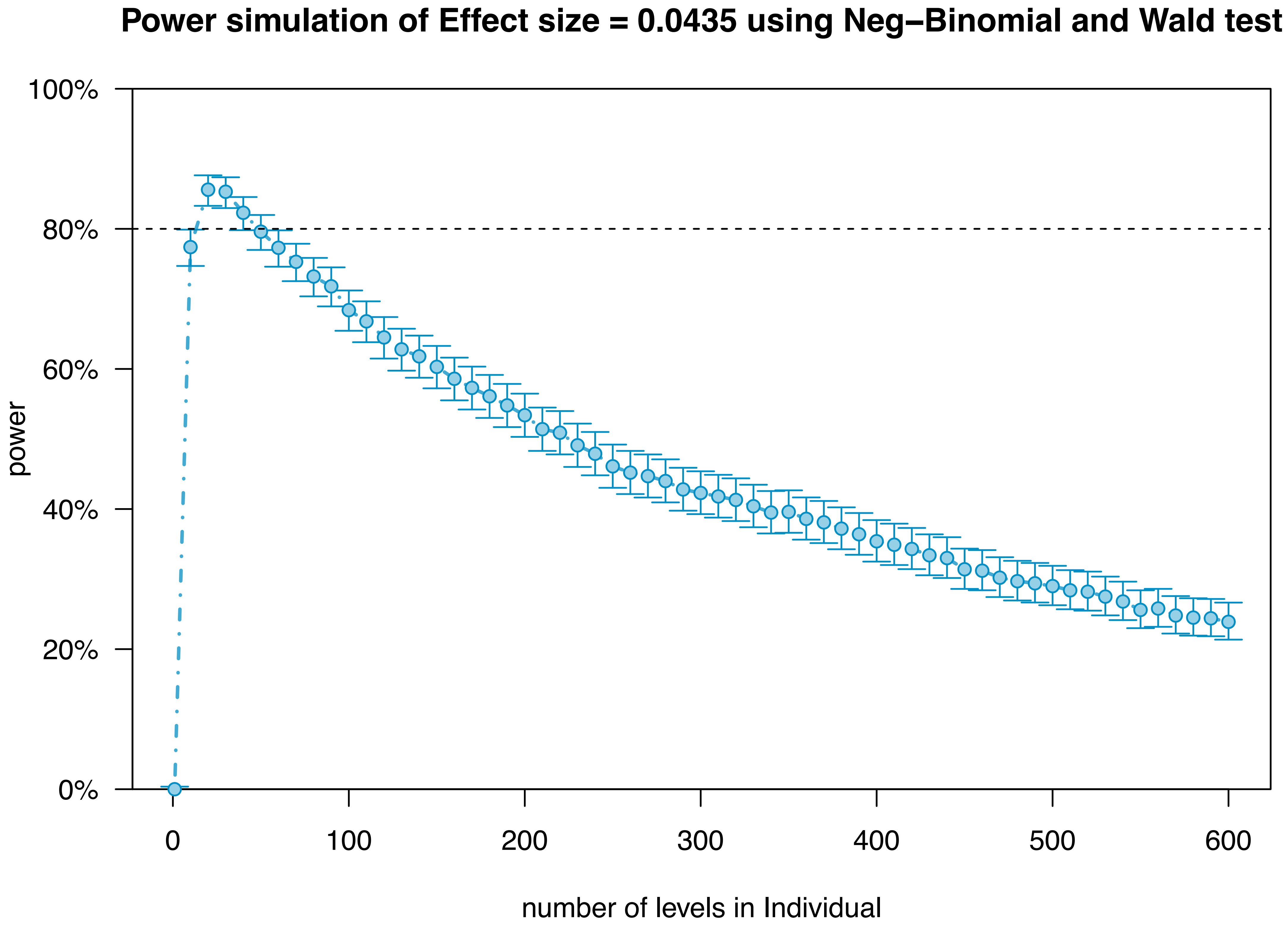




Incorrect power simulation curve

	Effect size	Absolute sparsity
Oscillibacter sp. 57_20	0.609	11
Lactobacillus ruminis	0.0435	44





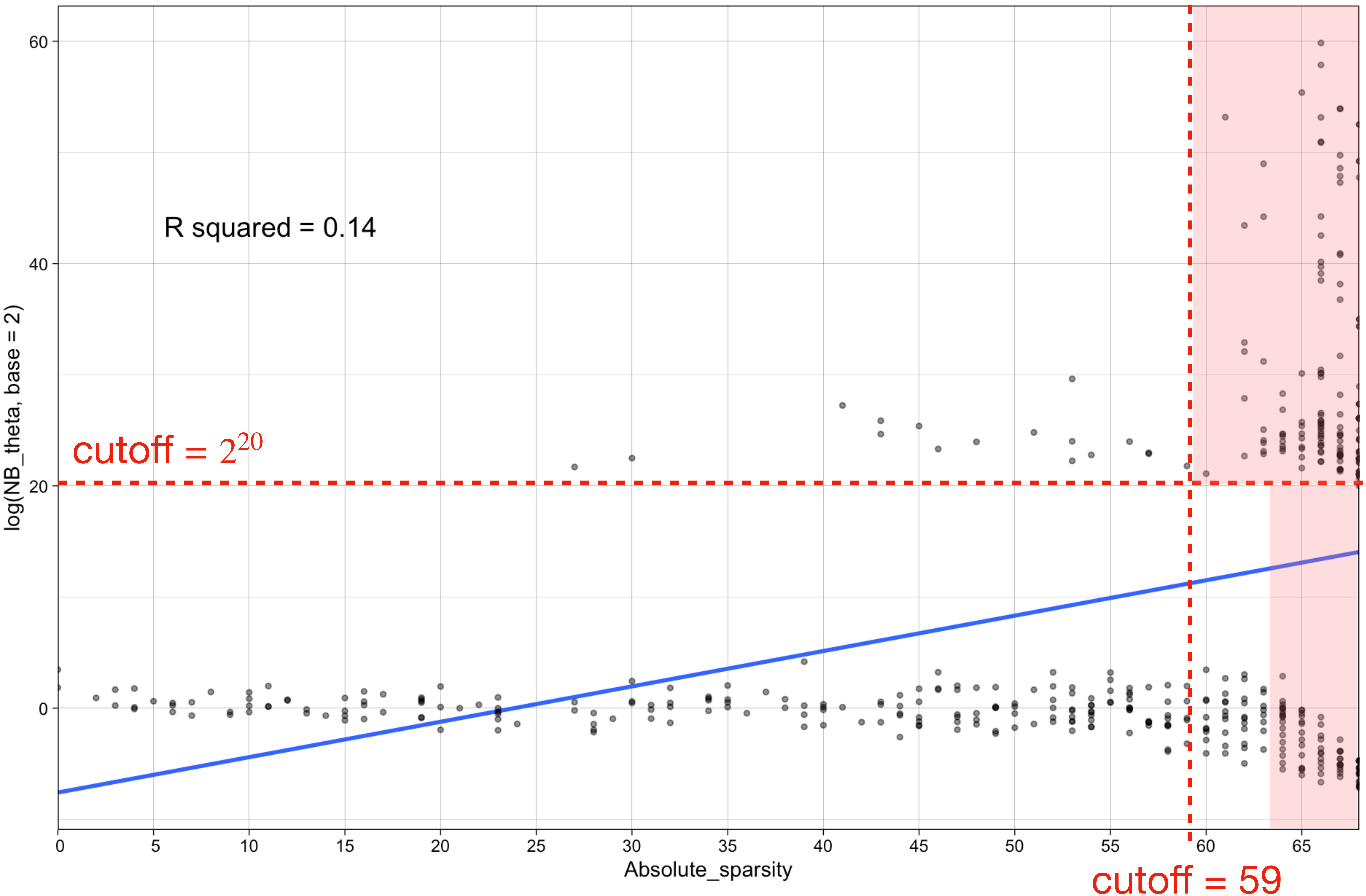
Theta threshold

Effect size	Absolute sparsity	Theta
0.609	11	2.9
0.565	15	3.0
0.478	11	1.8
0.304	28	0.5
0.0870	42	107827.6
0.0435	44	165287.6
0.000	44	2543.779

Results in extremely low, doubtful p values from models (1e-10, 1e-15…)

False positives with really low effect sizes

Absolute sparsity vs theta in CORONA bacteria data



R squared = 0.14

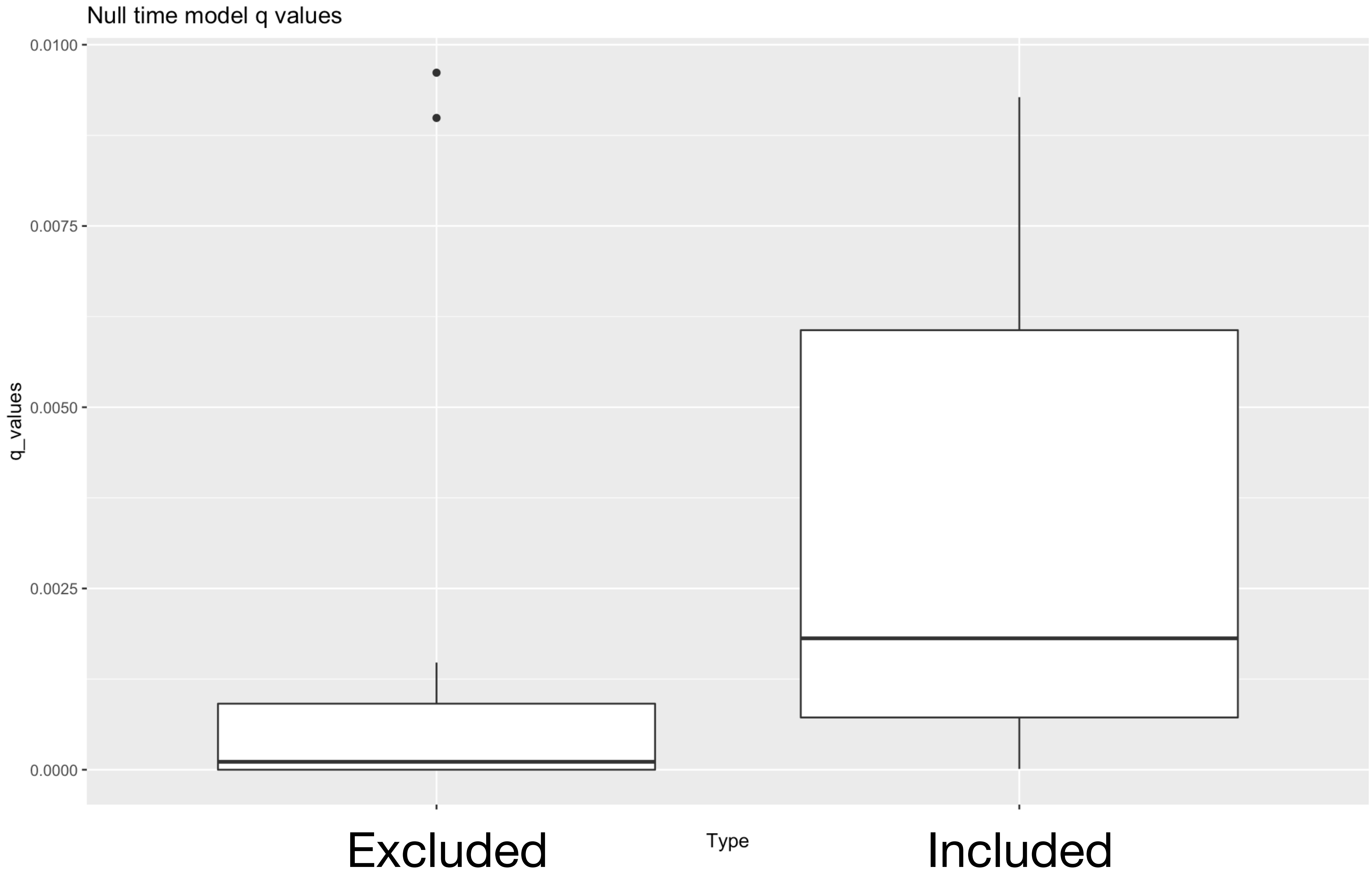
cutoff = 2^{20}

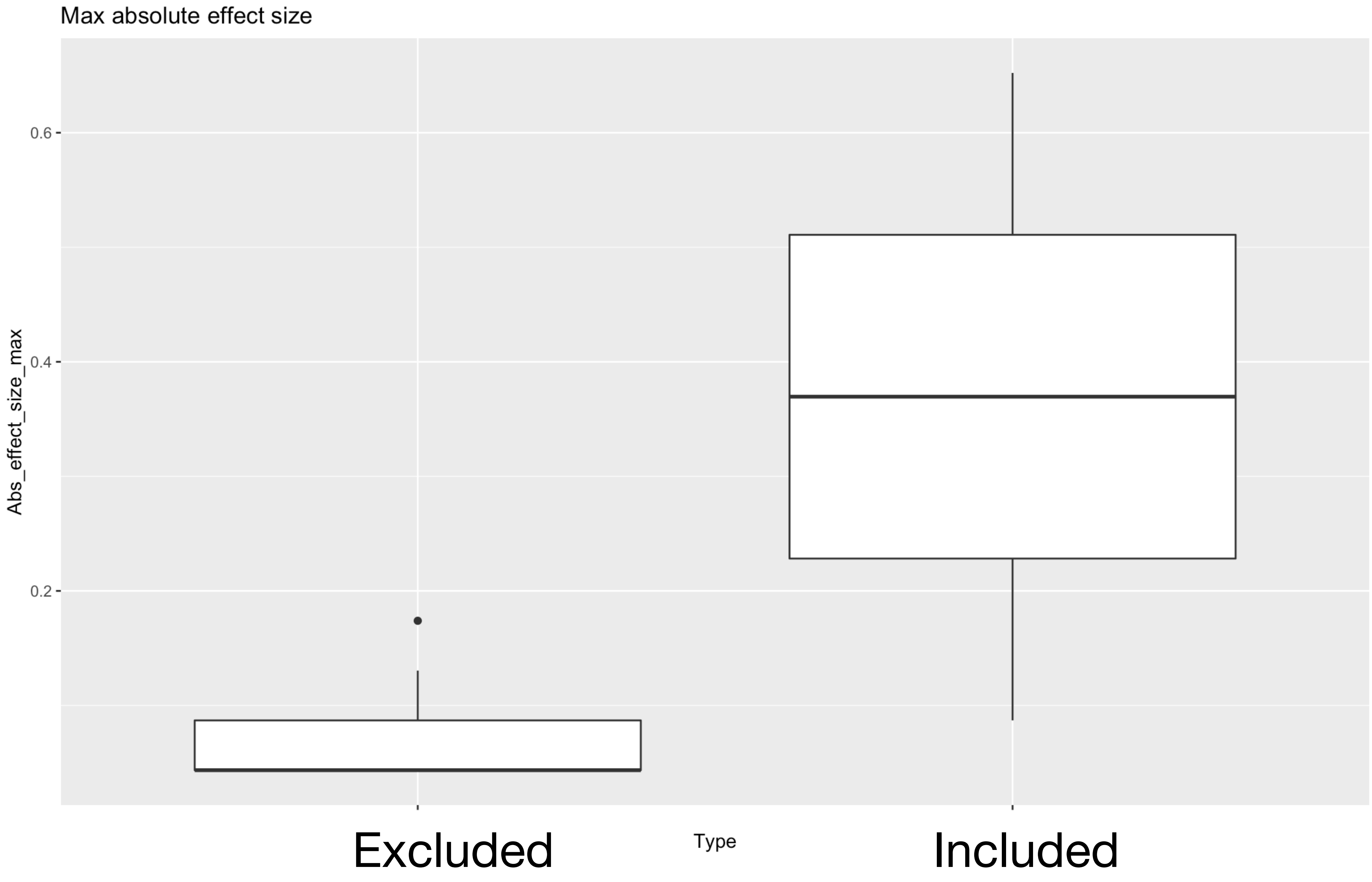
cutoff = 59

Negative binomial won't be fitted to the samples in the red area.

Theta threshold

The ones showing significance in negative binomial model





The extremely low p values are excluded

The really low effect sizes are excluded

Data types

Quantitative	Count	• Non-negative integers resulted from counting • Discrete	• 10 apples • 80 dogs

Data types

Quantitative	Measurement	• Can be measured at finer and finer scale • Continuous	• 1.6 g/ml • 9.5 cm

Measurement data

- Could be divided and reduced to finer and finer levels
- 1.5 kg ⇒ 1.52 kg
- Continuous
- Linear regression (not using generalized linear model!)
- Various types of data: height, weight, concentration, pressure……
- Normality not guaranteed
- Normalize first

Check distribution

- check_distribution( ) in the performance package
- Uses an internal random forest model to classify the distribution
- Possible distributions: bernoulli, beta-binomial, chi, exponential, F, gamma, lognormal, normal, negative binomial, poisson, ……

Check distribution

Metabolite data distribution

	Distribution
serotonin	lognormal
3-hydroxykynurenine	F
5-hydroxytryptophan	uniform
indole-3-propionic	lognormal
indolelactate	lognormal
indoxylsulfate	lognormal
kynurenic	weibull
Kynurenine	lognormal
tryptamin	weibull
tryptophan	lognormal
2-Aminophenol	weibull
3-Hydroxyanthranilate	chi
Melatonin	weibull
Methyltryptamine	weibull

Phenotype data distribution

	Distribution
HDL	beta-binomial
BMI	gamma
LDL	beta-binomial
RR_syst_mobilograph	beta-binomial
BP_sphygm_syst	beta-binomial
BP_sphygm_diast	beta-binomial
weight	gamma
hip_circumference	beta-binomial
waist_circumference	beta-binomial
body_fat_ratio	chi
urate	gamma
creatinine	gamma
eGFR	gamma
cholesterol	beta-binomial

Normalize data

- bestNormalize( ) in the bestNormalize package
- Selects the best transformation method according to the “Pearson P/df”, a relatively interpretable goodness of fit test.
- If the data is close to a normal distribution, “Pearson P/df” will be close to 1.

Distribution before and after normalization

Metabolite data before/after normalization

	Original	Normalized
3-hydroxykynurenine	F	normal
5-hydroxytryptophan	uniform	normal
indole-3-propionic	lognormal	normal
indolelactate	lognormal	normal
indoxylsulfate	lognormal	normal
kynurenic	weibull	normal
Kynurenine	lognormal	normal
serotonin	lognormal	normal
tryptamin	weibull	normal
tryptophan	lognormal	normal
2-Aminophenol	weibull	normal
3-Hydroxyanthranilate	chi	normal
Melatonin	weibull	normal
Methyltryptamine	weibull	normal

Phenotypes data before/after normalization

	Original	Normalized
RR_syst_mobilograph	beta-binomial	normal
HDL	beta-binomial	normal
BMI	gamma	normal
LDL	beta-binomial	normal
BP_sphygm_syst	beta-binomial	normal
BP_sphygm_diast	beta-binomial	normal
weight	gamma	normal
hip_circumference	beta-binomial	normal
waist_circumference	beta-binomial	normal
body_fat_ratio	chi	normal
urate	gamma	normal
creatinine	gamma	normal
eGFR	gamma	normal
cholesterol	beta-binomial	normal

Measurement data

Individual	Time point	Concentration (mg/l)
a	1	1.53
a	2	3.65
a	3	0.98
b	1	0.24
b	2	5.67
b	3	1.20
c	1	9.45
c	2	3.45
c	3	0.52

```
normalized_conc <- bestNormalize(Concentration)

m <- lmer( normalized_conc ~ (1|Individual) + Time_point, REML = F)

Anova(m, test.statistic=c("Chisq"))
```

Data types

Quantitative	Measurement	• Can be measured at finer and finer scale • Continuous	• 1.6 g/ml • 9.5 cm

Data types

Quantitative	Proportion	• Ranges from 0 to 1	• 25% classified as A • 10% classified as B

Proportion data

- Observations from 0 ~ 1
- Percentage of mortality
- Infection rates of diseases

Model for proportion data: beta regression

- Beta regression models continuous variables y that assume values in the interval (0,1)
- Beta distribution formula:

$$Pr\{Y = y\} = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} y^{p-1}(1-y)^{q-1}$$

p, q > 0, shape parameters
0 < y < 1

- Variance:

$$var(Y) = \frac{\mu(1-\mu)}{(1+\Phi)}$$

ϕ = dispersion parameter

Models for proportion data: beta regression

Individual	Time point	ratio of CD27+ % of CXCR3- Th17
a	1	0.42
a	2	0.36
a	3	0.97
b	1	0.12
b	2	0.20
b	3	0.98
c	1	0.39
c	2	0.41
c	3	0.68

```
m <- glmmTMB(ratio ~ (1|Individual) + Time_point, family = beta, REML = F)

Anova(m, test.statistic=c("Chisq"))
```

Data types

Quantitative	Proportion	• Ranges from 0 to 1	• 25% classified as A • 10% classified as B

Data types

Qualitative	Binary	• Sort things into one of two mutually exclusive categories	• True/False • Reject/Accept

Binary data

- Sort things into one of two mutually exclusive categories
- True/False
- Accept/Reject
- Passed/Failed

Model for binary data: binary logistic regression

- The distribution of y is assumed to be binomial
- Binomial distribution formula:

$$Pr\{Y = y\} = \binom{n}{y} p^y (1-p)^{n-y}$$

n Bernoulli trials
p the probability to succeed

- Mean and variance:

$$E(Y) = np$$
$$var(Y) = np(1-p)$$

Models for binary data: binary logistic regression

Individual	Time point	Survived
a	1	1
a	2	1
a	3	0
b	1	1
b	2	1
b	3	1
c	1	1
c	2	1
c	3	0

```
m <- glmmTMB(survived ~ (1|Individual) + Time_point, family = binomial, REML = F)

Anova(m, test.statistic=c("Chisq"))
```

Data types

Qualitative	Binary	• Sort things into one of two mutually exclusive categories	• True/False • Reject/Accept

Data types

Qualitative	Ordinal	• Ranked • The distance between two categories is not known	• Small/Medium/Large • Dislike/Neutral/Like

Ordinal data

- The variables have ordered categories and the distances between the categories is not known
- Satisfaction level on a scale of satisfied/indifferent/dissatisfied
- Pain level on a scale of no/mild/moderate/severe pain

Model for ordinal data: proportional odds logistic model

- Extension of the binary logistic model
- Instead of applying the transformation to the response probabilities π_i , we apply it to the cumulative response
- Sum probabilities up to a threshold, making the whole range of ordinal categories binary at that threshold.
- The ordered response is

$$y = 1,2,...,J$$

- The associated probabilities are

$$\{\pi_1, \pi_2, \ldots, \pi_J\}$$

- Cumulative probability of a response less than or equal to j is

$$P(Y \leq J) = \pi_1 + \ldots + \pi_J$$

Model for ordinal data: ordinal regression

Individual	Time point	Disease severity
a	1	1
a	2	2
a	3	5
b	1	2
b	2	3
b	3	2
c	1	5
c	2	4
c	3	1

```
library(MASS)
m <- polr(Severity ~ (1|Individual) + Time_point, method=“logistic")
Anova(m, test.statistic=c("Chisq"))
```

Data types

Quantitative	Count	• Non-negative integers resulted from counting • Discrete	• 10 apples • 80 dogs
	Measurement	• Can be measured at finer and finer scale • Continuous	• 1.6 g/ml • 9.5 cm
	Proportion	• Ranges from 0 to 1	• 25% classified as A • 10% classified as B
Qualitative	Binary	• Sort things into one of two mutually exclusive categories	• True/False • Reject/Accept
	Ordinal	• Ranked • The distance between two categories is not known	• Small/Medium/Large • Dislike/Neutral/Like

Data type	Count	Measurement	Proportion	Binary	Ordinal
Description	Non-negative integers resulted from counting	Continuous data Can be measured at finer scale	Ranges from 0 to 1	Either 0 or 1	Ranks
Example	Bacterial abundance	Height, weight, blood pressure	Immune cell: CD27+ % of CXCR3- Th17	Survived or not	Pain level
Model	**Negative binomial model**	**Normalize first, then apply linear model**	**Beta model**	**Binary logistic model**	**Proportional odds logistic model**

Longdat R package

- Longitudinal data analysis
- Takes longitudinal dataset as input
- Analyzes if there is significant change of the features over time
- The output table contains p values, effect sizes, confounders of features.
- Can handle the 5 types of data mentioned

1. longdat_disc( ): Time as discrete variable. V1, V2, V3…
2. longdat_cont( ): Time as continuous variable. Day1, day10, day20…
3. theta_plot( ): For count data, plots theta v.s. non-zero counts

longdat_disc(input, data_type, test_var, …)