

Modelos Lineares em Ecologia

Codificando e testando hipóteses usando modelos lineares

Versões desse curso já tem alguns anos

Biólogos tentando fazer outros biólogos sofrerem menos com estatística



Diogro, Diogo Melo (he/him)



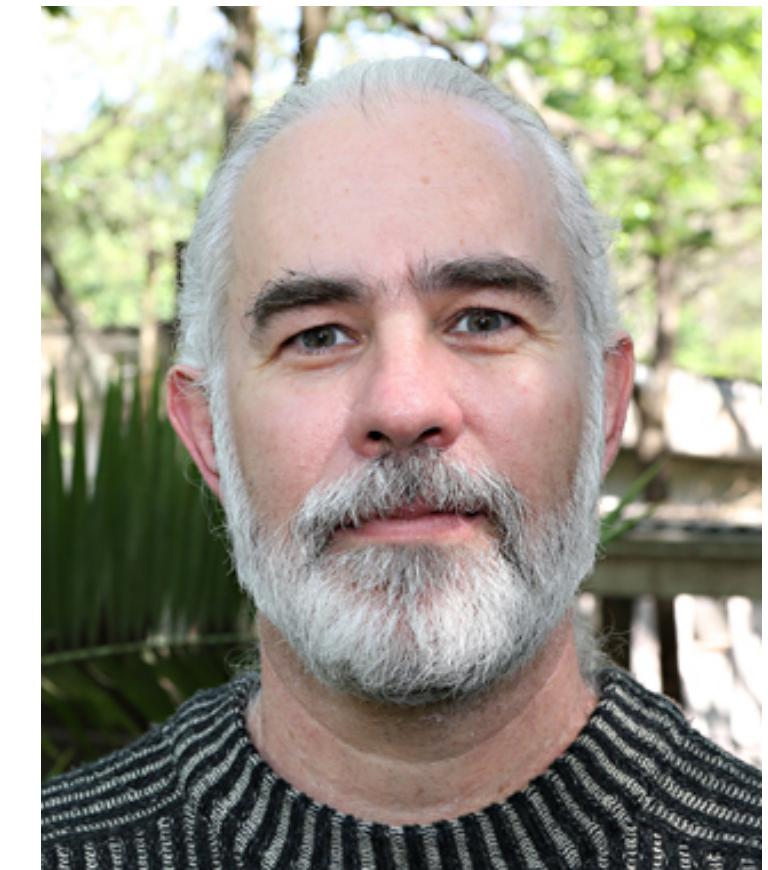
Sara, Sara Mortara (she/her)



Paulinha, Paula Lemos da Costa (she/her)



Andrea, Andrea Sanchez Tapia (she/her)



PI, Paulo Inácio Prado (he/him)

Conectando teoria e dados

Como abordar problemas ecológicos caracterizados por complexidade e incerteza?

- Use **modelos** para simplificar.
- **Teorias:** Resumir e entender processos.
- **Hipóteses:** Propor explicações
- **Dados:** Medidas informativas de sistemas reais
- **Modelos estatísticos:** Confrontar teorias e hipóteses com dados.



Ibā Huni Kuin, Nai Basa Masherī, 2014

Objetivos do curso

Poucas coisas que já são muitas

- Apresentar as bases de modelos lineares simples, com um foco em problemas típicos de ecologia
- Vamos usar uma filosofia genérica e bem fundamentada em perguntas científicas
- Introduzir as bases da inferência causal no contexto da criação de modelos e na abordagem de perguntas científicas
- Não traumatizar ninguém, e talvez tirar um pouco do ranço de estatística!

How to draw an Owl.

"A fun and creative guide for beginners"

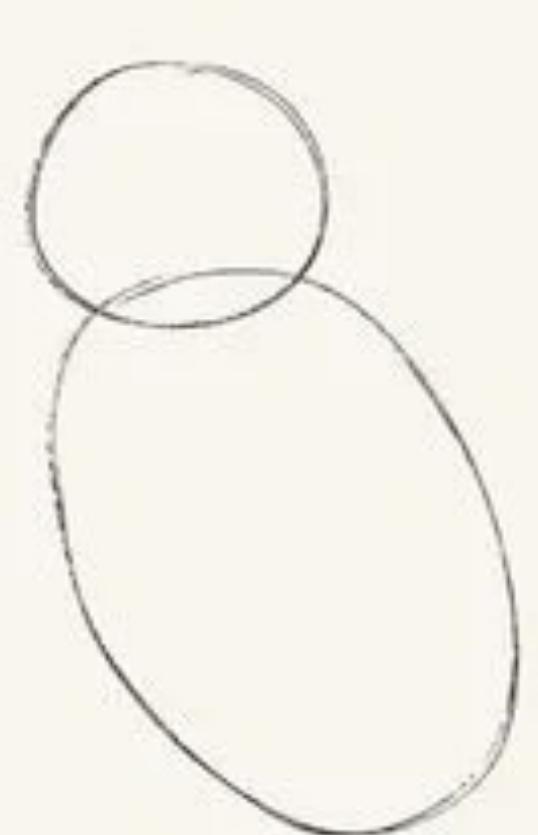


Fig 1. Draw two circles



Fig 2. Draw the rest of the damn Owl

Temas que não vamos abordar

Muita coisa, um mundo

- Modelos generalizados:
 - Regressão binomial, Poisson...
 - Usados para dados não contínuos
- Modelos mistos
 - Estruturas de variância mais complicadas
 - Desenhos experimentais complexos,
 - Observações correlacionadas
- Modelos Bayesianos (preferia, mas não dá tempo)
- Testes de hipótese específicos com nomes de russo
- Delineamento amostral e desenho experimental
- Testes não-paramétricos
- Mas podem perguntar tudo que quiserem!

Por que usar modelos probabilísticos?

Quais perguntas queremos responder?

- Modelos estatísticos devem responder perguntas científicas:
 - Qual a relação entre duas variáveis?
 - Qual a diferença entre dois grupos?
 - Quais são as causas de variação em um conjunto de dados?
 - Qual o resultado esperado de uma intervenção?

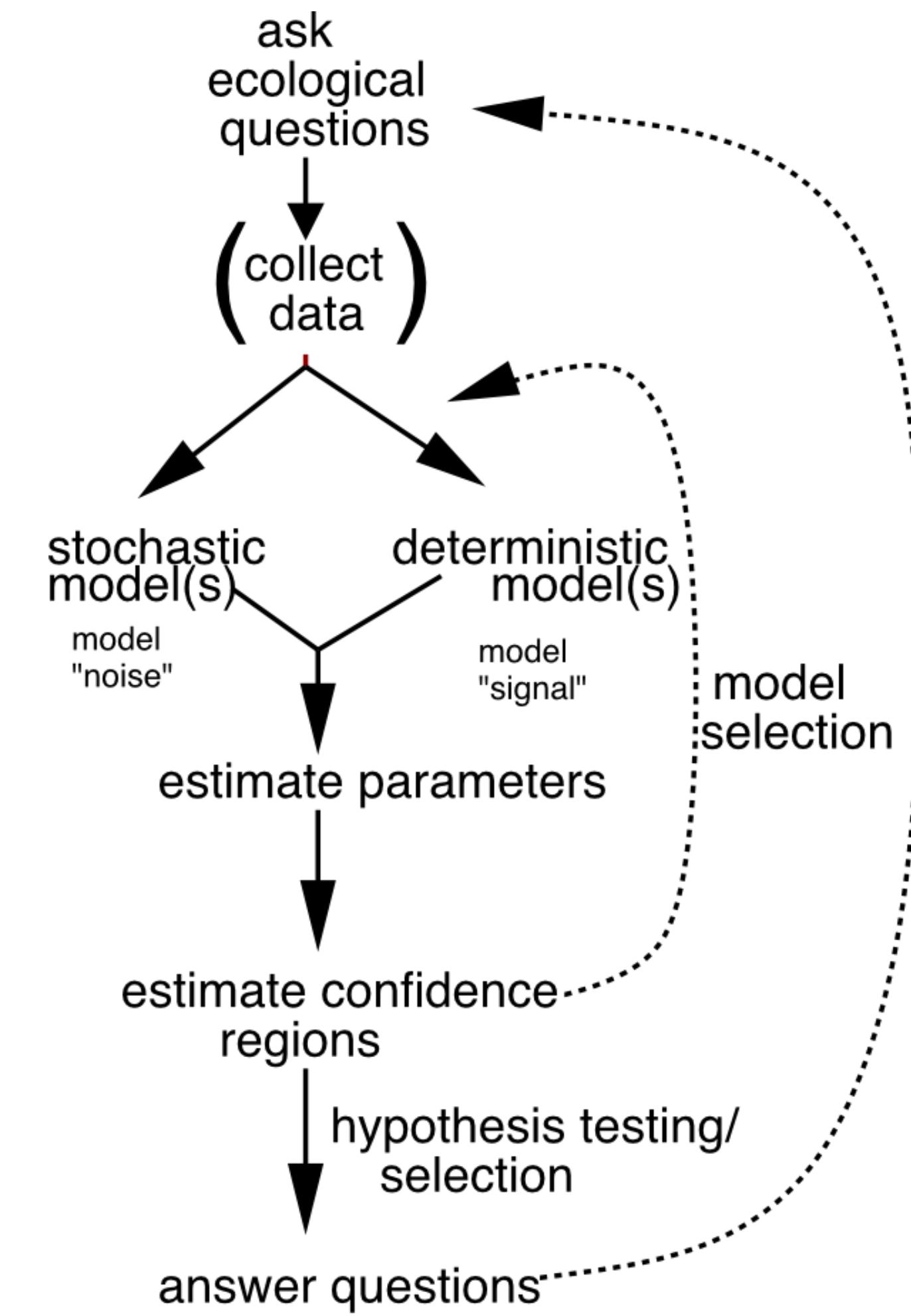


Figure 1.5 Flow of the modeling process.

Modelos Estatísticos e Teóricos

O modelo de estatístico é uma descrição de como **seus dados** devem se comportar de acordo com a teoria.

Portanto, o modelo estatístico é uma hipótese de como um **conjunto de dados** em particular foi gerado.

O modelo matemático descreve o comportamento de **quantidades teóricas**.

Já o modelo estatístico descreve o comportamento de **medidas** que são usadas como proxy para as variáveis teóricas.

O que queremos evitar?

Receitas enlatadas que não levam em conta as particularidades científicas da nossa pergunta.

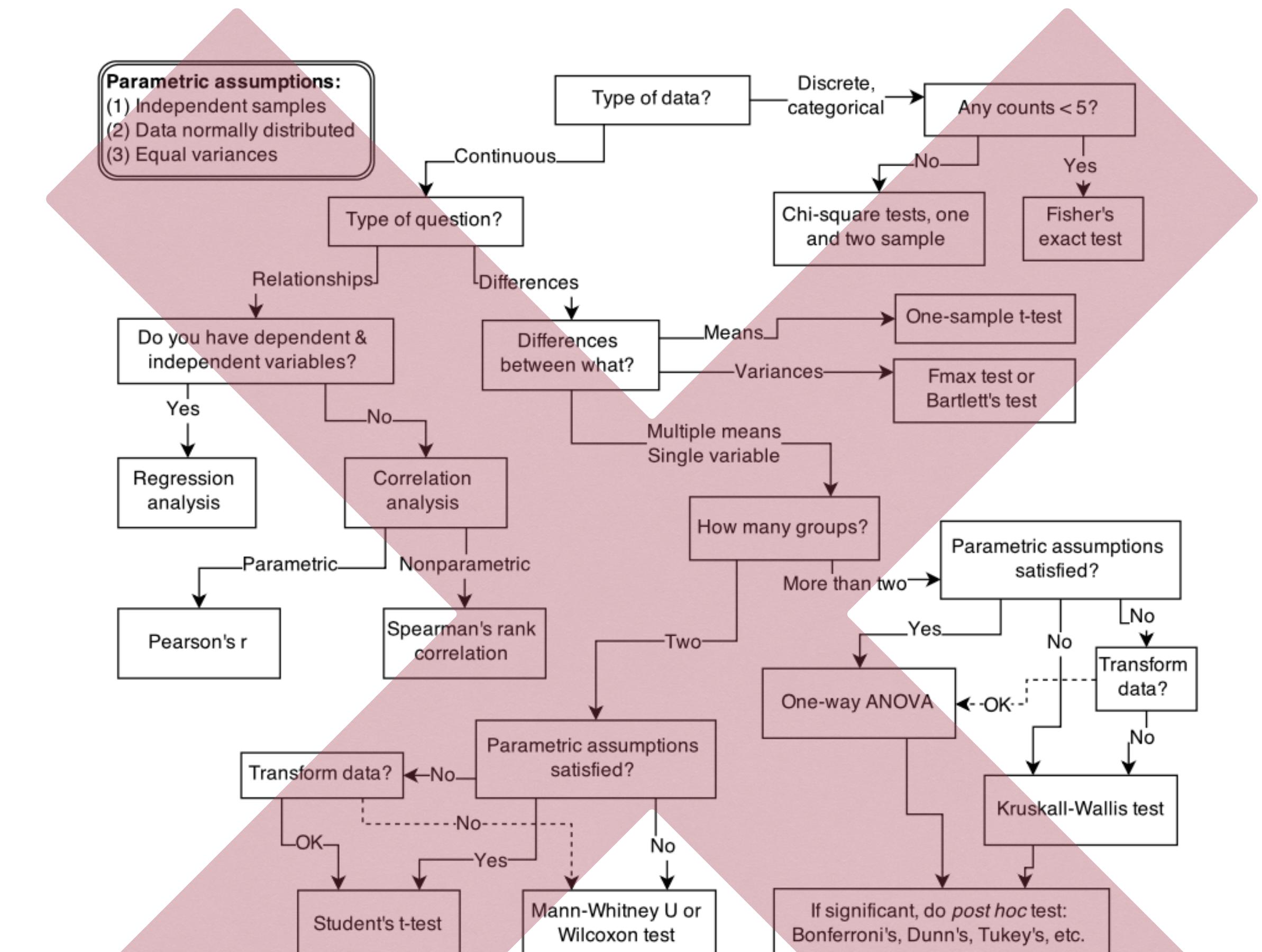


FIGURE 1.1. Example decision tree, or flowchart, for selecting an appropriate statistical procedure. Beginning at the top, the user answers a series of questions about measurement and intent, arriving eventually at the name of a procedure. Many such decision trees are possible.

Nosso modelo básico.

O martelo dos modelos lineares

- Nosso modelo deve:
 - Definir uma relação entre observações e parâmetros.
 - Criar uma descrição probabilística do modelo que estamos estudando.
 - Essa descrição deve capturar algum aspecto dos dados que estamos interessados em estudar.

Construindo modelos estatísticos

Hipótese geradora dos dados



Modelo estatístico



Estimar parâmetros



Verificar o ajuste
aos dados



Algumas definições em probabilidade

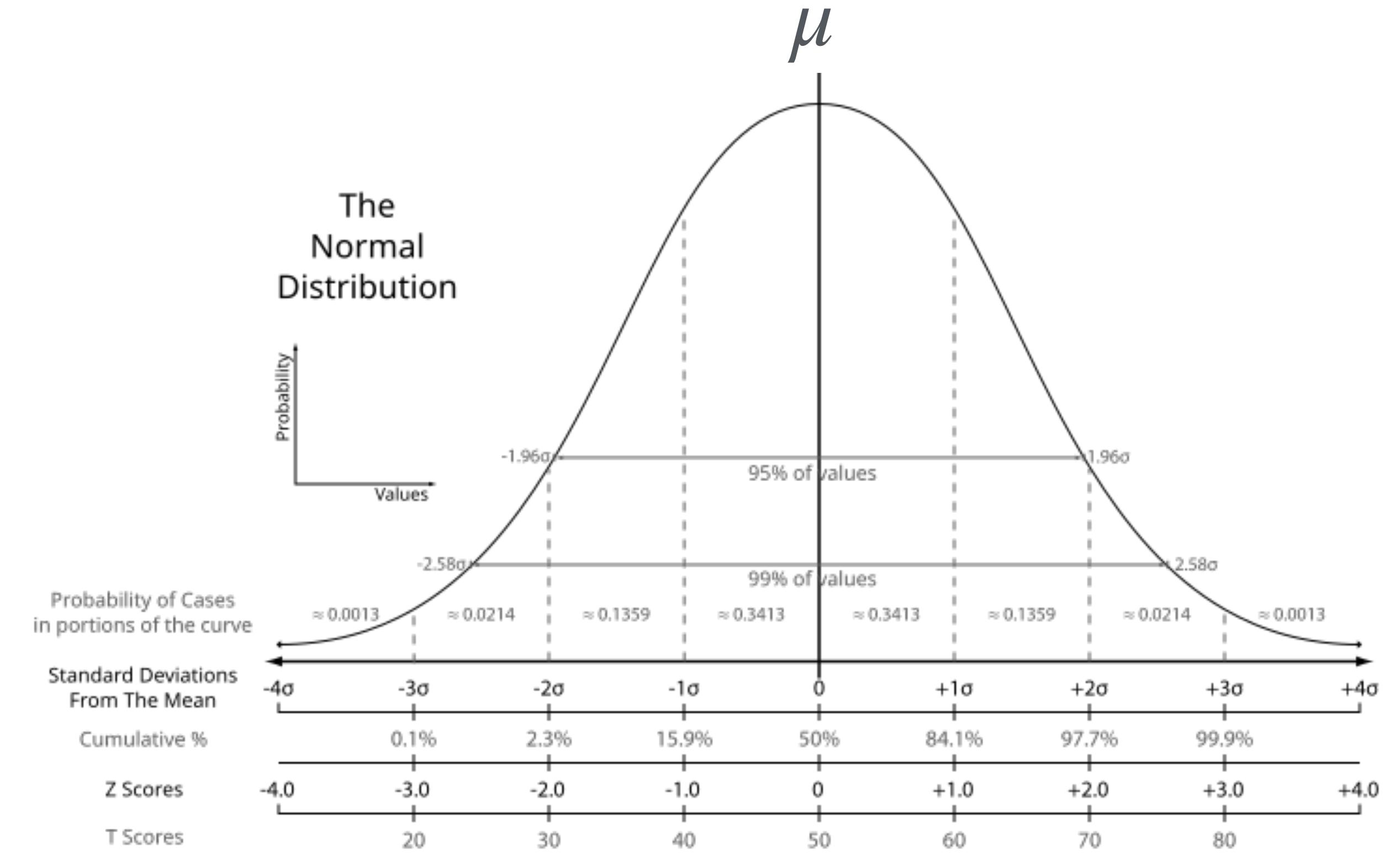
Crash course

- Podemos expressar a probabilidade de uma variável aleatória como:
 - $P(y)$: Leia como "**a probabilidade de y**"
 - Se a distribuição de uma variável depende de outras variáveis, podemos usar a probabilidade condicional:
 - $P(y|x)$: Leia como "**a probabilidade de y dado x**"

Distribuição de Probabilidade

Crash course

- Podemos usar distribuições de probabilidade padrões para descrever a relação entre variáveis
- Se a variável y segue uma distribuição normal, temos:
 - $P(y) = P(y | \mu, \sigma) = \text{Normal}(y | \mu, \sigma)$
 - Onde μ e σ são **parâmetros**
 - μ : mu, é a média, um parâmetro de localidade
 - σ : sigma, é o desvio padrão, um parâmetro de escala ou variação

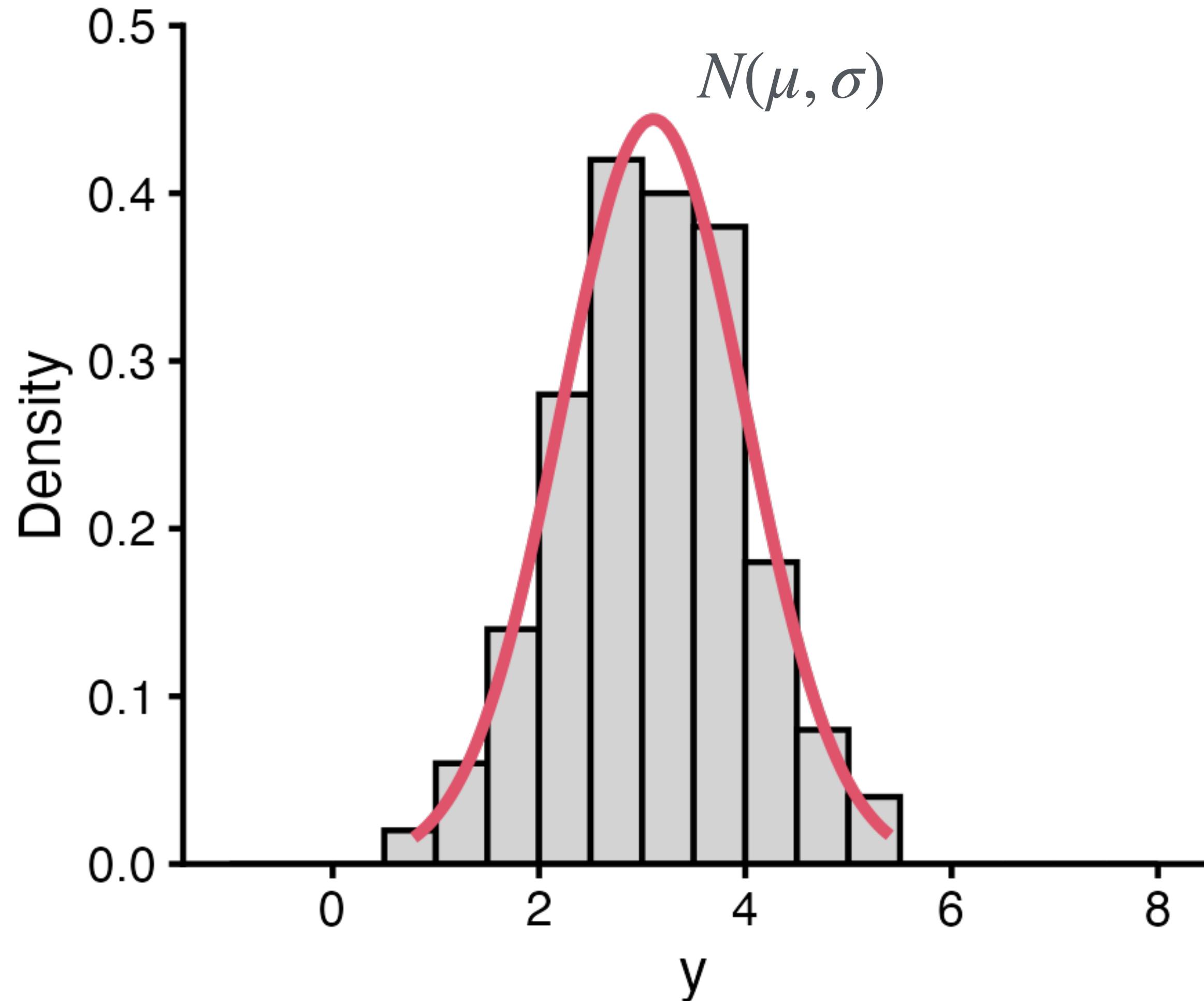


Modelo probabilístico mais simples

Encontrar a média e a variância para um conjunto de observações.

- Medir um conjunto de valores y_i :
- Achar a distribuição normal que melhor se ajusta aos dados encontrando os valores de μ e σ tal que a distribuição $N(\mu, \sigma)$ aproxime o histograma dos valores y_i :

$$y_i \sim N(\mu, \sigma)$$



A verossimilhança

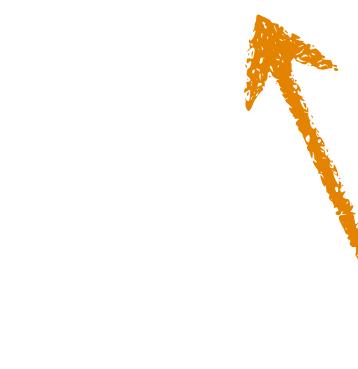
A probabilidade de observar cada valor de y

- O que isso significa?

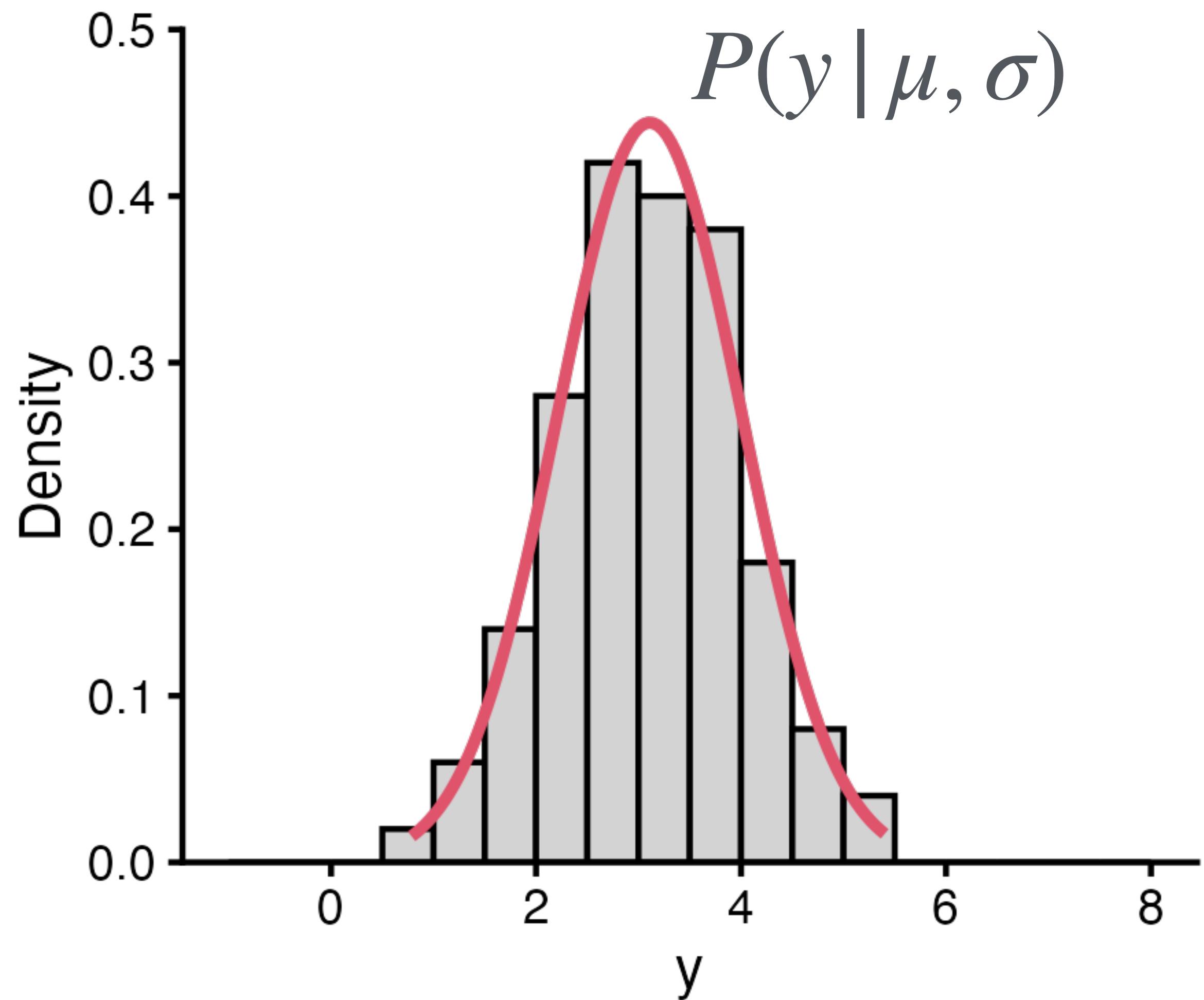
$$y_i \sim N(\mu, \sigma)$$

- Podemos escrever a mesma coisa com a expressão:

$$P(y | \mu, \sigma)$$



A verossimilhança de y



O modelo linear

Adicionando mais variáveis.

- A principal estratégia para fazer modelos probabilísticos úteis é **permitir que os parâmetros variem.**
- Se tivermos duas variáveis medidas:
 - y_i : variável dependente, a resposta, A variável a ser predita.
 - x_i : A variável independente, o tratamento, a variável controle, o preditor.

- Nós definimos o parâmetro μ (média), como sendo uma função linear da variável preditora:

$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

- Isso substitui a média por dois novos parâmetros:
 - α : o **intercepto**. O valor previsto de y quando x for zero.
 - β : a a **inclinação** da variável x .

Premissas do modelo linear

Eu sou obrigado a falar disso, mas eu acho bobagem.

- Modelos lineares fazem algumas suposições sobre os dados que estamos modelando.
- Desvios dessas premissas não são fatais para o modelo, mas entender o que cada uma delas implica nos ajuda a interpretar os desvios.

Premissas do modelo linear

Tres de cinco

1. Relação linear entre a resposta y e a variável preditora x :

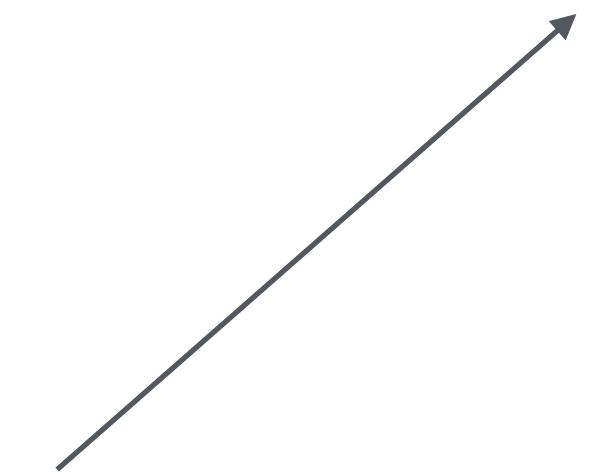
$$y_i = \alpha + \beta x_i + u_i$$



2. A amostra $[y_i, x_i]$ é uma amostra aleatória da população.



3. Os erros (u_i) tem média zero quando condicionados aos preditores x



$$E(u | x) = 0$$

**Estimativas OLS $\hat{\alpha}$ and $\hat{\beta}$ são
não-enviesadas**

Premissas do modelo linear

Mais duas...

1. Relação linear entre y e x :

2. A amostra $[y_i, x_i]$ é aleatória

3. $E(y|x) = \alpha + \beta x$

4. Homoscedasticidade, variância residual é constante: $Var(y|x) = \sigma^2$

5. A resposta y tem distribuição normal dado o preditor: $y \sim N(\alpha + \beta x, \sigma)$



**Estimativas OLS $\hat{\alpha}$ and $\hat{\beta}$ são
não-enviesadas**



**Estimativas OLS de
 σ^2 é não-enviesada**

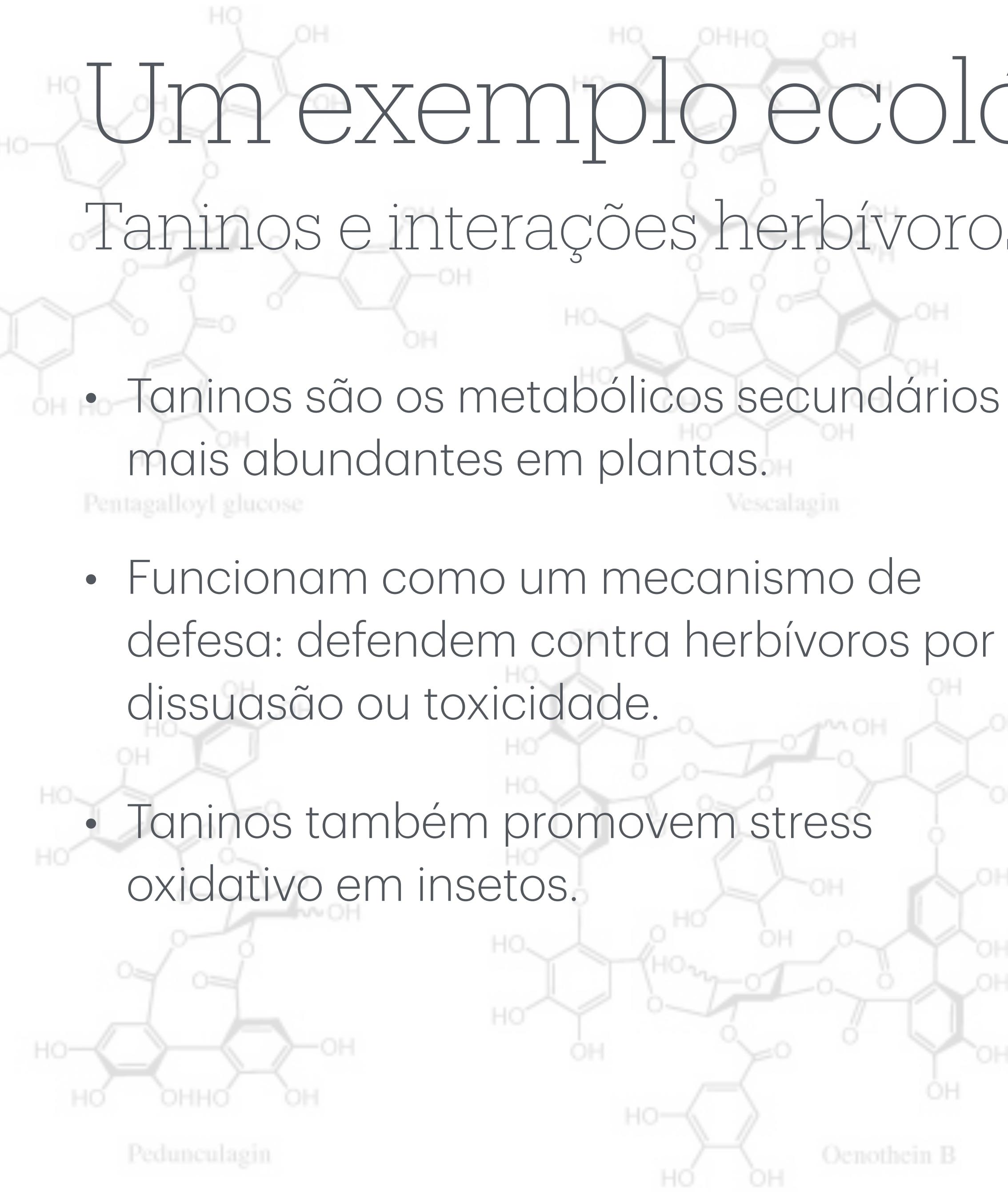
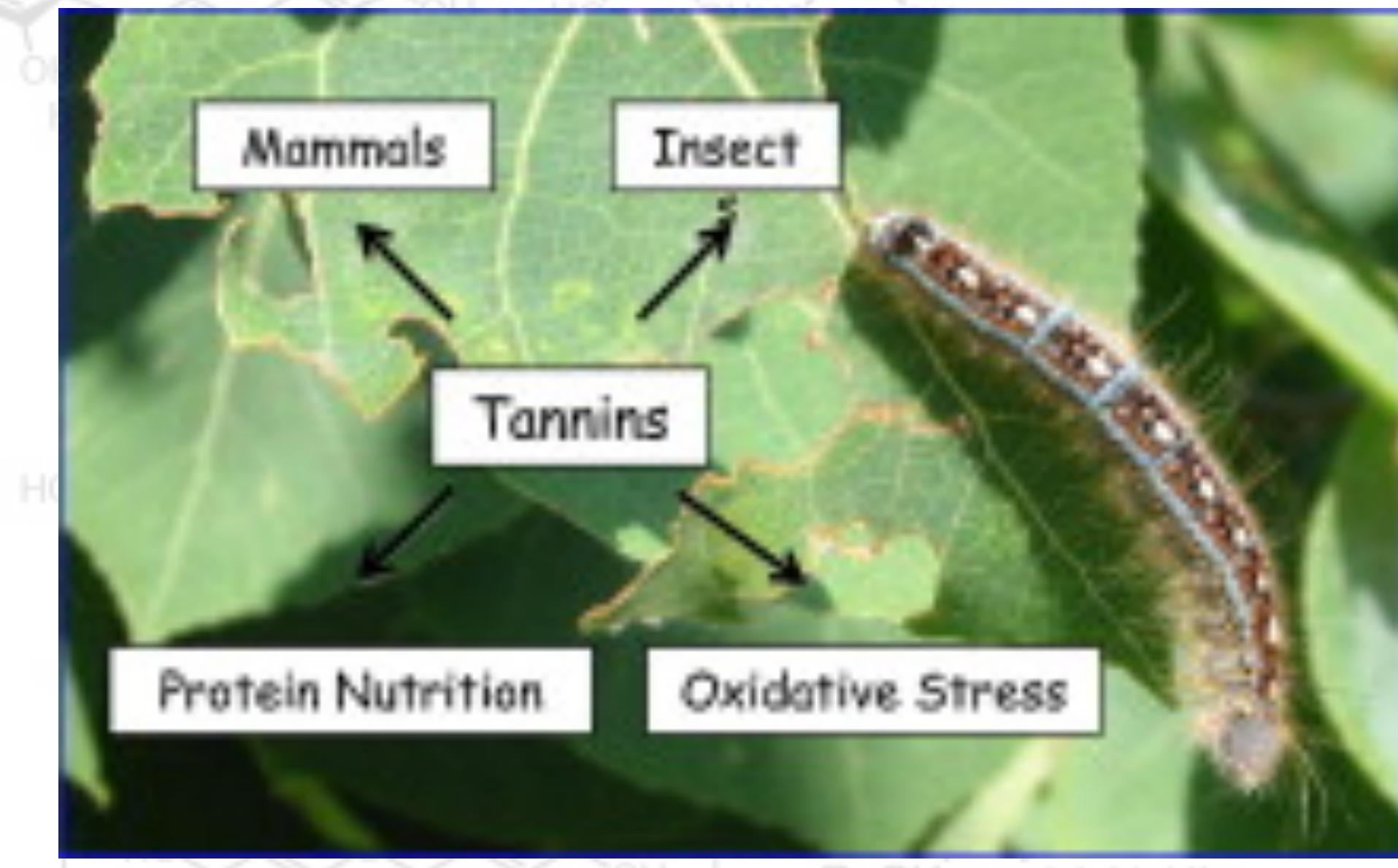


**$\hat{\alpha}$ e $\hat{\beta}$ Tem distribuições
conhecidas e podemos
calcular p-valores**

Um exemplo ecológico

Taninos e interações herbívoros-planta

- Taninos são os metabólicos secundários mais abundantes em plantas.
- Funcionam como um mecanismo de defesa: defendem contra herbívoros por dissuasão ou toxicidade.
- Taninos também promovem stress oxidativo em insetos.



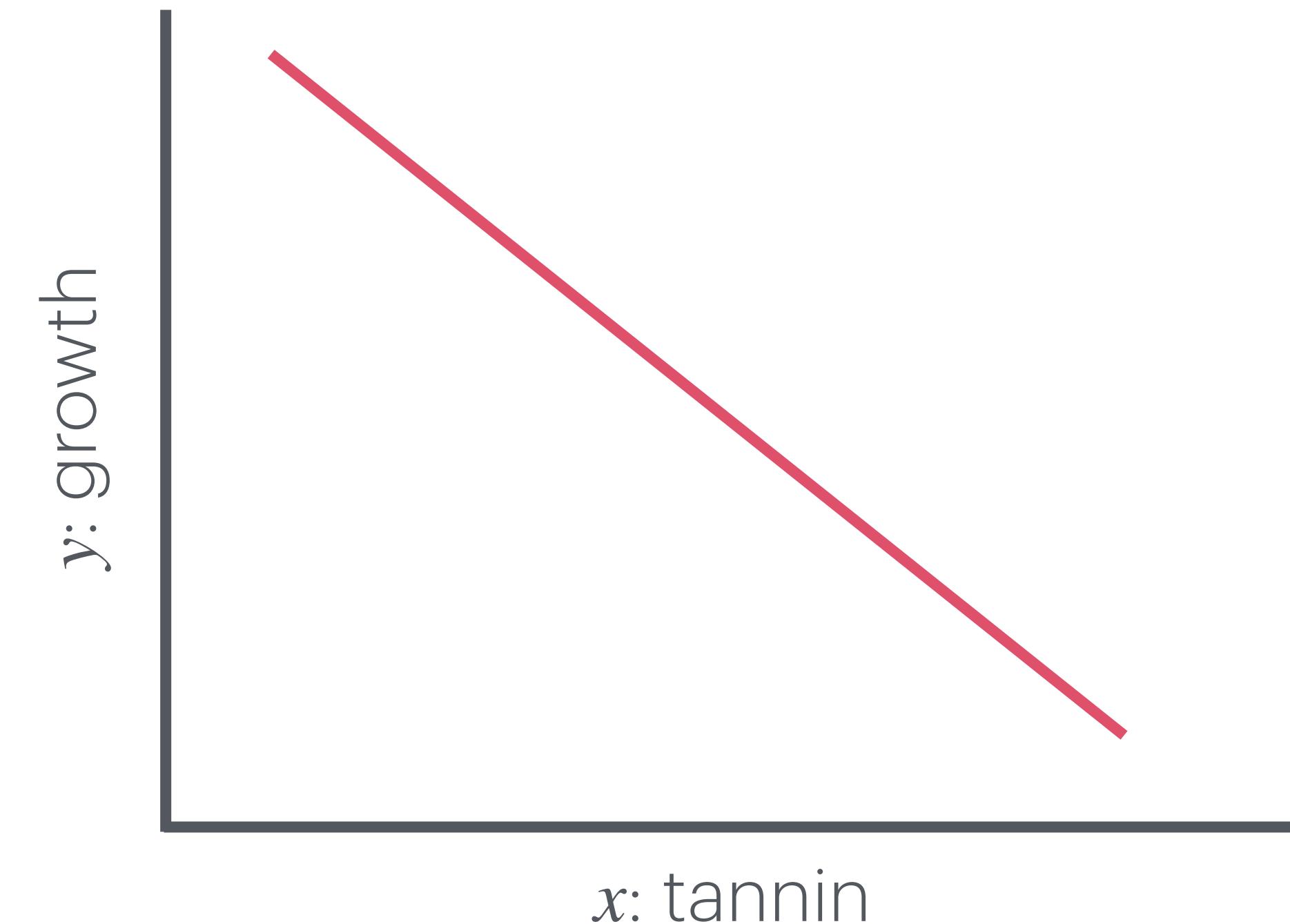
Nossas variáveis.

- y_i : **Crescimento de lagartas** - Variável resposta.
- x_i : **Quantidade de taninos na dieta das lagartas** - Variável preditora.



Nossas variáveis.

- y_i : **Crescimento de lagartas** - Variável resposta.
- x_i : **Quantidade de taninos na dieta das lagartas** - Variável preditora.



Articulando uma pergunta

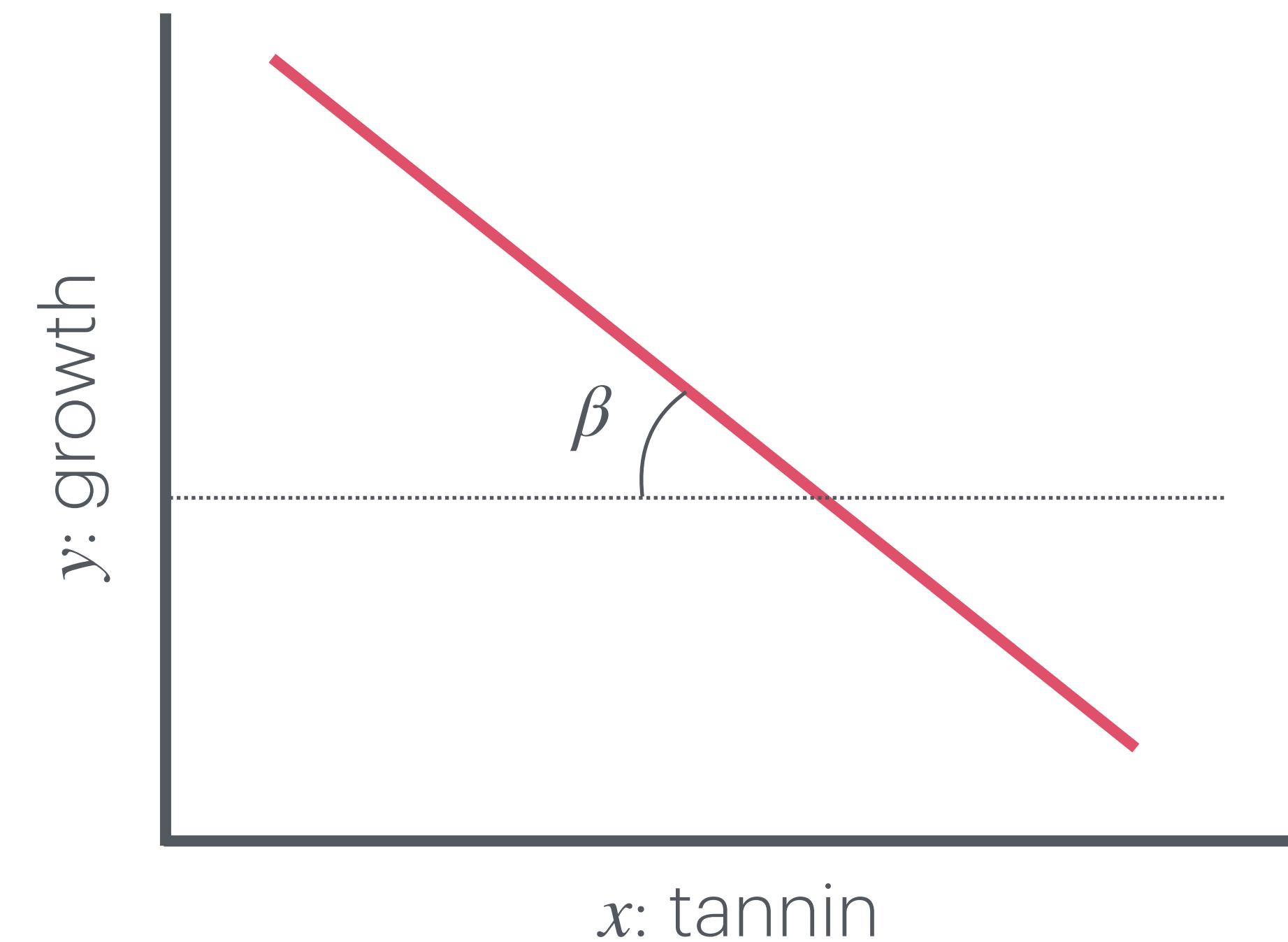
Os compostos químicos das folhas reduzem o crescimento das lagartas?

- Nosso modelo

$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

A relação entre o crescimento e a concentração de taninos é dada pela variável β



Modelo no computador

Centralizar ambas as variáveis é sempre uma boa ideia.

```
df <- data.frame(growth = c(12, 10, 8, 11, 6, 7, 2, 3, 3),  
                  tannin = c(0, 1, 2, 3, 4, 5, 6, 7, 8))  
df$tannin = scale(df$tannin, scale = FALSE)  
df$growth = scale(df$growth, scale = FALSE)  
  
fit = lm(growth ~ 1 + tannin, data = df)
```

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

Ajuste do modelo

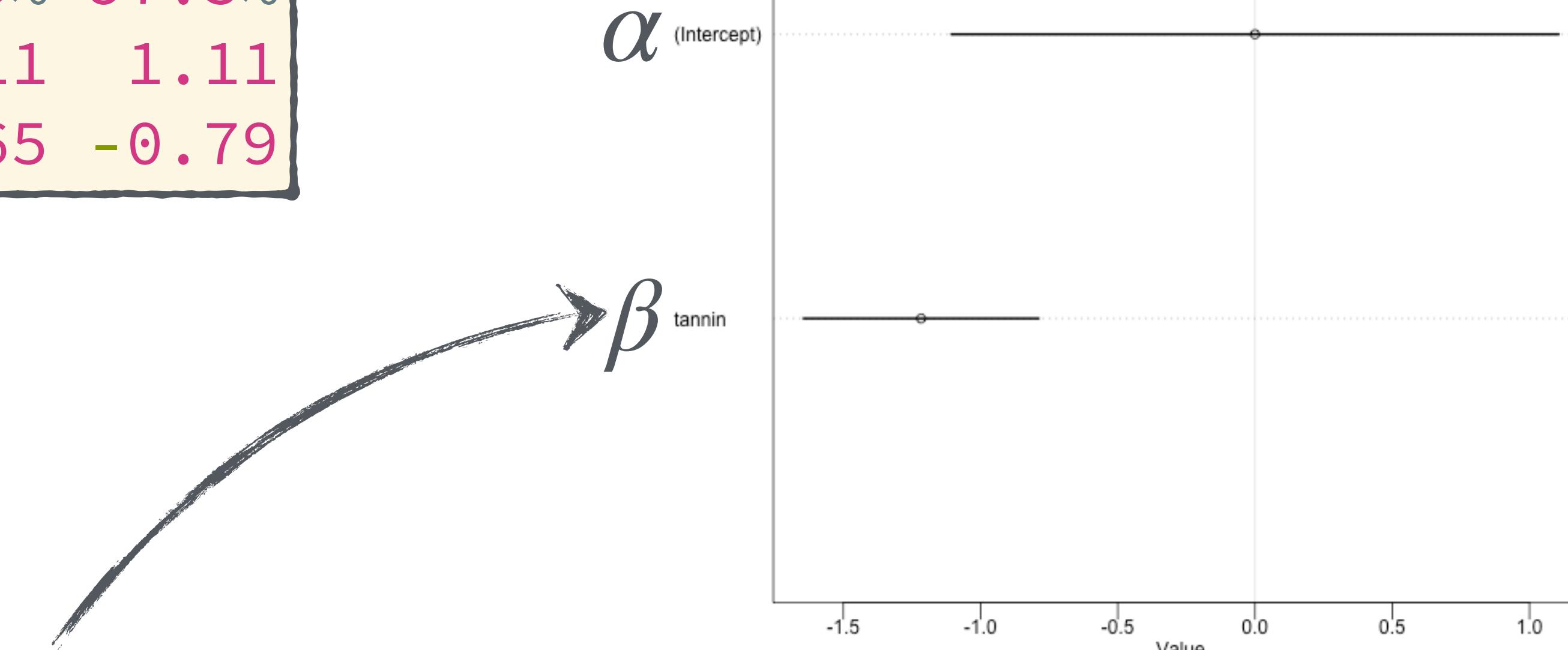
$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

```
> library(rethinking)
> precis(fit, prob = 0.95)
      mean    sd  2.5% 97.5%
α (Intercept) 0.00 0.56 -1.11  1.11
β tannin     -1.22 0.22 -1.65 -0.79
```

α
 β

Estimativas de α e β !

```
plot(precis(fit, prob = 0.95))
```

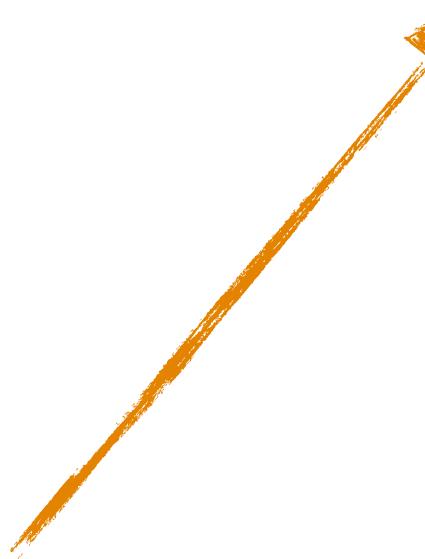


Ajuste do modelo

Summary funciona também, mas é pior

```
> summary(fit)$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.000000	0.5644526	0.000000	1.000000000000
tannin	-1.216667	0.2186115	-5.565427	0.0008460738



Estimativas!

```
> precis(fit, prob = 0.95)
```

	mean	sd	2.5%	97.5%
(Intercept)	0.00	0.56	-1.11	1.11
tannin	-1.22	0.22	-1.65	-0.79

Ajuste do modelo

Summary funciona também, na verdade é bem pior

```
> summary(fit)

Call:
lm(formula = growth ~ 1 + tannin, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4556 -0.8889 -0.2389  0.9778  2.8944 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.0000    0.5645   0.000 1.000000    
tannin      -1.2167    0.2186  -5.565 0.000846 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-squared:  0.8157,    Adjusted R-squared:  0.7893 
F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.0008461
```

Model plot

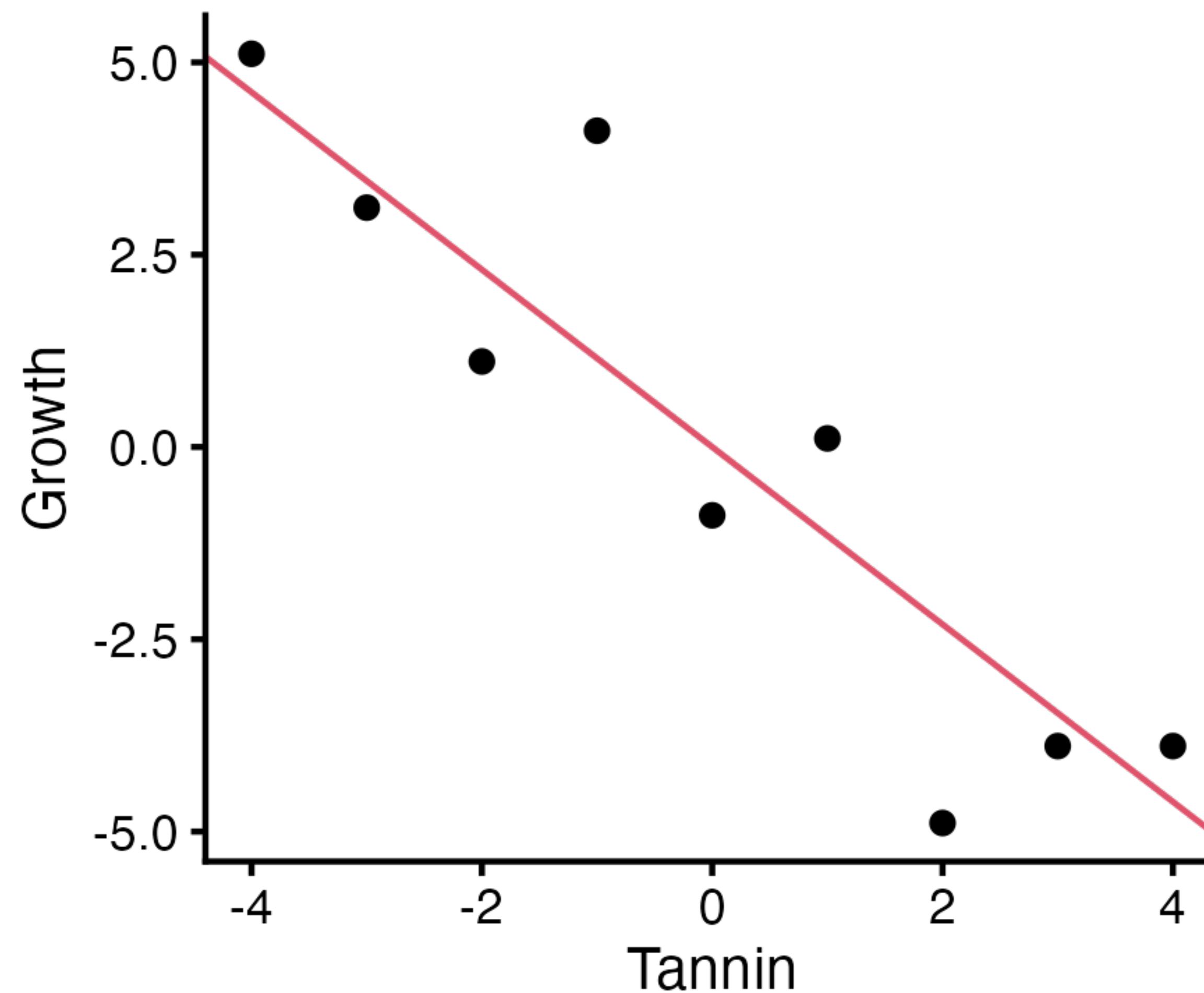
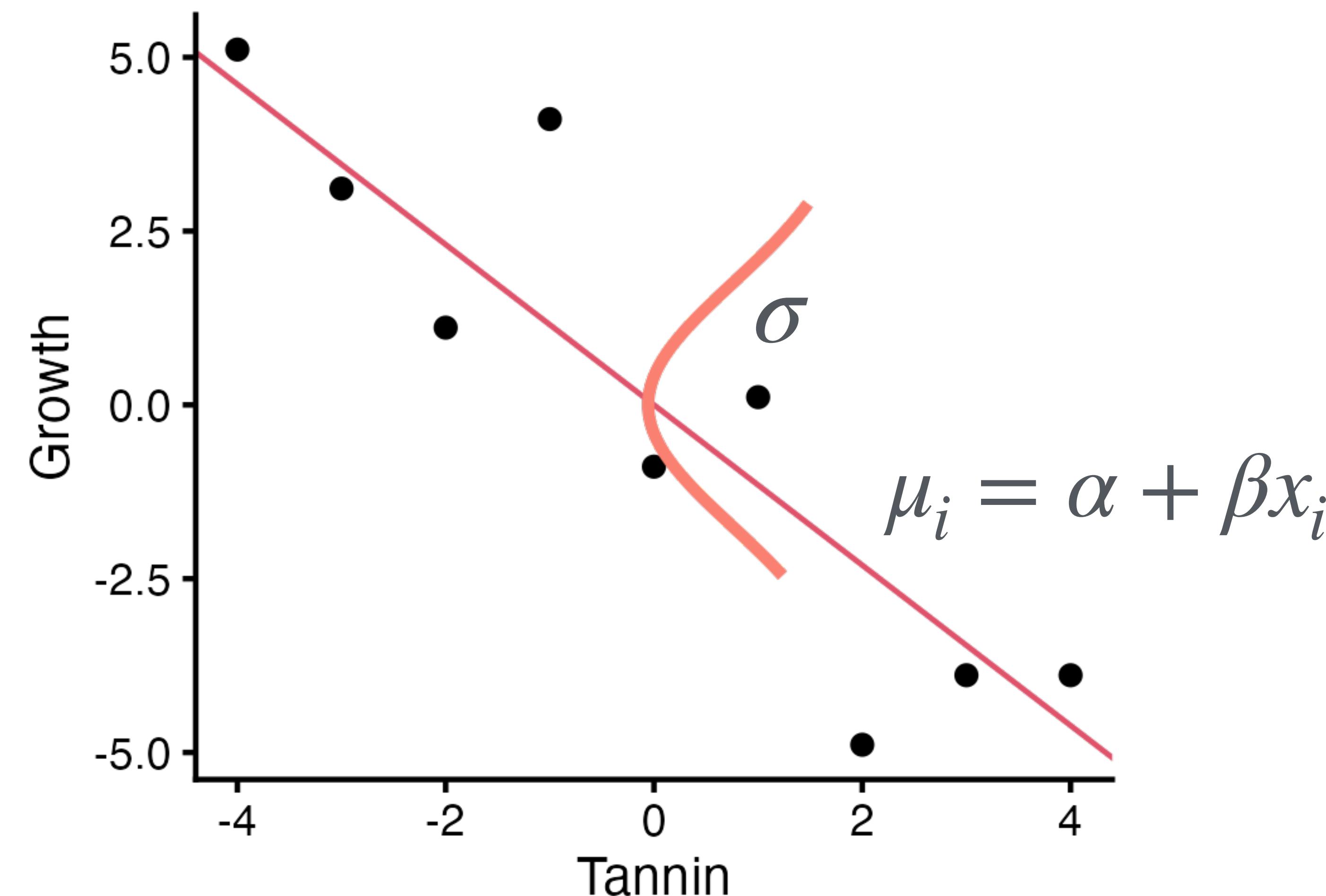


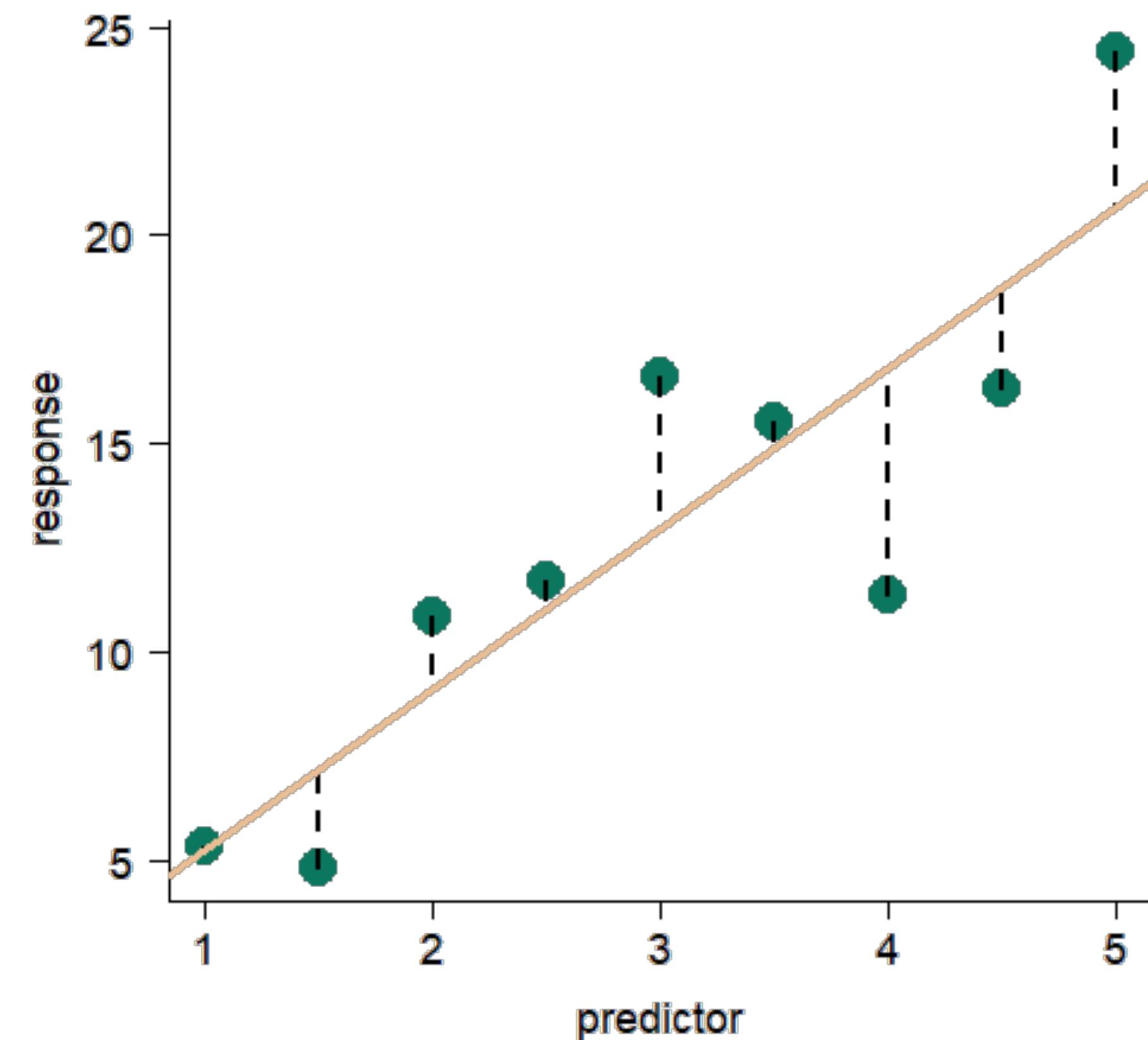
Gráfico do modelo



Como ajustar o modelo linear?

OLS e Maximum Likelihood

- O modelo linear simples tem muitas versões e muitas justificativas.
- Ordinary Least Squares (OLS), ou método de mínimos quadrados, é a introdução mais comum e consiste em minimizar o quadrado da distância entre as observações e a reta de regressão.
- Maximum likelihood (ML), a máxima verossimilhança, procura por parâmetros que maximizem a probabilidade de observar cada valor de y_i :
 - $P(y_i | \theta = \{\alpha, \beta, \sigma\})$
 - Tanto ML quanto OLS chegam na mesma solução no caso da regressão linear simples!



Usando a função lm() para tudo

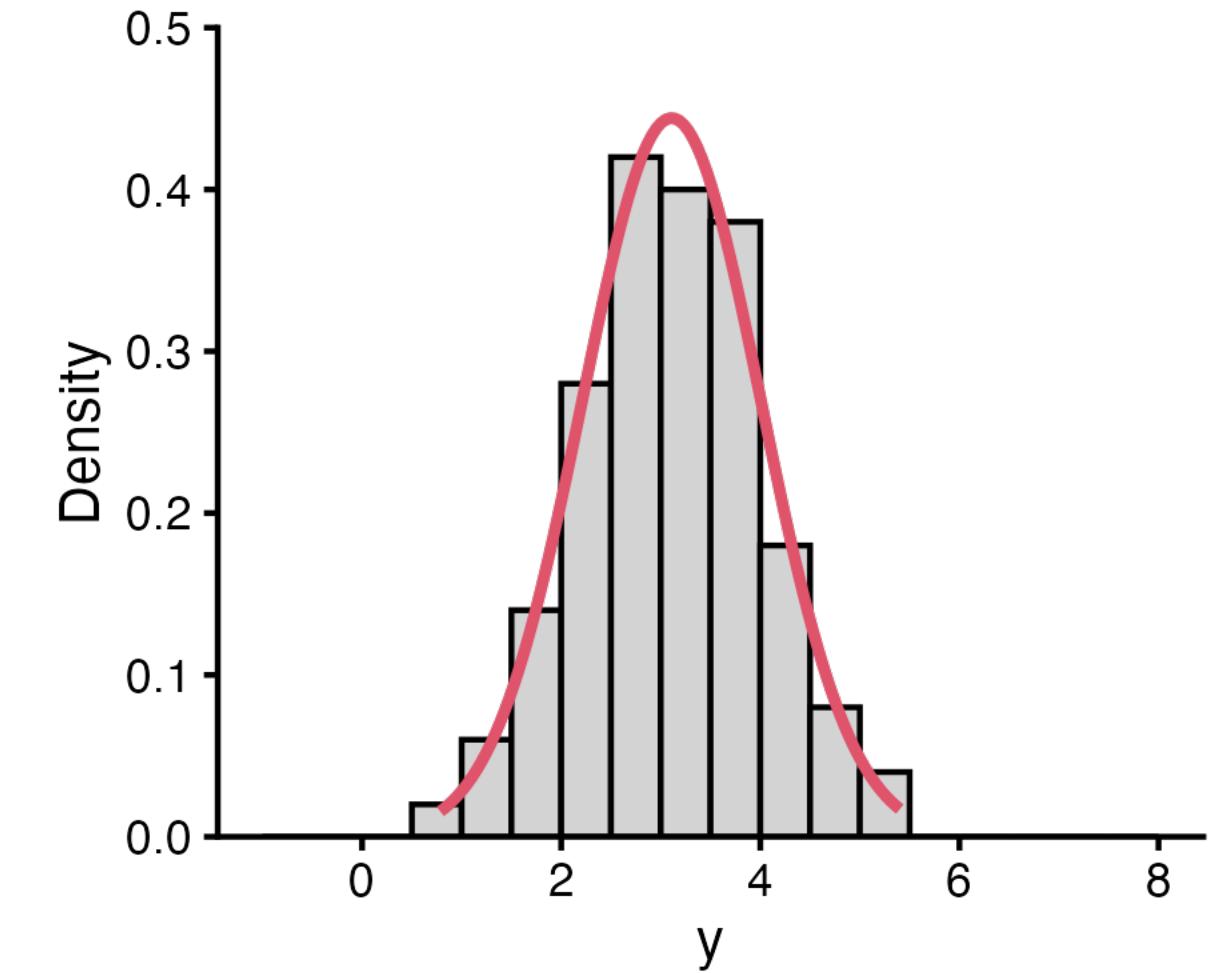
Exemplo, ajustando uma normal a um conjunto de observações

- A função lm() pode ajustar um grande conjunto de modelos usando OLS.
- R formulas:

- $y \sim 1$ (y como função de uma constante)

- $y \sim x$ (y em função de x, igual $1 + x$)

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha$$



```
> df <- data.frame(y = rnorm(100, 3, 1))

> ols_fit = lm(y ~ 1, data = df)

> precis(ols_fit, prob = 0.95)
               mean   sd 2.5% 97.5%
α (Intercept) 3.03 0.1 2.83 3.24
> (summary(ols_fit)$sigma)
σ [1] 1.037191
```

Outras formas de ajustar o mesmo modelo

stan_glm() para modelos lineares Bayesianos

$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

E priors
padrão...

```
> sglm_fit = stan_glm(growth ~ tannin, data = df, cores = 4)

> summary(sglm_fit, probs = c(0.025, 0.975))[, 1:7]
```

	mean	mcse	sd	2.5%	97.5%	n_eff
(Intercept) α	-0.01069275	0.015974907	0.6971944	-1.403377	1.3696905	1905
tannin β	-1.21608408	0.005609107	0.2482728	-1.716820	-0.7229236	1959
sigma	1.98129172	0.016244872	0.6447948	1.145132	3.5859302	1575
mean_PPD	-0.01273309	0.020249189	0.9958192	-2.030730	2.0076369	2418
log-posterior	-23.56539992	0.046443876	1.4480299	-27.365501	-21.9264227	972

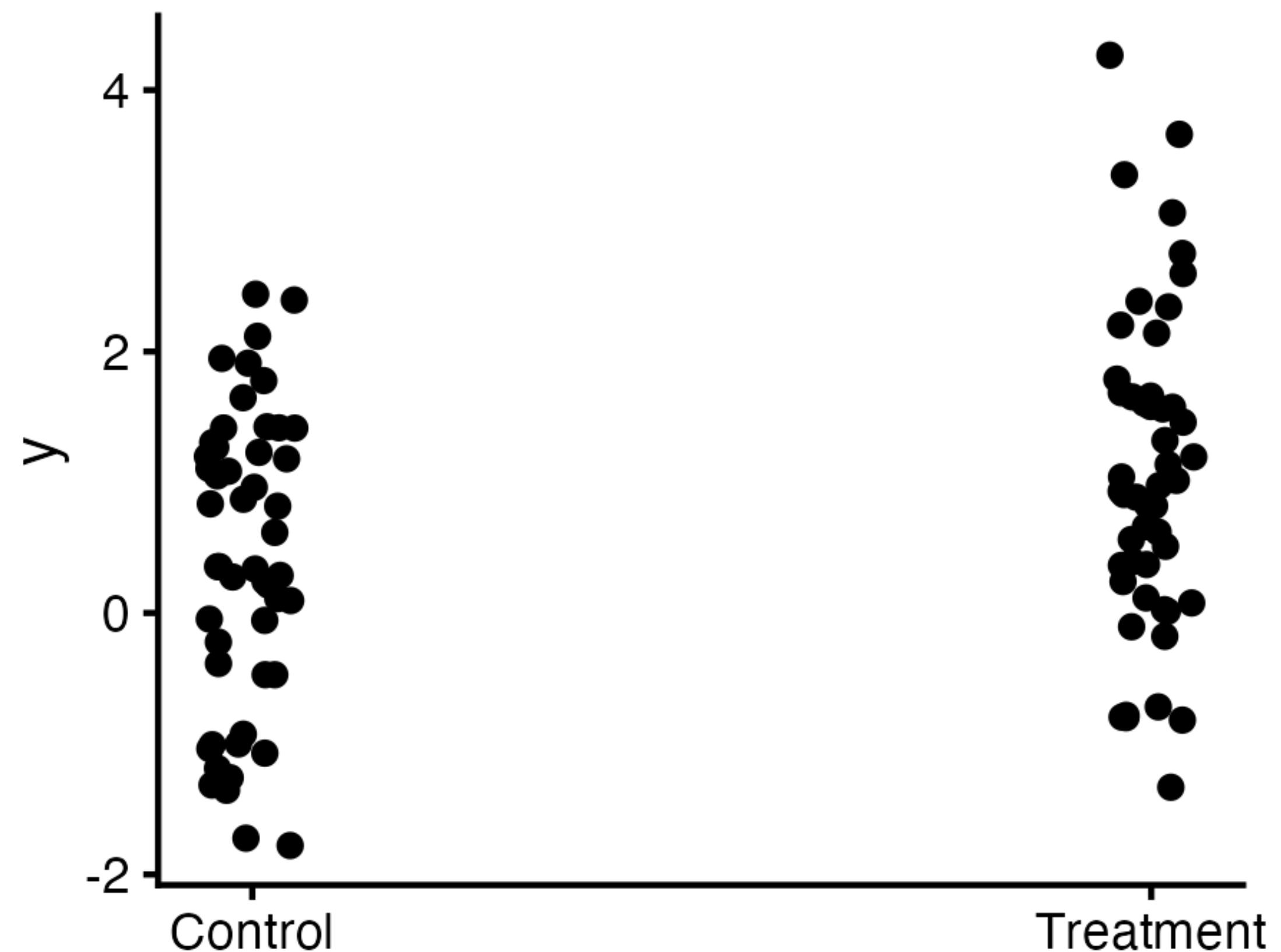
Rhat

(Intercept)	1.000676
tannin	1.001776
sigma	1.000847
mean_PPD	1.000271
log-posterior	1.003540

E variáveis preditoras categóricas?

Regressão linear é muito flexível.

- Nossas perguntas são frequentemente baseadas em categorias discretas:
 - Esse tratamento é efetivo em melhorar resultados?
 - Duas regiões geográficas são diferentes em algum aspecto?
 - A dieta de um grupo de espécies afeta o seu tamanho?



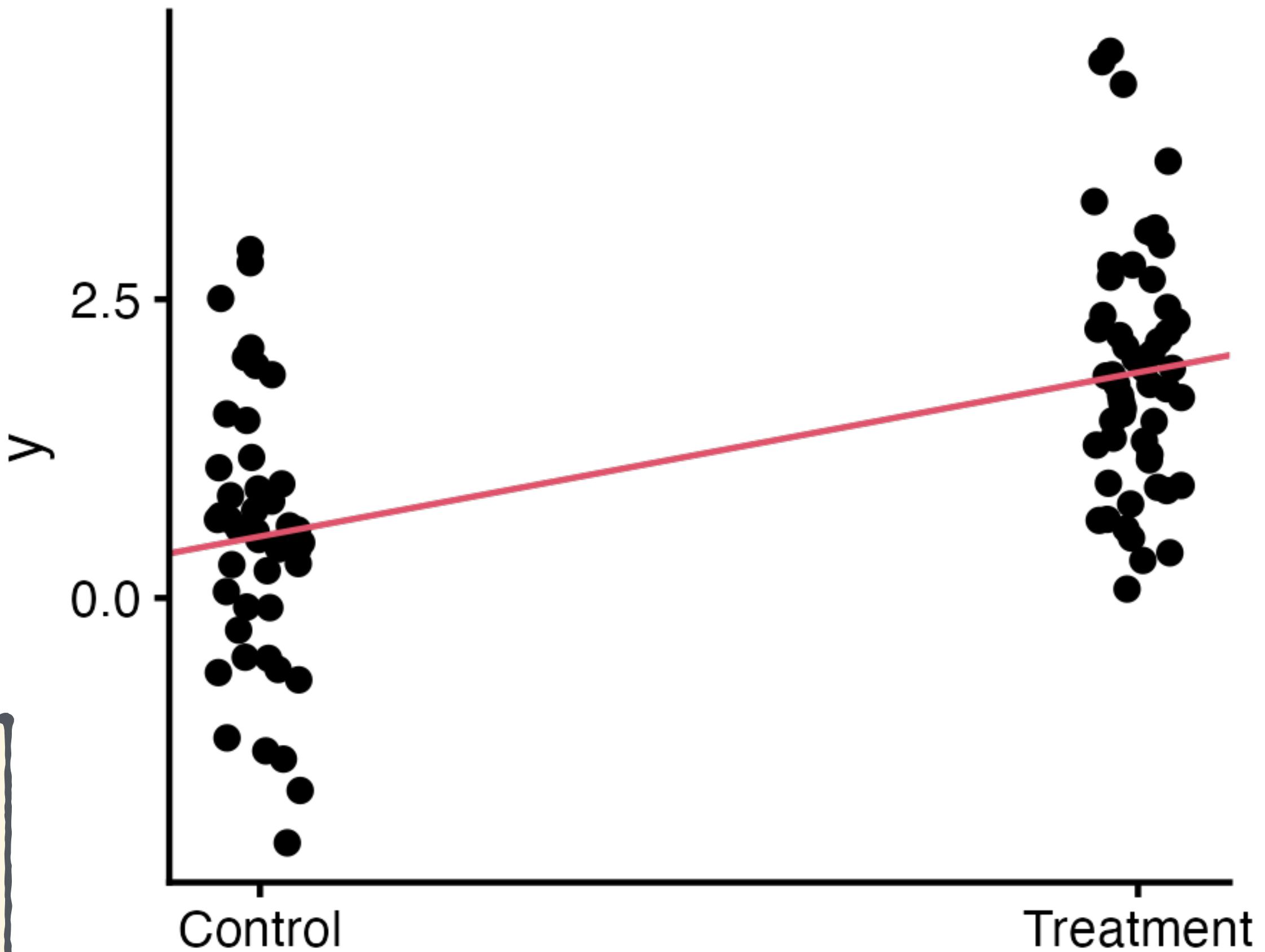
Predictor binário, o mesmo modelo

Tratamento-controle, duas categorias, etc.

$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$

- x_i : 0 para controle, 1 para tratamento
- α : intercepto é a **média do grupo controle**.
- β : A inclinação é a **diferença na médias dos grupos.**

```
> precis(fit, prob = 0.95)
      mean    sd 2.5% 97.5%
a     0.52  0.16  0.20  0.82
b     1.37  0.22  0.96  1.79
```



E se um preditor tiver mais de duas categorias?

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

```
> x = sample(LETTERS[1:3], 9, replace = TRUE)
> y = 1 + ifelse(x == "A", 0,
+                 ifelse(x == "B", 1, 2)) + rnorm(9)
> df = tibble(y, x)
> df
# A tibble: 9 × 2
      y     x
   <dbl> <chr>
1  1.25    B
2  0.870   A
3 -0.00180 A
4  1.18    B
5  2.03    C
6  2.60    B
7  2.55    B
8  3.92    C
9  4.66    B
```

E se um preditor tiver mais de duas categorias?

Contrasts, o padrão na maioria dos casos

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

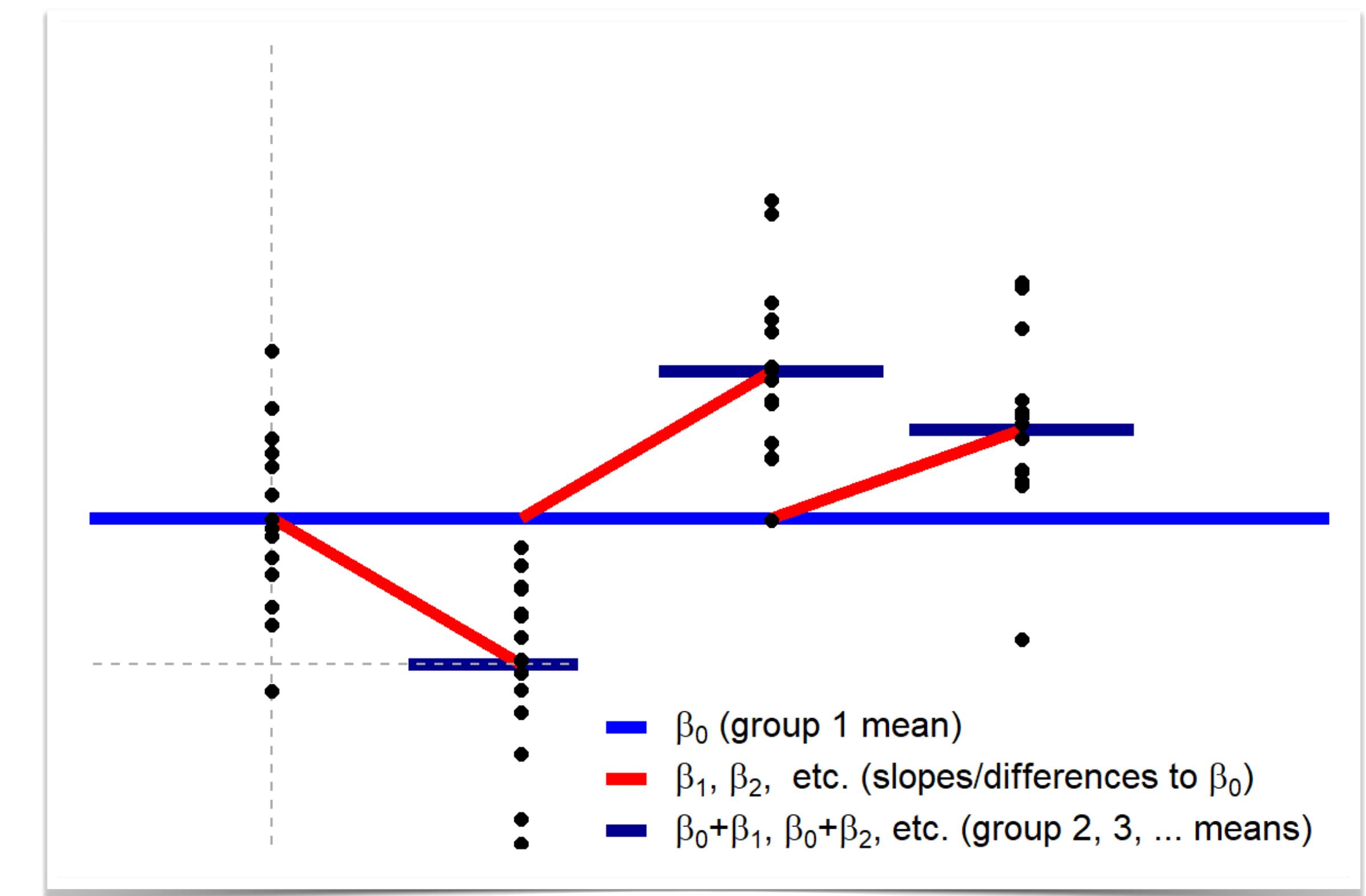
```
> fit_contrasts = stan_glm(y ~ x, data = df)
> summary(fit_contrasts)[1:3, 1:3]
            mean mcse   sd
(Intercept) 0.72    0 0.13
xB           1.32    0 0.20
xC           2.20    0 0.186
```

Q: Como construir estimativas da média de cada grupo?

E se um preditor tiver mais de duas categorias?

Contrasts, o padrão na maioria dos casos

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.



E se um preditor tiver mais de duas categorias?

One-hot

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

```
> x
[1] "B" "A" "A" "B" "C" "B" "B" "C" "B"
> onehot
  xA  xB  xC
1  0   1   0
2  1   0   0
3  1   0   0
4  0   1   0
5  0   0   1
6  0   1   0
7  0   1   0
8  0   0   1
9  0   1   0
```

E se um preditor tiver mais de duas categorias?

One-hot

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

```
> onehot = model.matrix(~0+x, data = df)
> fit_onehot = stan_glm(y ~ 0 + onehot, data =
df, cores = 4)
> summary(fit_onehot)[1:3, 1:3]
            mean   mcse     sd
onehotxA 0.72    0 0.13
onehotxB 2.04    0 0.14
onehotxC 2.93    0 0.12
```

E se um preditor tiver mais de duas categorias?

One-hot

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

```
> onehot = model.matrix(~0+x, data = df)
> fit_onehot = stan_glm(y ~ 0 + onehot, data =
df, cores = 4)
> summary(fit_onehot)[1:3, 1:3]
      mean  mcse    sd
onehotxA 0.72    0 0.13
onehotxB 2.04    0 0.14
onehotxC 2.93    0 0.12
```

```
> fit_contrasts = stan_glm(y ~ x, data = df)
> summary(fit_contrasts)[1:3, 1:3]
      mean  mcse    sd
(Intercept) 0.72    0 0.13
xB          1.32    0 0.20
xC          2.20    0 0.186
```

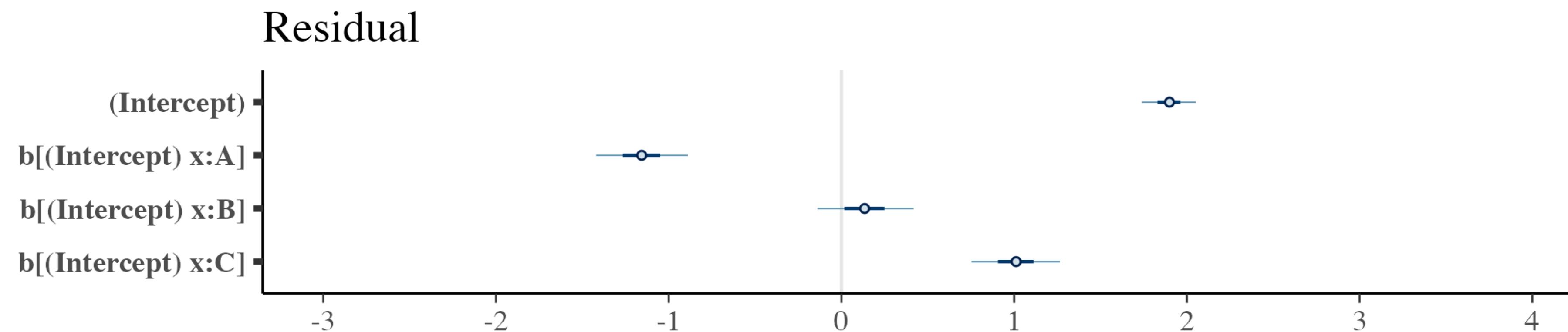
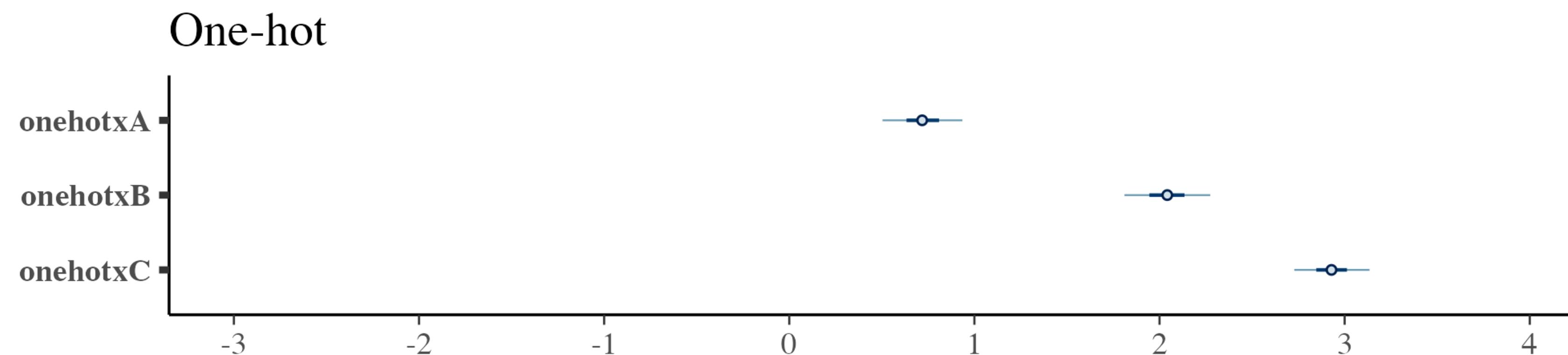
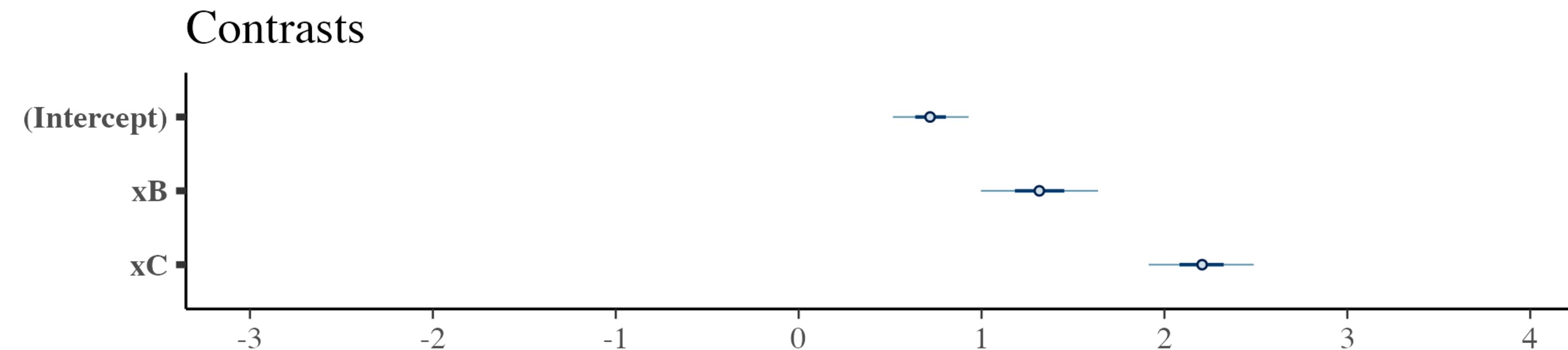
E se um preditor tiver mais de duas categorias?

- Existem várias maneiras de codificar um preditor com vários níveis:
 - **Contrastes**: Cada nível é comparado com uma **referência**, e os coeficientes são comparações entre a referência e cada nível.
 - **One-hot**: Coeficiente são a média de cada nível do preditor.
 - **Residuals**: Uma média geral é calculada, e os coeficientes são a diferença entre a média de cada nível e a média global.

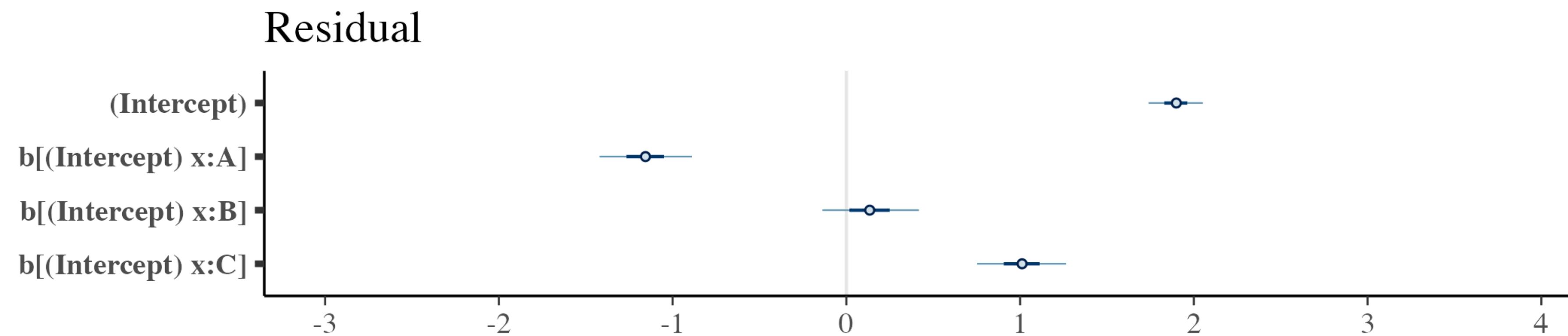
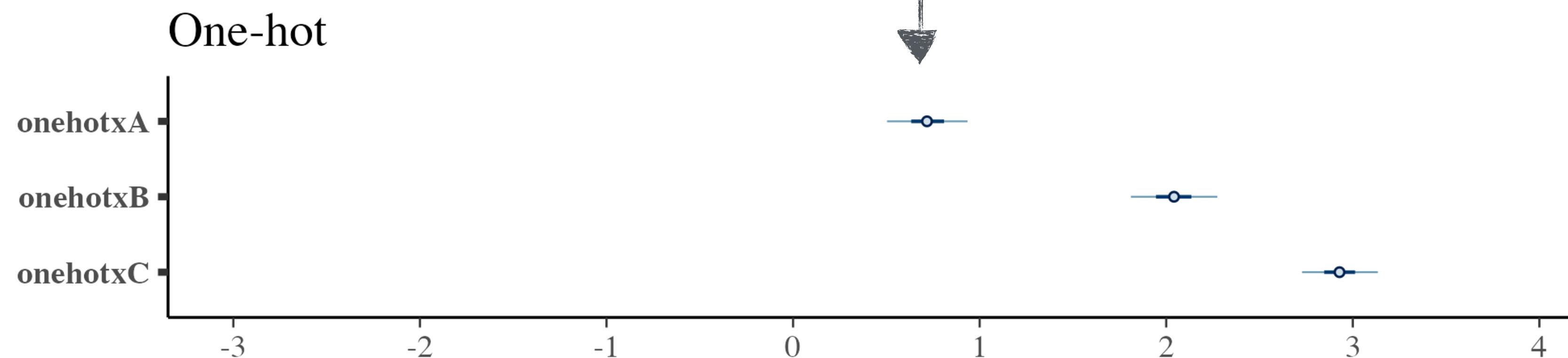
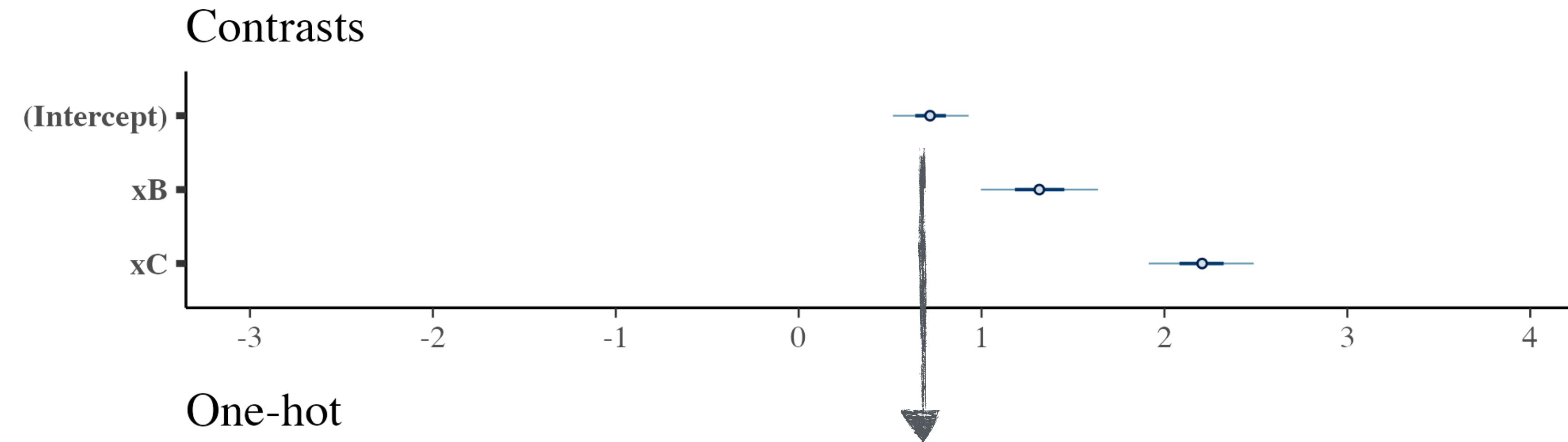
```
> fit_residual = stan_glmer(y ~ 1 + (1|x),
  data = df, prior_intercept =
  normal(mean(df$y), 0.1))
> summary(fit_residual)[1:4, 1:3]
```

	mean	mcse	sd
(Intercept)	1.90	0	0.10
b[(Intercept) x:A]	-1.16	0	0.16
b[(Intercept) x:B]	0.14	0	0.17
b[(Intercept) x:C]	1.01	0	0.15

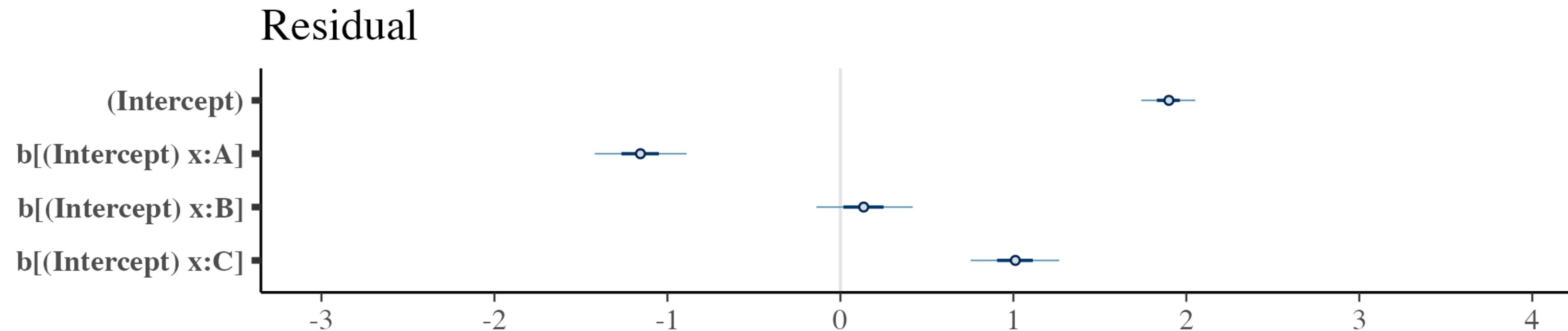
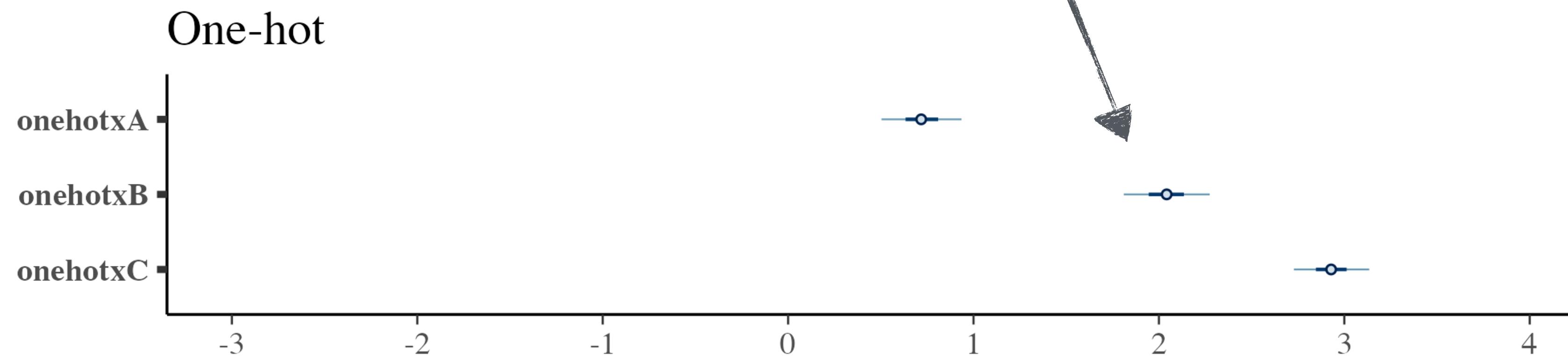
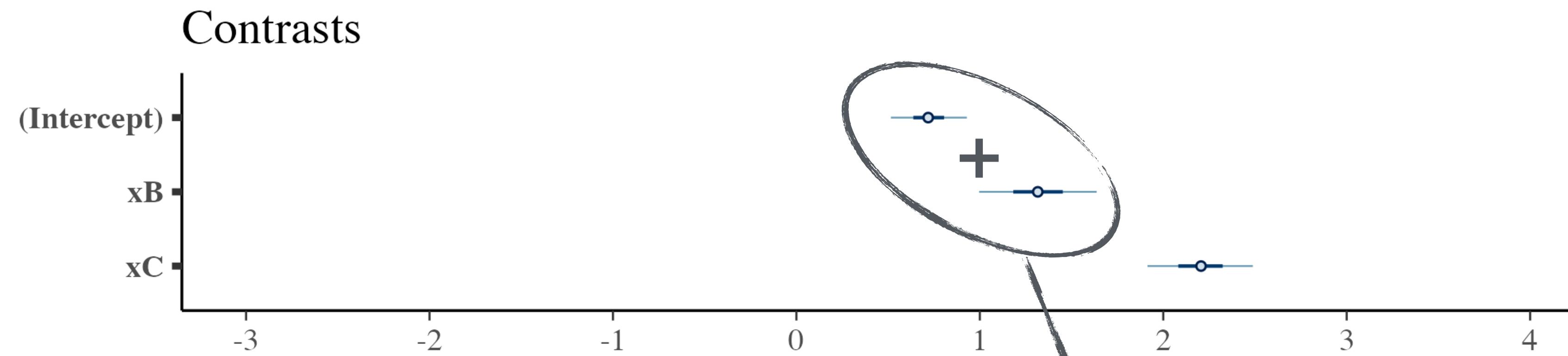
Comparando estimativas



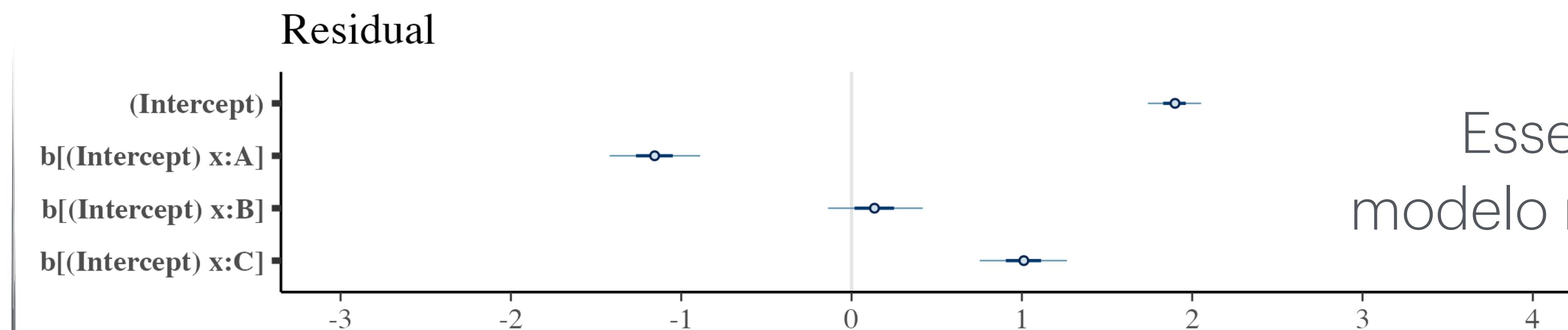
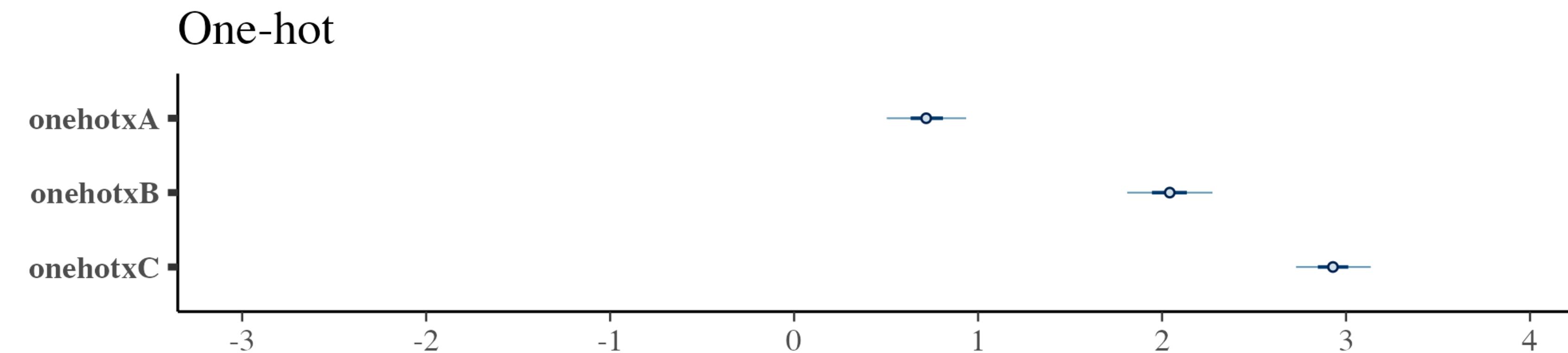
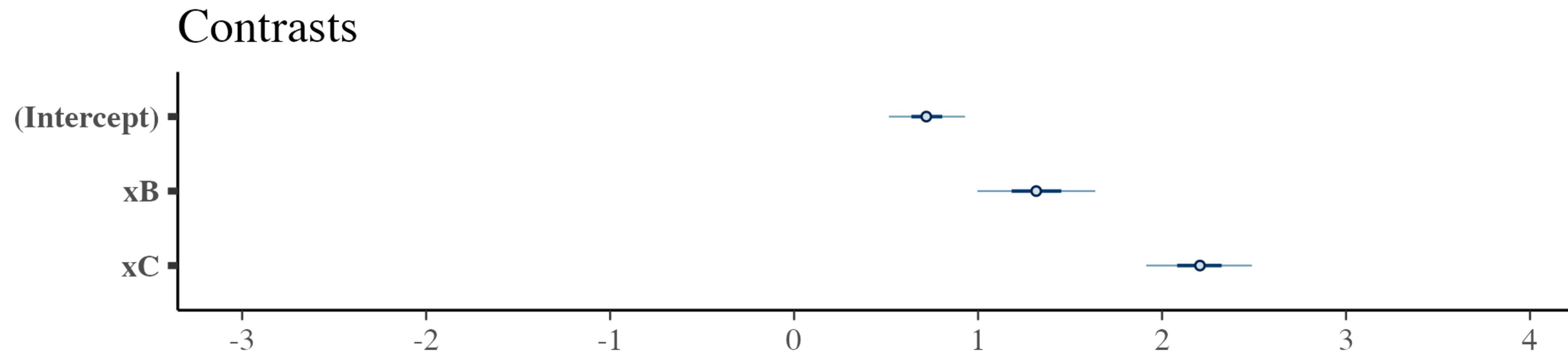
Comparando estimativas



Comparando estimativas



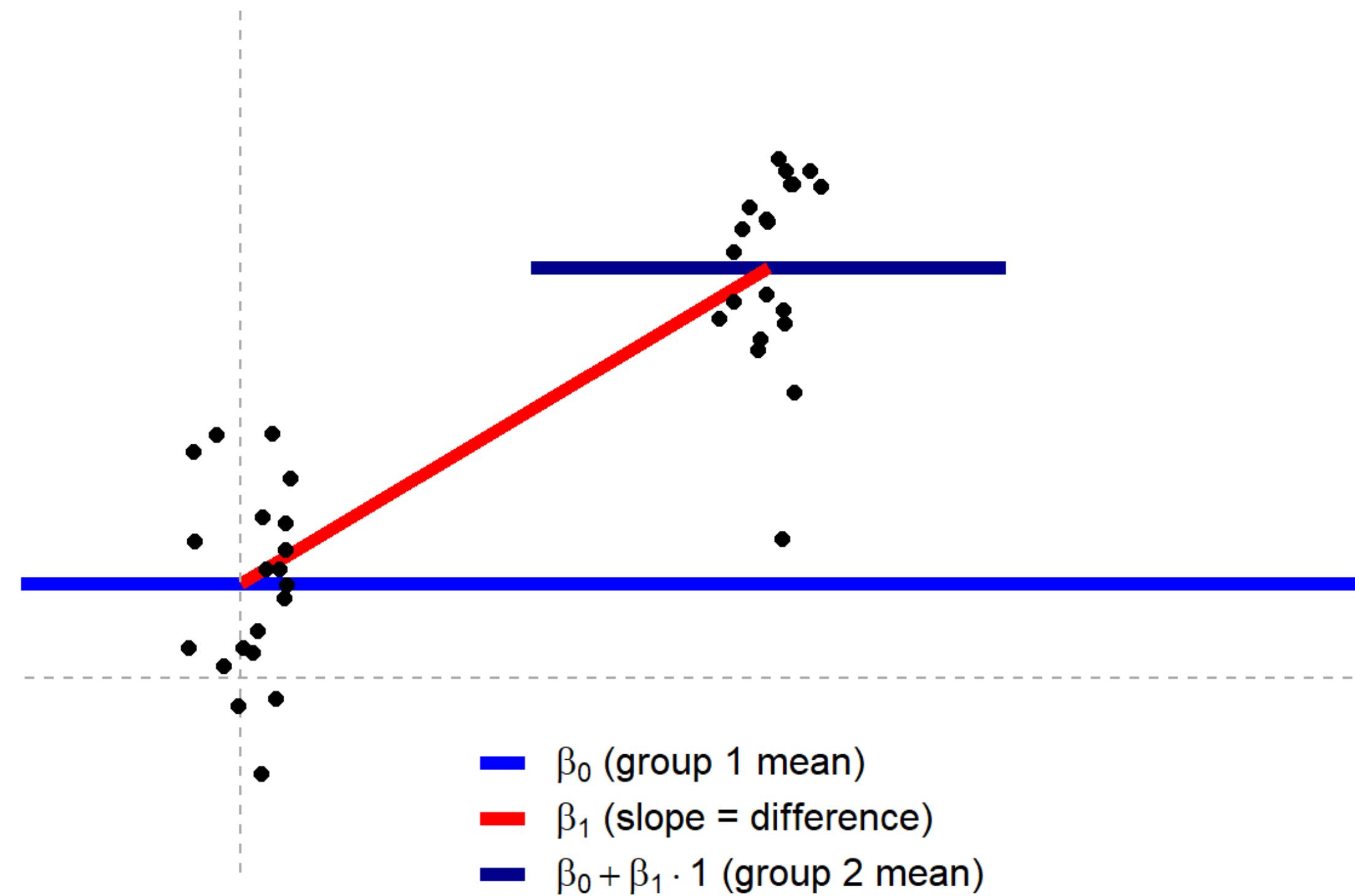
Comparando estimativas



Comparando testes com modelos lineares

Teste-t vs lm()

- Vários testes padrões podem ser expressos como modelos lineares:
 - <https://lindeloev.github.io/tests-as-linear/>
 - Comparando médias de duas amostras



Comparando testes com modelos lineares

Teste-t vs lm()

- Vamos utilizar o exemplo de tamanho de mandíbulas de chacal dourado que se encontra no livro do Bryan Manly).
- Os dados são provenientes da coleção do Museu Britânico de História Natural em Londres e foram publicados em um artigo de 1980.
- Aqui estamos interessados apenas nos tamanhos das mandíbulas e se há diferença entre fêmeas e machos.

```
macho <- c(120, 107, 110, 116, 114, 111,  
113, 117, 114, 112)  
femea <- c(110, 111, 107, 108, 110, 105,  
107, 106, 111, 111)  
  
chacal <- c(macho, femea)  
sexo <- factor(rep(c("macho", "femea"),  
each=10))
```

Comparando testes com modelos lineares

Teste-t vs lm()

```
macho <- c(120, 107, 110, 116, 114, 111, 113, 117, 114, 112)
femea <- c(110, 111, 107, 108, 110, 105, 107, 106, 111, 111)

> t.test(macho, femea, var.equal = TRUE)

  Two Sample t-test
data: macho and femea
t = 3.4843, df = 18, p-value = 0.002647
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.905773 7.694227
sample estimates:
mean of x mean of y
 113.4      108.6
```

Comparando testes com modelos lineares

Teste-t vs lm()

```
> chacal <- c(macho, femea)
> sexo <- factor(rep(c("macho", "femea"), each=10))

> fit2 <- lm(chacal ~ sexo)
> summary(fit2)

Call:
lm(formula = chacal ~ sexo)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.4     -1.8     0.1     2.4     6.6 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 108.6000    0.9741 111.486 < 2e-16 ***
sexomacho    4.8000    1.3776   3.484  0.00265 **  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.08 on 18 degrees of freedom
Multiple R-squared:  0.4028,    Adjusted R-squared:  0.3696 
F-statistic: 12.14 on 1 and 18 DF,  p-value: 0.002647
```

Comparando testes com modelos lineares

Teste-t vs lm()

```
> t.test(macho, femea, var.equal = TRUE)

  Two Sample t-test
data: macho and femea
t = 3.4843, df = 18, p-value = 0.002647
95 percent confidence interval:
 1.905773 7.694227
sample estimates:
mean of x mean of y
 113.4      108.6
```

```
> fit2 <- lm(chacal ~ sexo)
> summary(fit2)

Call:
lm(formula = chacal ~ sexo)

Residuals:
    Min     1Q Median     3Q    Max 
 -6.4   -1.8    0.1    2.4   6.6 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 108.6000    0.9741 111.486 < 2e-16 ***
sexomacho    4.8000    1.3776   3.484  0.00265 **  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.08 on 18 degrees of freedom
Multiple R-squared:  0.4028,    Adjusted R-squared:  0.3696 
F-statistic: 12.14 on 1 and 18 DF,  p-value: 0.002647
```

Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: <https://lindeloev.github.io/tests-as-linear>

Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon	
Simple regression: $\text{Im}(y \sim 1 + x)$	y is independent of x P: One-sample t-test N: Wilcoxon signed-rank	<code>t.test(y)</code> <code>wilcox.test(y)</code>	<code>lm(y ~ 1)</code> <code>lm(signed_rank(y) ~ 1)</code>	✓ for N >14	One number (intercept, i.e., the mean) predicts y. - (Same, but it predicts the <i>signed rank</i> of y.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	<code>t.test(y1, y2, paired=TRUE)</code> <code>wilcox.test(y1, y2, paired=TRUE)</code>	<code>lm(y2 - y1 ~ 1)</code> <code>lm(signed_rank(y2 - y1) ~ 1)</code>	✓ for N >14	One intercept predicts the pairwise $y_2 - y_1$ differences. - (Same, but it predicts the <i>signed rank</i> of $y_2 - y_1$.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	<code>cor.test(x, y, method='Pearson')</code> <code>cor.test(x, y, method='Spearman')</code>	<code>lm(y ~ 1 + x)</code> <code>lm(rank(y) ~ 1 + rank(x))</code>	✓ for N >10	One intercept plus x multiplied by a number (slope) predicts y. - (Same, but with <i>ranked x</i> and y)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	<code>t.test(y1, y2, var.equal=TRUE)</code> <code>t.test(y1, y2, var.equal=FALSE)</code> <code>wilcox.test(y1, y2)</code>	<code>lm(y ~ 1 + G₂)^A</code> <code>gls(y ~ 1 + G₂, weights=...^B)^A</code> <code>lm(signed_rank(y) ~ 1 + G₂)^A</code>	✓ ✓ for N >11	An intercept for group 1 (plus a difference if group 2) predicts y. - (Same, but with one variance <i>per group</i> instead of one common.) - (Same, but it predicts the <i>signed rank</i> of y.)	
Multiple regression: $\text{Im}(y \sim 1 + x_1 + x_2 + \dots)$	P: One-way ANOVA N: Kruskal-Wallis	<code>aov(y ~ group)</code> <code>kruskal.test(y ~ group)</code>	<code>lm(y ~ 1 + G₂ + G₃ + ... + G_N)^A</code> <code>lm(rank(y) ~ 1 + G₂ + G₃ + ... + G_N)^A</code>	✓ for N >11	An intercept for group 1 (plus a difference if group ≠ 1) predicts y. - (Same, but it predicts the <i>rank</i> of y.)	
	P: One-way ANCOVA	<code>aov(y ~ group + x)</code>	<code>lm(y ~ 1 + G₂ + G₃ + ... + G_N + x)^A</code>	✓	- (Same, but plus a slope on x.) <i>Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.</i>	
	P: Two-way ANOVA	<code>aov(y ~ group * sex)</code>	<code>lm(y ~ 1 + G₂ + G₃ + ... + G_N + S₂ + S₃ + ... + S_K + G₂*S₂+G₃*S₃+...+G_N*S_K)^A</code>	✓	Interaction term: changing sex changes the y ~ group parameters. <i>Note: G_{2 to N} is an indicator (0 or 1) for each non-intercept levels of the group variable. Similarly for S_{2 to K} for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the group × sex interaction. For two levels (e.g. male/female), line 2 would just be "S₂" and line 3 would be S₂ multiplied with each G_i.</i>	[Coming]
	Counts ~ discrete x N: Chi-square test	<code>chisq.test(groupXsex_table)</code>	Equivalent log-linear model <code>glm(y ~ 1 + G₂ + G₃ + ... + G_N + S₂ + S₃ + ... + S_K + G₂*S₂+G₃*S₃+...+G_N*S_K, family=...)^A</code>	✓	Interaction term: (Same as Two-way ANOVA.) <i>Note: Run glm using the following arguments: <code>glm(model, family=poisson())</code>. As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(a_i) + \log(\beta_j) + \log(a_i\beta_j)$ where a_i and β_j are proportions. See more info in the accompanying notebook.</i>	Same as Two-way ANOVA
N: Goodness of fit	<code>chisq.test(y)</code>	<code>glm(y ~ 1 + G₂ + G₃ + ... + G_N, family=...)^A</code>	✓	(Same as One-way ANOVA and see Chi-Square note.)	1W-ANOVA	

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation $y \sim 1 + x$ is R shorthand for $y = 1 \cdot b + a \cdot x$ which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see “Exact” column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are “[dummy coded](#)” indicator variables (either 0 or 1) exploiting the fact that when $\Delta x = 1$ between categories the difference equals the slope. Subscripts (e.g., G₂ or y₁) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at <https://lindeloev.github.io/tests-as-linear>.

^A See the note to the two-way ANOVA for explanation of the notation.

^B Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.

