

Regressão ordinal bayesiana - uma introdução

Igor Costa

18/05/2022

Motivação

Esse material pretende fazer uma breve introdução aos chamados modelos de regressão ordinais, ajustados a variáveis dependentes categorias ordinais, ou seja, em que a ordem dos elementos é importante (nível de escolaridade, respostas em escala do tipo Likert, etc). As discussões aqui feitas restringem-se ao campo da psicolinguística experimental e são parte dos encontros dos membros do [Laboratório de Psicolinguística e Aquisição da Linguagem - LAPAL/PUC-Rio](#).

Recomendações de leitura:

- Veríssimo, João (2021). *Analysis of rating scales: A pervasive problem in bilingualism research and a solution with Bayesian ordinal models*.
- Gelman & Hill (2009). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Mais especificamente: Cap. 5 Logistic regression (p. 79-105); Cap. 6.5 Multinomial regression (119-123); Cap. 15.2 Ordered categorical regression: storable votes (331-332).
- Nicenboim, B., Schad, D. & Vasishth, S. *An Introduction to Bayesian Data Analysis for Cognitive Science*.
- Kurz, S. *Doing Bayesian Data Analysis in brms and the tidyverse*. Mais especificamente: Cap. 23 - *Ordinal Predicted Variable*.

Carregando os pacotes adequados

```
require(ggplot2)
require(dplyr)
require(RColorBrewer)
require(tidyr)
require(scales)
require(brms) # Esse o pacote para fazermos o modelo ordinal bayesiano
require(sjPlot) # Fundamental para extrair os efeitos marginais (última etapa deste tutorial)
require(tidybayes)
```

Carregamento e organização dos dados brutos

Carregando os dados de um experimento com Escala Likert:

```
dados <-
  read.csv("https://raw.githubusercontent.com/igordeo-costa/ModelosOrdinais/main/ScalaLikertCosta2022.csv")
  stringsAsFactors = T)
```

Transformando demais as variáveis de modo adequado

```
dados$idade<-as.integer(dados$idade)
dados$subj <- as.factor(dados$subj)
```

Mudar os dados respostas para ordenados

```
dados$answer<-as.ordered(dados$answer)
```

Investigando os dados - Análise descritiva

Contagem dos valores brutos:

```
contag <- dados %>%
  group_by(Ordem, Num, answer) %>%
  tally() %>%
  group_by(Ordem, Num) %>%
  spread(answer, n)

colnames(contag) <- c("Ordem", "Num",
                     "Discordo_Totalmente", "Discordo", "Neutro",
                     "Concordo", "Concordo_Totalmente")

contag
```

```
## # A tibble: 4 x 7
## # Groups:   Ordem, Num [4]
##   Ordem  Num  Discordo_Totalmente Discordo Neutro Concordo Concordo_Totalmente
##   <fct> <fct>                <int>   <int> <int>   <int>          <int>
## 1 todo-um PL              156      72    20     27         23
## 2 todo-um SG               27      25    40     53        149
## 3 um-todo PL               66      26    42     66        108
## 4 um-todo SG               20      12    33     53        181
```

Cálculo das porcentagens respectivas (tabela em formato horizontal)

```
porc_horiz <- dados %>%
  group_by(Ordem, Num, answer) %>%
  tally() %>%
  mutate(perc=n/sum(n)) %>%
  dplyr::select(-n) %>%
  group_by(Ordem, Num) %>%
  spread(answer, perc)

colnames(porc_horiz) <- c("Ordem", "Num",
                        "Discordo_Totalmente", "Discordo", "Neutro",
                        "Concordo", "Concordo_Totalmente")

porc_horiz
```

```
## # A tibble: 4 x 7
## # Groups:   Ordem, Num [4]
##   Ordem  Num  Discordo_Totalmente Discordo Neutro Concordo Concordo_Totalmente
##   <fct> <fct>                <dbl>   <dbl> <dbl>   <dbl>          <dbl>
## 1 todo-um PL              0.523   0.242 0.0671  0.0906         0.0772
## 2 todo-um SG              0.0918   0.0850 0.136   0.180         0.507
## 3 um-todo PL              0.214    0.0844 0.136   0.214         0.351
```

```
## 4 um-todo SG
```

```
0.0669 0.0401 0.110 0.177
```

```
0.605
```

Gráfico de barras empilhadas para fácil visualização

Essa parte é difícil de compreender se não temos a noção do resultado final. Você pode rodá-la e voltar depois com calma para entender os códigos. Resumidamente: vamos fazer um gráfico de barras empilhadas centrado na categoria “neutro”, ou seja, no meio da escala. Para isso, vamos criar dois conjuntos de dados: as porcentagens da parte de cima (“concordo” e “concordo totalmente”) e as da parte de baixo (“discordo” e “discordo totalmente”). A categoria “neutro” será dividida em duas e cada pedaço anexado a uma das partes acima.

Dividir o meio da escala (os julgamentos “Neutro”):

```
dados_meio <- porc_horiz %>%
  mutate(c1 = Neutro / 2,
         c2 = Neutro / 2) %>%
  select(Ordem, Num,
         Discordo_Totalmente, Discordo, c1, c2, Concordo, Concordo_Totalmente) %>%
  gather(key = answer, value = perc, 3:8)
```

Separar a escala em dois conjuntos, o “alto” e o “baixo”:

```
meio_alto <- dados_meio %>%
  filter(answer %in% c("Concordo_Totalmente", "Concordo", "c2")) %>%
  # Níveis na ordem normal!
  mutate(answer = factor(answer, levels = c("Concordo_Totalmente", "Concordo", "c2")))

meio_baixo <- dados_meio %>%
  filter(answer %in% c("c1", "Discordo", "Discordo_Totalmente")) %>%
  # Níveis na ordem inversa!
  mutate(answer = factor(answer, levels = c("Discordo_Totalmente", "Discordo", "c1")))
```

Estabelecer uma paleta de cores para organização!

```
# Usar, do pacote RColorBrewer, a paleta de cores "spectral", com 5 cores
legend_pal <- brewer.pal(name = "Spectral", n = 5)
# Duplica a cor do meio manualmente
legend_pal <- c("#2B83BA", "#ABDDA4", "#FFFFBF", "#FFFFBF", "#FDAE61", "#D7191C")
# Substitui a cor do meio por um cinza
legend_pal <- gsub("#FFFFBF", "#9C9C9C", legend_pal)
# Atribui nomes às cores
names(legend_pal) <-
  c("Concordo_Totalmente", "Concordo", "c1", "c2", "Discordo", "Discordo_Totalmente")
```

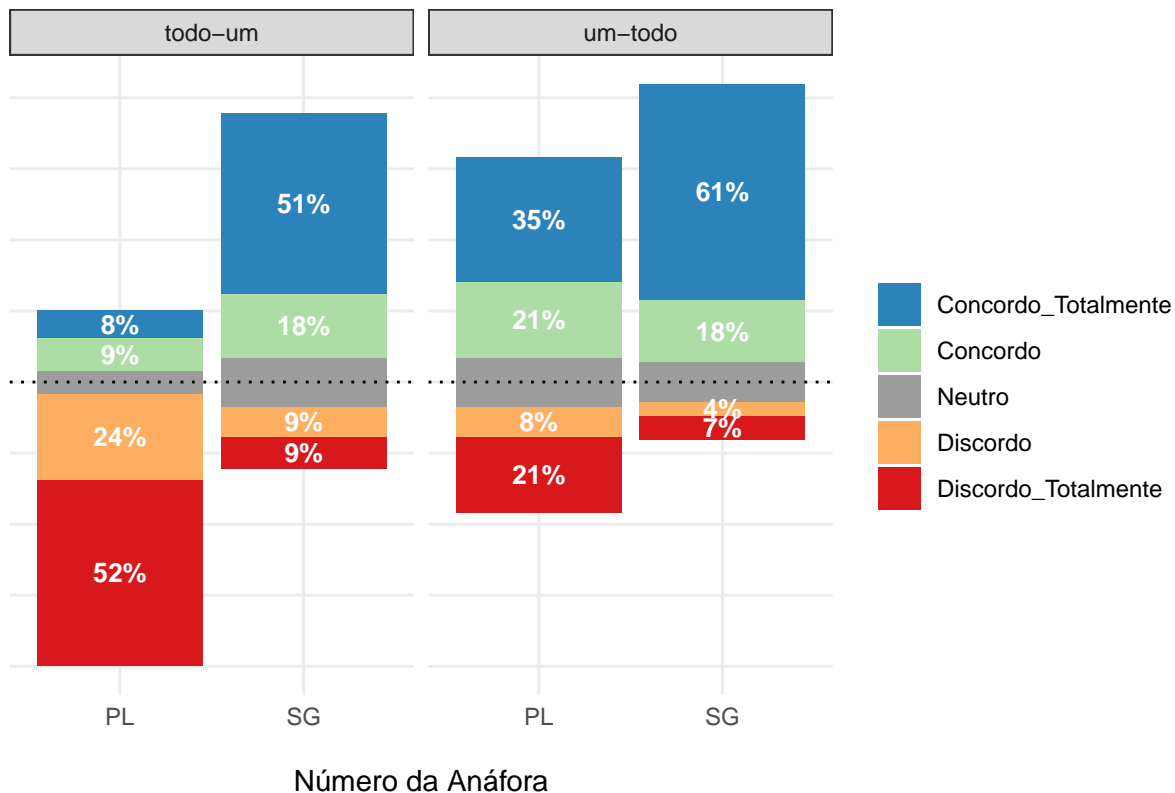
Produzir o gráfico, finalmente!

```
ggplot() +
  geom_bar(data = meio_alto, aes(x = Num, y=perc, fill = answer), stat="identity") +
  geom_bar(data = meio_baixo, aes(x = Num, y=-perc, fill = answer), stat="identity") +
  #-----
  # Essa parte do código é um pouco complicada...
  # Serve apenas para colocar os valores dentro das barras de modo alinhado...
  geom_text(data = meio_alto,
            aes(alpha = answer, x = Num, y=perc, group = answer,
               label = scales::percent(perc, accuracy = 1)),
            color = "white", size = 3.5, fontface = "bold",
            position = position_stack(vjust = .5)) +
```

```

geom_text(data = meio_baixo,
  aes(alpha = answer, x = Num, y=-perc, group = answer,
    label = scales::percent(perc, accuracy = 1)),
  size = 3.5, color = "white", fontface = "bold",
  position = position_stack(vjust = .5),) +
scale_alpha_manual(values = c("c1" = 0, "c2" = 0,
  "Discordo" = 1, "Discordo_Totalmente" = 1,
  "Concordo_Totalmente" = 1, "Concordo" = 1),
  guide = 'none') +
#-----
geom_hline(yintercept = 0, color = c("black"), linetype = "dotted") +
facet_wrap(~Ordem) +
scale_y_continuous(breaks = seq(from = -1, to = 1, by = .2),
  # Definir os valores negativos da legenda como positivos
  labels = function(x) percent(abs(x))) +
scale_fill_manual(values = legend_pal,
  breaks = c("Concordo_Totalmente", "Concordo", "c2",
    "Discordo", "Discordo_Totalmente"),
  labels = c("Concordo_Totalmente", "Concordo", "Neutro",
    "Discordo", "Discordo_Totalmente")) +
labs(x = "\nNúmero da Anáfora", y = "", fill = "") +
ggtitle("") +
theme_bw() +
theme(axis.ticks.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  plot.background = element_blank(),
  panel.grid.minor = element_blank(),
  panel.border = element_blank())

```



Ajustando um modelo ordinal misto

Agora vamos de fato ajustar um modelo ordinal bayesiano...

Verificando quais os contrastes dos dados:

```
contrasts(dados$Num)
```

```
##      SG
## PL    0
## SG    1
```

```
contrasts(dados$Ordem)
```

```
##          um-todo
## todo-um         0
## um-todo         1
```

Mudar o nível-base para “um-todo” para facilitar a comparação entre as condições “um-todo” SG x PL

```
dados$Ordem<-relevel(dados$Ordem, ref = "um-todo")
```

Ajustando um modelo de regressão ordinal bayesiano sem definição das priors informativas (“flat priors”). Este não é o melhor cenário, mas o objetivo aqui é apenas entender o funcionamento e implementação do modelo.

Vamos rodar o modelo mais completo, pois demora um pouco... ou muito, dependendo da capacidade de processamento do seu computador...

```
# Vamos garantir que o programa use toda a nossa capacidade de computação para acelerar o
# processo usando todos os núcleos do nosso processador
options(mc.cores = parallel::detectCores())
```

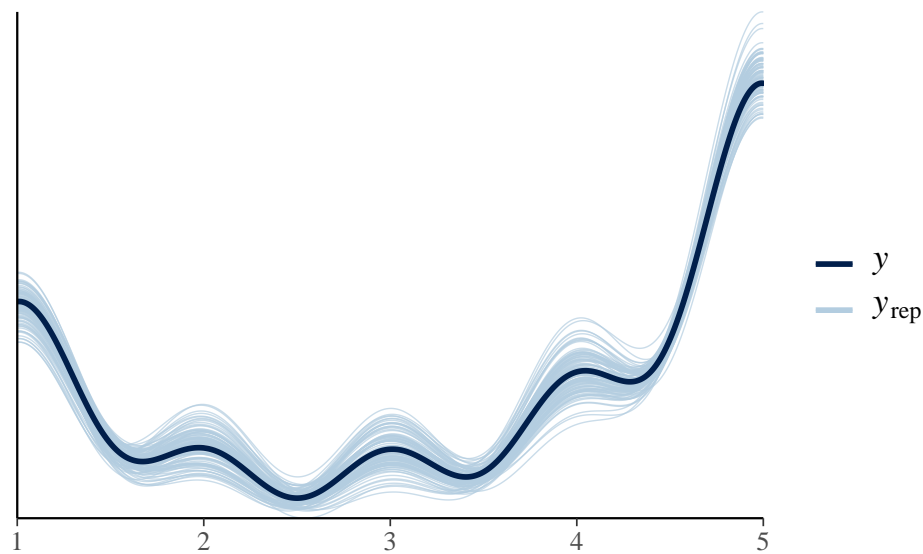
```
# Aqui de fato o modelo
m2 <- brm(answer ~ Ordem*Num + (1+Ordem*Num|subj)+(1+Ordem*Num|item),
  data = dados,
  family = cumulative(link = "logit", threshold = "flexible"),
  silent = 2, refresh = 0) # Apenas para não dar avisos no arquivo .pdf

# Investigando apenas os fatores fixos
fixef(m2)[5:7,]
```

##	Estimate	Est.Error	Q2.5	Q97.5
## OrdemtodoMum	-2.462673	0.3647994	-3.1935692	-1.787967
## NumSG	1.623304	0.4113184	0.8236970	2.437261
## OrdemtodoMum:NumSG	1.671090	0.4014726	0.8753537	2.486461

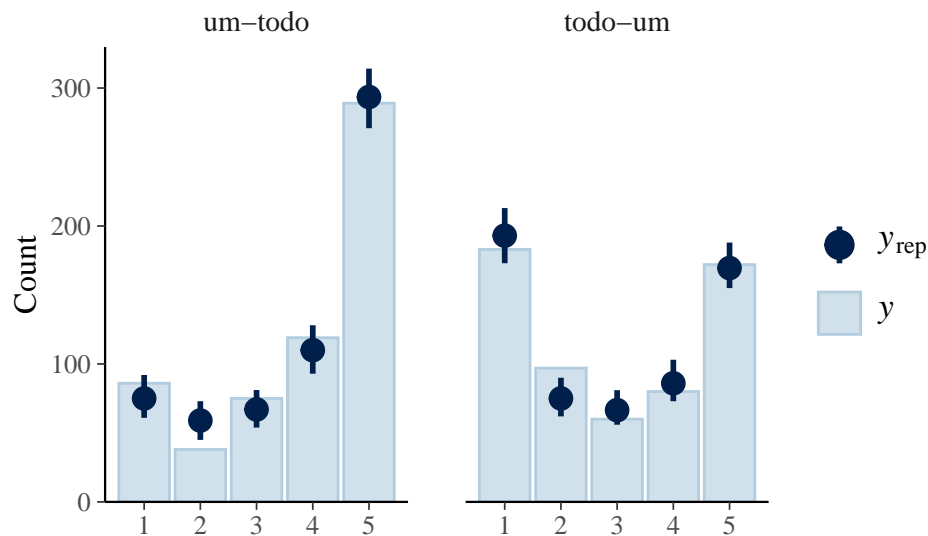
Você pode checar o ajuste do modelo, ou seja, a relação entre as priors e os dados. Neste caso, porém, ele será bom, visto que, com ‘flat priors’ o modelo usava os próprios dados como critério...

```
pp_check(m2, ndraws = 100)
```



```
# Investigue também no formato de barras, se desejar
# pp_check(m2, type = "bars", ndraws = 100)

# Você pode investigar, inclusive, por grupo:
pp_check(m2, type = "bars_grouped", ndraws = 100, group = "Ordem")
```



Visualizando as estimativas

Extraindo os coeficientes dos fatores fixos e intervalos de confiança calculados

```
fixos.m<-fixef(m2)[5:7,]
```

```
# Prepara um data frame com esses dados
```

```
colnames(fixos.m)<-c("Estimativas", "Est.Error", "lower", "upper")
```

```
fixos.m<-as.data.frame(fixos.m)
```

```
# Calcula da chance e das probabilidades correspondentes
```

```
fixos.m<-fixos.m %>%
```

```
  mutate(Odds=exp(Estimativas)) %>%
```

```
  mutate(Probs=if_else(Estimativas<0, # Se logOdds menor do que zero, então:
```

```
    (exp(Estimativas)/(1 + exp(Estimativas))-1)*100, # Probab. de descer na escala  
    exp(Estimativas)/(1 + exp(Estimativas))*100)) # Probab. de subir na escala
```

```
rownames(fixos.m)<-c("ORDEM: todo_um", "NÚMERO: singular", "INTERAÇÃO: ordem x número")
```

```
round(fixos.m, 3)
```

```
##               Estimativas Est.Error  lower  upper Odds  Probs  
## ORDEM: todo_um      -2.463    0.365 -3.194 -1.788 0.085 -92.148  
## NÚMERO: singular     1.623    0.411  0.824  2.437 5.070 83.525  
## INTERAÇÃO: ordem x número  1.671    0.401  0.875  2.486 5.318 84.172
```

Agora vamos plotar o gráfico para publicação...

```
fixos.m %>%
```

```
  ggplot(aes(x=reorder(rownames(fixos.m), Estimativas),  
             y=Estimativas))+
```

```
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey")+
```

```
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.0,
```

```
                position=position_dodge(.9))+
```

```
  geom_point(color="orange", size = 3)+
```

```
  geom_text(aes(label=round(Estimativas, 3)), vjust=-1)+
```

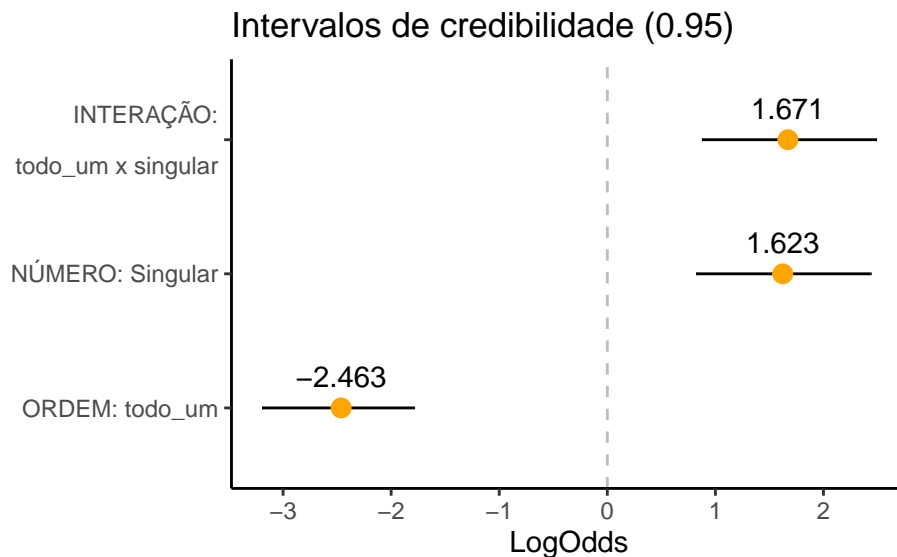
```
  # Ao mudar a ordem é preciso mudar aqui embaixo...
```

```
  scale_x_discrete(labels=c("ORDEM: todo_um", "NÚMERO: Singular",
```

```

"INTERAÇÃO:\n\ntodo_um x singular")) +
labs(y = "LogOdds", x = "") +
ggtitle("Intervalos de credibilidade (0.95)") +
coord_flip()+theme_classic()

```



Uma outra forma de pensar sobre esse gráfico é olhar para a relação entre a distribuição das estimativas dos fatores fixos e os interceptos (tau-cuts τ) estimados pelo modelo. A informação dada, no final, é a mesma do gráfico anterior, mas talvez fique mais claro o que aquele gráfico está mostrando.

Primeiro, vamos extrair os desvios padrão e estimativas da posterior:

```

(posterior.draws <- m2 %>%
  spread_draws(`b_.*`, regex = TRUE) %>%
  summarise_draws())

```

```

## # A tibble: 7 x 10
##   variable      mean median    sd  mad    q5    q95  rhat ess_bulk ess_tail
##   <chr>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 b_Intercept[1] -2.13  -2.12  0.309 0.302 -2.65 -1.64  1.00  2239.  2268.
## 2 b_Intercept[2] -1.15  -1.15  0.297 0.288 -1.64 -0.673 1.00  2356.  2478.
## 3 b_Intercept[3] -0.314 -0.309 0.295 0.283 -0.800 0.163 1.00  2342.  2610.
## 4 b_Intercept[4]  0.834  0.833 0.295 0.284  0.349  1.31  1.00  2270.  2919.
## 5 b_OrdemtodoMum -2.46  -2.45  0.365 0.354 -3.07 -1.89  1.00  2276.  2175.
## 6 b_NumSG        1.62   1.62  0.411 0.389  0.961  2.31  1.00  2705.  2728.
## 7 b_OrdemtodoMu~  1.67   1.66  0.401 0.382  1.02  2.35  1.00  2697.  2677.

```

Em seguida, vamos plotar um gráfico com alguns desses dados:

```

legend_pal <- c("#ABDDA4", "#FDAE61", "#D7191C")

data.frame(
  fatores = c("Todo_um", "Singular", "Interação"),
  posterior.draws[5:7,2:4]) %>%
  expand(nesting(fatores, mean, sd),
    y = seq(from = -4, to = 4, by = 0.001)) %>%
  mutate(d = dnorm(y, mean, sd)) %>%
  ggplot(aes(x = y, y = d, fill = fatores)) +
  geom_density(stat = "identity", alpha = 0.5,

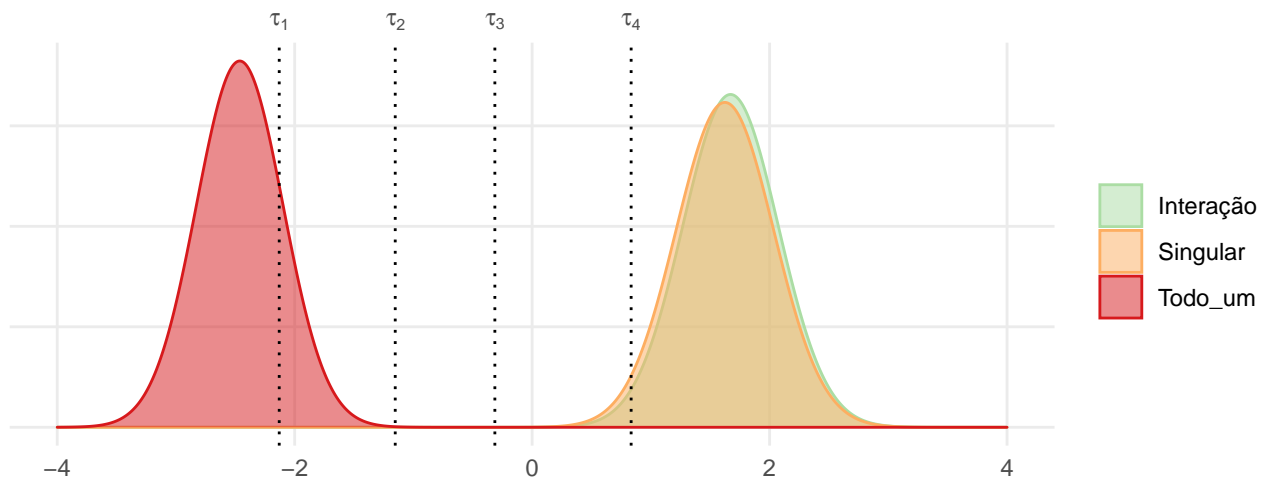
```



```

aes(color = fatores)) +
geom_vline(xintercept = fixef(m2)[1:4, 1], linetype = "dotted") +
scale_x_continuous(
  sec.axis = dup_axis(breaks = fixef(m2)[1:4, 1] %>%
    as.double(),
    labels = parse(text = stringr::str_c("tau[", 1:4, "]")))) +
scale_fill_manual(values = legend_pal, name = NULL) +
scale_color_manual(values = legend_pal, name = NULL) +
labs(y = "", x = "", fill = "") +
theme_bw() +
theme(axis.ticks.y = element_blank(),
      axis.ticks.x = element_blank(),
      axis.text.y = element_blank(),
      plot.background = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank())

```



Interpretando os resultados...

A fim de entender os resultados, primeiro devemos lembrar os contrastes do modelo. Neste caso, estamos usando ‘dummy code’ (0 e 1), também chamado de ‘treatment contrasts’. Não vamos entrar em detalhes aqui sobre outros tipos de contrastes. Para uma discussão introdutória, veja o excelente material disponível [aqui](#).

```
contrasts(dados$Num)
```

```
##      SG
## PL    0
## SG    1
```

```
contrasts(dados$Ordem)
```

```
##      todo-um
## um-todo      0
## todo-um      1
```

As categorias codificadas com 0 (zero) formam a base, a referência contra as quais as demais serão contrastadas. Neste caso, um_todo (0) PL (0) é a categoria de referência. Logo, podemos entender assim as estimativas do modelo:

- O preditor ‘Ordem: todo_um’ = -2.4 logOdds indica o contraste ‘todo_um (1) PL (0)’ × ‘um_todo (0) PL (0)’. Ou seja, mantendo-se o número constante, o efeito da ‘ordem’: todo_um aumenta a

probabilidade de se DESCER na escala, aumenta a probabilidade de se dar uma resposta mais abaixo na escala.

- O preditor ‘Número: singular’ = +1.6 logOdds indica o contraste ‘um_todo (0) SG (1)’ x ‘um_todo (0) PL (0)’. Ou seja, mantendo-se a ordem constante, o efeito do ‘número’: singular aumenta a probabilidade de se SUBIR na escala, aumenta a probabilidade de se dar uma resposta mais acima na escala.
- A interação ‘Ordem x Número (todo_um singular)’ = +1.6 logOdds indica que os níveis da variável ‘Ordem’ interagem com os níveis da variável ‘Número’, basicamente indicando que: o nível singular aumenta a probabilidade de SUBIR na escala com todo_um; o nível plural aumenta a probabilidade de DESCER na escala com todo_um.

IMPORTANTE! O fato de haver um efeito de interação aponta para olharmos com cuidado para os efeitos de Ordem e de Número. Se não houvesse interação, o efeito de um nível da variável seria o mesmo para ambos os níveis da outra, mas como há, os efeitos de ‘ordem’ e de ‘número’ indicam contrastes particulares. Ver explicação abaixo, se quiser saber mais sobre esse ponto.

Adendo 1: de logOdds para Odds para probailidades

Vamos começar com um vetor de probabilidades (valores entre 0 a 1, portanto), e vamos transformá-lo em logaritmo da chance (LogOdds). É esse valor que o modelo com `family = 'logit'` nos dá. Em seguida, vamos converter os LogOdds em Odds (chance) e depois em probabilidades.

```
a = c(0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1)
log_a = round(qlogis(a), 2)
odds = round(exp(log_a), 2)
# Abaixo, o mesmo que 'boot::inv.logit(log_a)'
backTo_a = round(exp(log_a)/(1+exp(log_a)), 2)
tags = c("extreme", rep("border", 4), "center", rep("border", 4), "extreme")

(myOddsGraph <- data.frame(a, log_a, odds, backTo_a, tags))
```

```
##      a log_a odds backTo_a  tags
## 1  0.0  -Inf 0.00      0.0 extreme
## 2  0.1 -2.20 0.11      0.1  border
## 3  0.2 -1.39 0.25      0.2  border
## 4  0.3 -0.85 0.43      0.3  border
## 5  0.4 -0.41 0.66      0.4  border
## 6  0.5  0.00 1.00      0.5  center
## 7  0.6  0.41 1.51      0.6  border
## 8  0.7  0.85 2.34      0.7  border
## 9  0.8  1.39 4.01      0.8  border
## 10 0.9  2.20 9.03      0.9  border
## 11 1.0   Inf Inf      NaN extreme
```

A partir dessa tabela, temos o seguinte gráfico resumitivo:

Odds	0	0.11	0.25	0.43	0.66	1	1.51	2.34	4.01	9.03	Inf
						\uparrow					
						$\exp(\log Odds)$					
LogOdds	-Inf	-2.2	-1.39	-0.85	-0.41	0	0.41	0.85	1.39	2.2	Inf
						\uparrow					
						$\exp(\log Odds)/(1+\exp(\log Odds))$					
						\downarrow					
Probabilidades	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
						\downarrow					
						qlogis(prob.)					

Olhando com atenção para esses dados, o que percebemos é que:

- Probabilidade de 50% é igual a 0 (zero) logOdds que é igual a chance de 1/1 (1 para 1, ou seja, uma chance de sucesso para uma de fracasso);
- Probabilidade de 80% é igual a 1.39 logOdds que é igual a chance de 4/1 (4 chances de sucesso para uma de fracasso);
- Probabilidade de 10% é igual a -2.2 logOdds que é igual a chance de 0.11 ou 1/9 (1 chance de sucesso para 9 de fracasso). A fim de converter probabilidade em Odds, basta dividir a probabilidade de sucesso pela de fracasso. Por exemplo: $\frac{0.1}{0.9} = 0.11$ ou $\frac{0.2}{0.8} = 0.25$ (uma probabilidade de 20% de sucesso é o mesmo que uma chance de 0.25 ou 1/4 - uma chance de sucesso para 4 de fracasso.)

Mas por que usar logOdds se fica tão difícil de compreender? O motivo é que probabilidade é um valor limitado entre 0 e 1 enquanto LogOdds é um valor que vai de -Inf a +Inf:

```
qlogis(0) # 0%
## [1] -Inf
qlogis(1) # 100%
## [1] Inf
qlogis(.999999999) # um valor muito próximo de 100% pode ser estimado em logOdds
## [1] 23.02585
qlogis(.000000001) # um valor muito próximo de 0% pode ser estimado em logOdds
## [1] -23.02585
```

Assim sendo, uma outra maneira de ler as estimativas do modelo é interpretar as probabilidades (ou chances) correspondentes na tabela:

```
round(fixos.m, 3)

##               Estimativas Est.Error  lower  upper Odds  Probs
## ORDEM: todo_um          -2.463    0.365 -3.194 -1.788 0.085 -92.148
## NÚMERO: singular         1.623    0.411  0.824  2.437 5.070 83.525
## INTERAÇÃO: ordem x número    1.671    0.401  0.875  2.486 5.318 84.172
```

Ao se mudar de uma frase com a ordem 'um_todo' para uma frase com a ordem 'todo_um', há cerca de 92% de probabilidade de a resposta ser 1 nível abaixo na escala (0.08 ou 1 chance para SUBIR e 12 para

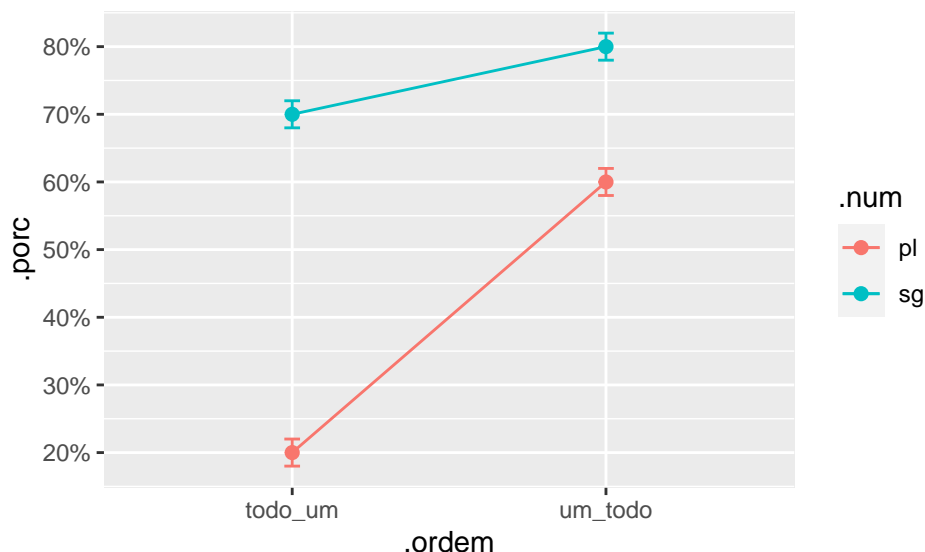
DESCER). Por exemplo, se as pessoas em geral marcam ‘concordo’ em uma frase com ‘um_todo’, há 92% de probabilidade de elas marcarem abaixo, ou seja, ‘neutro’, em uma frase com ‘todo_um’.

Se as pessoas em geral marcam ‘concordo’ em uma frase com ‘plural’, há 83% de probabilidade (ou 5 chances para SUBIR e 1 para DESCER) de elas marcarem acima, ou seja, ‘concordo totalmente’ em uma frase com anáfora ‘singular’. *Grosso modo*, é isso que o modelo está dizendo.

Adendo 2: entendendo o efeito de interação

Observe o gráfico abaixo, cujos valores foram inventados, apenas para ilustração:

```
data.frame(  
  .ordem = c("um_todo", "um_todo", "todo_um", "todo_um"),  
  .num = c("sg", "pl", "sg", "pl"),  
  .porc = c(.8, .6, .7, .2),  
  .sd = c(rep(.02, 4))) %>%  
  ggplot(aes(x = .ordem, y = .porc, group = .num, color = .num)) +  
  geom_errorbar(aes(ymin = .porc-.sd, ymax = .porc+.sd), width = .05) +  
  geom_point(size = 2) + geom_line() +  
  scale_y_continuous(labels = scales::label_percent(accuracy = 1L),  
    breaks = seq(from = 0, to = 1, by = .1))
```



Observe que mudar de plural (0) para singular (1) (ou vice-versa):

- Quando a ordem é ‘um_todo’, a mudança é de 20% (80 - 60) ou (60-80 = -20)
- Quando a ordem é ‘todo_um’, a mudança é de 50% (70 - 20) ou (20-70 = -50)
- A diferença é, portanto: 50 - 20 = 30 ou -50 - (-20) = -30

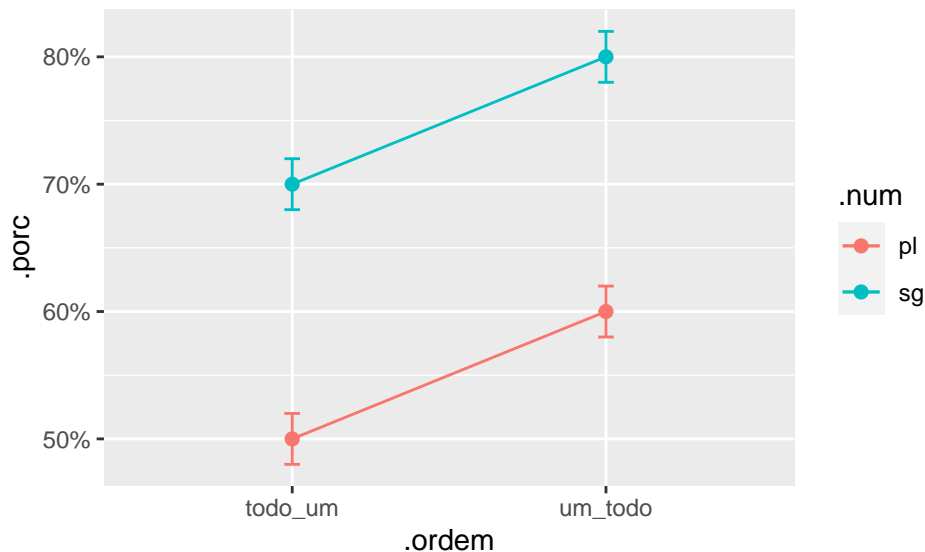
Observe também que mudar de ‘um_todo’ (0) para ‘todo_um’ (1) (ou vice-versa):

- Quando o número é ‘singular’, a mudança é de 10% (80 - 70) ou (70 - 80 = -10)
- Quando o número é plural, a mudança é de 40% (60 - 20) ou (20 - 60 = -40)
- A diferença é, portanto: 40 - 10 = 30 ou -40 - (-10) = -30

Repare que o “efeito de interação” é sempre o mesmo: a diferença das diferenças = 30% (ou -30% se for no sentido contrário, de 1 para 0). Por isso, não importa o modelo que você ajuste para esses dados, com quaisquer contrastes o valor da interação será o mesmo, mudando apenas o sinal em alguns casos!

Vamos fazer uma pequena mudança no gráfico, trocando aquele .2 por .5:

```
data.frame(
  .ordem = c("um_todo", "um_todo", "todo_um", "todo_um"),
  .num = c("sg", "pl", "sg", "pl"),
  .porc = c(.8, .6, .7, .5),
  .sd = c(rep(.02, 4))) %>%
  ggplot(aes(x = .ordem, y = .porc, group = .num, color = .num)) +
  geom_errorbar(aes(ymin = .porc-.sd, ymax = .porc+.sd), width = .05) +
  geom_point(size = 2) + geom_line() +
  scale_y_continuous(labels = scales::label_percent(accuracy = 1L),
    breaks = seq(from = 0, to = 1, by = .1))
```



Ao fazer isso, o que vemos é que o resultado, ao se mudar de ‘plural’ para ‘singular’ não é afetado pela ‘ordem’, que é 20% em ambos os casos: $20 - 20 = 0\%$ de alteração de um nível para outro; O resultado ao se mudar de ‘um_todo’ para ‘todo_um’ não é afetado pelo ‘número’, que é 10% em ambos os casos: $10 - 10 = 0\%$ de alteração de um nível para outro. Logo: não há efeito de interação!

Esse gráfico mostra bem, ainda, o que um modelo de regressão faz ao expor as ‘estimativas’. Se não há interação, o efeito de ‘singular’ (aumento de 20%) é o mesmo tanto em ‘todo_um’ quanto em ‘um_todo’. Esse pensamento é difícil para quem vem de um paradigma mental típico da Análise de Variância (Anova), que pensa sempre em contrastes de todas as categorias contra todas as categorias, como se faz em um teste *post-hoc*. O modelo de regressão contrasta apenas o que for estritamente necessário, não contrastando o que for redundante. Se, por exemplo, temos 3 categorias (A, B e C), ele apenas fará dois contrastes, porque o terceiro é inferível desses dois. Um exemplo pode deixar isso mais claro: se sabemos que Pedro é 5 anos mais velho do que João e que Maria é 2 anos mais velha do que Pedro, então não precisamos calcular a diferença de idade entre João e Maria, pois podemos inferi-la dos contrastes já dados.

No caso de 4 condições nos dados com que estamos trabalhando, ao fixar uma delas no intercepto, o modelo precisa apenas contrastar duas delas e estimar a interação para ter tudo o que é necessário. Se não há interação, o efeito do número obtido do contraste de duas condições é o mesmo para o contraste entre as demais (e o modelo não precisa perder tempo calculando esse valor). Se há interação, então o efeito do número obtido do contraste de duas condições não é o mesmo para as demais e o modelo está mostrando nas estimativas o contraste individual entre elas.

Interpretando os resultados à luz da teoria

Frase: O(a) estilista mostrou...	Anáfora	Ordem	Número
uma camisa para todo comprador que [...],	mas ela não...	Um_todo	Singular
uma camisa para todo comprador que [...],	mas elas não...	Um_todo	Plural
toda camisa que [...] para um comprador,	mas ele não...	Todo_um	Singular
toda camisa que [...] para um comprador,	mas eles não...	Todo_um	Plural

- Sentenças duplamente quantificadas com anáfora singular são aceitas pelos falantes de Português Brasileiro se comparadas com sentenças com anáfora plural (maior porcentagem de julgamentos concordo totalmente e concordo) - Efeito de NÚMERO Singular do modelo: 83%;
- No que diz respeito à ordem dos quantificadores, os resultados indicam que sentenças com a ordem todo_um não são aceitas pelos falantes de português - Efeito de ORDEM: Todo_um: -92%. Este efeito, contudo, foi modulado pela interação, mostrando que se mantém para a condição Todo_um × Plural, mas não para Todo_um × Singular - Efeito de INTERAÇÃO: Todo_um × Singular: 84%.

Investigando os dados do Mario

Carregando os dados e fazendo as alterações na tabela:

```
final_table <- read.csv(
  "https://raw.githubusercontent.com/igor-de-costa/ModelosOrdinais/main/AnaliseEscalaLikert_Mario.csv",
  stringsAsFactors = T)

# Preparando a tabela
final_table$escala <- as.ordered(final_table$escala) # variável resposta ordenada
```

Preparando a estatística descritiva

Calculando valores absolutos

```
resp_escala_abso <- final_table %>%
  filter(!is.na(escala)) %>% # Excluir células vazias NA
  group_by(Tamanho, Número, escala) %>% # agrupa pelas condições relevantes
  tally() %>% # Faz a contagem
  spread(escala, n)

colnames(resp_escala_abso) <- c("Tamanho", "Número",
  "Nada_natural", "Pouco_natural", "Neutro",
  "Muito_natural", "Totalmente_natural")
```

Calculando valores percentuais

```
resp_escala_perc <- final_table %>%
  filter(!is.na(escala)) %>%
  group_by(Tamanho, Número, escala) %>%
  tally() %>%
  mutate(perc=n/sum(n)) %>%
  select(-n) %>%
  spread(escala, perc)

colnames(resp_escala_perc) <- c("Tamanho", "Número",
  "Nada_natural", "Pouco_natural", "Neutro",
  "Muito_natural", "Totalmente_natural")
```

E aqui vamos fazer o gráfico de barras empilhadas:

```
# Dividir o meio da escala (os julgamentos "Neutro"):
dados_meio <- resp_escalaperc %>%
  mutate(c1 = Neutro / 2,
         c2 = Neutro / 2) %>%
  select(Tamanho, Número,
         Nada_natural, Pouco_natural, c1, c2, Muito_natural, Totalmente_natural) %>%
  gather(key = Escolha, value = perc, 3:8)

# Separando a escala em dois conjuntos, o "alto" e o "baixo":
meio_alto <- dados_meio %>%
  filter(Escolha %in% c("Totalmente_natural", "Muito_natural", "c2")) %>%
  # Níveis na ordem normal!
  mutate(Escolha = factor(Escolha,
                          levels = c("Totalmente_natural", "Muito_natural", "c2")))

meio_baixo <- dados_meio %>%
  filter(Escolha %in% c("c1", "Pouco_natural", "Nada_natural")) %>%
  # Níveis na ordem inversa!
  mutate(Escolha = factor(Escolha,
                          levels = c("Nada_natural", "Pouco_natural", "c1")))

# Estabelecimento de uma paleta de cores para organização!
legend_pal <- c("#2B83BA", "#ABDDA4", "#FFFFBF", "#FFFFBF", "#FDAE61", "#D7191C")
legend_pal <- gsub("#FFFFBF", "#9C9C9C", legend_pal)
names(legend_pal) <- c("Totalmente_natural", "Muito_natural", "c1",
                      "c2", "Pouco_natural", "Nada_natural")
```

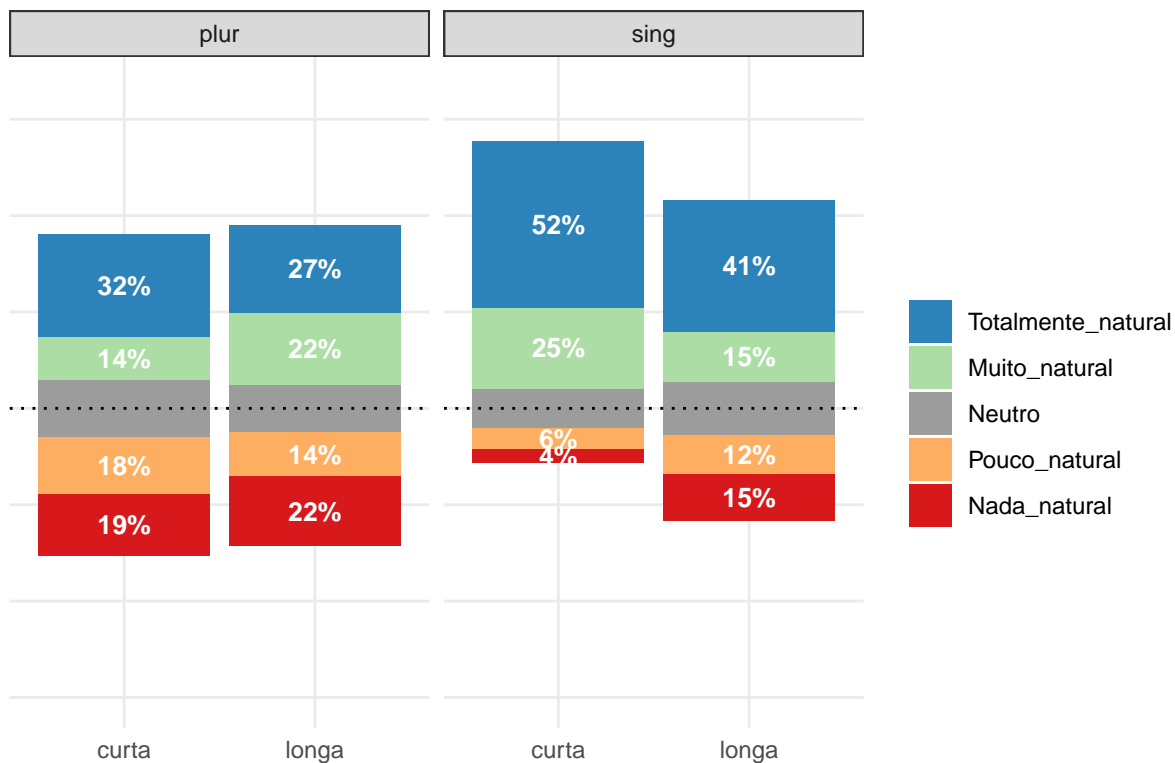
Produzindo o gráfico, finalmente!

```
ggplot() +
  geom_bar(data = meio_alto, aes(x = Tamanho,
                                y=perc, fill = Escolha), stat="identity") +
  geom_bar(data = meio_baixo, aes(x = Tamanho,
                                y=-perc, fill = Escolha), stat="identity") +
  #-----
  # Essa parte do código é um pouco complicada...
  # Serve apenas para colocar os valores dentro das barras
  geom_text(data = meio_alto,
            aes(x = Tamanho, y=perc, group = Escolha,
                alpha = Escolha,
                label = scales::percent(perc, accuracy = 1)),
            color = "white",
            size = 3.5, position = position_stack(vjust = .5),
            fontface = "bold") +
  geom_text(data = meio_baixo,
            aes(x = Tamanho, y=-perc, group = Escolha,
                alpha = Escolha,
                label = scales::percent(perc, accuracy = 1)),
            size = 3.5, position = position_stack(vjust = .5),
            color = "white",
            fontface = "bold") +
  # Não mostrar as % no meio da escala!
  scale_alpha_manual(values = c("c1" = 0, "c2" = 0,
```

```

    "Muito_natural" = 1, "Totalmente_natural" = 1,
    "Nada_natural" = 1, "Pouco_natural" = 1),
    guide = 'none') +
#-----
geom_hline(yintercept = 0, color = c("black"), linetype = "dotted") +
scale_fill_manual(values = legend_pal,
    breaks = c("Totalmente_natural", "Muito_natural", "c2",
    "Pouco_natural", "Nada_natural"),
    labels = c("Totalmente_natural", "Muito_natural",
    "Neutro", "Pouco_natural", "Nada_natural")) +
labs(x = "", y = "", fill="") +
facet_wrap(~Número) +
theme_bw() +
scale_y_continuous(labels = function(x) percent(abs(x)),
    limits = c(-0.9, 1),
    breaks = seq(from = -0.9, to = 1, by = .3)) +
theme(axis.ticks.y = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    plot.background = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank()) +
ggtitle("")

```



Aplicando um modelo ordinal aos dados usando 'Flat priors'

```

# Ativando acesso a todos os núcleos do processador
options(mc.cores = parallel::detectCores())

```



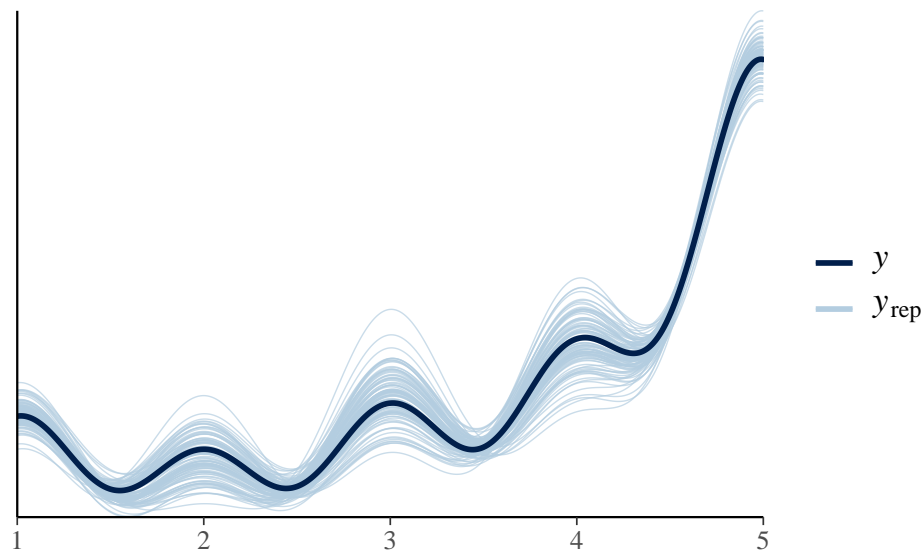
```
# Ajustando o modelo
ordinal_mario <- brm(escala ~ Tamanho*Número +
  (1+Tamanho*Número|Part) + (1+Tamanho*Número|num),
  data = final_table,
  family = cumulative(link = "logit", threshold = "flexible"),
  silent = 2, refresh = 0)

fixef(ordinal_mario)[5:7,]
```

```
##              Estimate Est.Error      Q2.5      Q97.5
## Tamanholonga      -0.3918196   1.421917  -3.483902   2.4459967
## Númerosing         6.3843323   2.882880   2.622058  13.5891708
## Tamanholonga:Número  -3.4523106   2.472428  -9.383174   0.1002761
```

Checando a qualidade do ajuste...

```
pp_check(ordinal_mario, ndraws = 100)
```



Extrair os coeficientes dos fatores fixos e intervalos de confiança calculados

```
fixos.m<-fixef(ordinal_mario)[5:7,]

# Preparar um data frame com esses dados
colnames(fixos.m)<-c("Estimativas", "Est.Error", "lower", "upper")
fixos.m<-as.data.frame(fixos.m)

# Cálculo da chance e das probabilidades
fixos.m<-fixos.m %>%
  mutate(Odds=exp(Estimativas)) %>%
  mutate(Probs=if_else(Estimativas<0, # Se logOdds menor do que zero, então:
    (exp(Estimativas)/(1 + exp(Estimativas))-1)*100, # Probab. de descer na escala
    exp(Estimativas)/(1 + exp(Estimativas))*100)) # Probab. de subir na escala

rownames(fixos.m)<-c("TAMANHO: longa", "NÚMERO: singular", "INTERAÇÃO: longa x sg")

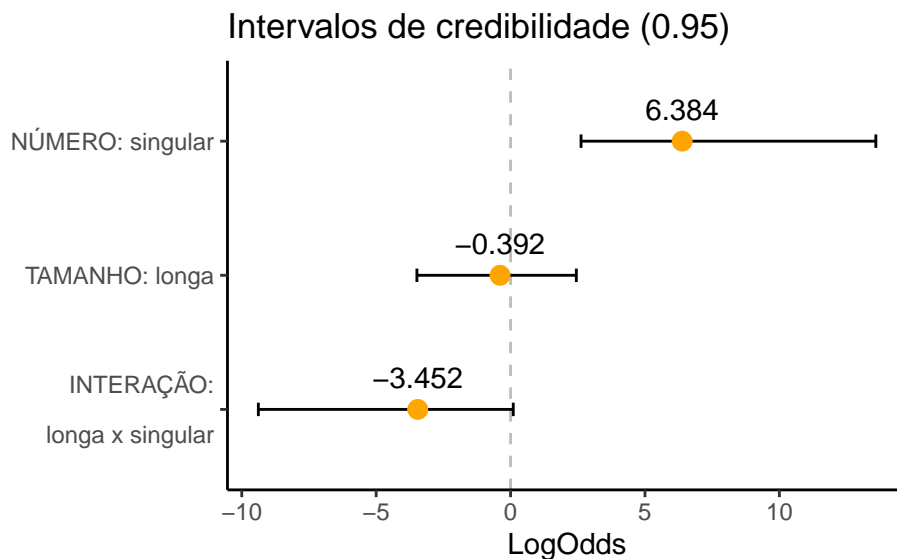
round(fixos.m, 3)
```

```
##              Estimativas Est.Error  lower  upper  Odds  Probs
```

```
## TAMANHO: longa          -0.392      1.422 -3.484  2.446  0.676 -59.672
## NÚMERO: singular        6.384      2.883  2.622 13.589 592.489 99.832
## INTERAÇÃO: longa x sg   -3.452      2.472 -9.383  0.100  0.032 -96.930
```

Plotar um gráfico para publicação...

```
fixos.m %>%
  ggplot(aes(x=reorder(rownames(fixos.m), Estimativas),
                y=Estimativas))+
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey")+
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.1,
                position=position_dodge(.9))+
  geom_point(color="orange", size = 3) +
  geom_text(aes(label=round(Estimativas, 3)), vjust=-1)+
  # ao mudar a ordem é preciso mudar aqui
  scale_x_discrete(labels=c("INTERAÇÃO:\n\nlonga x singular",
                            "TAMANHO: longa",
                            "NÚMERO: singular"))+
  labs(y = "LogOdds", x = "") +
  ggtitle("Intervalos de credibilidade (0.95)") +
  coord_flip()+theme_classic()
```



E também o gráfico com a distribuição estimada da posterior para cada fator...

```
(posterior.draws <- ordinal_mario %>%
  spread_draws(`b_.*`, regex = TRUE) %>%
  summarise_draws())
```

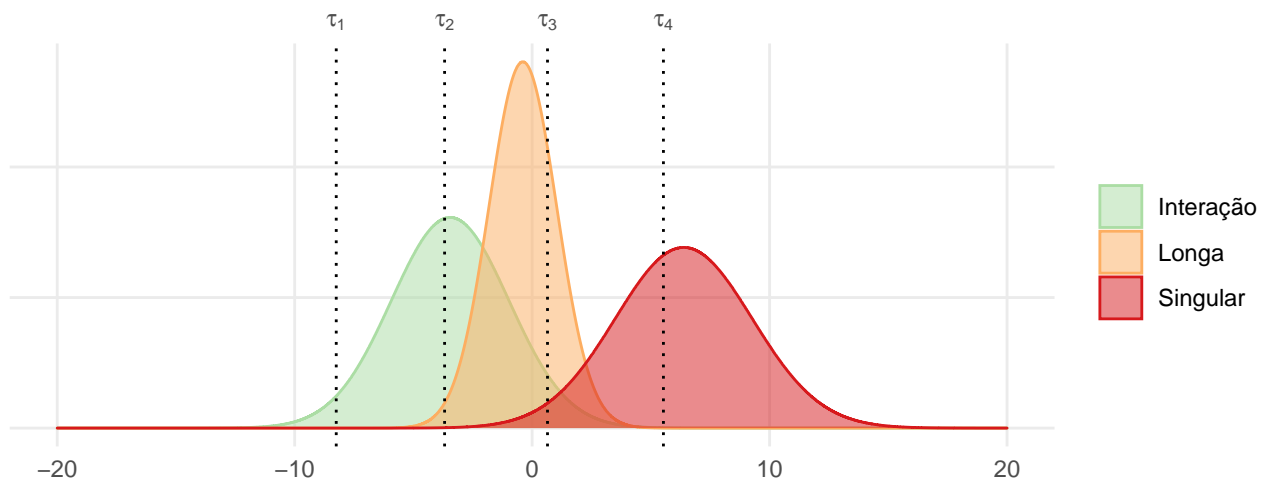
```
## # A tibble: 7 x 10
##   variable      mean median    sd  mad    q5    q95  rhat ess_bulk ess_tail
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 b_Intercept~ -8.25 -7.64  3.27 2.82 -14.2 -4.19  1.04   98.3  263.
## 2 b_Intercept~ -3.69 -3.39  1.66 1.45  -6.80 -1.61  1.03  133.  385.
## 3 b_Intercept~  0.648 0.555  1.04 0.835 -0.783  2.31  1.01  874.  748.
## 4 b_Intercept~  5.53  5.05  2.45 1.88   2.66  9.80  1.03  110.  248.
## 5 b_Tamanholon~ -0.392 -0.331  1.42 1.16  -2.84  1.80  1.01  843.  792.
## 6 b_Numerosing  6.38  5.79  2.88 2.21   3.01 11.6  1.03  111.  259.
## 7 b_Tamanholon~ -3.45 -3.04  2.47 1.87  -7.90 -0.396 1.01  259.  394.
```

```

legend_pal <- c("#ABDDA4", "#FDAE61", "#D7191C")

data.frame(
  fatores = c("Longa", "Singular", "Interação"),
  posterior.draws[5:7,2:4]) %>%
  expand(nesting(fatores, mean, sd),
    y = seq(from = -20, to = 20, by = 0.001)) %>%
  mutate(d = dnorm(y, mean, sd)) %>% # PDF da distribuição normal
  ggplot(aes(x = y, y = d, fill = fatores)) +
  geom_density(stat = "identity", alpha = 0.5,
    aes(color = fatores)) +
  geom_vline(xintercept = fixef(ordinal_mario)[1:4, 1], linetype = "dotted") +
  scale_x_continuous(
    sec.axis = dup_axis(breaks = fixef(ordinal_mario)[1:4, 1] %>%
      as.double(),
      labels = parse(text = stringr::str_c("tau[", 1:4, "]")))) +
  scale_fill_manual(values = legend_pal, name = NULL) +
  scale_color_manual(values = legend_pal, name = NULL) +
  labs(y = "", x = "", fill = "") +
  theme_bw() +
  theme(axis.ticks.y = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    plot.background = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank())

```



Interpretando os resultados...

Lembrando nossos contrastes...

```
contrasts(final_table$Tamanho)
```

```
##      longa
## curta    0
## longa    1
```

```
contrasts(final_table$Número)
```

```
##      sing
```

```
## plur    0
## sing    1
```

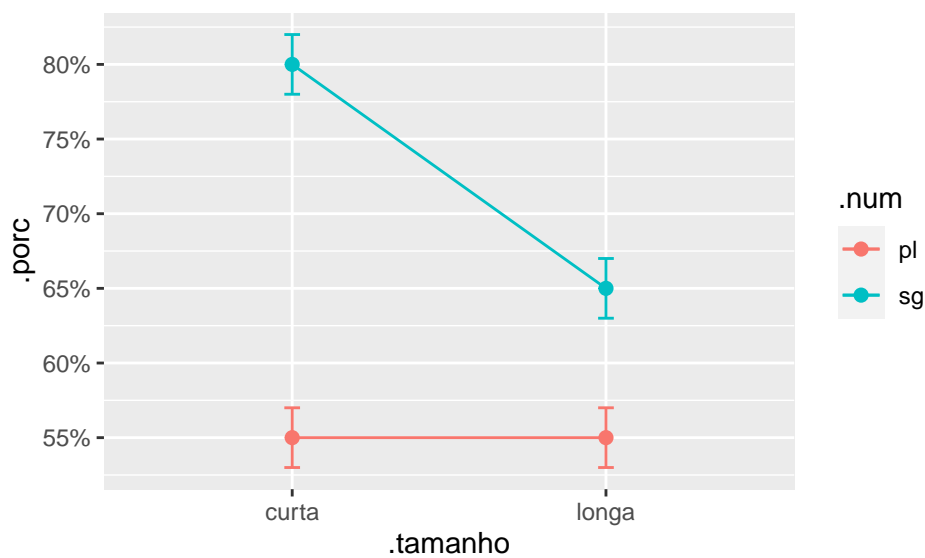
- O fator significativo número singular (+6 logOdds) indica o contraste Curta (0) sg (1) x Curta (0) pl (0): singular aumenta a chance de se SUBIR na escala...
- O efeito não significativo de tamanho indica que nada se pode inferir quanto ao tamanho. O intervalo de credibilidade cortando a linha do zero indica que o parâmetro populacional pode tanto ser menor quanto maior do que zero... ou mesmo zero. Mantendo-se o plural constante, ao se mudar de longa para curta, não se pode inferir que há diferença.
- O intervalo de credibilidade para a interação ora corta a linha do zero ora não. Apenas por motivos pedagógicos, vamos assumir que ele seja significativo. Se é assim, então o nível singular aumenta MUITO a probabilidade de SUBIR na escala com curta e aumenta POUCO ou NADA com longa. Mas por que o valor em logOdds é negativo (-3 logOdds)? Como falamos antes, o resultado da interação é sempre o mesmo, o que muda é apenas o sinal.

IMPORTANTE! Em uma abordagem bayesiana, em geral estamos mais preocupados com a estimativa do modelo e com o grau de confiança que temos nela e menos preocupados de essa estimativa ser ou não significativa, ou seja, menos preocupados com o fato de o intervalo de credibilidade conter ou não zero (Ver a [aula 03 do Prof. Shravan Vasishth](#), mais especificamente a partir de 2h27min13seg).

De novo um adendo para compreender a interação...

Vamos olhar para o gráfico abaixo:

```
# Valores abaixo são inventados, apenas para ilustração
data.frame(.tamanho = c("curta", "curta", "longa", "longa"),
           .num = c("sg", "pl", "sg", "pl"),
           .porc = c(.8, .55, .65, .55),
           .sd = c(rep(.02, 4))) %>%
  ggplot(aes(x = .tamanho, y = .porc, group = .num, color = .num)) +
  geom_line() +
  geom_errorbar(aes(ymin = .porc-.sd, ymax = .porc+.sd), width = .05) +
  geom_point(size = 2) +
  scale_y_continuous(labels = scales::label_percent(accuracy = 1L),
                     breaks = seq(from = 0, to = 1, by = .05))
```



Quando se muda de plural (0) para singular (1):

- na curta: $55 - 80 = -25$
- na longa: $55 - 65 = -10$
- Diferença = $-25 - (-10) = -15\%$ (Sinal negativo)

Quando se muda de singular (1) para plural (0):

- na curta: $80 - 55 = 25$
- na longa: $65 - 55 = 10$
- Diferença = $25 - 10 = 15\%$ (Sinal positivo)

Quando se muda de curta (0) para longa (1):

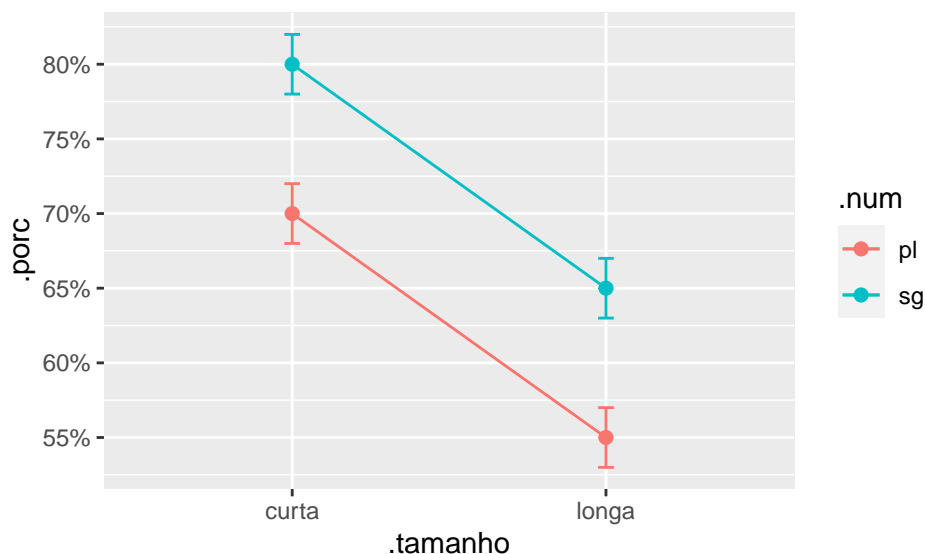
- no singular $80 - 65 = 15$
- no plural $60 - 60 = 0$
- Diferença = 15% (Sinal positivo)

Quando se muda de longa (1) para curta (0):

- no singular $65 - 80 = -15$
- no plural $60 - 60 = 0$
- Diferença = -15% (Sinal negativo)

Se você quiser ver como fica sem o efeito de interação, basta mudar o segundo .55 para .7:

```
# Valores abaixo são inventados, apenas para ilustração
data.frame(.tamanho = c("curta", "curta", "longa", "longa"),
           .num = c("sg", "pl", "sg", "pl"),
           .porc = c(.8, .7, .65, .55),
           .sd = c(rep(.02, 4))) %>%
  ggplot(aes(x = .tamanho, y = .porc, group = .num, color = .num)) +
  geom_line() +
  geom_errorbar(aes(ymin = .porc-.sd, ymax = .porc+.sd), width = .05) +
  geom_point(size = 2) +
  scale_y_continuous(labels = scales::label_percent(accuracy = 1L),
                     breaks = seq(from = 0, to = 1, by = .05))
```



Interpretando os resultados à luz da teoria

A partir de agora, o autor pode interpretar os resultados em vista das suas considerações teóricas, olhando para os contrastes que planejou, suas sentenças e categorizações respectivas:

Frase	Tamanho	Número
No escritório da firma desapareceu um documento sigiloso.	Curta	Singular
No escritório da firma desapareceu uns documentos sigilosos.	Curta	Plural
No canteiro do jardim desabrochou repentinamente uma azaleia belíssima.	Longa	Singular
No canteiro do jardim desabrochou repentinamente umas azaleias belíssimas.	Longa	Plural

- O singular aumenta a probabilidade de respostas ACIMA na escala, indicando que os falantes aceitam mais as sentenças com número singular do que com número plural - Efeito de NÚMERO: Singular: 99%;
- Esse efeito, porém, foi modulado pela interação: enquanto o plural tem o mesmo efeito nos falantes, independentemente do TAMANHO (efeito não significativo de TAMANHO: Longa), o singular é mais aceito com curtas do que com longas: Efeito de INTERAÇÃO: Longa \times Singular: -96%.

Ajustando um modelo com priors informativas (ou nem tanto)

NOTA DO AUTOR: Recomendo não discutirmos esses pontos agora. Acho que poderíamos nos aprofundar mais antes de avançarmos aqui. Comecei meus estudos pelas aulas disponíveis no Youtube do Prof. Shravan Vasishth. [A primeira aula](#) me parece fundamental para entender o que é uma função de verossimilhança (*Likelihood function*). Relembrem também as [aulas do Gilberto](#), em que ele trabalhou esse ponto com bastante atenção, dando vários exemplos. Sem esse conhecimento fica muito difícil avançar para [a segunda aula](#), em que ele discute a noção de priors, fundamentando os princípios matemáticos por trás dessa noção. Relembrem também a última aula do Gilberto, em que ele trabalhou a noção de priors, com um exemplo de uma moeda não viciada. Da [terceira aula](#), eu tirei a citação abaixo (1h38min08seg) que justifica o fato de, por enquanto, podermos ficar com esse modelo mais simples, sem priors tão informativas:

In general, we don't want our priors to have too much influence on our posterior. This is unless we have *very* good reasons for having informative priors, such as very small sample and/or a lot of prior information; an example would be if we have data from an impaired population, which makes it hard to increase our sample size.

[Andrew Gelman \(1h14min\)](#) parece ter uma postura semelhante:

In my analysis I'll typically - maybe because of my training as statistician - start with a super weak prior and hope It works and if It doesn't I'll start putting prior information in.

Eu ainda estou estudando o assunto e podemos ter uma conversa informal sobre o tema no dia da apresentação.