

Análise Fisiológica (CDA) e Estrutura Hierárquica do Afeto

Frederico Pedrosa

2025-12-05

Contents

1 Introdução e Objetivos do Estudo	1
1.1 O que estamos testando?	1
2 Análises estatísticas	2
2.1 Carregar pacotes	2
2.2 Carregar os dados	2
2.3 Verificação de Hipótese de um FG de Intensidade Fisiológica	3
2.4 Análise de Componentes Princípios	5
2.5 Validação do Fator Geral por Categoria	6
2.6 Plot para verificar as interações	10
3 Existem perfis latentes?	10

1 Introdução e Objetivos do Estudo

O presente estudo investiga a estrutura latente da resposta fisiológica emocional e sua relação com as dimensões afetivas de Valência e Arousal. Utilizando o conjunto de dados **EMO2018-SCR** (Juuse et al., 2024), analisamos a condutância da pele (*Galvanic Skin Response - GSR*) de **102 participantes** expostos a estímulos multimodais (expressões faciais e palavras) que evocam diferentes categorias emocionais.

O objetivo central é testar a hipótese da **Estrutura Hierárquica do Afeto** no domínio biológico. Argumentamos que, antes da diferenciação em dimensões de valência e arousal, existe um **Fator Geral de Intensidade Fisiológica** que governa a mobilização simpática.

1.1 O que estamos testando?

Esta análise busca responder a três perguntas para a validação da hipótese:

- Existência e Robustez do Fator Geral (PCA):** As diferentes métricas de condutância da pele convergem para uma única dimensão latente de Intensidade Fisiológica? Se a hipótese hierárquica estiver correta, este fator deve explicar a vasta maioria da variância compartilhada, discriminando linearmente estados de alta energia (ex: Raiva) de baixa energia (ex: Neutro).

Interação Contextual (LMER): A Intensidade Fisiológica responde ao Arousal normativo de forma linear (como prevê o modelo Circumplexo clássico) ou é dependente da Valência? Hipótese: Espera-se encontrar uma interação significativa, onde o efeito do Arousal na fisiologia é modulado pela qualidade hedônica do estímulo - dado que arousal foi o terceiro componente encontrado nos dados semânticos.

Tipologia da Reatividade (LPA): A intensidade fisiológica se manifesta em padrões qualitativamente distintos? Nova Hipótese: Utilizando Análise de Perfis Latentes (LPA), investigamos se o sistema nervoso simpático opera em “modos” discretos. Espera-se identificar um perfil de “Alta Magnitude” que seja predominantemente composto por emoções de alta mobilização (Raiva, Felicidade, Medo), segregando-as biologicamente do estado Neutro. Isso confirmaria que a Intensidade é a dimensão organizadora primária da resposta corporal.

2 Análises estatísticas

2.1 Carregar pacotes

```
library(dplyr)
library(readxl)
library(lmerTest)
library(janitor)
library(psych)
library(stringr)
library(ggeffects)
library(ggplot2)
library(performance)
library(emmeans)
library(tidyverse)
library(tidyLPA)
library(multcomp)
```

2.2 Carregar os dados

```
# =====
# 1. CARREGAR E PADRONIZAR DADOS FISIOLÓGICOS
# =====
dat4 <- read.csv("~/PIBIT2025/Circumplex/BasesDados/EM02018-SCR/dat4.csv")%>%
  janitor::clean_names() %>%
  # PADRONIZAÇÃO CRÍTICA:
  mutate(
    # Transformar emoção para MAIÚSCULO para bater com o excel
    emo_id_join = toupper(emo_id),
    # Garantir que exp_id (Face/Word) esteja padronizado (Title Case)
    exp_id_join = str_to_title(exp_id)
  )

# =====
# 2. CARREGAR E PADRONIZAR DADOS NORMATIVOS
# =====
normative_ratings <- read_excel("~/PIBIT2025/Circumplex/BasesDados/EM02018-SCR/arousal-valence-ratings..
```

```

normative_clean <- normative_ratings %>%
  # Selecionar apenas o que importa
  dplyr::select(EmoId, ExpId, Valence, Arousal) %>%
  mutate(
    # Transformar para MAIÚSCULO para bater com o dat4
    emo_id_join = toupper(EmoId),
    # Garantir Title Case ("Face", "Word")
    exp_id_join = str_to_title(ExpId)
  ) %>%
  # Renomear as métricas para não confundir
  rename(
    norm_valence = Valence,
    norm_arousal = Arousal
  )

# =====
# 3. CONECTANDO OS DOIS DATAS
# =====
# Usamos left_join para manter todos os 102 participantes do dat4
# Usamos DUAS chaves: Emoção E Tipo (Face/Word)
data_full_n102 <- left_join(dat4, normative_clean, by = c("emo_id_join", "exp_id_join"))

cat("Número de casos com Valência Normativa válida:", sum(!is.na(data_full_n102$norm_valence)), "\n")

## Número de casos com Valência Normativa válida: 7549

```

2.3 Verificação de Hipótese de um FG de Intensidade Fisiológica

Para a análise da intensidade fisiológica, optou-se exclusivamente pelas métricas derivadas da Análise de Decomposição Contínua (CDA - Continuous Decomposition Analysis) (Benedek & Kaernbach, 2010). Diferentemente dos métodos clássicos de *Trough-to-Peak (TTP)*, o CDA permite a separação matemática entre o nível tônico (atividade de fundo) e o componente fásico (resposta evocada pelo evento), além de corrigir distorções causadas pela superposição de respostas consecutivas.

Foram selecionadas cinco métricas que capturam diferentes facetas da magnitude da resposta: CDA.nSCR (frequência), CDA.AmpSum (soma das amplitudes), CDA.SCR (média da atividade fásica), CDA.ISCR (integral/área sob a curva) e CDA.PhasicMax (pico máximo). A alta correlação esperada entre estas métricas justifica a extração de um componente principal (PCA) para representar o Fator Geral de Intensidade Fisiológica.

2.3.1 Teste de pressupostos

```

# =====
# 4. ESCOLHA DAS VARIÁVEIS DE CDA
# =====
physio_vars <- c("cda_n_scr", "cda_amp_sum", "cda_scr", "cda_iscr", "cda_phasic_max")

# Preparar matriz sem NAs nas colunas fisiológicas
data_valid_physio <- data_full_n102 %>%

```

```

filter(!is.na(cda_scr)) # Garante que tem dados de SCR
physio_matrix <- scale(data_valid_physio[, physio_vars])

# =====
# 5. TESTE DE CORRELAÇÃO
# =====

normal <- MVN::mvn(physio_matrix, univariate_test = "AD")
print(normal$univariate_normality)

##          Test      Variable Statistic p.value Normality
## 1 Anderson-Darling    cda_n_scr   433.079 <0.001  Not normal
## 2 Anderson-Darling   cda_amp_sum  630.914 <0.001  Not normal
## 3 Anderson-Darling    cda_scr    621.107 <0.001  Not normal
## 4 Anderson-Darling   cda_iscr    621.085 <0.001  Not normal
## 5 Anderson-Darling cda_phasic_max  675.859 <0.001  Not normal

cor <- cor(physio_matrix, method = "spearman")
print(cor)

##           cda_n_scr  cda_amp_sum  cda_scr  cda_iscr  cda_phasic_max
## cda_n_scr 1.0000000  0.2866544  0.2701976  0.2702132   0.1485961
## cda_amp_sum 0.2866544  1.0000000  0.9680934  0.9680944   0.9319355
## cda_scr    0.2701976  0.9680934  1.0000000  0.9999978   0.9536508
## cda_iscr    0.2702132  0.9680944  0.9999978  1.0000000   0.9536532
## cda_phasic_max 0.1485961  0.9319355  0.9536508  0.9536532   1.0000000

# =====
# 6. Fatorabilidade
# =====

KMO(cor)

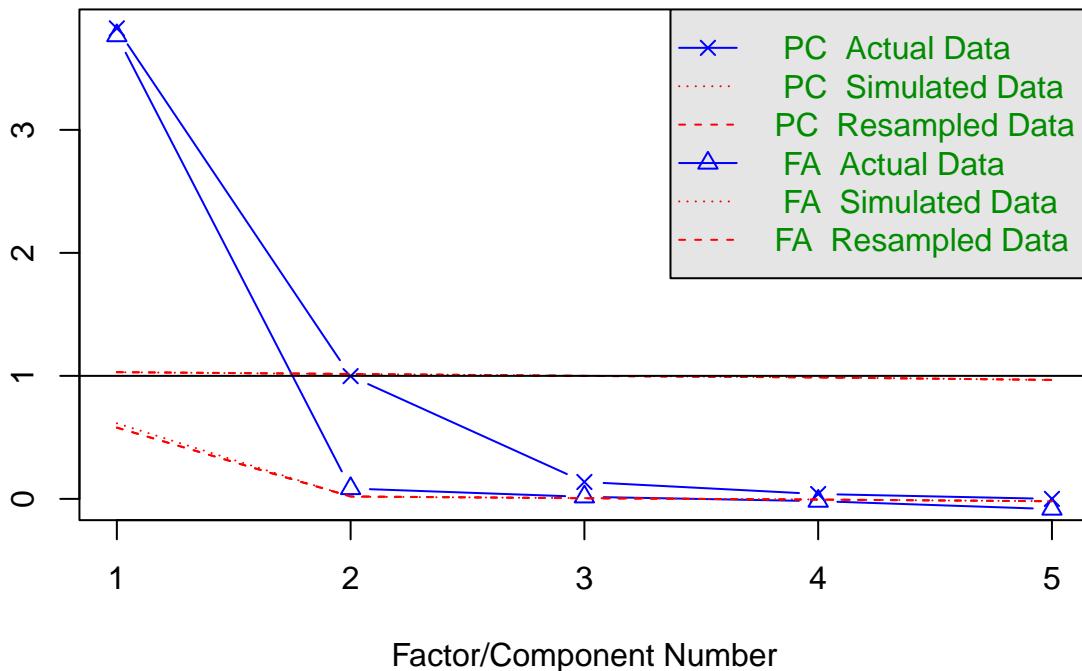
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor)
## Overall MSA =  0.83
## MSA for each item =
##           cda_n_scr  cda_amp_sum  cda_scr  cda_iscr  cda_phasic_max
##             0.58        0.98       0.74       0.74       0.94

psych::fa.parallel(physio_matrix)

```

eigenvalues of principal components and factor analysis

Parallel Analysis Scree Plots



```
## Parallel analysis suggests that the number of factors = 3 and the number of components = 1
```

2.4 Análise de Componentes Princípios

```
pca_res <- principal(physio_matrix, nfactors=1, rotate="none", scores=TRUE)
print(pca_res$loadings)
```

```
##
## Loadings:
##          PC1
## cda_n_scr    0.113
## cda_amp_sum   0.985
## cda_scr      0.988
## cda_iscr     0.988
## cda_phasic_max 0.944
##
##          PC1
## SS loadings   3.825
## Proportion Var 0.765

# Salvar o escore
data_valid_physio$physio_intensity <- as.numeric(pca_res$scores)
```

2.5 Validação do Fator Geral por Categoria

Antes de testar a interação dimensional, verificamos se o Fator Geral de Intensidade Fisiológica (PC1) discrimina adequadamente as categorias emocionais discretas. Espera-se que emoções de alta energia (ex: Raiva, Felicidade) apresentem escores significativamente maiores que estados de baixa energia (ex: Neutro, Tristeza).

```
# 1. Modelo Misto para Categorias
# Testamos se a intensidade muda conforme a emoção, controlando pelo participante
model_emotion_cat <- lmerTest::lmer(physio_intensity ~ emo_id + (1 | participant_id),
                                      data = data_valid_physio)

# 2. Tabela ANOVA
print("--- ANOVA: Diferença entre Emoções ---")

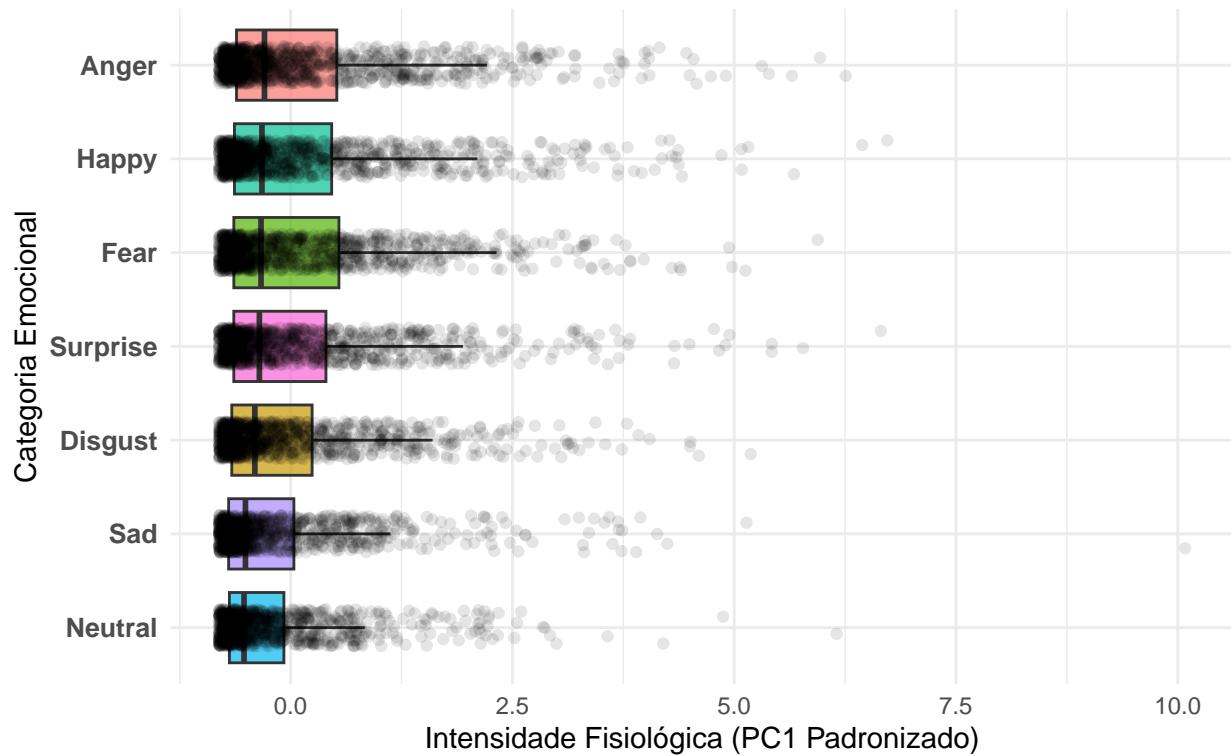
## [1] "--- ANOVA: Diferença entre Emoções ---"

anova(model_emotion_cat)

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## emo_id 153.02  25.504      6    7441.8 42.341 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# 3. Gráfico de Boxplot (A "Hierarquia Fisiológica")
# Ordenamos o eixo X pela mediana para mostrar a escada de intensidade
ggplot(data_valid_physio, aes(x = reorder(emo_id, physio_intensity, FUN = median),
                               y = physio_intensity,
                               fill = emo_id)) +
  # Boxplot para mostrar a distribuição
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  # Pontos ao fundo (jitter) para mostrar a densidade real dos dados
  geom_jitter(width = 0.2, alpha = 0.1, color = "black") +
  # Estética
  labs(
    title = "Hierarquia de Intensidade Fisiológica",
    subtitle = "",
    y = "Intensidade Fisiológica (PC1 Padronizado)",
    x = "Categoria Emocional"
  ) +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.y = element_text(size = 10, face = "bold")) +
  coord_flip() # Deita o gráfico para ler melhor os nomes
```

Hierarquia de Intensidade Fisiológica



2.5.1 Pos-hoc

```

# -----
# 1. POST-HOC ROBUSTO
# ----

# Calcular as médias marginais estimadas pelo modelo
emm_results <- emmeans(model_emotion_cat, ~ emo_id, pbkrtest.limit = 7549)

# A. Tabela de Contrastos (Pares Específicos)
# O ajuste "tukey" é o método robusto para múltiplas comparações
contrast_table <- pairs(emm_results, adjust = "sidak")

print("--- COMPARAÇÕES PAR A PAR (ROBUSTO - TUKEY) ---")

## [1] "--- COMPARAÇÕES PAR A PAR (ROBUSTO - TUKEY) ---"

# Vamos filtrar apenas os significativos para não poluir a tela,
# mas você pode ver todos removendo o filtro.
print(contrast_table)

```

contrast	estimate	SE	df	t.ratio	p.value
Anger - Happy	-0.05	0.05	6	-1.00	0.32
Anger - Fear	-0.05	0.05	6	-1.00	0.32
Anger - Surprise	-0.05	0.05	6	-1.00	0.32
Anger - Disgust	-0.05	0.05	6	-1.00	0.32
Anger - Sad	-0.05	0.05	6	-1.00	0.32
Anger - Neutral	-0.05	0.05	6	-1.00	0.32
Happy - Fear	-0.05	0.05	6	-1.00	0.32
Happy - Surprise	-0.05	0.05	6	-1.00	0.32
Happy - Disgust	-0.05	0.05	6	-1.00	0.32
Happy - Sad	-0.05	0.05	6	-1.00	0.32
Happy - Neutral	-0.05	0.05	6	-1.00	0.32
Fear - Surprise	-0.05	0.05	6	-1.00	0.32
Fear - Disgust	-0.05	0.05	6	-1.00	0.32
Fear - Sad	-0.05	0.05	6	-1.00	0.32
Fear - Neutral	-0.05	0.05	6	-1.00	0.32
Surprise - Disgust	-0.05	0.05	6	-1.00	0.32
Surprise - Sad	-0.05	0.05	6	-1.00	0.32
Surprise - Neutral	-0.05	0.05	6	-1.00	0.32
Disgust - Sad	-0.05	0.05	6	-1.00	0.32
Disgust - Neutral	-0.05	0.05	6	-1.00	0.32
Sad - Neutral	-0.05	0.05	6	-1.00	0.32

```

## Anger - Disgust      0.2060 0.0335 7442   6.153 <.0001
## Anger - Fear         0.0563 0.0331 7441   1.700  0.8593
## Anger - Happy        0.0325 0.0331 7441   0.983  0.9997
## Anger - Neutral      0.4026 0.0336 7442  11.981 <.0001
## Anger - Sad          0.3200 0.0336 7442   9.537 <.0001
## Anger - Surprise     0.0903 0.0331 7441   2.727  0.1264
## Disgust - Fear       -0.1497 0.0335 7442  -4.472  0.0002
## Disgust - Happy      -0.1735 0.0334 7442  -5.193 <.0001
## Disgust - Neutral    0.1966 0.0339 7442   5.794 <.0001
## Disgust - Sad         0.1139 0.0339 7442   3.363  0.0161
## Disgust - Surprise   -0.1157 0.0334 7442  -3.460  0.0114
## Fear - Happy         -0.0238 0.0331 7441  -0.720  1.0000
## Fear - Neutral       0.3463 0.0336 7442  10.308 <.0001
## Fear - Sad           0.2636 0.0335 7442   7.860 <.0001
## Fear - Surprise      0.0340 0.0331 7442   1.026  0.9995
## Happy - Neutral      0.3701 0.0335 7442  11.033 <.0001
## Happy - Sad          0.2874 0.0335 7442   8.584 <.0001
## Happy - Surprise     0.0578 0.0331 7441   1.748  0.8284
## Neutral - Sad        -0.0826 0.0340 7442  -2.431  0.2730
## Neutral - Surprise   -0.3123 0.0336 7442  -9.302 <.0001
## Sad - Surprise       -0.2297 0.0335 7442  -6.852 <.0001
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 21 tests

```

2.5.2 Análise Hierárquica

```
cld_results <- cld(emm_results, alpha = 0.05, Letters = letters, adjust = "sidak")
print("--- GRUPOS ESTATÍSTICOS (LETRAS IGUAIS = SEM DIFERENÇA) ---")
```

```
## [1] "--- GRUPOS ESTATÍSTICOS (LETRAS IGUAIS = SEM DIFERENÇA) ---"
```

```
print(cld_results)
```

```

## emo_id      emmean      SE  df lower.CL upper.CL .group
## Neutral   -0.30304 0.0651 130  -0.4805  -0.1256   a
## Sad        -0.22040 0.0651 130  -0.3978  -0.0430   a
## Disgust   -0.10645 0.0650 129  -0.2838   0.0709   b
## Surprise   0.00928 0.0649 128  -0.1676   0.1861   c
## Fear       0.04324 0.0649 128  -0.1336   0.2201   c
## Happy      0.06705 0.0648 128  -0.1097   0.2438   c
## Anger      0.09956 0.0649 128  -0.0773   0.2764   c
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 7 estimates
## P value adjustment: sidak method for 21 tests
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##       then we cannot show them to be different.
##       But we also did not show them to be the same.

```

```

# =====
# 5. O MODELO LINEAR MISTO
# =====
# Agora sim, testamos a interação com N=102 participantes (~7500 observações)

model_interaction_normative <-
  lmerTest::lmer(physio_intensity ~ norm_valence * norm_arousal +
    (1 | participant_id), data = data_valid_physio)

print("--- RESULTADO DO MODELO DE INTERAÇÃO (DADOS NORMATIVOS) ---")

## [1] "--- RESULTADO DO MODELO DE INTERAÇÃO (DADOS NORMATIVOS) ---"

summary(model_interaction_normative)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: physio_intensity ~ norm_valence * norm_arousal + (1 | participant_id)
##   Data: data_valid_physio
##
## REML criterion at convergence: 18181.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1962 -0.5202 -0.1593  0.2524 10.5247
##
## Random effects:
##   Groups            Name        Variance Std.Dev.
##   participant_id (Intercept) 0.3722   0.6101
##   Residual          0.6161   0.7849
## Number of obs: 7549, groups: participant_id, 102
##
## Fixed effects:
##                               Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)             -4.589e-02 1.215e-01 1.413e+03 -0.378 0.70568
## norm_valence            -7.702e-02 2.799e-02 7.445e+03 -2.752 0.00594 **
## norm_arousal             3.500e-03 1.810e-02 7.445e+03  0.193 0.84665
## norm_valence:norm_arousal 1.208e-02 4.861e-03 7.445e+03  2.485 0.01297 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) nrm_vl nrm_rs
## norm_valenc -0.822
## norm_arousl -0.848  0.938
## nrm_vlnc:n_  0.804 -0.984 -0.952

r2(model_interaction_normative)

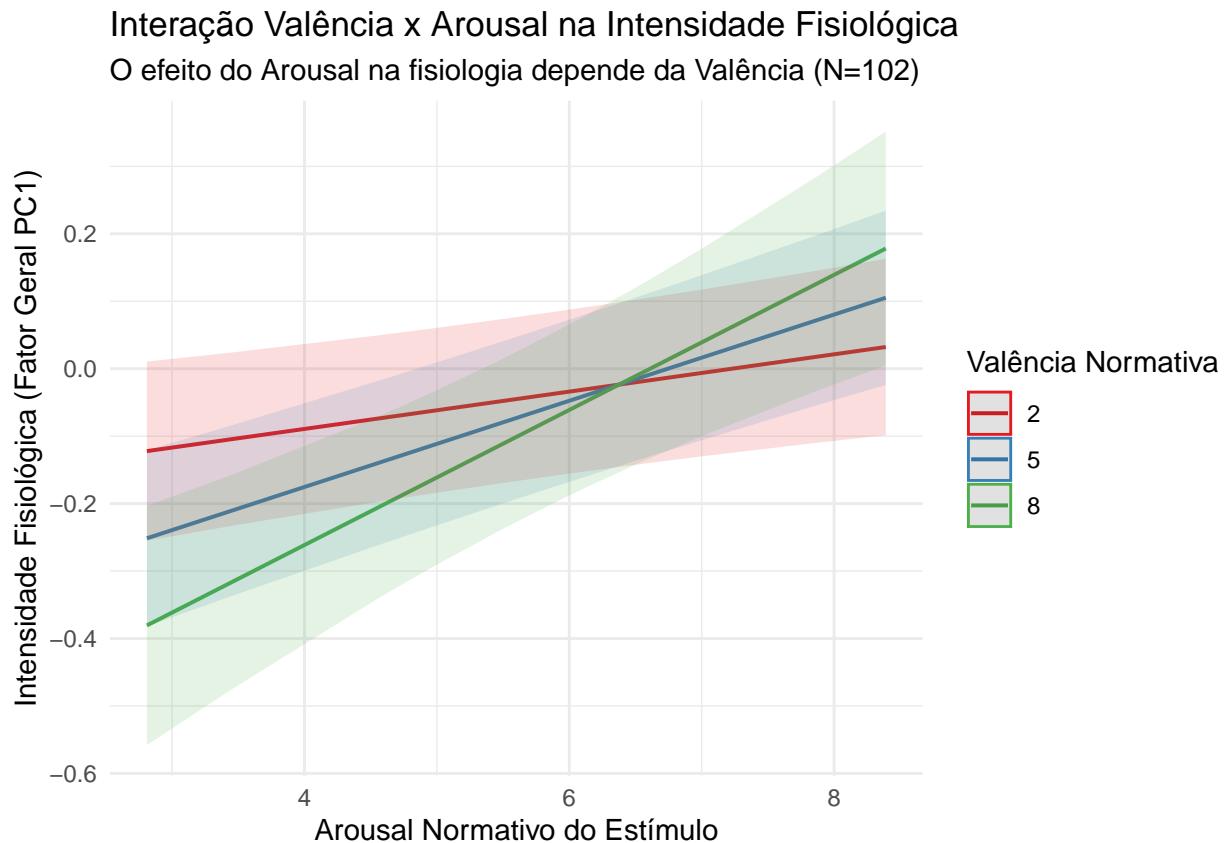
## # R2 for Mixed Models
##
## Conditional R2: 0.381
## Marginal R2: 0.007

```

2.6 Plot para verificar as interações

```
# Gerar previsões para o gráfico de interação
# Terms: eixo X = Arousal, Cores = Valence (definimos 3 níveis: Baixa/Negativa, Média, Alta/Positiva)
pred_interaction <- ggpredict(model_interaction_normative, terms = c("norm_arousal", "norm_valence [2, 5, 8]"))

# Plotar
plot(pred_interaction) +
  labs(
    title = "Interação Valência x Arousal na Intensidade Fisiológica",
    subtitle = "O efeito do Arousal na fisiologia depende da Valência (N=102)",
    y = "Intensidade Fisiológica (Fator Geral PC1)",
    x = "Arousal Normativo do Estímulo",
    colour = "Valência Normativa"
  ) +
  theme_minimal()
```



3 Existem perfis latentes?

Ainda que o teste tenha indicado 4 perfis, o quarto foi espúrio.

```

# 1. Selecionar variáveis menos redundantes
# Frequência (n_scr) e Magnitude Máxima (phasic_max) contam histórias complementares
df_lpa_simple <- data_valid_physio %>%
  dplyr::select(cda_n_scr, cda_phasic_max) %>%
  scale() %>%
  as.data.frame()

# 2. Estimar Perfis (Focando no Modelo 1)
set.seed(123)
lpa_result <- df_lpa_simple %>%
  estimate_profiles(1:4,
                    package = "mclust",
                    models = c(1)) # Apenas variância igual

## The 'variances'/'covariances' arguments were ignored in favor of the 'models' argument.

# 3. Comparar
print("--- Comparaçāo de Ajuste ---")

## [1] "--- Comparaçāo de Ajuste ---"

compare_solutions(lpa_result, statistics = c("AIC", "BIC", "Entropy", "BLRT_p"))

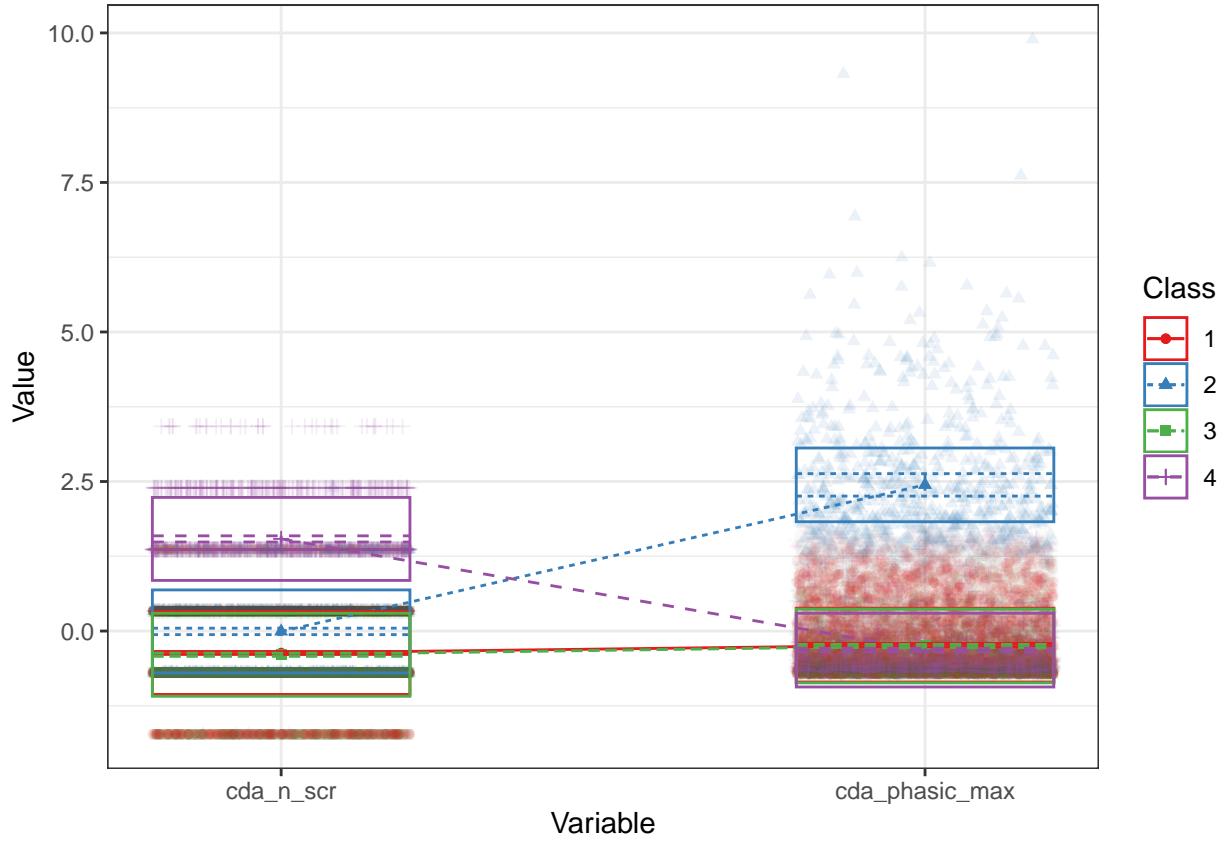
## Compare tidyLPA solutions:
##
##   Model Classes   AIC      BIC      Entropy    BLRT_p Warnings
##   1       1     42852.27 42879.98 1.0000000
##   1       2     42861.19 42909.70 0.2878684      Warning
##   1       3     38806.58 38875.87 0.8382346
##   1       4     38754.84 38844.92 0.6348751
##
## Best model according to AIC is Model 1 with 4 classes.
## Best model according to BIC is Model 1 with 4 classes.
## Best model according to Entropy is Model NA with NA classes.
## Best model according to BLRT_p is Model NA with NA classes.
##
## An analytic hierarchy process, based on the fit indices AIC, AWE, BIC, CLC, and KIC (Akogul & Erisoglu, 2012)

# 4. Escolher a Melhor Solução (Vamos assumir 3 ou 4 com base no BIC anterior)
# Se 4 classes foi o melhor antes, vamos validar com 4 aqui.
best_lpa <- df_lpa_simple %>%
  estimate_profiles(4, models = 1)

## The 'variances'/'covariances' arguments were ignored in favor of the 'models' argument.

# 5. Plotar Perfis
plot_profiles(best_lpa, add_line = TRUE)

```



Com 3 perfis

1. Baixa Reatividade: As linhas lá embaixo. O corpo quase não reage.
2. Alta Frequência (Frequent): A linha roxa no seu gráfico anterior. A pessoa tem muitos picos (n_scr alto), mas não necessariamente fortes. É um estado de “alerta constante”.
3. Alta Magnitude (Intense): A linha azul que sobe. A pessoa dá “tiros” de adrenalina (phasic_max alto).

```
# =====
# 4. RODAR O MELHOR MODELO (3 PERFIS - DECISÃO PELA ENTROPIA)
# =====
set.seed(123)
lpa_result <- df_lpa_simple %>%
  estimate_profiles(1:4,
    package = "mclust",
    models = c(1)) # Apenas variância igual

## The 'variances'/'covariances' arguments were ignored in favor of the 'models' argument.

## Warning:
## One or more analyses resulted in warnings! Examine these analyses carefully: model_1_class_2
```

```

# 3. Comparar
print("--- Comparação de Ajuste ---")

## [1] "--- Comparação de Ajuste ---"

compare_solutions(lpa_result, statistics = c("AIC", "BIC", "Entropy", "BLRT_p"))

## Warning:
## One or more analyses resulted in warnings! Examine these analyses carefully: model_1_class_2

## Warning: The solution with the maximum number of classes under consideration
## was considered to be the best solution according to one or more fit indices.
## Examine your results with care and consider estimating more classes.

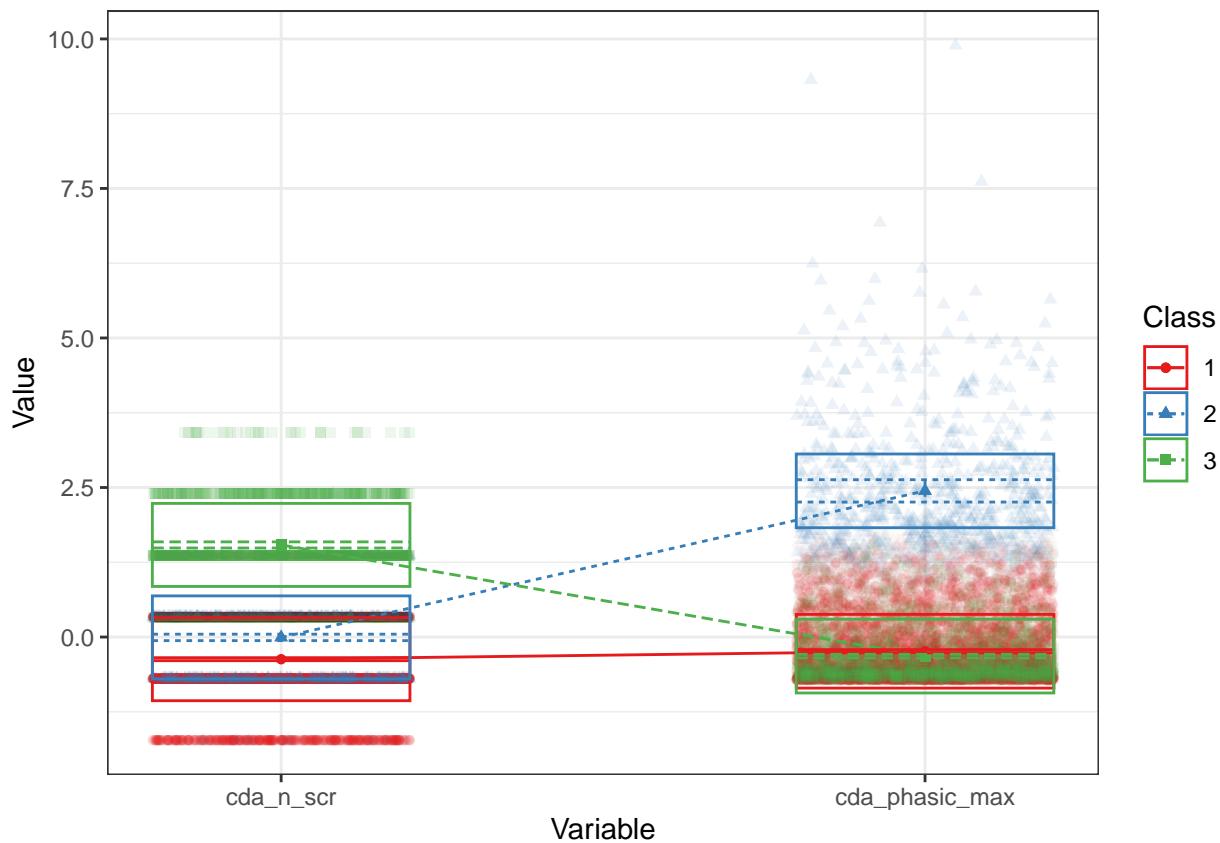
## Compare tidyLPA solutions:
##
##   Model Classes   AIC      BIC      Entropy    BLRT_p Warnings
## 1     1         42852.27 42879.98 1.0000000
## 1     2         42861.19 42909.70 0.2878684      Warning
## 1     3         38806.58 38875.87 0.8382346
## 1     4         38754.84 38844.92 0.6348751
##
## Best model according to AIC is Model 1 with 4 classes.
## Best model according to BIC is Model 1 with 4 classes.
## Best model according to Entropy is Model NA with NA classes.
## Best model according to BLRT_p is Model NA with NA classes.
##
## An analytic hierarchy process, based on the fit indices AIC, AWE, BIC, CLC, and KIC (Akogul & Erisoglu, 2017)

# 4. Escolher a Melhor Solução (Vamos assumir 3 ou 4 com base no BIC anterior)
# Se 4 classes foi o melhor antes, vamos validar com 4 aqui.
best_lpa <- df_lpa_simple %>%
  estimate_profiles(3, models = 1)

## The 'variances'/'covariances' arguments were ignored in favor of the 'models' argument.

# 5. Plotar Perfis
plot_profiles(best_lpa, add_line = TRUE)

```



3.0.1 “Esmiuçando”

```

# 1. Extrair classificação e probabilidades
df_classificado <- get_data(best_lpa)

# 2. Juntar com os dados originais (para pegar as emoções)
# Cuidado: O get_data retorna na mesma ordem do input.
data_analise_final <- bind_cols(data_valid_physio, Class = df_classificado$Class)

# Renomear os perfis para algo útil (Baseado na sua inspeção visual)
# Exemplo hipotético (AJUSTE conforme o gráfico que você viu):
# Class 1 = Baixo? Class 2 = Médio? Class 3 = Alto?
# Vou deixar genérico, depois você ajusta os labels no factor()
data_analise_final$Perfil <- factor(data_analise_final$Class,
                                       labels = c("Perfil 1", "Perfil 2", "Perfil 3"))

```

3.0.1.1 Organização dos dados

```

# Preparar dados para o plot (formato longo)
df_plot_summary <- data_analise_final %>%

```

```

dplyr::select(Perfil, cda_n_scr, cda_phasic_max) %>%
# Padronizar de novo só pra garantir que o gráfico fique na mesma escala visual
mutate(across(where(is.numeric), scale)) %>%
pivot_longer(cols = -Perfil, names_to = "Variavel", values_to = "Valor") %>%
group_by(Perfil, Variavel) %>%
summarise(
  Media = mean(Valor, na.rm = TRUE),
  SE = sd(Valor, na.rm = TRUE) / sqrt(n()), # Erro Padrão
  IC_Low = Media - 1.96 * SE,
  IC_High = Media + 1.96 * SE,
  .groups = 'drop'
) %>%
# Nomes bonitos para o gráfico
mutate(Variavel = recode(Variavel,
  "cda_n_scr" = "Frequênci(a) (nSCR)",
  "cda_phasic_max" = "Magnitude Máxima"))

# O GRÁFICO BONITÃO
ggplot(df_plot_summary, aes(x = Variavel, y = Media, group = Perfil, color = Perfil)) +
  geom_line(size = 1.2, position = position_dodge(0.1)) +
  geom_point(size = 3, position = position_dodge(0.1)) +
  geom_errorbar(aes(ymin = IC_Low, ymax = IC_High), width = 0.1, position = position_dodge(0.1)) +
  labs(
    title = "Caracterização dos Perfis Latentes de Reatividade",
    subtitle = "Médias padronizadas (Z-Score) com Intervalo de Confiança 95%",
    y = "Escore Z (Padronizado)",
    x = "Indicador Fisiológico"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "top") +
  scale_color_brewer(palette = "Dark2")

```

3.0.1.2 Gráfico mais simples

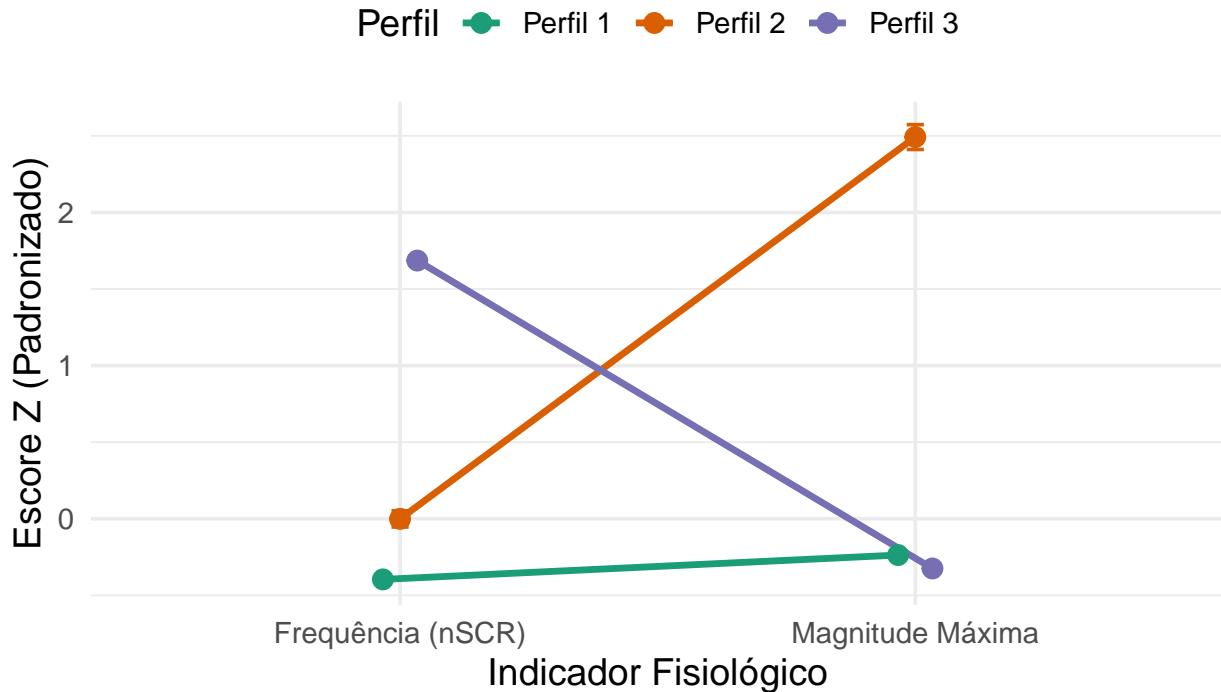
```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

Caracterização dos Perfis Latentes de Reatividade

Médias padronizadas (Z-Score) com Intervalo de Confiança 95%



```
# Tabela de Contingência
tabela_cruzada <- table(data_analise_final$emo_id, data_analise_final$Perfil)

# Teste Qui-Quadrado
teste_chi <- chisq.test(tabela_cruzada)

print("--- Associação entre Emoção e Perfil Fisiológico ---")
```

3.0.1.3 Os perfis diferem entre si?

```
## [1] "--- Associação entre Emoção e Perfil Fisiológico ---"
```

```
print(teste_chi)
```

```
##
## Pearson's Chi-squared test
##
## data: tabela_cruzada
## X-squared = 76.792, df = 12, p-value = 1.682e-11
```

```

# Pós-hoc: Resíduos Ajustados Padronizados
# Valores > 1.96 indicam que aquela emoção aparece MAIS do que o esperado naquele perfil.
# Valores < -1.96 indicam que aparece MENOS.
print("--- Resíduos Ajustados (Onde está a diferença?) ---")

## [1] "--- Resíduos Ajustados (Onde está a diferença?) ---"

round(teste_chi$stdres, 2)

##          Perfil 1 Perfil 2 Perfil 3
## Anger      -2.15     3.45   -0.13
## Disgust     -0.54    -0.63    1.12
## Fear        0.53     2.09   -2.22
## Happy       0.81     2.24   -2.66
## Neutral     -0.85    -4.40    4.36
## Sad         0.71    -3.87    2.14
## Surprise    1.47     0.93   -2.43

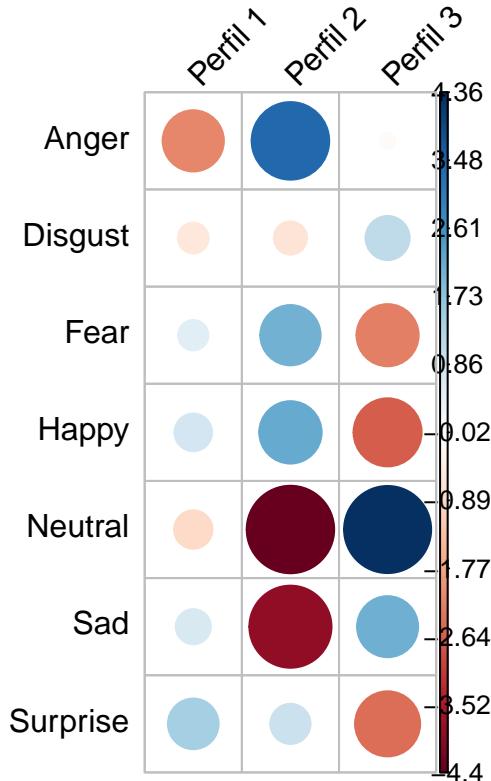
# Visualização da Associação (Heatmap dos Resíduos)
# Isso mostra visualmente o que o Qui-Quadrado achou
library(corrplot)

## corrplot 0.95 loaded

corrplot(teste_chi$stdres, is.cor = FALSE,
         title = "Associação Emoção x Perfil (Resíduos)",
         mar = c(0,0,2,0),
         tl.col = "black", tl.srt = 45)

```

Associação Emoção x Perfil (Resíduos)



```
sessionInfo()
```

```
## R version 4.5.0 (2025-04-11 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=Portuguese_Brazil.utf8  LC_CTYPE=Portuguese_Brazil.utf8
## [3] LC_MONETARY=Portuguese_Brazil.utf8 LC_NUMERIC=C
## [5] LC_TIME=Portuguese_Brazil.utf8
##
## time zone: America/Sao_Paulo
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices  utils       datasets   methods    base
##
## other attached packages:
## [1] corrplot_0.95      multcomp_1.4-29      TH.data_1.1-5      MASS_7.3-65
## [5] survival_3.8-3     mvtnorm_1.3-3       tidyLPA_1.1.0     lubridate_1.9.4
## [9] forcats_1.0.1      purrrr_1.0.4        readr_2.1.6       tidyr_1.3.1
## [13] tibble_3.2.1       tidyverse_2.0.0     emmeans_2.0.0     performance_0.15.2
## [17] ggplot2_4.0.1      ggeffects_2.3.1    stringr_1.6.0     psych_2.5.3
```

```

## [21] janitor_2.2.1      lmerTest_3.1-3      lme4_1.1-37       Matrix_1.7-3
## [25] readxl_1.4.5       dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##   [1] Rdpack_2.6.4        mnormt_2.1.1      gridExtra_2.3
##   [4] sandwich_3.1-1      rlang_1.1.6       magrittr_2.0.3
##   [7] snakecase_0.11.1    compiler_4.5.0    vctrs_0.6.5
##  [10] gsl_2.1-9          crayon_1.5.3     shape_1.4.6.1
##  [13] pkgconfig_2.0.3     fastmap_1.2.0    backports_1.5.0
##  [16] energy_1.7-12      labeling_0.4.3   pander_0.6.6
##  [19] rmarkdown_2.30      tzdb_0.5.0       haven_2.5.5
##  [22] nloptr_2.2.1      jomo_2.7-6      xfun_0.52
##  [25] glmnet_4.1-8       jsonlite_2.0.0   pan_1.9
##  [28] broom_1.0.10      parallel_4.5.0  R6_2.6.1
##  [31] stringi_1.8.7      RColorBrewer_1.1-3 rpart_4.1.24
##  [34] car_3.1-3          boot_1.3-32     cellranger_1.1.0
##  [37] numDeriv_2016.8-1.1 estimability_1.5.1 iterators_1.0.14
##  [40] Rcpp_1.1.0          knitr_1.50      zoo_1.8-14
##  [43] nnet_7.3-20         splines_4.5.0   timechange_0.3.0
##  [46] tidyselect_1.2.1    viridis_0.6.5   rstudioapi_0.17.1
##  [49] abind_1.4-8         yaml_2.3.10    codetools_0.2-20
##  [52] lattice_0.22-7     plyr_1.8.9     withr_3.0.2
##  [55] S7_0.2.1           coda_0.19-4.1  evaluate_1.0.5
##  [58] moments_0.14.1     fastDummies_1.7.5 mclust_6.1.2
##  [61] texreg_1.39.4       pillar_1.11.1   carData_3.0-5
##  [64] mice_3.17.0         nortest_1.0-4   checkmate_2.3.2
##  [67] foreach_1.5.2       reformulas_0.4.2 insight_1.4.2
##  [70] plotly_4.11.0      generics_0.1.4  hms_1.1.4
##  [73] scales_1.4.0        minqa_1.2.8    xtable_1.8-4
##  [76] glue_1.8.0          lazyeval_0.2.2  tools_4.5.0
##  [79] MVN_6.2             data.table_1.17.0 gsubfn_0.7
##  [82] grid_4.5.0          MplusAutomation_1.2 rbibutils_2.3
##  [85] datawizard_1.3.0    nlme_3.1-168   proto_1.0.0
##  [88] Formula_1.2-5       cli_3.6.5      viridisLite_0.4.2
##  [91] gtable_0.3.6         digest_0.6.38   pbkrtest_0.5.5
##  [94] htmlwidgets_1.6.4    farver_2.1.2   htmltools_0.5.8.1
##  [97] lifecycle_1.0.4     multcompView_0.1-10 httr_1.4.7
## [100] mitml_0.4-5

```