

# song\_sent\_scores(ICC'25)

Frederico Pedrosa

2025-06-14

## Contents

<b>1</b>	<b><i>Load packages</i></b>	<b>1</b>
<b>2</b>	<b><i>Load and prepare human judge data</i></b>	<b>2</b>
<b>3</b>	<b><i>Demographics</i></b>	<b>3</b>
<b>4</b>	<b><i>Preparing items for reliability testing</i></b>	<b>4</b>
<b>5</b>	<b><i>Intraclass correlation coefficient (ICC) - Reliability</i></b>	<b>5</b>
<b>6</b>	<b><i>Estimating average human scores and preparing AI scores</i></b>	<b>6</b>
<b>7</b>	<b><i>Rescaling human averages to the AI range (-1 to +1)</i></b>	<b>7</b>
<b>8</b>	<b><i>Creating comparison dataframe for correlation analysis and plots</i></b>	<b>8</b>
<b>9</b>	<b><i>Function to analyze and format correlation results</i></b>	<b>9</b>
<b>10</b>	<b><i>Estimating and presenting human-AI correlations</i></b>	<b>10</b>
<b>11</b>	<b><i>Plots</i></b>	<b>12</b>
11.1	<i>Preparing data taken from Colab mentioned ahead . . . . .</i>	12
11.2	<i>Temporal development of Arousal and Valence in Negro Drama - Racionais MC's . . . . .</i>	13
11.3	<i>Significance of correlation between human and AI scores . . . . .</i>	16
<b>12</b>	<b><i>Dynamic Psychometric Network</i></b>	<b>17</b>
<b>1</b>	<b><i>Load packages</i></b>	

```
# install.packages(c("readxl", "haven", "psych", "irr", "dplyr", "tidyr", "stringr", "ggplot2"))
library(readxl)
library(haven)      # Para as_factor
library(psych)      # Para describe() e ICC()
library(irr)        # Alternativa para ICC, não usada explicitamente abaixo mas pode ser útil
```

```
## Carregando pacotes exigidos: lpSolve
```

```
library(dplyr)      # Para manipulação de dados (mutate, select, filter, etc.)
```

```
##
```

```
## Anexando pacote: 'dplyr'
```

```
## Os seguintes objetos são mascarados por 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## Os seguintes objetos são mascarados por 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(tidyr)      # Para pivot_longer
library(stringr)     # Para manipulação de strings (str_extract, sub)
library(ggplot2)     # Para gráficos
```

```
##
```

```
## Anexando pacote: 'ggplot2'
```

```
## Os seguintes objetos são mascarados por 'package:psych':
```

```
##
```

```
##      %+%, alpha
```

## 2 Load and prepare human judge data

```
cat("--- 1. Carregando e Preparando Dados Humanos ---\n")
```

```
## --- 1. Carregando e Preparando Dados Humanos ---
```

```
tryCatch({
  song_sent_human <- read_excel("~/ICCC2025/song_sent_human.xlsx")
  data_raw <- as_factor(song_sent_human)
}, error = function(e) {
  stop("Erro ao carregar o arquivo Excel. Verifique o caminho e o arquivo: ", e$message)
})
df_juizes <- data_raw[1:13, ]
```

### 3 *Demographics*

```
cat("\n--- 2. Análise Descritiva da Amostra ---\n")
```

```
##  
## --- 2. Análise Descritiva da Amostra ---
```

```
cat("Idade:\n")
```

```
## Idade:
```

```
print(describe(df_juizes$Age)) # Usar df_juizes aqui e abaixo
```

```
##      vars  n  mean      sd median trimmed   mad min max range skew kurtosis   se  
## X1      1 13 45.92 14.13      38   45.27 11.86  30  69    39 0.31    -1.71 3.92
```

```
cat("\nSexo:\n")
```

```
##  
## Sexo:
```

```
print(prop.table(table(df_juizes$Sex)))
```

```
##  
## Feminino Masculino  
## 0.6923077 0.3076923
```

```
cat("\nEstado:\n")
```

```
##  
## Estado:
```

```
print(prop.table(table(df_juizes$State)) * 100)
```

```
##  
## Distrito Federal (DF)      Goiás (GO)      Minas Gerais (MG)  
##           7.692308           7.692308           46.153846  
##           Paraná (PR)      Rio de Janeiro (RJ)      São Paulo (SP)  
##           23.076923           7.692308           7.692308
```

```
cat("\nIdentidade de Gênero:\n")
```

```
##  
## Identidade de Gênero:
```

```
print(prop.table(table(df_juizes$Gender)))
```

```
##  
## Cisgênero Não-binário  
## 0.92307692 0.07692308
```

```
cat("\nEscolaridade:\n")
```

```
##  
## Escolaridade:
```

```
print(prop.table(table(df_juizes$Scholarship)))
```

```
##  
##      Doutorado Especialização      Graduação      Mestrado Pós-graduação  
##      0.30769231      0.07692308      0.23076923      0.30769231      0.07692308
```

```
cat("\nEstudo de Música:\n")
```

```
##  
## Estudo de Música:
```

```
print(prop.table(table(df_juizes$Music_study)))
```

```
##  
##      5 anos ou mais De 1 ano a 5 anos De 1 mês a 1 ano Mais de 15 anos  
##      0.46153846      0.07692308      0.15384615      0.07692308  
## Não estudei música  
##      0.23076923
```

```
cat("\nPrática Musical:\n")
```

```
##  
## Prática Musical:
```

```
print(prop.table(table(df_juizes$Music_practice)))
```

```
##  
##      5 anos ou mais De 1 mês a 1 ano  
##      0.6923077      0.1538462  
## Nunca toquei/cantei música  
##      0.1538462
```

#### 4 *Preparing items for reliability testing*

```
cat("\n--- 3. Preparando Itens Avaliados pelos Humanos ---\n")
```

```
##
## --- 3. Preparando Itens Avaliados pelos Humanos ---
```

```
items_avalizados <- df_juizes[, 8:39]
tryCatch({
  items_numeric <- data.frame(lapply(items_avalizados, function(x) as.numeric(as.character(x))))
  if(any(sapply(items_numeric, function(col) sum(is.na(col)) > 0)) {
    warning("NAs introduzidos durante a conversão para numérico. Verifique os dados originais dos itens")
  }

  if (nrow(items_numeric) != 13 || ncol(items_numeric) != 32) {
    stop(paste0("Erro: 'items_numeric' não tem as dimensões esperadas (13x32). Dimensões atuais: ",
               nrow(items_numeric), "x", ncol(items_numeric)))
  }
  cat("Dimensões de 'items_numeric' (Juizes x Itens):", dim(items_numeric), "\n")
}, error = function(e) {
  stop("Erro ao processar 'items_avalizados' para 'items_numeric': ", e$message)
})
```

```
## Dimensões de 'items_numeric' (Juizes x Itens): 13 32
```

## 5 Intraclass correlation coefficient (ICC) - Reliability

```
cat("\n--- 4. Calculando ICC para Confiabilidade Interavaliadores ---\n")
```

```
##
## --- 4. Calculando ICC para Confiabilidade Interavaliadores ---
```

```
# Transpor para ter Itens como Linhas e Juizes como Colunas para psych::ICC
items_numeric_transposed <- as.data.frame(t(items_numeric))
# Renomear colunas para Juiz_1, Juiz_2, etc. (opcional, mas bom para clareza)
colnames(items_numeric_transposed) <- paste0("Juiz_", 1:13)

icc_geral_juizes <- psych::ICC(items_numeric_transposed)
print(icc_geral_juizes)
```

```
## Call: psych::ICC(x = items_numeric_transposed)
```

```
##
```

```
## Intraclass correlation coefficients
```

	type	ICC	F	df1	df2	p	lower bound	upper bound
## Single_raters_absolute	ICC1	0.26	5.6	31	384	2.3e-17	0.16	0.41
## Single_random_raters	ICC2	0.27	6.9	31	372	3.2e-22	0.17	0.42
## Single_fixed_raters	ICC3	0.31	6.9	31	372	3.2e-22	0.20	0.47
## Average_raters_absolute	ICC1k	0.82	5.6	31	384	2.3e-17	0.72	0.90
## Average_random_raters	ICC2k	0.83	6.9	31	372	3.2e-22	0.73	0.90
## Average_fixed_raters	ICC3k	0.86	6.9	31	372	3.2e-22	0.77	0.92

```
##
## Number of subjects = 32      Number of Judges = 13
## See the help file for a discussion of the other 4 McGraw and Wong estimates,
```

```
cat("ICC(2,k) - Confiabilidade da MÉDIA dos 13 juízes (acordo absoluto):\n")
```

```
## ICC(2,k) - Confiabilidade da MÉDIA dos 13 juízes (acordo absoluto):
```

```
print(icc_geral_juizes$results["Average_random_raters", "ICC"]) # ICC(2,k)
```

```
## [1] 0.8279541
```

```
cat("ICC(3,k) - Confiabilidade da MÉDIA dos 13 juízes (consistência):\n")
```

```
## ICC(3,k) - Confiabilidade da MÉDIA dos 13 juízes (consistência):
```

```
print(icc_geral_juizes$results["Average_fixed_raters", "ICC"]) # ICC(3,k)
```

```
## [1] 0.8559224
```

## 6 *Estimating average human scores and preparing AI scores*

Zero-shot classifications were estimated with the Python function (<https://colab.research.google.com/drive/1LxyqgjIgPoC1hBSSS56J634CZ8YqdO0B?usp=sharing>)

```
cat("\n--- 5. Calculando Médias Humanas e Definindo Scores da IA ---\n")
```

```
##
## --- 5. Calculando Médias Humanas e Definindo Scores da IA ---
```

```
# Médias Humanas (média de cada coluna/item em 'items_numeric')
medias_humanos_brutas <- colMeans(items_numeric, na.rm = TRUE)
cat("Médias brutas das avaliações humanas por item (escala 1-7):\n")
```

```
## Médias brutas das avaliações humanas por item (escala 1-7):
```

```
print(round(medias_humanos_brutas, 2))
```

```
##      X1.baby_1_v_1      X2.baby_1_a_1      X3.baby_a_v_1      X4.baby_a_a_1
##      5.15           3.85           5.77           4.23
##      X5.baby_1_v_2      X6.baby_1_a_2      X7.baby_a_v_2      X8.baby_a_a_2
##      5.38           4.31           5.77           5.15
##      X9.drama_1_v_1     X10.drama_1_a_1     X11.drama_a_v_1     X12.drama_a_a_1
##      3.69           4.31           3.54           4.31
##      X13.drama_1_v_2    X14.drama_1_a_2    X15.drama_a_v_2    X16.drama_a_a_2
##      2.92           4.54           3.62           4.54
##      X17.moinho_1_v_1   X18.moinho_1_a_1   X19.moinho_a_v_1   X20.moinho_a_a_1
```

```
##           4.77           3.54           5.08           3.54
## X21.moinho_l_v_2 X22.moinho_l_a_2 X23.moinho_a_v_2 X24.moinho_a_a_2
##           4.31           4.85           5.46           4.62
## X25.territory_l_v_1 X26.territory_l_a_1 X27.territory_a_v_1 X28.territory_a_a_1
##           3.08           5.62           2.69           6.23
## X29.territory_l_v_2 X30.territory_l_a_2 X31.territory_a_v_2 X32.territory_a_a_2
##           2.31           5.69           2.00           5.46
```

```
# Scores da IA (conforme calculado anteriormente)
scores_ia <- c(
  -0.0826, -0.0264, -0.6894, -0.2552, -0.0826, -0.0264, -0.7532, -0.2408,
  -0.7846, -0.2060, -0.7110,  0.3928, -0.6706, -0.1528, -0.6174,  0.5080,
  0.2240, -0.5026,  0.2514, -0.0266, -0.9932, -0.8470, -0.1982,  0.3804,
  -0.3870, -0.0222, -0.9586,  0.4332, -0.9766, -0.7262, -0.9762,  0.5094
)
# Atribuir nomes aos scores da IA para garantir alinhamento
# Os nomes devem corresponder exatamente aos de 'medias_humanos_brutas'
# que são os nomes das colunas de 'items_numeric' (ex: "X1.baby_l_v_1")
if (length(names(items_numeric)) == length(scores_ia)) {
  names(scores_ia) <- names(items_numeric)
} else {
  stop("Número de scores da IA não corresponde ao número de itens.")
}
```

## 7 Rescaling human averages to the AI range (-1 to +1)

```
cat("\n--- 6. Reescalando Médias Humanas ---\n")
```

```
##
## --- 6. Reescalando Médias Humanas ---
```

```
reescalar_humano_para_ia <- function(score_humano, min_humano = 1, max_humano = 7, neutro_humano = 4) {
  if (is.na(score_humano)) return(NA) # Lidar com NAs
  if (score_humano == neutro_humano) return(0)
  if (score_humano > neutro_humano) return((score_humano - neutro_humano) / (max_humano - neutro_humano))
  return((score_humano - neutro_humano) / (neutro_humano - min_humano))
}
medias_humanos_reescaladas <- sapply(medias_humanos_brutas, reescalar_humano_para_ia)
cat("Médias humanas reescaladas para a faixa -1 a +1:\n")
```

```
## Médias humanas reescaladas para a faixa -1 a +1:
```

```
print(round(medias_humanos_reescaladas, 2))
```

```
## X1.baby_l_v_1 X2.baby_l_a_1 X3.baby_a_v_1 X4.baby_a_a_1
## 0.38 -0.05 0.59 0.08
## X5.baby_l_v_2 X6.baby_l_a_2 X7.baby_a_v_2 X8.baby_a_a_2
## 0.46 0.10 0.59 0.38
## X9.drama_l_v_1 X10.drama_l_a_1 X11.drama_a_v_1 X12.drama_a_a_1
```

```
##          -0.10          0.10          -0.15          0.10
##      X13.drama_l_v_2      X14.drama_l_a_2      X15.drama_a_v_2      X16.drama_a_a_2
##          -0.36          0.18          -0.13          0.18
##      X17.moinho_l_v_1      X18.moinho_l_a_1      X19.moinho_a_v_1      X20.moinho_a_a_1
##          0.26          -0.15          0.36          -0.15
##      X21.moinho_l_v_2      X22.moinho_l_a_2      X23.moinho_a_v_2      X24.moinho_a_a_2
##          0.10          0.28          0.49          0.21
## X25.territory_l_v_1 X26.territory_l_a_1 X27.territory_a_v_1 X28.territory_a_a_1
##          -0.31          0.54          -0.44          0.74
## X29.territory_l_v_2 X30.territory_l_a_2 X31.territory_a_v_2 X32.territory_a_a_2
##          -0.56          0.56          -0.67          0.49
```

## 8 *Creating comparison dataframe for correlation analysis and plots*

```
cat("\n--- 7. Criando Dataframe de Comparação ---\n")
```

```
##
## --- 7. Criando Dataframe de Comparação ---
```

```
nomes_originais_itens <- names(medias_humanos_reescaladas) # Ex: "X1.baby_l_v_1"
itens_base_limpos <- sub("^X[0-9]+\\.\\.", "", nomes_originais_itens) # Remove "X1.", etc. -> "baby_l_v_1"

df_comparacao <- data.frame(
  Item_Orig = nomes_originais_itens,
  Item_Base_Limpo = itens_base_limpos,
  Humano_Reescalado = medias_humanos_reescaladas,
  IA_Score = scores_ia,
  stringsAsFactors = FALSE # Evitar fatores automáticos
)

df_comparacao <- df_comparacao %>%
  mutate(
    Musica = sub("_.*$", "", Item_Base_Limpo),
    Modalidade = case_when(
      grepl("_l_", Item_Base_Limpo) ~ "Lyrics",
      grepl("_a_v_|_a_a_", Item_Base_Limpo) ~ "Audio",
      TRUE ~ NA_character_
    ),
    Dimensao = case_when(
      grepl("_v_", Item_Base_Limpo) ~ "Valence",
      grepl("_l_a_|_a_a_", Item_Base_Limpo) ~ "Arousal", # Mais específico para arousal
      TRUE ~ NA_character_
    ),
    Trecho = as.numeric(str_extract(Item_Base_Limpo, "[0-9]$")),
    Musica_Trecho_ID = paste(Musica, Trecho, sep = "_Trecho")
  )
cat("Head do df_comparacao finalizado:\n")
```

```
## Head do df_comparacao finalizado:
```



```
print(head(df_comparacao))
```

```
##           Item_Orig Item_Base_Limpo Humano_Reescalado IA_Score Musica
## X1.baby_l_v_1 X1.baby_l_v_1      baby_l_v_1      0.38461538 -0.0826  baby
## X2.baby_l_a_1 X2.baby_l_a_1      baby_l_a_1     -0.05128205 -0.0264  baby
## X3.baby_a_v_1 X3.baby_a_v_1      baby_a_v_1      0.58974359 -0.6894  baby
## X4.baby_a_a_1 X4.baby_a_a_1      baby_a_a_1      0.07692308 -0.2552  baby
## X5.baby_l_v_2 X5.baby_l_v_2      baby_l_v_2      0.46153846 -0.0826  baby
## X6.baby_l_a_2 X6.baby_l_a_2      baby_l_a_2      0.10256410 -0.0264  baby
##           Modalidade Dimensao Trecho Musica_Trecho_ID
## X1.baby_l_v_1      Lyrics  Valence      1      baby_Trecho1
## X2.baby_l_a_1      Lyrics  Arousal      1      baby_Trecho1
## X3.baby_a_v_1       Audio  Valence      1      baby_Trecho1
## X4.baby_a_a_1       Audio  Arousal      1      baby_Trecho1
## X5.baby_l_v_2      Lyrics  Valence      2      baby_Trecho2
## X6.baby_l_a_2      Lyrics  Arousal      2      baby_Trecho2
```

```
cat("Verificação - Nomes únicos de Músicas:\n"); print(unique(df_comparacao$Musica))
```

```
## Verificação - Nomes únicos de Músicas:
```

```
## [1] "baby"      "drama"      "moinho"     "territory"
```

```
cat("Verificação - Contagem por Música:\n"); print(table(df_comparacao$Musica))
```

```
## Verificação - Contagem por Música:
```

```
##
##      baby      drama      moinho territory
##         8         8         8         8
```

## 9 *Function to analyze and format correlation results*

```
cat("\n--- 8. Definindo Função de Análise de Correlação ---\n")
```

```
##
## --- 8. Definindo Função de Análise de Correlação ---
```

```
analisar_resultado_cor_reporte_p <- function(nome_analise, resultado_cor_test) {
  if (!inherits(resultado_cor_test, "htest")) {
    return(data.frame(
      Analise = nome_analise, r = NA, p_valor_num = NA, p_valor_str = "N/A",
      IC_inferior = NA, IC_superior = NA,
      Magnitude = "N/A (Error/Insufficient Data)" # Removida coluna Significancia
    ))
  }
  r_est <- resultado_cor_test$estimate
```

```

p_val <- resultado_cor_test$p.value
conf_int <- resultado_cor_test$conf.int

ic_inf <- if (is.null(conf_int) || any(is.na(conf_int))) NA else conf_int[1]
ic_sup <- if (is.null(conf_int) || any(is.na(conf_int))) NA else conf_int[2]

abs_r <- abs(r_est)
mag_r <- ifelse(is.na(abs_r), "N/A",
               ifelse(abs_r < 0.1, "Very Weak/None",
                     ifelse(abs_r < 0.3, "Weak",
                           ifelse(abs_r < 0.5, "Moderate",
                                 ifelse(abs_r < 0.7, "Strong",
                                       ifelse(abs_r < 0.9, "Very Strong", "Near Perfect"))))))))

return(data.frame(
  Analise = nome_analise, r = round(r_est, 2), p_valor_num = p_val,
  p_valor_str = format.pval(p_val, digits = 3, eps = 0.001), # Aumentar precisão do p-valor
  IC_inferior = round(ic_inf, 2), IC_superior = round(ic_sup, 2),
  Magnitude = mag_r
))
}

```

## 10 *Estimating and presenting human-AI correlations*

```

cat("\n--- 9. Calculando Correlações Humano-IA (sem categorização de significância explícita) ---\n")

##
## --- 9. Calculando Correlações Humano-IA (sem categorização de significância explícita) ---

# 9.1 Correlação Geral
cor_geral <- cor.test(df_comparacao$Humano_Reescalado, df_comparacao$IA_Score, method = "pearson")
cat("\n** Correlação Geral (Todos os 32 itens) **\n")

##
## ** Correlação Geral (Todos os 32 itens) **

print(analisar_resultado_cor_reporte_p("Geral (Todos os Itens)", cor_geral))

##
##           Analise      r p_valor_num p_valor_str IC_inferior IC_superior
## cor Geral (Todos os Itens) 0.47 0.007282409      0.00728      0.14      0.7
##      Magnitude
## cor Moderate

# 9.2 Correlações por Música
musicas_unicas <- unique(df_comparacao$Musica)
resultados_cor_por_musica <- list() # Reinicializar se já existia
for (musica_atual in musicas_unicas) {
  df_filtrado <- filter(df_comparacao, Musica == musica_atual)

```

```

if (nrow(df_filtrado) >= 3) {
  resultados_cor_por_musica[[musica_atual]] <- cor.test(df_filtrado$Humano_Reescalado, df_filtrado$IA)
} else {
  resultados_cor_por_musica[[musica_atual]] <- NA
}
}
# Criar df_resultados_musica com a nova função
df_resultados_musica <- do.call(rbind, lapply(names(resultados_cor_por_musica), function(nome) {
  analisar_resultado_cor_reporte_p(paste("Música:", nome), resultados_cor_por_musica[[nome]])
}))
# Adicionar MusicaNome para o plot
df_resultados_musica$MusicaNome <- sub("Música: ", "", df_resultados_musica$Analise)

cat("\n** Correlações por Música **\n")

```

```

##
## ** Correlações por Música **

```

```

print(df_resultados_musica[order(-abs(df_resultados_musica$r)), ], row.names = FALSE)

```

```

##           Analise      r p_valor_num p_valor_str IC_inferior IC_superior
## Música: drama    0.79  0.01941214      0.0194      0.19      0.96
## Música: territory 0.78  0.02111878      0.0211      0.18      0.96
## Música: baby    -0.70  0.05366427      0.0537     -0.94      0.01
## Música: moinho   0.20  0.63797120      0.638      -0.59      0.79
## Magnitude MusicaNome
## Very Strong      drama
## Very Strong      territory
## Strong           baby
## Weak             moinho

```

```

# 9.3 Correlações por Dimensão Afetiva (Valence vs Arousal)
#dimensoes_unicas <- unique(df_comparacao$Dimensao)
#resultados_cor_por_dimensao <- list() # Reinicializar
#for (dimensao_atual in dimensoes_unicas) {
#  df_filtrado <- filter(df_comparacao, Dimensao == dimensao_atual)
#  if (nrow(df_filtrado) >= 3) {
#    resultados_cor_por_dimensao[[dimensao_atual]] <- #cor.test(df_filtrado$Humano_Reescalado, df_filtrado$IA)
#  } else {
#    resultados_cor_por_dimensao[[dimensao_atual]] <- NA
#  }
#}
#df_resultados_dimensao <- do.call(rbind, lapply(names(resultados_cor_por_dimensao), function(nome) {
#  analisar_resultado_cor_reporte_p(paste("Dimensão:", nome), resultados_cor_por_dimensao[[nome]])
#}))
#cat("\n** Correlações por Dimensão Afetiva **\n")
#print(df_resultados_dimensao[order(-abs(df_resultados_dimensao$r)), ], row.names = FALSE)

```

## 11 Plots

### 11.1 Preparing data taken from Colab mentioned ahead

```
drama_dynamics_df <- data.frame(  
  chunk_index = 0:27, # Ajuste o limite superior se houver mais linhas  
  start_sec = c(0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 90.0, 105.0, 120.0, 135.0,  
                150.0, 165.0, 180.0, 195.0, 210.0, 225.0, 240.0, 255.0, 270.0,  
                285.0, 300.0, 315.0, 330.0, 345.0, 360.0, 375.0, 390.0, 405.0),  
  end_sec = c(15.000000, 30.000000, 45.000000, 60.000000, 75.000000, 90.000000,  
              105.000000, 120.000000, 135.000000, 150.000000, 165.000000,  
              180.000000, 195.000000, 210.000000, 225.000000, 240.000000,  
              255.000000, 270.000000, 285.000000, 300.000000, 315.000000,  
              330.000000, 345.000000, 360.000000, 375.000000, 390.000000,  
              405.000000, 413.094604), # Último end_sec é diferente  
  a_v = c(-0.665753, -0.657330, -0.615731, -0.658784, -0.662083, -0.644427,  
          -0.657671, -0.665114, -0.634251, -0.733225, -0.621964, -0.370273,  
          -0.395543, -0.577507, -0.584650, -0.554828, -0.585612, -0.512236,  
          -0.569394, -0.544546, -0.696251, -0.342278, -0.427922, -0.547013,  
          -0.535473, -0.486693, -0.278924, 0.057519),  
  a_a = c(0.482093, 0.367038, 0.422882, 0.377440, 0.420690, 0.446748, 0.327155,  
          0.401789, 0.429925, 0.350500, 0.382882, 0.029830, 0.101933, 0.557221,  
          0.578816, 0.563559, 0.547528, 0.546595, 0.543269, 0.534901, 0.540644,  
          0.097132, 0.365725, 0.255547, 0.378836, 0.286943, 0.258139, -0.289718),  
  t_v = c(-0.956614, -0.964118, -0.971945, -0.829183, -0.435768, 0.212308,  
          0.996886, 0.212308, 0.240586, -0.948319, 0.392665, -0.515499,  
          -0.740753, -0.960305, -0.664196, 0.131511, -0.722270, -0.466326,  
          -0.237309, 0.333603, -0.692122, 0.023676, 0.576573, 0.653284,  
          -0.850350, 0.959888, 0.117958, 0.117958),  
  t_a = c(-0.683256, -0.783604, 0.020486, -0.377621, -0.244989, 0.209801,  
          -0.213847, 0.209801, -0.788298, -0.497455, 0.288773, -0.605028,  
          -0.318877, -0.867789, 0.649291, -0.467914, -0.533064, -0.515904,  
          -0.343820, -0.535258, 0.518485, -0.069823, -0.094264, -0.074851,  
          -0.473008, 0.967696, 0.117318, 0.117318)  
)  
  
# --- Transformação para VALÊNCIA ---  
valence_long_drama_df <- drama_dynamics_df %>%  
  select(start_sec, end_sec, a_v, t_v) %>%  
  pivot_longer(  
    cols = c("a_v", "t_v"),  
    names_to = "modality_code",  
    values_to = "valence_score"  
  ) %>%  
  mutate(Modality = case_when(  
    modality_code == "a_v" ~ "Audio Valence",  
    modality_code == "t_v" ~ "Text Valence",  
    TRUE ~ modality_code  
  ))  
  
# --- Transformação para AROUSAL ---  
arousal_long_drama_df <- drama_dynamics_df %>%
```

```

select(start_sec, end_sec, a_a, t_a) %>%
pivot_longer(
  cols = c("a_a", "t_a"),
  names_to = "modality_code",
  values_to = "arousal_score"
) %>%
mutate(Modality = case_when(
  modality_code == "a_a" ~ "Audio Arousal",
  modality_code == "t_a" ~ "Text Arousal",
  TRUE ~ modality_code
))

# --- [OPCIONAL mas recomendado] Combinar em um único dataframe longo ---
full_long_drama_df <- bind_rows(
  valence_long_drama_df %>%
    rename(score = valence_score) %>%
    mutate(Dimension = "Valence"),
  arousal_long_drama_df %>%
    rename(score = arousal_score) %>%
    mutate(Dimension = "Arousal")
) %>%
mutate(Modality_Type = case_when( # Para cores/formas mais simples
  grepl("Audio", Modality) ~ "Audio",
  grepl("Text", Modality) ~ "Text",
  TRUE ~ "Unknown"
))

```

## 11.2 Temporal development of Arousal and Valence in Negro Drama - Racionais MC's

```

# --- 10. Gerar Gráficos ---
cat("\n--- 10. Gerando Gráficos para o Pictorial ---\n")

##
## --- 10. Gerando Gráficos para o Pictorial ---

# 10.1 Plot para Dinâmica do Arousal (Negro Drama)
# Cores do seu gráfico: Audio Arousal (vermelho/rosa), Text Arousal (verde-azulado claro)
plot_arousal_dynamics_drama_pictorial <- ggplot(arousal_long_drama_df,
  aes(x = start_sec, y = arousal_score, color = Modality, group = Modality)) +
  geom_line(linewidth = 1) + # Ajuste linewidth conforme a imagem original
  geom_point(size = 2.5) + # Ajuste size conforme a imagem original
  scale_color_manual(values = c("Audio Arousal" = "#D8828E", "Text Arousal" = "#8DD3C7")) +
  scale_x_continuous(breaks = seq(0, max(arousal_long_drama_df$start_sec, na.rm = TRUE) + 15, by = 30),
    name = "Time (seconds)") +
  scale_y_continuous(limits = c(-1.0, 1.0), breaks = seq(-1.0, 1.0, 0.5),
    name = "Arousal Score (-1 Low to +1 High)") +
  labs(
    # title = "Negro Drama: Arousal Dynamics Over Time", # Título pode ser removido se estiver na legend
    color = "Modality"
  ) +

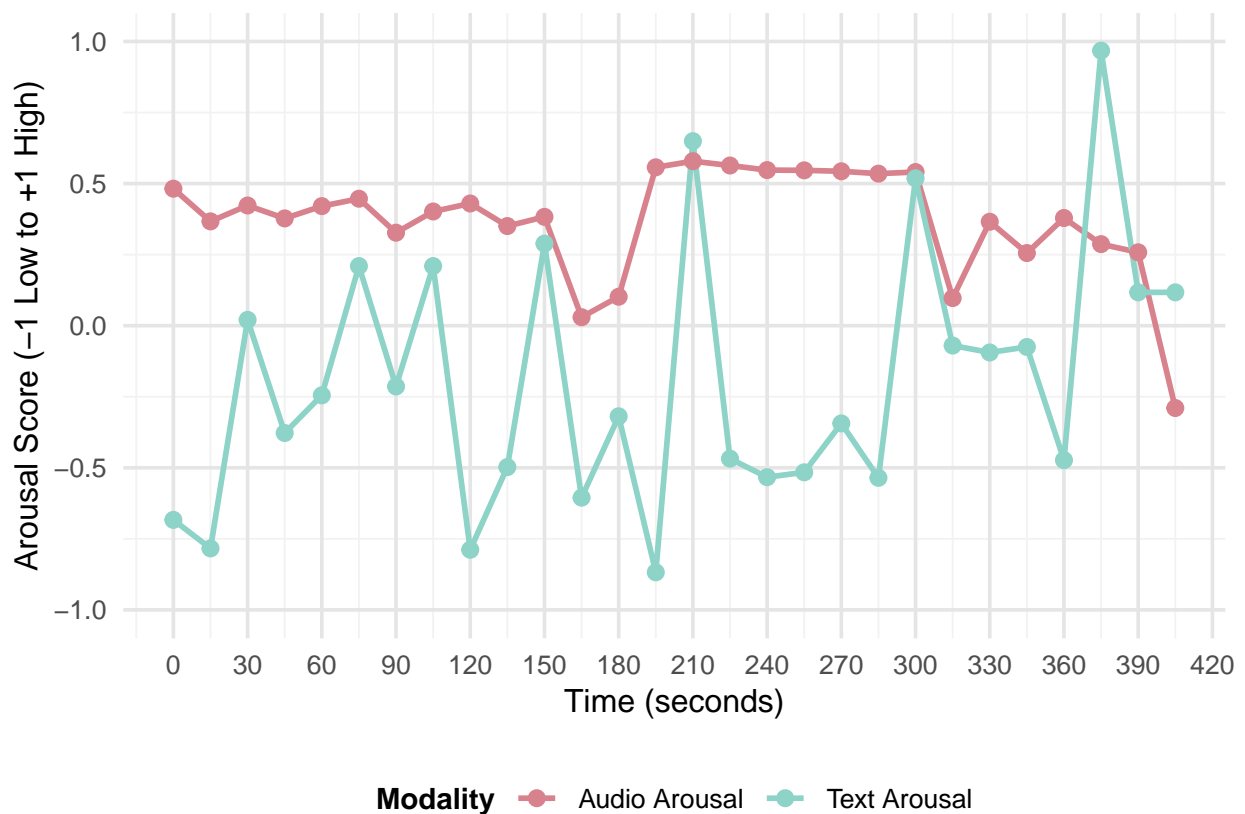
```

```

theme_minimal(base_size = 14) + # Ajuste base_size para corresponder ao tamanho da fonte na imagem
theme(
  legend.position = "bottom",
  # plot.title = element_text(hjust = 0.5, face="bold"),
  axis.title.x = element_text(size = 12), # Ajuste tamanho
  axis.title.y = element_text(size = 12), # Ajuste tamanho
  axis.text = element_text(size = 10),    # Ajuste tamanho
  legend.text = element_text(size = 10),  # Ajuste tamanho
  legend.title = element_text(size = 11, face = "bold"),
  panel.grid.major = element_line(colour = "grey90"), # Linhas de grade como na imagem
  panel.grid.minor = element_line(colour = "grey95") # Linhas de grade menores
)

print(plot_arousal_dynamics_drama_pictorial)

```



```

# ggsave("arousal_dynamics_drama_pictorial.png", plot_arousal_dynamics_drama_pictorial,
#         width = 8, height = 4, dpi = 300, bg = "white") # Ajuste width/height

```

```

# 10.2 Plot para Dinâmica da Valência (Nego Drama)
# Cores do seu gráfico: Audio Valence (verde escuro/azulado), Text Valence (lilás claro)
plot_valence_dynamics_drama_pictorial <- ggplot(valence_long_drama_df,
  aes(x = start_sec, y = valence_score, color = Modality, group = Modality)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2.5) +
  scale_color_manual(values = c("Audio Valence" = "#1F78B4", "Text Valence" = "#B2ABD2")) +

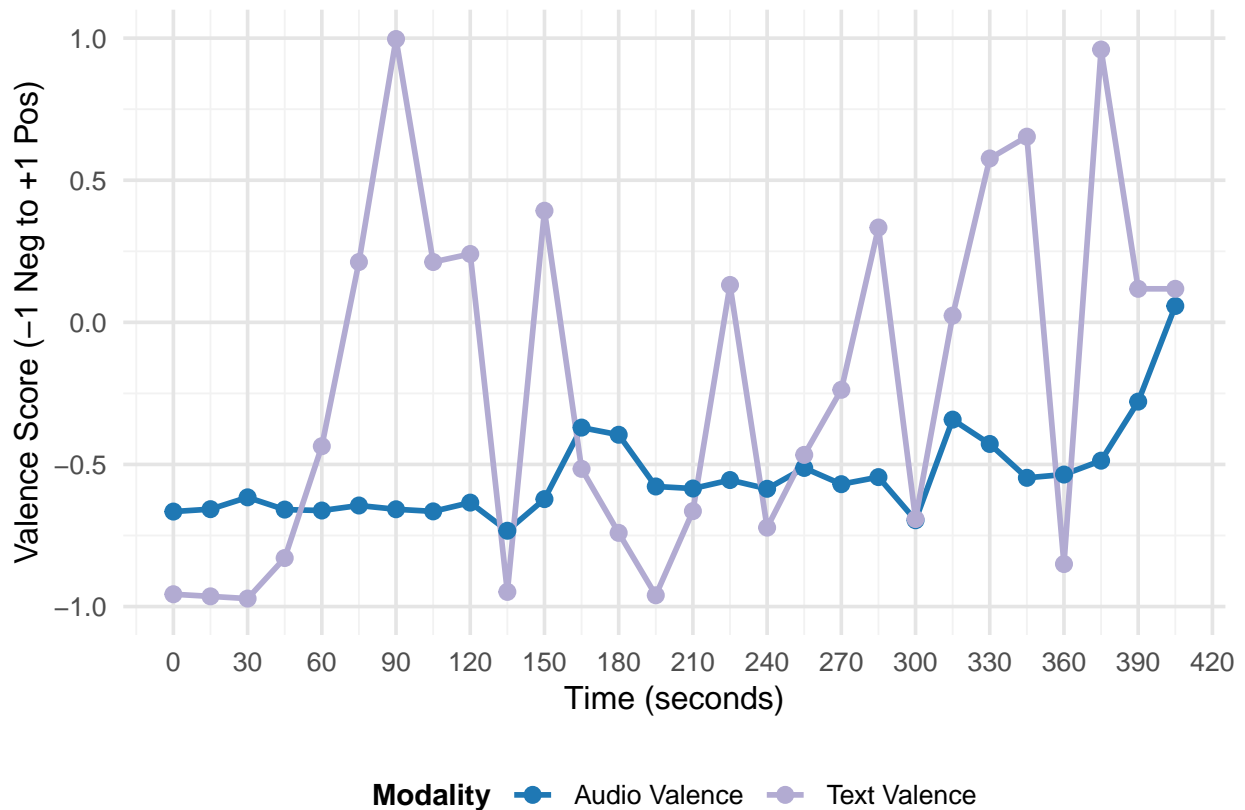
```

```

scale_x_continuous(breaks = seq(0, max(valence_long_drama_df$start_sec, na.rm = TRUE) + 15, by = 30),
  name = "Time (seconds)" +
scale_y_continuous(limits = c(-1.0, 1.0), breaks = seq(-1.0, 1.0, 0.5),
  name = "Valence Score (-1 Neg to +1 Pos)" +
labs(
  # title = "Nego Drama: Valence Dynamics Over Time",
  color = "Modality"
) +
theme_minimal(base_size = 14) +
theme(
  legend.position = "bottom",
  # plot.title = element_text(hjust = 0.5, face = "bold"),
  axis.title.x = element_text(size = 12),
  axis.title.y = element_text(size = 12),
  axis.text = element_text(size = 10),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 11, face = "bold"),
  panel.grid.major = element_line(colour = "grey90"),
  panel.grid.minor = element_line(colour = "grey95")
)

print(plot_valence_dynamics_drama_pictorial)

```



```

# ggsave("valence_dynamics_drama_pictorial.png", plot_valence_dynamics_drama_pictorial,
#         width = 8, height = 4, dpi = 300, bg = "white")

```

### 11.3 Significance of correlation between human and AI scores

```
df_resultados_musica <- df_resultados_musica %>%
  mutate(Cor_Significancia = ifelse(p_valor_num < 0.05, "Significant", "Not Significant"))

plot_cor_musica_pictorial <- ggplot(df_resultados_musica,
                                   aes(x = r, y = reorder(MusicaNome, r))) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey80", linewidth = 0.5) +

  geom_errorbarh(aes(xmin = IC_inferior, xmax = IC_superior, color = Cor_Significancia),
                 height = 0.0, # Remove as pequenas barras verticais nas pontas do IC
                 linewidth = 1) + # Ajuste a espessura da linha do IC

  # Pontos para a estimativa de r
  geom_point(aes(fill = Cor_Significancia), size = 3.5, shape = 21, color = "black", stroke = 0.5) +

  # Escalas e Nomes dos Eixos
  scale_x_continuous(limits = c(-1.05, 1.05), breaks = seq(-1.0, 1.0, 0.25),
                     name = "Correlation Coefficient (r) [95% CI]" +
  scale_y_discrete(name = "Song") +

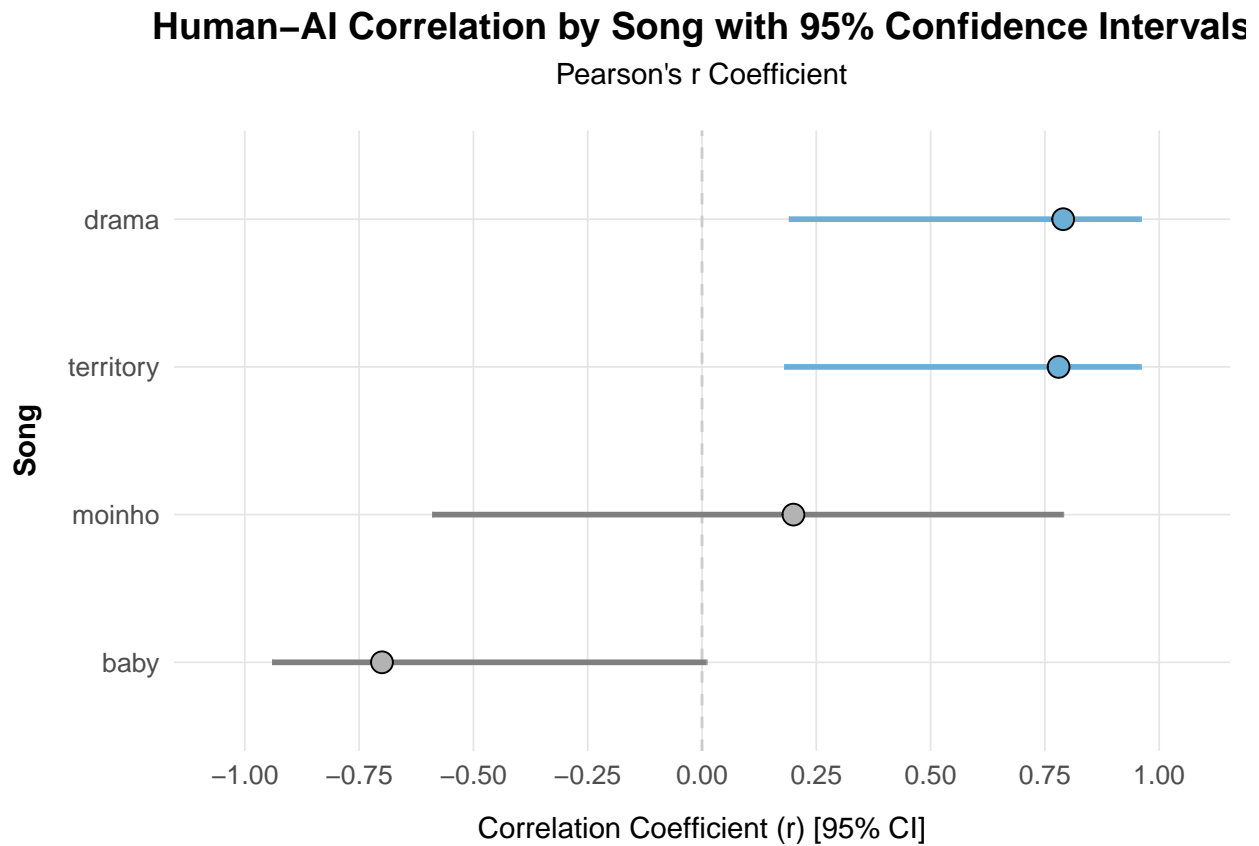
  # Cores Manuais para preenchimento dos pontos e linhas de IC
  # A cor da sua imagem para significativa é um azul claro, para não significativa é cinza.
  scale_fill_manual(name = "Statistical Significance", # Este nome não será usado se guide="none"
                    values = c("Significant" = "#6BAED6", # Azul claro (exemplo, ajuste)
                              "Not Significant" = "grey70"),
                    guide = "none") + # Esconde a legenda de preenchimento
  scale_color_manual(name = "Statistical Significance",
                     values = c("Significant" = "#6BAED6", # Mesma cor azul para a linha do IC
                               "Not Significant" = "grey50"), # Cinza um pouco mais escuro para a linha
                     guide = "none") + # Esconde a legenda de cor

  # Títulos
  labs(
    title = "Human-AI Correlation by Song with 95% Confidence Intervals",
    subtitle = "Pearson's r Coefficient"
  ) +

  # Tema
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    plot.subtitle = element_text(hjust = 0.5, size = 11, margin = margin(b = 15)), # Margem abaixo do s
    axis.title.x = element_text(size=11, margin = margin(t = 10)),
    axis.title.y = element_text(size=11, face="bold", margin = margin(r = 10)),
    axis.text = element_text(size=10),
    panel.grid.major.x = element_line(colour = "grey90", linewidth = 0.3),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(colour = "grey90", linewidth = 0.3, linetype = "solid"), # Linhas
    panel.grid.minor.y = element_blank(),
    legend.position = "none" # Remove a legenda completamente, já que a cor indica a significância
  )
```



```
print(plot_cor_musica_pictorial)
```



```
# ggsave("correlation_by_song_pictorial_final.png", plot_cor_musica_pictorial,
#         width = 7.5, height = 4.5, dpi = 300, bg = "white") # Ajuste width/height conforme necessário
```

## 12 *Dynamic Psychometric Network*

```
## This is psychonetrics 0.13! For questions, issues, and bug reports, please see github.com/SachaEpskar
```

```
##
```

```
## Anexando pacote: 'psychonetrics'
```

```
## O seguinte objeto é mascarado por 'package:psych':
```

```
##
```

```
##     bifactor
```

```
## O seguinte objeto é mascarado por 'package:graphics':
```

```
##
```

```
##     identify
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
```

```
## i Please use a list of either functions or lambdas:
```

```

##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## i The deprecated feature was likely used in the psychonetrics package.
## Please report the issue at
## <https://github.com/SachaEpskamp/psychonetrics/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning in runmodel(.): One or more parameters were estimated to be near its
## bounds. This may be indicative of, for example, a Heywood case, but also of an
## optimization problem. Interpret results and fit with great care. For
## unconstrained estimation, set bounded = FALSE.

## Warning in addSEs_cpp(x, verbose = verbose, approximate_SEs = approximate_SEs):
## Standard errors could not be obtained because the Fischer information matrix
## could not be inverted. This may be a symptom of a non-identified model or due
## to convergence issues. You can try to approximate standard errors by setting
## approximate_SEs = TRUE at own risk.

## Warning in runmodel(.): Model might not have converged properly:
## mean(abs.gradient)) > 1.

##           Measure           Value
##           logl             -853.84
## unrestricted.logl         -853.84
##           baseline.logl      -24.85
##           nvar                8
##           nobs               44
##           npar               44
##           df                  ~ 0
##           objective -21121407017356.47
##           chisq              ~ 0
##           pvalue              1
##           baseline.chisq      -1657.98
##           baseline.npar        22
##           baseline.df         22
##           baseline.pvalue      1
##           nfi                 1
##           pnfi
##           tli
##           nnfi                1
##           rfi
##           ifi                 1
##           rni                 1
##           cfi                 1
##           rmsea
##           rmsea.ci.lower      ~ 0
##           rmsea.ci.upper      ~ 0

```

```

##          rmsea.pvalue          ~ 0
##          aic.ll              1795.67
##          aic.ll2             1562.73
##          aic.x                ~ 0
##          aic.x2               88
##          bic                 1854.29
##          bic2                1717.49
##          ebic.25             1945.79
##          ebic.5              2037.28
##          ebic.75             2110.48
##          ebic1               2220.27

##
## Parameters for group fullsample
## - mu
##      var1 op var2    est se p row col par
## a_v_lag1 ~1    -0.54      1  1  1
## a_a_lag1 ~1     0.41      2  1  2
## t_v_lag1 ~1    -0.23      3  1  3
## t_a_lag1 ~1    -0.20      4  1  4
##      a_v ~1    -0.56      5  1  5
##      a_a ~1     0.36      6  1  6
##      t_v ~1    -0.21      7  1  7
##      t_a ~1    -0.19      8  1  8
##
## - exo_cholesky
##      var1      op      var2      est se p row col par
## a_v_lag1 ~chol~ a_v_lag1  0.78      1  1  9
## a_a_lag1 ~chol~ a_v_lag1  0.30      2  1 10
## t_v_lag1 ~chol~ a_v_lag1  0.12      3  1 11
## t_a_lag1 ~chol~ a_v_lag1  0.070     4  1 12
## a_a_lag1 ~chol~ a_a_lag1  0.40      2  2 13
## t_v_lag1 ~chol~ a_a_lag1 -0.060     3  2 14
## t_a_lag1 ~chol~ a_a_lag1  0.047     4  2 15
## t_v_lag1 ~chol~ t_v_lag1  0.61      3  3 16
## t_a_lag1 ~chol~ t_v_lag1  0.16      4  3 17
## t_a_lag1 ~chol~ t_a_lag1  0.43      4  4 18
##
## - beta
## var1 op var2      est se p row col par
## a_v <- a_v  0.26      1  1 19
## a_a <- a_v -0.13      2  1 20
## t_v <- a_v  0.16      3  1 21
## t_a <- a_v  0.30      4  1 22
## a_v <- a_a -0.020     1  2 23
## a_a <- a_a  0.19      2  2 24
## t_v <- a_a  0.31      3  2 25
## t_a <- a_a  0.24      4  2 26
## a_v <- t_v -0.058     1  3 27
## a_a <- t_v -0.066     2  3 28
## t_v <- t_v  0.072     3  3 29
## t_a <- t_v  0.018     4  3 30
## a_v <- t_a  0.028     1  4 31
## a_a <- t_a -0.22      2  4 32

```

```

##   t_v <- t_a 0.56      3  4 33
##   t_a <- t_a -0.14     4  4 34
##
## - omega_zeta (symmetric)
## var1 op var2      est se p row col par
##   a_a -- a_v -0.098      2  1 35
##   t_v -- a_v 0.022      3  1 36
##   t_a -- a_v -0.042     4  1 37
##   t_v -- a_a 0.021      3  2 38
##   t_a -- a_a -0.040     4  2 39
##   t_a -- t_v 0.12      4  3 40
##
## - delta_zeta (diagonal)
## var1 op var2      est se p row col par
##   a_v ~/~ a_v 0.00065      1  1 41
##   a_a ~/~ a_a ~ 0         2  2 42
##   t_v ~/~ t_v 0.49        3  3 43
##   t_a ~/~ t_a 0.38        4  4 44

##
## Top 20 modification indices:
##
##      var1      op      var2      est mi pmi epc      matrix row col
##   a_v_lag1    ~1    <NA> -0.54207173      mu    1    1
##   a_a_lag1    ~1    <NA> 0.40633947      mu    2    1
##   t_v_lag1    ~1    <NA> -0.22658388      mu    3    1
##   t_a_lag1    ~1    <NA> -0.20450791      mu    4    1
##       a_v     ~1    <NA> -0.55663358      mu    5    1
##       a_a     ~1    <NA> 0.36338340      mu    6    1
##       t_v     ~1    <NA> -0.21383745      mu    7    1
##       t_a     ~1    <NA> -0.19409992      mu    8    1
##   a_v_lag1 ~chol~ a_v_lag1 0.77990210    exo_cholesky 1    1
##   a_a_lag1 ~chol~ a_v_lag1 0.29720846    exo_cholesky 2    1
##   t_v_lag1 ~chol~ a_v_lag1 0.11917505    exo_cholesky 3    1
##   t_a_lag1 ~chol~ a_v_lag1 0.07048973    exo_cholesky 4    1
##   a_a_lag1 ~chol~ a_a_lag1 0.40433821    exo_cholesky 2    2
##   t_v_lag1 ~chol~ a_a_lag1 -0.06025043    exo_cholesky 3    2
##   t_a_lag1 ~chol~ a_a_lag1 0.04667463    exo_cholesky 4    2
##   t_v_lag1 ~chol~ t_v_lag1 0.61446661    exo_cholesky 3    3
##   t_a_lag1 ~chol~ t_v_lag1 0.15950468    exo_cholesky 4    3
##   t_a_lag1 ~chol~ t_a_lag1 0.42731677    exo_cholesky 4    4
##       a_v     <-      a_v 0.26001931      beta    1    1
##       a_a     <-      a_v -0.12590094      beta    2    1
##
##      group group_id
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1
##   fullsample      1

```

```
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
## fullsample      1
```

