

# Análise de dados - Irisa

2025-04-29

## Estatísticas descritivas

```
# Pacotes são carregados no setup

colunas_interesse_nomes <- c("Grupo", "Idade", "Sexo", "Estatura (m)",
                             "Massa Corporal (Kg) Avaliação 1", "IMC avaliação 1",
                             "IDATE-E_Sessão 1_Pré", "PANAS-SESSÃO 1_Pré_AFETO NEGATIVO",
                             "PANAS-SESSÃO 1_Pré_AFETO POSITIVO")

descritivas_gt <- irisa %>%
  dplyr::select(all_of(colunas_interesse_nomes)) %>%
  mutate(Grupo = factor(Gruo), Sexo = factor(Sexo)) %>%
  group_by(Gruo) %>%
  summarise(
    N = n(), Idade_Media = mean(Idade, na.rm = TRUE), Idade_DP = sd(Idade, na.rm = TRUE),
    Estatura_Media = mean(`Estatura (m)`, na.rm = TRUE), Estatura_DP = sd(`Estatura (m)`, na.rm = TRUE),
    Massa_Media = mean(`Massa Corporal (Kg) Avaliação 1`, na.rm = TRUE), Massa_DP = sd(`Massa Corporal
    IMC_Media = mean(`IMC avaliação 1`, na.rm = TRUE), IMC_DP = sd(`IMC avaliação 1`, na.rm = TRUE),
    IDATE_Media = mean(`IDATE-E_Sessão 1_Pré`, na.rm = TRUE), IDATE_DP = sd(`IDATE-E_Sessão 1_Pré`, na.
    PANAS_Neg_Media = mean(`PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`, na.rm = TRUE), PANAS_Neg_DP = sd(`PANAS
    PANAS_Pos_Media = mean(`PANAS-SESSÃO 1_Pré_AFETO POSITIVO`, na.rm = TRUE), PANAS_Pos_DP = sd(`PANAS
    Sexo_M_N = sum(Sexo == "M", na.rm = TRUE), Sexo_F_N = sum(Sexo == "F", na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  mutate(
    Sexo_Total_N = Sexo_M_N + Sexo_F_N,
    Sexo_M_Perc = ifelse(Sexo_Total_N > 0, (Sexo_M_N / Sexo_Total_N) * 100, 0),
    Sexo_F_Perc = ifelse(Sexo_Total_N > 0, (Sexo_F_N / Sexo_Total_N) * 100, 0)
  )

descritivas_formatadas <- descritivas_gt %>%
  mutate(
    `Idade` = paste0(sprintf("%.2f", Idade_Media), " (", sprintf("%.2f", Idade_DP), ")"),
    `Estatura (m)` = paste0(sprintf("%.2f", Estatura_Media), " (", sprintf("%.2f", Estatura_DP), ")"),
    `Massa Corporal (Kg) Avaliação 1` = paste0(sprintf("%.2f", Massa_Media), " (", sprintf("%.2f", Mass
    `IMC avaliação 1` = paste0(sprintf("%.2f", IMC_Media), " (", sprintf("%.2f", IMC_DP), ")"),
    `IDATE-E_Sessão 1_Pré` = paste0(sprintf("%.2f", IDATE_Media), " (", sprintf("%.2f", IDATE_DP), ")"),
    `PANAS-SESSÃO 1_Pré_AFETO NEGATIVO` = paste0(sprintf("%.2f", PANAS_Neg_Media), " (", sprintf("%.2f"
    `PANAS-SESSÃO 1_Pré_AFETO POSITIVO` = paste0(sprintf("%.2f", PANAS_Pos_Media), " (", sprintf("%.2f"
    `Sexo (Masc)` = paste0(Sexo_M_N, " (", sprintf("%.1f", Sexo_M_Perc), "%)"),
    `Sexo (Fem)` = paste0(Sexo_F_N, " (", sprintf("%.1f", Sexo_F_Perc), "%)"),
    N = as.character(N)
  ) %>%
```

```

dplyr::select(
  Grupo, N, `Idade`, `Estatura (m)`, `Massa Corporal (Kg) Avaliação 1`,
  `IMC avaliação 1`, `IDATE-E_Sessão 1_Pré`, `PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`,
  `PANAS-SESSÃO 1_Pré_AFETO POSITIVO`, `Sexo (Masc)`, `Sexo (Fem)`
)

descritivas_transposta <- descritivas_formatadas %>%
  pivot_longer(
    cols = -Grupo,
    names_to = "Métrica",
    values_to = "Valor" # Coluna de valores é "Valor"
  ) %>%
  pivot_wider(
    names_from = Grupo,
    values_from = Valor # <-- ESPECIFICA A COLUNA CORRETA
  )

kable(
  descritivas_transposta,
  format = "pipe",
  caption = "", # Removido caption aqui para usar texto markdown abaixo
  col.names = c("Métrica", "Grupo 1", "Grupo 2"),
  align = 'l'
)

```

Métrica	Grupo 1	Grupo 2
N	17	14
Idade	26.59 (6.08)	25.00 (5.74)
Estatura (m)	1.66 (0.09)	1.65 (0.11)
Massa Corporal (Kg) Avaliação 1	66.63 (15.21)	65.35 (15.42)
IMC avaliação 1	24.21 (4.99)	23.98 (5.11)
IDATE-E_Sessão 1_Pré	48.88 (13.55)	44.36 (9.80)
PANAS-SESSÃO 1_Pré_AFETO NEGATIVO	20.71 (10.77)	16.71 (4.65)
PANAS-SESSÃO 1_Pré_AFETO POSITIVO	35.12 (6.23)	33.36 (7.32)
Sexo (Masc)	4 (23.5%)	1 (7.1%)
Sexo (Fem)	13 (76.5%)	13 (92.9%)

*Valores para métricas contínuas apresentados como Média (DP). DP = Desvio Padrão. Valores para Sexo apresentados como N (%).*

**Verificação sobre a distribuição dos dados** Dividindo os dados por grupo e testando a normalidade das variáveis

```

grupo1_data <- irisa[irisa$Grupo == 1, ] # Seleciona linhas onde Grupo é 1, e todas as colunas
grupo2_data <- irisa[irisa$Grupo == 2, ] # Seleciona linhas onde Grupo é 2, e todas as colunas

variaveis_numericas_teste <- c("Idade",
                                "Estatura (m)",
                                "Massa Corporal (Kg) Avaliação 1",
                                "IMC avaliação 1",

```

```

        "IDATE-E_Sessão 1_Pré",
        "PANAS-SESSÃO 1_Pré_AFETO NEGATIVO",
        "PANAS-SESSÃO 1_Pré_AFETO POSITIVO")

# --- Teste para o Grupo 1 ---

resultados_shapiro_grupo1 <- lapply(variaveis_numericas_teste, function(nome_var) {
  dados_coluna <- grupo1_data[[nome_var]]
  dados_coluna_limpos <- na.omit(dados_coluna)
  if (length(dados_coluna_limpos) >= 3) {
    tryCatch({
      shapiro.test(dados_coluna_limpos)
    }, error = function(e) {
      list(statistic = NA, p.value = NA, method = paste("Erro no Shapiro-Wilk para", nome_var, ":", e$message))
    })
  } else {
    list(statistic = NA, p.value = NA, method = paste("Shapiro-Wilk: Pontos de dados insuficientes (<3)"))
  }
})

names(resultados_shapiro_grupo1) <- variaveis_numericas_teste

# --- Teste para o Grupo 2 ---

# Repetir o processo para o grupo 2
resultados_shapiro_grupo2 <- lapply(variaveis_numericas_teste, function(nome_var) {
  dados_coluna <- grupo2_data[[nome_var]]
  dados_coluna_limpos <- na.omit(dados_coluna)
  if (length(dados_coluna_limpos) >= 3) {
    tryCatch({
      shapiro.test(dados_coluna_limpos)
    }, error = function(e) {
      list(statistic = NA, p.value = NA, method = paste("Erro no Shapiro-Wilk para", nome_var, ":", e$message))
    })
  } else {
    list(statistic = NA, p.value = NA, method = paste("Shapiro-Wilk: Pontos de dados insuficientes (<3)"))
  }
})

names(resultados_shapiro_grupo2) <- variaveis_numericas_teste

# --- Visualizar os Resultados ---

tabela_shapiro_g1 <- data.frame(
  Variavel = names(resultados_shapiro_grupo1),
  W_Statistic = sapply(resultados_shapiro_grupo1, function(x) ifelse(is.list(x) && !is.null(x$statistic), x$statistic, NA)),
  P_Value = sapply(resultados_shapiro_grupo1, function(x) ifelse(is.list(x) && !is.null(x$p.value), x$p.value, NA)),
  row.names = NULL # Remover nomes de linha automáticos
)

tabela_shapiro_g2 <- data.frame(
  Variavel = names(resultados_shapiro_grupo2),
  W_Statistic = sapply(resultados_shapiro_grupo2, function(x) ifelse(is.list(x) && !is.null(x$statistic), x$statistic, NA)),
  P_Value = sapply(resultados_shapiro_grupo2, function(x) ifelse(is.list(x) && !is.null(x$p.value), x$p.value, NA))
)

```

```

row.names = NULL
)

tabela_shapiro_combinada <- left_join(
  tabela_shapiro_g1,
  tabela_shapiro_g2,
  by = "Variavel",
  suffix = c("_G1", "_G2")
)

# Selecionar e Renomear colunas para a tabela final
tabela_shapiro_formatada <- tabela_shapiro_combinada %>%
  dplyr::select(
    Variavel,
    W_Statistic_G1, P_Value_G1,
    W_Statistic_G2, P_Value_G2
  ) %>%
  rename(
    "W Grupo 1" = W_Statistic_G1,
    "p Grupo 1" = P_Value_G1,
    "W Grupo 2" = W_Statistic_G2,
    "p Grupo 2" = P_Value_G2
  )

kable(
  tabela_shapiro_formatada,
  format = "pipe", # Formato bom para visualização no console
  digits = 3,      # Número de casas decimais para os valores W e p
  caption = "Resultados do Teste de Normalidade Shapiro-Wilk por Grupo",
  # col.names pode ser omitido se você usou rename() como acima,
  # caso contrário, use para definir os cabeçalhos:
  # col.names = c("Variável", "W Grupo 1", "p Grupo 1", "W Grupo 2", "p Grupo 2")
  align = c('l', 'c', 'c', 'c', 'c') # Alinhamento: l=left, c=center, r=right
)

```

Table 2: Resultados do Teste de Normalidade Shapiro-Wilk por Grupo

Variavel	W Grupo 1	p Grupo 1	W Grupo 2	p Grupo 2
Idade	0.933	0.241	0.876	0.051
Estatura (m)	0.929	0.208	0.852	0.023
Massa Corporal (Kg) Avaliação 1	0.918	0.136	0.931	0.311
IMC avaliação 1	0.954	0.517	0.933	0.339
IDATE-E_Sessão 1_Pré	0.923	0.169	0.962	0.754
PANAS-SESSÃO 1_Pré_AFETO NEGATIVO	0.835	0.006	0.885	0.069
PANAS-SESSÃO 1_Pré_AFETO POSITIVO	0.935	0.261	0.906	0.136

```
tabela_contingencia <- table(irisa$Sexo, irisa$Grupo)
fisher.test(tabela_contingencia)
```

*Apenas estatura e Afetos negativos não seguem uma distribuição normal.*

```
##
## Fisher's Exact Test for Count Data
##
## data:  tabela_contingencia
## p-value = 0.3445
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.004726187 3.116026293
## sample estimates:
## odds ratio
##  0.2603913
```

```
irisa <- irisa %>% mutate(Gruo = factor(Gruo))

# Definir as variáveis
vars_teste_t <- c("Idade", "Massa Corporal (Kg) Avaliação 1", "IMC avaliação 1",
                  "IDATE-E_Sessão 1_Pré", "PANAS-SESSÃO 1_Pré_AFETO POSITIVO")
vars_teste_wilcoxon <- c("Estatura (m)", "PANAS-SESSÃO 1_Pré_AFETO NEGATIVO")

# Inicializar a lista
resultados_testes <- list()

# --- Loop Teste t ---
for (variavel in vars_teste_t) {
  formula_teste <- as.formula(paste0("`", variavel, "` ~ Gruo"))

  teste_t_resultado <- irisa %>%
    rstatix::t_test(formula = formula_teste, var.equal = FALSE)

  resultados_testes[[variavel]] <- data.frame(
    Variavel = variavel,
    Teste = "T-test",
    Estatistica = paste0("t(", round(teste_t_resultado$df, 2), ") = ", round(teste_t_resultado$statistic, 2)),
    P_Value = teste_t_resultado$p
  )
}

# --- Loop Wilcoxon ---
for (variavel in vars_teste_wilcoxon) {
  formula_teste <- as.formula(paste0("`", variavel, "` ~ Gruo"))

  wilcox_test_resultado <- irisa %>%
    rstatix::wilcox_test(formula = formula_teste)
```

```

resultados_testes[[variavel]] <- data.frame(
  Variavel = variavel,
  Teste = "Mann-Whitney",
  # Usar a estatística e p-valor do wilcox_test
  Estatistica = paste0("W = ", wilcox_test_resultado$statistic),
  P_Value = wilcox_test_resultado$p
)
}

# Combinar todos os resultados
tabela_resultados_finais <- bind_rows(resultados_testes)

#Gerar a tabela com kable
kable(
  tabela_resultados_finais,
  format = "pipe",
  digits = 3,
  caption = "Resultados dos Testes de Comparação entre Grupos",
  col.names = c("Variável", "Teste Aplicado", "Estatística", "Valor-p"),
  align = c('l', 'l', 'c', 'c'),
  rownames = FALSE
)

```

*Não existe diferença estatisticamente significativa sobre quantidade de pessoas de cada sexo em cada grupo.*

Table 3: Resultados dos Testes de Comparação entre Grupos

Variável	Teste Aplicado	Estatística	Valor-p
Idade	T-test	t(28.42) = 0.75	0.462
Massa Corporal (Kg) Avaliação 1	T-test	t(27.72) = 0.23	0.819
IMC avaliação 1	T-test	t(27.61) = 0.13	0.900
IDATE-E_Sessão 1_Pré	T-test	t(28.59) = 1.08	0.291
PANAS-SESSÃO 1_Pré_AFETO POSITIVO	T-test	t(25.71) = 0.71	0.483
Estatura (m)	Mann-Whitney	W = 132.5	0.605
PANAS-SESSÃO 1_Pré_AFETO NEGATIVO	Mann-Whitney	W = 130.5	0.661

*Não existe diferença estatisticamente significativa entre os grupos no tempo 1.*

Teste sobre a distribuição dos dados entre as variáveis de interesse no tempo “pós”

```

#IDATE
shapiro.test(grupo1_data$`IDATE-E_Sessão 1_Pré`)

```

*As variáveis de IDATE em todos os grupos seguem uma distribuição normal, o que não verdade para as outras variáveis.*

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`IDATE-E_Sessão 1_Pré`
## W = 0.92342, p-value = 0.1687
```

```
shapiro.test(grupo1_data$`IDATE-E_Sessão 1_Pós`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`IDATE-E_Sessão 1_Pós`
## W = 0.90935, p-value = 0.09754
```

```
shapiro.test(grupo2_data$`IDATE-E_Sessão 1_Pré`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`IDATE-E_Sessão 1_Pré`
## W = 0.96189, p-value = 0.754
```

```
shapiro.test(grupo2_data$`IDATE-E_Sessão 1_Pós`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`IDATE-E_Sessão 1_Pós`
## W = 0.91812, p-value = 0.2066
```

```
#PANAS Positivos
```

```
shapiro.test(grupo1_data$`PANAS-SESSÃO 1_Pré_AFETO POSITIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`PANAS-SESSÃO 1_Pré_AFETO POSITIVO`
## W = 0.93473, p-value = 0.2609
```

```
shapiro.test(grupo1_data$`PANAS-SESSÃO 1_Pós_AFETO POSITIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`PANAS-SESSÃO 1_Pós_AFETO POSITIVO`
## W = 0.92066, p-value = 0.1515
```

```
shapiro.test(grupo2_data$`PANAS-SESSÃO 1_Pré_AFETO POSITIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`PANAS-SESSÃO 1_Pré_AFETO POSITIVO`
## W = 0.90567, p-value = 0.1364
```

```
shapiro.test(grupo2_data$`PANAS-SESSÃO 1_Pós_AFETO POSITIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`PANAS-SESSÃO 1_Pós_AFETO POSITIVO`
## W = 0.86486, p-value = 0.03556
```

*#PANAS Negativos*

```
shapiro.test(grupo1_data$`PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`
## W = 0.83461, p-value = 0.006302
```

```
shapiro.test(grupo1_data$`PANAS-SESSÃO 1_Pós_AFETO NEGATIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo1_data$`PANAS-SESSÃO 1_Pós_AFETO NEGATIVO`
## W = 0.86447, p-value = 0.01797
```

```
shapiro.test(grupo2_data$`PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`
## W = 0.88528, p-value = 0.06913
```

```
shapiro.test(grupo2_data$`PANAS-SESSÃO 1_Pós_AFETO NEGATIVO`)
```

```
##
## Shapiro-Wilk normality test
##
## data: grupo2_data$`PANAS-SESSÃO 1_Pós_AFETO NEGATIVO`
## W = 0.76773, p-value = 0.002046
```

## ANOVAs fatoriais de medidas repetidas

### IDATE-E



```

names(iris)[names(iris) == "Código"] <- "ID"

data_long_idate <- iris %>%
  dplyr::select(
    ID, Grupo,
    `IDATE-E_Sessão 1_Pré`, `IDATE-E_Sessão 1_Pós`
  ) %>%
  pivot_longer(
    cols = c(`IDATE-E_Sessão 1_Pré`, `IDATE-E_Sessão 1_Pós`),
    names_to = c(".value", "Tempo"),
    names_pattern = "(.*)_(Pré|Pós)$",
    values_to = "Score"
  ) %>%
  rename(Score = `IDATE-E_Sessão 1`) %>%
  mutate(
    Tempo = case_when(
      Tempo == "Pré" ~ 1,
      Tempo == "Pós" ~ 2,
      TRUE ~ NA_real_
    ),
    ID = as.numeric(ID)
  ) %>%
  filter(!is.na(Score))

str(data_long_idate)

## tibble [62 x 4] (S3: tbl_df/tbl/data.frame)
## $ ID : num [1:62] 101 101 102 102 103 103 104 104 106 106 ...
## $ Grupo: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
## $ Tempo: num [1:62] 1 2 1 2 1 2 1 2 1 2 ...
## $ Score: num [1:62] 48 33 33 24 44 32 44 36 34 27 ...

```

```

anova_idate_param <- anova_test(
  data = data_long_idate,
  dv = Score,           # Variável dependente
  wid = ID,             # Identificador do sujeito
  between = Grupo,      # Fator entre-grupos
  within = Tempo,       # Fator intra-sujeitos
  effect.size = "ges",  # Generalized Eta Squared
  type = 3              # Tipo III SS
)

get_anova_table(anova_idate_param)

```

*A ANOVA paramétrica não encontrou efeito da interação entre grupo e tempo nos escores para IDATE, apenas dentro dos grupos (pré e pós).*

```
## ANOVA Table (type III tests)
```

```
##
##      Effect DFn DFd      F      p p<.05      ges
## 1      Grupo   1  29  0.262 6.13e-01      0.008
## 2      Tempo   1  29 36.325 1.48e-06      * 0.165
## 3 Grupo:Tempo   1  29  3.162 8.60e-02      0.017
```

```
posthoc_grupo_por_tempo <- data_long_idate %>%
  group_by(Grupo) %>%
  t_test(Score ~ Tempo, paired = TRUE, var.equal = FALSE) %>% # Teste t
  adjust_pvalue(method = "bonferroni") # Ajusta p-valor para os 2 testes

print(posthoc_grupo_por_tempo)
```

Se quiser vasculhar dentro dos grupos, ambos tiveram diminuição estatisticamente significativa

```
## # A tibble: 2 x 10
##   Grupo .y. group1 group2   n1   n2 statistic    df      p    p.adj
##   <fct> <chr> <chr>  <chr> <int> <int>    <dbl> <dbl>    <dbl>    <dbl>
## 1 1      Score 1      2      17   17      4.91    16 0.000156 0.000312
## 2 2      Score 1      2      14   14      4.01    13 0.00148  0.00296
```

## PANAS Afetos negativos

```
data_long_panas_neg <- irisa %>%
  dplyr::select(
    ID, Grupo,
    `PANAS-SESSÃO 1_Pré_AFETO NEGATIVO`, `PANAS-SESSÃO 1_Pós_AFETO NEGATIVO`
  ) %>%
  pivot_longer(
    cols = -c(ID, Grupo),
    names_to = "Nome_Original",
    values_to = "Score"
  ) %>%
  mutate(
    Tempo = case_when(
      grepl("_Pré_", Nome_Original, fixed = TRUE) ~ 1,
      grepl("_Pós_", Nome_Original, fixed = TRUE) ~ 2,
      TRUE ~ NA_real_
    ),
    Variavel = "PANAS_AFETO NEGATIVO",
    ID = as.numeric(ID)
  ) %>%
  dplyr::select(ID, Grupo, Variavel, Tempo, Score) %>%
  filter(!is.na(Score) & !is.na(Tempo))

mod_panas_neg <- nparLD(formula = Score ~ Tempo * Grupo,
```

```

data = data_long_panas_neg,
subject = data_long_panas_neg$ID,
description = FALSE
)

```

O mesmo aconteceu para PANAS, utilizando a ANOVA não paramétrica

```

## F1 LD F1 Model
## -----
## Check that the order of the time and group levels are correct.
## Time level: 1 2
## Group level: 1 2
## If the order is not correct, specify the correct order in time.order or group.order.

```

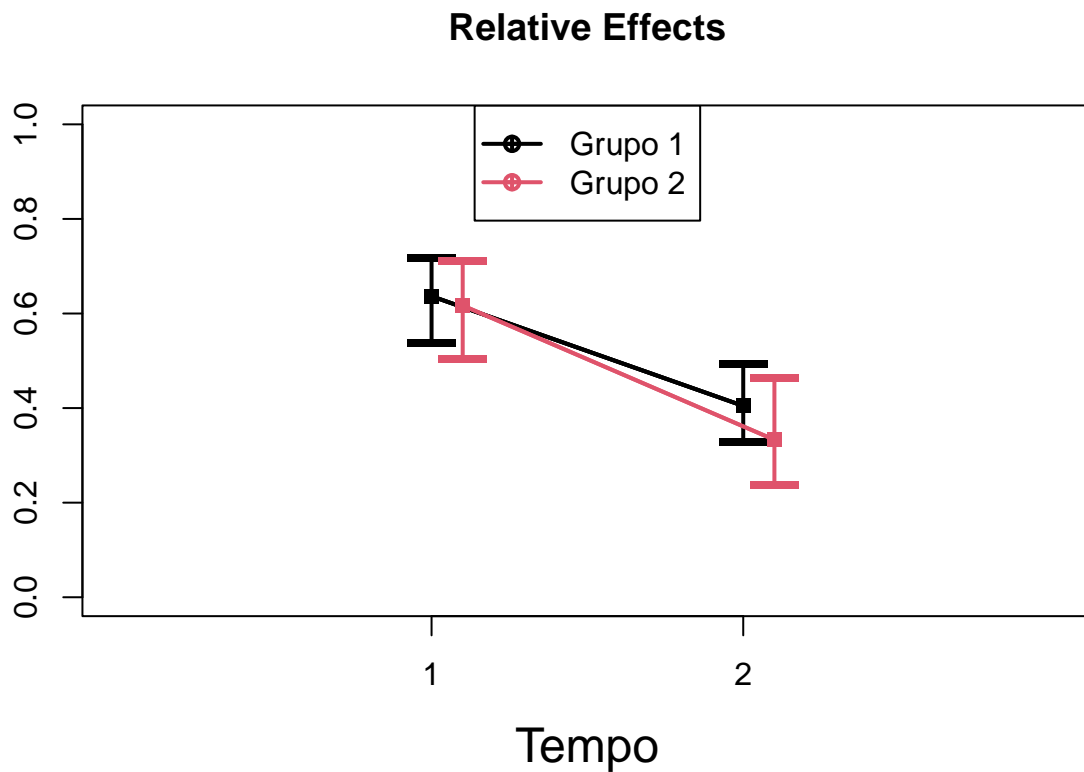
```
summary(mod_panas_neg)
```

```

## Model:
## F1 LD F1 Model
##
## Call:
## Score ~ Tempo * Grupo
##
## Relative Treatment Effect (RTE):
##           RankMeans Nobs      RTE
## Grupo1      32.77941   34 0.5206357
## Grupo2      29.94643   28 0.4749424
## Tempo1      39.36029   31 0.6267789
## Tempo2      23.36555   31 0.3687991
## Grupo1:Tempo1 39.97059   17 0.6366224
## Grupo1:Tempo2 25.58824   17 0.4046490
## Grupo2:Tempo1 38.75000   14 0.6169355
## Grupo2:Tempo2 21.14286   14 0.3329493
##
## Wald-Type Statistic (WTS):
##           Statistic df      p-value
## Grupo      0.2786623  1 5.975793e-01
## Tempo     46.0541665  1 1.150283e-11
## Grupo:Tempo 0.4680120  1 4.939032e-01
##
## ANOVA-Type Statistic (ATS):
##           Statistic df      p-value
## Grupo      0.2786623  1 5.975793e-01
## Tempo     46.0541665  1 1.150283e-11
## Grupo:Tempo 0.4680120  1 4.939032e-01
##
## Modified ANOVA-Type Statistic for the Whole-Plot Factors:
##           Statistic df1      df2      p-value
## Grupo 0.2786623    1 28.90494 0.6016087

```

```
plot(mod_panas_neg)
```



```
data_long_panas_neg1 <- data_long_panas_neg[data_long_panas_neg$Grupo == 1, ]
data_long_panas_neg2 <- data_long_panas_neg[data_long_panas_neg$Grupo == 2, ]

posthoc_2x1_inter <- npar.t.test.paired(Score ~ Tempo, data_long_panas_neg1,
                                         conf.level = 0.95,
                                         alternative = c("two.sided"), plot=F)
```

Dá pra “vasculhar” também. Verifica-se que em ambos os grupos houve uma diminuição estatisticamente significativa

```
##
## #-----Nonparametric Paired t Test-----#
##
## - Sample Size: 17
## - Factor Levels: 1 2
## - H0: p=1/2
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence Level: 95 %
## - Method: Brunner-Munzel (BM), Permutation (PERM)
##
## #-----#
##
```

```
summary(posthoc_2x1_inter)
```

```
##
## # Nonparametric Paired t Test Procedures and Confidence Intervals for the relative effect #
##
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence level: 95 %
## - Method = Brunner-Munzel Test (BM), Studentized Permutation Test (PERM)
## #-----Interpretation-----#
## p(a,b) > 1/2 : b tends to be larger than a
## #-----#
##
## #---Data Info-----#
## Sample Size
## 1      1      17
## 2      2      17
##
## #---Analysis-----#
## Lower p.hat Upper      T p.value
## BM  0.176 0.272 0.367 -5.077      0
## PERM 0.181 0.272 0.362 -5.077      0
##
## #-----#
```

```
posthoc_2x1_contr <- npar.t.test.paired(Score ~ Tempo, data_long_panas_neg2,
                                       conf.level = 0.95,
                                       alternative = c("two.sided"),plot=F)
```

```
##
## #-----Nonparametric Paired t Test-----#
##
## - Sample Size: 14
## - Factor Levels: 1 2
## - H0: p=1/2
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence Level: 95 %
## - Method: Brunner-Munzel (BM), Permutation (PERM)
##
## #-----#
##
```

```
summary(posthoc_2x1_contr)
```

```
##
## # Nonparametric Paired t Test Procedures and Confidence Intervals for the relative effect #
##
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence level: 95 %
## - Method = Brunner-Munzel Test (BM), Studentized Permutation Test (PERM)
## #-----Interpretation-----#
## p(a,b) > 1/2 : b tends to be larger than a
```

```
## #-----#
##
## #---Data Info-----#
## Sample Size
## 1      1    14
## 2      2    14
##
## #---Analysis-----#
## Lower p.hat Upper      T p.value
## BM  0.090 0.219 0.349 -4.681  0.000
## PERM 0.097 0.219 0.342 -4.681  0.001
##
## #-----#
```

## PANAS Afetos positivos

```
data_long_panas_pos <- irisa %>%
  dplyr::select(
    ID, Grupo,
    `PANAS-SESSÃO 1_Pré_AFETO POSITIVO`, `PANAS-SESSÃO 1_Pós_AFETO POSITIVO`
  ) %>%
  pivot_longer(
    cols = -c(ID, Grupo),
    names_to = "Nome_Original",
    values_to = "Score"
  ) %>%
  mutate(
    Tempo = case_when(
      grepl("_Pré_", Nome_Original, fixed = TRUE) ~ 1,
      grepl("_Pós_", Nome_Original, fixed = TRUE) ~ 2,
      TRUE ~ NA_real_
    ),
    Variavel = "PANAS_AFETO POSITIVO",
    ID = as.numeric(ID)
  ) %>%
  dplyr::select(ID, Grupo, Variavel, Tempo, Score) %>%
  filter(!is.na(Score) & !is.na(Tempo))

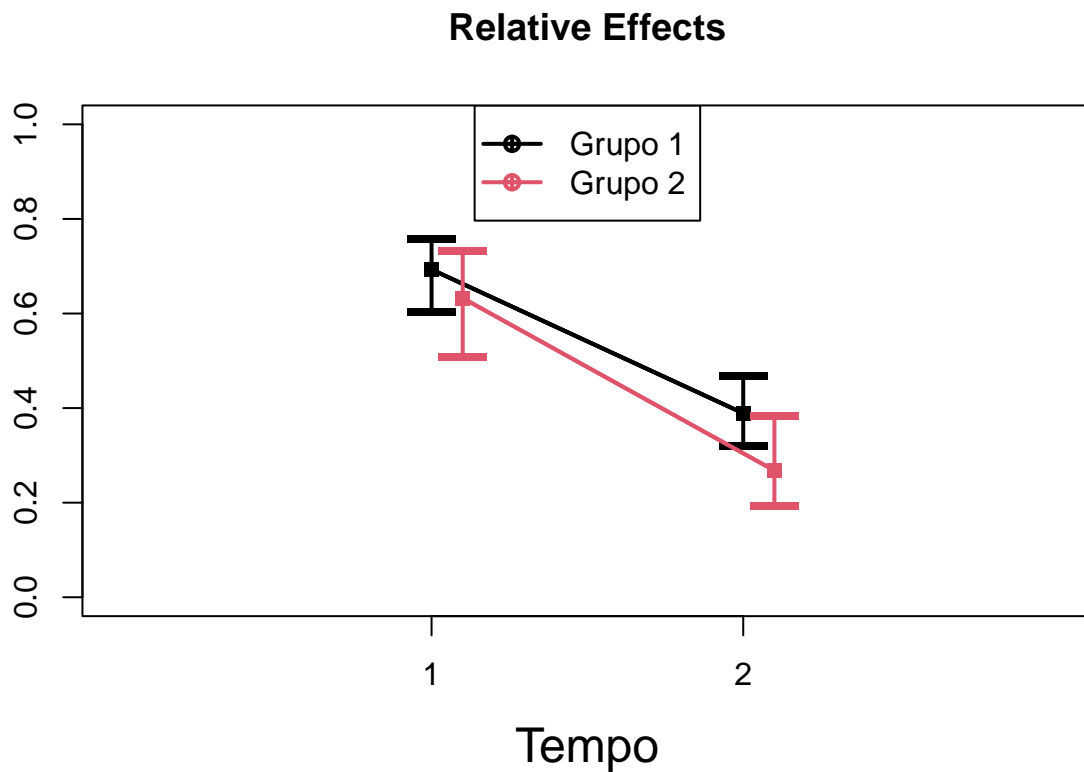
mod_panas_pos <- nparLD(formula = Score ~ Tempo * Grupo,
  data = data_long_panas_pos,
  subject = data_long_panas_pos$ID,
  description = FALSE
)
```

```
## F1 LD F1 Model
## -----
## Check that the order of the time and group levels are correct.
## Time level: 1 2
## Group level: 1 2
## If the order is not correct, specify the correct order in time.order or group.order.
```

```
summary(mod_panas_pos)
```

```
## Model:
## F1 LD F1 Model
##
## Call:
## Score ~ Tempo * Grupo
##
## Relative Treatment Effect (RTE):
##           RankMeans Nobs      RTE
## Grupo1      34.04412   34 0.5410342
## Grupo2      28.41071   28 0.4501728
## Tempo1      41.59244   31 0.6627812
## Tempo2      20.86239   31 0.3284257
## Grupo1:Tempo1 43.47059   17 0.6930740
## Grupo1:Tempo2 24.61765   17 0.3889943
## Grupo2:Tempo1 39.71429   14 0.6324885
## Grupo2:Tempo2 17.10714   14 0.2678571
##
## Wald-Type Statistic (WTS):
##           Statistic df      p-value
## Grupo      1.3314048  1 2.485555e-01
## Tempo     78.5245853  1 7.900879e-19
## Grupo:Tempo 0.6438438  1 4.223223e-01
##
## ANOVA-Type Statistic (ATS):
##           Statistic df      p-value
## Grupo      1.3314048  1 2.485555e-01
## Tempo     78.5245853  1 7.900879e-19
## Grupo:Tempo 0.6438438  1 4.223223e-01
##
## Modified ANOVA-Type Statistic for the Whole-Plot Factors:
##           Statistic df1    df2    p-value
## Grupo  1.331405    1 27.48782 0.2584897
```

```
plot(mod_panas_pos)
```



Dá pra vasculhar também e verificar que em ambos os grupos houve uma diminuição estatisticamente significativa

```
data_long_panas_pos1 <- data_long_panas_pos[data_long_panas_pos$Grupo == 1, ]
data_long_panas_pos2 <- data_long_panas_pos[data_long_panas_pos$Grupo == 2, ]

posthoc_2x1_inter <- npar.t.test.paired(Score ~ Tempo, data_long_panas_pos1,
                                       conf.level = 0.95,
                                       alternative = c("two.sided"), plot=F)
```

```
##
## #-----Nonparametric Paired t Test-----#
##
## - Sample Size: 17
## - Factor Levels: 1 2
## - H0: p=1/2
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence Level: 95 %
## - Method: Brunner-Munzel (BM), Permutation (PERM)
##
## #-----#
##
```

```
summary(posthoc_2x1_inter)
```

```
##
```



```
## # Nonparametric Paired t Test Procedures and Confidence Intervals for the relative effect #
##
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence level: 95 %
## - Method = Brunner-Munzel Test (BM), Studentized Permutation Test (PERM)
## #-----Interpretation-----#
## p(a,b) > 1/2 : b tends to be larger than a
## #-----#
##
## #---Data Info-----#
## Sample Size
## 1      1      17
## 2      2      17
##
## #---Analysis-----#
## Lower p.hat Upper      T p.value
## BM  0.094 0.182 0.269 -7.697      0
## PERM 0.097 0.182 0.266 -7.697      0
##
## #-----#
```

```
posthoc_2x1_contr <- npar.t.test.paired(Score ~ Tempo, data_long_panas_pos2,
                                       conf.level = 0.95,
                                       alternative = c("two.sided"),plot=F)
```

```
##
## #-----Nonparametric Paired t Test-----#
##
## - Sample Size: 14
## - Factor Levels: 1 2
## - H0: p=1/2
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence Level: 95 %
## - Method: Brunner-Munzel (BM), Permutation (PERM)
##
## #-----#
##
```

```
summary(posthoc_2x1_contr)
```

```
##
## # Nonparametric Paired t Test Procedures and Confidence Intervals for the relative effect #
##
## - Alternative Hypothesis: True relative effect is less or greater than 1/2
## - Confidence level: 95 %
## - Method = Brunner-Munzel Test (BM), Studentized Permutation Test (PERM)
## #-----Interpretation-----#
## p(a,b) > 1/2 : b tends to be larger than a
## #-----#
##
## #---Data Info-----#
## Sample Size
## 1      1      14
```

```
## 2      2    14
##
## #----Analysis-----#
##      Lower p.hat Upper      T p.value
## BM    0.038 0.145 0.253 -7.142    0.000
## PERM 0.033 0.145 0.258 -7.142    0.001
##
## #-----#
```

```
sessionInfo()
```

## Análise feita por Frederico Pedrosa

```
## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
##
## 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] effectsize_1.0.0 rstatix_0.7.2  nparcomp_3.0  multcomp_1.4-26
## [5] TH.data_1.1-3    survival_3.7-0 mvtnorm_1.3-3  nparLD_2.2
## [9] MASS_7.3-61      lubridate_1.9.4 forcats_1.0.0  stringr_1.5.1
## [13] purrr_1.0.2      readr_2.1.5    tibble_3.2.1   ggplot2_3.5.2
## [17] tidyverse_2.0.0  tidyr_1.3.1    dplyr_1.1.4    knitr_1.50
## [21] gt_0.11.1        kableExtra_1.4.0 psych_2.5.3    readxl_1.4.3
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.6      bayestestR_0.15.1 xfun_0.52        insight_1.1.0
## [5] lattice_0.22-6    tzdb_0.5.0        vctr_0.6.5       tools_4.4.2
## [9] generics_0.1.3    datawizard_1.0.0  parallel_4.4.2   sandwich_3.1-1
## [13] pkgconfig_2.0.3   Matrix_1.7-1      lifecycle_1.0.4  compiler_4.4.2
## [17] munsell_0.5.1     mnormt_2.1.1      codetools_0.2-20 carData_3.0-5
## [21] htmltools_0.5.8.1 yaml_2.3.10       Formula_1.2-5    car_3.1-3
## [25] pillar_1.10.2     abind_1.4-8       nlme_3.1-166     tidyselect_1.2.1
## [29] digest_0.6.37     stringi_1.8.7     splines_4.4.2    fastmap_1.2.0
## [33] grid_4.4.2        colorspace_2.1-1  cli_3.6.4        magrittr_2.0.3
## [37] utf8_1.2.4        broom_1.0.8       withr_3.0.2      scales_1.3.0
## [41] backports_1.5.0   estimability_1.5.1 timechange_0.3.0  rmarkdown_2.29
```

```
## [45] emmeans_1.10.6      cellranger_1.1.0    zoo_1.8-14          hms_1.1.3
## [49] coda_0.19-4.1        evaluate_1.0.3      parameters_0.24.1    viridisLite_0.4.2
## [53] rlang_1.1.5          xtable_1.8-4        glue_1.8.0           formatR_1.14
## [57] xml2_1.3.8           svglite_2.1.3       rstudioapi_0.17.1    R6_2.6.1
## [61] systemfonts_1.2.2
```