# Técnicas e Métodos de Pesquisa no R

Luiz G. de Almeida

2024-03-12

# Table of contents

# Introdução

Tutorial produzido com base nas aulas práticas da disciplina "Estatística Aplicada a Psicobiologia II - 2023", ministrada pelo Professor Altay Lino de Souza e oferecida pelo Departamento de Psicobiologia da UNIFESP.

## Sobre as aulas

As aulas são gravadas e disponibilizadas gratuitamente por meio de lives no canal Científtica & Podcast Naruhodo do YouTube. Destacando aqui o agradecimento mais do que especial para a Maria Lucia Oliveira De Souza Formigoni, por tornar possível a disciplina.

## Sobre o tutorial

Este tutorial tem como objetivo oferecer uma introdução prática à análise estatística de dados no R, utilizando diversos bancos de dados para cada tipo de anáise. O público-alvo abrange estudantes de Estatística Aplicada a Psicobiologia II, pós-graduandos e pesquisadores que buscam aprimorar suas habilidades em análise de dados. É recomendado ter conhecimento básico em estatística, particularmente Estatística Aplicada a Psicobiologia I, e alguma familiaridade com o ambiente R para acompanhar este tutorial. Abordaremos as seguintes análises:

- Transformação de dados para análises
- Modelos lineares:
  - Modelo linear geral (GLM ) de medidas repetidas

  - Generalized Estimated Equations (GEE)
  - Modelos mistos e hierárquicos (GMM)
  - Generalized linear models (GzLM)
- Análise de sobrevida
  - Kaplan-Meier
  - Regressão de Cox
  - Cox Tempo dependente
- Séries temporais (ARIMA)
- Modelagem de Equação Estrutural (SEM)
  - Path analysis
  - Confrmatory Factor Analysis (CFA)
  - Moderação e mediação

Ao fim de cada capítulos, nas seções intituladas "Extras", vamos mostrar dicas sobre pacotes que podem ser úteis para suas análises, mas que não estão disponíveis no SPSS ou no Jamovi.

> **!** Importante!
>
> O material apresentado aqui é **complementar** às aulas teóricas e práticas. É imprescindível que você assista às aulas antes de resolver os exercícios no R.

## R e Rstudio

Embora as aulas práticas tenham sido gravadas utilizando o SPSS, o intuito do tutorial é replicar as análises no R, que é gratuito! Portanto você precisa baixar o R e o Rstudio.

Download do R
Download do Rstudio

## Aulas práticas gravadas

Os vídeos das aulas práticas no SPSS foram anexados ao fim de cada capítulo para que você possa ter uma referência do tipo de análise realizada. Em alguns casos você notarão que os resultados não serão idênticos no SPSS e no R. Isso ocorre devido aos diferentes algorítimos de estimação de coeficientes utilizados nos programas. O importante é você sempre reportar como a análise foi feita, quais programas e sempre que possível, disponibilizar o código ou o passo-a-passo utilizado para realizar a análise.

## Boas práticas no Rstudio

Criar um projeto separado para cada tipo de análise no R é uma prática recomendada porque mantém o ambiente organizado, evita conflitos entre projetos, facilita a colaboração e torna a reprodução e compartilhamento de trabalho mais eficientes.

### Criando o projeto e alocando os arquivos

Para criar um novo projeto no R, siga estes passos simples:

1. Abra o RStudio.
2. Vá até a guia "File" (Arquivo) e selecione "New Project" (Novo Projeto).
3. Escolha um diretório para o seu projeto, onde todas as pastas e arquivos relacionados a ele serão armazenados. Isso ajudará na organização.
4. Clique em "Create Project" (Criar Projeto).

Feito isso você terá um novo projeto configurado. Qualquer arquivo que você deseje usar para o tutorial deve ser colocado dentro da pasta desse projeto. Isso garantirá que todos os caminhos e referências aos arquivos sejam relativos ao diretório do projeto, facilitando a portabilidade e compartilhamento do tutorial.

Com esses passos, você terá um ambiente de projeto limpo e organizado para trabalhar com seus arquivos e conduzir seu tutorial no R.

## Instalando e carregando os pacotes

No início de cada capítulo, você encontrará uma lista completa dos pacotes necessários para reproduzir as análises correspondentes.

Para instalar um pacote, basta executar o comando `install.packages("nome_do_pacote")` uma única vez.

Por exemplo: `install.packages("effects")`. Este comando instalará o pacote "effects", que contém funções para calcular os estimadores de modelos lineares. É importante colocar o nome do pacote entre aspas (" ")

Após a instalação do pacote, será necessário carregá-lo sempre que desejar utilizar alguma função associada a ele. Para isso basta executar o comando library(nome_do_pacote). Note que aqui não há a necessidade de colocar o nome do pacote entre aspas.

Exemplo: library(effects).

Pronto! Agora você está familiarizado com o processo de instalação e carregamento dos pacotes que serão utilizados ao longo deste tutorial. Pode-se fazer uma analogia com uma biblioteca: adquirir os livros seria como instalar os pacotes (`install.packages`), e retirar um livro da prateleira seria como carregar o pacote (`library`) quando necessário.

## Referências

https://r4ds.hadley.nz/

# Part I

# GLM, GEE, GMM, GzLM

## Resumo sobre os modelos lineares abordados

- **Modelo Linear Geral (GLM) de Medidas Repetidas:** O Modelo Linear Geral de Medidas Repetidas é uma extensão do modelo linear geral tradicional, projetado para lidar com dados repetidos ao longo do tempo. Ele é utilizado quando há correlação entre as observações, como em estudos longitudinais, e permite modelar a estrutura de covariância entre as medições repetidas.
  - **Desenho de Estudo Sugerido:**
    - Um estudo longitudinal com medições repetidas ao longo do tempo em um grupo de participantes.
  - **Exemplo:**
    - Acompanhamento de pacientes com uma condição médica específica, medindo regularmente os níveis de uma variável biológica para observar mudanças ao longo do tratamento.
- **Generalized Estimated Equations (GEE):** As Equações Estimadas Generalizadas (GEE) são uma abordagem estatística para análise de dados longitudinais ou correlacionados. Elas proporcionam uma estrutura robusta para lidar com a dependência entre as observações, permitindo estimativas eficientes dos parâmetros, mesmo quando a especificação da covariância não é precisa.
  - **Desenho de Estudo Sugerido:**
    - Um estudo observacional ou ensaio clínico longitudinal onde as medições podem ser correlacionadas, como em estudos epidemiológicos.
  - **Exemplo:**
    - Investigação sobre a eficácia de um programa de intervenção de saúde em que as observações estão correlacionadas dentro dos grupos de participantes.

- **Modelos Mistos e Hierárquicos (GMM):** Os Modelos Mistos e Hierárquicos, também conhecidos como Modelos de Efeitos Misto, combinam componentes fixos e aleatórios para modelar tanto a variabilidade fixa quanto a aleatória nos dados. Esses modelos são particularmente úteis quando há hierarquia nos dados, como em estudos multicêntricos, onde as observações podem ser agrupadas em diferentes níveis (por exemplo, centros de pesquisa). Eles permitem capturar a variabilidade tanto dentro quanto entre os grupos, oferecendo uma abordagem flexível para análise de dados complexos.
  - **Desenho de Estudo Sugerido:**
    * Estudos multicêntricos ou experimentos com estrutura hierárquica, onde as unidades de observação estão agrupadas em diferentes níveis.
  - **Exemplo:**
    * Avaliação do desempenho acadêmico de alunos em escolas, onde os alunos (nível inferior) estão agrupados em salas de aula (níveis superiores), considerando o efeito tanto do ensino individual quanto do ambiente escolar.

## Principais vantagens dos modelos lineres de medidas repetidas em comparação com a ANOVA

1. **Modelagem flexível:** O GLM permite modelar e analisar experimentos com medidas repetidas de forma mais flexível. Você pode incluir múltiplos fatores independentes (variáveis independentes) em um único modelo e estudar suas interações, o que é especialmente útil em experimentos complexos.
2. **Tratamento de dados desequilibrados:** O GLM pode lidar eficazmente com desequilíbrio nas amostras ou tamanhos diferentes de grupos, o que é comum em

experimentos do mundo real. A ANOVA tradicional é mais sensível a desequilíbrio.

3. **Modelagem de covariáveis:** O GLM permite incorporar covariáveis (variáveis de controle) em sua análise para controlar o efeito de variáveis que não são o foco principal do estudo. Isso melhora a precisão das estimativas dos efeitos de interesse.

4. **Correções para violações de pressupostos:** Quando os pressupostos da ANOVA, como a homogeneidade de variâncias ou normalidade dos resíduos, são violados, o GLM oferece opções para corrigir ou lidar com essas violações, tornando os resultados mais robustos.

5. **Modelagem de medidas contínuas e categóricas:** O GLM pode acomodar variáveis dependentes contínuas e categóricas (nominais ou ordinais), o que é útil em situações em que a variável dependente é de natureza diferente.

6. **Maior poder estatístico:** O GLM pode ser mais poderoso do que a ANOVA em situações em que as medidas repetidas têm alta correlação entre si, permitindo detectar diferenças significativas mesmo com tamanhos de amostra menores.

7. **Análise de interações complexas:** O GLM é especialmente eficaz na análise de interações complexas entre fatores independentes em designs experimentais com medidas repetidas, o que é difícil de realizar com a ANOVA.

## Pacotes que vamos utilizar

```r
library(emmeans)   # Cálculo de médias estimadas após análises estatísticas.
library(lme4)      # Ajuste de modelos lineares mistos.
library(nlme)      # Ajuste de modelos mistos não lineares.
library(flexplot)  # Criação de gráficos flexíveis e personalizados.
library(foreign)   # Importação/exportação de dados de outros formatos.
library(tidyr)     # Manipulação de dados.
library(dplyr)     # Manipulação e transformação de dados de maneira eficiente.
```

```r
library(multcomp)  # Correção de múltiplas comparações pós-teste.
library(effects)   # Visualização de efeitos de modelos estatísticos.
library(sjstats)   # Estatísticas descritivas e sumarização de modelos.
#library(tm)        # Análise de texto e mineração de texto.
#library(car)       # Análise de regressão e diagnóstico de regressão.
#library(pwr)       # Cálculo do poder estatístico em estudos de amostragem.
library(rstatix)   # Análise estatística simplificada.
library(geepack)   # Ajuste de modelos de equações de estimação generalizadas.
#library(htmltools) # Ferramentas para trabalhar com HTML.
#library(mime)      # Ferramentas para manipulação de tipos MIME.
library(performance) # Avaliação e melhoria do desempenho do modelo linear
library(see)       # Simplificar a exploração de dados
library(rempsyc)   # Métodos psicométricos e estatísticas relacionadas à psicometria
library(easystats) # Simplifica a análise estatística
```

## Banco de dados, script e Lista 1

Faça o download do arquivo compactado abaixo.

Lembre-se de descompactar os três arquivos na **mesma pasta do projeto** que você acabou de criar!

O arquivo compactado contém:

1. `bd_New drug_respiratory&pulse.sav`: Este é o arquivo de banco de dados que será usado ao longo do tutorial. Ele contém os dados que serão analisados e explorados durante os exercícios.
2. `lista1_parcial.R`: Este arquivo .R contém o script parcialmente preenchido para praticar e estudar os códigos abordados no tutorial. Você pode usar este script como um guia interativo para aprender e executar as análises estatísticas.
3. A `"Lista de Exercícios 1"` contém os exercícios para serem resolvidos.

## Carregando os Dados

Vamos começar carregando o conjunto de dados original e realizando algumas transformações para tornar possível nossas análises.

```
original_wide = read.spss("bd_New drug_respiratory&pulse.sav", to.data.frame=TRUE)
head(original_wide)
```

```
      drug resp1 resp2 resp3 pulse1 pulse2 pulse3
1 New Drug   3.4   3.3   3.3    2.2    2.1    2.1
2 New Drug   3.4   3.4   3.3    2.2    2.1    2.2
3 New Drug   3.3   3.4   3.4    2.3    2.4    2.3
4 New Drug   3.4   3.4   3.4    2.3    2.4    2.3
5 New Drug   3.3   3.4   3.3    2.2    2.2    2.4
6 New Drug   3.3   3.3   3.3    2.0    2.1    2.4
```

O código fornecido tem como objetivo carregar um conjunto de dados a partir de um arquivo SPSS chamado "bd_New drug_respiratory&pulse.sav" e exibir as primeiras linhas desse conjunto de dados.

1. `original_wide = read.spss("bd_New drug_respiratory&pulse.sav", to.data.frame=TRUE)`: Esta linha de código utiliza a função `read.spss` para ler o arquivo SPSS "bd_New drug_respiratory&pulse.sav" e convertê-lo em um objeto de data frame do R. A opção `to.data.frame=TRUE` especifica que queremos que os dados sejam armazenados em um data frame.
2. `head(original_wide)`: Após a leitura do conjunto de dados, esta linha de código utiliza a função `head` para mostrar as primeiras linhas do data frame `original_wide`. Isso ajuda a visualizar rapidamente os dados e verificar sua estrutura.

## Transformando o Banco de Dados de Wide para Long

Para tornar possível algumas análises, precisamos transformar o banco de dados de formato "wide" para "long". Isso nos permitirá realizar análises de medidas repetidas.

```
bd <- original_wide %>%
  rename_with(~gsub("(resp|pulse)(\\d+)", "\\1_\\2", .), -drug) %>%
  mutate(ID = row_number()) %>%
  dplyr::select(ID, everything())


head(bd)
```

```
  ID     drug resp_1 resp_2 resp_3 pulse_1 pulse_2 pulse_3
1  1 New Drug    3.4    3.3    3.3     2.2     2.1     2.1
2  2 New Drug    3.4    3.4    3.3     2.2     2.1     2.2
3  3 New Drug    3.3    3.4    3.4     2.3     2.4     2.3
4  4 New Drug    3.4    3.4    3.4     2.3     2.4     2.3
5  5 New Drug    3.3    3.4    3.3     2.2     2.2     2.4
6  6 New Drug    3.3    3.3    3.3     2.0     2.1     2.4
```

Os códigos fornecidos têm como objetivo renomear colunas e transformar um conjunto de dados de formato "wide" para "long", onde uma expressão regular está sendo utilizada para facilitar esse processo.

Uma expressão regular, ou regex, é uma sequência de caracteres que define um padrão de busca em texto, permitindo operações avançadas de busca e manipulação. Elas são amplamente usadas na programação para validação, extração e transformação de dados em texto.

No primeiro trecho de código, estamos renomeando as colunas do conjunto de dados `original_wide`. Aqui, a expressão regular `(resp|pulse)(\\d+)` está sendo usada na função `gsub`. Vamos explicar essa expressão regular:

- `(resp|pulse)`: Isso corresponde à palavra "resp" OU "pulse". O operador `|` atua como uma escolha, permitindo que corresponda a uma das duas palavras.
- `(\\d+)`: Isso corresponde a um ou mais dígitos numéricos. O `\\d+` é usado para extrair os números que seguem "resp" ou "pulse".

A expressão regular `(resp|pulse)(\\d+)` funciona para identificar colunas com nomes como "resp1", "resp2", "pulse1", "pulse2" etc. A função `gsub` substitui esses nomes de colunas por um novo formato, onde mantém "resp" ou "pulse" e adiciona o número correspondente. Por exemplo, "resp1" será renomeado para "resp_1", "pulse2" será renomeado para "pulse_2" e assim por diante.

Isso é útil para o próximo passo porque torna mais fácil identificar e separar os dados de resp e pulse em diferentes colunas. Além disso, os números extraídos da expressão regular serão usados para criar a variável "Tempo", que indicará as medidas repetidas ao longo do tempo.

```
bd_long = pivot_longer(bd,
                       cols = resp_1:pulse_3,
                       names_to = c(".value", "Tempo"),
                       names_pattern = "(.+)_(.+)")

head(bd_long)
```

```
# A tibble: 6 x 5
     ID drug     Tempo  resp pulse
  <int> <fct>    <chr> <dbl> <dbl>
1     1 New Drug 1       3.4   2.2
2     1 New Drug 2       3.3   2.1
3     1 New Drug 3       3.3   2.1
4     2 New Drug 1       3.4   2.2
5     2 New Drug 2       3.4   2.1
6     2 New Drug 3       3.3   2.2
```

No segundo trecho de código, estamos usando a função `pivot_longer` para transformar o conjunto de dados `bd` de

formato “wide” para “long”. A opção `names_pattern` usa a expressão regular `(.)_(.)` para dividir os nomes das colunas em duas partes:

- `(.+)`: Isso corresponde a qualquer sequência de caracteres, representando os nomes originais das colunas.
- `_(.+)`: Isso corresponde ao caractere sublinhado “_” seguido de qualquer sequência de caracteres. Essa parte será usada para identificar os valores correspondentes nas colunas no formato “long”.

Portanto, a expressão regular `(.)_(.)` ajuda a extrair informações dos nomes das colunas originais e organizá-las adequadamente no formato “long” do conjunto de dados `bd_long`, onde a primeira parte é armazenada na coluna `"Tempo"` e a segunda parte é usada para identificar os valores correspondentes na coluna `"valor"` (geralmente representada como `.value` no R).

## Alterando o tipo da variável “Tempo”

```
# Suponha que sua variável "Tempo" esteja em um dataframe chamado "seu_data_frame"
bd_long$Tempo <- factor(bd_long$Tempo)
```

O código assume que a variável `"Tempo"` está no dataframe chamado `"bd_long"`. Ele usa a função `factor()` para converter a variável “Tempo” em uma variável categórica. A conversão para uma variável categórica é útil quando você deseja tratar “Tempo” como uma variável de fator com níveis distintos em vez de uma variável numérica contínua. Essa transformação pode ser útil em análises estatísticas que envolvam categorias ou grupos de tempo, como em modelos de medidas repetidas.

## Pressupostos da variável dependente

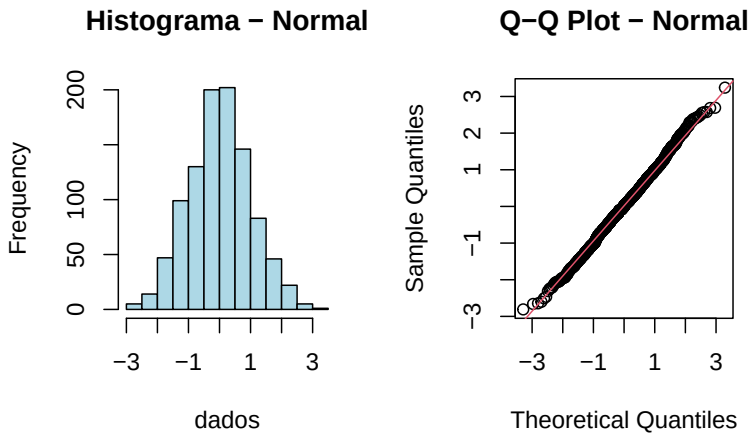
A distribuição dos dados da variável independente é fundamental para inferências estatísticas robustas. Quando os dados seguem uma distribuição normal, isso implica que a maioria

das observações está centralizada em torno da média, proporcionando uma simetria e previsibilidade desejáveis. Esta normalidade é frequentemente pressuposta em muitos métodos estatísticos clássicos. Ao observar o histograma da variável independente, esperamos ver uma forma de sino simétrica. No Q-Q plot, quando os dados são normalmente distribuídos, os pontos devem seguir aproximadamente uma linha diagonal. Em contrapartida, quando os dados não são normalmente distribuídos, o histograma pode revelar assimetria ou padrões diferentes, e o Q-Q plot apresentará desvios significativos da linha diagonal, indicando divergências da normalidade. Analisar a normalidade dos dados e interpretar o Q-Q plot ajuda a guiar a escolha adequada de métodos estatísticos e a compreender possíveis limitações na inferência.

Abaixo exemplos de distribuições normais, não normais e seus respectivos gráicos de disperção e Q-Q plot.

### Dados com distribuição normal

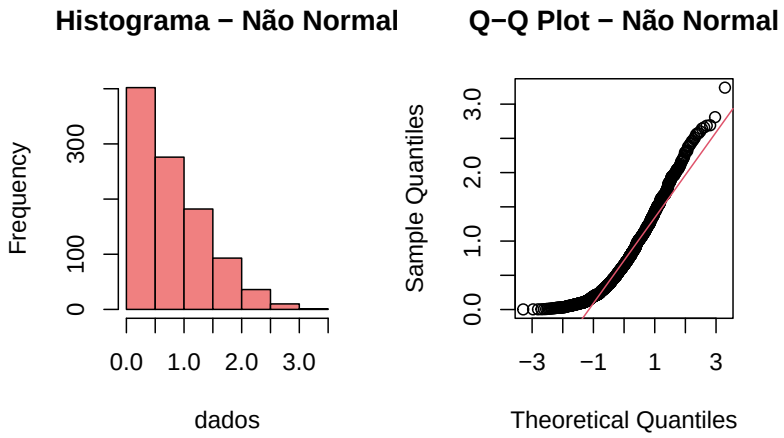
```r
# Bloco para dados normalmente distribuídos
set.seed(123)
dados_normais <- rnorm(1000, mean = 0, sd = 1)

# Função para criar histograma e Q-Q plot
criar_graficos_normais <- function(dados, titulo) {
  par(mfrow = c(1, 2))  # Organiza os gráficos em uma linha com duas colunas

  # Histograma
  hist(dados, main = paste("Histograma -", titulo), col = "lightblue", border = "black")

  # Q-Q plot
  qqnorm(dados, main = paste("Q-Q Plot -", titulo))
  qqline(dados, col = 2)
}

# Crie os gráficos para dados normalmente distribuídos
criar_graficos_normais(dados_normais, "Normal")
```




```
# Restaure o layout gráfico padrão
par(mfrow = c(1, 1))
```

### Dados com distribuição não-normal

```
# Bloco para dados não normalmente distribuídos
set.seed(123)
dados_nao_normais <- abs(rnorm(1000, mean = 0, sd = 1))

# Função para criar histograma e Q-Q plot
criar_graficos_nao_normais <- function(dados, titulo) {
  par(mfrow = c(1, 2))  # Organiza os gráficos em uma linha com duas colunas

  # Histograma
  hist(dados, main = paste("Histograma -", titulo), col = "lightcoral", border = "black")

  # Q-Q plot
  qqnorm(dados, main = paste("Q-Q Plot -", titulo))
  qqline(dados, col = 2)
}

# Crie os gráficos para dados não normalmente distribuídos
criar_graficos_nao_normais(dados_nao_normais, "Não Normal")
```




```r
# Restaure o layout gráfico padrão
par(mfrow = c(1, 1))
```

Além das análises visuais dos gráficos, temos também o teste de Shapiro-Wilk. A hipótese nula no teste de Shapiro-Wilk é que a variável analisada segue uma distribuição normal. Em termos mais formais, a hipótese nula (H0) é:

H0:Os dados são provenientes de uma distribuição normal.

> ⚠ Cuidado!
>
> Se o p-valor for **maior que 0,05**, não há evidências suficientes para rejeitar a hipótese nula, **indicando que os dados podem ser considerados normalmente distribuídos**. Por outro lado, um p-valor **menor que 0,05** sugere que há evidências significativas contra a hipótese nula, **indicando não normalidade nos dados**. É importante considerar o contexto do estudo ao interpretar os resultados e ter em mente que o teste pode ser sensível a tamanhos amostrais muito grandes, resultando em rejeições mesmo para desvios pequenos da normalidade.

Podemos verificar o teste de Shapiro-Wilk para os dois exemplos anteriores e treinar a leitura dos resultados

### Shapiro-Wilk para os dados com distribuição normal

```
shapiro.test(dados_normais)
```

```
    Shapiro-Wilk normality test

data:  dados_normais
W = 0.99838, p-value = 0.4765
```

O valor de p é maior do que 0,05, portanto podemos assumir que os dados possuem distribuição normal.

### Shapiro-Wilk para os dados com distribuição não-normal

```
shapiro.test(dados_nao_normais)
```

```
    Shapiro-Wilk normality test

data:  dados_nao_normais
W = 0.92344, p-value < 2.2e-16
```

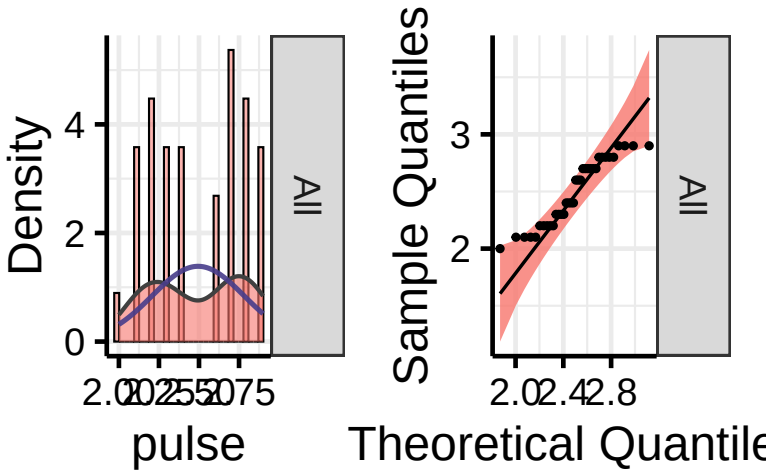
O valor de p é menor do que 0,05, portanto podemos assumir que os dados possuem distribuição não-normal.

Vamos agora verificar a distribuição, o Q-Q plot e o teste de Shapiro-Wilk das variáveis Resp e Pulse do nosso banco de dados.

### Densidade (distribuição) + Q-Q plot da variável "Pulse"

Para verificar a distribuição e o Q-Q plot vamos utilizar a função `nice_normality()` do pacote `rempsyc`, que cria os dois gráficos em poucas linhas de código!

```
nice_normality(data = bd_long,
               variable = "pulse",
               histogram = TRUE)
```




Shapiro-Wilk para a variável pulse:

```
shapiro.test(bd_long$pulse)
```

```
    Shapiro-Wilk normality test

data:  bd_long$pulse
W = 0.90791, p-value = 0.005655
```

Tanto pela análise gráfica quanto pelo teste de Shapiro-Wilk podemos observar que a variável "Pulse" não pussui distribuião normal.

### Esfericidade da Variável "Pulse"

Em casos de medidas repetidas também precisamos avaliar a esfericidade intra-sujeito ao longo do tempo. A esfericidade intra-sujeitos avalia se as variações nas diferenças entre as medidas

ao longo do tempo são consistentes para todas as combinações de momentos. Se essa homogeneidade não for atendida, ajustes como a correção de Greenhouse-Geisser ou Huynh-Feldt podem ser necessários para garantir conclusões estatisticamente válidas. Essas correções ajustam os graus de liberdade dos testes para lidar com a falta de esfericidade intra-sujeitos.

Vamos verificar a esfericidade da variável "pulse" usando o teste de Mauchly.

```
pulse_mauchly = anova_test(data = bd_long,
                           dv = pulse,
                           wid = ID,
                           within = Tempo)
```

Os códigos fornecidos têm como objetivo realizar um teste de Mauchly para verificar a esfericidade da variável "pulse" em um conjunto de dados no formato longo (`bd_long`). Vamos explicar o que cada linha de código faz:

1. `pulse_mauchly = anova_test(data = bd_long, dv = pulse, wid = ID, within = Tempo)`: Esta linha de código executa o teste de Mauchly para verificar a esfericidade da variável "pulse". A função `anova_test` é usada para realizar esse teste. Os argumentos passados para a função são:

    - `data`: O conjunto de dados no formato longo (`bd_long`), onde os dados estão organizados em formato apropriado para análises de medidas repetidas.
    - `dv`: A variável dependente sendo analisada, que neste caso é "pulse".
    - `wid`: A variável de identificação única (`ID`), que indica quais observações pertencem ao mesmo sujeito.
    - `within`: A variável categórica que representa o fator dentro dos sujeitos, neste caso, "Tempo".

2. `pulse_mauchly`: Esta linha de código armazena os resultados do teste de Mauchly na variável `pulse_mauchly`.

Os resultados incluem estatísticas relacionadas à esfericidade e os valores associados (valor-p).

Vamos interpretar os resultados da análise de esfericidade usando o Teste de Mauchly:

```
pulse_mauchly
```

```
ANOVA Table (type III tests)

$ANOVA
  Effect DFn DFd    F     p p<.05   ges
1  Tempo   2  22 3.48 0.049     * 0.024

$`Mauchly's Test for Sphericity`
  Effect     W    p p<.05
1  Tempo 0.781 0.29

$`Sphericity Corrections`
  Effect  GGe      DF[GG] p[GG] p[GG]<.05   HFe      DF[HF] p[HF] p[HF]<.05
1  Tempo 0.82 1.64, 18.04 0.061           0.945 1.89, 20.78 0.052
```

**ANOVA Table (type III tests): Effect (Efeito):** "Tempo". DFn e DFd: Graus de liberdade para o numerador (DFn) e denominador (DFd) da estatística F. F e p: Estatística F e valor p associado ao efeito "Tempo". p<.05: Indica se o valor p é menor que 0,05, sugerindo significância estatística. ges (generalized eta-squared): Uma medida da força do efeito.

- **Interpretação**: O efeito "Tempo" apresentou uma estatística F de 3,48 com um valor p de 0,049, indicando uma possível significância estatística. O valor p é menor que 0,05, sugerindo que há diferenças significativas entre os níveis de tempo.

**Mauchly's Test for Sphericity: Effect (Efeito):** "Tempo". W (Estátistica de Mauchly): 0,781. p: Valor p associado ao teste de Mauchly.

- **Interpretação**: O teste de Mauchly avalia a esfericidade. Para o efeito "Tempo", o valor p é 0,29, indicando que não há evidência estatística para rejeitar a esfericidade. Ou seja, a esfericidade não é violada.

**Sphericity Corrections: Effect (Efeito):** "Tempo". GGe (Greenhouse-Geisser epsilon): 0,82. DF[GG] e p[GG]: Graus de liberdade e valor p corrigidos pelo método Greenhouse-Geisser. HFe (Huynh-Feldt epsilon): 0,945. DF[HF] e p[HF]: Graus de liberdade e valor p corrigidos pelo método Huynh-Feldt.

- Interpretação: Se a esfericidade fosse violada, você usaria essas correções para ajustar os graus de liberdade e valores p. Os valores de GGe e HFe estão próximos de 1, indicando que a esfericidade não foi severamente violada. Os valores p corrigidos são 0,061 (GGe) e 0,052 (HFe), indicando que mesmo com a correção, o efeito "Tempo" pode ainda ser significativo.

## Esfericidade da Variável "Resp"

Da mesma forma, vamos verificar a esfericidade da variável "resp" usando o teste de Mauchly.

```
resp_mauchly = anova_test(data = bd_long,
                          dv = resp,
                          wid = ID,
                          within = Tempo)

resp_mauchly
```

```
ANOVA Table (type III tests)

$ANOVA
  Effect DFn DFd     F     p p<.05  ges
1  Tempo   2  22 0.344 0.713       0.01

$`Mauchly's Test for Sphericity`
  Effect     W     p p<.05
1  Tempo 0.501 0.032     *
```

```
$`Sphericity Corrections`
  Effect   GGe      DF[GG] p[GG] p[GG]<.05   HFe      DF[HF] p[HF] p[HF]<.05
1  Tempo 0.667 1.33, 14.68 0.629           0.725 1.45, 15.94 0.646
```

Vamos interpretar os resultados da segunda análise de esfericidade:

**ANOVA Table (type III tests): Effect (Efeito):** "Tempo". DFn e DFd: Graus de liberdade para o numerador (DFn) e denominador (DFd) da estatística F. F e p: Estatística F e valor p associado ao efeito "Tempo". p<.05: Indica se o valor p é menor que 0,05, sugerindo significância estatística. ges (generalized eta-squared): Uma medida da força do efeito.

- **Interpretação**: O efeito "Tempo" apresentou uma estatística F de 0,344 com um valor p de 0,713, indicando que não há evidência estatística para rejeitar a hipótese nula. O valor p é maior que 0,05, sugerindo que não há diferenças significativas entre os níveis de tempo.

**Mauchly's Test for Sphericity: Effect (Efeito):** "Tempo". W (Estátistica de Mauchly): 0,501. p e p<.05: Valor p associado ao teste de Mauchly e indicação de significância.

- **Interpretação**: O teste de Mauchly indica que a esfericidade foi violada, pois o valor p é menor que 0,05. Isso sugere que as covariâncias das diferenças entre os níveis de tempo não são iguais.

**Sphericity Corrections: Effect (Efeito): "Tempo".** GGe (Greenhouse-Geisser epsilon): 0,667. DF[GG] e p[GG]: Graus de liberdade e valor p corrigidos pelo método Greenhouse-Geisser. HFe (Huynh-Feldt epsilon): 0,725. DF[HF] e p[HF]: Graus de liberdade e valor p corrigidos pelo método Huynh-Feldt.

- **Interpretação**: As correções (GGe e HFe) sugerem que, mesmo com a correção para a violação da esfericidade, o efeito "Tempo" não é significativo. Os valores p corrigidos são 0,629 (GGe) e 0,646 (HFe), indicando que a falta de esfericidade afeta a significância do efeito "Tempo".

Os resultados indicam que a esfericidade foi violada, e mesmo com as correções, não há evidências significativas para o efeito "Tempo". Isso destaca a importância de considerar a esfericidade ao interpretar os resultados de análises de variância com medidas repetidas.

## GGe (Greenhouse-Geisser epsilon) e o HFe (Huynh-Feldt epsilon)

O GGe (Greenhouse-Geisser epsilon) e o HFe (Huynh-Feldt epsilon) são coeficientes de correção usados em análises de variância com medidas repetidas para lidar com a violação da esfericidade intra-sujeitos. Esses coeficientes ajustam os graus de liberdade dos testes estatísticos para compensar a falta de esfericidade, ajudando a evitar conclusões incorretas sobre a significância dos efeitos.

- **Greenhouse-Geisser epsilon (GGe):** Este coeficiente é uma estimativa da magnitude da não esfericidade intra-sujeitos. O GGe é usado para corrigir os graus de liberdade dos testes estatísticos, tornando-os mais conservadores quando a esfericidade é violada. Um GGe próximo de 1 indica menos violação da esfericidade.
- **Huynh-Feldt epsilon (HFe):** Similar ao GGe, o HFe é outro coeficiente de correção. Ele é um ajuste mais conservador que o GGe. Seu valor próximo de 1 indica menos violação da esfericidade. O HFe é geralmente mais utilizado quando os tamanhos amostrais são pequenos.

No contexto da variável Resp:

- **GGe:** 0,667 - Indica que, após a correção de Greenhouse-Geisser, os graus de liberdade foram reduzidos em cerca de 33% para compensar a violação da esfericidade intra-sujeitos.
- **HFe:** 0,725 - Similar ao GGe, o HFe fornece outra correção mais conservadora. Neste caso, os graus de liberdade são reduzidos em aproximadamente 27,5%.

Já no contexto da variável Pulse:

- GGe = 0,82 - Indica que, após a correção de Greenhouse-Geisser, os graus de liberdade foram reduzidos em cerca de 18% para compensar a violação da esfericidade intra-sujeitos.
- HFe = 0,945 - Indica que, após a correção de Huynh-Feldt, os graus de liberdade foram reduzidos em cerca de 5,5% para compensar a violação da esfericidade intra-sujeitos.

Agora finalmente vamos para a lista de exercícios, começando com o GLM e continuando na sequencia com o GEE e GMM.

# 1 Lista 1 - GLM, GEE e GMM

Baixe o banco **"new drug respiratory&pulse"**. O objetivo desta primeira aula prática será entender como os testes são realizados no R. A comparação entre os diferentes métodos será objeto de listas de exercícios subsequentes.

## 1.1 GLM

### 1.1.1 Análise para a Variável "resp"

Vamos ajustar o seguinte modelo de medidas repetidas para a variável dependente "resp":

$$resp = \beta_0 + \beta_1 drug + \beta_2 Tempo + \beta_3 drug * Tempo + \varepsilon$$

```
modelo1_resp = lm(resp ~ drug + Tempo + drug*Tempo, data = bd_long)

summary(modelo1_resp)
```

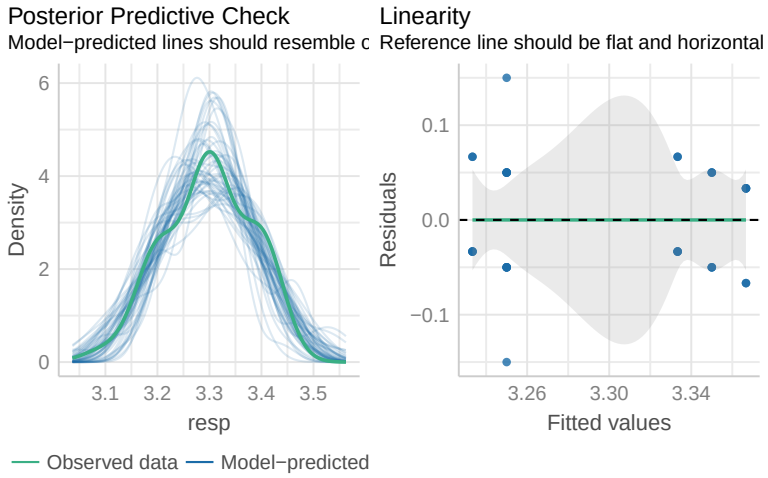
```
Call:
lm(formula = resp ~ drug + Tempo + drug * Tempo, data = bd_long)

Residuals:
     Min       1Q   Median       3Q      Max
-0.15000 -0.05000 -0.03333  0.05000  0.15000

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.350e+00  2.635e-02 127.124  < 2e-16 ***
```

```
drugPlacebo         -1.167e-01  3.727e-02  -3.130  0.00387 **
Tempo2               1.667e-02  3.727e-02   0.447  0.65793
Tempo3              -1.667e-02  3.727e-02  -0.447  0.65793
drugPlacebo:Tempo2   1.904e-15  5.270e-02   0.000  1.00000
drugPlacebo:Tempo3   3.333e-02  5.270e-02   0.632  0.53188
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06455 on 30 degrees of freedom
Multiple R-squared:  0.4559,    Adjusted R-squared:  0.3652
F-statistic: 5.027 on 5 and 30 DF,  p-value: 0.001836
```

Como houve diferença apenas entre os grupos que receberam a droga e o placebo, não vamos realizar o post hoc

### Pressupostos do modelo lm(resp)

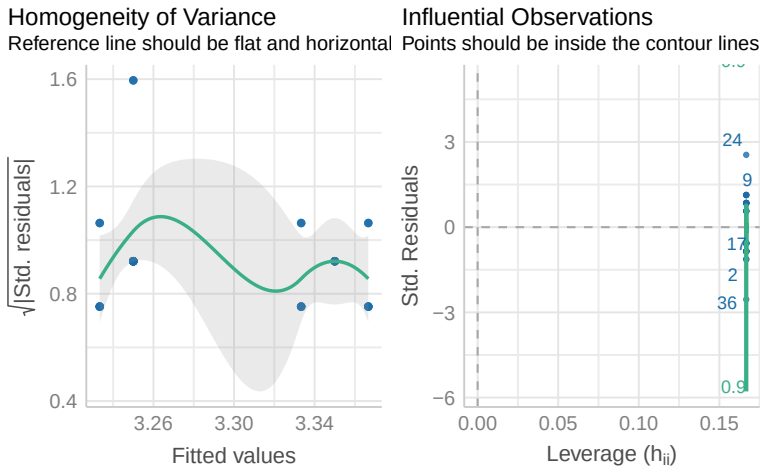
```
check_model(modelo1_resp, check = c("pp_check", "linearity"))
```




- **Posterior Predictive Checks:** utilizadas para identificar discrepâncias sistemáticas entre dados reais e simulados, auxiliando a avaliar se o tipo de modelo (família de distribuição) se ajusta adequadamente aos dados.

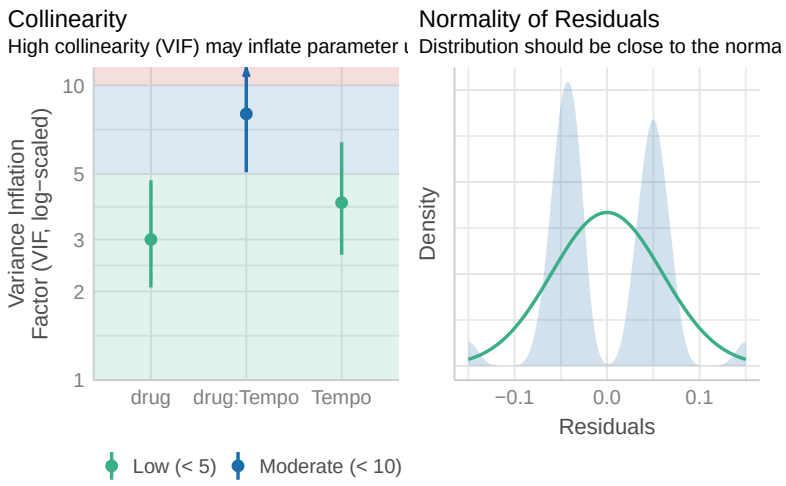
- **Linearity Assumption:** o gráfico de Linearidade verifica a suposição de relação linear. No entanto, a dispersão dos pontos também indica possíveis heterocedasticidades (ou seja, variância não constante, daí o termo "ncv" para este gráfico), mostrando se os resíduos têm padrões não lineares. Esse gráfico ajuda a observar se os preditores têm uma relação não linear com o resultado, indicada aproximadamente pela linha de referência. Uma linha reta e horizontal sugere que a especificação do modelo parece estar adequada. Contudo, se a linha for em forma de U, alguns preditores provavelmente devem ser modelados como termos quadráticos.

```
check_model(modelo1_resp, check = c("homogeneity", "outliers"))
```




- **Homogeneity of Variance:** verifica a suposição de variância igual (homocedasticidade). O padrão desejado é que os pontos se espalhem igualmente acima e abaixo de uma linha reta horizontal, sem desvios aparentes.
- **Influential Observations**: identifica observações influentes. Se algum ponto estiver fora da distância de Cook (linhas tracejadas), é considerado uma observação influente.

```
check_model(modelo1_resp, check = c("vif", "normality"))
```




**Multicollinearity:** verifica possíveis problemas de multicolinearidade entre os preditores. Em resumo, a multicolinearidade significa que, uma vez conhecido o efeito de um preditor, o valor de conhecer o outro preditor é relativamente baixo. Isso pode ocorrer quando uma terceira variável não observada tem um efeito causal em cada um dos dois preditores associados ao resultado. Nesses casos, a relação relevante seria entre a variável não observada e o resultado.

**Normality of Residuals:** determina se os resíduos do modelo de regressão têm uma distribuição normal. Geralmente, os pontos devem seguir a linha. Desvios (principalmente nas caudas) indicam que o modelo não prevê bem o resultado para a faixa que apresenta maiores desvios da linha. Para modelos lineares generalizados, é exibido um gráfico Q-Q meio-normal dos resíduos padronizados de desvio absoluto, mas a interpretação do gráfico permanece a mesma.

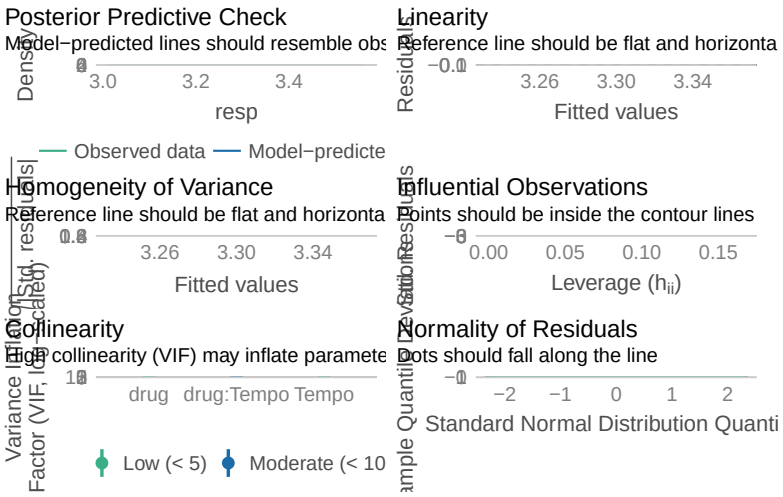
### Conclusão sobre os pressupostos

Fica nítido que o modelo violou diversos pressupostos. Ao fim do capítulo você encontrará uma lista de ações que podem ser

tomadas para cada uma das violações do modelo. Caso tudo mais falhe, apenas aceite que você tem um modelo ruim.

> **Tudo em um painel só!**
>
> Você pode utilizar a função `check_model()` sem especificar o parâmetro check. O resultado é um único painel com todas as análises. Nos exemplos anteriores os plots apareceram separados por questões didáticas e de formatação. Veja abaixo como fica o plot em painel.

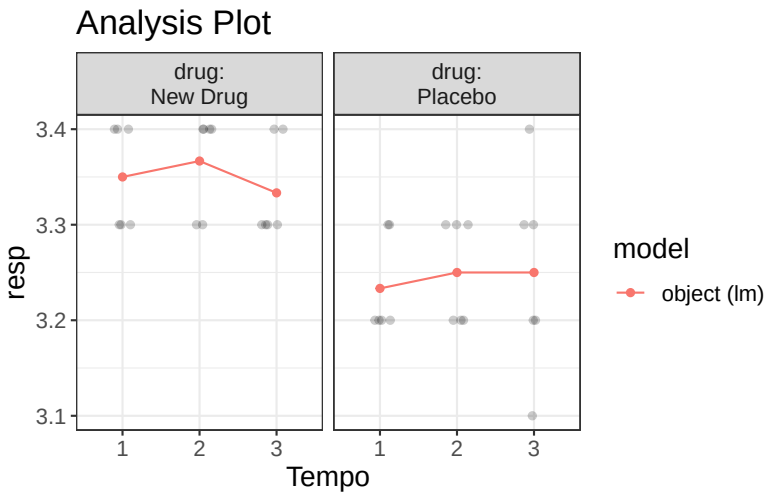
```
check_model(modelo1_resp)
```




### Plot do modelo lm(resp)

Agora vamos fazer um plot do modelo.

```
visualize(modelo1_resp, plot = "model")
```




O código `visualize(modelo1_resp, plot = "model")` usa a função `visualize` do pacote `flexplot` para criar um gráfico que visualiza o modelo estatístico denominado `modelo1_resp`. Este gráfico é uma representação visual do modelo, ajudando a compreender a relação entre as variáveis independentes e a variável dependente no contexto da análise estatística em questão.

### Escrevendo os Resultados com a função `report()`

Grandes poderes...

Podemos utilizar a função `report()` do pacote `easystat` que gerar um texto formatado para publicação e em inglês com os principais resultados de diversos modelos lineares! Com isso conseguimos diminuir erros de digitação, confusão com os estimadores e uma maior reprodutibilidade. Mas lembre-se! É **fundamental** você treinar como escrever os resultados. Use o poder do `report()` com sabedoria e sempre revise o texto gerado!

**Resultados**

```
report(modelo1_resp)
```

```
We fitted a linear model (estimated using OLS) to predict resp with drug and
Tempo (formula: resp ~ drug + Tempo + drug * Tempo). The model explains a
statistically significant and substantial proportion of variance (R2 = 0.46,
F(5, 30) = 5.03, p = 0.002, adj. R2 = 0.37). The model's intercept,
corresponding to drug = New Drug and Tempo = 1, is at 3.35 (95% CI [3.30,
3.40], t(30) = 127.12, p < .001). Within this model:

  - The effect of drug [Placebo] is statistically significant and negative (beta
= -0.12, 95% CI [-0.19, -0.04], t(30) = -3.13, p = 0.004; Std. beta = -1.44,
95% CI [-2.38, -0.50])
  - The effect of Tempo [2] is statistically non-significant and positive (beta =
0.02, 95% CI [-0.06, 0.09], t(30) = 0.45, p = 0.658; Std. beta = 0.21, 95% CI
[-0.73, 1.15])
  - The effect of Tempo [3] is statistically non-significant and negative (beta =
-0.02, 95% CI [-0.09, 0.06], t(30) = -0.45, p = 0.658; Std. beta = -0.21, 95%
CI [-1.15, 0.73])
  - The effect of drug [Placebo] × Tempo [2] is statistically non-significant and
positive (beta = 1.90e-15, 95% CI [-0.11, 0.11], t(30) = 3.61e-14, p > .999;
Std. beta = -1.65e-15, 95% CI [-1.33, 1.33])
  - The effect of drug [Placebo] × Tempo [3] is statistically non-significant and
positive (beta = 0.03, 95% CI [-0.07, 0.14], t(30) = 0.63, p = 0.532; Std. beta
= 0.41, 95% CI [-0.92, 1.74])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald t-distribution approximation.
```

### 1.1.2 Análise para a Variável “pulse”

Agora, ajustaremos o mesmo modelo para a variável dependente “pulse”:

$$pulse = \beta_0 + \beta_1 drug + \beta_2 Tempo + \beta_3 drug * Tempo + \varepsilon$$

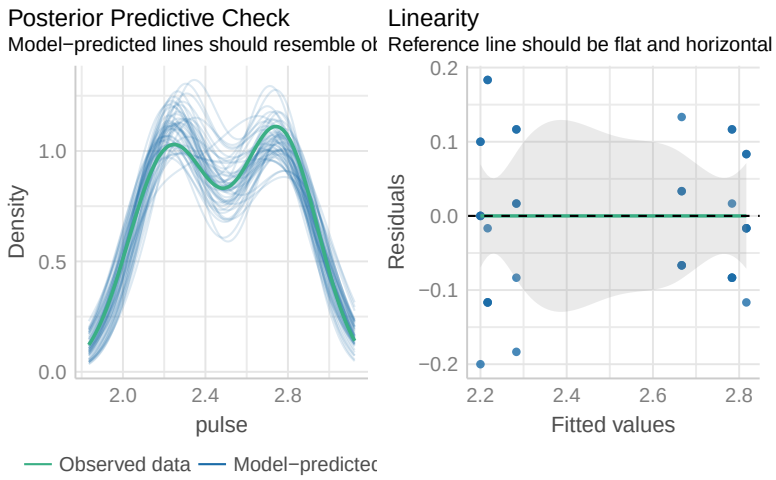
```r
# Ajustando o modelo
modelo1_pulse = glm(pulse ~ drug + Tempo + drug*Tempo, data = bd_long)

summary(modelo1_pulse)
```

```
Call:
glm(formula = pulse ~ drug + Tempo + drug * Tempo, data = bd_long)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.20000    0.04389  50.131  < 2e-16 ***
drugPlacebo        0.46667    0.06206   7.519 2.21e-08 ***
Tempo2             0.01667    0.06206   0.269    0.790
Tempo3             0.08333    0.06206   1.343    0.189
drugPlacebo:Tempo2 0.13333    0.08777   1.519    0.139
drugPlacebo:Tempo3 0.03333    0.08777   0.380    0.707
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.01155556)

    Null deviance: 2.89889  on 35  degrees of freedom
Residual deviance: 0.34667  on 30  degrees of freedom
AIC: -50.981

Number of Fisher Scoring iterations: 2
```

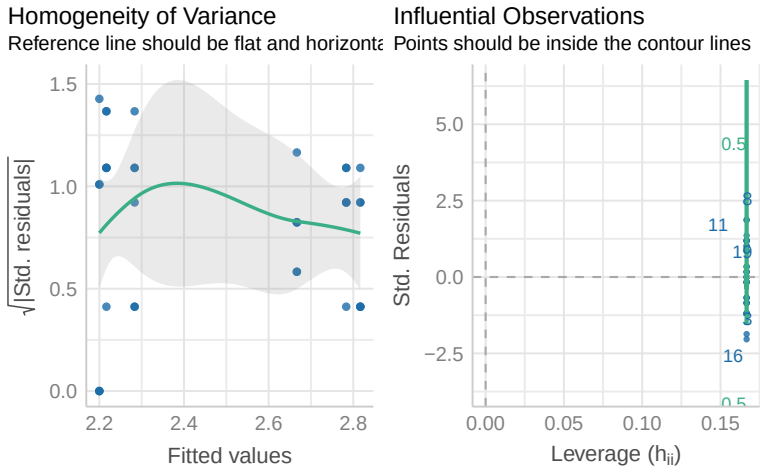
*Descrição dos resultados*

### Pressupostos do modelo

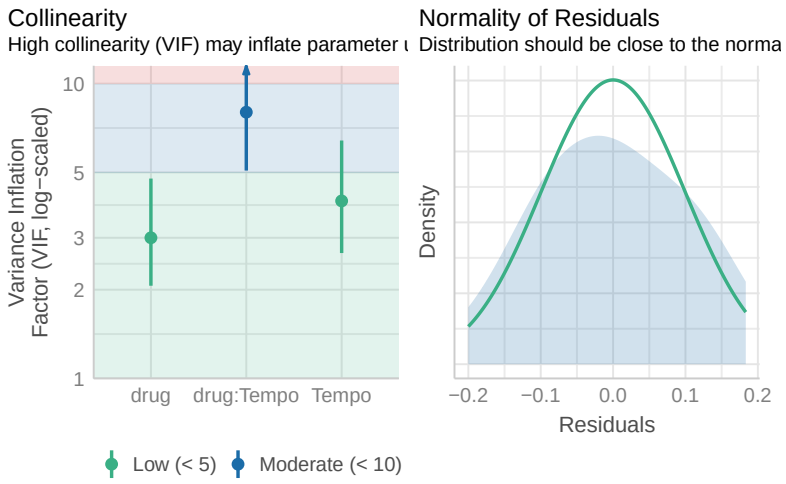
```r
check_model(modelo1_pulse, check = c("pp_check", "linearity"))
```




```
check_model(modelo1_pulse, check = c("homogeneity", "outliers"))
```




```
check_model(modelo1_pulse, check = c("vif", "normality"))
```

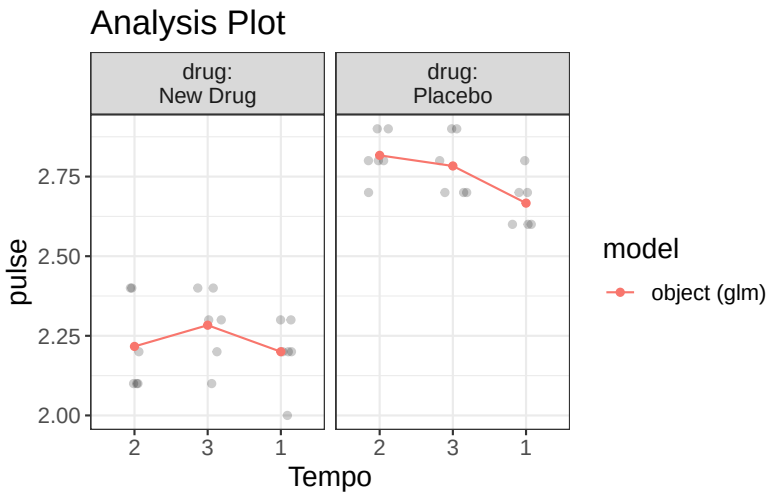



A interpretação dos resultados é a mesma para o modelo com a variável resp.

### Plot do modelo lm(pulse)

Agora vamos fazer um plot do modelo.

```
visualize(modelo1_pulse, plot = "model")
```



**Resultados**

```
report(modelo1_pulse)
```

```
We fitted a linear model (estimated using ML) to predict pulse with drug and
Tempo (formula: pulse ~ drug + Tempo + drug * Tempo). The model's explanatory
power is substantial (R2 = 0.88). The model's intercept, corresponding to drug
= New Drug and Tempo = 1, is at 2.20 (95% CI [2.11, 2.29], t(30) = 50.13, p <
.001). Within this model:

  - The effect of drug [Placebo] is statistically significant and positive (beta
= 0.47, 95% CI [0.35, 0.59], t(30) = 7.52, p < .001; Std. beta = 1.62, 95% CI
[1.20, 2.04])
  - The effect of Tempo [2] is statistically non-significant and positive (beta =
0.02, 95% CI [-0.10, 0.14], t(30) = 0.27, p = 0.788; Std. beta = 0.06, 95% CI
[-0.36, 0.48])
  - The effect of Tempo [3] is statistically non-significant and positive (beta =
0.08, 95% CI [-0.04, 0.20], t(30) = 1.34, p = 0.179; Std. beta = 0.29, 95% CI
[-0.13, 0.71])
  - The effect of drug [Placebo] × Tempo [2] is statistically non-significant and
positive (beta = 0.13, 95% CI [-0.04, 0.31], t(30) = 1.52, p = 0.129; Std. beta
= 0.46, 95% CI [-0.13, 1.06])
  - The effect of drug [Placebo] × Tempo [3] is statistically non-significant and
positive (beta = 0.03, 95% CI [-0.14, 0.21], t(30) = 0.38, p = 0.704; Std. beta
= 0.12, 95% CI [-0.48, 0.71])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald t-distribution approximation.
```

## 1.2 GEE

### 1.2.1 Análise para a Variável "Resp"

Realizaremos uma análise usando Generalized Estimating Equations (GEE) para a variável "resp".

```r
bd_long$ID = as.factor(bd_long$ID)

bd_long$Tempo = as.factor(bd_long$Tempo)
```

```r
# Ajustando o modelo GEE para "resp"
modelo_gee_resp <- geeglm(resp ~ drug + Tempo + drug*Tempo,
                          data = bd_long,
                          id = ID,
                          family = gaussian,
                          corstr = "unstructured")


summary(modelo_gee_resp)
```

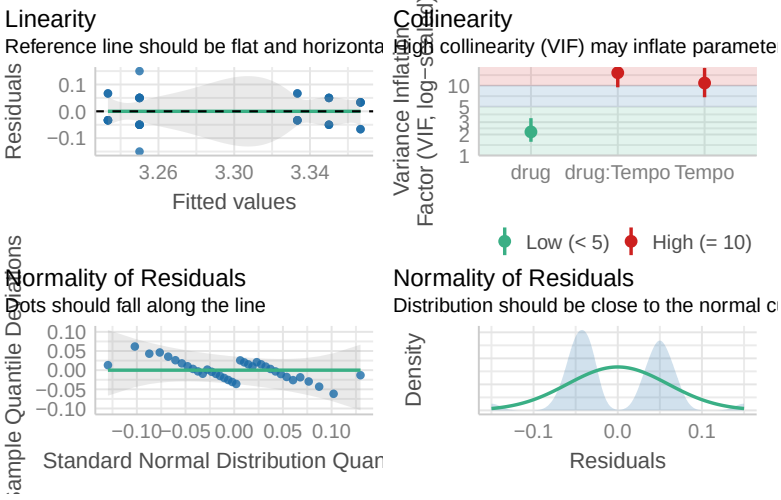
```
Call:
geeglm(formula = resp ~ drug + Tempo + drug * Tempo, family = gaussian,
    data = bd_long, id = ID, corstr = "unstructured")

 Coefficients:
                     Estimate    Std.err      Wald Pr(>|W|)
(Intercept)         3.350e+00  2.041e-02 26934.000  < 2e-16 ***
drugPlacebo        -1.167e-01  2.805e-02    17.294  3.2e-05 ***
Tempo2              1.667e-02  2.805e-02     0.353    0.552
Tempo3             -1.667e-02  2.805e-02     0.353    0.552
drugPlacebo:Tempo2  4.022e-18  3.967e-02     0.000    1.000
drugPlacebo:Tempo3  3.333e-02  5.693e-02     0.343    0.558
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = unstructured
Estimated Scale Parameters:

            Estimate   Std.err
(Intercept) 0.003472 0.0007618
  Link = identity

Estimated Correlation Parameters:
            Estimate Std.err
```

```
alpha.1:2 -9.252e-18 0.19596
alpha.1:3 -2.400e-01 0.27321
alpha.2:3  7.600e-01 0.09074
Number of clusters:   12  Maximum cluster size: 3
```

### Pressupostos do modelo GEE (resp)

Agora vamos fazer um plot do histograma dos resíduos do modelo

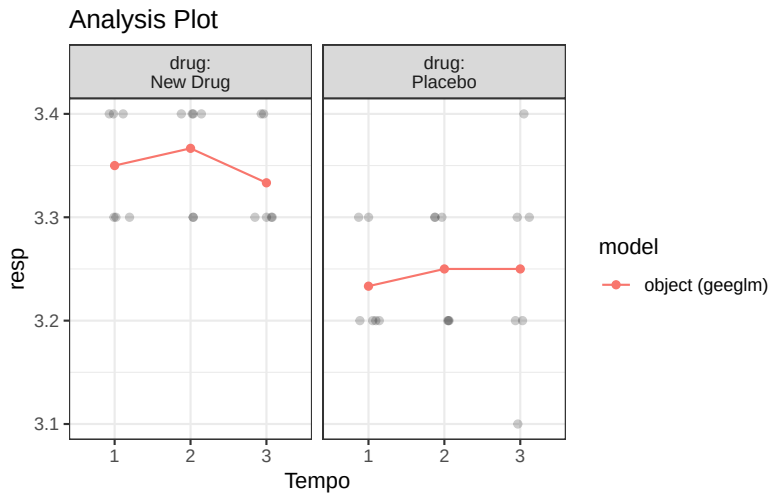
```
check_model(modelo_gee_resp)
```




### Plot do modelo GEE (resp)

Agora vamos fazer um plot do modelo.

```
visualize(modelo_gee_resp, plot = "model")
```




**Resultados do modelo**

> **Warning**
>
> A função `report()` não funciona para modelos GEE. Então treine para escrever os seus resultados!

### 1.2.2 Análise para a Variável "Pulse"

Agora, realizaremos uma análise GEE para a variável "pulse".

```
modelo_gee_pulse <- geeglm(pulse ~ drug + Tempo + drug*Tempo,
                           data = bd_long,
                           id = ID,
                           family = gaussian,
                           corstr = "unstructured")

summary(modelo_gee_pulse)
```

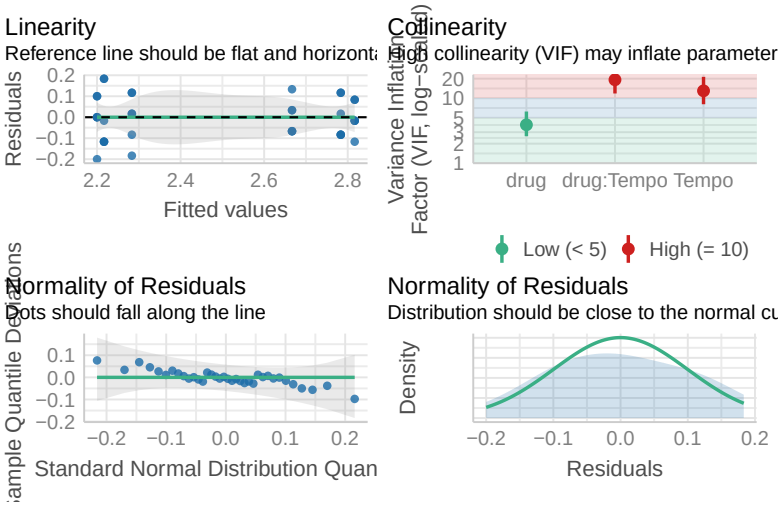
```
Call:
geeglm(formula = pulse ~ drug + Tempo + drug * Tempo, family = gaussian,
    data = bd_long, id = ID, corstr = "unstructured")
```

```
 Coefficients:
                  Estimate Std.err    Wald Pr(>|W|)
(Intercept)         2.2000  0.0408 2904.00   <2e-16 ***
drugPlacebo         0.4667  0.0509   84.00   <2e-16 ***
Tempo2              0.0167  0.0366    0.21   0.6492
Tempo3              0.0833  0.0684    1.49   0.2230
drugPlacebo:Tempo2  0.1333  0.0419   10.11   0.0015 **
drugPlacebo:Tempo3  0.0333  0.0877    0.14   0.7038
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = unstructured
Estimated Scale Parameters:

            Estimate Std.err
(Intercept)  0.00963 0.00168
  Link = identity

Estimated Correlation Parameters:
          Estimate Std.err
alpha.1:2    0.721   0.169
alpha.1:3   -0.288   0.208
alpha.2:3    0.115   0.267
Number of clusters:   12  Maximum cluster size: 3
```

### Pressupostos do modelo GEE (pulse)

Agora vamos fazer um plot do histograma dos resíduos do modelo

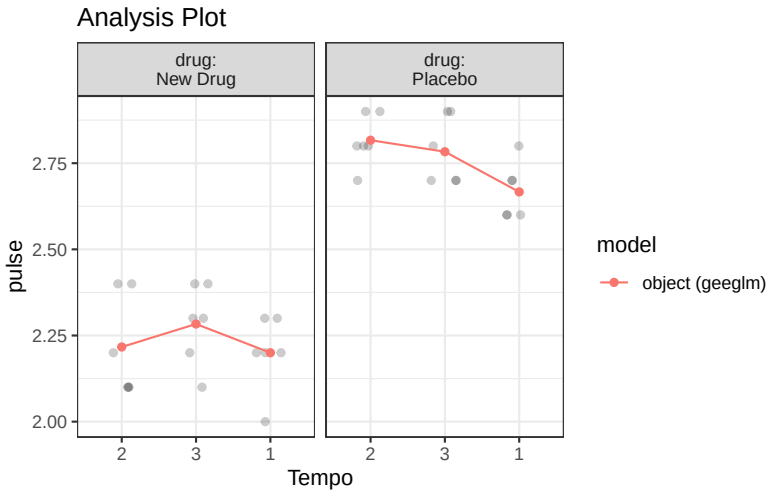
```
check_model(modelo_gee_pulse)
```




## Plot do modelo GEE (pulse)

Agora vamos fazer um plot do modelo.

```
visualize(modelo_gee_pulse, plot = "model")
```



#### Resultados do modelo

> **Warning**
>
> A função `report()` não funciona para modelos GEE. Então treine para escrever os seus resultados!

## 1.3 GMM

### 1.3.1 Análise para a Variável "resp"

Agora, realizaremos uma análise usando Generalized Mixed Models (GMM) para a variável "resp".

```
modelo_gmm_resp = lme(   fixed = resp ~ drug + Tempo + drug * Tempo,
                         random = ~1|ID,
                         data = bd_long )


summary(modelo_gmm_resp)
```

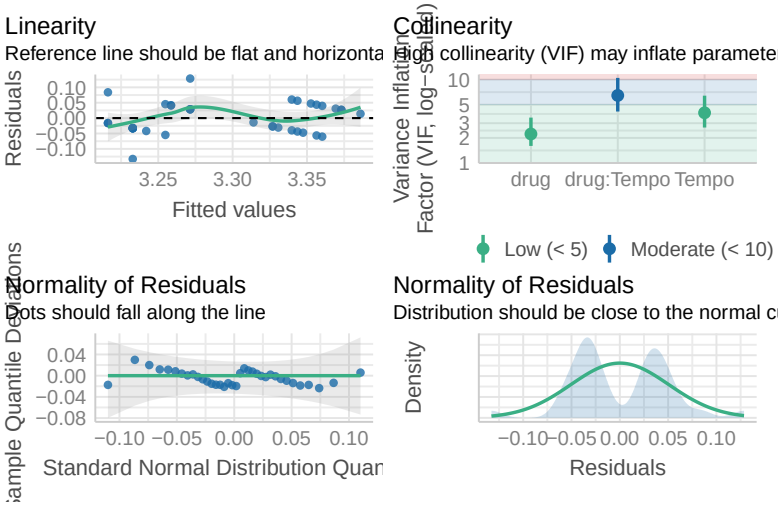
```
Linear mixed-effects model fit by REML
  Data: bd_long
    AIC   BIC logLik
  -53.4 -42.2   34.7

Random effects:
 Formula: ~1 | ID
        (Intercept) Residual
StdDev:      0.0269   0.0587

Fixed effects:  resp ~ drug + Tempo + drug * Tempo
                   Value Std.Error DF t-value p-value
(Intercept)         3.35    0.0264 20   127.1  0.0000
drugPlacebo        -0.12    0.0373 10    -3.1  0.0107
Tempo2              0.02    0.0339 20     0.5  0.6282
Tempo3             -0.02    0.0339 20    -0.5  0.6282
drugPlacebo:Tempo2  0.00    0.0479 20     0.0  1.0000
```

```
drugPlacebo:Tempo3  0.03    0.0479 20     0.7  0.4947
 Correlation:
                   (Intr) drgPlc Tempo2 Tempo3 drP:T2
drugPlacebo        -0.707
Tempo2             -0.643  0.455
Tempo3             -0.643  0.455  0.500
drugPlacebo:Tempo2  0.455 -0.643 -0.707 -0.354
drugPlacebo:Tempo3  0.455 -0.643 -0.354 -0.707  0.500

Standardized Within-Group Residuals:
   Min     Q1    Med     Q3    Max
-2.263 -0.560 -0.257  0.685  2.190

Number of Observations: 36
Number of Groups: 12
```

### Pressupostos do modelo GMM (resp)

Agora vamos fazer um plot do histograma dos resíduos do modelo

```
check_model(modelo_gmm_resp)
```

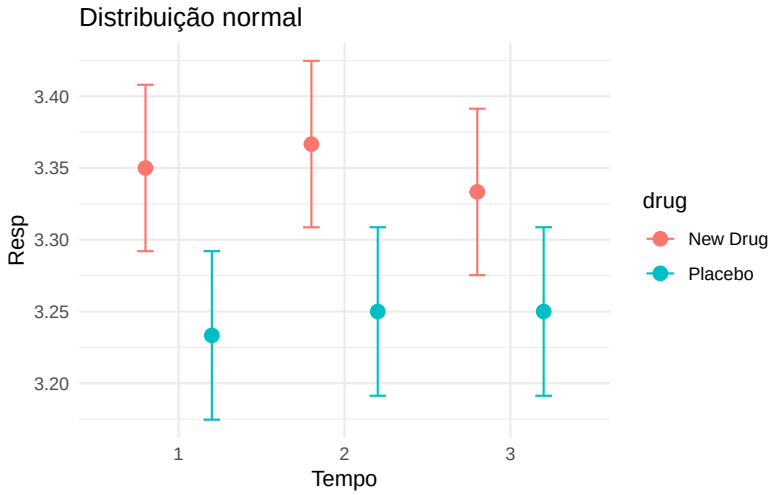


### Plot do modelo GMM (resp)

Agora vamos fazer um plot do modelo.

> Dica!
>
> A função `visualize()` as vezes não funciona com determinados modelos. Vamos fazer o gráfico na mão utilizando a função `ggplot()` do pacote `ggplot2`

```
# Vamos guardar os resultados do modelo em uma variável
means_ci_gmm_resp = emmeans(modelo_gmm_resp, specs = ~drug:Tempo)

# Código para criar nosso gráfico
ggplot(as.data.frame(means_ci_gmm_resp), aes(x = Tempo, y = emmean, color = drug)) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.2, position = position_dodg
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição normal",
       x = "Tempo",
       y = "Resp") +
  theme_minimal()
```



**Resultados do modelo**

```
report(modelo_gmm_resp)
```

```
We fitted a linear mixed model (estimated using REML and nlminb optimizer) to
predict resp with drug and Tempo (formula: resp ~ drug + Tempo + drug * Tempo).
The model included ID as random effect (formula: ~1 | ID). The model's total
explanatory power is substantial (conditional R2 = 0.52) and the part related
to the fixed effects alone (marginal R2) is of 0.42. The model's intercept,
corresponding to drug = New Drug and Tempo = 1, is at 3.35 (95% CI [3.30,
3.40], t(20) = 127.12, p < .001). Within this model:

  - The effect of drug [Placebo] is statistically significant and negative (beta
= -0.12, 95% CI [-0.20, -0.03], t(10) = -3.13, p = 0.011; Std. beta = -1.44,
95% CI [-2.47, -0.42])
  - The effect of Tempo [2] is statistically non-significant and positive (beta =
0.02, 95% CI [-0.05, 0.09], t(20) = 0.49, p = 0.628; Std. beta = 0.21, 95% CI
[-0.67, 1.08])
  - The effect of Tempo [3] is statistically non-significant and negative (beta =
-0.02, 95% CI [-0.09, 0.05], t(20) = -0.49, p = 0.628; Std. beta = -0.21, 95%
CI [-1.08, 0.67])
  - The effect of drug [Placebo] × Tempo [2] is statistically non-significant and
negative (beta = -6.16e-16, 95% CI [-0.10, 0.10], t(20) = -1.29e-14, p > .999;
Std. beta = -3.14e-16, 95% CI [-1.23, 1.23])
  - The effect of drug [Placebo] × Tempo [3] is statistically non-significant and
positive (beta = 0.03, 95% CI [-0.07, 0.13], t(20) = 0.70, p = 0.495; Std. beta
= 0.41, 95% CI [-0.82, 1.65])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald t-distribution approximation.
```

### 1.3.2 Análise para a Variável "Pulse"

Agora, realizaremos uma análise GMM para a variável "pulse".

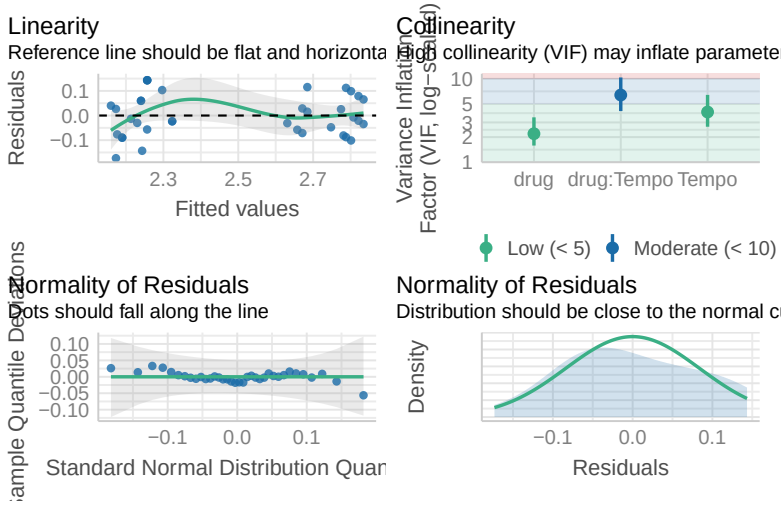
```r
# Ajustando o modelo GMM para "pulse"
modelo_gmm_pulse = lme(   fixed = pulse ~ drug + Tempo + drug * Tempo,
                          random = ~1|ID,
                          data = bd_long )

summary(modelo_gmm_pulse)
```

```
Linear mixed-effects model fit by REML
  Data: bd_long
    AIC   BIC logLik
  -22.9 -11.6   19.4

Random effects:
 Formula: ~1 | ID
        (Intercept) Residual
StdDev:      0.0459   0.0972

Fixed effects:  pulse ~ drug + Tempo + drug * Tempo
                   Value Std.Error DF t-value p-value
(Intercept)        2.200    0.0439 20    50.1   0.000
drugPlacebo        0.467    0.0621 10     7.5   0.000
Tempo2             0.017    0.0561 20     0.3   0.769
Tempo3             0.083    0.0561 20     1.5   0.153
drugPlacebo:Tempo2 0.133    0.0793 20     1.7   0.108
drugPlacebo:Tempo3 0.033    0.0793 20     0.4   0.679
 Correlation:
                   (Intr) drgPlc Tempo2 Tempo3 drP:T2
drugPlacebo        -0.707
Tempo2             -0.639  0.452
Tempo3             -0.639  0.452  0.500
drugPlacebo:Tempo2  0.452 -0.639 -0.707 -0.354
drugPlacebo:Tempo3  0.452 -0.639 -0.354 -0.707  0.500

Standardized Within-Group Residuals:
   Min     Q1    Med     Q3    Max
-1.783 -0.629 -0.178  0.630  1.476

Number of Observations: 36
Number of Groups: 12
```

### Pressupostos do modelo GMM (pulse)

Agora vamos fazer um plot do histograma dos resíduos do modelo

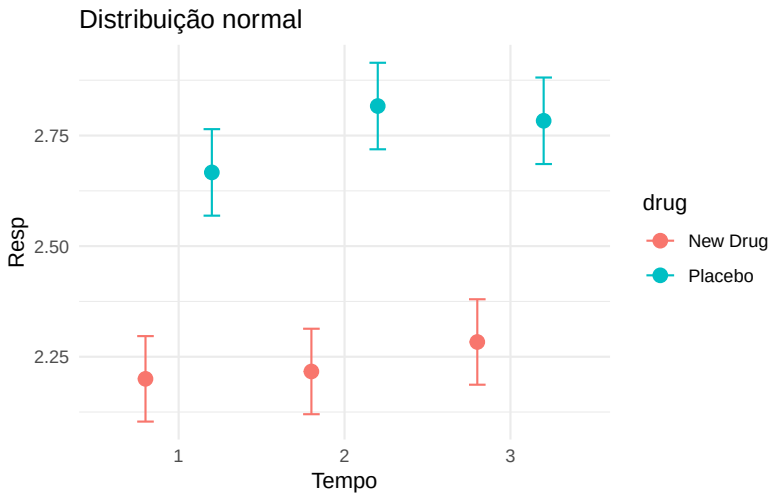
```
check_model(modelo_gmm_pulse)
```




### Plot do modelo GMM (pulse)

Agora vamos fazer um plot do modelo utilizando mais uma vez a função `ggplot()`

```
# Vamos guardar os resultados do modelo em uma variável
means_ci_gmm_pulse = emmeans(modelo_gmm_pulse, specs = ~drug:Tempo)

# Código para criar nosso gráfico
ggplot(as.data.frame(means_ci_gmm_pulse), aes(x = Tempo, y = emmean, color = drug)) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.2, position = position_dodg
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição normal",
       x = "Tempo",
       y = "Resp") +
  theme_minimal()
```




### Resultados do modelo

```
report(modelo_gmm_pulse)
```

```
We fitted a linear mixed model (estimated using REML and nlminb optimizer) to
predict pulse with drug and Tempo (formula: pulse ~ drug + Tempo + drug *
Tempo). The model included ID as random effect (formula: ~1 | ID). The model's
total explanatory power is substantial (conditional R2 = 0.89) and the part
related to the fixed effects alone (marginal R2) is of 0.86. The model's
intercept, corresponding to drug = New Drug and Tempo = 1, is at 2.20 (95% CI
[2.11, 2.29], t(20) = 50.13, p < .001). Within this model:

  - The effect of drug [Placebo] is statistically significant and positive (beta
= 0.47, 95% CI [0.33, 0.60], t(10) = 7.52, p < .001; Std. beta = 1.62, 95% CI
[1.14, 2.10])
  - The effect of Tempo [2] is statistically non-significant and positive (beta =
0.02, 95% CI [-0.10, 0.13], t(20) = 0.30, p = 0.769; Std. beta = 0.06, 95% CI
[-0.35, 0.46])
  - The effect of Tempo [3] is statistically non-significant and positive (beta =
0.08, 95% CI [-0.03, 0.20], t(20) = 1.49, p = 0.153; Std. beta = 0.29, 95% CI
[-0.12, 0.70])
  - The effect of drug [Placebo] × Tempo [2] is statistically non-significant and
positive (beta = 0.13, 95% CI [-0.03, 0.30], t(20) = 1.68, p = 0.108; Std. beta
= 0.46, 95% CI [-0.11, 1.04])
```

```
  - The effect of drug [Placebo] × Tempo [3] is statistically non-significant and
positive (beta = 0.03, 95% CI [-0.13, 0.20], t(20) = 0.42, p = 0.679; Std. beta
= 0.12, 95% CI [-0.46, 0.69])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald t-distribution approximation.
```

## 1.4 Violação dos pressupostos: o que pode ser feito?

- **Multicolinearidade**: Identifique quais variáveis independentes estão altamente correlacionadas entre si. Considere remover uma das variáveis altamente correlacionadas ou combinar variáveis para criar índices compostos.
- **Normalidade dos Resíduos:** Verifique a presença de padrões nos resíduos e considere transformações nos dados, como a transformação logarítmica. Considere a utilização de modelos mais robustos que não dependam da normalidade dos resíduos.
- **Homogeneidade de Variância:** Se a variância dos resíduos não é constante, considere transformações nos dados ou na variável dependente.
- **Outliers**: Identifique e investigue pontos de dados que se destacam nos resíduos. Avalie se a exclusão dos outliers é justificada ou se é necessário aplicar transformações aos dados. Considere modelos mais robustos que não sejam sensíveis a outliers.
- **Linearidade**: Se a relação entre variáveis independentes e dependentes não é linear, considere transformações nos dados ou nas variáveis. Utilize técnicas de modelagem não linear.
- **Posterior Predictive Checks:** Se houver discrepâncias entre dados reais e simulados nos checks pós-predição, reveja a especificação do modelo. Considere ajustes nas

distribuições ou estruturas do modelo para melhorar o ajuste.

## 1.5 Conclusão

Neste tutorial, exploramos como conduzir análises estatísticas no R utilizando diferentes abordagens, incluindo modelos de medidas repetidas, Generalized Estimating Equations (GEE) e Generalized Mixed Models (GMM). Essas técnicas nos permitem entender melhor o efeito do tempo e do grupo sobre as variáveis "resp" e "pulse" em nosso conjunto de dados.

Lembre-se de que as tabelas e resultados aqui apresentados são apenas parte da análise completa. Assista aos vídeos das aulas para entender melhor a teoria.

## 1.6 Lista 1 resolvida no SPSS

https://www.youtube.com/watch?v=_KtjZcaMYhk&list=PLZjaOxYREinslupYDLvknGMB-U-Kl8G7k&index=1

## 1.7 Referências


- Mauchly's Test of Sphericity in R
- Pivoting multiple variables: A simpler (more complex?) way
- Wide to long format part 2 - Pivoting with multivariate data


## 1.8 Versões dos pacotes

```
report(sessionInfo())
```

Analyses were conducted using the R Statistical language (version 4.3.1; R Core Team, 2023) on Windows 11 x64 (build 22621), using the packages lme4 (version 1.1.34; Bates D et al., 2015), Matrix (version 1.6.0; Bates D et al., 2023), effectsize (version 0.8.6; Ben-Shachar MS et al., 2020), flexplot (version 0.20.5; Fife D, 2024), effects (version 4.2.2; Fox J, Weisberg S, 2019), carData (version 3.0.5; Fox J et al., 2022), mvtnorm (version 1.2.3; Genz A, Bretz F, 2009), geepack (version 1.3.9; Halekoh U et al., 2006), TH.data (version 1.1.2; Hothorn T, 2023), multcomp (version 1.4.25; Hothorn T et al., 2008), rstatix (version 0.7.2; Kassambara A, 2023), emmeans (version 1.8.8; Lenth R, 2023), sjstats (version 0.18.2; Lüdecke D, 2022), parameters (version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), datawizard (version 0.9.0; Patil I et al., 2022), nlme (version 3.1.163; Pinheiro J et al., 2023), foreign (version 0.8.85; R Core Team, 2023), rempsyc (version 0.1.6; Thériault R, 2023), survival (version 3.5.7; Therneau T, 2023), MASS (version 7.3.60; Venables WN, Ripley BD, 2002), ggplot2 (version 3.4.4; Wickham H, 2016), dplyr (version 1.1.3; Wickham H et al., 2023) and tidyr (version 1.3.0; Wickham H et al., 2023).

References
---


- Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." _Journal of Statistical Software_, *67*(1), 1-48. doi:10.18637/jss.v067.i01 <https://doi.org/10.18637/jss.v067.i01>.
- Bates D, Maechler M, Jagan M (2023). _Matrix: Sparse and Dense Matrix Classes and Methods_. R package version 1.6-0, <https://CRAN.R-project.org/package=Matrix>.
- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, 3rd edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html>. Fox J, Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression

Models with Predictor Effect Plots and Partial Residuals." _Journal of Statistical Software_, *87*(9), 1-27. doi:10.18637/jss.v087.i09 <https://doi.org/10.18637/jss.v087.i09>. Fox J (2003). "Effect Displays in R for Generalised Linear Models." _Journal of Statistical Software_, *8*(15), 1-27. doi:10.18637/jss.v008.i15 <https://doi.org/10.18637/jss.v008.i15>. Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." _Journal of Statistical Software_, *32*(1), 1-24. doi:10.18637/jss.v032.i01 <https://doi.org/10.18637/jss.v032.i01>.

- Fox J, Weisberg S, Price B (2022). _carData: Companion to Applied Regression Data Sets_. R package version 3.0-5, <https://CRAN.R-project.org/package=carData>.

- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.

- Halekoh U, Højsgaard S, Yan J (2006). "The R Package geepack for Generalized Estimating Equations." _Journal of Statistical Software_, *15/2*, 1-11. Yan J, Fine JP (2004). "Estimating Equations for Association Structures." _Statistics in Medicine_, *23*, 859-880. Yan J (2002). "geepack: Yet Another Package for Generalized Estimating Equations." _R-News_, *2/3*, 12-14.

- Hothorn T (2023). _TH.data: TH's Data Archive_. R package version 1.1-2, <https://CRAN.R-project.org/package=TH.data>.

- Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." _Biometrical Journal_, *50*(3), 346-363.

- Kassambara A (2023). _rstatix: Pipe-Friendly Framework for Basic Statistical Tests_. R package version 0.7.2, <https://CRAN.R-project.org/package=rstatix>.

- Lenth R (2023). _emmeans: Estimated Marginal Means, aka Least-Squares Means_. R package version 1.8.8, <https://CRAN.R-project.org/package=emmeans>.

- Lüdecke D (2022). _sjstats: Statistical Functions for Regression Models (Version 0.18.2)_. doi:10.5281/zenodo.1284472 <https://doi.org/10.5281/zenodo.1284472>, <https://CRAN.R-project.org/package=sjstats>.

- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.

- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.

- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats:

Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.

- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.

- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.

- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.

- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_. <https://github.com/easystats/modelbased>.

- Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.

- Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.

- Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.

- Pinheiro J, Bates D, R Core Team (2023). _nlme: Linear and Nonlinear Mixed Effects Models_. R package version 3.1-163, <https://CRAN.R-project.org/package=nlme>. Pinheiro JC, Bates DM (2000). _Mixed-Effects Models in S and S-PLUS_. Springer, New York. doi:10.1007/b98882 <https://doi.org/10.1007/b98882>.

- R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.

- R Core Team (2023). _R: A Language and Environment for Statistical

Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

- Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.

- Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.

- Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.

- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.

- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.

- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.

# 2 Lista 2 - GEE

## 2.1 Introdução

Na lista 2 vamos utilizar mais uma vez o banco de dados New Drug com medidas de resp e pulse. Não se esqueça de transformá-lo para o formato long como no exercício anterior.

```
# Carregar o banco de dados e armazena na variável original_wide

original_wide = read.spss("bd_New drug_respiratory&pulse.sav", to.data.frame=TRUE)
```

```
# Renomear as colunas do banco de dados para facilitar a conversão para o formato longo.

bd <- original_wide %>%   rename_with(~gsub("(resp|pulse)(\\d+)", "\\1_\\2", .), -drug) %>%
```

```
# Organizar os dados para o formato longo

bd_long = pivot_longer(bd, cols=resp_1:pulse_3, names_to = c(".value", "Tempo"),   names_pat
```

Primeiras linhas do banco de dados depois de transformado.

```
head(bd_long)
```

```
# A tibble: 6 x 5
     ID drug     Tempo  resp pulse
  <int> <fct>    <chr> <dbl> <dbl>
1     1 New Drug 1       3.4   2.2
2     1 New Drug 2       3.3   2.1
3     1 New Drug 3       3.3   2.1
4     2 New Drug 1       3.4   2.2
5     2 New Drug 2       3.4   2.1
```

```
6     2 New Drug 3       3.3   2.2
```

---

## 2.2 Exercícios

### 2.2.1 a) GEE com a VD "Pulse"

Utilize um GEE para verificar o efeito de tempo e grupo sobre os resultados de resp e pulse. Faça 3 modelos para cada variável dependente (com as distribuições Normal, Gamma e Tweedie) e cole aqui apenas as tabelas relevantes para a análise.

#### Distribuição normal

#### Criando o modelo

```r
modelo_gee_pulse_normal <- geeglm(pulse ~ drug + Tempo + drug*Tempo,
                            data = bd_long,
                            id = ID,
                            family = gaussian, #Distribuição normal
                            corstr = "unstructured")
```

#### Resumo do modelo e contrastes

```r
summary(modelo_gee_pulse_normal)
```

```
Call:
geeglm(formula = pulse ~ drug + Tempo + drug * Tempo, family = gaussian,
    data = bd_long, id = ID, corstr = "unstructured")

 Coefficients:
                  Estimate Std.err     Wald Pr(>|W|)
(Intercept)        2.20000 0.04082 2904.000  < 2e-16 ***
drugPlacebo        0.46667 0.05092   84.000  < 2e-16 ***
```

```
Tempo2              0.01667 0.03664    0.207  0.64921
Tempo3              0.08333 0.06838    1.485  0.22297
drugPlacebo:Tempo2  0.13333 0.04194   10.105  0.00148 **
drugPlacebo:Tempo3  0.03333 0.08767    0.145  0.70377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = unstructured
Estimated Scale Parameters:

            Estimate  Std.err
(Intercept)  0.00963 0.001676
  Link = identity

Estimated Correlation Parameters:
          Estimate Std.err
alpha.1:2   0.7212  0.1690
alpha.1:3  -0.2885  0.2076
alpha.2:3   0.1154  0.2665
Number of clusters:   12  Maximum cluster size: 3
```

```
emmeans(modelo_gee_pulse_normal, pairwise ~ drug*Tempo)
```

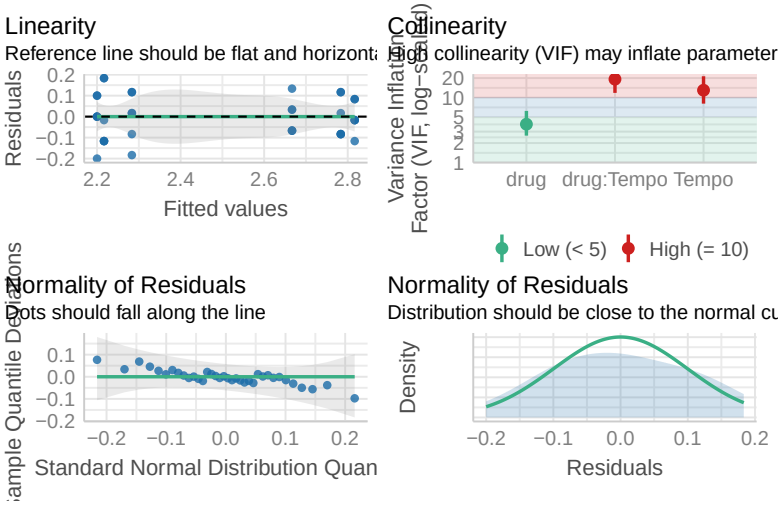
```
$emmeans
 drug     Tempo emmean     SE  df asymp.LCL asymp.UCL
 New Drug 1       2.20 0.0408 Inf      2.12      2.28
 Placebo  1       2.67 0.0304 Inf      2.61      2.73
 New Drug 2       2.22 0.0549 Inf      2.11      2.32
 Placebo  2       2.82 0.0280 Inf      2.76      2.87
 New Drug 3       2.28 0.0436 Inf      2.20      2.37
 Placebo  3       2.78 0.0366 Inf      2.71      2.86

Covariance estimate used: vbeta
Confidence level used: 0.95

$contrasts
 contrast                           estimate     SE  df z.ratio p.value
 New Drug Tempo1 - Placebo Tempo1   -0.4667 0.0509 Inf  -9.165  <.0001
 New Drug Tempo1 - New Drug Tempo2  -0.0167 0.0366 Inf  -0.455  0.9976
 New Drug Tempo1 - Placebo Tempo2   -0.6167 0.0495 Inf -12.449  <.0001
```

```
 New Drug Tempo1 - New Drug Tempo3  -0.0833 0.0684 Inf  -1.219  0.8279
 New Drug Tempo1 - Placebo Tempo3   -0.5833 0.0549 Inf -10.634  <.0001
 Placebo Tempo1 - New Drug Tempo2    0.4500 0.0627 Inf   7.173  <.0001
 Placebo Tempo1 - Placebo Tempo2    -0.1500 0.0204 Inf  -7.348  <.0001
 Placebo Tempo1 - New Drug Tempo3    0.3833 0.0531 Inf   7.213  <.0001
 Placebo Tempo1 - Placebo Tempo3    -0.1167 0.0549 Inf  -2.127  0.2733
 New Drug Tempo2 - Placebo Tempo2   -0.6000 0.0616 Inf  -9.738  <.0001
 New Drug Tempo2 - New Drug Tempo3  -0.0667 0.0609 Inf  -1.095  0.8834
 New Drug Tempo2 - Placebo Tempo3   -0.5667 0.0660 Inf  -8.590  <.0001
 Placebo Tempo2 - New Drug Tempo3    0.5333 0.0518 Inf  10.292  <.0001
 Placebo Tempo2 - Placebo Tempo3     0.0333 0.0509 Inf   0.655  0.9867
 New Drug Tempo3 - Placebo Tempo3   -0.5000 0.0569 Inf  -8.783  <.0001

P value adjustment: tukey method for comparing a family of 6 estimates
```

### Verificando os pressupostos

```
# Plotar o diagnóstico do modelo GEE para a variável 'pulse'

check_model(modelo_gee_pulse_normal)
```

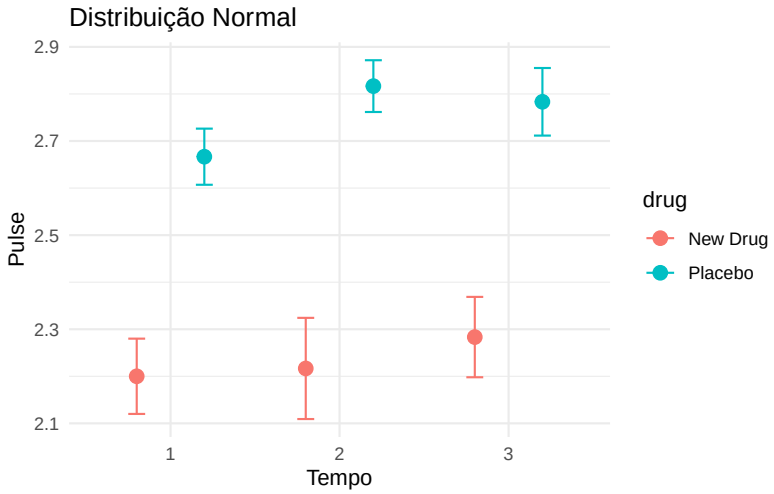
```
Converting missing values (`NA`) into regular values currently not
  possible for variables of class `NULL`.
```



### Plot dos resultados

```
# Salvando os resultados do modelo em uma variável

means_ci_normal = emmeans(modelo_gee_pulse_normal, specs = ~drug:Tempo)
```

```
# Plotar as médias e intervalos de confiança

ggplot(as.data.frame(means_ci_normal), aes(x = Tempo, y = emmean, color = drug)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, position = position_do
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição Normal",
       x = "Tempo",
       y = "Pulse") +
  theme_minimal()
```




---

### Distribuição gamma

### Criando o modelo

```r
modelo_gee_pulse_gamma <- geeglm(pulse ~ drug + Tempo + drug*Tempo,
                            data = bd_long,
                            id = ID,
                            family = Gamma(link = "identity"), #Distribuição Gamma
                            corstr = "unstructured")
```

**Resumo do modelo e contrastes**

```r
summary(modelo_gee_pulse_gamma)
```

```
Call:
geeglm(formula = pulse ~ drug + Tempo + drug * Tempo, family = Gamma(link = "identity"),
    data = bd_long, id = ID, corstr = "unstructured")

 Coefficients:
                  Estimate Std.err    Wald Pr(>|W|)
(Intercept)         2.2000  0.0408 2904.00   <2e-16 ***
drugPlacebo         0.4667  0.0509   84.00   <2e-16 ***
Tempo2              0.0167  0.0366    0.21   0.6492
Tempo3              0.0833  0.0684    1.49   0.2230
drugPlacebo:Tempo2  0.1333  0.0419   10.11   0.0015 **
drugPlacebo:Tempo3  0.0333  0.0877    0.14   0.7038
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = unstructured
Estimated Scale Parameters:

            Estimate  Std.err
(Intercept)  0.00172 0.000372
  Link = identity

Estimated Correlation Parameters:
          Estimate Std.err
alpha.1:2    0.745   0.178
alpha.1:3   -0.279   0.208
alpha.2:3    0.156   0.279
Number of clusters:   12  Maximum cluster size: 3
```

```r
emmeans(modelo_gee_pulse_gamma, pairwise ~ drug*Tempo)
```

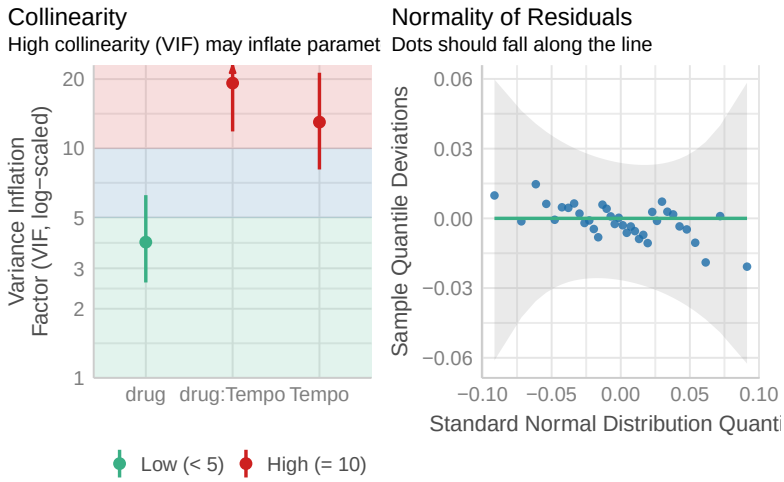
```
$emmeans
 drug     Tempo emmean     SE  df asymp.LCL asymp.UCL
 New Drug 1       2.20 0.0408 Inf      2.12      2.28
 Placebo  1       2.67 0.0304 Inf      2.61      2.73
 New Drug 2       2.22 0.0549 Inf      2.11      2.32
 Placebo  2       2.82 0.0281 Inf      2.76      2.87
 New Drug 3       2.28 0.0436 Inf      2.20      2.37
 Placebo  3       2.78 0.0366 Inf      2.71      2.85

Covariance estimate used: vbeta
Confidence level used: 0.95

$contrasts
 contrast                           estimate     SE  df z.ratio p.value
 New Drug Tempo1 - Placebo Tempo1    -0.467 0.0509 Inf  -9.170  <.0001
 New Drug Tempo1 - New Drug Tempo2   -0.017 0.0366 Inf  -0.450  0.9980
 New Drug Tempo1 - Placebo Tempo2    -0.617 0.0495 Inf -12.450  <.0001
 New Drug Tempo1 - New Drug Tempo3   -0.083 0.0684 Inf  -1.220  0.8280
 New Drug Tempo1 - Placebo Tempo3    -0.583 0.0549 Inf -10.630  <.0001
 Placebo Tempo1 - New Drug Tempo2     0.450 0.0627 Inf   7.170  <.0001
 Placebo Tempo1 - Placebo Tempo2     -0.150 0.0204 Inf  -7.350  <.0001
 Placebo Tempo1 - New Drug Tempo3     0.383 0.0531 Inf   7.210  <.0001
 Placebo Tempo1 - Placebo Tempo3     -0.117 0.0549 Inf  -2.130  0.2730
 New Drug Tempo2 - Placebo Tempo2    -0.600 0.0616 Inf  -9.740  <.0001
 New Drug Tempo2 - New Drug Tempo3   -0.067 0.0609 Inf  -1.100  0.8830
 New Drug Tempo2 - Placebo Tempo3    -0.567 0.0660 Inf  -8.590  <.0001
 Placebo Tempo2 - New Drug Tempo3     0.533 0.0518 Inf  10.290  <.0001
 Placebo Tempo2 - Placebo Tempo3      0.033 0.0509 Inf   0.650  0.9870
 New Drug Tempo3 - Placebo Tempo3    -0.500 0.0569 Inf  -8.780  <.0001

P value adjustment: tukey method for comparing a family of 6 estimates
```

### Verificando os pressupostos

```r
# Plotar o diagnóstico do modelo GEE para a variável 'pulse'
```

```
check_model(modelo_gee_pulse_gamma)
```

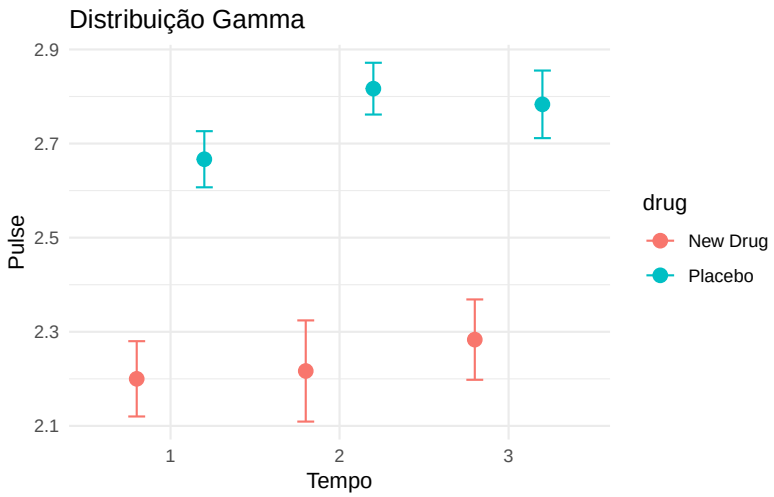
```
Converting missing values (`NA`) into regular values currently not
  possible for variables of class `NULL`.
```




### Plot dos resultados

```
# Salvando os resultados do modelo em uma variável

means_ci_gamma = emmeans(modelo_gee_pulse_gamma, specs = ~drug:Tempo)
```

```
# Plotar as médias e intervalos de confiança

ggplot(as.data.frame(means_ci_gamma), aes(x = Tempo, y = emmean, color = drug)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, position = position_do
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição Gamma",
       x = "Tempo",
       y = "Pulse") +
  theme_minimal()
```




---

### Distribuição tweedie

### Criando o modelo

```
modelo_gee_pulse_tweedie <- glm(pulse ~ drug + Tempo + drug*Tempo,
                            data = bd_long,
                           # id = ID,
                            family = tweedie(var.power=2, link.power = 0),
                           contrasts = )
```

> **Aviso!**
>
> Utilizamos a função glm para criar o modelo Tweedie. Estamos trabalhando para criar o modelo com a função GEE. Por hora utilize o SPSS .

### Resumo do modelo e contrastes

```
summary(modelo_gee_pulse_tweedie)
```

```
Call:
glm(formula = pulse ~ drug + Tempo + drug * Tempo, family = tweedie(var.power = 2,
    link.power = 0), data = bd_long)

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.78846    0.01857   42.47  < 2e-16 ***
drugPlacebo         0.19237    0.02626    7.33  3.7e-08 ***
Tempo2              0.00755    0.02626    0.29     0.78
Tempo3              0.03718    0.02626    1.42     0.17
drugPlacebo:Tempo2  0.04718    0.03713    1.27     0.21
drugPlacebo:Tempo3  0.00564    0.03713    0.15     0.88
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Tweedie family taken to be 0.00207)

    Null deviance: 0.473395  on 35  degrees of freedom
Residual deviance: 0.062212  on 30  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 3
```

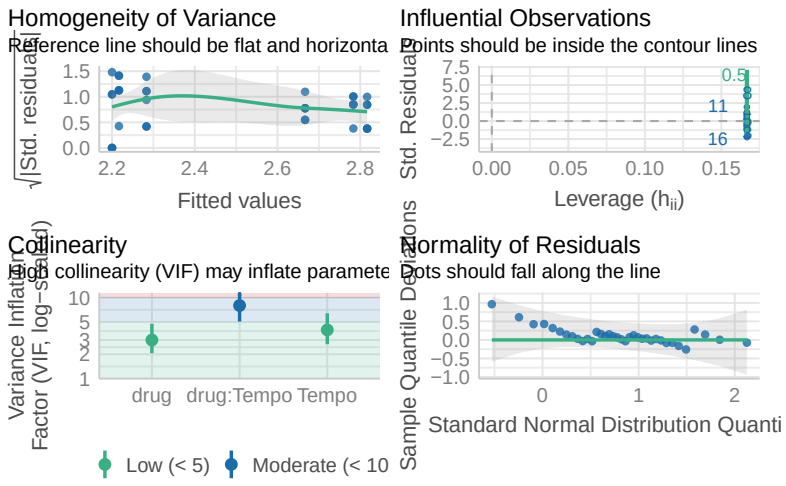
```r
emmeans(modelo_gee_pulse_tweedie, pairwise ~ drug*Tempo)
```

```
$emmeans
 drug     Tempo emmean     SE  df asymp.LCL asymp.UCL
 New Drug 1      0.788 0.0186 Inf     0.752     0.825
 Placebo  1      0.981 0.0186 Inf     0.944     1.017
 New Drug 2      0.796 0.0186 Inf     0.760     0.832
 Placebo  2      1.036 0.0186 Inf     0.999     1.072
 New Drug 3      0.826 0.0186 Inf     0.789     0.862
 Placebo  3      1.024 0.0186 Inf     0.987     1.060

Results are given on the mu^0 (not the response) scale.
Confidence level used: 0.95

$contrasts
 contrast                        estimate     SE  df z.ratio p.value
```

```
 New Drug Tempo1 - Placebo Tempo1   -0.1924 0.0263 Inf  -7.330  <.0001
 New Drug Tempo1 - New Drug Tempo2  -0.0075 0.0263 Inf  -0.290  1.0000
 New Drug Tempo1 - Placebo Tempo2   -0.2471 0.0263 Inf  -9.410  <.0001
 New Drug Tempo1 - New Drug Tempo3  -0.0372 0.0263 Inf  -1.420  0.7170
 New Drug Tempo1 - Placebo Tempo3   -0.2352 0.0263 Inf  -8.960  <.0001
 Placebo Tempo1 - New Drug Tempo2    0.1848 0.0263 Inf   7.040  <.0001
 Placebo Tempo1 - Placebo Tempo2    -0.0547 0.0263 Inf  -2.080  0.2950
 Placebo Tempo1 - New Drug Tempo3    0.1552 0.0263 Inf   5.910  <.0001
 Placebo Tempo1 - Placebo Tempo3    -0.0428 0.0263 Inf  -1.630  0.5780
 New Drug Tempo2 - Placebo Tempo2   -0.2395 0.0263 Inf  -9.120  <.0001
 New Drug Tempo2 - New Drug Tempo3  -0.0296 0.0263 Inf  -1.130  0.8700
 New Drug Tempo2 - Placebo Tempo3   -0.2276 0.0263 Inf  -8.670  <.0001
 Placebo Tempo2 - New Drug Tempo3    0.2099 0.0263 Inf   7.990  <.0001
 Placebo Tempo2 - Placebo Tempo3     0.0119 0.0263 Inf   0.450  0.9980
 New Drug Tempo3 - Placebo Tempo3   -0.1980 0.0263 Inf  -7.540  <.0001

Note: contrasts are still on the mu^0 scale
P value adjustment: tukey method for comparing a family of 6 estimates
```

### Verificando os pressupostos

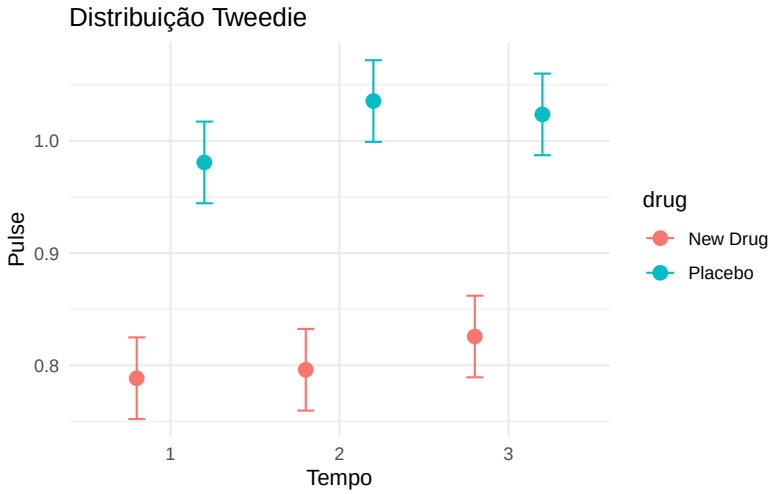
```
# Plotar o diagnóstico do modelo GEE para a variável 'pulse'

check_model(modelo_gee_pulse_tweedie)
```



**Plot dos resultados**

```
# Salvando os resultados do modelo em uma variável

means_ci_tweedie = emmeans(modelo_gee_pulse_tweedie, specs = ~drug:Tempo)
```

```
# Plotar as médias e intervalos de confiança

ggplot(as.data.frame(means_ci_tweedie), aes(x = Tempo, y = emmean, color = drug)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, position = position_do
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição Tweedie",
       x = "Tempo",
       y = "Pulse") +
  theme_minimal()
```




---

### 2.2.2 b) QIC

Compare cada um dos modelos com diferentes distribuições utilizando o QIC (Quasi Likehood Independence Criterion). Os modelos têm diferença entre si nos resultados?

> **Nota**
>
> A função `QIC()` não funciona para modelos gerados com as funções glm e lm, apenas com o GEE. Resolveremos em breve! Por hora utilize o SPSS .

```
QIC(modelo_gee_pulse_normal)
```

```
    QIC      QICu Quasi Lik       CIC    params      QICC
 12.347    12.347    -0.173     6.000     6.000   102.347
```

```
QIC(modelo_gee_pulse_gamma)
```

```
    QIC      QICu Quasi Lik       CIC    params      QICC
  125.2     125.2     -56.6       6.0       6.0     215.2
```

```
#QIC(modelo_gee_pulse_tweedie)
```

### 2.2.3 c) Sumarizando os resultados

> **Nota**
>
> A função `report()` não funciona para modelos gerados com as funções GEE. Aproveite para treinar a escrita no formato de uma publicação acadêmica.

#### Resutados com distribuição Tweedie

```
report(modelo_gee_pulse_tweedie)
```

```
We fitted a general linear model (Tweedie family with a mu^0 link) (estimated
using ML) to predict pulse with drug and Tempo (formula: pulse ~ drug + Tempo +
drug * Tempo). The model's explanatory power is substantial (Nagelkerke's R2 =
0.87). The model's intercept, corresponding to drug = New Drug and Tempo = 1,
```

```
is at 0.79 (95% CI [0.75, 0.83], p < .001). Within this model:

  - The effect of drug [Placebo] is statistically significant and positive (beta
= 0.19, 95% CI [0.14, 0.24], p < .001; Std. beta = 0.19, 95% CI [0.14, 0.24])
  - The effect of Tempo [2] is statistically non-significant and positive (beta =
7.55e-03, 95% CI [-0.04, 0.06], p = 0.774; Std. beta = 7.55e-03, 95% CI [-0.04,
0.06])
  - The effect of Tempo [3] is statistically non-significant and positive (beta =
0.04, 95% CI [-0.01, 0.09], p = 0.157; Std. beta = 0.04, 95% CI [-0.01, 0.09])
  - The effect of drug [Placebo] × Tempo [2] is statistically non-significant and
positive (beta = 0.05, 95% CI [-0.03, 0.12], p = 0.204; Std. beta = 0.05, 95%
CI [-0.03, 0.12])
  - The effect of drug [Placebo] × Tempo [3] is statistically non-significant and
positive (beta = 5.64e-03, 95% CI [-0.07, 0.08], p = 0.879; Std. beta =
5.64e-03, 95% CI [-0.07, 0.08])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald z-distribution approximation.
```

## 2.3 Considerações finais

Realizamos todas as análises para a VD Pulse! Agora faça as análises para a variável Reps!

> Dica!
>
> Não faça apenas um copy/paste dos scripts! Treine escrever os códigos e lembre-se de mudar o nome das variáveis do modelo para que não ocorra nenhum conflito! Compare seus resultados com os da aula prática.

## 2.4 Lista 2 resolvida no SPSS

https://www.youtube.com/watch?v=qd0qF2lRqIs

## 2.5 Referências

## 2.6 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages lme4 (version
1.1.34; Bates D et al., 2015), Matrix (version 1.6.0; Bates D et al., 2023),
effectsize (version 0.8.6; Ben-Shachar MS et al., 2020), gee (version 4.13.26;
Carey VJ, 2023), pwr (version 1.3.0; Champely S, 2020), htmltools (version
0.5.7; Cheng J et al., 2023), fitdistrplus (version 1.1.11; Delignette-Muller
ML, Dutang C, 2015), tweedie (version 2.3.5; Dunn PK, Smyth GK, 2005), tm
(version 0.7.11; Feinerer I, Hornik K, 2023), flexplot (version 0.20.5; Fife D,
2024), effects (version 4.2.2; Fox J, Weisberg S, 2019), car (version 3.1.2;
Fox J, Weisberg S, 2019), carData (version 3.0.5; Fox J et al., 2022), mvtnorm
(version 1.2.3; Genz A, Bretz F, 2009), statmod (version 1.5.0; Giner G, Smyth
GK, 2016), geepack (version 1.3.9; Halekoh U et al., 2006), NLP (version 0.2.1;
Hornik K, 2020), TH.data (version 1.1.2; Hothorn T, 2023), multcomp (version
1.4.25; Hothorn T et al., 2008), rstatix (version 0.7.2; Kassambara A, 2023),
emmeans (version 1.8.8; Lenth R, 2023), sjstats (version 0.18.2; Lüdecke D,
2022), parameters (version 0.21.3; Lüdecke D et al., 2020), performance
(version 0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D
et al., 2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version
0.19.6; Lüdecke D et al., 2019), survey (version 4.2.1; Lumley T, 2023),
bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version
0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al.,
2023), correlation (version 0.8.4; Makowski D et al., 2022), datawizard
(version 0.9.0; Patil I et al., 2022), nlme (version 3.1.163; Pinheiro J et
al., 2023), foreign (version 0.8.85; R Core Team, 2023), GGally (version 2.2.0;
Schloerke B et al., 2023), rempsyc (version 0.1.6; Thériault R, 2023), survival
(version 3.5.7; Therneau T, 2023), MASS (version 7.3.60; Venables WN, Ripley
BD, 2002), ggplot2 (version 3.4.4; Wickham H, 2016), dplyr (version 1.1.3;
Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et al., 2023) and mime
(version 0.12; Xie Y, 2021).

References
----------
```

- Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." _Journal of Statistical Software_, *67*(1), 1-48. doi:10.18637/jss.v067.i01 <https://doi.org/10.18637/jss.v067.i01>.
- Bates D, Maechler M, Jagan M (2023). _Matrix: Sparse and Dense Matrix Classes and Methods_. R package version 1.6-0, <https://CRAN.R-project.org/package=Matrix>.
- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Carey VJ (2023). _gee: Generalized Estimation Equation Solver_. R package version 4.13-26, <https://CRAN.R-project.org/package=gee>.
- Champely S (2020). _pwr: Basic Functions for Power Analysis_. R package version 1.3-0, <https://CRAN.R-project.org/package=pwr>.
- Cheng J, Sievert C, Schloerke B, Chang W, Xie Y, Allen J (2023). _htmltools: Tools for HTML_. R package version 0.5.7, <https://CRAN.R-project.org/package=htmltools>.
- Delignette-Muller ML, Dutang C (2015). "fitdistrplus: An R Package for Fitting Distributions." _Journal of Statistical Software_, *64*(4), 1-34. doi:10.18637/jss.v064.i04 <https://doi.org/10.18637/jss.v064.i04>.
- Dunn PK, Smyth GK (2005). "Series evaluation of Tweedie exponential dispersion models." _Statistics and Computing_, *15*(4), 267-280. Dunn PK, Smyth GK (2008). "Evaluation of Tweedie exponential dispersion models using Fourier inversion." _Statistics and Computing_, *18*(1), 73-86. Dunn PK (2022). _Tweedie: Evaluation of Tweedie Exponential Family Models_. R package version 2.3.5.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, 3rd edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html>. Fox J, Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression Models with Predictor Effect Plots and Partial Residuals." _Journal of Statistical Software_, *87*(9), 1-27. doi:10.18637/jss.v087.i09 <https://doi.org/10.18637/jss.v087.i09>. Fox J (2003). "Effect Displays in R for Generalised Linear Models." _Journal of Statistical Software_, *8*(15),

1-27. doi:10.18637/jss.v008.i15 <https://doi.org/10.18637/jss.v008.i15>. Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." _Journal of Statistical Software_, *32*(1), 1-24. doi:10.18637/jss.v032.i01 <https://doi.org/10.18637/jss.v032.i01>.

- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, Third edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.

- Fox J, Weisberg S, Price B (2022). _carData: Companion to Applied Regression Data Sets_. R package version 3.0-5, <https://CRAN.R-project.org/package=carData>.

- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.

- Giner G, Smyth GK (2016). "statmod: probability calculations for the inverse Gaussian distribution." _R Journal_, *8*(1), 339-351. Phipson B, Smyth GK (2010). "Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn." _Statistical Applications in Genetics and Molecular Biology_, *9*(1), Article 39. Hu Y, Smyth GK (2009). "ELDA: extreme limiting dilution analysis for comparing depleted and enriched populations in stem cell and other assays." _Journal of Immunological Methods_, *347*(1), 70-78. Smyth GK (2005). "Optimization and nonlinear equations." _Encyclopedia of Biostatistics_, 3088-3095. Smyth GK (2005). "Numerical integration." _Encyclopedia of Biostatistics_, 3088-3095. Smyth GK (2002). "An efficient algorithm for REML in heteroscedastic regression." _Journal of Computational and Graphical Statistics_, *11*, 836-847. Dunn PK, Smyth GK (1996). "Randomized quantile residuals." _J. Comput. Graph. Statist_, *5*, 236-244.

- Halekoh U, Højsgaard S, Yan J (2006). "The R Package geepack for Generalized Estimating Equations." _Journal of Statistical Software_, *15/2*, 1-11. Yan J, Fine JP (2004). "Estimating Equations for Association Structures." _Statistics in Medicine_, *23*, 859-880. Yan J (2002). "geepack: Yet Another Package for Generalized Estimating Equations." _R-News_, *2/3*, 12-14.

- Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.

- Hothorn T (2023). _TH.data: TH's Data Archive_. R package version 1.1-2, <https://CRAN.R-project.org/package=TH.data>.

- Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." _Biometrical Journal_, *50*(3), 346-363.

- Kassambara A (2023). _rstatix: Pipe-Friendly Framework for Basic Statistical Tests_. R package version 0.7.2, <https://CRAN.R-project.org/package=rstatix>.

- Lenth R (2023). _emmeans: Estimated Marginal Means, aka Least-Squares Means_. R package version 1.8.8, <https://CRAN.R-project.org/package=emmeans>.
- Lüdecke D (2022). _sjstats: Statistical Functions for Regression Models (Version 0.18.2)_. doi:10.5281/zenodo.1284472 <https://doi.org/10.5281/zenodo.1284472>, <https://CRAN.R-project.org/package=sjstats>.
- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.
- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
- Lumley T (2023). "survey: analysis of complex survey samples." R package version 4.2. Lumley T (2004). "Analysis of Complex Survey Samples." _Journal of Statistical Software_, *9*(1), 1-19. R package verson 2.2. Lumley T (2010). _Complex Surveys: A Guide to Analysis Using R: A Guide to Analysis Using R_. John Wiley and Sons.
- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.
- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_. <https://github.com/easystats/modelbased>.
- Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.

- Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.
- Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
- Pinheiro J, Bates D, R Core Team (2023). _nlme: Linear and Nonlinear Mixed Effects Models_. R package version 3.1-163, <https://CRAN.R-project.org/package=nlme>. Pinheiro JC, Bates DM (2000). _Mixed-Effects Models in S and S-PLUS_. Springer, New York. doi:10.1007/b98882 <https://doi.org/10.1007/b98882>.
- R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
- R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Schloerke B, Cook D, Larmarange J, Briatte F, Marbach M, Thoen E, Elberg A, Crowley J (2023). _GGally: Extension to 'ggplot2'_. R package version 2.2.0, <https://CRAN.R-project.org/package=GGally>.
- Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
- Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
- Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.
- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.

- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
- Xie Y (2021). _mime: Map Filenames to MIME Types_. R package version 0.12, <https://CRAN.R-project.org/package=mime>.

# 3 Lista 3 - Matriz de Covariância

## 3.1 Pacotes que vamos utilizar

```
library(emmeans)
library(nlme)
library(flexplot)
library(foreign)
library(dplyr)
library(multcomp)
library(effects)
library(performance)
library(easystats)
```

## 3.2 Instruções e carregando o banco de dados

Vamos utilizar um GMM para verificar o efeito de tempo e grupo sobre os resultados de resp (o banco já está no formato correto). Porém, antes disso vamos avaliar qual a melhor matriz de covariância que o modelo deve aplicar aos dados.

```
dataset = read.spss("bd_New drug_respiratory&pulseRESHAPE.sav", to.data.frame=TRUE)
```

> **!** Muito importante!
>
> SEMPRE verifique o tipo das variáveis no banco de dados. Elas podem ser fatores, números íntegros, números decimais, etc. As análises mudam MUITO dependendo

> do tipo de variável que utilizamos no modelo de regressão linear!

Para verificar o tipo das variáveis podemos utilizar a função `glimpse()`.

```
glimpse(dataset)
```

```
Rows: 36
Columns: 5
$ Sujeito <dbl> 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,~
$ drug    <fct> New Drug, New Drug, New Drug, New Drug, New Drug, New Drug, Ne~
$ Tempo   <dbl> 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3,~
$ resp    <dbl> 3.4, 3.3, 3.3, 3.4, 3.4, 3.3, 3.3, 3.4, 3.4, 3.4, 3.4, 3.4, 3.~
$ pulse   <dbl> 2.2, 2.1, 2.1, 2.2, 2.1, 2.2, 2.3, 2.4, 2.3, 2.3, 2.4, 2.3, 2.~
```

Notem que a variável `Tempo` está como `<dbl>`, indicando que é uma variável contínua. Vejam a aula prática em que o Altay explica que os tempos de coleta na verdade são fatores e não indicam uma ordem ou um contínuo. Seria como dizer que as amostras foram coletadas nos Tempos "X", "Y" e "Z".

Para transformar a variável `Tempo` em um fator podemos utilizar o seguinte código:

```
dataset$Tempo = as.factor(dataset$Tempo)
```

Rodando novamente a função `glimpse()` é possível observar que agora sim temos a variável `Tempo` como `<fct>`, indicando que ela é do tipo fator .

```
glimpse(dataset)
```

```
Rows: 36
Columns: 5
$ Sujeito <dbl> 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,~
$ drug    <fct> New Drug, New Drug, New Drug, New Drug, New Drug, New Drug, Ne~
$ Tempo   <fct> 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3,~
$ resp    <dbl> 3.4, 3.3, 3.3, 3.4, 3.4, 3.3, 3.3, 3.4, 3.4, 3.4, 3.4, 3.4, 3.~
$ pulse   <dbl> 2.2, 2.1, 2.1, 2.2, 2.1, 2.2, 2.3, 2.4, 2.3, 2.3, 2.4, 2.3, 2.~
```

> **Dica**
>
> Faça os modelos antes e depois de transformar a variável `Tempo` em um fator e compare os resultados.

Agora sim podemos realizar nossas análises e comparar com os resultados do SPSS.

## 3.3 a) Criando os modelos para a variável Resp

Há diversos pacotes no R que podemos alterar a matriz de covariância do modelo. Optamos por escolher a função `lme()` do pacote `nlme` por ela ter de forma bem direta todas as matrizes escolhidas na aula prática utilizando o SPSS. Recomendamos também o pacote `geepack`para outras matrizes.

### 3.3.1 Matriz simétrica

```
model_resp_sim = lme(
  fixed = resp ~ 1 + drug + Tempo + drug * Tempo,
  random =~ 1|Sujeito,
  correlation = corCompSymm(form = ~1|Sujeito), # Aqui definimos a matriz
  data = dataset)
```

### 3.3.2 Matriz Ar(1)

```
model_resp_AR1 = lme(
  fixed = resp ~ Tempo + drug + Tempo * drug,
  random = ~1|Sujeito,
  correlation = corAR1(form = ~ 1|Sujeito), # Aqui definimos a matriz
  data = dataset)
```

### 3.3.3 Matriz Diagonal (identidade)

```
model_resp_Iden = lme(
  fixed = resp ~ drug + Tempo + drug * Tempo,
  random = ~1|Sujeito,
  correlation = corIdent(form = ~ 1|Sujeito), # Aqui definimos a matriz
  data = dataset)
```

### 3.3.4 Matriz Não estruturada (Unstructured)

```
model_resp_Uns = lme(
  fixed = resp ~ drug + Tempo + drug * Tempo,
  random = ~1|Sujeito,
  correlation = corSymm(form = ~ 1|Sujeito), # Aqui definimos a matriz
  data = dataset)
```

## 3.4 b) Comparando os valores de AIC e BIC

Podemos colocar os resultados dos valores de AIC e BIC dos modelos em um dataframe para poder compará-los.

```
# Crie um dataframe
df_aderencia <- data.frame(
  Modelo = c("model_resp_sim", "model_resp_AR1", "model_resp_Iden", "model_resp_Uns"),
  AIC = c(AIC(model_resp_sim), AIC(model_resp_AR1), AIC(model_resp_Iden), AIC(model_resp_Uns
  BIC = c(BIC(model_resp_sim), BIC(model_resp_AR1), BIC(model_resp_Iden), BIC(model_resp_Uns
)

# Arredonde os valores para 3 casas decimais
df_aderencia$AIC <- round(df_aderencia$AIC, 3)
df_aderencia$BIC <- round(df_aderencia$BIC, 3)

# Adicione um asterisco às células correspondentes aos menores valores de AIC e BIC
df_aderencia$AIC <- ifelse(df_aderencia$AIC == min(df_aderencia$AIC), paste0(df_aderencia$AI
df_aderencia$BIC <- ifelse(df_aderencia$BIC == min(df_aderencia$BIC), paste0(df_aderencia$BI
```

```r
# Exiba o dataframe
print(df_aderencia)
```

```
           Modelo     AIC      BIC
1  model_resp_sim -51.363  -38.752
2  model_resp_AR1  -54.3*   -41.69
3 model_resp_Iden -53.363 -42.153*
4  model_resp_Uns -53.455  -38.042
```

> Dica!
>
> Podemos utilizar a função `compare_performance` para comparar diversos valores de aderência dos modelos!

```r
compare_performance(model_resp_sim,
                    model_resp_AR1,
                    model_resp_Iden,
                    model_resp_Uns,
                    metrics = c("AIC","BIC"),
                    rank = TRUE)
```

```
# Comparison of Model Performance Indices

Name            | Model | AIC weights | BIC weights | Performance-Score
-------------------------------------------------------------------------
model_resp_AR1  |   lme |       0.394 |       0.419 |            98.26%
model_resp_Iden |   lme |       0.184 |       0.432 |            67.80%
model_resp_Uns  |   lme |       0.354 |       0.077 |            44.62%
model_resp_sim  |   lme |       0.068 |       0.072 |             0.00%
```

Das duas formas constatamos que o modelo com a matriz de covariância AR1 apresentou os melhores índices de aderência. Note que os valores de AIC e BIC apresentados na saída da função `compare_factors` aparecem ponderados. Os valores são calculados dividindo o peso AIC/BIC de um modelo pelo peso AIC/BIC dos outros modelos ajustados.

## 3.5 c) Resultado do modelo escolhido - AR1

A matriz de covariância AR(1) é uma escolha adequada para o exemplo da droga e do placebo devido à sua capacidade de capturar a dependência temporal nas respostas dos pacientes em estudos longitudinais. Essa matriz reflete a ideia de que as observações próximas no tempo têm uma correlação mais forte, enquanto as observações mais distantes têm uma correlação mais fraca. Isso é consistente com a possibilidade de que os efeitos da droga persistam ao longo do tempo, mas diminuam com o passar dos dias após a administração.

```
report(model_resp_AR1)
```

```
We fitted a linear mixed model (estimated using REML and nlminb optimizer) to
predict resp with Tempo, drug and Sujeito (formula: resp ~ Tempo + drug + Tempo
* drug). The model included Sujeito as random effect (formula: ~1 | Sujeito).
The model's total explanatory power is substantial (conditional R2 = 0.41) and
the part related to the fixed effects alone (marginal R2) is of 0.41. The
model's intercept, corresponding to Tempo = 1 and drug = New Drug, is at 3.35
(95% CI [3.29, 3.41], t(20) = 125.70, p < .001). Within this model:

  - The effect of Tempo [2] is statistically non-significant and positive (beta =
0.02, 95% CI [-0.04, 0.08], t(20) = 0.59, p = 0.559; Std. beta = 0.21, 95% CI
[-0.52, 0.93])
  - The effect of Tempo [3] is statistically non-significant and negative (beta =
-0.02, 95% CI [-0.09, 0.05], t(20) = -0.49, p = 0.627; Std. beta = -0.21, 95%
CI [-1.07, 0.66])
  - The effect of drug [Placebo] is statistically significant and negative (beta
= -0.12, 95% CI [-0.20, -0.03], t(10) = -3.10, p = 0.011; Std. beta = -1.44,
95% CI [-2.48, -0.40])
  - The effect of Tempo [2] × drug [Placebo] is statistically non-significant and
positive (beta = 8.08e-17, 95% CI [-0.08, 0.08], t(20) = 2.04e-15, p > .999;
Std. beta = -1.43e-15, 95% CI [-1.02, 1.02])
  - The effect of Tempo [3] × drug [Placebo] is statistically non-significant and
positive (beta = 0.03, 95% CI [-0.07, 0.13], t(20) = 0.70, p = 0.493; Std. beta
= 0.41, 95% CI [-0.82, 1.64])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
```

```
computed using a Wald t-distribution approximation.
```

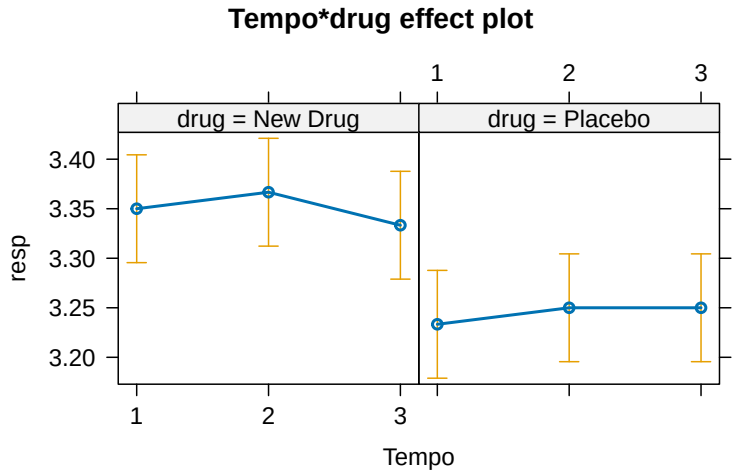
Faça as análises para a variável Pulse! Lembre-se de não ficar apenas copiando e colando os scripts e mude os nomes das variáveis!

## 3.6 Extras!

### 3.6.1 Gráfico do modelo em uma linha

Já vimos algumas formas de apresentar os gráficos dos modelos. A ideia ao longo destes tutoriais é oferecer várias ferramentas para você poder escolher a mais adequada para seus objetivos. Vamos ver uma forma bem prática de criar um gráfico do nosso modelo escolhido com a função `plot()` em conjunto com a função `allEffects()`!

```
plot(allEffects(model_resp_AR1))
```




### 3.6.2 Mudando a referência de um fator

Na análise dos modelos observamos que o grupo "New Drug" foi escolhido como referência. Isso é evidenciado pelo fato de que

apenas os valores dos estimadores para o grupo "Placebo" são apresentados nos resultados. O R escolhe, por padrão, o valor de referência inicial com base na ordem alfabética dos níveis da variável categórica. Nesse caso, "New Drug" é escolhido como referência por ser o primeiro nível alfabeticamente.

Caso queira confirmar qual é o valor de referência de uma variável, basta utiliza a função `levels()`, que já vem instalada com o R.

```
levels(dataset$drug)
```

```
[1] "New Drug" "Placebo"
```

Podemos alterar facilmente qual será o grupo de referência de nossas análises utilizando a função `relevel()`, que também já vem instalada no pacote base do R.

```
dataset$drug <- relevel(dataset$drug, ref = "Placebo")
levels(dataset$drug)
```

```
[1] "Placebo"  "New Drug"
```

Note agora que "Placebo" aparece em primeiro lugar, indicando o novo valor de referência. Escreva um novo modelo e compare os resultados com os anteriores.

## 3.7 Lista 2 resolvida no SPSS

https://youtu.be/gYWld1qSh9c?si=VJdZsrSYAE-2BJpA

## 3.8 Referências

https://bcheggeseth.github.io/CorrelatedData/marginal-models.html

## 3.9 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages effectsize
(version 0.8.6; Ben-Shachar MS et al., 2020), flexplot (version 0.20.5; Fife D,
2024), effects (version 4.2.2; Fox J, Weisberg S, 2019), carData (version
3.0.5; Fox J et al., 2022), mvtnorm (version 1.2.3; Genz A, Bretz F, 2009),
TH.data (version 1.1.2; Hothorn T, 2023), multcomp (version 1.4.25; Hothorn T
et al., 2008), emmeans (version 1.8.8; Lenth R, 2023), parameters (version
0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke D et al.,
2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see (version 0.8.1;
Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019),
bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version
0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al.,
2023), correlation (version 0.8.4; Makowski D et al., 2022), datawizard
(version 0.9.0; Patil I et al., 2022), nlme (version 3.1.163; Pinheiro J et
al., 2023), foreign (version 0.8.85; R Core Team, 2023), survival (version
3.5.7; Therneau T, 2023), MASS (version 7.3.60; Venables WN, Ripley BD, 2002)
and dplyr (version 1.1.3; Wickham H et al., 2023).

References
----------
  - Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of
Effect Size Indices and Standardized Parameters." _Journal of Open Source
Software_, *5*(56), 2815. doi:10.21105/joss.02815
<https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
  - Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_.
R package version 0.20.5.
  - Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, 3rd
edition. Sage, Thousand Oaks CA.
<https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html>. Fox J,
Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression
Models with Predictor Effect Plots and Partial Residuals." _Journal of
Statistical Software_, *87*(9), 1-27. doi:10.18637/jss.v087.i09
<https://doi.org/10.18637/jss.v087.i09>. Fox J (2003). "Effect Displays in R
for Generalised Linear Models." _Journal of Statistical Software_, *8*(15),
1-27. doi:10.18637/jss.v008.i15 <https://doi.org/10.18637/jss.v008.i15>. Fox J,
```

Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." _Journal of Statistical Software_, *32*(1), 1-24. doi:10.18637/jss.v032.i01 <https://doi.org/10.18637/jss.v032.i01>.

- Fox J, Weisberg S, Price B (2022). _carData: Companion to Applied Regression Data Sets_. R package version 3.0-5, <https://CRAN.R-project.org/package=carData>.

- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.

- Hothorn T (2023). _TH.data: TH's Data Archive_. R package version 1.1-2, <https://CRAN.R-project.org/package=TH.data>.

- Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." _Biometrical Journal_, *50*(3), 346-363.

- Lenth R (2023). _emmeans: Estimated Marginal Means, aka Least-Squares Means_. R package version 1.8.8, <https://CRAN.R-project.org/package=emmeans>.

- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.

- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.

- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.

- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.

- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.

- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.

- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_.

<https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
  - Pinheiro J, Bates D, R Core Team (2023). _nlme: Linear and Nonlinear Mixed Effects Models_. R package version 3.1-163, <https://CRAN.R-project.org/package=nlme>. Pinheiro JC, Bates DM (2000). _Mixed-Effects Models in S and S-PLUS_. Springer, New York. doi:10.1007/b98882 <https://doi.org/10.1007/b98882>.
  - R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
  - Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
  - Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.
  - Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.

# 4 Lista 4 - GMM e ICC

Para resolver a lista de exercícios 4 vamos utilizar o banco de dados THKS2. O banco apresenta dados do programa "Television, School and Family Smoking Prevention and Cessation Project (TVSFP)", que avaliou a eficácia de um programa presencial para parar de fumar (Currículo) em conjunto com um programa em vídeo (TV) para prevenir o início do tabagismo e fortalecer a resiliência daqueles que deixaram de fumar.

O estudo adotou um delineamento 2x2 com quatro grupos distintos, considerando a presença do "school-based social-resistance curriculum (CC)" e do "television-based prevention program (TV)". Esses grupos foram categorizados como "Curriculum & TV", "Curriculum", "TV" e "Neither". Este último indicando que as pessoas do grupo não participaram de nenhuma intervenção.

A randomização da amostra ocorreu em dois níveis: por escolas e por salas de aula. O banco de dados inclui informações de 1600 alunos de 135 classes distintas em 28 escolas localizadas em Los Angeles. A variável dependente, a Escala de Conhecimento em Tabaco e Saúde (THKS), foi avaliada antes da randomização e após a implementação dos efeitos de cada grupo.

Com base nestes dados, por favor, apresente as questões específicas e descreva os resultados utilizando as notações apropriadas.

## 4.1 Pacotes que vamos utilizar

```
# Seu código R aqui
library(emmeans)
library(lme4)
library(nlme)
library(flexplot)
library(foreign)
library(dplyr)
library(multcomp)
library(effects)
library(sjstats)
library(tm)
library(report)
library(ggplot2)
library(forcats)
library(performance)
library(rempsyc)
library(easystats)
library(fitdistrplus)
library(sjPlot)
library(kableExtra)
library(psychometric)
library(misty)
```

```
dataset = read.spss("THKS2.sav", to.data.frame=TRUE)
```

Como já mencionado no capítulo anterior, é muito importante averiguar os tipos de variáveis antes de começar as análises. Para isso vamos utilizar a função `glimpse()`.

```
glimpse(dataset)
```

```
Rows: 1,600
Columns: 5
$ SchoolID <dbl> 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 4~
$ ClassID  <dbl> 404101, 404101, 404101, 404101, 404101, 404101, 404101, 40410~
$ PreTHKS  <dbl> 1, 2, 4, 3, 1, 1, 2, 0, 1, 2, 2, 1, 1, 0, 4, 1, 2, 1, 2, 2, 2~
$ PosTHKS  <dbl> 3, 4, 2, 3, 2, 1, 3, 5, 2, 3, 4, 3, 1, 0, 4, 5, 3, 3, 4, 2, 3~
```

```
$ Group    <fct> Curriculum & TV, Curriculum & TV, Curriculum & TV, Curriculum~
```

Ao analisar os arquivos provenientes de outros programas, percebe-se que todas as variáveis numéricas são tratadas como contínuas. No entanto, todas as variáveis no banco de dados são, na verdade, categóricas. Portanto, é necessário modificar o tipo das variáveis antes de iniciar as análises. Para isso vamos utilizar a função `as.factor()` nas 4 variáveis contínuas.

```
dataset$SchoolID = as.factor(dataset$SchoolID)
dataset$ClassID = as.factor(dataset$ClassID)
dataset$PreTHKS = as.integer(dataset$PreTHKS)
dataset$PosTHKS = as.integer(dataset$PosTHKS)
```

Rodando novamente a função `glimpse()` podemos verificar se a mudança aconteceu.

```
glimpse(dataset)
```

```
Rows: 1,600
Columns: 5
$ SchoolID <fct> 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 404, 4~
$ ClassID  <fct> 404101, 404101, 404101, 404101, 404101, 404101, 404101, 40410~
$ PreTHKS  <int> 1, 2, 4, 3, 1, 1, 2, 0, 1, 2, 2, 1, 1, 0, 4, 1, 2, 1, 2, 2, 2~
$ PosTHKS  <int> 3, 4, 2, 3, 2, 1, 3, 5, 2, 3, 4, 3, 1, 0, 4, 5, 3, 3, 4, 2, 3~
$ Group    <fct> Curriculum & TV, Curriculum & TV, Curriculum & TV, Curriculum~
```

Podemos também calcular o número de alunos por classe também. Isso será bem útil para uma análise Extra no fim do capítulo.

```
dataset$Tamanho_Classe <- ave(dataset$PreTHKS, dataset$SchoolID, dataset$ClassID, FUN = leng
```

## 4.2 a) Modelos hierárquicos

> **Exercício**
>
> Com base no desenho apresentado, qual é a pergunta que este estudo quer responder?

Para abordar as questões específicas relacionadas ao banco de dados THKS2 utilizando um modelo linear GMM hierárquico, precisamos formular perguntas específicas que desejamos responder com a análise. Dado que o THKS é a variável dependente e foi avaliado antes e depois da implementação dos diferentes grupos de intervenção, podemos considerar algumas perguntas relevantes:

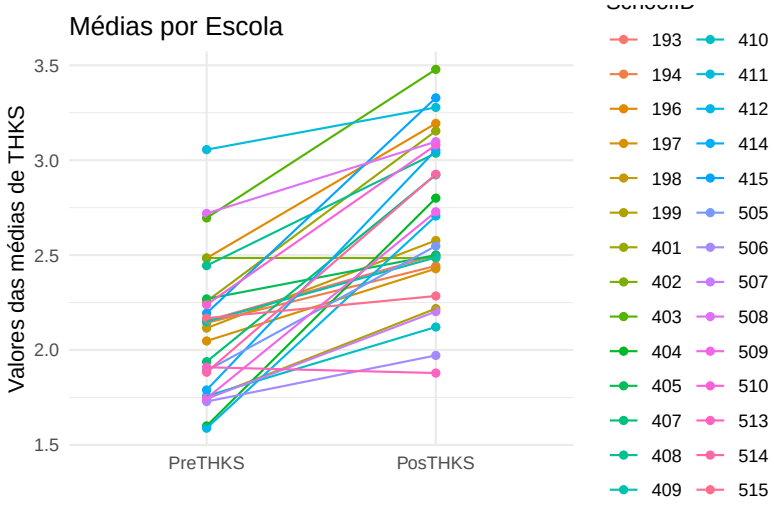
- Efeito geral da intervenção: Como a média da escala THKS varia entre os grupos "Curriculum & TV", "Curriculum", "TV" e "Neither" após a implementação das intervenções?
- Diferenças entre grupos específicos: Há diferenças significativas nas mudanças médias da escala THKS entre os grupos "Curriculum & TV", "Curriculum", "TV" e "Neither"?
- Variação entre escolas e salas de aula: A variação nas médias da escala THKS é significativa entre as escolas ou entre as salas de aula, considerando o efeito das intervenções?

A análise gráfica pode ser fundamental para avaliar a validade da escolha de um modelo hierárquico. Ao comparar as médias do PreTHKS e do PosTHKS para diferentes escolas e salas de aula, os gráficos podem revelar padrões ou tendências que indicam se há variação sistemática ou não nas médias entre esses níveis hierárquicos. A identificação de padrões específicos pode orientar a decisão de usar um modelo hierárquico para capturar a estrutura aninhada dos dados.

Vamos criar um gráfico das médias de THKS entre as escolas antes e depois das intervenções.

### 4.2.1 Média por Escola

```
# Instale o pacote ggplot2 se ainda não o tiver instalado
# install.packages("ggplot2")


# Crie um novo dataframe para armazenar a média das notas por escola
media_por_escola <- aggregate(cbind(PosTHKS, PreTHKS) ~ SchoolID, data = dataset, FUN = mean

# Transforme os dados em formato longo (tidy)
media_por_escola_long <- tidyr::pivot_longer(media_por_escola, cols = c("PreTHKS", "PosTHKS"

# Crie um gráfico de dispersão com uma linha contínua conectando as médias das notas
ggplot(data = media_por_escola_long, aes(x = forcats::fct_rev(tempo), y = media, color = Sch
  geom_point() +
  geom_line() +
  labs(title = "Médias por Escola",
       x = "",
       y = "Valores das médias de THKS") +
  theme_minimal() +
  theme(legend.position = "right")  # Posição da leg
```



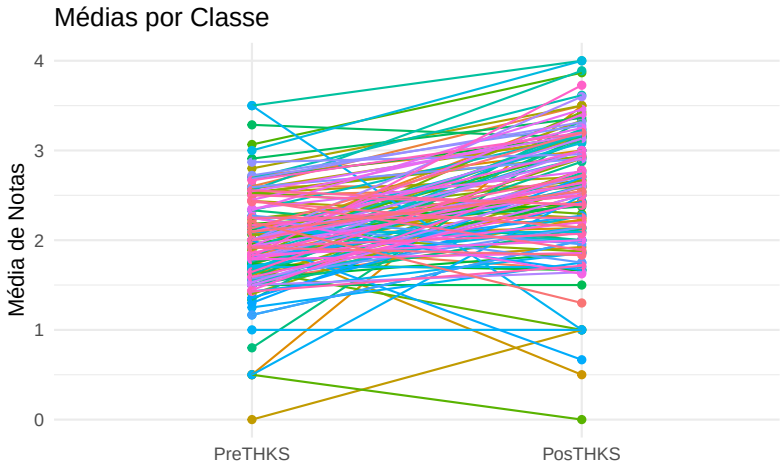

Observe no gráfico que o efeito da intervenção é basicamente constante. Antes da intervenção a média de THKS era menor

e após a intervenção a média aumentou em praticamente todas as escolas analisadas.

Vamos fazer o mesmo mas separando as médias por classes.

### 4.2.2 Média por classe

```
# Instale o pacote ggplot2 se ainda não o tiver instalado
# install.packages("ggplot2")


# Crie um novo dataframe para armazenar a média das notas por escola
media_por_classe <- aggregate(cbind(PreTHKS, PosTHKS) ~ ClassID, data = dataset, FUN = mean)

# Transforme os dados em formato longo (tidy)
media_por_classe_long <- tidyr::pivot_longer(media_por_classe, cols = c("PreTHKS", "PosTHKS"

# Crie um gráfico de dispersão com uma linha contínua conectando as médias das notas
ggplot(data = media_por_classe_long, aes(x = forcats::fct_rev(tempo), y = media, color = Cla
  geom_point() +
  geom_line() +
  labs(title = "Médias por Classe",
       x = "",
       y = "Média de Notas") +
  #scale_color_manual(values = rainbow(length(top_50_escolas))) +  # Ajuste as cores manualm
  theme_minimal() +
  theme(legend.position = "none")  # Posição da legenda
```




Observamos que, em alguns casos, a média de THKS diminui após a intervenção, enquanto em outros ocorre um aumento. A falta de um padrão claro sugere que a classe também desempenha um papel na resposta à intervenção, indicando a necessidade de utilizar modelos hierárquicos com fatores aleatórios nessa variável.

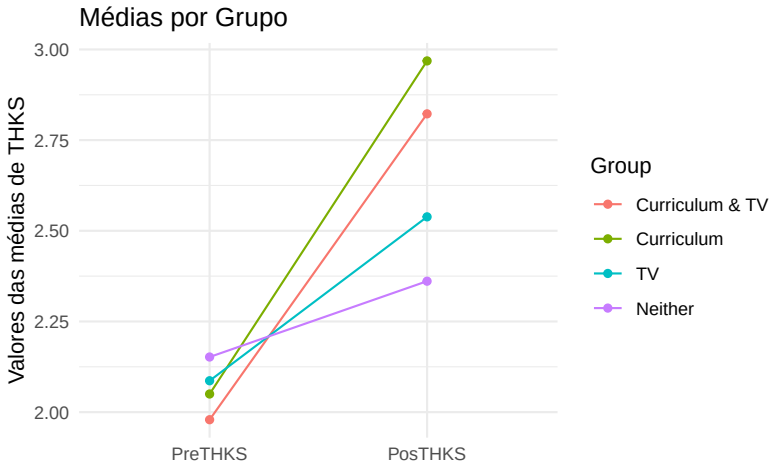
### 4.2.3 Média por Grupo

```
# Instale o pacote ggplot2 se ainda não o tiver instalado
# install.packages("ggplot2")
head(dataset)
```

```
  SchoolID ClassID PreTHKS PosTHKS           Group Tamanho_Classe
1      404  404101       1       3 Curriculum & TV             11
2      404  404101       2       4 Curriculum & TV             11
3      404  404101       4       2 Curriculum & TV             11
4      404  404101       3       3 Curriculum & TV             11
5      404  404101       1       2 Curriculum & TV             11
6      404  404101       1       1 Curriculum & TV             11
```

```
# Crie um novo dataframe para armazenar a média das notas por escola
media_por_grupo <- aggregate(cbind(PosTHKS, PreTHKS) ~ Group, data = dataset, FUN = mean)
```

```r
# Transforme os dados em formato longo (tidy)
media_por_grupo_long <- tidyr::pivot_longer(media_por_grupo, cols = c("PreTHKS", "PosTHKS"),

# Crie um gráfico de dispersão com uma linha contínua conectando as médias das notas
ggplot(data = media_por_grupo_long, aes(x = forcats::fct_rev(tempo), y = media, color = Grou
  geom_point() +
  geom_line() +
  labs(title = "Médias por Grupo",
       x = "",
       y = "Valores das médias de THKS") +
  theme_minimal() +
  theme(legend.position = "right")  # Posição da leg
```




Os grupos também parecem ter um padrão constante de aumento na média de THKS após a intervenção, indicano que não há necessidade de colocar essa variável como efeito aleatório.

## 4.3 b) Efeitos fixos e aleatórios

> **Exercício**
>
> Dentre os efeitos observados – Grupo, Classe e Escola – quais são efeitos fixos e aleatórios pelo menos do ponto de vista teórico?

os gráficos são apenas uma das diversas maneiras de verificar a necessidade ou não de efeitos aleatórios. Mais a frente vamos ver outras métricas que podemos nos ajudar com a decisão.

Com base nos gráficos anteriores, podemos inferir que a classe é um efeito aleatório, enquanto a escola e o grupo são efeitos fixos. Para seguir a abordagem prática exemplificada durante a aula no SPSS, iremos construir vários modelos considerando efeitos fixos e aleatórios. Posteriormente, compararemos os índices de aderência e os resultados obtidos, com o intuito de selecionar o modelo mais adequado para os nossos dados.

## 4.4 c) GLM univariado

> **Exercício**
>
> Faça um GLM univariado tendo o THKS pós como VD e os grupos, escolas e classes como variáveis independentes. Coloque as variáveis como efeitos fixos e aleatórios adequadamente conforme a questão anterior. Descreva os resultados encontrados.

Caso queira repetir o modelo que o Altay apresentou no vídeo, execute o código abaixo. Assim como no SPSS, no R os valores serão calculados por muito tempo e o modelo não vai convergir.

> **Por conta e risco!**

```
modelo1 <- lm(PosTHKS ~ Group * SchoolID *
ClassID * PreTHKS, data=dataset)
```

## 4.5 d) Componentes da variância e ICC

> **Exercício**
>
> Utilizando o "Variance Components", verifique se Classe e Escola podem ser considerados fatores aleatórios. Utilize o ICC (Coeficiente de Correlação Intraclasse) como critério para decidir.

Vamos utilizar a função `lmer()` do pacote `lme4` para criar nossos primeiro modelo com efeitos fixos e aleatórios **(modelo 1)**. Em seguida vamos extrair os componentes da variância dos resultados.

```
modelo_1 = lmer(PosTHKS ~ 1 + Group * PreTHKS + # Efeitos fixos
                  (1|SchoolID:ClassID) + # intercepto aleatório da classe aninhado na escola
                  (1|SchoolID), # intercepto aleatório apenas da escola
                data = dataset,
                REML = TRUE) # Método de estimação dos parâmetros
```

Importante notar em nosso modelo que os efeitos fixos estão fora dos parênteses, ao passo que os efeitos aleatórios (SchooID e ClassID) estão contidos dentro dos parênteses. Essa estrutura informa à função lmer quais variáveis têm efeitos fixos e quais têm efeitos aleatórios.

Quanto os métodos de estimação dos parâmetros, não deixe de ler a seção "Section 4.7"

O primeiro passo para mostrar os componentes da variância é extrair os valores do modelo utilizando a função `VarCorr`. Vamos guardar a saída da função em um data-frame, tornando a visualização mais acessível. Em seguida, utilizaremos os estimadores de variância desejados no cálculo do ICC.

```
var_modelo_1 = as.data.frame(VarCorr(modelo_1))
var_modelo_1
```

```
              grp        var1 var2       vcov     sdcor
1 SchoolID:ClassID (Intercept) <NA> 0.06467071 0.2543044
```

```
2         SchoolID (Intercept) <NA> 0.03844644 0.1960776
3         Residual             <NA> <NA> 1.59946785 1.2647007
```

Os valores que precisamos para calcular o ICC estão na coluna `vcov`.

vamos agora armazenar os valores desejados em outras variáveis.

```
var_classe_1 = var_modelo_1$vcov[1] # classe
var_escola_1 = var_modelo_1$vcov[2] # school
var_erro_1 = var_modelo_1$vcov[3] # total
```

O que fizemos aqui foi acessar o data-frame (comp_var_modelo_1), indicar a coluna que queremos acessar O cifrão (`$vcoc`) e a linha em que se encontra o valor, indicada pelo número dentra das chaves `[]`.

Tudo o que precisamos fazer agora é calcular o ICC, que se dá pela seguinte fórmula:

$$ICC = \frac{\sigma^2_{\text{entre grupos}}}{\sigma^2_{\text{entre grupos}} + \sigma^2_{\text{do erro}}}$$

### 4.5.1 ICC Escola (modelo 1)

Vamos primeiro calcular o ICC da Escola.

```
# ICC Escola
icc_escola_1 = var_escola_1 / (var_escola_1 + var_erro_1)
round(icc_escola_1, 3)
```

```
[1] 0.023
```

Arredondando o valor do cálculo temos que o valor do ICC da escola é de 0,023, ou de aproximadamente 2,3%. Se um valor de 5% fosse estabelecido para considerar uma variabilidade significativa entre os grupos, o ICC de 0,023 seria bastante baixo em relação a esse limiar.

### 4.5.2 ICC Classe (modelo 1)

Para calcular o ICC da classe temos:

```
# ICC Classe
icc_class_1 = var_classe_1 / (var_classe_1 + var_erro_1)
round(icc_class_1, 3)
```

```
[1] 0.039
```

Aqui também temos um valor de ICC abaixo dos 5%, indicando que, por esse critério, a Classe também não deveria ser considerada como um fator aletaório.

Uma manipulação viável para avaliar o ICC exclusivamente a partir das variáveis que você considera como aleatórias é incluir apenas essas variáveis no modelo, excluindo todas as outras que tenham efeito fixo. Vamos refazer todos os passos anteriores, apenas mudando o modelo **(modelo 2)**.

```
modelo_2 = lmer(PosTHKS ~ 1 +
                  (1|SchoolID:ClassID) +
                  (1|SchoolID),
                data = dataset,
                REML = TRUE) # Método de estimação dos parâmetros
```

```
var_modelo_2 = as.data.frame(VarCorr(modelo_2))
var_modelo_2
```

```
              grp        var1 var2       vcov     sdcor
1 SchoolID:ClassID (Intercept) <NA> 0.08497895 0.2915115
2         SchoolID (Intercept) <NA> 0.11659673 0.3414626
3         Residual        <NA> <NA> 1.72359029 1.3128558
```

```
var_classe_2 = var_modelo_2$vcov[1] # classe
var_escola_2 = var_modelo_2$vcov[2] # escola
var_erro_2 = var_modelo_2$vcov[3] # total
```

### 4.5.3 ICC Escola (modelo 2)

Calculando o ICC da escola para o modelo 2 temos

```
# ICC escola

icc_school_2 = var_escola_2 / (var_escola_2 + var_erro_2)
round(icc_school_2, 3)
```

```
[1] 0.063
```

Agora temos que o ICC da escola é maior que 5%, indicando que a variável é uma boa candidata para ser designada tendo efeito aleatório.

### 4.5.4 ICC Classe (modelo 2)

```
# ICC classe

icc_class_2 = var_classe_2 / (var_classe_2 + var_erro_2)
round(icc_class_2, 3)
```

```
[1] 0.047
```

Já a classe continua com um valor de ICC abaixo dos 5%

### 4.5.5 ICC com função

Podemos utilizar a função `multilevel.icc` do pacote `misty` para não precisar calcular na mão o ICC. Digno de nota que a função não aceita efeitos fixos, portanto teremos **APENAS** o ICC do modelo com efeitos aleatórios. Além disso a função pode assumir 3 tipos:

- ICC(1) - Mostra quanto da variação ocorre entre os grupos (nível 2) e entre os grupos de grupos (nível 3), que é semelhante ao que calculamos na mão.

```
multilevel.icc(PosTHKS, # Variável dependente
               data = dataset, # Banco de dados
               cluster = c("SchoolID", "ClassID")) # Ordem dos clusters importa. Primeiro o
```

```
      L3        L2
0.0605645 0.0441411
```

- ICC(1b) - Representa a correlação esperada entre dois elementos escolhidos aleatoriamente no mesmo grupo.

```
multilevel.icc(PosTHKS,
               data = dataset,
               cluster = c("SchoolID", "ClassID"),
               type = "1b")
```

```
      L3        L2
0.0605645 0.1607378
```

- ICC(2) Indica quão confiáveis são as médias dos grupos (nível 2 e 3). Ou seja, o quão representativas são as médias dos grupos em relação às diferenças individuais dentro desses grupos.

```
multilevel.icc(PosTHKS, data = dataset, cluster = c("SchoolID", "ClassID"),
type = "2")
```

```
      L3        L2
0.7092913 0.3688212
```

Notem que a primeira fórmula apresenta resultado similar ao que calculamos na mão.

> **!** Importante!
>
> Não existe um conceito fechado de como definir se uma variável deve ser considerada ou não como efeito aleatório. A teoria deve sempre prevalecer sobre os demais critérios.

Pelo critério teórico, vamos assumir que tanto escola quanto classe terão efeito aleatório em nosso modelo final.

## 4.6 e) Interpretando os resultados

> **Exercício**
>
> Realize um Modelo Misto Hierárquico (caso os fatores aleatórios sejam relevantes com base em d). Descreva os resultados adequadamente e verifique qual combinação de fatores aleatórios é a mais adequada para explicar a variação dos resultados do THKS (com base no ICC).

### 4.6.1 Verificando a referência do Grupo

Para seguir os passos do vídeo feito pelo Altay no SPSS primeiro temos que ajustar o nível de referência da variável Grupo. No SPSS a referência é o grupo que não fez nada (Neither). Para verificar qual o nível de referência aqui no R vamos utilizar a função `levels()`.

```
levels(dataset$Group)
```

```
[1] "Curriculum & TV" "Curriculum"      "TV"              "Neither"
```

O nível de referência é sempre primeiro que aparece na lista, no caso "Curriculum & TV".

Vamos mudar para que a referência seja "Neither", utilizando a função relevel.

```
dataset$Group <- relevel(dataset$Group, ref = "Neither")
levels(dataset$Group)
```

```
[1] "Neither"         "Curriculum & TV" "Curriculum"      "TV"
```

Agora sim podemos seguir com nossa análise.

### 4.6.2 Criando o modelo

Ao contrário do SPSS, não enfrentaremos problemas de convergência em nossos modelos se a matriz de covariância não for modificada. Para demonstrar que alterar a matriz de covariância não afeta significativamente os coeficientes, podemos criar dois modelos para verificação:

- a) modelo com matriz de covariância simétrica;
- b) modelo com matriz de covariância diagonal (padrão caso não definamos explicitamente a matriz).

A função `lmer()` não oferece uma maneira direta de modificar a matriz de covariância. Portanto, da mesma forma que fizemos na Lista de Exercícios 3, vamos utilizar a função `lme()`.

```
# Modelo a)
modelo_a = lme(
  fixed = PosTHKS ~ 1 + PreTHKS + Group,
  random =~ 1|SchoolID/ClassID,
  correlation = corCompSymm(form = ~1|SchoolID/ClassID), # Aqui definimos a matriz simétrica
  data = dataset,
  method = "REML")

# Armazenando os valores dos coeficientes do modelo a) em uma variável
coef_a = modelo_a$coefficients$fixed

# Modelo b)
modelo_b = lme(
  fixed = PosTHKS ~ 1 + PreTHKS + Group,
  random =~ 1|SchoolID/ClassID,
  data = dataset,
  method = "REML") # Matriz diagonal por padrão

# Armazenando os valores dos coeficientes do modelo b) em uma variável
coef_b = modelo_b$coefficients$fixed


# Criar um dataframe
df_coeficientes <- data.frame(Modelo_a = coef_a,
```

```
                                Modelo_b = coef_b)
df_coeficientes
```

```
                      Modelo_a  Modelo_b
(Intercept)          1.7019847 1.7019852
PreTHKS              0.3053629 0.3053628
GroupCurriculum & TV 0.4924670 0.4924662
GroupCurriculum      0.6413248 0.6413260
GroupTV              0.1820783 0.1820802
```

Podemos observar que os valores mudam apenas depois da 3 casa após a vírgula. Portanto podemos construir os modelos sem alterar a matriz de covariância neste caso específico.

Vamos ao modelo:

```
modelo_3 = lme(
  fixed = PosTHKS ~ 1 + PreTHKS + Group,
  random =~ 1|SchoolID/ClassID,
  data = dataset,
  method = "REML")

# Escolhi utilizar o lme() por ele apresentar mais resultados na saída da função anova()
```

### 4.6.3 ICC do modelo

Não encontramos uma maneira fácil de mostrar o ICC para modelos de 3 níveis com variáveis independentes fixas. Por isso mostramos como calcular na mão o ICC anteriormente. Podemos acessar os valores de variância do modelo com a seguinte função:

```
kable(VarCorr(modelo_3)) # O kable é só pra deixar com um visual melhor a saída.
```

| | Variance | StdDev |
|---|---|---|
| SchoolID = | pdLogChol(1) | |
| (Intercept) | 0.03864002 | 0.1965706 |
| ClassID = | pdLogChol(1) | |
| (Intercept) | 0.06466151 | 0.2542863 |
| Residual | 1.60229394 | 1.2658175 |

Agora queremos acessar cada variância separadamente. Para isso executamos o scritp a seguir.

```
var_escola = VarCorr(modelo_3)[2] # Variancia da Escola

var_classe = VarCorr(modelo_3)[4] # Variancia da Classe

var_res = VarCorr(modelo_3)[5] # Variancia do resíduo
```

Se você tentar fazer contas com essas variáveis vai notar algo bem estranho

```
var_classe + var_res
```

Isso acontece porque elas saíram como caracteres (símbolos, letras...) e não como números!

```
typeof(var_classe)
```

```
[1] "character"
```

Vamos resolver isso transformando elas para números

```
var_escola = as.numeric(var_escola)
var_classe = as.numeric(var_classe)
var_res = as.numeric(var_res)
```

Agora sim!

```
typeof(var_escola)
```

```
[1] "double"
```

Calculando o ICC da Escola temos:

```
var_escola/(var_escola+var_res) #ICC da escola
```

```
[1] 0.02354758
```

ICC da classe:

```
var_classe/(var_classe+var_res)
```

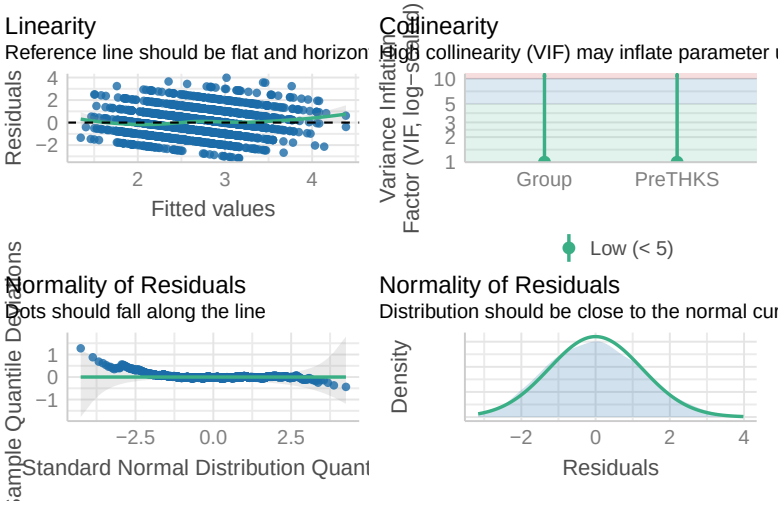
```
[1] 0.03879018
```

### 4.6.4 Pressupostos do modelo

Como já vimos, parte importante de analisar os modelos é verificar os pressupostos. Não entraremos em detalhes, vamos apenas vislumbrar quando todos os pressupostos são atendidos! O melhor de tudo, usando apenas 3 palavras na linha de código, graças à função `check_model()`.

```
check_model(modelo_3)
```

```
Converting missing values (`NA`) into regular values currently not
  possible for variables of class `NULL`.
```




Que beleza, não? Resíduos normais, baixa colinearidade e ótima linearidade do modelo! Podemos interpretar os resultados com tranquilidade!

### 4.6.5 Resultados

Vamos verificar se o efeito do grupo é significante, que é a principal variável dependente do nosso modelo. Para isso podemos utilizar a função `anova()` que é muito versátil para diversas ocasiões.

```
kable(anova(modelo_3)) #função kable apenas para deixar mais bonita a tabela
```

| | numDF | denDF | F-value | p-value |
|---|---|---|---|---|
| (Intercept) | 1 | 1464 | 2240.036555 | 0.0000000 |
| PreTHKS | 1 | 1464 | 136.795261 | 0.0000000 |
| Group | 3 | 24 | 6.610438 | 0.0020567 |

Boa! Descobrimos que o efeito do grupo é significativo. Agora precisamos saber entre quais grupos está a diferença e de quanto ela é.

Para tanto vamos utilizar mais uma vez a função `summary()`.

```
summary(modelo_3)
```

```
Linear mixed-effects model fit by REML
  Data: dataset
       AIC      BIC    logLik
  5389.335 5432.332 -2686.668

Random effects:
 Formula: ~1 | SchoolID
        (Intercept)
StdDev:   0.1965706

 Formula: ~1 | ClassID %in% SchoolID
        (Intercept) Residual
StdDev:   0.2542863 1.265817

Fixed effects:  PosTHKS ~ 1 + PreTHKS + Group
                         Value  Std.Error   DF   t-value p-value
(Intercept)          1.7019852 0.12543004 1464 13.569199  0.0000
PreTHKS              0.3053628 0.02589132 1464 11.794021  0.0000
GroupCurriculum & TV 0.4924662 0.15864165   24  3.104268  0.0048
GroupCurriculum      0.6413260 0.16094729   24  3.984696  0.0005
GroupTV              0.1820802 0.15724054   24  1.157972  0.2583
 Correlation:
                     (Intr) PrTHKS GrC&TV GrpCrr
PreTHKS              -0.442
GroupCurriculum & TV -0.649  0.029
GroupCurriculum      -0.634  0.015  0.496
GroupTV              -0.645  0.008  0.508  0.501

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.49874557 -0.69757194 -0.01721254  0.68240735  3.14602049

Number of Observations: 1600
Number of Groups:
             SchoolID ClassID %in% SchoolID
                   28                   135
```

Como vocês já podem ter percebido as saídas da função `summary()` no R não geram as saídas mais fáceis de interpretar, como podemos ver no exemplo abaixo.

Agora que você enfrentou a busca nos detalhes desse fascinante output gerado pela função summary, é com satisfação que compartilhamos a boa notícia de que muitos desenvolvedores compartilham da sua experiência e criaram vários pacotes para aprimorar a visualização dos resultados. Ao longo dos exercícios, apresentaremos algumas abordagens para alcançar isso. No final da seção de modelos lineares, você encontrará um glossário que ajudará na geração de outputs mais amigáveis e formatados para publicações acadêmicas.

Por hora, vamos compartilhar uma abordagem mais "na mão" para melhorar a visualização dos resultados, para caso algum pacote não atenda completamente às suas necessidades.

```
# Resumo do modelo

resumo_modelo <- summary(modelo_3)

# Extração de estimadores, intervalos de confiança e p-valores

coeficientes <- resumo_modelo$coefficients$fixed # essa linha varia muito dependendo do mode
intervalos_confianca <- intervals(modelo_3, which = "fixed") # Função `confint()` pode ser u
p_valores <- resumo_modelo$tTable[, "p-value"]

# Criar um data frame

resultados_modelo <- data.frame(
  Estimador = round(coeficientes, 3),
  IC_Inf = round(intervalos_confianca$fixed[, 1], 3),
  IC_Sup = round(intervalos_confianca$fixed[, 2], 3),
  p = round(p_valores, 3)
)

# Apresentando os resultados
kable(resultados_modelo)
```

| | Estimador | IC_Inf | IC_Sup | p |
|---|---|---|---|---|
| (Intercept) | 1.702 | 1.456 | 1.702 | 0.000 |
| PreTHKS | 0.305 | 0.255 | 0.305 | 0.000 |
| GroupCurriculum & TV | 0.492 | 0.165 | 0.492 | 0.005 |
| GroupCurriculum | 0.641 | 0.309 | 0.641 | 0.001 |
| GroupTV | 0.182 | -0.142 | 0.182 | 0.258 |

---

Melhorou um pouco né? Achou muito trabalhoso??

Que tal fazer tudo em uma linha de código e ainda com correção de Bonferroni!?

```
emmeans(modelo_3, pairwise ~ Group, adjust = "bonferroni") # por padrão temos a correção de
```

```
$emmeans
 Group           emmean    SE df lower.CL upper.CL
 Neither           2.33 0.113 27     2.10     2.56
 Curriculum & TV   2.83 0.112 24     2.60     3.06
 Curriculum        2.98 0.115 24     2.74     3.21
 TV                2.52 0.110 24     2.29     2.74

Degrees-of-freedom method: containment
Confidence level used: 0.95

$contrasts
 contrast                     estimate    SE df t.ratio p.value
 Neither - Curriculum & TV      -0.492 0.159 24  -3.104  0.0290
 Neither - Curriculum           -0.641 0.161 24  -3.985  0.0033
 Neither - TV                   -0.182 0.157 24  -1.158  1.0000
 Curriculum & TV - Curriculum   -0.149 0.160 24  -0.928  1.0000
 Curriculum & TV - TV            0.310 0.157 24   1.981  0.3550
 Curriculum - TV                 0.459 0.159 24   2.888  0.0485

Degrees-of-freedom method: containment
P value adjustment: bonferroni method for 6 tests
```

Os valores estão negativos porque ajustamos o nível de referência da variável Group para "Neither". No resultado temos

que “Curriculum” apresenta a maior média geral. Logo seria interessante deixá-lo como variável de referência, caso queira que seus estimadores fiquem positivo. Já vimos como fazer isso anteriormente!

Tente modificar a referência para “Curriculum”, mas **CUIDADO!** Não se esqueça de criar o modelo novamente, caso contrário os resultados ficarão errados!

Para acessar apenas os resultados de contraste podemos fazer o seguinte:

```
emmeans(modelo_3, pairwise ~ Group, adjust = "bonferroni")$contrasts
```

```
contrast                      estimate    SE df t.ratio p.value
Neither - Curriculum & TV       -0.492 0.159 24  -3.104  0.0290
Neither - Curriculum            -0.641 0.161 24  -3.985  0.0033
Neither - TV                    -0.182 0.157 24  -1.158  1.0000
Curriculum & TV - Curriculum    -0.149 0.160 24  -0.928  1.0000
Curriculum & TV - TV             0.310 0.157 24   1.981  0.3550
Curriculum - TV                  0.459 0.159 24   2.888  0.0485

Degrees-of-freedom method: containment
P value adjustment: bonferroni method for 6 tests
```

## 4.7 Extras!

### 4.7.1 Métodos de estimação dos parâmetros do modelo

O REML (Residual Maximum Likelihood) e o ML (Maximum Likelihood) são duas abordagens distintas para a estimação de parâmetros em modelos de regressão linear mista (ou modelos hierárquicos). Ambas são baseadas no método da máxima verossimilhança, mas diferem na maneira como tratam os graus de liberdade.

**Maximum Likelihood (ML):**

Na abordagem ML, o foco é maximizar a verossimilhança do modelo, considerando tanto os efeitos fixos quanto os efeitos

aleatórios. O ML leva em conta todos os parâmetros do modelo para maximizar a probabilidade de observar os dados dados os parâmetros. É mais adequado quando o interesse principal é fazer inferências sobre os parâmetros fixos do modelo.

**Residual Maximum Likelihood (REML):**

A abordagem REML é uma variação do ML que remove os efeitos fixos do modelo antes de calcular a verossimilhança. O REML estima a verossimilhança condicional dos efeitos aleatórios, removendo a contribuição dos efeitos fixos. Ele tende a ser mais eficiente na estimação dos efeitos aleatórios, especialmente em amostras pequenas, e fornece estimativas menos enviesadas para a variância dos efeitos aleatórios. O REML é frequentemente preferido quando o foco está na estimação dos parâmetros aleatórios e quando a inferência sobre os parâmetros fixos não é o objetivo principal.

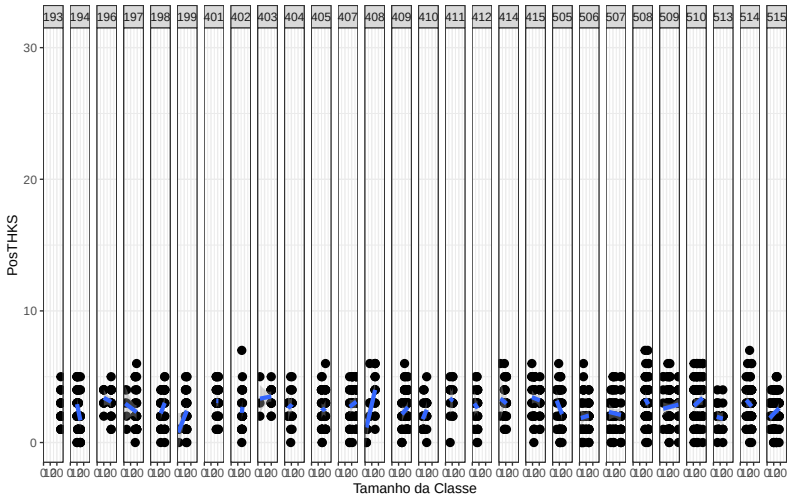
### 4.7.2 Extraindo valores de summary

Podemos extrair diversos valores individualmente da função `summary()`.

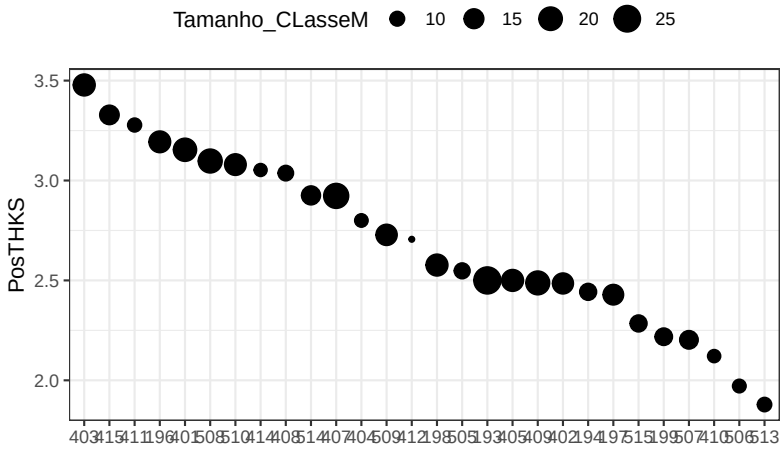
### 4.7.3 Tamanho da classe importa?

```
theme_set(theme_bw(base_size = 7, base_family = ""))

ggplot(data = dataset, aes(x = Tamanho_Classe, y=PosTHKS))+
  facet_grid(~SchoolID)+
  coord_cartesian(ylim=c(0,30))+
  geom_point()+
  geom_smooth(method = "lm", se = TRUE)+
  xlab("Tamanho da Classe")+ylab("PosTHKS")+
  theme(legend.position = "top")
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
Plot.Means<-dataset %>% group_by(SchoolID) %>%
  dplyr::summarize(PosTHKSM = mean(PosTHKS, na.rm=TRUE),
                   Tamanho_CLasseM=mean(Tamanho_Classe, na.rm=TRUE))



ggplot(data = Plot.Means, aes(x = reorder(SchoolID, -PosTHKSM), y=PosTHKSM))+
  geom_point(aes(size = Tamanho_CLasseM))+
  xlab("")+ylab("PosTHKS")+
  theme_bw()+
  theme(legend.position = "top")
```




### 4.7.4 Comparando modelos

Podemos comparar diversos modelos utilizando a função `model.comparison()` do pacote `flexplot`.

```
model.comparison(modelo_1, modelo_2)
```

```
refitting model(s) with ML (instead of REML)

$statistics
              aic      bic bayes.factor      p
modelo_1 5400.422 5459.578 2.090546e+16 <2e-16
modelo_2 5513.224 5534.735 0.000000e+00

$predicted_differences
   0%   25%   50%   75%  100%
0.001 0.113 0.243 0.436 1.310

$r_squared_change
   Residual (Intercept) (Intercept)
 0.07201389  0.67026142  0.23897964
```

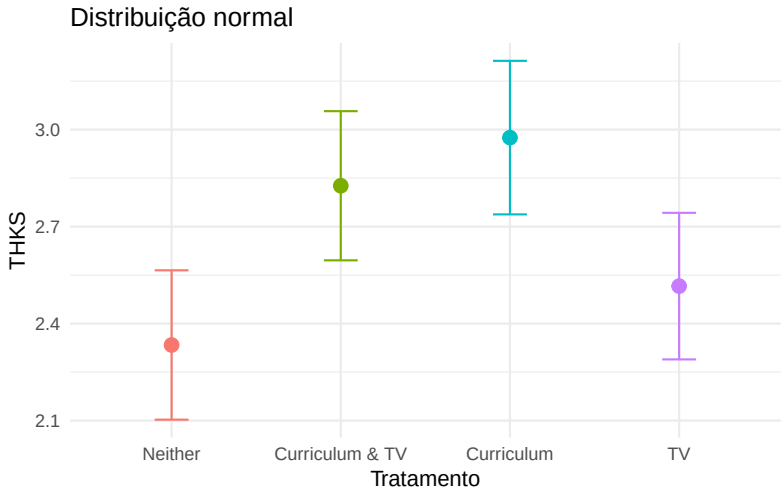
```
# O modelo 1 apresenta melhores resultados
```

### 4.7.5 Plot do modelo

Criando um gráfico com os coeficientes gerados pelo modelo.

```
# Com a correção de Bonferroni
results_modelo_3 = emmeans(modelo_3, pairwise ~ Group, adjust = "bonferroni")
```

```
ggplot(as.data.frame(results_modelo_3$emmeans), aes(x = Group, y = emmean, color = Group)) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.2, position = position_dodg
  geom_point(position = position_dodge(0.8), size = 3) +
  labs(title = "Distribuição normal",
       x = "Tratamento",
       y = "THKS") +
  theme_minimal() +
  theme(legend.position = "none")
```




### 4.7.6 Função para calcular o ICC

Caso você queira calcular o ICC para diferentes modelos de 3 níveis, sugiro criar uma função que faça o trabalho repetitivo ao invés de ficar calculando tudo sempre na mão.

> **!** Importante
>
> Funciona apenas para modelos gerados pela função `lme()`.

```
# Criando minha própria função

icc_lme_3nv = function(modelo) {
  # Extração da variância entre grupos e total
  var_escola = as.numeric(VarCorr(modelo)[2])
  var_classe = as.numeric(VarCorr(modelo)[4])
  var_total = as.numeric(VarCorr(modelo)[5])

  # Cálculo do ICC
  icc_escola = var_escola/(var_escola+var_res)
  icc_classe = var_classe/(var_classe+var_res)

  # Retorna o valor do ICC
  return(list("ICC-Escola" = icc_escola, "ICC-Classe" = icc_classe))
}

# Uso
# icc_lme_3nv(modelo) - basta substitui "modelo" pelo nome da variável que você escolheu par
```

## 4.8 Observações

Treine criar mais modelos multinível, inclusive com apenas 2 níveis. Inclusive, se for utilzar a função `lmer()`, MUITO CUIDADO!

Este modelo:

```
modelo_5b = lmer(PosTHKS ~ 1 + Group +
                   (1|SchoolID:ClassID),
                 dataset,
                 REML = TRUE)
```

É diferente deste modelo:

```
modelo_5b = lmer(PosTHKS ~ 1 + Group +
                   (1|SchoolID) + # Escola como efeito aleatório
                   (1|SchoolID:ClassID), # Classe como efeito aleatório
                 dataset,
                 REML = TRUE)
```

Com a função `lmer()` precisamos indicar no modelo que queremos Escola e Classe como efeito aleatórios em linhas separadas!

## 4.9 Lista 4 resolvida no SPSS

https://www.youtube.com/watch?v=_1lUvEu8M9c

## 4.10 Referências


https://lmudge13.github.io/sample_code/mixed_effects.html
# Tabelas e gráficos de modelos lme

https://rpsychologist.com/r-guide-longitudinal-lme-lmer#three-level-models

https://search.r-project.org/CRAN/refmans/misty/html/multilevel.icc.html

https://www.rdocumentation.org/packages/psychometric/versions/2.4/topics/ICC.lme

https://www.alexanderdemos.org/Mixed5.html

https://cran.r-project.org/web/packages/rempsyc/vignettes/assumptions.html#categorical-predictors # pressupostos dos modelos com variáveis categóricas como preditoras.


## 4.11 Versões dos pacotes

```
report(sessionInfo())
```

Analyses were conducted using the R Statistical language (version 4.3.1; R Core Team, 2023) on Windows 11 x64 (build 22621), using the packages lme4 (version 1.1.34; Bates D et al., 2015), Matrix (version 1.6.0; Bates D et al., 2023), effectsize (version 0.8.6; Ben-Shachar MS et al., 2020), multilevel (version 2.7; Bliese P, 2022), fitdistrplus (version 1.1.11; Delignette-Muller ML, Dutang C, 2015), tm (version 0.7.11; Feinerer I, Hornik K, 2023), flexplot (version 0.20.5; Fife D, 2024), psychometric (version 2.4; Fletcher TD, 2023), effects (version 4.2.2; Fox J, Weisberg S, 2019), carData (version 3.0.5; Fox J et al., 2022), mvtnorm (version 1.2.3; Genz A, Bretz F, 2009), NLP (version 0.2.1; Hornik K, 2020), TH.data (version 1.1.2; Hothorn T, 2023), multcomp (version 1.4.25; Hothorn T et al., 2008), emmeans (version 1.8.8; Lenth R, 2023), sjstats (version 0.18.2; Lüdecke D, 2022), sjPlot (version 2.8.15; Lüdecke D, 2023), parameters (version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), datawizard (version 0.9.0; Patil I et al., 2022), nlme (version 3.1.163; Pinheiro J et al., 2023), foreign (version 0.8.85; R Core Team, 2023), rempsyc (version 0.1.6; Thériault R, 2023), survival (version 3.5.7; Therneau T, 2023), MASS (version 7.3.60; Venables WN, Ripley BD, 2002), ggplot2 (version 3.4.4; Wickham H, 2016), forcats (version 1.0.0; Wickham H, 2023), dplyr (version 1.1.3; Wickham H et al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), misty (version 0.6.1; Yanagida T, 2024) and kableExtra (version 1.3.4; Zhu H, 2021).

References
---


- Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." _Journal of Statistical Software_, *67*(1), 1-48. doi:10.18637/jss.v067.i01 <https://doi.org/10.18637/jss.v067.i01>.
- Bates D, Maechler M, Jagan M (2023). _Matrix: Sparse and Dense Matrix Classes and Methods_. R package version 1.6-0, <https://CRAN.R-project.org/package=Matrix>.
- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Bliese P (2022). _multilevel: Multilevel Functions_. R package version 2.7, <https://CRAN.R-project.org/package=multilevel>.

- Delignette-Muller ML, Dutang C (2015). "fitdistrplus: An R Package for Fitting Distributions." _Journal of Statistical Software_, *64*(4), 1-34. doi:10.18637/jss.v064.i04 <https://doi.org/10.18637/jss.v064.i04>.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Fletcher TD (2023). _psychometric: Applied Psychometric Theory_. R package version 2.4, <https://CRAN.R-project.org/package=psychometric>.
- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, 3rd edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html>. Fox J, Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression Models with Predictor Effect Plots and Partial Residuals." _Journal of Statistical Software_, *87*(9), 1-27. doi:10.18637/jss.v087.i09 <https://doi.org/10.18637/jss.v087.i09>. Fox J (2003). "Effect Displays in R for Generalised Linear Models." _Journal of Statistical Software_, *8*(15), 1-27. doi:10.18637/jss.v008.i15 <https://doi.org/10.18637/jss.v008.i15>. Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." _Journal of Statistical Software_, *32*(1), 1-24. doi:10.18637/jss.v032.i01 <https://doi.org/10.18637/jss.v032.i01>.
- Fox J, Weisberg S, Price B (2022). _carData: Companion to Applied Regression Data Sets_. R package version 3.0-5, <https://CRAN.R-project.org/package=carData>.
- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.
- Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.
- Hothorn T (2023). _TH.data: TH's Data Archive_. R package version 1.1-2, <https://CRAN.R-project.org/package=TH.data>.
- Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." _Biometrical Journal_, *50*(3), 346-363.
- Lenth R (2023). _emmeans: Estimated Marginal Means, aka Least-Squares Means_. R package version 1.8.8, <https://CRAN.R-project.org/package=emmeans>.
- Lüdecke D (2022). _sjstats: Statistical Functions for Regression Models (Version 0.18.2)_. doi:10.5281/zenodo.1284472

<https://doi.org/10.5281/zenodo.1284472>,
<https://CRAN.R-project.org/package=sjstats>.
  - Lüdecke D (2023). _sjPlot: Data Visualization for Statistics in Social Science_. R package version 2.8.15,
<https://CRAN.R-project.org/package=sjPlot>.
  - Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445
<https://doi.org/10.21105/joss.02445>.
  - Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
  - Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
  - Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393
<https://doi.org/10.21105/joss.03393>.
  - Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
  - Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>,
<https://joss.theoj.org/papers/10.21105/joss.01541>.
  - Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_.
<https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_.
<https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3,
<https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306
<https://doi.org/10.21105/joss.02306>,

<https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
  - Pinheiro J, Bates D, R Core Team (2023). _nlme: Linear and Nonlinear Mixed Effects Models_. R package version 3.1-163, <https://CRAN.R-project.org/package=nlme>. Pinheiro JC, Bates DM (2000). _Mixed-Effects Models in S and S-PLUS_. Springer, New York. doi:10.1007/b98882 <https://doi.org/10.1007/b98882>.
  - R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
  - Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
  - Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
  - Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.
  - Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
  - Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
  - Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
  - Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
  - Yanagida T (2024). _misty: Miscellaneous Functions 'T. Yanagida'_. R package version 0.6.1, <https://CRAN.R-project.org/package=misty>.
  - Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4,

<https://CRAN.R-project.org/package=kableExtra>.

# 5 Lista 5 - Generalized Linear Model Aula Prática

Esta é uma lista focada em GLzM independente. Vamos realizar testes baseados no banco de dados Dados Amostra.

```
library(emmeans)
library(tidyverse)
library(lme4)
library(nlme)
library(flexplot)
library(foreign)
library(dplyr)
library(multcomp)
library(effects)
library(sjstats)
library(sjPlot)
library(tm)
library(report)
library(ggplot2)
library(forcats)
library(performance)
library(rempsyc)
library(easystats)
library(kableExtra)
library(fitdistrplus)
library(AER)
library(gtsummary)
library(broom)
```

## 5.1 Carregando os dados e modificando o tipo de variável

```
original = read.spss("Dados Amostra.sav", to.data.frame=TRUE)
```

## 5.2 Boas práticas

Ter um clone do banco de dados e manter ele no formato original. Podemos ir comparando todas as mudanças de maneira ágil. Vamos também já verificar os tipos de variáveis que temos no banco de dados e realizar mudanças, caso necessário.

```
db = original
kable(head(db))
```

| id | childs | age | educ | sex | life | tvhours | attsprts | tempo_obs | aderencia |
|---|---|---|---|---|---|---|---|---|---|
| 188 | 2 | 76 | 14 | Female | Routine | 5 | NA | 123 | SIM |
| 730 | 2 | 48 | 12 | Female | Routine | 1 | NA | 424 | SIM |
| 855 | 0 | 19 | 13 | Male | Exciting | 1 | NA | 124 | SIM |
| 866 | 0 | 38 | 16 | Female | Exciting | NA | NA | 500 | SIM |
| 1165 | 3 | 54 | 16 | Female | NA | 3 | NA | 500 | SIM |
| 1225 | 2 | 53 | NA | Female | Dull | 4 | NA | 500 | SIM |

Não queremos que o número de filhos (childs), idade (age), nível escolaridade (educ) e horas de TV (tvhours) sejam categóricas. Vamos alterar para que sejam numéricas com a função `as.numeric()`.

```
# Modificando o tipo das variáveis. Apenas Life, Sex, attsprts e aderencia devem ser categór
db$childs = as.integer(db$childs)
db$age = as.numeric(db$age)
db$educ = as.numeric(db$educ)
db$tvhours = as.numeric(db$tvhours)
```

Observando como elas estão agora, podemos também utilizar a função `glimpse()`.

```r
glimpse(db)
```

```
Rows: 1,500
Columns: 10
$ id        <dbl> 188, 730, 855, 866, 1165, 1225, 1294, 1339, 1343, 168, 1390,~
$ childs    <int> 3, 3, 1, 1, 4, 3, 4, NA, NA, 1, 3, 2, 1, 1, 1, 3, 1, 2, 5, 4~
$ age       <dbl> 59, 31, 2, 21, 37, 36, 48, 35, 55, 65, 38, 26, 27, 26, 28, 6~
$ educ      <dbl> 13, 11, 12, 15, 15, NA, 11, 11, 8, 14, 11, 10, 15, 15, 14, 1~
$ sex       <fct> Female, Female, Male, Female, Female, Female, Male, Female, ~
$ life      <fct> Routine, Routine, Exciting, Exciting, NA, Dull, NA, NA, NA, ~
$ tvhours   <dbl> 6, 2, 2, NA, 4, 5, 2, NA, NA, 3, 3, 5, 6, 3, 5, 5, 5, 4, 7, ~
$ attsprts  <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, No, No, No, No, ~
$ tempo_obs <dbl> 123.0000, 424.0000, 124.0000, 500.0000, 500.0000, 500.0000, ~
$ aderencia <fct> SIM, SIM, SIM, SIM, SIM, SIM, SIM, SIM, SIM, SIM, SIM, SIM, ~
```

## 5.3 Verificando a representatividade dos dados

```r
xtabs(~ attsprts + sex, data = db)
```

```
        sex
attsprts Male Female
     Yes  384    407
     No   254    444
```

```r
xtabs(~ attsprts + life, data = db)
```

```
        life
attsprts Dull Routine Exciting
     Yes   16     226      281
     No    48     230      190
```

```r
xtabs(~ attsprts + aderencia , data = db)
```

```
        aderencia
attsprts Não SIM
     Yes 750  41
     No    0 698
```

## 5.4 a) e b) GLzM - Praticar ou não esportes

> **Exercício**
>
> Verifique qual o efeito do sexo, o que as pessoas acham da vida (life), número de filhos, idade, anos de escolaridade e horas de tv sobre o fato dela praticar ou não esportes (Attsports). Faça um GLzM e descreva os resultados adequadamente.

Verificando quais os níveis de referência.

```
levels(db$attsprts)
```

```
[1] "Yes" "No"
```

```
levels(db$sex)
```

```
[1] "Male"   "Female"
```

```
levels(db$life)
```

```
[1] "Dull"     "Routine"  "Exciting"
```

```
levels(db$aderencia)
```

```
[1] "Não" "SIM"
```

Para os resultados ficarem similares aos da aula prática, vamos modificar o nível de referência da variável attsprts para “No”

```r
db$attsprts =  relevel(db$attsprts, ref = "No")
levels(db$attsprts)
```

```
[1] "No"  "Yes"
```

## 5.5 Criando o modelo

```r
modelo_1 <- glm(attsprts ~ 1 + sex + life + childs + educ + tvhours +  age,
                data = db,
                family = "binomial")
                #na.action = na.exclude -> para remover os NAs
```

## 5.6 Resultados do modelo

Vamos começar mais uma vez vendo o resultado que a função `summary()` nos oferece.

```r
summary(modelo_1)
```

```
Call:
glm(formula = attsprts ~ 1 + sex + life + childs + educ + tvhours +
    age, family = "binomial", data = db)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.625849   0.560416  -2.901  0.00372 **
sexFemale    -0.452288   0.145961  -3.099  0.00194 **
lifeRoutine   0.649746   0.333770   1.947  0.05157 .
lifeExciting  0.787760   0.336597   2.340  0.01926 *
childs        0.121782   0.049603   2.455  0.01408 *
educ          0.200783   0.029138   6.891 5.55e-12 ***
tvhours      -0.097926   0.039787  -2.461  0.01384 *
age          -0.036856   0.004969  -7.417 1.20e-13 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1352.4  on 977  degrees of freedom
Residual deviance: 1143.6  on 970  degrees of freedom
  (522 observations deleted due to missingness)
AIC: 1159.6

Number of Fisher Scoring iterations: 4
```

Novamente não é o melhor dos mundos mas temos os valores de p e podemos observar quais variáveis deram resultados significativos.

Para melhorar a visualização e trazer os resultados em Odds Ratio podemos utilizar a função `tab_model()` do pacote `sjPlot`.

```
tab_model(modelo_1,
          show.se = TRUE,
          show.aic = TRUE,
          show.loglik=TRUE,
          show.ci = FALSE)
```

| | attsprts | | |
|---|---|---|---|
| Predictors | Odds Ratios | std. Error | p |
| (Intercept) | 0.20 | 0.11 | **0.004** |
| sex [Female] | 0.64 | 0.09 | **0.002** |
| life [Routine] | 1.92 | 0.64 | 0.052 |
| life [Exciting] | 2.20 | 0.74 | **0.019** |
| childs | 1.13 | 0.06 | **0.014** |
| educ | 1.22 | 0.04 | **<0.001** |
| tvhours | 0.91 | 0.04 | **0.014** |
| age | 0.96 | 0.00 | **<0.001** |
| Observations | 978 | | |
| $R^2$ Tjur | 0.194 | | |
| AIC | 1159.552 | | |
| log-Likelihood | -571.776 | | |

Bem melhor!

> Dica!
>
> Veja a seção Section 5.10 para uma explicação sobre correções para os valores de p, como Bonferroni, Holm, Hochberg e Hommel.

Para não precisar ficar mudando a referência e conseguir interpretar valores menores que 1, podemos utilizar a função `estimates()` do pacote `flexplot`.

```
estimates(modelo_1)
```

```
             raw.coefficients    OR inverse.OR standardized.OR
(Intercept)            -1.626 0.197      5.083           1.000
sexFemale              -0.452 0.636      1.572           0.799
lifeRoutine             0.650 1.915      0.522           1.383
lifeExciting            0.788 2.198      0.455           1.482
childs                  0.122 1.130      0.885           1.230
educ                    0.201 1.222      0.818           1.837
tvhours                -0.098 0.907      1.103           0.821
age                    -0.037 0.964      1.038           0.527
             inverse.standardized.OR Prediction Difference (+/- 1 SD)
(Intercept)                   1.000                              <NA>
sexFemale                     1.251      -0.11 (relative to sexMale)
lifeRoutine                   0.723      0.16 (relative to lifeDull)
lifeExciting                  0.675      0.19 (relative to lifeDull)
childs                        0.813                               0.1
educ                          0.544                              0.29
tvhours                       1.219                               0.1
age                           1.899                               0.3
```

Na coluna "inverse.OR" temos os valores invertendo a ordem das referências. No caso da variável sexo, podemos observar que o valor de odds ratio para "Female" quando comparado com "Male" (Female - Male) é de 0,636. O Inverse.OR nos mostra o valor caso o nível de referência fosse invertido (Male - Female).

A interpretação do resultado, levando em conta o inverse.OR, também será invertida. Lembrando sempre que o valor de referência para a VD é "Não fazer esportes" (sedentarismo). Portanto podemos escrever um parágrafo de resultados assim:

"Pessoas do sexo feminino tem 1,57 mais chance de pertencer ao grupo que **faz** exercícios em relação à pessoas do sexo masculino."

Caso fique na dúvida, podemos sempre mudar o nível de referência da variável independente de interesse, rodar novamente o modelo e comparar os resultados.

```
# Alterando o nível de referência

db$sex = relevel(db$sex, ref =  "Female")
```

```
# Verificando se a troca ocorreu

levels(db$sex)
```

```
[1] "Female" "Male"
```

```
# Rodando novamente o modelo

modelo_1b <- glm(attsprts ~ 1 + sex + life + childs + educ + tvhours +  age,
                 data = db,
                 family = "binomial")
                 #na.action = na.exclude -> para remover os NAs
```

```
estimates(modelo_1b)
```

```
            raw.coefficients    OR inverse.OR standardized.OR
(Intercept)           -2.078 0.125      7.990           1.000
sexMale                0.452 1.572      0.636           1.251
lifeRoutine            0.650 1.915      0.522           1.383
lifeExciting           0.788 2.198      0.455           1.482
childs                 0.122 1.130      0.885           1.230
educ                   0.201 1.222      0.818           1.837
```

```
tvhours                -0.098 0.907      1.103           0.821
age                    -0.037 0.964      1.038           0.527
             inverse.standardized.OR Prediction Difference (+/- 1 SD)
(Intercept)                   1.000                              <NA>
sexMale                       0.799     0.11 (relative to sexFemale)
lifeRoutine                   0.723      0.16 (relative to lifeDull)
lifeExciting                  0.675      0.19 (relative to lifeDull)
childs                        0.813                               0.1
educ                          0.544                              0.29
tvhours                       1.219                               0.1
age                           1.899                               0.3
```

Podemos observar que o valor de OR para "Male" quando comparado com "Female" (Male - Female) é idêntico ao inverse.OR quando a referência era "Male".

Agora temos que interpretar de forma direta os resultados e ficaria assim:

"Pessoas do sexo masculino tem 1,57 mais chance de pertencer ao grupo que **não faz** exercícios em relação à pessoas do sexo feminino"

> ⚠ Cuidado!
>
> Percebam que é fácil se enrolar com a descrição dos resultados. Faça da maneira que se sentir mais a vontade dentre as duas apresentadas e verifique sempre o nível de referência das variáveis.

## 5.7 c) Comparando modelos

> ℹ Exercício
>
> Os resultados das questões A e B são similares? Se sim, porque? Se não, qual dos modelos é mais adequado?

Aqui no R vamos apenas criar o modelo com a função do GzLM. Você pode criar um modelo com o módulo de regressão logís-

tica no Jamovi e comparar os resultados apresentados anteriormente.

## 5.8 d) e e) Número de filhos (VD)

> **Exercício**
>
> Verifique o efeito do sexo, opinião sobre a vida (life) e prática de exercícios (attsports) sobre o número de filhos. Controle os resultados para idade e anos de escolaridade. Faça um GLM Univariado e um GLzM para a mesma pergunta

Podemos utilizar a mesma função `glm()` para criar os dois modelos.

### 5.8.1 Modelo GLM

```
modelo_2 <- glm(childs ~ 1 + sex + life + attsprts + age + educ,
                data = db)
```

Como estamos analisando um modelo linear univariado assumindo que a distribuição da VD é normal, podemos interpretar diretamente os estimadores que são retornados pela função summary.

```
kable(summary(modelo_2)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 2.8726230 | 0.2926723 | 9.8151502 | 0.0000000 |
| sexMale | -0.1863170 | 0.0989903 | -1.8821732 | 0.0601095 |
| lifeRoutine | -0.3467610 | 0.2066828 | -1.6777451 | 0.0937167 |
| lifeExciting | -0.2098896 | 0.2104225 | -0.9974676 | 0.3187845 |
| attsprtsYes | 0.2586856 | 0.1075859 | 2.4044569 | 0.0163818 |
| age | 0.0387229 | 0.0029938 | 12.9342252 | 0.0000000 |
| educ | -0.0822842 | 0.0178006 | -4.6225436 | 0.0000043 |

E chamar a função `report()` para gerar os resultados.

```r
report(modelo_2)
```

```
We fitted a linear model (estimated using ML) to predict childs with sex, life,
attsprts, age and educ (formula: childs ~ 1 + sex + life + attsprts + age +
educ). The model's explanatory power is moderate (R2 = 0.20). The model's
intercept, corresponding to sex = Female, life = Dull, attsprts = No, age = 0
and educ = 0, is at 2.87 (95% CI [2.30, 3.45], t(977) = 9.82, p < .001). Within
this model:

  - The effect of sex [Male] is statistically non-significant and negative (beta
= -0.19, 95% CI [-0.38, 7.70e-03], t(977) = -1.88, p = 0.060; Std. beta =
-0.11, 95% CI [-0.22, 4.53e-03])
  - The effect of life [Routine] is statistically non-significant and negative
(beta = -0.35, 95% CI [-0.75, 0.06], t(977) = -1.68, p = 0.093; Std. beta =
-0.20, 95% CI [-0.44, 0.03])
  - The effect of life [Exciting] is statistically non-significant and negative
(beta = -0.21, 95% CI [-0.62, 0.20], t(977) = -1.00, p = 0.319; Std. beta =
-0.12, 95% CI [-0.37, 0.12])
  - The effect of attsprts [Yes] is statistically significant and positive (beta
= 0.26, 95% CI [0.05, 0.47], t(977) = 2.40, p = 0.016; Std. beta = 0.15, 95% CI
[0.03, 0.28])
  - The effect of age is statistically significant and positive (beta = 0.04, 95%
CI [0.03, 0.04], t(977) = 12.93, p < .001; Std. beta = 0.40, 95% CI [0.34,
0.46])
  - The effect of educ is statistically significant and negative (beta = -0.08,
95% CI [-0.12, -0.05], t(977) = -4.62, p < .001; Std. beta = -0.15, 95% CI
[-0.21, -0.08])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald t-distribution approximation.
```

### 5.8.2 Modelo Poisson

```r
modelo_3 <- glm(childs ~ 1 + sex + life + attsprts + age + educ,
                data = db,
                family = "poisson")
```

### 5.8.3 Comparando AIC os modelos

Podemos comparar os índices de aderência dos dois modelos para verificar qual se ajusta melhor aos dados.

```
AIC(modelo_2, modelo_3)
```

```
         df      AIC
modelo_2  8 3630.810
modelo_3  7 3498.782
```

```
BIC(modelo_2, modelo_3)
```

```
         df      BIC
modelo_2  8 3669.943
modelo_3  7 3533.024
```

Podemos observar que tanto o AIC quanto o BIC favorecem o modelo_3 com distribuição Poisson.

### 5.8.4 Resultados

Sempre começando com a boa e velha função `summary()`.

```
summary(modelo_3)
```

```
Call:
glm(formula = childs ~ 1 + sex + life + attsprts + age + educ,
    family = "poisson", data = db)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.988656   0.110271   8.966  < 2e-16 ***
sexMale      -0.067072   0.039024  -1.719   0.0857 .
lifeRoutine  -0.102725   0.074303  -1.383   0.1668
lifeExciting -0.052165   0.075989  -0.686   0.4924
```

```
attsprtsYes   0.095588   0.042782   2.234   0.0255 *
age           0.013111   0.001144  11.462  < 2e-16 ***
educ         -0.026711   0.006805  -3.925 8.67e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 951.59  on 983  degrees of freedom
Residual deviance: 757.85  on 977  degrees of freedom
  (516 observations deleted due to missingness)
AIC: 3498.8

Number of Fisher Scoring iterations: 4
```

Temos que idade (age), nível de educação (educ) e praticar esportes (attsportsYes) são significativos.

E agora podemos utilizando mais uma vez a função estimates() para ver os valores de de cada variável exp(B). Reparem que aqui não teremos as Odds Ratio, mas sim uma coluna chamada **multiplicative.coef**, que no caso de modelos Poisson de desenho transversal é a **razão de prevalência**. A maneira de interpretar é a mesma da regressão logística.

```
estimates(modelo_3)
```

```
            raw.coefficients multiplicative.coef std.mult.coef
(Intercept)            0.989               2.688         1.000
sexMale               -0.067               0.935         0.967
lifeRoutine           -0.103               0.902         0.950
lifeExciting          -0.052               0.949         0.974
attsprtsYes            0.096               1.100         1.049
age                    0.013               1.013         1.257
educ                  -0.027               0.974         0.922
            Prediction Difference (+/- 1 SD)
(Intercept)                             <NA>
sexMale         -0.17 (relative to sexFemale)
lifeRoutine      -0.26 (relative to lifeDull)
lifeExciting     -0.14 (relative to lifeDull)
```

```
attsprtsYes     0.24 (relative to attsprtsNo)
age                                     1.12
educ                                    0.39
```

```
exp(modelo_3$coefficients)
```

```
 (Intercept)      sexMale  lifeRoutine lifeExciting  attsprtsYes          age
   2.6876196    0.9351281    0.9023752    0.9491727    1.1003059    1.0131971
        educ
   0.9736426
```

Escrevendo o parágrafo de um dos resultados temos algo como:

"Pessoas que pertencem ao grupo que fazem esportes tem 10% a mais de chance de terem um filho quando comparadas com pessoas que são sedentárias.

No caso do nível educacional precisamos calcular o exp(B) inverso e ter cuidado na interpretação do resultado.

```
kable(exp(-coef(modelo_3)))
```

| | x |
|---|---|
| (Intercept) | 0.3720765 |
| sexMale | 1.0693722 |
| lifeRoutine | 1.1081865 |
| lifeExciting | 1.0535491 |
| attsprtsYes | 0.9088382 |
| age | 0.9869748 |
| educ | 1.0270709 |

Temos que para cada nível a mais de educação a chance de ter filhos **diminui** em aproximadamente 3%.

> Cuidado!
>
> Não recomendamos utilizar a função `report()` para modelos Poisson e de regressão logística. Os resultados apresenta

```r
report(modelo_3)
```

```
We fitted a poisson model (estimated using ML) to predict childs with sex,
life, attsprts, age and educ (formula: childs ~ 1 + sex + life + attsprts + age
+ educ). The model's explanatory power is substantial (Nagelkerke's R2 = 0.29).
The model's intercept, corresponding to sex = Female, life = Dull, attsprts =
No, age = 0 and educ = 0, is at 0.99 (95% CI [0.77, 1.20], p < .001). Within
this model:

  - The effect of sex [Male] is statistically non-significant and negative (beta
= -0.07, 95% CI [-0.14, 9.25e-03], p = 0.086; Std. beta = -0.07, 95% CI [-0.14,
9.25e-03])
  - The effect of life [Routine] is statistically non-significant and negative
(beta = -0.10, 95% CI [-0.25, 0.05], p = 0.167; Std. beta = -0.10, 95% CI
[-0.25, 0.05])
  - The effect of life [Exciting] is statistically non-significant and negative
(beta = -0.05, 95% CI [-0.20, 0.10], p = 0.492; Std. beta = -0.05, 95% CI
[-0.20, 0.10])
  - The effect of attsprts [Yes] is statistically significant and positive (beta
= 0.10, 95% CI [0.01, 0.18], p = 0.025; Std. beta = 0.10, 95% CI [0.01, 0.18])
  - The effect of age is statistically significant and positive (beta = 0.01, 95%
CI [0.01, 0.02], p < .001; Std. beta = 0.23, 95% CI [0.19, 0.27])
  - The effect of educ is statistically significant and negative (beta = -0.03,
95% CI [-0.04, -0.01], p < .001; Std. beta = -0.08, 95% CI [-0.12, -0.04])

Standardized parameters were obtained by fitting the model on a standardized
version of the dataset. 95% Confidence Intervals (CIs) and p-values were
computed using a Wald z-distribution approximation.
```

Crie mais modelos com interações entre as variáveis para praticar. Compare os índices de aderência dos modelos e depois descreva o que melhor se adequa aos dados.

## 5.9 Lista 5 resolvida no SPSS

https://www.youtube.com/watch?v=IHhhsXYZ-1A

## 5.10 Extras!

### 5.10.1 Resultados dos modelos na unha

```
modelo_4 <- glm(childs ~ sex * life * attsprts + age + educ,
                data = db,
                family = "poisson")
```

```
# Obter o resumo estatístico do modelo
resumo_modelo <- summary(modelo_4)

# Extrair os valores de p e os coeficientes
valores_p <- resumo_modelo$coefficients[, "Pr(>|z|)"]
coeficientes <- resumo_modelo$coefficients[, "Estimate"]

# Calcular os asteriscos para os níveis de significância
asteriscos <- ifelse(valores_p < 0.001, "***",
            ifelse(valores_p < 0.01, "**",
            ifelse(valores_p < 0.05, "*", "")))

# Calcular as estimativas de RR e intervalos de confiança
RP <- exp(coef(modelo_4))
IC <- exp(confint(modelo_4))
```

```
Waiting for profiling to be done...
```

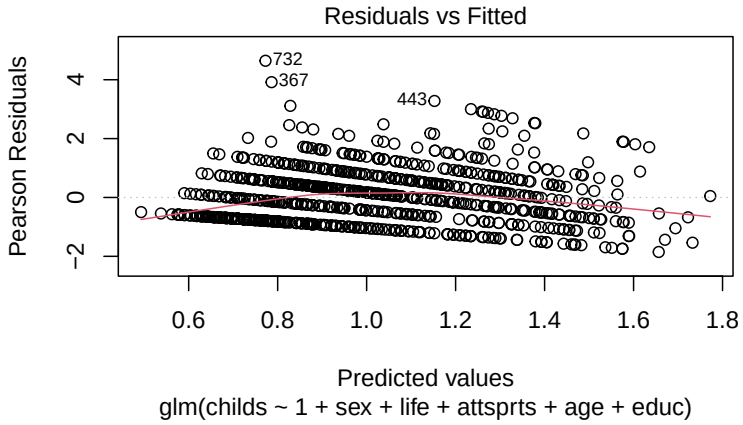
```
# Criar o dataframe parametros_modelo_2
parametros_modelo_4 <- data.frame(
  RP = round(RP, 2),
  IC_Lower = round(IC[, 1], 2),
  IC_Upper = round(IC[, 2], 2),
  Valores_p = round(valores_p, 4),
  Significância = asteriscos
)

parametros_modelo_4
```

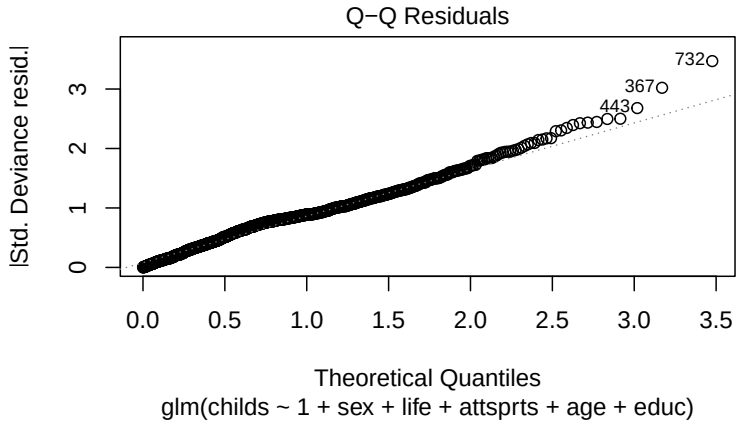
```
                                RP IC_Lower IC_Upper Valores_p Significância
```

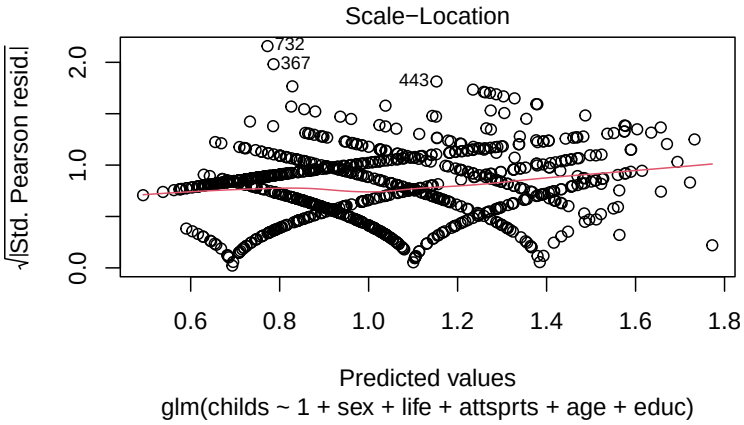
| | | | | | |
|---|---|---|---|---|---|
| (Intercept) | 2.67 | 2.09 | 3.39 | 0.0000 | *** |
| sexMale | 0.87 | 0.61 | 1.21 | 0.4045 | |
| lifeRoutine | 0.91 | 0.74 | 1.12 | 0.3565 | |
| lifeExciting | 0.93 | 0.76 | 1.15 | 0.5210 | |
| attsprtsYes | 1.19 | 0.81 | 1.72 | 0.3573 | |
| age | 1.01 | 1.01 | 1.02 | 0.0000 | *** |
| educ | 0.97 | 0.96 | 0.99 | 0.0002 | *** |
| sexMale:lifeRoutine | 1.12 | 0.78 | 1.64 | 0.5468 | |
| sexMale:lifeExciting | 1.03 | 0.71 | 1.53 | 0.8619 | |
| sexMale:attsprtsYes | 1.03 | 0.51 | 2.01 | 0.9373 | |
| lifeRoutine:attsprtsYes | 0.91 | 0.61 | 1.38 | 0.6476 | |
| lifeExciting:attsprtsYes | 0.93 | 0.63 | 1.40 | 0.7128 | |
| sexMale:lifeRoutine:attsprtsYes | 0.88 | 0.43 | 1.84 | 0.7297 | |
| sexMale:lifeExciting:attsprtsYes | 1.07 | 0.52 | 2.23 | 0.8600 | |

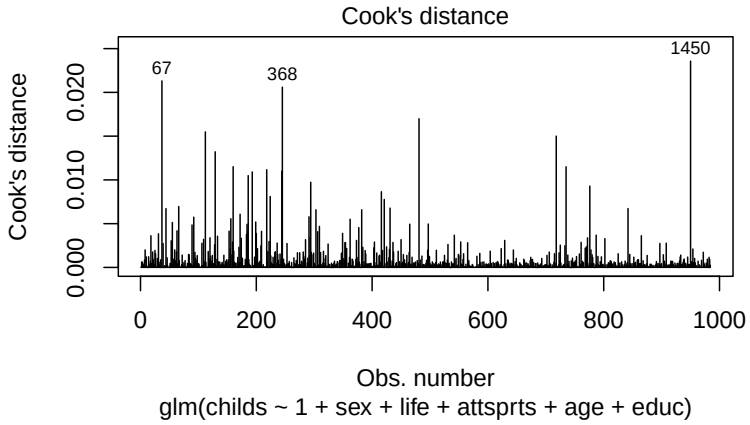
### 5.10.2 Pressupostos dos modelos Poisson

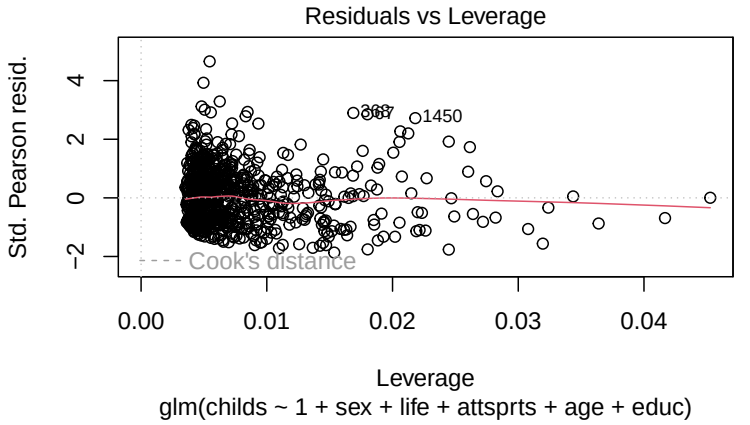
```
plot(modelo_3, which = 1:6)
```

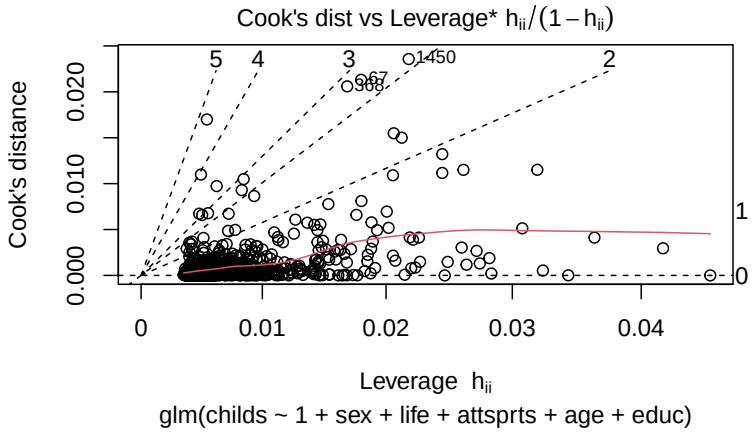



Q−Q Residuals
|Std. Deviance resid.|
732
367
443
0
1
2
3
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Theoretical Quantiles
glm(childs ~ 1 + sex + life + attsprts + age + educ)


Scale−Location
√|Std. Pearson resid.|
732
367
443
0.0
1.0
2.0
0.6
0.8
1.0
1.2
1.4
1.6
1.8
Predicted values
glm(childs ~ 1 + sex + life + attsprts + age + educ)


Cook's distance
67
368
1450
Cook's distance
0.000
0.010
0.020
0
200
400
600
800
1000
Obs. number
glm(childs ~ 1 + sex + life + attsprts + age + educ)


Residuals vs Leverage
368
1450
Cook's distance
Std. Pearson resid.
−2
0
2
4
0.00
0.01
0.02
0.03
0.04
Leverage
glm(childs ~ 1 + sex + life + attsprts + age + educ)




### Overdispersion

```
dispersiontest(modelo_3, trafo = 1)
```

```
    Overdispersion test

data:  modelo_3
z = -5.1887, p-value = 1
alternative hypothesis: true alpha is greater than 0
sample estimates:
     alpha
-0.2078915
```

Podemos também chamar a correção de Bonferroni para sermos mais conservadores com nossos resutlados.

```
tab_model(modelo_1,
          show.se = TRUE,
          show.aic = TRUE,
          show.loglik=TRUE,
          show.ci = FALSE,
          p.adjust = "bonferroni")
```

| | attsprts | | |
|---|---|---|---|
| Predictors | Odds Ratios | std. Error | p |
| (Intercept) | 0.20 | 0.11 | **0.030** |
| sex [Female] | 0.64 | 0.09 | **0.016** |
| life [Routine] | 1.92 | 0.64 | 0.413 |
| life [Exciting] | 2.20 | 0.74 | 0.154 |
| childs | 1.13 | 0.06 | 0.113 |
| educ | 1.22 | 0.04 | **<0.001** |
| tvhours | 0.91 | 0.04 | 0.111 |
| age | 0.96 | 0.00 | **<0.001** |
| Observations | 978 | | |
| $R^2$ Tjur | 0.194 | | |
| AIC | 1159.552 | | |
| log-Likelihood | -571.776 | | |

A correção de Bonferroni é um método utilizado para controlar o erro tipo I (falso positivo) em testes de hipóteses múltiplos. Quando você realiza vários testes simultaneamente, há um aumento no risco de obter resultados significativos simplesmente devido ao acaso (erro tipo I).

O método de Bonferroni ajusta os valores-p obtidos nos testes individuais para reduzir a probabilidade global de cometer um erro tipo I. A correção é feita dividindo o nível de significância (geralmente 0,05) pelo número total de testes realizados. Cada teste individual deve, então, ter um valor de significância ajustado para compensar o número de comparações.

A fórmula para a correção de Bonferroni é:

$$Valor\_p\_Ajustado = Valor\_de\_Significncia\_Original/Nmero\_Total\_de\_Testes$$

Por exemplo, suponha que você esteja conduzindo 5 testes de hipóteses e deseje manter um nível global de significância de 0,05. A correção de Bonferroni ajustaria o valor de significância para cada teste individual, resultando em 0,05/5=0,010,05/5=0,01.

Contudo, é importante destacar que a correção de Bonferroni tende a ser conservadora, o que significa que pode aumentar a probabilidade de erro tipo II (falso negativo), dificultando a detecção de diferenças ou efeitos reais. Existem alternativas menos conservadoras, como as correções de Holm ou Hochberg, que buscam um equilíbrio entre controle de erro e poder estatístico. A escolha da correção a ser utilizada depende do contexto específico da análise.

```
tidy(modelo_1, exponentiate = TRUE,
                        conf.int = TRUE)
```

```
# A tibble: 8 x 7
  term         estimate std.error statistic  p.value conf.low conf.high
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 (Intercept)     0.197   0.560       -2.90 3.72e- 3   0.0646     0.583
2 sexFemale       0.636   0.146       -3.10 1.94e- 3   0.477      0.846
3 lifeRoutine     1.92    0.334        1.95 5.16e- 2   1.01       3.77
4 lifeExciting    2.20    0.337        2.34 1.93e- 2   1.15       4.34
5 childs          1.13    0.0496       2.46 1.41e- 2   1.03       1.25
6 educ            1.22    0.0291       6.89 5.55e-12   1.16       1.30
7 tvhours         0.907   0.0398      -2.46 1.38e- 2   0.838      0.980
8 age             0.964   0.00497     -7.42 1.20e-13   0.954      0.973
```

### 5.10.3 Pseudo $R^2$

Função para calcular todos os três R2

Comentando cada linha temos:

- dev<-LogModel$deviance extrai o desvio do modelo ($-2$LL(new)) do modelo inserido na função

  e chama isso de dev.

- nullDev<-LogModel$null.deviance extrai o desvio da linha de base ($-2$LL(linha de base)) do modelo inserido a função e as chamadas são nullDev.

- modelN<-length(LogModel$fitted.values) usa a função length() no valor ajustado para calcular a amostra size, que ele chama de modelN.

- R.l <- 1 - dev/nullDev calcula a medida de Hosmer e Lemeshow (R2L) usando os valores extraídos do

  modelo e o chama de R.l.

- R.cs<- 1- exp ( -(nullDev - dev)/modelN): calcula a medida de Cox e Snell (R2CS) usando os valores extraídos do modelo e o chama de R.cs.

- R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN)))) calcula a medida de Nagelkerke (R2N) usando os valores extraídos do modelo e o chama de R.n.

```
logisticPseudoR2s <- function(LogModel) {
  dev <- LogModel$deviance
  nullDev <- LogModel$null.deviance
  modelN <- length(LogModel$fitted.values)
  R.l <- 1 - dev / nullDev
  R.cs <- 1- exp ( -(nullDev - dev) / modelN)
  R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))
  resultados <- data.frame(
  Metodo = c("Hosmer-Lemeshow", "Cox-Snell", "Nagelkerke"),
  Pseudo_R2 = c(round(R.l, 3), round(R.cs, 3), round(R.n, 3)))
  return(resultados)
}

logisticPseudoR2s(modelo_3)
```

```
          Metodo Pseudo_R2
1 Hosmer-Lemeshow     0.204
2       Cox-Snell     0.179
3      Nagelkerke     0.288
```

```
#exp(modelo_1$coefficients)
```

O $R^2$ (R-squared) em modelos de regressão linear é uma métrica que representa a proporção da variabilidade da variável dependente que é explicada pelo modelo. No entanto, ao lidar com modelos de regressão logística ou outros modelos generalizados, a interpretação direta do $R^2$ torna-se mais complexa devido à natureza da função de ligação utilizada.

Por isso, foi desenvolvido o Pseudo $R^2$ como uma medida análoga ao $R^2$, mas adaptada para modelos logísticos. Existem várias versões de Pseudo $R^2$, e a interpretação pode variar dependendo da versão específica utilizada. Aqui, abordarei uma interpretação geral.

### 5.10.4 Diferenças principais entre $R^2$ e Pseudo $R^2$:

**Interpretação Direta:** $R^2$ (em modelos lineares): Representa a proporção da variância explicada pela variável independente(s). Pseudo $R^2$ (em modelos logísticos): Oferece uma medida análoga, mas a interpretação é menos direta, pois está relacionada à verossimilhança e à diferença entre a verossimilhança do modelo ajustado e a verossimilhança de um modelo nulo.

**Intervalo de Valores:** $R^2$ (em modelos lineares): Pode variar de 0 a 1, indicando a porcentagem da variabilidade explicada pela variável independente. Pseudo $R^2$ (em modelos logísticos): Pode variar de 0 a 1, mas o significado exato depende da versão específica. Em alguns casos, um Pseudo $R^2$ mais alto indica um melhor ajuste, mas a interpretação exata pode variar.

### 5.10.5 Quando usar poisson e bin negativa?

Comparar AIC/BIC Variância maior que a média -> Usar bin negativa Poisson overdisperssion = Evento muito raro, com muitos zeros no banco.

## 5.11 Referências

https://www.youtube.com/watch?v=QPY4zuxs1W0

https://bookdown.org/drki_musa/dataanalysis/poisson-regression.html#prepare-r-environment-for-analysis-1

## 5.12 Versões dos pacotes

```
report(sessionInfo())
```

Analyses were conducted using the R Statistical language (version 4.3.1; R Core Team, 2023) on Windows 11 x64 (build 22621), using the packages lme4 (version 1.1.34; Bates D et al., 2015), Matrix (version 1.6.0; Bates D et al., 2023), effectsize (version 0.8.6; Ben-Shachar MS et al., 2020), fitdistrplus (version 1.1.11; Delignette-Muller ML, Dutang C, 2015), tm (version 0.7.11; Feinerer I, Hornik K, 2023), flexplot (version 0.20.5; Fife D, 2024), effects (version 4.2.2; Fox J, Weisberg S, 2019), car (version 3.1.2; Fox J, Weisberg S, 2019), carData (version 3.0.5; Fox J et al., 2022), mvtnorm (version 1.2.3; Genz A, Bretz F, 2009), lubridate (version 1.9.3; Grolemund G, Wickham H, 2011), NLP (version 0.2.1; Hornik K, 2020), TH.data (version 1.1.2; Hothorn T, 2023), multcomp (version 1.4.25; Hothorn T et al., 2008), AER (version 1.2.10; Kleiber C, Zeileis A, 2008), emmeans (version 1.8.8; Lenth R, 2023), sjstats (version 0.18.2; Lüdecke D, 2022), sjPlot (version 2.8.15; Lüdecke D, 2023), parameters (version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), tibble (version 3.2.1; Müller K, Wickham H, 2023), datawizard (version 0.9.0; Patil I et al., 2022), nlme (version 3.1.163; Pinheiro J et al., 2023), foreign (version 0.8.85; R Core Team, 2023), broom (version 1.0.5; Robinson D et al., 2023), gtsummary (version 1.7.2; Sjoberg D et al., 2021), rempsyc (version 0.1.6; Thériault R, 2023), survival (version 3.5.7; Therneau T, 2023), MASS (version 7.3.60; Venables WN, Ripley BD, 2002), ggplot2 (version 3.4.4; Wickham H, 2016), forcats (version 1.0.0; Wickham H, 2023), stringr (version 1.5.1; Wickham H, 2023), tidyverse (version 2.0.0; Wickham H et al., 2019), dplyr (version 1.1.3; Wickham H et al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), readr (version 2.1.4; Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et al., 2023), zoo (version 1.8.12; Zeileis A, Grothendieck G, 2005), lmtest (version 0.9.40; Zeileis A, Hothorn T, 2002), sandwich (version 3.1.0; Zeileis A et al., 2020) and kableExtra (version 1.3.4; Zhu H, 2021).

References
---

- Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." _Journal of Statistical Software_, *67*(1), 1-48. doi:10.18637/jss.v067.i01 <https://doi.org/10.18637/jss.v067.i01>.
- Bates D, Maechler M, Jagan M (2023). _Matrix: Sparse and Dense Matrix Classes and Methods_. R package version 1.6-0, <https://CRAN.R-project.org/package=Matrix>.
- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Delignette-Muller ML, Dutang C (2015). "fitdistrplus: An R Package for Fitting Distributions." _Journal of Statistical Software_, *64*(4), 1-34. doi:10.18637/jss.v064.i04 <https://doi.org/10.18637/jss.v064.i04>.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, 3rd edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html>. Fox J, Weisberg S (2018). "Visualizing Fit and Lack of Fit in Complex Regression Models with Predictor Effect Plots and Partial Residuals." _Journal of Statistical Software_, *87*(9), 1-27. doi:10.18637/jss.v087.i09 <https://doi.org/10.18637/jss.v087.i09>. Fox J (2003). "Effect Displays in R for Generalised Linear Models." _Journal of Statistical Software_, *8*(15), 1-27. doi:10.18637/jss.v008.i15 <https://doi.org/10.18637/jss.v008.i15>. Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." _Journal of Statistical Software_, *32*(1), 1-24. doi:10.18637/jss.v032.i01 <https://doi.org/10.18637/jss.v032.i01>.
- Fox J, Weisberg S (2019). _An R Companion to Applied Regression_, Third edition. Sage, Thousand Oaks CA. <https://socialsciences.mcmaster.ca/jfox/Books/Companion/>.
- Fox J, Weisberg S, Price B (2022). _carData: Companion to Applied Regression Data Sets_. R package version 3.0-5, <https://CRAN.R-project.org/package=carData>.
- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag,

Heidelberg. ISBN 978-3-642-01688-2.
  - Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate." _Journal of Statistical Software_, *40*(3), 1-25. <https://www.jstatsoft.org/v40/i03/>.
  - Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.
  - Hothorn T (2023). _TH.data: TH's Data Archive_. R package version 1.1-2, <https://CRAN.R-project.org/package=TH.data>.
  - Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." _Biometrical Journal_, *50*(3), 346-363.
  - Kleiber C, Zeileis A (2008). _Applied Econometrics with R_. Springer-Verlag, New York. ISBN 978-0-387-77316-2, <https://CRAN.R-project.org/package=AER>.
  - Lenth R (2023). _emmeans: Estimated Marginal Means, aka Least-Squares Means_. R package version 1.8.8, <https://CRAN.R-project.org/package=emmeans>.
  - Lüdecke D (2022). _sjstats: Statistical Functions for Regression Models (Version 0.18.2)_. doi:10.5281/zenodo.1284472 <https://doi.org/10.5281/zenodo.1284472>, <https://CRAN.R-project.org/package=sjstats>.
  - Lüdecke D (2023). _sjPlot: Data Visualization for Statistics in Social Science_. R package version 2.8.15, <https://CRAN.R-project.org/package=sjPlot>.
  - Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.
  - Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
  - Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
  - Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
  - Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
  - Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian

Framework." _Journal of Open Source Software_, *4*(40), 1541.
doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>,
<https://joss.theoj.org/papers/10.21105/joss.01541>.
  - Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of
Model-Based Predictions, Contrasts and Means." _CRAN_.
<https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023).
"Automated Results Reporting as a Practical Tool to Improve Reproducibility and
Methodological Best Practices Adoption." _CRAN_.
<https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022).
"correlation: Methods for Correlation Analysis." Version 0.8.3,
<https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M,
Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in
R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306
<https://doi.org/10.21105/joss.02306>,
<https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version
3.2.1, <https://CRAN.R-project.org/package=tibble>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022).
"datawizard: An R Package for Easy Data Preparation and Statistical
Transformations." _Journal of Open Source Software_, *7*(78), 4684.
doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
  - Pinheiro J, Bates D, R Core Team (2023). _nlme: Linear and Nonlinear Mixed
Effects Models_. R package version 3.1-163,
<https://CRAN.R-project.org/package=nlme>. Pinheiro JC, Bates DM (2000).
_Mixed-Effects Models in S and S-PLUS_. Springer, New York. doi:10.1007/b98882
<https://doi.org/10.1007/b98882>.
  - R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS',
'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85,
<https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical
Computing_. R Foundation for Statistical Computing, Vienna, Austria.
<https://www.R-project.org/>.
  - Robinson D, Hayes A, Couch S (2023). _broom: Convert Statistical Objects into
Tidy Tibbles_. R package version 1.0.5,
<https://CRAN.R-project.org/package=broom>.
  - Sjoberg D, Whiting K, Curry M, Lavery J, Larmarange J (2021). "Reproducible
Summary Tables with the gtsummary Package." _The R Journal_, *13*, 570-580.
doi:10.32614/RJ-2021-053 <https://doi.org/10.32614/RJ-2021-053>,
<https://doi.org/10.32614/RJ-2021-053>.

- Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
- Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
- Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.
- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
- Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
- Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
- Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
- Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
- Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.
- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
- Zeileis A, Grothendieck G (2005). "zoo: S3 Infrastructure for Regular and Irregular Time Series." _Journal of Statistical Software_, *14*(6), 1-27. doi:10.18637/jss.v014.i06 <https://doi.org/10.18637/jss.v014.i06>.
- Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." _R News_, *2*(3), 7-10. <https://CRAN.R-project.org/doc/Rnews/>.
- Zeileis A, Köll S, Graham N (2020). "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." _Journal of

Statistical Software_, *95*(1), 1-36. doi:10.18637/jss.v095.i01
<https://doi.org/10.18637/jss.v095.i01>. Zeileis A (2004). "Econometric
Computing with HC and HAC Covariance Matrix Estimators." _Journal of
Statistical Software_, *11*(10), 1-17. doi:10.18637/jss.v011.i10
<https://doi.org/10.18637/jss.v011.i10>. Zeileis A (2006). "Object-Oriented
Computation of Sandwich Estimators." _Journal of Statistical Software_,
*16*(9), 1-16. doi:10.18637/jss.v016.i09
<https://doi.org/10.18637/jss.v016.i09>.

- Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe
Syntax_. R package version 1.3.4,
<https://CRAN.R-project.org/package=kableExtra>.

# Part II

# SURVIVAL

A análise de sobrevida é uma técnica estatística utilizada para investigar o tempo até a ocorrência de um evento, como a falha de um dispositivo, o desenvolvimento de uma doença ou a morte. No contexto da análise de sobrevida, a regressão de Cox, também conhecida como modelo de riscos proporcionais de Cox, é uma ferramenta essencial. Essa abordagem, desenvolvida por David R. Cox, permite avaliar a influência de variáveis independentes no tempo até o evento de interesse, mantendo a suposição de proporções constantes de risco ao longo do tempo. No ambiente estatístico R, a implementação da análise de sobrevida e da Cox regression é amplamente realizada por meio de pacotes como "survival" e "survminer". Essas ferramentas possibilitam a modelagem e visualização de dados de sobrevida, oferecendo uma compreensão mais aprofundada dos fatores que influenciam a taxa de ocorrência de eventos ao longo do tempo.

# 6 Lista 6 - Kaplan-Meier e Cox Regression

Temos os dados de 124 pacientes que estão na fila para um transplante de rim. Temos as variáveis:

- T_seg – tempo de seguimento (em meses);
- tx - fez transplante (sim, não);
- óbito - morte do paciente (sim, não);
- t_tx - tempo até o transplante (em meses).

Com base nos dados:

## 6.1 Carregando pacotes

```
library(tidyverse)
library(flexplot)
library(foreign)
library(dplyr)
library(tm)
library(ggplot2)
library(forcats)
library(rempsyc)
library(easystats)
library(kableExtra)
library(gtsummary)

#Específicos para survival
library(survival)
library(ggsurvfit)
library(survminer)
```

```r
library(broom)
library(survMisc)
library(PHInfiniteEstimates)
library(coin)
library(condSURV)
```

## 6.2 Limpando o ambiente

Quando executamos diversos comandos no R muitas vezes acabamos deixando o ambiente meio "sujo". Cheio de variáveis que não estamos mais utilizando, ou pacotes que estão carregados e não serão utilizados no momento.

Em longas sessões utilizando o R é sempre bom dar uma limpada no ambiente entre um projeto e outro. Para isso podemos executar o código abaixo:

```r
# Limpa o ambiente
rm(list = ls(all.names = TRUE)) # will clear all objects including hidden objects
gc() # free up memory and report the memory usage
```

```
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 3171438 169.4    4837031 258.4  4730248 252.7
Vcells 5395114  41.2   10146329  77.5  8388592  64.0
```

```r
options(max.print = .Machine$integer.max, scipen = 999, stringsAsFactors = F, dplyr.summaris

# Set seed
set.seed(42)
```

## 6.3 Definindo um tema para os gráficos

Preenhcer todos os parâmetros da função ggplot() é uma tarefa morosa e repetitiva. Podemos criar um tema para todos os nossos gráficos e assim manter a consistência nas figuras e não

precisar ficar escrevendo toda hora aquele parâmetro para mudar a espessura da linha do eixo X...

Uma vez definido o tema, podemos apenas chamá-lo dentro da função ggplot para repetir o padrão. Vamos armazenar todas as informações da padronização em uma variável com o código a seguir:

```
meu_tema <- theme(plot.title = element_text(size = rel(2)),
                  panel.grid.major.y = element_line(colour = 'gray'),
                  panel.grid.minor.y = element_line(colour = 'gray'),
                  panel.grid.major.x = element_blank(),
                  panel.grid.minor.x = element_blank(),
                  plot.background = element_rect(fill = NULL, colour = 'white'),
                  panel.background = element_rect(fill = 'white'),
                  # Axis stuff
                  axis.line = element_line(colour = 'black', linewidth = 1),
                  axis.text = element_text(colour = "black", face = 'bold'),
                  axis.text.x = element_text(size = rel(1)),
                  axis.text.y = element_text(size = rel(1)),
                  axis.title = element_text(size = rel(1.2)),
                  axis.ticks = element_line(colour = 'black', linewidth = 1.2),
                  # Legend stuff
                  legend.position = "bottom",
                  legend.margin = margin(6, 6, 6, 6),
                  legend.title = element_text(face = 'bold'),
                  legend.background = element_blank(),
                  legend.box.background = element_rect(colour = "black"))
```

Vamos utilizar o tema em nossos gráficos mais adiante!

## 6.4 Carregando os dados e modificando o tipo de variável

Como de costume, vamos carregar os dados e ver os tipos das variáveis que temos no banco de dados.

```
original = read.spss("teste Cox tempo dep Tx.sav", to.data.frame=TRUE)
glimpse(original)
```

```
Rows: 124
Columns: 5
$ id    <chr> "13758618I                      ", "13750502G                      ~
$ t_seg <dbl> 99, 98, 97, 97, 97, 96, 92, 90, 89, 87, 83, 83, 82, 82, 80, 77, ~
$ t_tx  <dbl> 22, 81, NA, 25, 93, 5, 1, 30, 88, 28, 30, 13, 49, NA, NA, 10, NA~
$ tx    <fct> sim, sim, não, sim, sim, sim, sim, sim, sim, sim, sim, sim, sim,~
$ obito <fct> não, não, não, não, não, não, não, não, sim, não, não, não, não,~
```

> ⚠ Cuidado!
>
> A variável do evento (`óbito` em nosso exemplo) PRECISA ser recodificada para uma variável numérica binária, ou seja, 1 e 0 caso queira realizar a análise de sobrevida.

Na seção Section 6.16, exploraremos as distinções entre conduzir as análises com os fatores "sim" e "não" versus os números 1 e 0.

Inicialmente, ajustaremos a variável para aceitar os valores 1 e 0, representando a ocorrência do evento e a censura, respectivamente. Para isso, empregaremos o operador pipe %>% para duplicar a base de dados original e efetuar a modificação no mesmo script. O operador pipe é útil para executar várias operações em uma única sequência de código.

```
db <- original %>%
  mutate(
    obito = as.integer(obito == "sim") # para transformar sim e não em 1 e 0, respectivament
  )

glimpse(db)
```

```
Rows: 124
Columns: 5
$ id    <chr> "13758618I                      ", "13750502G                      ~
$ t_seg <dbl> 99, 98, 97, 97, 97, 96, 92, 90, 89, 87, 83, 83, 82, 82, 80, 77, ~
$ t_tx  <dbl> 22, 81, NA, 25, 93, 5, 1, 30, 88, 28, 30, 13, 49, NA, NA, 10, NA~
$ tx    <fct> sim, sim, não, sim, sim, sim, sim, sim, sim, sim, sim, sim, sim,~
$ obito <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Pronto, agora temos que óbito assumiu os valores de números 1 e 0.

Outra análise exploratória importante a fazer nos dados é observar se há dados faltantes (NA) e onde eles estão, caso estejam presentes. Se uma variável tiver muitos NAs, vamos precisar de cautela para inserir a variável na análise.

```
# Verificando NAs
data.frame(
  nas_t_seg = sum(is.na(db$t_seg)),
  nas_t_seg = sum(is.na(db$t_tx)),
  nas_tx = sum(is.na(db$tx)),
  nas_obito = sum(is.na(db$obito))
)
```

```
  nas_t_seg nas_t_seg.1 nas_tx nas_obito
1         0          64      0         0
```

```
kable(report(db))
```

| | Variable | Level | n_Obs | percentage_Obs | percentage_Missing | Mean | SD | Median | MAD | Min |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | id | NA | 124 | | 0.00 | | | | | |
| 5 | t_seg | NA | 124 | | 0.00 | 45.22 | 24.08 | 42.00 | 19.27 | 0.00 |
| 6 | t_tx | NA | 124 | | 51.61 | 19.90 | 20.05 | | 16.31 | 1.00 |
| 1 | tx | sim | 60 | 48.39 | | | | | | |
| 2 | tx | não | 64 | 51.61 | | | | | | |
| 4 | obito | NA | 124 | | 0.00 | 0.27 | 0.45 | 0.00 | 0.00 | 0.00 |

O número de NAs na variável t_tx é alto (51.61%) pelo simples motivo de que pessoas que não fizeram transplante não possuem a marca do tempo que fizeram o transplante. Em todo caso podemos verificar se existem indivíduos que fizeram o transplante mas não possuem a marca do tempo em que fizeram o transplante.

```
db %>%
  filter(tx == "sim" & is.na(t_tx))
```

```
[1] id    t_seg t_tx  tx    obito
<0 linhas> (ou row.names de comprimento 0)
```

O código acima filtra os dados de pessoas que fizeram o transplante (tx sim) e que tenham NA na coluna t_tx. Como o resultado volta com zero elementos, podemos concluir que todas as pessoas que fizeram o transplante, possuem a marca do horário em que o transplante foi feito.

Na tabela acima podemos perceber também que a porcentagem de pessoas que não fizeram o transplante (51.61) é a mesma porcentagem de dados faltantes (missing) da variável t_tx (51.61)

Por fim, podemos ver quantas pessoas morreram pela causa de morte do desfecho durante o período de observação.

```
kable(table(db$obito))
```

| Var1 | Freq |
|---|---|
| 0 | 90 |
| 1 | 34 |

Lembrando que 1 é o evento, que no nosso exemplo é ocorrência do óbito

Vamos agora às análises.

## 6.5 Criando a estrutura de dados

Iniciamos especificando para a função `Surv()` as colunas referentes ao tempo observado e aos eventos de interesse, que, neste caso, são os óbitos.

```
surv_obj <- Surv(time = db$t_seg, event = db$obito)
```

## 6.6 a) Tábua de vida

> **Exercício**
>
> Faça duas tábuas de vida em função da variável óbito comparando grupos que fizeram ou não transplante: Ambas com período 0 até 99 meses. A primeira dividida em períodos de 20 meses e a segunda com períodos de 1 mês. Faça um parágrafo descrevendo as diferenças nos gráficos.

Agora vamos criar a tabela de vida. Por enquanto, não faremos a separação dos dados por grupos.

```
fit1 <- survfit(surv_obj ~ 1, data = db)
```

A função `summary()` também pode ser utilizada para verificar os resultados dos modelos de sobrevida.

```
summary(fit1)
```

```
Call: survfit(formula = surv_obj ~ 1, data = db)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    0    124       2    0.984  0.0113        0.962        1.000
    3    121       2    0.968  0.0159        0.937        0.999
    4    119       2    0.951  0.0194        0.914        0.990
    6    117       1    0.943  0.0208        0.903        0.985
    8    116       2    0.927  0.0234        0.882        0.974
   11    114       1    0.919  0.0246        0.872        0.968
   13    113       1    0.911  0.0257        0.862        0.962
   16    110       1    0.902  0.0268        0.851        0.956
   19    108       1    0.894  0.0278        0.841        0.950
   24    106       2    0.877  0.0297        0.821        0.937
   25    104       1    0.869  0.0306        0.811        0.931
   26    103       1    0.860  0.0314        0.801        0.924
   27    102       1    0.852  0.0323        0.791        0.917
   29    101       1    0.843  0.0330        0.781        0.911
   34     89       3    0.815  0.0358        0.748        0.888
   36     82       1    0.805  0.0367        0.736        0.880
```

```
  38     70       1    0.794  0.0379        0.723        0.871
  40     68       1    0.782  0.0391        0.709        0.862
  41     65       2    0.758  0.0414        0.681        0.844
  44     59       1    0.745  0.0427        0.666        0.834
  45     55       1    0.731  0.0440        0.650        0.823
  46     53       1    0.718  0.0453        0.634        0.812
  49     47       1    0.702  0.0468        0.616        0.800
  58     32       1    0.680  0.0502        0.589        0.786
  66     24       1    0.652  0.0556        0.552        0.771
  89      9       1    0.580  0.0843        0.436        0.771
```

E a função função `tidy_survfit()` nos oferece uma tabela bem mais completa.

```
tidy_survfit(fit1)
```

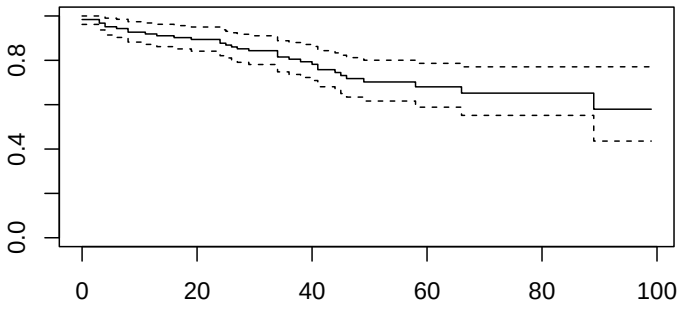
```
# A tibble: 63 x 14
    time n.risk n.event n.censor cum.event cum.censor estimate std.error
   <dbl>  <dbl>   <dbl>    <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
 1     0    124       2        1         2          1    0.984    0.0115
 2     3    121       2        0         4          1    0.968    0.0165
 3     4    119       2        0         6          1    0.951    0.0204
 4     6    117       1        0         7          1    0.943    0.0221
 5     8    116       2        0         9          1    0.927    0.0253
 6    11    114       1        0        10          1    0.919    0.0268
 7    13    113       1        0        11          1    0.911    0.0282
 8    14    112       0        1        11          2    0.911    0.0282
 9    15    111       0        1        11          3    0.911    0.0282
10    16    110       1        0        12          3    0.902    0.0297
# i 53 more rows
# i 6 more variables: conf.high <dbl>, conf.low <dbl>, estimate_type <chr>,
#   estimate_type_label <chr>, monotonicity_type <chr>, conf.level <dbl>
```

## 6.7 b) Kaplan-Meier

> **Exercício**
>
> Faça uma curva de Kaplan-meyer comparando os grupos que fizeram vs não fizeram transplante em relação ao óbito. Analise o gráfico e as saídas do teste.

Para produzir um gráfico Kaplan-Meier simples podemos utilizar a função `plot()`.

```
plot(fit1)
```

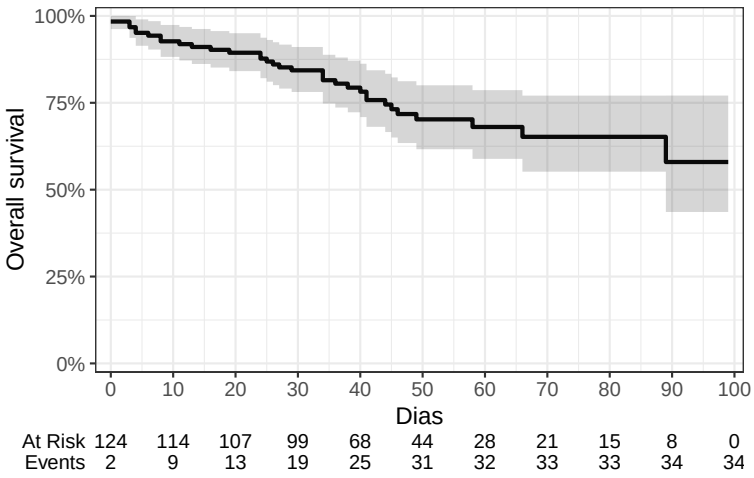



Meio pobrezinho e sem cor ne?

Podemos melhorar utilizando a função `ggsurvfit()`, do pacote com o mesmo nome.

```
fit1_km = ggsurvfit(fit1, linewidth = 1) +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
  add_risktable() +
  scale_ggsurvfit()
```

```
fit1_km
```




Até aqui estamos vendo o gráfico da sobrevida sem separar por grupos. A seguir vamos comparar entre os grupos que receberam ou não o transplante de rins.

## 6.8 Separando por transplante e nos tempos 0, 20, 40, 60, 80

Queremos comparar a sobrevida entre quem fez e não fez o transplante. Para isso podemos especificar no modelo que o transplante (tx) será uma das variáveis independentes.

```
fit2 = survfit(surv_obj ~ tx, # basta colocar tx como uma variável preditora no modelo
               data = db)


summary(fit2)
```

```
Call: survfit(formula = surv_obj ~ tx, data = db)

                tx=sim
```

```
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    3     59       1    0.983  0.0168        0.951        1.000
    4     58       1    0.966  0.0236        0.921        1.000
   24     57       1    0.949  0.0286        0.895        1.000
   26     56       1    0.932  0.0327        0.870        0.999
   29     55       1    0.915  0.0363        0.847        0.989
   38     44       1    0.894  0.0410        0.818        0.978
   41     41       1    0.873  0.0454        0.788        0.966
   45     37       1    0.849  0.0499        0.757        0.953
   49     32       1    0.823  0.0550        0.722        0.938
   66     19       1    0.779  0.0670        0.658        0.922
   89      8       1    0.682  0.1083        0.499        0.931

                tx=não
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    0     64       2    0.969  0.0217        0.927        1.000
    3     62       1    0.953  0.0264        0.903        1.000
    4     61       1    0.938  0.0303        0.880        0.999
    6     60       1    0.922  0.0335        0.858        0.990
    8     59       2    0.891  0.0390        0.817        0.970
   11     57       1    0.875  0.0413        0.798        0.960
   13     56       1    0.859  0.0435        0.778        0.949
   16     53       1    0.843  0.0456        0.758        0.937
   19     51       1    0.827  0.0476        0.738        0.925
   24     49       1    0.810  0.0495        0.718        0.913
   25     48       1    0.793  0.0513        0.699        0.900
   27     47       1    0.776  0.0529        0.679        0.887
   34     38       3    0.715  0.0594        0.607        0.841
   36     35       1    0.694  0.0611        0.584        0.825
   40     26       1    0.668  0.0643        0.553        0.806
   41     24       1    0.640  0.0674        0.520        0.786
   44     21       1    0.609  0.0707        0.485        0.765
   46     18       1    0.575  0.0745        0.447        0.742
   58     11       1    0.523  0.0841        0.382        0.717
```

A função `summary()` aceita um parâmetro com intervalos específicos para aparecer nos resultados. Vamos utilizar a função `seq()` para criar uma sequência de números que vai do 0 ao 100 com intervalos de 20 em 20.

```r
# Cria o intervalo de tempo

tempos_específicos <- seq(0, 100, by = 20) # sequencia de 0 a 100 em intervalos de 20.
```

Aplicando o intervalo na função temos o seguinte script:

```r
summary(fit2, times = tempos_específicos)
```

```
Call: survfit(formula = surv_obj ~ tx, data = db)

                tx=sim
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    0     60       0    1.000  0.0000        1.000        1.000
   20     57       2    0.966  0.0236        0.921        1.000
   40     42       4    0.894  0.0410        0.818        0.978
   60     21       3    0.823  0.0550        0.722        0.938
   80     12       1    0.779  0.0670        0.658        0.922

                tx=não
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    0     64       2    0.969  0.0217        0.927        1.000
   20     50       9    0.827  0.0476        0.738        0.925
   40     26       8    0.668  0.0643        0.553        0.806
   60      7       4    0.523  0.0841        0.382        0.717
   80      3       0    0.523  0.0841        0.382        0.717
```

Podemos nos perguntar também qual é a probabilidade de sobreviver após um certo tempo. Para obter a resposta basta ajustar o parâmetro `times` da função `summary()` para o tempo desejado.

```r
summary(fit2, times = 75)
```

```
Call: survfit(formula = surv_obj ~ tx, data = db)

                tx=sim
       time       n.risk      n.event     survival      std.err lower 95% CI
     75.000       14.000       10.000        0.779        0.067        0.658
upper 95% CI
```

```
      0.922

                tx=não
       time       n.risk      n.event     survival      std.err lower 95% CI
    75.0000       4.0000      23.0000       0.5232       0.0841       0.3818
upper 95% CI
      0.7169
```

Na análise do tempo de sobrevivência neste modelo, observamos o seguinte:

Para o grupo que **realizou** o transplante (tx=sim):

- Aos 75 meses, havia 14 indivíduos em risco.
- 10 eventos ocorreram até esse momento.
- A taxa de sobrevivência foi de 0.779, com um desvio padrão de 0.067.
- O intervalo de confiança de 95% para a taxa de sobrevivência variou de 0.658 a 0.922.

Para o grupo que **não realizou** o transplante (tx=não):

- Aos 75 meses, havia 4 indivíduos em risco.
- 23 eventos ocorreram até esse momento.
- A taxa de sobrevivência foi de 0.5232, com um desvio padrão de 0.0841.
- O intervalo de confiança de 95% para a taxa de sobrevivência variou de 0.3818 a 0.7169.

Podemos ainda calcular quantas vezes a probabilidade de sobrevivência é maior no grupo que realizou o transplante em comparação com o grupo que não o fez.

```r
summary(fit2, times = 75)$surv[1] / summary(fit2, times = 75)$surv[2]
```

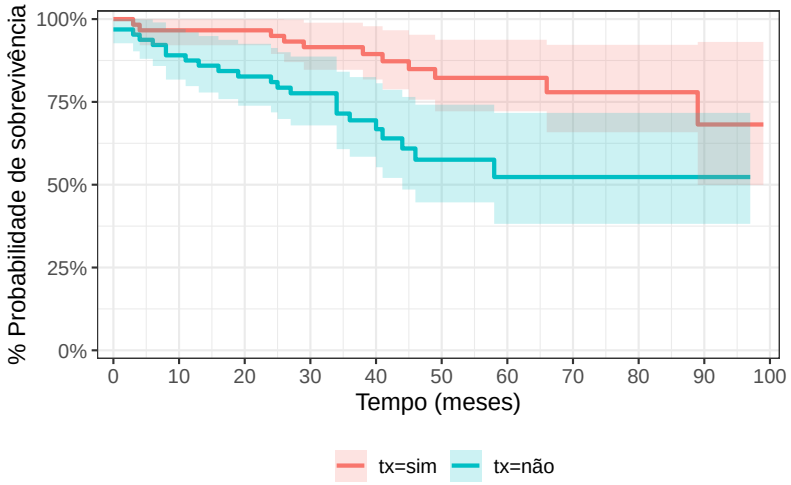
```
[1] 1.489431
```

O resultado revela que a probabilidade de sobrevivência no grupo que fez o transplante é aproximadamente 1.5 vezes maior do que no grupo que não o realizou.

### 6.8.1 Kaplan-Meir do novo modelo

Vamos salvar o plot padrão do segundo modelo para adicionar mais alguns parâmetros e incrementar a visualização dos resultados.

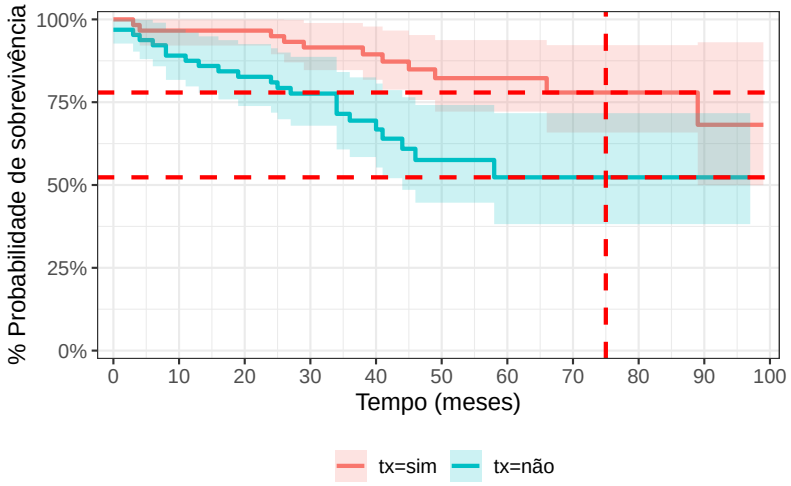
```
fit2_km = ggsurvfit(fit2, linewidth = 1) +
  labs(x = 'Tempo (meses)', y = '% Probabilidade de sobrevivência') +
  add_confidence_interval() +
  #add_risktable() +
  scale_ggsurvfit()

fit2_km
```




Com o plot salvo, podemos adicionar mais elementos aos poucos, como as linhas tracejadas para enfatizar diferenças.

```
fit2_km +
  geom_vline(xintercept = 75,
```

```
            linetype = 'dashed',
            colour = 'red',
            size = 1) + # adiciona a linha vermelha vertical
  geom_hline(yintercept = summary(fit2, times = 75)$surv,
            linetype = 'dashed',
            colour = 'red', size = 1) # adiciona as linhas vermelhas horizontais.
```



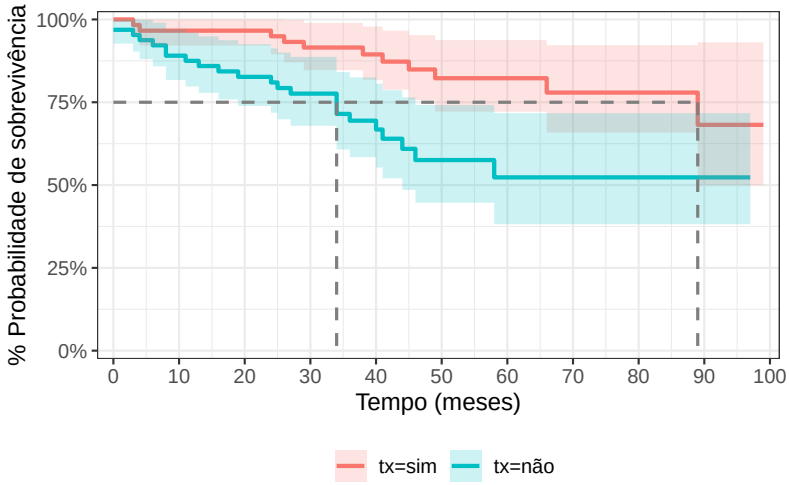

### 6.8.2 Porcentagem fixa, tempos diferentes

A função ggsurvfit oferece vários parâmetros interessantes. Um deles, bastante útil, permite traçar uma linha para comparar o tempo em que a probabilidade de sobrevivência X ocorre entre grupos diferentes.

Em quanto tempo será que a probabilidade de sobrevida chega a 75% nos dois grupos? Vamos utilizar o parâmetro `add_quantile()` para ter uma estimativa gráfica.

```
fit2 %>%
  ggsurvfit(linewidth = 1) +
  labs(x = 'Tempo (meses)', y = '% Probabilidade de sobrevivência') +
  add_confidence_interval() +
  #  add_risktable() +
  add_quantile(y_value = 0.75, color = "gray50", linewidth = 0.75) +
```

```
  scale_ggsurvfit()
```

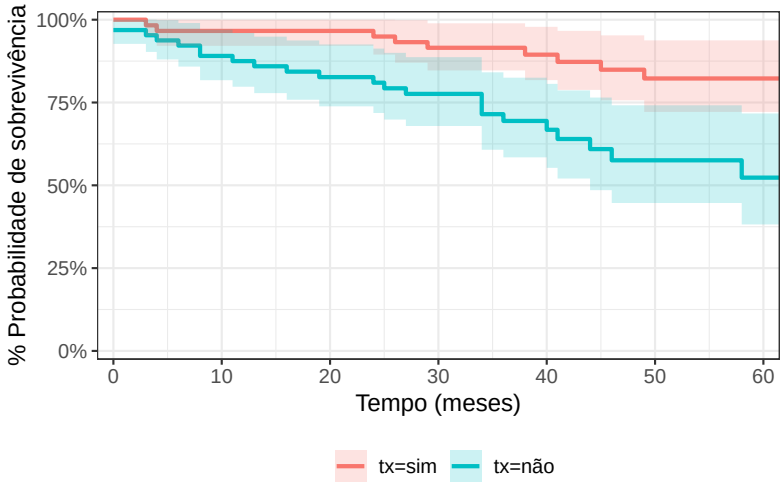



Ao examinarmos a imagem, observamos que o grupo que não passou pelo transplante atinge uma probabilidade de sobrevida de 75% em aproximadamente 35 meses. Por outro lado, no grupo que se submeteu ao transplante, essa mesma probabilidade só ocorre por volta do 90º mês, sendo ainda maior antes desse período.

### 6.8.3 Escolhendo um intervalo de tempo

Caso você queira apresentar apenas um período específico de tempo em sua análise, podemos fazer isso utilizando o parâmetro `coord_cartesian()`.

```
ggsurvfit(fit2, linewidth = 1) +
  labs(x = 'Tempo (meses)', y = '% Probabilidade de sobrevivência') +
  add_confidence_interval() +
  # add_risktable() +
  scale_ggsurvfit() +
  coord_cartesian(xlim = c(0, 60)) # coloque os números que
```




Personalize os limites do intervalo de tempo em sua análise ajustando os valores “0” e “60”, de acordo com suas necessidades específicas.

## 6.9 Comparando as curvas

- Log-rank: utilizar para comparar o primeiro terço do gráfico
- Gehan: utilizar para comparar o meio do gráfico
- Tarone: utilizar para comparar o final do gráfico
- Peto-Peto: parecido com o Log-rank, utilizar para comparar o primeiro terço do gráfico

Pacote mais indicado para utilizar é o `coin`.

### 6.9.1 Tipos de testes possíveis

“logrank”, “Gehan-Breslow”, “Tarone-Ware”, “Peto-Peto”, “Prentice”, “Prentice-Marek”, “Andersen-Borgan-Gill-Keiding”, “Fleming-Harrington”, “Gaugler-Kim-Liao”, “Self”

### Log-rank

```
coin::logrank_test(surv_obj ~ tx, data = db, type = "logrank" ) # padrão é o log-rank
```

```
    Asymptotic Two-Sample Logrank Test

data:  surv_obj by tx (sim, não)
Z = 2.9275, p-value = 0.003417
alternative hypothesis: true theta is not equal to 1
```

### Gehan-Breslow

```
coin::logrank_test(surv_obj ~ tx ,data = db, type = "Gehan-Breslow")
```

```
    Asymptotic Two-Sample Gehan-Breslow Test

data:  surv_obj by tx (sim, não)
Z = 3.0103, p-value = 0.00261
alternative hypothesis: true theta is not equal to 1
```

### Tarone-Ware

```
coin::logrank_test(surv_obj ~ tx ,data = db, type = "Tarone-Ware")
```

```
    Asymptotic Two-Sample Tarone-Ware Test

data:  surv_obj by tx (sim, não)
Z = 3.0338, p-value = 0.002415
alternative hypothesis: true theta is not equal to 1
```

#### Peto-Peto

```
coin::logrank_test(surv_obj ~ tx ,data = db, type = "Peto-Peto")
```

```
    Asymptotic Two-Sample Peto-Peto Test

data:  surv_obj by tx (sim, não)
Z = 2.9857, p-value = 0.002829
alternative hypothesis: true theta is not equal to 1
```

Em todos os testes a hipótese alternativa sugere que o verdadeiro parâmetro theta não é igual a 1, indicando assim que há diferenças significativas nas curvas de sobrevida entre os dois grupos analisados. Em termos práticos, isso sugere que a probabilidade de sobrevivência varia de maneira estatisticamente significativa entre os grupos que fizeram ou não o transplante.

## 6.10 c) Cox Regression

> **Exercício**
>
> Reproduza a análise do item b) com uma Cox Regression. Descreva os resultados

Compare com base no resultado da Cox, qual seria a diferença na sobrevida (HR) entre uma pessoa que fez e outra que não fez transplante com 50 meses de observação

A Regressão de Cox é uma técnica estatística utilizada para analisar a relação entre variáveis explicativas e o tempo até um evento ocorrer, como a morte. Ao contrário de modelos de regressão linear, a Regressão de Cox lida com dados de sobrevida, levando em consideração o tempo até o evento ou a censura. O código apresentado realiza uma Regressão de Cox com a função `coxph()`.

```r
# Cox regression ======================================================
# Fit the model
cox_res <- coxph(Surv(time = db$t_seg, event = db$obito) ~ tx, data = db)
```

O código acima ajusta o modelo de Regressão de Cox. A variável dependente é definida como o tempo (t_seg) até o evento (obito) ocorrer, e a variável independente é tx.

Notem que dentro da função `coxph()`, repetimos o código para gerar a tabela de vida.Durante a criação da estrutura dos dados, armazenamos a tabela de vida em uma variável chamada `surv_obj`. Podemos reutilizá-la na Regressão de Cox, evitando a necessidade de reescrever o código.

Vamos fazer isso!

```r
cox_res_2 = coxph(surv_obj ~ tx, data = db)
```

Bem mais limpo, não? E como vamos escrever mais alguns modelos, é uma boa prática salvar o padrão que se repete em uma variável.

### 6.10.1 Resultados dos modelos

Sabe qual função vamos utilizar para verificar o resultado? Sim, a `summary()`.

Primeiro vamos verificar se as duas formas que escrevemos os modelos geram os mesmos resultados.

```r
summary(cox_res)
```

```
Call:
coxph(formula = Surv(time = db$t_seg, event = db$obito) ~ tx,
    data = db)

  n= 124, number of events= 34

        coef exp(coef) se(coef)     z Pr(>|z|)
txnão 1.0787    2.9409   0.3753 2.874  0.00405 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
txnão     2.941       0.34     1.409     6.136

Concordance= 0.638  (se = 0.04 )
Likelihood ratio test= 8.99  on 1 df,   p=0.003
Wald test            = 8.26  on 1 df,   p=0.004
Score (logrank) test = 9.01  on 1 df,   p=0.003
```

---

```
summary(cox_res_2)
```

```
Call:
coxph(formula = surv_obj ~ tx, data = db)

  n= 124, number of events= 34

        coef exp(coef) se(coef)     z Pr(>|z|)
txnão 1.0787    2.9409   0.3753 2.874  0.00405 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
txnão     2.941       0.34     1.409     6.136

Concordance= 0.638  (se = 0.04 )
Likelihood ratio test= 8.99  on 1 df,   p=0.003
Wald test            = 8.26  on 1 df,   p=0.004
Score (logrank) test = 9.01  on 1 df,   p=0.003
```

---

Boa! Os resultados são idênticos, então podemos manter o padrão de escrevr o modelo utilizando a tábua de vida salva em uma variável.

Embora o resultado da função summary() para modelos de Regressão de Cox possa não ser visualmente atraente, ele oferece informações detalhadas sobre como o modelo se ajusta aos dados. Vamos analisar cada componente separadamente:

1. **Sumário do Modelo:**
    - **Call**: Indica a chamada da função utilizada para ajustar o modelo.
    - **n= 124, number of events= 34:** Informa o número total de observações (n) e o número de eventos ocorridos (number of events).
2. **Coeficientes:**
    - **coef:** O coeficiente estimado para a variável tx.
    - **exp(coef):** A interpretação deste valor é que, para pessoas do grupo que não fizeram o transplante (txnão), o risco de o evento (morte) ocorrer aumenta em 2.941 vezes.
    - **se(coef):** O erro padrão do coeficiente.
3. **Teste de Hipótese para Coeficientes:**
    - **z:** O valor z do teste de Wald, indicando quão longe o coeficiente está da média em termos de erros padrão.
    - **Pr(>|z|):** O p-valor associado ao teste de Wald. No exemplo, 0.00405 sugere que o efeito da variável tx é estatisticamente significativo.
    - **Significância codes:** ** indica significância a 0.01.
4. **Intervalo de Confiança para Exp(Coef):**
    - **exp(coef) exp(-coef) lower .95 upper .95:** O intervalo de confiança de 95% para o efeito da variável tx.
5. **Medidas de Desempenho do Modelo:**

- **Concordance= 0.638:** A concordância é uma medida de quão bem o modelo prevê a ordem de eventos.
- **Likelihood ratio test= 8.99, p=0.003:** O teste de razão de verossimilhança avalia se o modelo é significativamente melhor do que um modelo nulo. O p-valor sugere que o modelo é estatisticamente significativo.
- **Wald test= 8.26, p=0.004:** O teste de Wald também avalia a significância global do modelo.
- **Score (logrank) test= 9.01, p=0.003:** O teste de log-rank compara as curvas de sobrevivência entre os grupos.

E claro que temos formas melhores de visualizar e mostrar os dados mais importantes. Vamos utilizar a função `tbl_regression()` do pacote `gtsummary`, que pega um objeto de modelo de regressão e retorna uma tabela formatada pronta para publicação.

```
tbl_regression(cox_res, exponentiate = TRUE)
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

| **Characteristic** | **HR** | **95% CI** | **p-value** |
|---|---|---|---|
| tx | | | |
| sim | — | — | |
| não | 2.94 | 1.41, 6.14 | 0.004 |

Pra fazer isso aí no word demoraria uns 30 minutos hein? E ficaria feia ainda. Com uma linha de código fizemos miséria!

### 6.10.2 Plots do modelo e do resultado

Tendo ajustado um modelo de Cox aos dados, é possível visualizar a proporção de sobrevivência prevista em qualquer momento para um determinado grupo de risco.

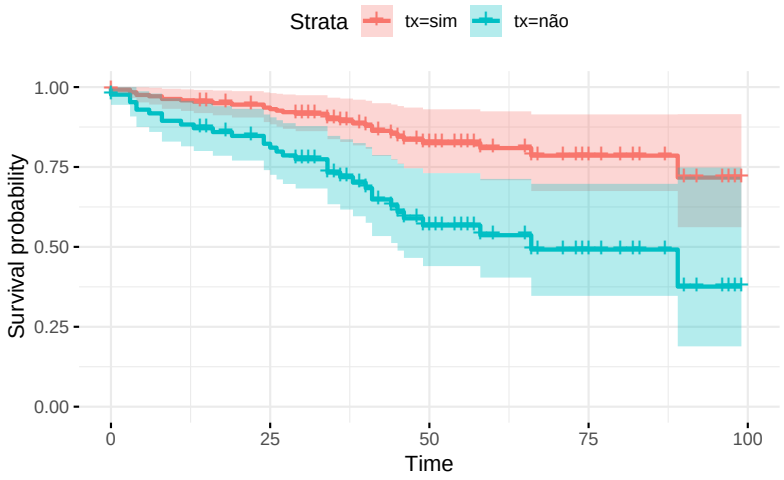
Neste caso, construímos um novo banco de dados com duas linhas, uma para cada valor de tx.

```
tx_df <- with(db,
              data.frame(tx = c("sim", "não")
              )
)
kable(tx_df)
```

| tx |
|---|
| sim |
| não |

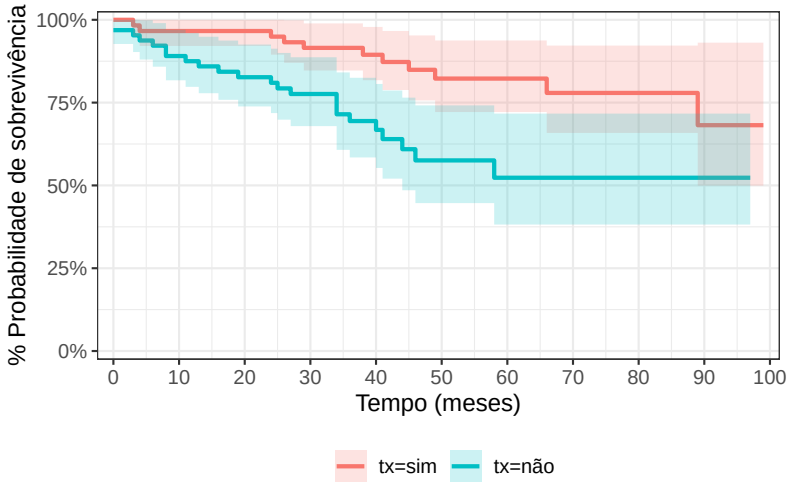
Agora podemos utilizar o nosso modelo para prever os valores de sobrevida e criar um gráfico da Regressão de Cox.

```
cox_graph <- survfit(cox_res, newdata = tx_df)

ggsurvplot(cox_graph, conf.int = TRUE, legend.labs=c("tx=sim", "tx=não"),
           ggtheme = theme_minimal(),
           data = db)
```



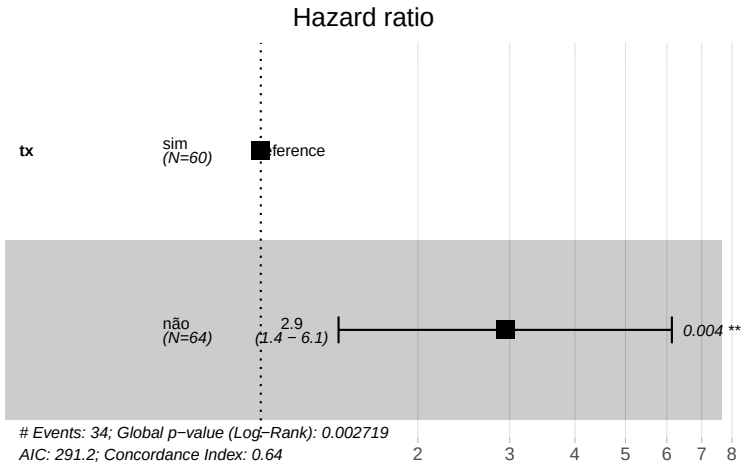
> **Cuidado!**
>
> O gráfico do modelo da Regressão de Cox é diferente do gráfico da Kaplan-Meir! O cálculo da regressão distorce os valores e encaixa o modelo aos dados. Observe a diferença!

```
# Gráfico da Kaplan-Meir
fit2_km
```




Não podemos deixar e fora o gráfico do modelo. Com pouca tinta (e pouco código) vamos mostrar tudo o que a função `summary()` nos proporcionou. Para isso vamos utilizar a função `ggforest()` do pacote `survminer`.

```
ggforest(cox_res, data = db)
```




Se não escorreu uma lágrima aí do outro lado da tela agora, eu desisto. E olha que utilizamos apenas uma variável independente no modelo!

## 6.11 Hazard ratio e risco relativo (CONFIRMAR O CONTEÚDO)

Vocês repararam que tanto na tabela quanto no gráfico com os resultados no modelo aparece o resultado como "Hazard Ratio"... pois bem, isso está errado!

> ⚠ Atenção!
>
> O risco relativo compara a probabilidade **cumulativa** de um evento ocorrer entre dois grupos ao longo de um período específico, enquanto o hazard ratio avalia a **razão instantânea** de riscos proporcionais entre os grupos, considerando a variação no risco ao longo do tempo. Enquanto o risco relativo se concentra em eventos cumulativos, o hazard ratio destaca as diferenças nas taxas instantâneas de falha, sendo especialmente útil em análises de sobrevida e estudos onde a dinâmica temporal do risco é crucial.

## 6.12 d) Hazard Ratio

> **Exercício**
>
> Compare com base no resultado da Cox, qual seria a diferença na sobrevida (HR) entre uma pessoa que fez e outra que não fez transplante com 50 meses de observação

Para de fato calcular o Hazard Ratio precisamos utilizar nosso modelo para prever a sobrevida em um tempo específico de nosso interesse.

Vamos começar salvando nosso modelo em uma variável

```
# Ajuste do modelo de regressão de Cox
cox_res <- coxph(Surv(time = t_seg, event = obito) ~ tx, data = db)
```

Agora vamos criar um conjunto de dados com informações simuladas sobre tempo de seguimento, ocorrência de evento (óbito), e uma variável indicadora de tratamento. Como queremos comparar o tempo de sobrevida entre quem fez ou não o transplante, a única variável que terá valores diferentes será a tx.

```
pred_dat <- data.frame(t_seg = c(41,41),
                       obito = c(0,0),
                       tx = c("sim","não")
                       )
kable(pred_dat)
```

| t_seg | obito | tx |
|---|---|---|
| 41 | 0 | sim |
| 41 | 0 | não |

A seguir vamos utiliza a função `predict()` para fazer previsões com base em nosso modelo previamente ajustado (cox_res).

```
preds <- predict(cox_res, newdata = pred_dat, type = "survival", se.fit = TRUE)
```

Salvamos o resultado da função em uma variável para poder adicionar os resultados das predições em nosso dataframe criado

anteriormente (pred_dat). Queremos os resultados da média e do Intervalo de Confiança. Para isso executamos o código a seguir:

```
pred_dat$prob <- preds$fit
pred_dat$lcl <- preds$fit - 1.96*preds$se.fit
pred_dat$ucl <- preds$fit + 1.96*preds$se.fit
kable(pred_dat)
```

| t_seg | obito | tx | prob | lcl | ucl |
|---|---|---|---|---|---|
| 41 | 0 | sim | 0.8630231 | 0.7805057 | 0.9455404 |
| 41 | 0 | não | 0.6484075 | 0.5224067 | 0.7744083 |

Por fim, podemos finalmente verificar o Hazard Ratio no tempo de 41 meses, dividindo a probabilidade de sobrevida do grupo que fez o transplante pela probabilidade de sobrevida do grupo que não fez o transplante.

```
HR_41 = pred_dat$prob[1] / pred_dat$prob[2] # Diferença na sobrevida (HR) no tempo 41 meses
HR_41
```

```
[1] 1.330989
```

---

Temos que no tempo de 41 meses a probabilidade de sobrevida de quem não fez o transplante é 1.33 menor do que quem fez o transplante.

Caso tenha interesse em mais pontos, podemos criar vários tempos de interesse em um único dataframe e repetir o código.

```
multi_pred_dat <- data.frame(t_seg = c(41,41, 50, 50, 80, 80),
                         obito = c(0,0,0,0,0,0),
                         tx = c("sim","não","sim","não","sim","não")
                         )

preds <- predict(cox_res, newdata = multi_pred_dat, type = "survival", se.fit = TRUE)

multi_pred_dat$prob <- preds$fit
```

```r
multi_pred_dat$lcl <- preds$fit - 1.96*preds$se.fit
multi_pred_dat$ucl <- preds$fit + 1.96*preds$se.fit

HR_41 = multi_pred_dat$prob[1] / multi_pred_dat$prob[2] # 41
HR_50 = multi_pred_dat$prob[3] / multi_pred_dat$prob[4] # 50
HR_80 = multi_pred_dat$prob[5] / multi_pred_dat$prob[6] # 80

tabela_HR = data.frame(Tempo = c(41, 50, 80),
                       HR_Não = c(HR_41, HR_50, HR_80))
kable(tabela_HR)
```

| Tempo | HR_Não |
|---|---|
| 41 | 1.330989 |
| 50 | 1.454309 |
| 80 | 1.597592 |

## 6.13 Verificando os pressupostos da Cox regression

A Regressão de Cox é uma técnica robusta, mas, como qualquer método estatístico, possui alguns pressupostos importantes. Os principais pressupostos da Regressão de Cox são:

1. **Proporcionalidade dos Riscos:**
    - O pressuposto fundamental é que os riscos relativos entre dois grupos são constantes ao longo do tempo. Em outras palavras, a razão instantânea de riscos (hazard ratio) entre grupos não muda com o tempo. Este é o pressuposto de proporcionalidade dos riscos.
2. **Independência Censura:**
    - A censura dos dados deve ser independente da probabilidade de falha. Isso significa que a probabilidade de um evento censurado (ocorrido após o fim do acompanhamento) deve ser a mesma para todos os grupos.
3. **Linearidade no Logaritmo dos Riscos:**

- A relação entre as variáveis independentes e o logaritmo do risco deve ser linear. Isso é crucial para a interpretação dos coeficientes como log-riscos instantâneos.

4. **Auscência de Colinearidade:**

    - As variáveis independentes no modelo não devem estar altamente correlacionadas (colinearidade). A colinearidade pode levar a estimativas imprecisas dos coeficientes.

5. **Ausência de Efeito de Interferência:**

    - Não deve haver efeito de interferência entre indivíduos, o que significa que o status de um indivíduo não deve influenciar diretamente o tempo de falha de outro indivíduo.

6. **Adequação do Modelo:**

    - O modelo escolhido deve ser apropriado para os dados. Avaliações de adequação, como testes de resíduos, podem ser úteis para verificar a qualidade do ajuste do modelo aos dados.

Os pressupostos de 2 a 6 são inerentes ao desenho do experimento e do acompanhamento durante as observações. O único que vamos abordar aqui no tutorial é o de proporcionalidade dos riscos.

### 6.13.1 Proporcionalidade dos riscos

Temos duas formas de avaliar a proporcionalidade dos riscos

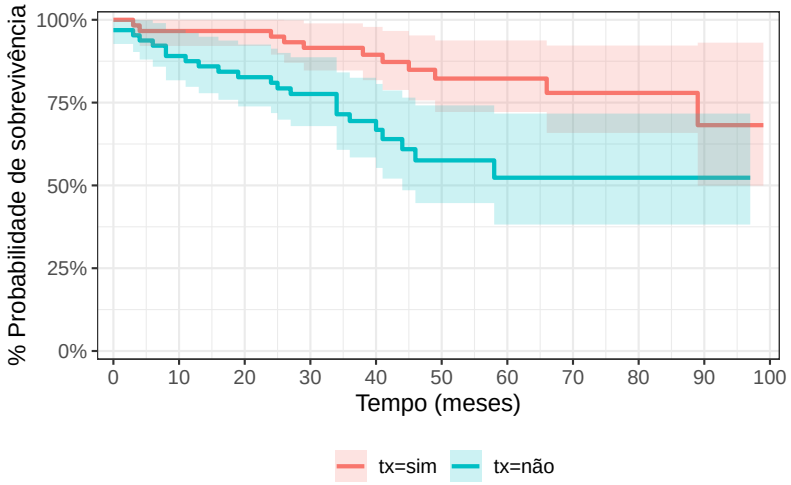
#### 1) Análise do gráfico da Kaplan-Meier

Ao analisar o gráfico de Kaplan-Meier para diferentes grupos, é crucial observar se as curvas de sobrevivência são aproximadamente paralelas ou se cruzam entre si. Se as curvas são paralelas, isso sugere proporcionalidade dos riscos, indicando que as

diferenças nas taxas de falha entre os grupos são constantes ao longo do tempo. No entanto, se as curvas se cruzam, isso indica uma possível violação da proporcionalidade dos riscos.

Cruzamentos nas curvas podem indicar mudanças na relação de risco entre os grupos ao longo do tempo. Essa mudança pode ser devido a diferentes dinâmicas de risco em períodos distintos do estudo. Se as curvas se cruzarem, a aplicação da Regressão de Cox não deve ser feita para não gerar interpretações erradas!

```
fit2_km
```




Podemos observar que em nosso exemplo as linhas de sobrevida não cruzam, portanto podemos assumir que os riscos são proporcionais pela análise gráfica.

### 2) Resíduos de Schoenfeld

A segunda forma para se avaliar a suposição de proporcionalidade dos riscos na Regressão de Cox vamos utilizar o teste de Schoenfeld, que verifica se há uma relação sistemática entre os resíduos de Schoenfeld e o tempo, o que indicaria uma violação dessa suposição.

A ideia central é que, se os resíduos de Schoenfeld não apresentarem uma relação significativa com o tempo, isso sugere que a proporcionalidade dos riscos é razoável. Logo, a hipótese nula é que não há relação entre os resíduos e o tempo, o que indicaria proporcionalidade dos riscos. O teste estatístico avalia se é razoável rejeitar essa hipótese nula.

> ❗ Importante!
>
> Vamos torcer para o valor de p ser **MAIOR** que 0.05!

Utilizando a função `cox.zph()` do pacote `survival` temos o seguinte código:

```
test <- survival::cox.zph(cox_res)
test
```

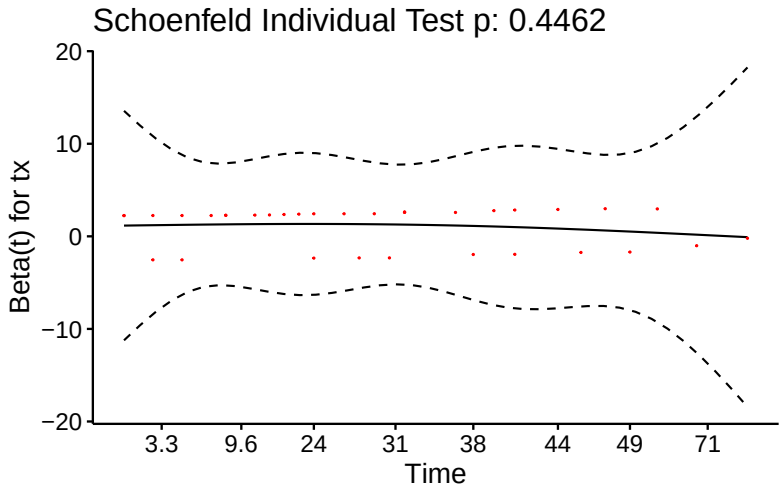
```
       chisq df    p
tx      0.58  1 0.45
GLOBAL  0.58  1 0.45
```

Ok! Temos riscos proporcionais!

Outra forma de verificar a proporcionalidade dos riscos é com o gráfico dos resíduos de Schoenfeld.

```
# Plot the Schoenfeld residuals over time for each covariate
survminer::ggcoxzph(test, point.size = 0.1)[1]
```

```
$`1`
```




Se os resíduos mostrarem um padrão claro ao longo do tempo, isso pode indicar uma violação da suposição de riscos proporcionais.

Algumas dicas para ajudar na interpretação:

- Sem Padrão (Resíduos Constantes): Se os resíduos aparecerem aleatoriamente espalhados em torno de zero, sem nenhuma tendência ou padrão claro, isso sugere que a suposição de riscos proporcionais é razoável.
- Tendência Linear: Uma tendência linear (aumentando ou diminuindo) nos resíduos ao longo do tempo pode sugerir uma violação da suposição de riscos proporcionais. Por exemplo, se os resíduos forem consistentemente positivos ou negativos ao longo do tempo, isso indica um efeito dependente do tempo.
- Padrão Não Linear: Se os resíduos exibirem um padrão não linear ou formatos específicos (por exemplo, formato de U, formato de V), isso pode indicar desvios dos riscos proporcionais.
- Paralelismo: Paralelismo significa que a propagação e distribuição dos resíduos são relativamente constantes ao longo do tempo. Se os resíduos aumentarem ou diminuirem ao longo do tempo, isso pode sugerir uma violação da suposição.

## 6.14 Conlcusões

Muito bacana a análise de sobrevida e a Regressão de Cox! Na seção Extras! vamos ver mais algumas formas de plotar os gráficos e avaliar a proporcionalidade dos riscos caso a Variável Independente seja contínua!

Próximo capitulo: Cox tempo-dependente!

## 6.15 Lista 6 resolvida no SPSS

https://www.youtube.com/watch?v=oyhA4EiE1eM

## 6.16 Extras!

### 6.16.1 Evento como fator ou como número

Como mencionado na seção Section 6.4, o tipo da variável do evento (morte) afeta os resultados tanto da Kaplan-Meir quanto na Regressão de Cox.

Vamos criar alguns modelos utilizando o banco de dados `original` (variável óbito é um fator) e também o `db` (variável óbito é binária, 1 e 0).

Vamos começar observando a diferença do tipo da variável nos bancos utilizando a função `glimpse()`:

```
glimpse(original$obito)
```

```
Factor w/ 2 levels "não","sim": 1 1 1 1 1 1 1 1 2 1 ...
```

```
glimpse(db$obito)
```

```
int [1:124] 0 0 0 0 0 0 0 0 1 0 ...
```

E tem mais! Temos que lembrar que quando deixamos as variáveis como fatores elas sempre possuem um nível de referência. Já verificamos isso em outros exercícios utilizando a função `levels()`.

```
levels(original$obito)
```

```
[1] "não" "sim"
```

Veja só! A referência para a variável óbito é o "não". Para fins didáditcos vamos criar três modelos:

Óbito como variável binária (banco `db`) Óbito como fator com nível de referência "não" (banco `original`) Óbito como fator com nível de referência "sim" (banco `original_sim`)

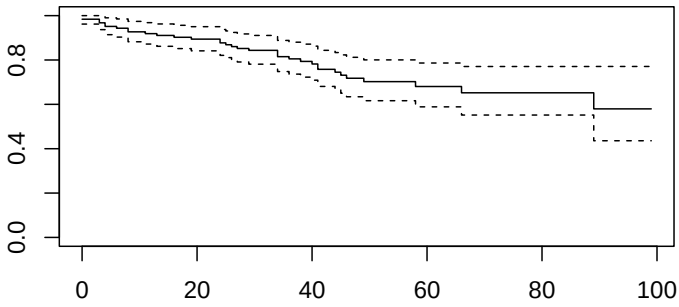
```
original_sim = original
original_sim$obito = relevel(original_sim$obito, ref = "sim")
```

Agora vamos repetir todo o procedimento já demonstrado no início das análises, utilizando os três bancos de dados.

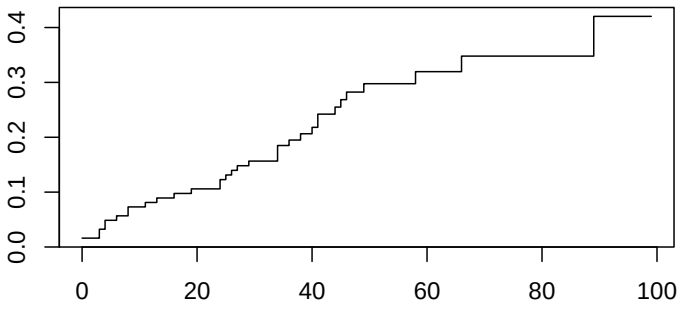
```
surv_db <- Surv(time = db$t_seg, event = db$obito)
surv_oiriginal_não <- Surv(time = original$t_seg, event = original$obito)
surv_oiriginal_sim <- Surv(time = original_sim$t_seg, event = original_sim$obito)
```

```
fit_db <- survfit(surv_db ~ 1, data = db)
fit_original_não <- survfit(surv_oiriginal_não ~ 1, data = original)
fit_original_sim <- survfit(surv_oiriginal_sim ~ 1, data = original_sim)
```

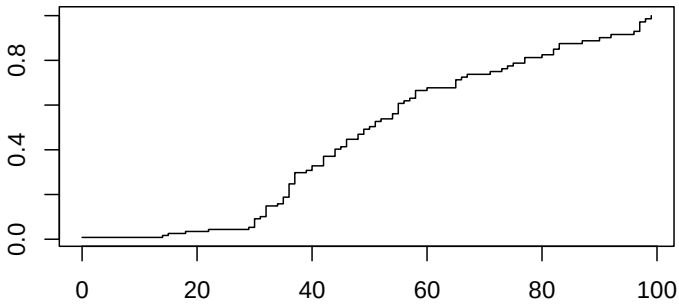
```
plot(fit_db)
```



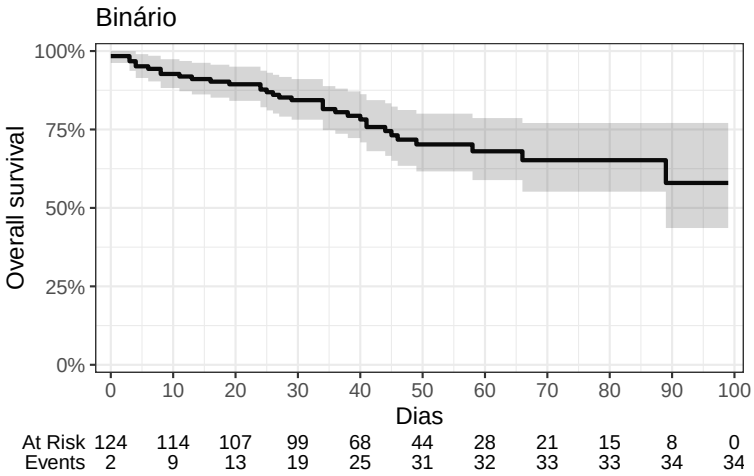

```
plot(fit_original_não)
```




```
plot(fit_original_sim)
```



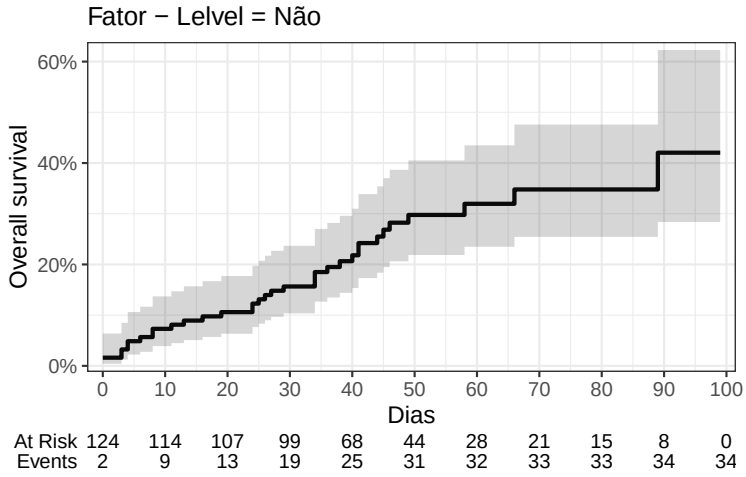

```
ggsurvfit(fit_db, linewidth = 1) +
  ggtitle("Binário") +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
    add_risktable() +
  scale_ggsurvfit()
```



```
ggcuminc(fit_original_não, linewidth = 1, type = "survival" ) +
  ggtitle("Fator - Lelvel = Não") +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
  add_risktable() +
  scale_ggsurvfit()
```

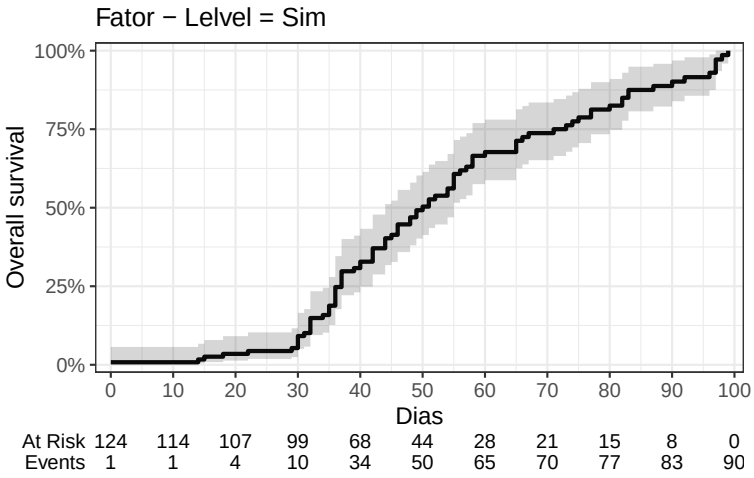
```
Plotting outcome "sim".
```

```
Warning in ggplot2::geom_step(ggplot2::aes(x = .data$time, y = .data$estimate),
: Ignoring unknown parameters: `type`
```




```
ggcuminc(fit_original_sim, linewidth = 1) +
  ggtitle("Fator - Lelvel = Sim") +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
  add_risktable() +
  scale_ggsurvfit()
```

```
Plotting outcome "não".
```

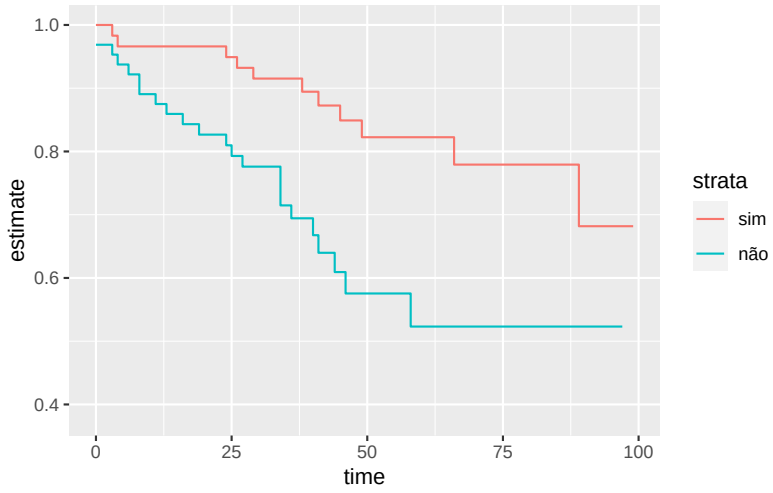



Como podemos observar, quando utilizamos a variável de evento como um fator, acabamos analisando o risco cumulativo e não a sobrevida.

### 6.16.2 Mais gráficos!

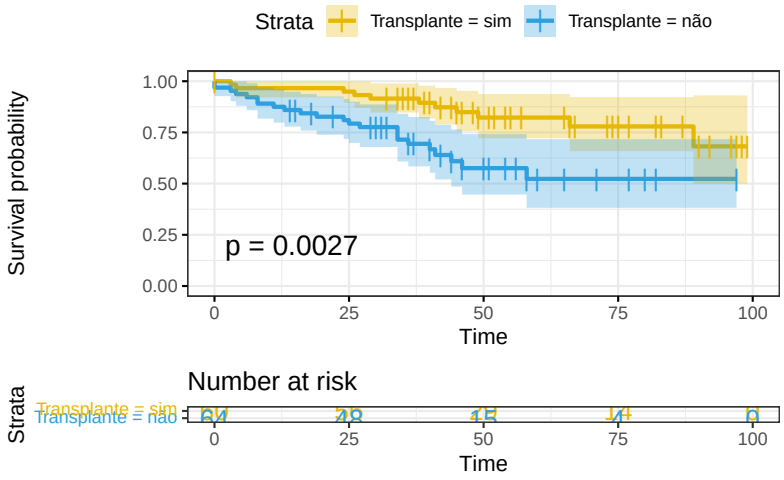
Com o ggplot2

```
km_plot = survfit2(surv_obj ~ tx, data = db) %>%
  tidy_survfit() %>%
  ggplot(aes(x = time, y = estimate,
             min = conf.low, ymax = conf.low,
             color = strata, fill = strata)) +
  geom_step()

km_plot
```




Com a função `ggsurvplot()` do pacote `survminer`.

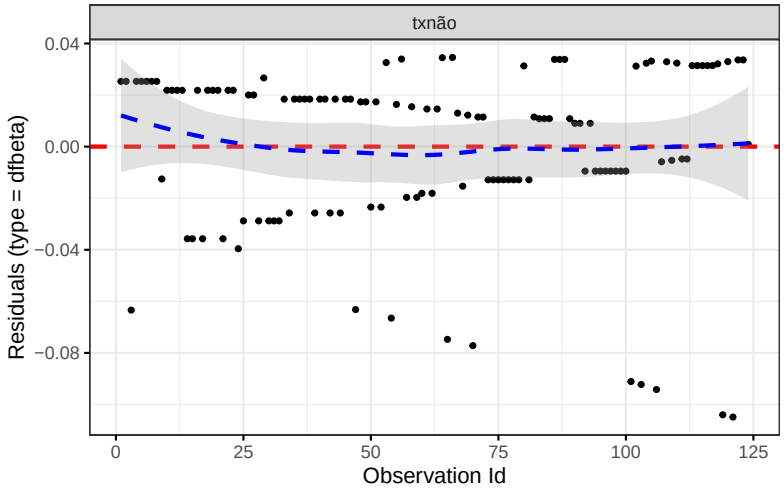
```
ggsurvplot(fit2, data = db,
           size = 1,
           palette = c('#E7B800', '#2e9fdf'),
           censor.shape = '|', censor.size = 4,
           conf.int = TRUE,
           pval = TRUE,
           risk.table = TRUE,
           risk.table.col = 'strata',
           legend.labs = list('0' = 'Transplante = sim', '1' = 'Transplante = não'),
           risk.table.height = 0.25,
           ggtheme = theme_bw())
```




Gráficos de proporcionalidade com outras funções

```
ggcoxdiagnostics(cox_res, type = "dfbeta", linear.predictions = FALSE)
```

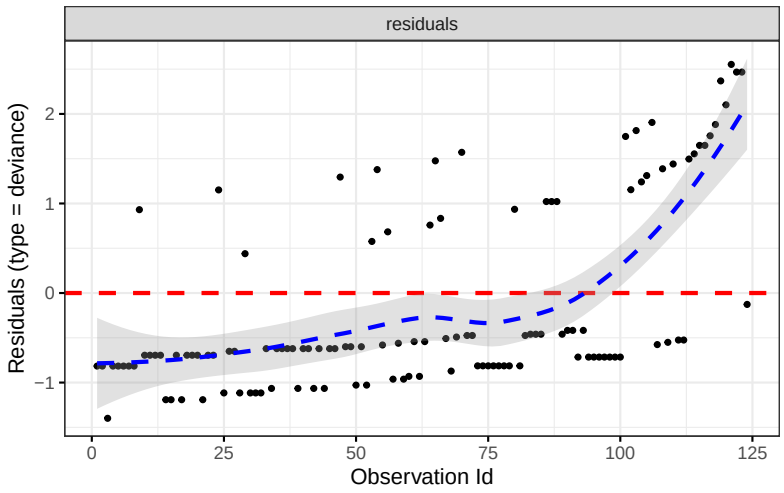
```
`geom_smooth()` using formula = 'y ~ x'
```




E um específico para variáveis contínuas.

```
# Não é importante para variáveis categóricas, mas fica o código para eventual consulta.
ggcoxdiagnostics(cox_res, type = "deviance", linear.predictions = FALSE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```




### 6.16.3 Pacots alternativos para comparar curvas

```
gehan.wilcoxon.test(surv_obj ~ tx ,data=db)
```

```
    Gehan-Wilcoxon

data:
= 9.1531, p-value = 0.002483
alternative hypothesis: two-sided
```

`survdiff` Com rho = 0 este é o teste log-rank ou Mantel-Haenszel, e com rho = 1 é equivalente à modificação Peto & Peto do teste Gehan-Wilcoxon.

```
survdiff(surv_obj ~ tx, data=db, rho = 2)
```

```
Call:
survdiff(formula = surv_obj ~ tx, data = db, rho = 2)

        N Observed Expected (O-E)^2/E (O-E)^2/V
tx=sim 60      7.4     14.0      3.12      9.19
tx=não 64     17.9     11.3      3.89      9.19

 Chisq= 9.2  on 1 degrees of freedom, p= 0.002
```

### 6.16.4 Tabela completa do modelo 2

```
life_table2 = survfit2(Surv(time = t_seg, event = obito) ~ tx, data = db) %>%
  tidy_survfit()

kable(life_table2)
```

| time | n.risk | n.event | n.censor | cum.event | cum.censor | estimate | std.error | conf.high | conf.low | s |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 60 | 0 | 1 | 0 | 1 | 1.0000000 | 0.0000000 | 1.0000000 | 1.0000000 | s |
| 3 | 59 | 1 | 0 | 1 | 1 | 0.9830508 | 0.0170946 | 1.0000000 | 0.9506595 | s |
| 4 | 58 | 1 | 0 | 2 | 1 | 0.9661017 | 0.0243866 | 1.0000000 | 0.9210112 | s |
| 24 | 57 | 1 | 0 | 3 | 1 | 0.9491525 | 0.0301329 | 1.0000000 | 0.8947194 | s |
| 26 | 56 | 1 | 0 | 4 | 1 | 0.9322034 | 0.0351093 | 0.9986097 | 0.8702130 | s |
| 29 | 55 | 1 | 0 | 5 | 1 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 32 | 54 | 0 | 3 | 5 | 4 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 34 | 51 | 0 | 1 | 5 | 5 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 35 | 50 | 0 | 3 | 5 | 8 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 36 | 47 | 0 | 1 | 5 | 9 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 37 | 46 | 0 | 2 | 5 | 11 | 0.9152542 | 0.0396152 | 0.9891503 | 0.8468787 | s |
| 38 | 44 | 1 | 0 | 6 | 11 | 0.8944530 | 0.0458029 | 0.9784644 | 0.8176548 | s |
| 39 | 43 | 0 | 1 | 6 | 12 | 0.8944530 | 0.0458029 | 0.9784644 | 0.8176548 | s |
| 40 | 42 | 0 | 1 | 6 | 13 | 0.8944530 | 0.0458029 | 0.9784644 | 0.8176548 | s |
| 41 | 41 | 1 | 0 | 7 | 13 | 0.8726371 | 0.0520352 | 0.9663315 | 0.7880272 | s |
| 42 | 40 | 0 | 2 | 7 | 15 | 0.8726371 | 0.0520352 | 0.9663315 | 0.7880272 | s |
| 44 | 38 | 0 | 1 | 7 | 16 | 0.8726371 | 0.0520352 | 0.9663315 | 0.7880272 | s |
| 45 | 37 | 1 | 1 | 8 | 17 | 0.8490523 | 0.0588083 | 0.9527789 | 0.7566181 | s |
| 46 | 35 | 0 | 1 | 8 | 18 | 0.8490523 | 0.0588083 | 0.9527789 | 0.7566181 | s |
| 48 | 34 | 0 | 2 | 8 | 20 | 0.8490523 | 0.0588083 | 0.9527789 | 0.7566181 | s |
| 49 | 32 | 1 | 2 | 9 | 22 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 51 | 29 | 0 | 1 | 9 | 23 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 52 | 28 | 0 | 1 | 9 | 24 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 54 | 27 | 0 | 1 | 9 | 25 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 55 | 26 | 0 | 4 | 9 | 29 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 57 | 22 | 0 | 1 | 9 | 30 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 65 | 21 | 0 | 2 | 9 | 32 | 0.8225194 | 0.0668317 | 0.9376341 | 0.7215375 | s |
| 66 | 19 | 1 | 1 | 10 | 33 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 67 | 17 | 0 | 1 | 10 | 34 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 73 | 16 | 0 | 1 | 10 | 35 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 74 | 15 | 0 | 1 | 10 | 36 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 75 | 14 | 0 | 1 | 10 | 37 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 77 | 13 | 0 | 1 | 10 | 38 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 82 | 12 | 0 | 1 | 10 | 39 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 83 | 11 | 0 | 2 | 10 | 41 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 87 | 9 | 0 | 1 | 10 | 42 | 0.7792289 | 0.0859678 | 0.9222336 | 0.6583990 | s |
| 89 | 8 | 1 | 0 | 11 | 42 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 90 | 7 | 0 | 1 | 11 | 43 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 92 | 6 | 0 | 1 | 11 | 44 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 96 | 5 | 0 | 1 | 11 | 45 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 97 | 4 | 0 | 2 | 11 | 47 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 98 | 2 | 0 | 1 | 11 | 48 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 99 | 1 | 0 | 1 | 11 | 49 | 0.6818253 | 0.1588949 | 0.9309465 | 0.4993689 | s |
| 0 | 64 | 2 | 0 | 2 | 0 | 0.9687500 | 0.0224507 | 1.0000000 | 0.9270468 | r |
| 3 | 62 | 1 | 0 | 3 | 0 | 0.9531250 | 0.0277208 | 1.0000000 | 0.9027217 | r |
| 4 | 61 | 1 | 0 | 4 | 0 | 0.9375000 | 0.0322749 | 0.9987199 | 0.8800328 | r |
| 6 | 60 | 1 | 0 | 5 | 0 | 0.9218750 | 0.0363889 | 0.9900254 | 0.8584159 | r |
| 8 | 59 | 2 | 0 | 7 | 0 | 0.8906250 | 0.0438048 | 0.9704688 | 0.8173502 | r |
| 11 | 57 | 1 | 0 | 8 | 0 | 0.8750000 | 0.0472456 | 0.9598946 | 0.7976136 | r |

```
summary(life_table2)
```

```
     time           n.risk        n.event          n.censor
Min.   : 0.00   Min.   : 1.0   Min.   :0.0000   Min.   :0.000
1st Qu.:28.00   1st Qu.:13.5   1st Qu.:0.0000   1st Qu.:0.000
Median :42.00   Median :34.0   Median :0.0000   Median :1.000
Mean   :46.25   Mean   :31.9   Mean   :0.4096   Mean   :1.084
3rd Qu.:65.50   3rd Qu.:50.0   3rd Qu.:1.0000   3rd Qu.:1.000
Max.   :99.00   Max.   :64.0   Max.   :3.0000   Max.   :5.000
  cum.event       cum.censor       estimate        std.error
Min.   : 0.00   Min.   : 0.00   Min.   :0.5232   Min.   :0.00000
1st Qu.: 7.00   1st Qu.: 4.00   1st Qu.:0.6818   1st Qu.:0.05056
Median :10.00   Median :20.00   Median :0.7929   Median :0.06683
Mean   :11.48   Mean   :20.04   Mean   :0.7757   Mean   :0.08234
3rd Qu.:14.00   3rd Qu.:34.50   3rd Qu.:0.8726   3rd Qu.:0.11068
Max.   :23.00   Max.   :49.00   Max.   :1.0000   Max.   :0.16071
  conf.high         conf.low      strata   estimate_type
Min.   :0.7169   Min.   :0.3818   sim:43   Length:83
1st Qu.:0.8869   1st Qu.:0.5099   não:40   Class :character
Median :0.9309   Median :0.6985            Mode  :character
Mean   :0.9015   Mean   :0.6716
3rd Qu.:0.9663   3rd Qu.:0.7880
Max.   :1.0000   Max.   :1.0000
estimate_type_label monotonicity_type  strata_label         conf.level
Length:83           Length:83          Length:83          Min.   :0.95
Class :character    Class :character   Class :character   1st Qu.:0.95
Mode  :character    Mode  :character   Mode  :character   Median :0.95
                                                          Mean   :0.95
                                                          3rd Qu.:0.95
                                                          Max.   :0.95
```

```
summary(life_table2, times = tempos_específicos)
```

```
     time           n.risk        n.event          n.censor
Min.   : 0.00   Min.   : 1.0   Min.   :0.0000   Min.   :0.000
1st Qu.:28.00   1st Qu.:13.5   1st Qu.:0.0000   1st Qu.:0.000
Median :42.00   Median :34.0   Median :0.0000   Median :1.000
Mean   :46.25   Mean   :31.9   Mean   :0.4096   Mean   :1.084
```

```
 3rd Qu.:65.50   3rd Qu.:50.0   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :99.00   Max.   :64.0   Max.   :3.0000   Max.   :5.000
   cum.event       cum.censor       estimate        std.error
 Min.   : 0.00   Min.   : 0.00   Min.   :0.5232   Min.   :0.00000
 1st Qu.: 7.00   1st Qu.: 4.00   1st Qu.:0.6818   1st Qu.:0.05056
 Median :10.00   Median :20.00   Median :0.7929   Median :0.06683
 Mean   :11.48   Mean   :20.04   Mean   :0.7757   Mean   :0.08234
 3rd Qu.:14.00   3rd Qu.:34.50   3rd Qu.:0.8726   3rd Qu.:0.11068
 Max.   :23.00   Max.   :49.00   Max.   :1.0000   Max.   :0.16071
   conf.high         conf.low      strata   estimate_type
 Min.   :0.7169   Min.   :0.3818   sim:43   Length:83
 1st Qu.:0.8869   1st Qu.:0.5099   não:40   Class :character
 Median :0.9309   Median :0.6985            Mode  :character
 Mean   :0.9015   Mean   :0.6716
 3rd Qu.:0.9663   3rd Qu.:0.7880
 Max.   :1.0000   Max.   :1.0000
 estimate_type_label monotonicity_type  strata_label         conf.level
 Length:83           Length:83          Length:83          Min.   :0.95
 Class :character    Class :character   Class :character   1st Qu.:0.95
 Mode  :character    Mode  :character   Mode  :character   Median :0.95
                                                           Mean   :0.95
                                                           3rd Qu.:0.95
                                                           Max.   :0.95
```

```
head(life_table2)
```

```
# A tibble: 6 x 16
   time n.risk n.event n.censor cum.event cum.censor estimate std.error
  <dbl>  <dbl>   <dbl>    <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
1     0     60       0        1         0          1    1        0
2     3     59       1        0         1          1    0.983    0.0171
3     4     58       1        0         2          1    0.966    0.0244
4    24     57       1        0         3          1    0.949    0.0301
5    26     56       1        0         4          1    0.932    0.0351
6    29     55       1        0         5          1    0.915    0.0396
# i 8 more variables: conf.high <dbl>, conf.low <dbl>, strata <fct>,
#   estimate_type <chr>, estimate_type_label <chr>, monotonicity_type <chr>,
#   strata_label <chr>, conf.level <dbl>
```

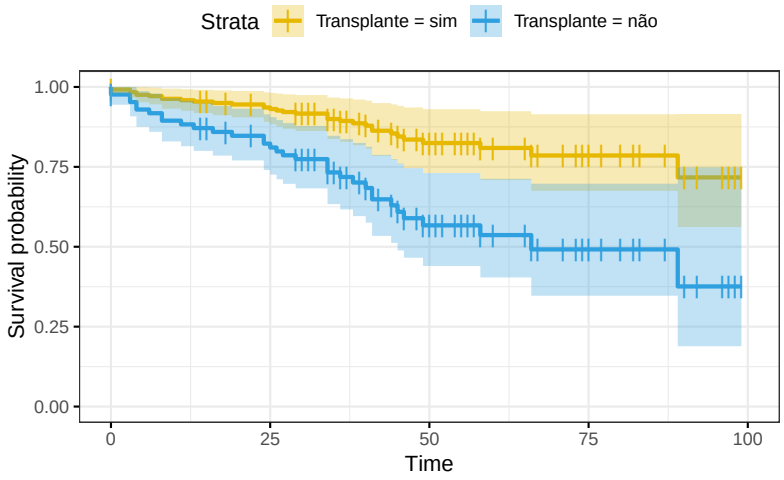
### 6.16.5 Código não usado

```
# Create the new data
new_df <- with(db,
               data.frame(tx = c("sim", "não")
               )
)
glimpse(new_df)
```

```
Rows: 2
Columns: 1
$ tx <chr> "sim", "não"
```

```
new_df$tx = as.factor(new_df$tx)


# Survival curves with new data
#%%%%%%%%%%%%%%%%%%%%%%%%%%%
fit_cox <- survfit(cox_res, newdata = new_df)


ggsurvplot(fit_cox, data = db,
           size = 1,
           palette = c('#E7B800', '#2e9fdf'),
           censor.shape = '|', censor.size = 4,
           conf.int = TRUE,
           pval = TRUE,
          # risk.table = TRUE,
         #  risk.table.col = 'strata',
           legend.labs = list('0' = 'Transplante = sim', '1' = 'Transplante = não'),
           risk.table.height = 0.25,
           ggtheme = theme_bw())
```

```
Warning in .pvalue(fit, data = data, method = method, pval = pval, pval.coord = pval.coord, : T
 This is a null model.
```




### 6.16.6 Para salvar os valores de sobrevida

```
surv_fit_cox = survfit(cox_res)

# Extrai os tempos de sobrevida e as estimativas de sobrevida
surv_df_cox <- data.frame(time = surv_fit_cox$time, surv = surv_fit_cox$surv)
surv_df_cox
```

```
   time      surv
1     0 0.9919270
2     3 0.9837071
3     4 0.9754200
4     6 0.9712508
5     8 0.9628064
6    11 0.9585298
7    13 0.9542158
8    14 0.9542158
9    15 0.9542158
10   16 0.9497436
11   18 0.9497436
12   19 0.9451662
13   22 0.9451662
14   24 0.9357670
15   25 0.9310329
```

```
16   26 0.9262516
17   27 0.9214703
18   29 0.9166403
19   30 0.9166403
20   31 0.9166403
21   32 0.9166403
22   34 0.8995899
23   35 0.8995899
24   36 0.8936099
25   37 0.8936099
26   38 0.8862225
27   39 0.8862225
28   40 0.8787730
29   41 0.8630231
30   42 0.8630231
31   44 0.8544152
32   45 0.8449676
33   46 0.8354131
34   48 0.8354131
35   49 0.8245091
36   50 0.8245091
37   51 0.8245091
38   52 0.8245091
39   54 0.8245091
40   55 0.8245091
41   56 0.8245091
42   57 0.8245091
43   58 0.8091983
44   60 0.8091983
45   65 0.8091983
46   66 0.7855423
47   67 0.7855423
48   71 0.7855423
49   73 0.7855423
50   74 0.7855423
51   75 0.7855423
52   77 0.7855423
53   80 0.7855423
54   82 0.7855423
55   83 0.7855423
56   87 0.7855423
```

```
57   89 0.7169271
58   90 0.7169271
59   92 0.7169271
60   96 0.7169271
61   97 0.7169271
62   98 0.7169271
63   99 0.7169271
```

## 6.17 Referências


https://bookdown.org/mpfoley1973/survival/semiparametric.html#fitting-the-model-1

https://biostatsquid.com/easy-survival-analysis-r-tutorial/

https://www.youtube.com/watch?v=XrvCCFQRCZE

https://www.youtube.com/watch?v=vX3l36ptrTU&list=PLqzoL9-eJTNDdnKvep_YHIwk2AMqHhuJ0

http://www.sthda.com/english/wiki/cox-proportional-hazards-model


## 6.18 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages effectsize
(version 0.8.6; Ben-Shachar MS et al., 2020), lpSolve (version 5.6.19;
Berkelaar M, others, 2023), survMisc (version 0.5.6; Dardis C, 2022), tm
(version 0.7.11; Feinerer I, Hornik K, 2023), flexplot (version 0.20.5; Fife D,
2024), lubridate (version 1.9.3; Grolemund G, Wickham H, 2011), coxphf (version
1.13.4; Heinze G et al., 2023), NLP (version 0.2.1; Hornik K, 2020), coin
(version 1.4.3; Hothorn T et al., 2006), ggpubr (version 0.6.0; Kassambara A,
2023), survminer (version 0.4.9; Kassambara A et al., 2021),
PHInfiniteEstimates (version 2.9.5; Kolassa JE, Zhang J, 2023), parameters
(version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke
D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see
```

(version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), condSURV (version 2.0.4; Meira-Machado L, Sestelo M, 2023), tibble (version 3.2.1; Müller K, Wickham H, 2023), datawizard (version 0.9.0; Patil I et al., 2022), foreign (version 0.8.85; R Core Team, 2023), nph (version 2.1; Ristl R et al., 2021), broom (version 1.0.5; Robinson D et al., 2023), ggsurvfit (version 1.0.0; Sjoberg D et al., 2023), gtsummary (version 1.7.2; Sjoberg D et al., 2021), rempsyc (version 0.1.6; Thériault R, 2023), survival (version 3.5.7; Therneau T, 2023), ggplot2 (version 3.4.4; Wickham H, 2016), forcats (version 1.0.0; Wickham H, 2023), stringr (version 1.5.1; Wickham H, 2023), tidyverse (version 2.0.0; Wickham H et al., 2019), dplyr (version 1.1.3; Wickham H et al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), readr (version 2.1.4; Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et al., 2023) and kableExtra (version 1.3.4; Zhu H, 2021).

References
----------


- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Berkelaar M, others (2023). _lpSolve: Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs_. R package version 5.6.19, <https://CRAN.R-project.org/package=lpSolve>.
- Dardis C (2022). _survMisc: Miscellaneous Functions for Survival Data_. R package version 0.5.6, <https://CRAN.R-project.org/package=survMisc>.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate." _Journal of Statistical Software_, *40*(3), 1-25. <https://www.jstatsoft.org/v40/i03/>.
- Heinze G, Ploner M, Jiricka L, Steiner G (2023). _coxphf: Cox Regression with Firth's Penalized Likelihood_. R package version 1.13.4, <https://CRAN.R-project.org/package=coxphf>.

- Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.
- Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2006). "A Lego system for conditional inference." _The American Statistician_, *60*(3), 257-263. doi:10.1198/000313006X118430 <https://doi.org/10.1198/000313006X118430>.
Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2008). "Implementing a class of permutation tests: The coin package." _Journal of Statistical Software_, *28*(8), 1-23. doi:10.18637/jss.v028.i08 <https://doi.org/10.18637/jss.v028.i08>.
- Kassambara A (2023). _ggpubr: 'ggplot2' Based Publication Ready Plots_. R package version 0.6.0, <https://CRAN.R-project.org/package=ggpubr>.
- Kassambara A, Kosinski M, Biecek P (2021). _survminer: Drawing Survival Curves using 'ggplot2'_. R package version 0.4.9, <https://CRAN.R-project.org/package=survminer>.
- Kolassa JE, Zhang J (2023). _PHInfiniteEstimates: Tools for Inference in the Presence of a Monotone Likelihood_. R package version 2.9.5, <https://CRAN.R-project.org/package=PHInfiniteEstimates>.
- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.
- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.
- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of

Model-Based Predictions, Contrasts and Means." _CRAN_.
<https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023).
"Automated Results Reporting as a Practical Tool to Improve Reproducibility and
Methodological Best Practices Adoption." _CRAN_.
<https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022).
"correlation: Methods for Correlation Analysis." Version 0.8.3,
<https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M,
Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in
R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306
<https://doi.org/10.21105/joss.02306>,
<https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Meira-Machado L, Sestelo M (2023). _condSURV: Estimation of the Conditional
Survival Function for Ordered Multivariate Failure Time Data_. R package
version 2.0.4, <https://CRAN.R-project.org/package=condSURV>.
  - Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version
3.2.1, <https://CRAN.R-project.org/package=tibble>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022).
"datawizard: An R Package for Easy Data Preparation and Statistical
Transformations." _Journal of Open Source Software_, *7*(78), 4684.
doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
  - R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS',
'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85,
<https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical
Computing_. R Foundation for Statistical Computing, Vienna, Austria.
<https://www.R-project.org/>.
  - Ristl R, Ballarini N, Götte H, Schüler A, Posch M, König F (2021). "Delayed
treatment effects, treatment switching and heterogeneous patient populations:
How to design and analyze RCTs in oncology." _Pharmaceutical statistics_,
*20*(1), 129-145.
  - Robinson D, Hayes A, Couch S (2023). _broom: Convert Statistical Objects into
Tidy Tibbles_. R package version 1.0.5,
<https://CRAN.R-project.org/package=broom>.
  - Sjoberg D, Baillie M, Fruechtenicht C, Haesendonckx S, Treis T (2023).
_ggsurvfit: Flexible Time-to-Event Figures_. R package version 1.0.0,
<https://CRAN.R-project.org/package=ggsurvfit>.
  - Sjoberg D, Whiting K, Curry M, Lavery J, Larmarange J (2021). "Reproducible
Summary Tables with the gtsummary Package." _The R Journal_, *13*, 570-580.
doi:10.32614/RJ-2021-053 <https://doi.org/10.32614/RJ-2021-053>,

<https://doi.org/10.32614/RJ-2021-053>.
- Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
- Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
- Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
- Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
- Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
- Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
- Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.
- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
- Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4, <https://CRAN.R-project.org/package=kableExtra>.

# 7 Lista 6.1 - Cox tempo dependente

O banco de dados Cox_tempo_dependente1.sav apresenta os dados de 628 pacientes de um serviço de nefrologia. Os pacientes foram acompanhados por pouco mais de 1000 dias e durante este período alguns pacientes fizeram transplante renal (treat=1). Gostariamos de saber se a realização de transplante aumenta significativamente a sobrevida destes pacientes em relação àqueles que não fizeram. Com base nestas informações, responda as questões abaixo

## 7.1 Carregando pacotes

```
library(tidyverse)
library(flexplot)
library(foreign)
library(dplyr)
library(tm)
library(ggplot2)
library(forcats)
library(rempsyc)
library(easystats)
library(kableExtra)
library(gtsummary)

#Específicos para survival
library(survival)
library(ggsurvfit)
library(survminer)
library(broom)
library(survMisc)
```

```
library(PHInfiniteEstimates)
library(coin)
library(condSURV)
```

## 7.2 Carregando os dados e modificando o tipo de variável

Mantendo as boas práticas das análises, logo após carregar os dados em uma variável, vamos verificar os tipos de variávels que temos em nosso banco.

```
original = read.spss("Cox tempo dependente 2_1.sav", to.data.frame=TRUE)
glimpse(original)
```

```
Rows: 628
Columns: 7
$ ID            <dbl> 112, 91, 113, 150, 22, 139, 104, 590, 189, 171, 338, 492~
$ age           <dbl> 35, 33, 35, 31, 33, 43, 25, 35, 29, 27, 35, 23, 24, 23, ~
$ race          <fct> branco, branco, branco, branco, branco, branco, branco, ~
$ treat         <dbl> 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,~
$ Tempo_dialise <dbl> 51, 67, 88, 156, 12, 139, 90, 25, 187, 34, 29, 78, 63, 8~
$ time          <dbl> 1172, 762, 734, 720, 659, 658, 655, 654, 634, 630, 621, ~
$ morte         <fct> Não, Não, Não, Não, Não, Não, Não, Não, Não, Não, Não, N~
```

Novamente podemos observar que o evento de interesse (morte) está como um fator. Vamos modificar como já fizemos a lista 6 e também já vamos ajustar a variável "treat" para que ela seja um fator e não um número.

```
db <- original %>%
  mutate(
    morte = as.integer(morte == "Sim"), # para transformar sim e não em 1 e 0, respectivamen
    treat = as.factor(treat)
  )
glimpse(db)
```

```
Rows: 628
Columns: 7
$ ID            <dbl> 112, 91, 113, 150, 22, 139, 104, 590, 189, 171, 338, 492~
$ age           <dbl> 35, 33, 35, 31, 33, 43, 25, 35, 29, 27, 35, 23, 24, 23, ~
$ race          <fct> branco, branco, branco, branco, branco, branco, branco, ~
$ treat         <fct> 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,~
$ Tempo_dialise <dbl> 51, 67, 88, 156, 12, 139, 90, 25, 187, 34, 29, 78, 63, 8~
$ time          <dbl> 1172, 762, 734, 720, 659, 658, 655, 654, 634, 630, 621, ~
$ morte         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

Feito! Vamos também verificar se há presença de dados faltantes e em quais variáveis.

```
# Verificando NAs
resumo_nas <- db %>%
  summarise(
    nas_age = sum(is.na(age)),
    nas_race = sum(is.na(race)),
    nas_treat = sum(is.na(treat)),
    nas_t_dialise = sum(is.na(Tempo_dialise)),
    nas_time = sum(is.na(time)),
    nas_morte = sum(is.na(morte)),
  )
kable(resumo_nas)
```

| nas_age | nas_race | nas_treat | nas_t_dialise | nas_time | nas_morte |
|---|---|---|---|---|---|
| 5 | 6 | 0 | 0 | 0 | 0 |

Até chegar na Cox tempo dependente, vamos repetir basicamente o que já fizemos no Capítulo 6

## 7.3 Criando a estrutura de dados

```
# Create a survival object
surv_obj <- Surv(time = db$time, event = db$morte)
```

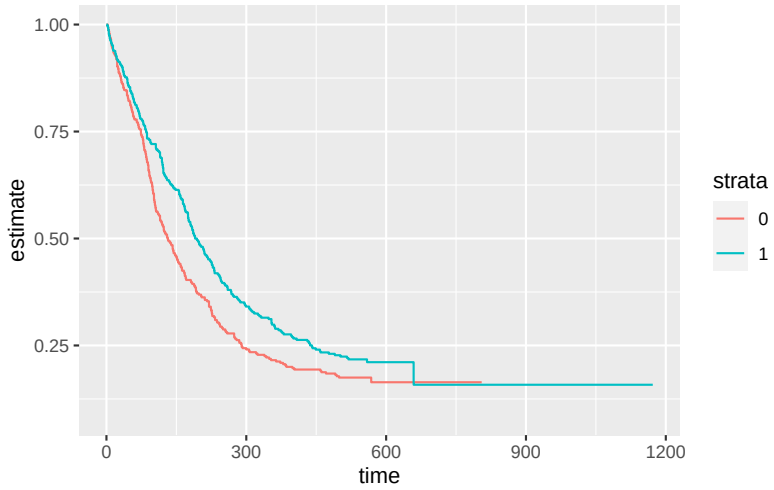
### 7.3.1 Tábua de vida

```
# Create survival curve
fit1 <- survfit(surv_obj ~ treat, data = db)
kable(head(tidy(fit1)))
```

| time | n.risk | n.event | n.censor | estimate | std.error | conf.high | conf.low | strata |
|---|---|---|---|---|---|---|---|---|
| 3 | 320 | 2 | 0 | 0.993750 | 0.0044333 | 1.0000000 | 0.9851526 | treat=0 |
| 4 | 318 | 2 | 0 | 0.987500 | 0.0062894 | 0.9997483 | 0.9754017 | treat=0 |
| 5 | 316 | 1 | 0 | 0.984375 | 0.0070430 | 0.9980575 | 0.9708801 | treat=0 |
| 6 | 315 | 2 | 0 | 0.978125 | 0.0083599 | 0.9942837 | 0.9622289 | treat=0 |
| 7 | 313 | 2 | 0 | 0.971875 | 0.0095097 | 0.9901593 | 0.9539283 | treat=0 |
| 8 | 311 | 1 | 0 | 0.968750 | 0.0100402 | 0.9880024 | 0.9498728 | treat=0 |

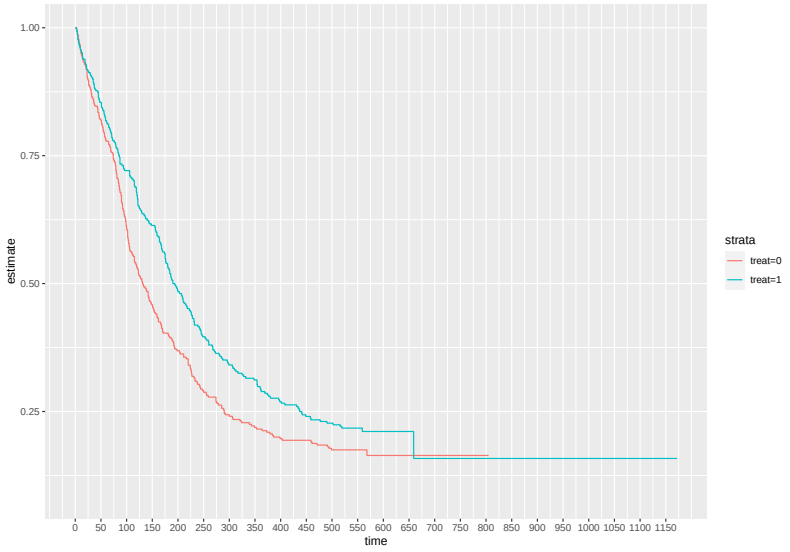
### 7.3.2 Gráfico Kaplan-Meir

```
km_plot = survfit2(surv_obj ~ treat, data = db) %>%
  tidy_survfit() %>%
  ggplot(aes(x = time, y = estimate,
             min = conf.low, ymax = conf.low,
             color = strata, fill = strata)) +
  geom_step()
km_plot
```



Podemos ajustar as configurações do eixo X para exibir uma escala temporal com intervalos de 50 unidades.

```
# km_plot2 = fit1 %>%
#   tidy_survfit() %>%
#   ggplot(aes(x = time, y = estimate,
#              min = conf.low, ymax = conf.low,
#              color = strata, fill = strata)) +
#   geom_step()

km_plot2 = fit1 %>%
  tidy_survfit() %>%
  ggplot(aes(x = time, y = estimate,
             min = conf.low, ymax = conf.low,
             color = strata, fill = strata)) +
  geom_step() +
  scale_x_continuous(breaks = seq(0, max(fit1$time), by = 50))


km_plot2
```



### 7.3.3 Tabela com Sobrevida em tempos espcíficos.

```
tbl_survfit_ex3 <-
  list(
    survfit(surv_obj ~ 1, db),
    survfit(surv_obj ~ treat, db)
  ) %>%
  tbl_survfit(times = c(100, 600))
tbl_survfit_ex3
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

| **Characteristic** | **Time 100** | **Time 600** |
|---|---|---|
| Overall | 66% (63%, 70%) | 19% (16%, 22%) |
| treat | | |
| 0 | 61% (56%, 66%) | 16% (13%, 21%) |
| 1 | 72% (67%, 77%) | 21% (17%, 26%) |

### 7.3.4 Log-rank

```
coin::logrank_test(surv_obj ~ treat, data = db, type = "logrank" ) # padrão é o log-rank
```

```

    Asymptotic Two-Sample Logrank Test

data:  surv_obj by treat (0, 1)
Z = -2.5984, p-value = 0.009365
alternative hypothesis: true theta is not equal to 1
```

### 7.3.5 Gehan-Breslow

```
coin::logrank_test(surv_obj ~ treat ,data = db, type = "Gehan-Breslow")
```

```
    Asymptotic Two-Sample Gehan-Breslow Test

data:  surv_obj by treat (0, 1)
Z = -3.0713, p-value = 0.002132
alternative hypothesis: true theta is not equal to 1
```

### 7.3.6 Tarone-Ware

```
coin::logrank_test(surv_obj ~ treat ,data = db, type = "Tarone-Ware")
```

```
    Asymptotic Two-Sample Tarone-Ware Test

data:  surv_obj by treat (0, 1)
Z = -2.9622, p-value = 0.003055
alternative hypothesis: true theta is not equal to 1
```

### 7.3.7 Peto-Peto

```
coin::logrank_test(surv_obj ~ treat ,data = db, type = "Peto-Peto")
```

```
    Asymptotic Two-Sample Peto-Peto Test

data:  surv_obj by treat (0, 1)
Z = -3.0608, p-value = 0.002207
alternative hypothesis: true theta is not equal to 1
```

## 7.4 Cox regression

```
# Cox regression ======================================================
# Fit the model

cox_res <- coxph(Surv(time = db$time, event = db$morte) ~ treat, data = db)
```

```
### Para testar todas as variáveis
#cox_res <- coxph(Surv(time = db$time, event = db$morte2) ~ treat + age + Tempo_dialise, dat

tbl_regression(cox_res, exponentiate = TRUE)
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

| **Characteristic** | **HR** | **95% CI** | **p-value** |
|---|---|---|---|
| treat | | | |
| 0 | — | — | |
| 1 | 0.79 | 0.67, 0.94 | 0.009 |

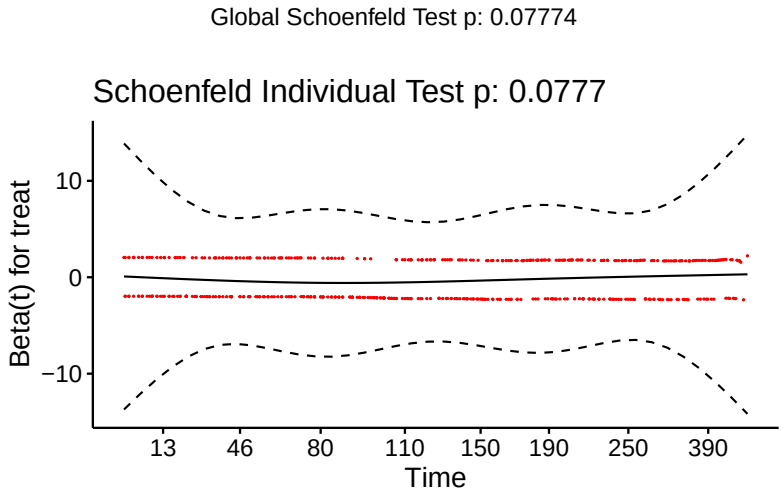
### 7.4.1 Verificando os pressupostos da Cox regression

Relembrando a análise dos riscos proporcionais com base nos resíduos de Schoenfeld:

- p-val $<$ 0,05: há evidências **contra** a pressuposto de riscos proporcionais, os HRs **não** são constantes ao longo do tempo
- chisq: quanto maior o valor, mais forte a violação dos pressupostos

## 7.5 Plot dos resíduos de Schoenfeld

```
# Plot the Schoenfeld residuals over time for each covariate
survminer::ggcoxzph(cox.zph(cox_res), point.size = 0.1)
```




Se os resíduos mostrarem um padrão claro ao longo do tempo, isso pode indicar uma violação da suposição de riscos proporcionais.

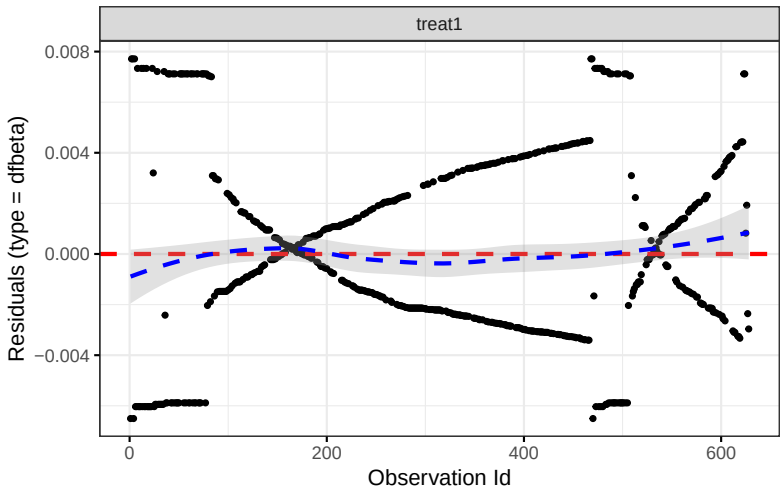
Algumas dicas para ajudar na interpretação:

- Sem Padrão (Resíduos Constantes): Se os resíduos aparecerem aleatoriamente espalhados em torno de zero, sem nenhuma tendência ou padrão claro, isso sugere que a suposição de riscos proporcionais é razoável.
- Tendência Linear: Uma tendência linear (aumentando ou diminuindo) nos resíduos ao longo do tempo pode sugerir uma violação da suposição de riscos proporcionais. Por exemplo, se os resíduos forem consistentemente positivos ou negativos ao longo do tempo, isso indica um efeito dependente do tempo.
- Padrão Não Linear: Se os resíduos exibirem um padrão não linear ou formatos específicos (por exemplo, formato de U, formato de V), isso pode indicar desvios dos riscos proporcionais.
- Paralelismo: Paralelismo significa que a propagação e distribuição dos resíduos são relativamente constantes ao longo do tempo. Se os resíduos aumentarem ou diminuirem ao longo do tempo, isso pode sugerir uma violação da suposição.

```
ggcoxdiagnostics(cox_res, type = "dfbeta", linear.predictions = FALSE)
```

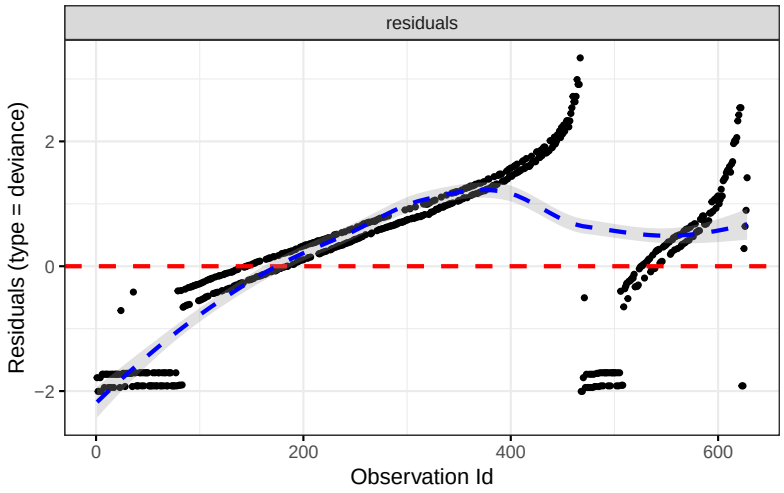
```
Warning: `gather_()` was deprecated in tidyr 1.2.0.
i Please use `gather()` instead.
i The deprecated feature was likely used in the survminer package.
  Please report the issue at <https://github.com/kassambara/survminer/issues>.
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
# Não é importante para variáveis categóricas, mas fica o código para eventual consulta.
ggcoxdiagnostics(cox_res, type = "deviance", linear.predictions = FALSE)
```

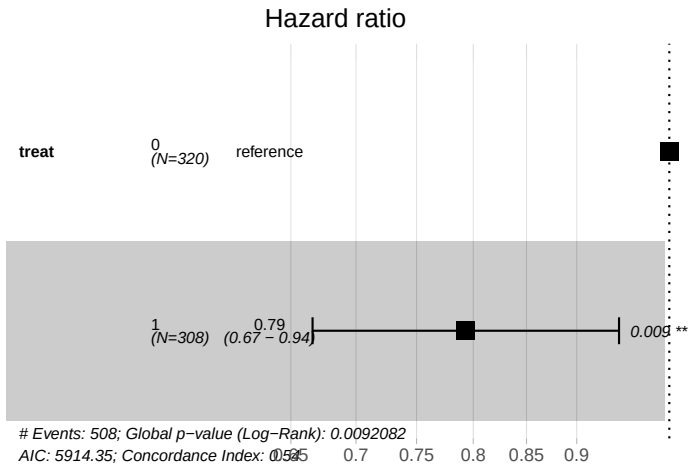
```
`geom_smooth()` using formula = 'y ~ x'
```




## 7.6 Plots do modelo

### 7.6.1 Forest plot

```
# Forest plots ====================================================================
# Visualise your Cox model results
ggforest(cox_res, data = db)
```



### 7.6.2 Gráfico de sobrevida

Assim como fizemos no exercício anterior, precisamos criar um novo banco de dados para visualizar o gráfico da Regressão de Cox:

```
# Precisa ser feito apenas com uma variável
cox_res2 <- coxph(Surv(time = db$time, event = db$morte) ~ treat, data = db)

# Criando o novo banco de dados
new_df <- with(db,
               data.frame(treat = c("0", "1"))
)
```

E precisamos transformar a variável `treat` em um fator.

```
new_df$treat = as.factor(new_df$treat)
kable(new_df)
```

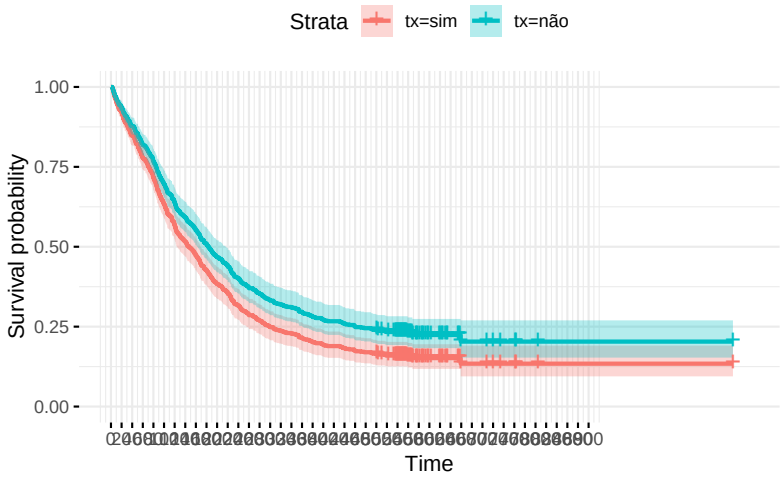
| treat |
|---|
| 0 |
| 1 |

Criando os dados com base no modelo e plotando o gráfico.

```
fit_cox <- survfit(cox_res2, newdata = new_df)

J = ggsurvplot(fit_cox, conf.int = TRUE, legend.labs=c("tx=sim", "tx=não"),
           ggtheme = theme_minimal(),
           data = db)

J$plot = J$plot +
  scale_x_continuous(breaks = seq(0, 900, 20))
```

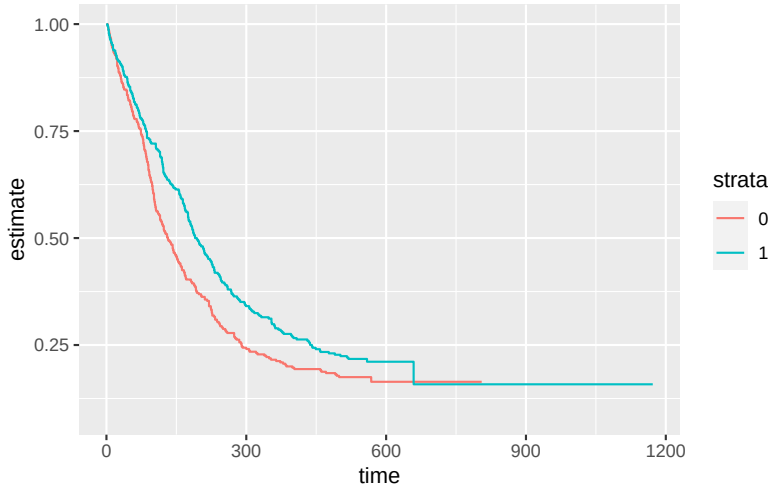
```
Scale for x is already present.
Adding another scale for x, which will replace the existing scale.
```

```
J
```




Reparem na distorção do gráfico em relação à Kaplan-Meir

```
km_plot
```




## 7.7 Cox tempo-dependente

Já vimos que os riscos não são proporcionais neste caso. Porém, nem tudo está perdido. Podemos finalmente agora falar da Cox

Tempo-dependente.

O primeiro passo é identificar um possível fator que esteja afetando a proporcionalidade dos riscos no estudo. Pela literatura tempos que o tempo em diálise afeta os riscos entre pessoas que fizeram ou não o transplante de rim. Daí a importância de entender bem o fenômeno que estamos estudando. Como bons pesquisadores, também coletamos o tempo em diálise e esses dados estão no banco de dados

```
glimpse(db$Tempo_dialise)
```

```
num [1:628] 51 67 88 156 12 139 90 25 187 34 ...
```

A variável é numérica e contínua, logo ela já está formatada para continuarmos com a análise.

Não existe regras escritas na pedra para contornar o problema de não proporcionalidade. Vamos mostrar uma abordagem aqui. Não deixe de ver as referências para outros casos.

### 7.7.1 Covariáveis tempo dependente

No R, há diversas formas de indicar uma variável como tempo-dependente. A escolha do método dependerá da natureza da variável independente e da sua relação teórica com o evento em estudo. A função `coxph()` oferece a opção de utilizar o argumento `tt()`, o qual especifica qual variável independente será considerada uma covariável tempo-dependente e como o coeficiente associado a ela deve ser modificado ao longo do tempo.

O modelo deve seguir a seguinte estrutura

```
coxph(Surv(time, event) ~ covariavel1 + covariavel2
+ tt(covariavel2), data, tt=function(x,t,…) x*t)
```

Podemos substituir o `Surv(time, event)` pela variável que salvamos com o objeto survival, `surv_obj`.

A função tt (`function(x,t,…)___`) pode assumir alguns modelos. A seguir trazemos três exemplos mais utilizados em diversas análises:

- x*t permitirá que o coeficiente mude linearmente com o tempo
- x*log(t) permite que o coeficiente mude com o log do tempo
- x*(t>tempo) permite que o coeficiente assuma 2 valores diferentes, um valor quando t<=tempo e outro valor t>tempo

Vamos gerar vários modelos e avaliá-los comparando os índices de ajuste e os resultados obtidos.

### 7.7.2 Sem variável tempo dependente

```
dialise <- coxph(surv_obj ~ treat + Tempo_dialise,
                          data=db) # corte no 660

summary(dialise)
```

```
Call:
coxph(formula = surv_obj ~ treat + Tempo_dialise, data = db)

  n= 628, number of events= 508

                    coef  exp(coef)   se(coef)       z            Pr(>|z|)
treat1         0.0618983  1.0638541  0.0899990   0.688               0.492
Tempo_dialise -0.0084493  0.9915863  0.0007709 -10.960 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
treat1           1.0639      0.940    0.8918    1.2691
Tempo_dialise    0.9916      1.008    0.9901    0.9931

Concordance= 0.75  (se = 0.012 )
Likelihood ratio test= 151.4  on 2 df,   p=<0.0000000000000002
Wald test            = 121.2  on 2 df,   p=<0.0000000000000002
Score (logrank) test = 120.6  on 2 df,   p=<0.0000000000000002
```

### 7.7.3 Mudança linear

```
dialise_linear <- coxph(surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
                        data=db,
                          tt=function(x,t,...) x*t)
summary(dialise_linear)
```

```
Call:
coxph(formula = surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
    data = db, tt = function(x, t, ...) x * t)

  n= 628, number of events= 508

                          coef    exp(coef)     se(coef)       z
treat1             -0.016001236  0.984126103  0.091676088  -0.175
Tempo_dialise      -0.023777102  0.976503346  0.001506128 -15.787
tt(Tempo_dialise)  0.000070699  1.000070701  0.000005184  13.637
                             Pr(>|z|)
treat1                          0.861
Tempo_dialise      <0.0000000000000002 ***
tt(Tempo_dialise) <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
treat1               0.9841     1.0161    0.8223    1.1778
Tempo_dialise        0.9765     1.0241    0.9736    0.9794
tt(Tempo_dialise)    1.0001     0.9999    1.0001    1.0001

Concordance= 0.761  (se = 0.009 )
Likelihood ratio test= 337.6  on 3 df,   p=<0.0000000000000002
Wald test            = 249.3  on 3 df,   p=<0.0000000000000002
Score (logrank) test = 229.1  on 3 df,   p=<0.0000000000000002
```

### 7.7.4 Modelo log

```
dialise_log <- coxph(surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
                         data=db,
                         tt=function(x,t,...) x*log(t))

summary(dialise_log)
```

```
Call:
coxph(formula = surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
    data = db, tt = function(x, t, ...) x * log(t))

  n= 628, number of events= 508

                      coef exp(coef)  se(coef)      z            Pr(>|z|)
treat1            -0.071106  0.931363  0.092369  -0.77               0.441
Tempo_dialise     -0.097417  0.907178  0.005942 -16.40 <0.0000000000000002 ***
tt(Tempo_dialise)  0.017178  1.017326  0.001087  15.80 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
treat1               0.9314      1.074    0.7771    1.1162
Tempo_dialise        0.9072      1.102    0.8967    0.9178
tt(Tempo_dialise)    1.0173      0.983    1.0152    1.0195

Concordance= 0.759  (se = 0.01 )
Likelihood ratio test= 461.5  on 3 df,   p=<0.0000000000000002
Wald test            = 284.6  on 3 df,   p=<0.0000000000000002
Score (logrank) test = 223.9  on 3 df,   p=<0.0000000000000002
```

### 7.7.5 Modelo temporal

```
dialise_tempo_650 <- coxph(surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
                         data=db,
                         tt=function(x,t,...) x*(t>650))

summary(dialise_tempo_650)
```

```
Call:
coxph(formula = surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
    data = db, tt = function(x, t, ...) x * (t > 650))

  n= 628, number of events= 508

                       coef  exp(coef)   se(coef)       z            Pr(>|z|)
treat1             0.0617615  1.0637086  0.0900051   0.686               0.493
Tempo_dialise     -0.0084524  0.9915832  0.0007713 -10.958 <0.0000000000000002
tt(Tempo_dialise)  0.0028519  1.0028560  0.0221016   0.129               0.897

treat1
Tempo_dialise     ***
tt(Tempo_dialise)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
treat1               1.0637     0.9401    0.8917    1.2689
Tempo_dialise        0.9916     1.0085    0.9901    0.9931
tt(Tempo_dialise)    1.0029     0.9972    0.9603    1.0473

Concordance= 0.75  (se = 0.01 )
Likelihood ratio test= 151.5  on 3 df,   p=<0.0000000000000002
Wald test            = 121.2  on 3 df,   p=<0.0000000000000002
Score (logrank) test = 120.6  on 3 df,   p=<0.0000000000000002
```

### 7.7.6 Índices de aderência (AIC e BIC)

Podemos comparar os modelos computando os valores de AIC
e BIC

```r
combined_df <- data.frame(
  Model = c("dialise", "dialise_linear", "dialise_log", "dialise_tempo_650"),
  AIC = c(AIC(dialise), AIC(dialise_linear), AIC(dialise_log), AIC(dialise_tempo_650)),
  BIC = c(BIC(dialise), BIC(dialise_linear), BIC(dialise_log), BIC(dialise_tempo_650))
)

kable(combined_df)
```

| Model | AIC | BIC |
|---|---|---|
| dialise | 5771.688 | 5780.149 |
| dialise_linear | 5587.564 | 5600.256 |
| dialise_log | 5463.651 | 5476.343 |
| dialise_tempo_650 | 5773.672 | 5786.363 |

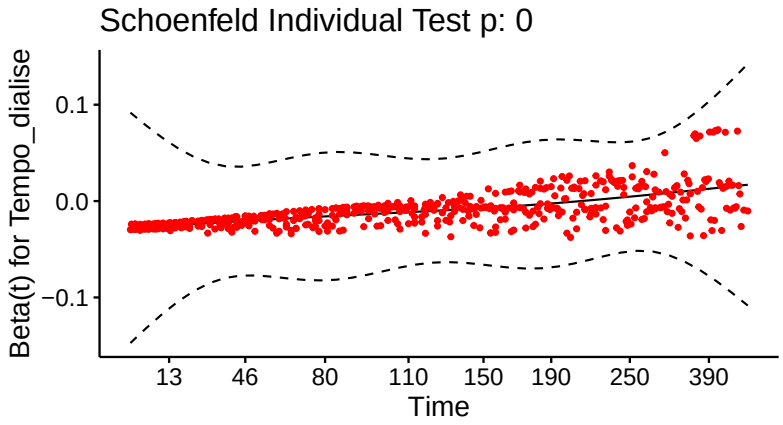
Por esse critério, temos que o melhor modelo é o log em seguida o linear.

### 7.7.7 Resíduos de Schoenfeld

Agora vamos analisar mais uma vez os resíduos de Schoenfeld, mas agora variando pelo "Tempo em Diálise".

```
cox_res_T_Cov <- coxph(Surv(time = db$time, event = db$morte) ~ treat + Tempo_dialise, data
ggcoxzph(cox.zph(cox_res_T_Cov), var ="Tempo_dialise")
```

nfeld Test p: 0.000000000000000000000000000000000000000000000000




Podemos observar que o Beta do tempo em diálise tem um aumento linear conforme maior o tempo. O resultado pode indicar que o efeito do tempo sobre a o tempo em diálise pode ser linear.

### 7.7.8 Interpretando os resultados.

A interpretação dos coeficientes da Cox Tempo-dependente é diferente das outras regressõs.

Vamos interpretar o valor do modelo com mudança linear.

```
summary(dialise_linear)
```

```
Call:
coxph(formula = surv_obj ~ treat + Tempo_dialise + tt(Tempo_dialise),
    data = db, tt = function(x, t, ...) x * t)

  n= 628, number of events= 508

                          coef    exp(coef)     se(coef)       z
treat1            -0.016001236  0.984126103  0.091676088  -0.175
Tempo_dialise     -0.023777102  0.976503346  0.001506128 -15.787
tt(Tempo_dialise)  0.000070699  1.000070701  0.000005184  13.637
                             Pr(>|z|)
treat1                          0.861
Tempo_dialise     <0.0000000000000002 ***
tt(Tempo_dialise) <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
treat1               0.9841     1.0161    0.8223    1.1778
Tempo_dialise        0.9765     1.0241    0.9736    0.9794
tt(Tempo_dialise)    1.0001     0.9999    1.0001    1.0001

Concordance= 0.761  (se = 0.009 )
Likelihood ratio test= 337.6  on 3 df,   p=<0.0000000000000002
Wald test            = 249.3  on 3 df,   p=<0.0000000000000002
Score (logrank) test = 229.1  on 3 df,   p=<0.0000000000000002
```

O coeficiente Tempo_dialise = -0.023, deve ser interpretado como o efeito do tempo de diálise no tempo zero. Já o coeficiente tt(Tempo_dialise) = deve ser interpretado como o a mudança do efeito do tempo em diálise a cada unidade de tempo a mais.

### 7.7.9 Observações SPSS e R

Na aula prática o modelo não é feito com o Tempo em Diálise fora da variável tempo dependente. Já na aula teórica do curso II de 2023, o modelo é escrito como foi feito aqui no R, levando em conta o Tempo em Diálise como uma variável tempo dependente e também como covariável no modelo.

## 7.8 Covariando para idade e raça

O conjunto de dados ainda possui duas variáveis que não foram incluídas no modelo: idade e raça. Conforme o procedimento padrão, vamos examinar a natureza dessas variáveis. Começando com a idade.

```
glimpse(db)
```

```
Rows: 628
Columns: 7
$ ID            <dbl> 112, 91, 113, 150, 22, 139, 104, 590, 189, 171, 338, 492~
$ age           <dbl> 35, 33, 35, 31, 33, 43, 25, 35, 29, 27, 35, 23, 24, 23, ~
$ race          <fct> branco, branco, branco, branco, branco, branco, branco, ~
$ treat         <fct> 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1,~
$ Tempo_dialise <dbl> 51, 67, 88, 156, 12, 139, 90, 25, 187, 34, 29, 78, 63, 8~
$ time          <dbl> 1172, 762, 734, 720, 659, 658, 655, 654, 634, 630, 621, ~
$ morte         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

Ótimo, idade já está como uma variável numérica e contínua e raça está como um fator. Por fim, vamos verificar qual o nível de referência da variável "race".

```
levels(db$race)
```

```
[1] "branco"      "negro/pardo"
```

O nível "branco" está como referência, logo, os resultados do modelo mostrarão os valores dos coeficientes do nível "negro/pardo" em relação ao nível "branco".

Vamos ao modelo completo.

### 7.8.1 Modelo completo Cox tempo dependente

```
cox_full_model <- coxph(surv_obj ~ age + race + treat + Tempo_dialise + tt(Tempo_dialise),
                        data=db,
                          tt=function(x,t,...) x*t)
summary(cox_full_model)
```

```
Call:
coxph(formula = surv_obj ~ age + race + treat + Tempo_dialise +
    tt(Tempo_dialise), data = db, tt = function(x, t, ...) x *
    t)

  n= 617, number of events= 500
   (11 observations deleted due to missingness)

                          coef    exp(coef)     se(coef)       z
age                -0.005531327  0.994483942  0.007272881  -0.761
racenegro/pardo    -0.342009991  0.710341107  0.108147939  -3.162
treat1              0.042207387  1.043110784  0.093368582   0.452
Tempo_dialise      -0.023671737  0.976606241  0.001511519 -15.661
tt(Tempo_dialise)   0.000068807  1.000068809  0.000005196  13.242
                               Pr(>|z|)
age                             0.44693
racenegro/pardo                 0.00156 **
treat1                          0.65123
Tempo_dialise     < 0.0000000000000002 ***
tt(Tempo_dialise) < 0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                  exp(coef) exp(-coef) lower .95 upper .95
age                  0.9945     1.0055    0.9804    1.0088
racenegro/pardo      0.7103     1.4078    0.5747    0.8781
treat1               1.0431     0.9587    0.8687    1.2526
Tempo_dialise        0.9766     1.0240    0.9737    0.9795
tt(Tempo_dialise)    1.0001     0.9999    1.0001    1.0001
```

```
Concordance= 0.765  (se = 0.009 )
Likelihood ratio test= 343.2  on 5 df,   p=<0.0000000000000002
Wald test            = 259.4  on 5 df,   p=<0.0000000000000002
Score (logrank) test = 243.3  on 5 df,   p=<0.0000000000000002
```

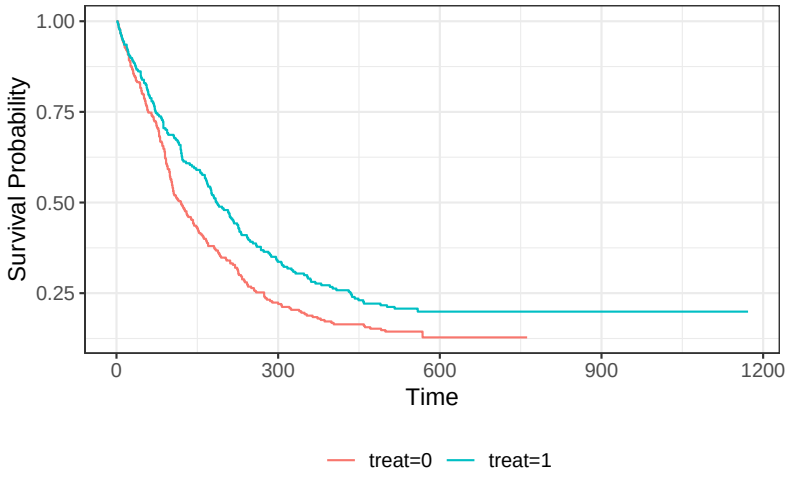
Agora temos que a raça tem um efeito significativo no modelo. Seguindo o vídeo da aula prática, podemos segmentar o banco de dados para as duas raças que temos no banco de dados.

### 7.8.2 Segmentando o banco de dados por raça

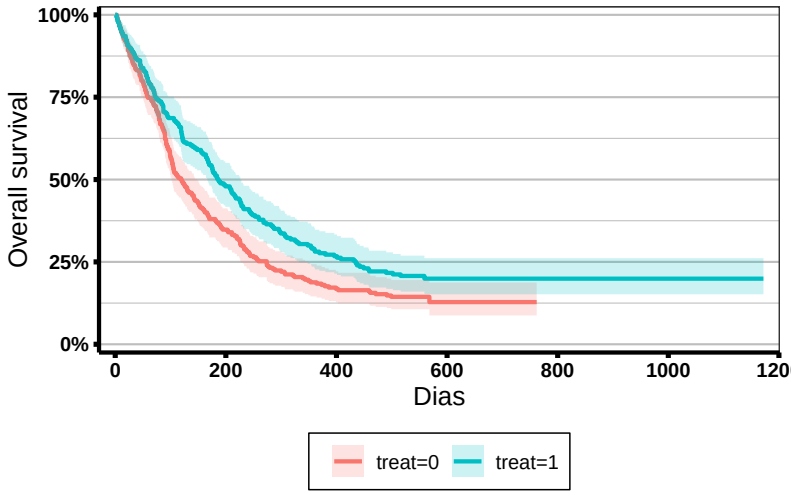
```
db_branco = db %>%
  filter(race == "branco")

db_pardo_negro = db %>%
  filter(race == "negro/pardo")
```

### 7.8.3 KM por raça = Branco

```
# Criando um novo objeto Surv
surv_obj_branco <- Surv(time = db_branco$time, event = db_branco$morte)

fit_br = survfit(surv_obj_branco ~ treat, data = db_branco)
ggsurvfit(fit_br)
```




```
ggsurvfit(fit_br, linewidth = 1) +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
 # add_risktable() +
  scale_ggsurvfit() +
  biostatsquid_theme #+  coord_cartesian(xlim = c(0, 8))
```



### 7.8.4 Modelo completo para brancos

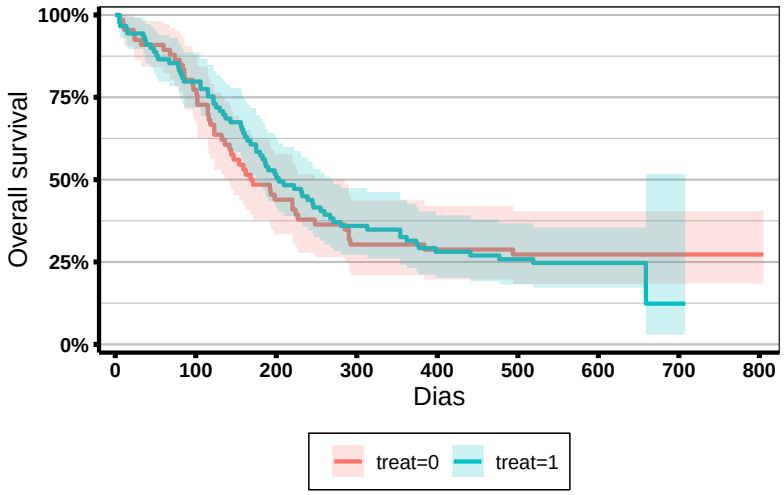
```
# Escrevendo o modelo

cox_full_model_branco <- coxph(surv_obj_branco ~ age + treat + Tempo_dialise + tt(Tempo_dial
                         data=db_branco,
                           tt=function(x,t,...) x*t)

summary(cox_full_model_branco)
```

```
Call:
coxph(formula = surv_obj_branco ~ age + treat + Tempo_dialise +
    tt(Tempo_dialise), data = db_branco, tt = function(x, t,
    ...) x * t)

  n= 464, number of events= 385
   (3 observations deleted due to missingness)

                          coef    exp(coef)     se(coef)      z
age                -0.000811624  0.999188705  0.008137628  -0.10
treat1              0.028995294  1.029419750  0.107276897   0.27
Tempo_dialise      -0.024166842  0.976122838  0.001707544 -14.15
tt(Tempo_dialise)   0.000070240  1.000070242  0.000005841  12.03
                             Pr(>|z|)
age                             0.921
treat1                          0.787
Tempo_dialise      <0.0000000000000002 ***
tt(Tempo_dialise)  <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                   exp(coef) exp(-coef) lower .95 upper .95
age                   0.9992     1.0008    0.9834    1.0153
treat1                1.0294     0.9714    0.8342    1.2703
Tempo_dialise         0.9761     1.0245    0.9729    0.9794
tt(Tempo_dialise)     1.0001     0.9999    1.0001    1.0001

Concordance= 0.78  (se = 0.01 )
Likelihood ratio test= 286.5  on 4 df,   p=<0.0000000000000002
Wald test            = 201.8  on 4 df,   p=<0.0000000000000002
```

```
Score (logrank) test = 185.6  on 4 df,   p=<0.0000000000000002
```

### 7.8.5 KM por raça = negro/pardo

```r
# Criando um novo objeto Surv

surv_obj_pardo_negro<- Surv(time = db_pardo_negro$time, event = db_pardo_negro$morte)

fit_pn = survfit(surv_obj_pardo_negro ~ treat, data = db_pardo_negro)

ggsurvfit(fit_pn, linewidth = 1) +
  labs(x = 'Dias', y = 'Overall survival') +
  add_confidence_interval() +
 # add_risktable() +
  scale_ggsurvfit() +
  biostatsquid_theme #+  coord_cartesian(xlim = c(0, 8))
```



### 7.8.6 Modelo completo para pardo/negro

```
# Escrevendo o modelo

cox_full_model_pardo_negro <- coxph(surv_obj_pardo_negro ~ age + treat + Tempo_dialise + tt(
                        data=db_pardo_negro,
                          tt=function(x,t,...) x*t)

summary(cox_full_model_pardo_negro)
```

```
Call:
coxph(formula = surv_obj_pardo_negro ~ age + treat + Tempo_dialise +
    tt(Tempo_dialise), data = db_pardo_negro, tt = function(x,
    t, ...) x * t)

  n= 153, number of events= 115
   (2 observations deleted due to missingness)

                         coef   exp(coef)    se(coef)      z      Pr(>|z|)
age                -0.02336224  0.97690855  0.01673857 -1.396         0.163
treat1              0.07806397  1.08119182  0.19431411  0.402         0.688
Tempo_dialise      -0.02367178  0.97660620  0.00397339 -5.958 0.00000000256 ***
tt(Tempo_dialise)   0.00007639  1.00007639  0.00001784  4.282 0.00001853157 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                   exp(coef) exp(-coef) lower .95 upper .95
age                   0.9769     1.0236    0.9454    1.0095
treat1                1.0812     0.9249    0.7388    1.5824
Tempo_dialise         0.9766     1.0240    0.9690    0.9842
tt(Tempo_dialise)     1.0001     0.9999    1.0000    1.0001

Concordance= 0.696  (se = 0.025 )
Likelihood ratio test= 48.53  on 4 df,   p=0.0000000007
Wald test            = 39.77  on 4 df,   p=0.00000005
Score (logrank) test = 42.56  on 4 df,   p=0.00000001
```

## 7.9 Lista 6.1 resolvida no SPSS

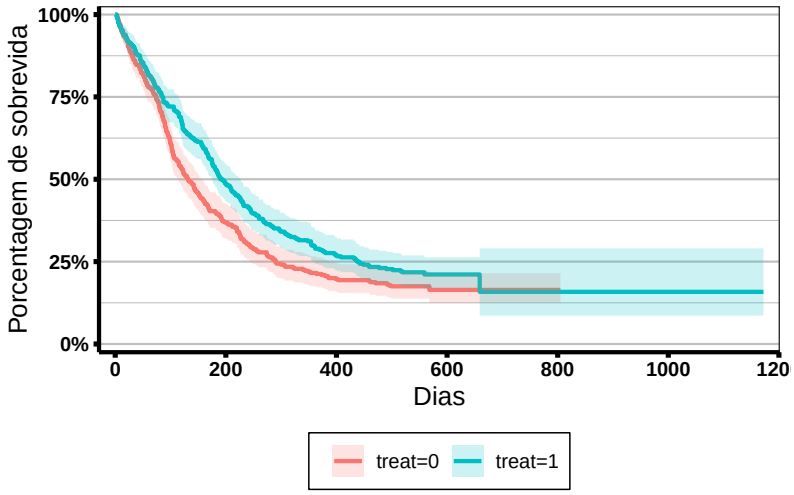
https://www.youtube.com/watch?v=6FvdrOpz0XU

## 7.10 Extras

### 7.10.1 Mais gráficos

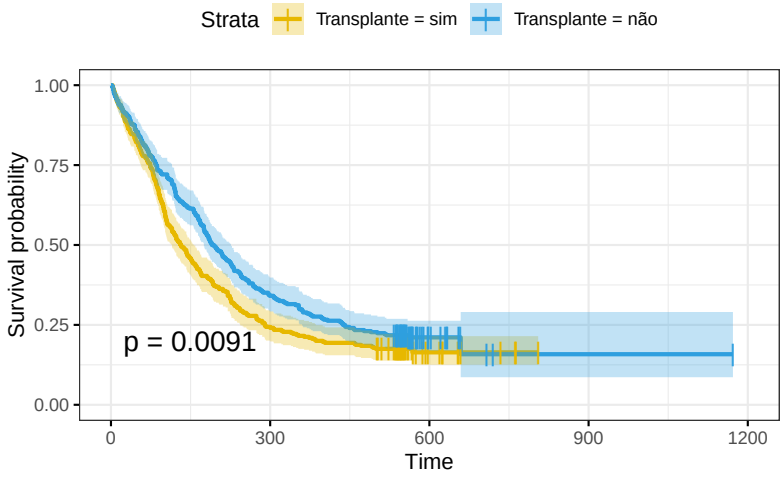
E utilizar nosso tema para personalizar e padronizar.

```
fit2_km <- ggsurvfit(fit1, linewidth = 1) +
  labs(x = 'Dias', y = 'Porcentagem de sobrevida') +
  add_confidence_interval() +
 # add_risktable() +
  scale_ggsurvfit() +
  biostatsquid_theme #+  coord_cartesian(xlim = c(0, 8))

fit2_km
```




**Cuidado** com o p-value do gráfico a seguir, ele se refere apenas ao Log-rank

```
ggsurvplot(fit1, data = db,
           size = 1,
           palette = c('#E7B800', '#2e9fdf'),
           censor.shape = '|', censor.size = 4,
           conf.int = TRUE,
           pval = TRUE, # CUIDADO, apenas log-rank
          # risk.table = TRUE,
         #  risk.table.col = 'strata',
           legend.labs = list('0' = 'Transplante = sim', '1' = 'Transplante = não'),
           risk.table.height = 0.25,
           ggtheme = theme_bw())
```




### 7.10.2 Cox tempo dependente log

```
cox_full_model_2 <- coxph(surv_obj ~ age + race + treat + Tempo_dialise + tt(Tempo_dialise),
                          data=db,
                            tt=function(x,t,...) x*log(t))
summary(cox_full_model_2)
```

```
Call:
coxph(formula = surv_obj ~ age + race + treat + Tempo_dialise +
    tt(Tempo_dialise), data = db, tt = function(x, t, ...) x *
    log(t))
```

```
  n= 617, number of events= 500
   (11 observations deleted due to missingness)

                       coef exp(coef)  se(coef)       z             Pr(>|z|)
age                -0.005656  0.994360  0.007318  -0.773              0.43960
racenegro/pardo    -0.296846  0.743159  0.108704  -2.731              0.00632
treat1             -0.007682  0.992348  0.094183  -0.082              0.93499
Tempo_dialise      -0.096272  0.908217  0.005996 -16.056 < 0.0000000000000002
tt(Tempo_dialise)   0.016885  1.017028  0.001098  15.383 < 0.0000000000000002

age
racenegro/pardo    **
treat1
Tempo_dialise      ***
tt(Tempo_dialise) ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                   exp(coef) exp(-coef) lower .95 upper .95
age                   0.9944     1.0057    0.9802    1.0087
racenegro/pardo       0.7432     1.3456    0.6006    0.9196
treat1                0.9923     1.0077    0.8251    1.1935
Tempo_dialise         0.9082     1.1011    0.8976    0.9190
tt(Tempo_dialise)     1.0170     0.9833    1.0148    1.0192

Concordance= 0.764  (se = 0.009 )
Likelihood ratio test= 459.7  on 5 df,   p=<0.0000000000000002
Wald test            = 290.9  on 5 df,   p=<0.0000000000000002
Score (logrank) test = 237.8  on 5 df,   p=<0.0000000000000002
```

```
AIC(cox_full_model_2)
```

```
[1] 5356.724
```

## 7.11 Referencias

https://stats.oarc.ucla.edu/wp-content/uploads/2022/05/survival_r.html#(48)

https://www.youtube.com/watch?v=Y_83HXuHMdc

https://youtu.be/Y_83HXuHMdc?t=9151

## 7.12 Códigos não utilizados

```
# Ajustando o banco de dados

db3 = db


# db3$time = pmax(0.5, db3$time - 0) caso eu tenha zeros no tempo de morte
# db3$time660 = as.integer(db3$time660)
# head(db3)

# db3$time660 = as.integer(db3$time660)

db3 <- tmerge(
  data1 = db3,
  data2 = db3,
  id = ID,
 # death = event(T1, delta1), caso tenha dois eventos de morte independentes. Duas doenças d
  death = event(time, morte),
  T_Cov = tdc(Tempo_dialise) # indicando a covariavel tempo-dependente
)

head(db3)
```

```
   ID age   race treat Tempo_dialise time morte tstart tstop death T_Cov
1 112  35 branco     1            51 1172     0      0    51     0     0
2 112  35 branco     1            51 1172     0     51  1172     0     1
3  91  33 branco     0            67  762     0      0    67     0     0
4  91  33 branco     0            67  762     0     67   762     0     1
5 113  35 branco     0            88  734     0      0    88     0     0
6 113  35 branco     0            88  734     0     88   734     0     1
```

```
# Duvida para Altay - colocar o evento como morte2 ou death
cox_model_T_Cov <- coxph(Surv(time = tstart, time2 = tstop, event = morte) ~ treat + T_Cov,
```

```r
summary(cox_model_T_Cov)
```

```
Call:
coxph(formula = Surv(time = tstart, time2 = tstop, event = morte) ~
    treat + T_Cov, data = db3)

  n= 1174, number of events= 934

           coef exp(coef) se(coef)      z Pr(>|z|)
treat1 -0.17205   0.84194  0.06668 -2.580  0.00987 **
T_Cov   0.03829   1.03903  0.08493  0.451  0.65210
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

       exp(coef) exp(-coef) lower .95 upper .95
treat1    0.8419     1.1877    0.7388    0.9595
T_Cov     1.0390     0.9624    0.8797    1.2272

Concordance= 0.539  (se = 0.01 )
Likelihood ratio test= 7.53  on 2 df,   p=0.02
Wald test            = 7.53  on 2 df,   p=0.02
Score (logrank) test = 7.54  on 2 df,   p=0.02
```

```r
db3 %>%
  coxph(Surv(time = tstart, time2 = tstop, event = death) ~ treat + age + race + T_Cov, data
  gtsummary::tbl_regression(exp = TRUE)
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

| **Characteristic** | **HR** | **95% CI** | **p-value** |
|---|---|---|---|
| treat | | | |
| 0 | — | — | |
| 1 | 1.05 | 0.88, 1.25 | 0.6 |
| age | 1.00 | 0.98, 1.01 | 0.5 |
| race | | | |
| branco | — | — | |
| negro/pardo | 0.68 | 0.55, 0.84 | <0.001 |
| T_Cov | 13.6 | 10.1, 18.4 | <0.001 |

### 7.12.1 Tempo em diálise como covariante tempo-dependente

```
# Ajustando o banco de dados

db2 = db


#db2$time = pmax(0.5, db2$time - 0)


db2 <- tmerge(
  data1 = db,
  data2 = db,
  id = ID,
 # death = event(T1, delta1), caso tenha dois eventos de morte independentes. Duas doenças d
  death = event(time, morte),
  T_Tempo_dialise = tdc(Tempo_dialise) # indicando a covariavel tempo-dependente
)
head(db2)
```

```
   ID age   race treat Tempo_dialise time morte tstart tstop death
1 112  35 branco     1            51 1172     0      0    51     0
2 112  35 branco     1            51 1172     0     51  1172     0
3  91  33 branco     0            67  762     0      0    67     0
4  91  33 branco     0            67  762     0     67   762     0
5 113  35 branco     0            88  734     0      0    88     0
6 113  35 branco     0            88  734     0     88   734     0
  T_Tempo_dialise
```

```
1               0
2               1
3               0
4               1
5               0
6               1
```

```
cox_model_time_dependent <- coxph(Surv(time = tstart, time2 = tstop, event = death) ~ T_Temp

summary(cox_model_time_dependent)
```

```
Call:
coxph(formula = Surv(time = tstart, time2 = tstop, event = death) ~
    T_Tempo_dialise + treat, data = db2)

  n= 1174, number of events= 508

                    coef exp(coef)  se(coef)      z            Pr(>|z|)
T_Tempo_dialise  2.585761 13.273384  0.150925 17.133 <0.0000000000000002 ***
treat1          -0.003201  0.996804  0.089471 -0.036               0.971
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                exp(coef) exp(-coef) lower .95 upper .95
T_Tempo_dialise   13.2734    0.07534    9.8745    17.842
treat1             0.9968    1.00321    0.8365     1.188

Concordance= 0.699  (se = 0.014 )
Likelihood ratio test= 382.7  on 2 df,   p=<0.0000000000000002
Wald test            = 294.6  on 2 df,   p=<0.0000000000000002
Score (logrank) test = 354.7  on 2 df,   p=<0.0000000000000002
```

```
db2 %>%
  coxph(Surv(time = tstart, time2 = tstop, event = death) ~ T_Tempo_dialise * treat, data =
  gtsummary::tbl_regression(exp = TRUE)
```

```
Table printed with `knitr::kable()`, not {gt}. Learn why at
https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
To suppress this message, include `message = FALSE` in code chunk header.
```

| **Characteristic** | **HR** | **95% CI** | **p-value** |
|---|---|---|---|
| T_Tempo_dialise | 9.78 | 6.78, 14.1 | <0.001 |
| treat | | | |
| 0 | — | — | |
| 1 | 0.59 | 0.38, 0.92 | 0.020 |
| T_Tempo_dialise * treat | | | |
| T_Tempo_dialise * 1 | 1.86 | 1.15, 3.02 | 0.012 |

## 7.13 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages effectsize
(version 0.8.6; Ben-Shachar MS et al., 2020), lpSolve (version 5.6.19;
Berkelaar M, others, 2023), survMisc (version 0.5.6; Dardis C, 2022), tm
(version 0.7.11; Feinerer I, Hornik K, 2023), flexplot (version 0.20.5; Fife D,
2024), lubridate (version 1.9.3; Grolemund G, Wickham H, 2011), coxphf (version
1.13.4; Heinze G et al., 2023), NLP (version 0.2.1; Hornik K, 2020), coin
(version 1.4.3; Hothorn T et al., 2006), ggpubr (version 0.6.0; Kassambara A,
2023), survminer (version 0.4.9; Kassambara A et al., 2021),
PHInfiniteEstimates (version 2.9.5; Kolassa JE, Zhang J, 2023), parameters
(version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke
D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see
(version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et
al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased
(version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et
al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), condSURV
(version 2.0.4; Meira-Machado L, Sestelo M, 2023), tibble (version 3.2.1;
Müller K, Wickham H, 2023), datawizard (version 0.9.0; Patil I et al., 2022),
foreign (version 0.8.85; R Core Team, 2023), nph (version 2.1; Ristl R et al.,
2021), broom (version 1.0.5; Robinson D et al., 2023), ggsurvfit (version
1.0.0; Sjoberg D et al., 2023), gtsummary (version 1.7.2; Sjoberg D et al.,
2021), rempsyc (version 0.1.6; Thériault R, 2023), survival (version 3.5.7;
Therneau T, 2023), ggplot2 (version 3.4.4; Wickham H, 2016), forcats (version
1.0.0; Wickham H, 2023), stringr (version 1.5.1; Wickham H, 2023), tidyverse
(version 2.0.0; Wickham H et al., 2019), dplyr (version 1.1.3; Wickham H et
al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), readr (version
```

2.1.4; Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et al., 2023) and kableExtra (version 1.3.4; Zhu H, 2021).

References
---


- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Berkelaar M, others (2023). _lpSolve: Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs_. R package version 5.6.19, <https://CRAN.R-project.org/package=lpSolve>.
- Dardis C (2022). _survMisc: Miscellaneous Functions for Survival Data_. R package version 0.5.6, <https://CRAN.R-project.org/package=survMisc>.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate." _Journal of Statistical Software_, *40*(3), 1-25. <https://www.jstatsoft.org/v40/i03/>.
- Heinze G, Ploner M, Jiricka L, Steiner G (2023). _coxphf: Cox Regression with Firth's Penalized Likelihood_. R package version 1.13.4, <https://CRAN.R-project.org/package=coxphf>.
- Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.
- Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2006). "A Lego system for conditional inference." _The American Statistician_, *60*(3), 257-263. doi:10.1198/000313006X118430 <https://doi.org/10.1198/000313006X118430>. Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2008). "Implementing a class of permutation tests: The coin package." _Journal of Statistical Software_, *28*(8), 1-23. doi:10.18637/jss.v028.i08 <https://doi.org/10.18637/jss.v028.i08>.
- Kassambara A (2023). _ggpubr: 'ggplot2' Based Publication Ready Plots_. R package version 0.6.0, <https://CRAN.R-project.org/package=ggpubr>.
- Kassambara A, Kosinski M, Biecek P (2021). _survminer: Drawing Survival Curves using 'ggplot2'_. R package version 0.4.9, <https://CRAN.R-project.org/package=survminer>.

- Kolassa JE, Zhang J (2023). _PHInfiniteEstimates: Tools for Inference in the Presence of a Monotone Likelihood_. R package version 2.9.5, <https://CRAN.R-project.org/package=PHInfiniteEstimates>.
- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.
- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.
- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_. <https://github.com/easystats/modelbased>.
- Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.
- Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.
- Meira-Machado L, Sestelo M (2023). _condSURV: Estimation of the Conditional

Survival Function for Ordered Multivariate Failure Time Data_. R package version 2.0.4, <https://CRAN.R-project.org/package=condSURV>.
- Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version 3.2.1, <https://CRAN.R-project.org/package=tibble>.
- Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
- R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
- R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Ristl R, Ballarini N, Götte H, Schüler A, Posch M, König F (2021). "Delayed treatment effects, treatment switching and heterogeneous patient populations: How to design and analyze RCTs in oncology." _Pharmaceutical statistics_, *20*(1), 129-145.
- Robinson D, Hayes A, Couch S (2023). _broom: Convert Statistical Objects into Tidy Tibbles_. R package version 1.0.5, <https://CRAN.R-project.org/package=broom>.
- Sjoberg D, Baillie M, Fruechtenicht C, Haesendonckx S, Treis T (2023). _ggsurvfit: Flexible Time-to-Event Figures_. R package version 1.0.0, <https://CRAN.R-project.org/package=ggsurvfit>.
- Sjoberg D, Whiting K, Curry M, Lavery J, Larmarange J (2021). "Reproducible Summary Tables with the gtsummary Package." _The R Journal_, *13*, 570-580. doi:10.32614/RJ-2021-053 <https://doi.org/10.32614/RJ-2021-053>, <https://doi.org/10.32614/RJ-2021-053>.
- Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
- Therneau T (2023). _A Package for Survival Analysis in R_. R package version 3.5-7, <https://CRAN.R-project.org/package=survival>. Terry M. Therneau, Patricia M. Grambsch (2000). _Modeling Survival Data: Extending the Cox Model_. Springer, New York. ISBN 0-387-98784-3.
- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
- Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.

- Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
- Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
- Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
- Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.
- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
- Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4, <https://CRAN.R-project.org/package=kableExtra>.

# Part III

# ARIMA

A Análise de Séries Temporais é uma ferramenta fundamental em diversas áreas, inclusive a análise de sujeito único, permitindo a compreensão e previsão de padrões temporais em conjuntos de dados. Nesse contexto, o modelo ARIMA (AutoRegressive Integrated Moving Average) surge como uma abordagem robusta para modelar séries temporais.

## Fundamentos do ARIMA:

**AutoRegressivo (AR)**: Refere-se à relação entre uma observação atual e suas observações passadas. O termo "AutoRegressivo" destaca a dependência linear de uma observação em relação a suas antecessoras.

**Integrated (I)**: Indica o número de diferenciações necessárias para tornar a série temporal estacionária, ou seja, para remover tendências e padrões sistemáticos. A estacionarização é crucial para garantir a estabilidade do modelo.

**Média Móvel (MA)**: Considera os erros residuais das observações anteriores para prever a próxima. O componente "Média Móvel" reflete a média dos erros anteriores, incorporando informações sobre o comportamento recente da série.

**Número de observações:** O número ideal de observações repetidas para uma única unidade de análise é de pelo menos 40, sendo preferível alcançar 50 observações. Não é necessário ter um grande número de pessoas ou unidades de análise; até mesmo com N = 1, você pode obter várias observações do mesmo indivíduo, tornando o ARIMA uma ferramenta eficaz de análise.

## Condições e Pressupostos:

**Estacionariedade**: O ARIMA assume que a série temporal seja estacionária, o que significa que a média, a variância e a estrutura de autocorrelação não devem variar significativamente ao longo do tempo. Se a série não for estacionária, é necessário aplicar diferenciação até atingir a estacionariedade.

**Identificação de Ordem**: A escolha adequada dos parâmetros p, d, e q (ordens AR, I, e MA) é crucial. Isso geralmente é feito por meio de análise visual, funções de autocorrelação (ACF) e autocorrelação parcial (PACF), bem como métodos estatísticos como o critério de informação de Akaike (AIC).

**Ruído Branco**: Os resíduos do modelo ARIMA devem se comportar como um "ruído branco", ou seja, serem independentes, terem média zero e variância constante. Isso garante que não haja padrões significativos nos erros residuais não capturados pelo modelo.

Além dos componentes fundamentais, o ARIMA pode ser estendido para lidar com sazonalidade através do SARIMA (Seasonal ARIMA), que incorpora parâmetros adicionais para modelar padrões recorrentes em determinados intervalos de tempo.

A adequada compreensão dos fundamentos, condições e pressupostos é essencial para explorar todo o potencial desse método e fazer previsões precisas em uma variedade de contextos.

## Passo a Passo da ARIMA

1. **Coleta e Exploração de Dados:**
    - Inicie coletando dados temporais relevantes para sua análise.
    - Explore graficamente a série temporal para identificar padrões, sazonalidades e tendências.
2. **Estacionarização da Série:**
    - Diferencie a série temporal para torná-la estacionária.
    - Utilize gráficos, como sequence charts, para visualizar mudanças ao longo do tempo.
3. **Identificação dos Parâmetros (p, d, q):**

- Analise as funções de autocorrelação (ACF) e autocorrelação parcial (PACF) para determinar os valores ideais de p (ordem AR) e q (ordem MA).
- Estabeleça a ordem de diferenciação d necessária para atingir a estacionariedade.

4. **Divisão dos Dados:**
   - Separe os dados em conjuntos de treinamento e teste para avaliar o desempenho do modelo posteriormente.

5. **Ajuste do Modelo ARIMA:**
   - Utilize os parâmetros (p, d, q) identificados para ajustar o modelo ARIMA aos dados de treinamento.
   - Ajuste também os parâmetros sazonais, se aplicável (SARIMA).

6. **Validação do Modelo:**
   - Avalie a qualidade do modelo usando critérios de informação como AIC (Akaike Information Criterion) e BIC (Bayesian Information Criterion) para modelos com os **mesmos valores** de p, d, e q.
   - Calcule o erro médio quadrático (RMSE) para comparar modelos com **diferentes configurações** de p, d, e q.

7. **Previsões e Avaliação:**
   - Faça previsões utilizando o modelo ARIMA ajustado nos dados de teste.
   - Avalie a precisão das previsões comparando-as com os valores reais.

8. **Ajustes Finais e Refinamentos:**
   - Se necessário, ajuste os parâmetros do modelo com base na análise da qualidade das previsões.

- Considere iterar nos passos anteriores para melhorar a performance do modelo.

9. **Interpretação e Comunicação dos Resultados:**

    - Comunique os resultados do modelo de forma clara, destacando as tendências identificadas e a capacidade de previsão.

Na lista prática de exercícios vamos analisar dois bancos de dados, um apenas para verificar se o modelo é estacionário ou não e outro para de fato criar modelos ARIMA.

Para mais informações sobre os parâmetros p, d, q, consulte as referências

## Referências

https://people.duke.edu/~rnau/411arim.htm

# 8 Lista 7 - Séries Temporais (ARIMA)

## 8.1 Pacotes

```
library(tidyverse)
library(flexplot)
library(foreign)
library(dplyr)
library(tm)
library(ggplot2)
library(forcats)
library(rempsyc)
library(easystats)
library(kableExtra)

#Específicos para series temporais
library(prophet)
library(forecast)
library(tseries)
```

## 8.2 Limpando o ambiente

# 9 Cigarro

## 9.1 Carregando os dados e modificando o tipo de variável

```
original = read.spss("CigarrosROD_1.sav", to.data.frame=TRUE)
glimpse(original)
```

```
Rows: 20
Columns: 2
$ Dia         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
$ cigarrosROD <dbl> 6, 10, 4, 13, 4, 11, 4, 6, 4, 15, 5, 14, 5, 21, 10, 31, 13~
```

```
db = original
```

## 9.2 Verificando se os dados são estacionários

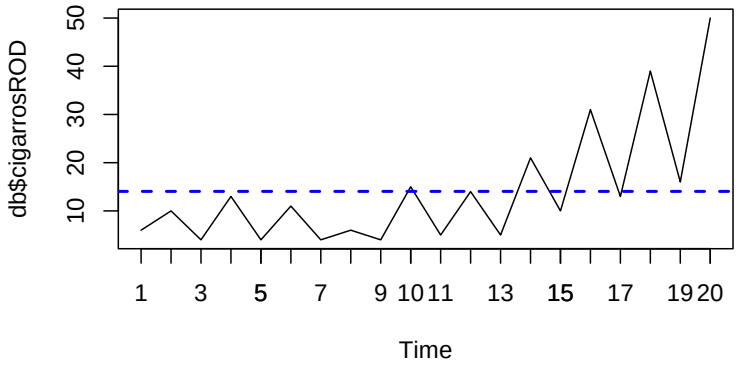
Iniciaremos nossa primeira análise para verificar a estacionaridade dos dados por meio de uma abordagem gráfica. Este gráfico simples exibirá o número de cigarros consumidos ao longo do tempo, apresentando uma linha média que atravessa toda a linha temporal. A ideia é observar se os números de cigarros oscilam próximos à média, proporcionando uma visualização intuitiva da estacionariedade dos dados.

```
# Plot estilizado
# media_cigarros <- mean(db$cigarrosROD)
#
# # Cria o gráfico com ggplot
# ggplot(data = data.frame(cigarrosROD = db$cigarrosROD), aes(x = seq_along(cigarrosROD), y
```

```
#   geom_line(color = "black", size = 1) +
#   geom_point(color = "black", size = 3) +
#   geom_hline(yintercept = media_cigarros, linetype = "dashed", color = "blue", size = 1) +
#   labs(x = "Dias", y = "Cigarros por dia") +
#   scale_x_continuous(breaks = seq_along(db$cigarrosROD), labels = seq_along(db$cigarrosROD
#   theme_minimal() +
#   theme(panel.grid = element_blank(),
#         axis.ticks = element_line())  # Adiciona ticks nos eixos x e y
```

### 9.2.1 Plot simples

```
media_cigarros <- mean(db$cigarrosROD)

# plot mais simples
plot.ts(db$cigarrosROD)
abline(h = media_cigarros, col = "blue", lty = 2, lwd = 2)
axis(1, at = db$Dia, labels = db$Dia)
```




Claramente os dados desviam bastante da média, logo essa não é uma série estacionária.

### 9.2.2 Adf teste

Podemos também utilizar o Augmented Dickey-Fuller (ADF) Test para avaliar a estacionaridade em séries temporais. A função para realizar o teste é a `adf.test()`.

Interpretação do Resultado:

- Se a estatística do teste for menor que o valor crítico ($p < 0.05$), rejeitamos a hipótese nula e concluímos que a série **é estacionária**.
- Se a estatística do teste for maior que o valor crítico ($p > 0.05$), falhamos em rejeitar a hipótese nula, sugerindo que a série é **não estacionária**.

```
# Adf teste
adf.test(db$cigarrosROD)
```

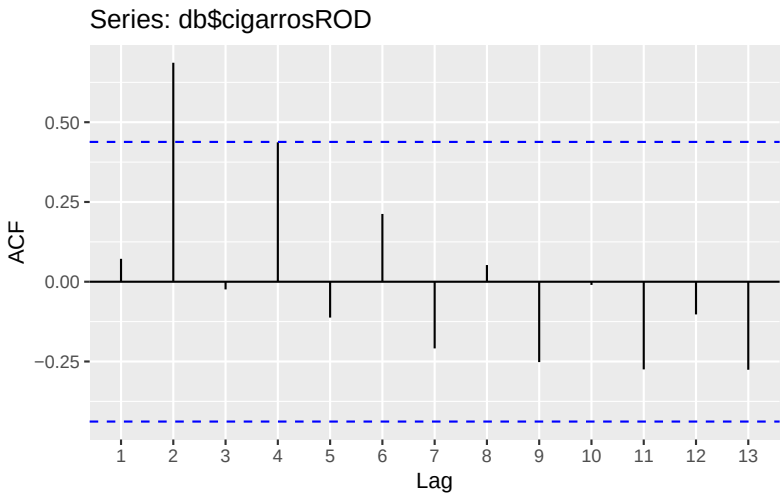
```
    Augmented Dickey-Fuller Test

data:  db$cigarrosROD
Dickey-Fuller = -0.38979, Lag order = 2, p-value = 0.9797
alternative hypothesis: stationary
```

Corroborando a análise visual, falhamos em rejeitar a hipótese nula, logo podemos assumir que a série temporal em questão não é estacionária. Em seguida vamos ver como podemos ajustar os dados.

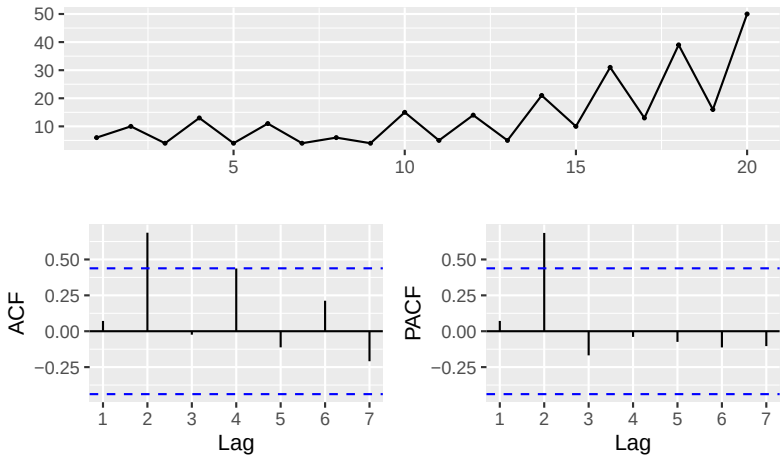
### 9.2.3 Autocorrelação

A função `acf()` (AutoCorrelation Function) no R é utilizada para calcular e visualizar os coeficientes de autocorrelação em uma série temporal. A autocorrelação mede a correlação entre uma observação e suas observações anteriores em diferentes defasagens (lags de tempo).

```r
# Calcula as autocorrelações e cria o gráfico

autocorrelacoes = acf(db$cigarrosROD, plot = FALSE)

autoplot(autocorrelacoes)
```




```r
ggtsdisplay(db$cigarrosROD)
```



Os valores de lag que tiveram um AFC além do intervalo de confiança (linha tracejada), são candidatos para utilizarmos em nosso modelo ARIMA. Portanto lag 2 e 4 são candidatos. Além disso podemos basear nossa decisão também o teste de Ljung-Box.

### 9.2.4 Teste Ljung-Box

O Teste Ljung-Box avalia para cada lag se a séria é estacionária ou não. Podemos testar individualmente para cada lag.

```
# Teste Ljung-Box com lag 2
Box.test(autocorrelacoes$acf , lag = 3, type = "Ljung-Box")
```

```
    Box-Ljung test

data:  autocorrelacoes$acf
X-squared = 8.5179, df = 3, p-value = 0.03644
```

```
lag(db$cigarrosROD,1)
```

```
 [1] NA  6 10  4 13  4 11  4  6  4 15  5 14  5 21 10 31 13 39 16
```

Uma outra forma é criar um dataframe com todos os valores de lags calculados na autocorrelação.

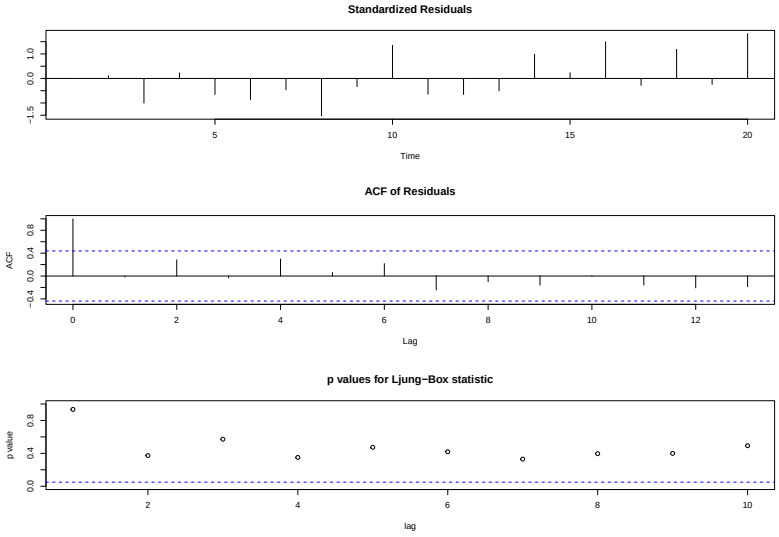
```
# Obtém o número máximo de lags disponíveis
max_lags <- length(autocorrelacoes$acf) - 1

# Inicialize os vetores para armazenar os resultados
lags <- numeric(max_lags)
p_values <- numeric(max_lags)

# Itere sobre os lags
for (lag in 1:max_lags) {
  # Execute o teste de Ljung-Box para o lag atual
```

```
  resultado_teste <- Box.test(autocorrelacoes$acf, lag = lag, type = "Ljung-Box")

  # Armazene os resultados
  lags[lag] <- lag
  p_values[lag] <- resultado_teste$p.value
}

# Crie um dataframe com os resultados
resultados_df <- data.frame(Lag = lags, P_Value = p_values)
kable(resultados_df)
```

| Lag | P_Value |
|---|---|
| 1 | 0.9409683 |
| 2 | 0.0166341 |
| 3 | 0.0364377 |
| 4 | 0.0200419 |
| 5 | 0.0236724 |
| 6 | 0.0359384 |
| 7 | 0.0197526 |
| 8 | 0.0337181 |
| 9 | 0.0101406 |
| 10 | 0.0150763 |
| 11 | 0.0030557 |
| 12 | 0.0038952 |
| 13 | 0.0006256 |

Ou ainda

```
tsdiag(auto.arima(db$cigarrosROD))
```



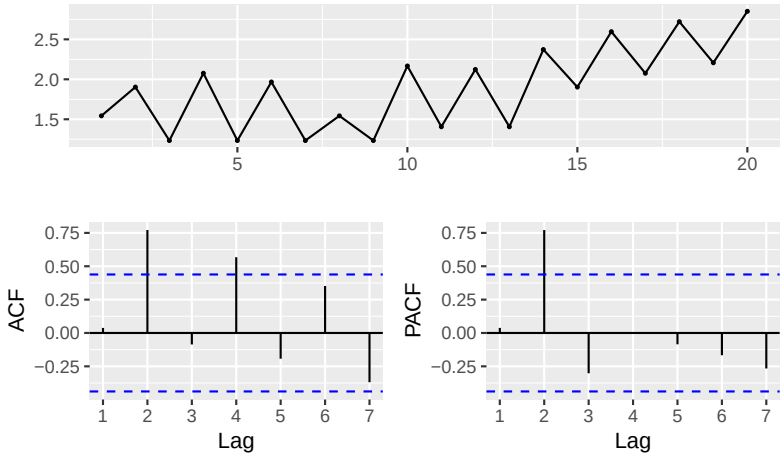

## 9.3 Transformação variabilidade e estacionária

Primeiro vamos modificar a série para que tenha variabilidade constante

```
lambda = BoxCox.lambda(db$cigarrosROD)
lambda
```

```
[1] -0.1713367
```

```
var_const = BoxCox(db$cigarrosROD, lambda = lambda)

ggtsdisplay(var_const)
```

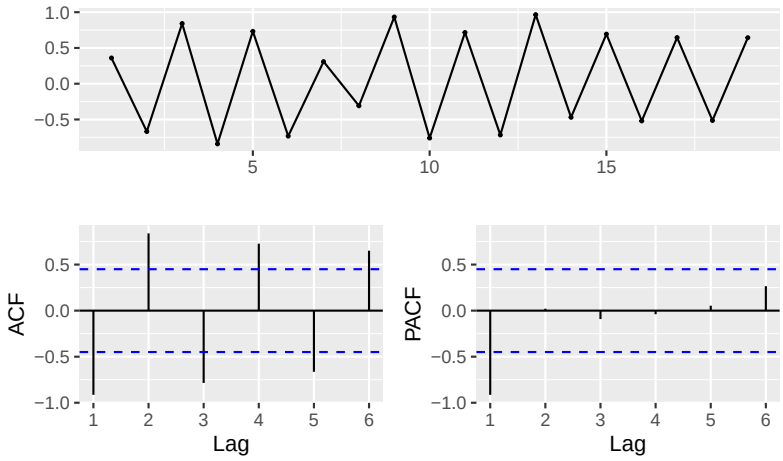



E agora podemos ajustar a serie para que ela fique estacionária.

```
ndiffs(var_const)
```

```
[1] 1
```

```
estacio = diff(var_const, 1)
ggtsdisplay(estacio)
```



De acordo com os resultados, qualquer lag, a não ser o lag 1, poderá ser utilizado para transformar os dados.

Para decidir devemos levar em conta tanto a análise gráfica da autocorrelação quanto o teste de Ljung-Box.

Logo os lags 2 e 4 são bons candidatos. Por parcimônia e sem nenhum critério teórico, vamos optar pelo lag menor, ou seja, lag 2.

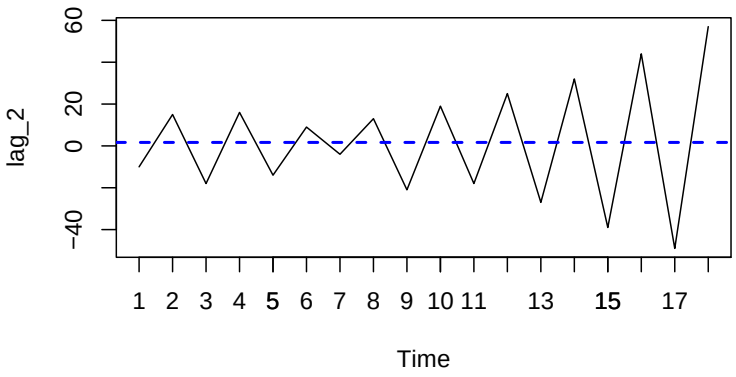
## 9.4 Transformando os dados para estacionários

Para modificar nossa série temporal utilizando lag 2 vamos utilizar a função `diff()`.

```
# Log
lag_2 = diff(db$cigarrosROD, differences = 2) # posso colocar o log da diferença também caso
```

### 9.4.1 Plot

```
media_lag_2 = mean(lag_2)
plot.ts(lag_2)
abline(h = media_lag_2, col = "blue", lty = 2, lwd = 2)
axis(1, at = db$Dia, labels = db$Dia)
```




Pronto! Agora os valores estão ocilando em torno da média. Apenas para confirmar que agora temos uma série temporal estacionária, podemos rodar novamente o adf.test.

```
adf.test(lag_2) # testar com outros valores de K(lag) para verificar o p-value
```

```
    Augmented Dickey-Fuller Test

data:  lag_2
Dickey-Fuller = -4.3237, Lag order = 2, p-value = 0.01196
alternative hypothesis: stationary
```

Esse banco de dados era apenas para transformar os dados não estacionários para estacionários. Vamos agora carregar outro banco de dados e criar o modelo ARIMA.

## 9.5 Dados séries temporais

```
original_ts = read.spss("dados series temporais.sav", to.data.frame=TRUE)
```

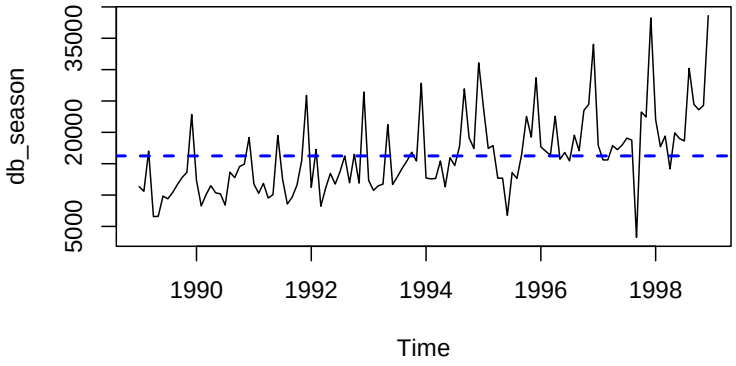
```
re-encoding from CP1252
```

```r
db_ts = original_ts

glimpse(db_ts)
```

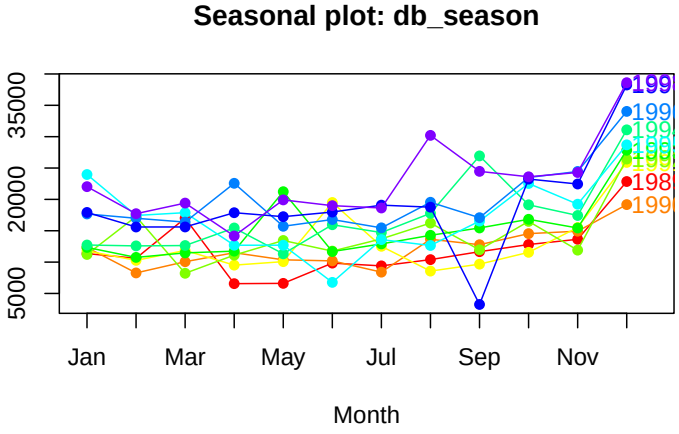
```
Rows: 120
Columns: 11
$ date         <dbl> 12818995200, 12821673600, 12824092800, 12826771200, 12829~
$ men          <dbl> 11357.92, 10605.95, 16998.57, 6563.75, 6607.69, 9839.00, ~
$ women        <dbl> 16578.93, 18236.13, 43393.55, 30908.49, 28701.58, 29647.5~
$ horas        <dbl> 7978, 8290, 8029, 7752, 8685, 7847, 7881, 8121, 7811, 870~
$ divida       <dbl> 73, 88, 65, 85, 74, 87, 79, 72, 83, 111, 74, 105, 66, 59,~
$ idade        <dbl> 34, 29, 24, 20, 17, 30, 28, 27, 35, 25, 30, 45, 35, 20, 2~
$ propaganda   <dbl> 22294.48, 27426.47, 27978.66, 28949.65, 22642.27, 27210.6~
$ escolaridade <dbl> 20, 20, 26, 22, 21, 23, 22, 20, 15, 20, 16, 29, 22, 28, 2~
$ YEAR_        <dbl> 1989, 1989, 1989, 1989, 1989, 1989, 1989, 1989, 1989, 198~
$ MONTH_       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, ~
$ DATE_        <chr> "JAN 1989", "FEB 1989", "MAR 1989", "APR 1989", "MAY 1989~
```

### 9.5.1 Plot simples

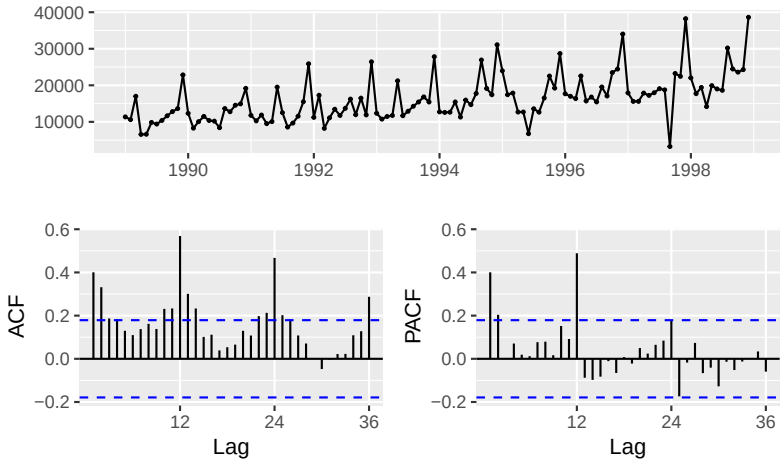
```r
media_sal_men <- mean(db_ts$men)
db_season = ts(db_ts$men,
               frequency = 12,
               start = c(1989, 1))

ts.plot(db_season)
abline(h = media_sal_men, col = "blue", lty = 2, lwd = 2)
```




```
seasonplot(db_season,
           col = rainbow(12),
           year.labels = TRUE,
           type = "o",
           pch = 16)
```




```
ggtsdisplay(db_season)
```




Semelhante ao ARIMA (0,0,0)

### 9.5.2 Adf teste

```
# Adf teste
adf.test(db_ts$men, k =1) #já está no formato estacionário
```

```
Warning in adf.test(db_ts$men, k = 1): p-value smaller than printed p-value
```

```

    Augmented Dickey-Fuller Test

data:  db_ts$men
Dickey-Fuller = -6.1931, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

### 9.5.3 Ljung-Box

Descrever

```
# Teste Ljung-Box com lag 2
Box.test(db_ts$men , lag = 1, type = "Ljung-Box")
```

```
    Box-Ljung test

data:  db_ts$men
X-squared = 19.742, df = 1, p-value = 0.000008865
```

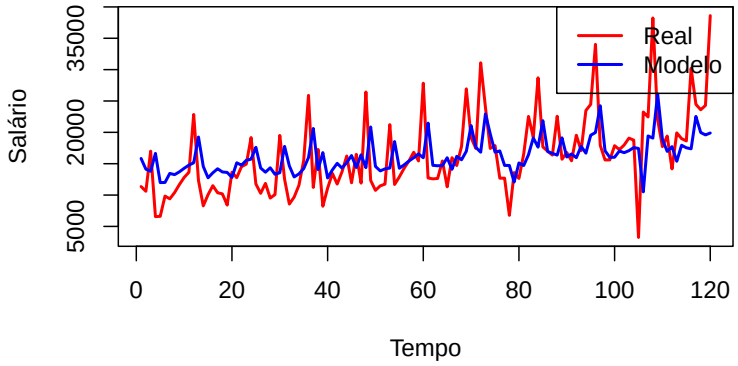
## 9.6 Modelo ARIMA (1,0,0)

```r
modelo_sal_men = Arima(db_ts$men, order = c(1,0,0))
```

### Plot 1 do modelo (1,0,0)

```r
# Supondo que você tenha as séries temporais 'modelo_sal_men$fitted' e 'db_ts$men'

# Cria o gráfico
plot(modelo_sal_men$x, type = "l", col = "red", lty = 1, lwd = 2, xlab = "Tempo", ylab = "Sa
lines(modelo_sal_men$fitted, col = "blue", lty = 1, lwd = 2)

# Adiciona uma legenda
legend("topright", legend = c("Real", "Modelo"), col = c("red", "blue"), lty = c(1, 1), lwd

# Adiciona um título ao gráfico
title(main = "Valores e Reais e do Modelo ARIMA de Salário para Homens")
```

**Valores e Reais e do Modelo ARIMA de Salário para Home**

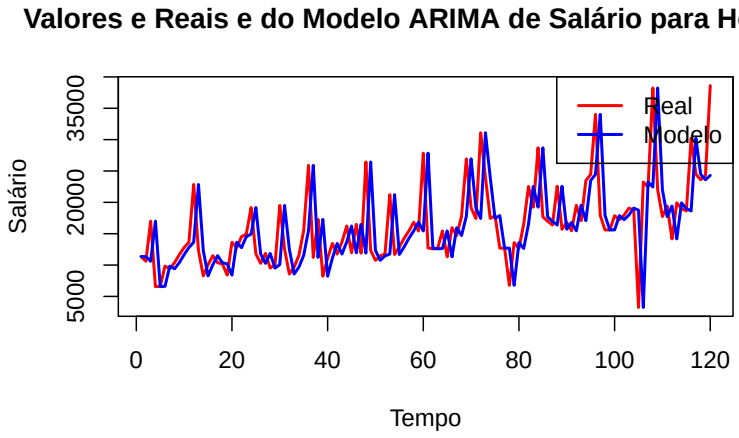



## 9.7 Modelo ARIMA (0,1,0)

```
modelo2_sal_men = Arima(db_ts$men, order = c(0,1,0))
```

**Plot 1 do modelo (0,1,0)**

```
# Supondo que você tenha as séries temporais 'modelo2_sal_men$fitted' e 'db_ts$men'

# Cria o gráfico
plot(modelo2_sal_men$x, type = "l", col = "red", lty = 1, lwd = 2, xlab = "Tempo", ylab = "S
lines(modelo2_sal_men$fitted, col = "blue", lty = 1, lwd = 2)

# Adiciona uma legenda
legend("topright", legend = c("Real", "Modelo"), col = c("red", "blue"), lty = c(1, 1), lwd

# Adiciona um título ao gráfico
title(main = "Valores e Reais e do Modelo ARIMA de Salário para Homens")
```




## 9.8 Modelo autoARIMA

Assim como o SPSS o R também tem uma função que determina automaticamente os parâmetros p, d, q. Vamos verificar qual modelo a função `auto.arima()`sugere.

```
# Para verificar qual o modelo sugerido pela função auto.arima
auto.arima(db_ts$men, trace = TRUE)
```

```
ARIMA(2,1,2) with drift         : Inf
ARIMA(0,1,0) with drift         : 2434.823
ARIMA(1,1,0) with drift         : 2412.096
ARIMA(0,1,1) with drift         : Inf
ARIMA(0,1,0)                    : 2432.897
ARIMA(2,1,0) with drift         : 2411.86
ARIMA(3,1,0) with drift         : 2408.495
ARIMA(4,1,0) with drift         : 2407.461
ARIMA(5,1,0) with drift         : 2408.674
ARIMA(4,1,1) with drift         : Inf
ARIMA(3,1,1) with drift         : Inf
ARIMA(5,1,1) with drift         : Inf
ARIMA(4,1,0)                    : 2405.816
ARIMA(3,1,0)                    : 2406.731
```

```
 ARIMA(5,1,0)                    : 2407.075
 ARIMA(4,1,1)                    : 2394.525
 ARIMA(3,1,1)                    : 2392.515
 ARIMA(2,1,1)                    : 2392.416
 ARIMA(1,1,1)                    : 2391.073
 ARIMA(0,1,1)                    : 2393.07
 ARIMA(1,1,0)                    : 2410.255
 ARIMA(1,1,2)                    : 2392.894
 ARIMA(0,1,2)                    : 2391.92
 ARIMA(2,1,0)                    : 2410.02
 ARIMA(2,1,2)                    : Inf
 ARIMA(1,1,1) with drift         : Inf

 Best model: ARIMA(1,1,1)

Series: db_ts$men
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.2036  -0.9139
s.e.  0.1002   0.0347

sigma^2 = 29737029:  log likelihood = -1192.43
AIC=2390.86   AICc=2391.07   BIC=2399.2
```

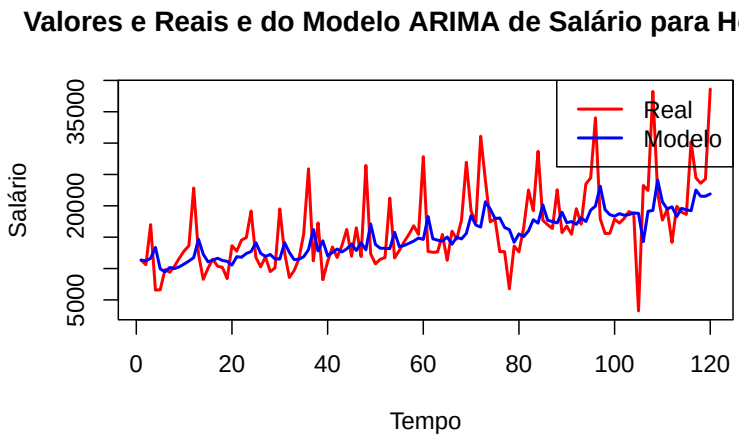
A função sugeriu o modelo 1, 1, 1. Vamos verificar os resultados.

```r
modelo_auto_sal_men = Arima(db_ts$men, order = c(1,1,1))
```

### Plot 1 do modelo (1,1,1)

```r
# Supondo que você tenha as séries temporais 'modelo_atuo_sal_men$fitted' e 'db_ts$men'

# Cria o gráfico
plot(modelo_auto_sal_men$x, type = "l", col = "red", lty = 1, lwd = 2, xlab = "Tempo", ylab
lines(modelo_auto_sal_men$fitted, col = "blue", lty = 1, lwd = 2)
```

```r
# Adiciona uma legenda
legend("topright", legend = c("Real", "Modelo"), col = c("red", "blue"), lty = c(1, 1), lwd

# Adiciona um título ao gráfico
title(main = "Valores e Reais e do Modelo ARIMA de Salário para Homens")
```




## 9.9 Homens - Modelo com variáveis independentes

> ⚠ Atenção!
>
> Ainda falta modificar os índices p, d, q das variáveis independentes como foi feito no SPSS.

### 9.9.1 Auto arima

```r
# Defina as variáveis independentes originais
nomes_variaveis <- c("horas", "divida", "idade", "propaganda", "escolaridade")

# Crie a matriz de covariáveis
covars <- as.matrix(db_ts[, nomes_variaveis, drop = FALSE])
```

```r
# Atribua os nomes diretamente à matriz de covariáveis
colnames(covars) <- nomes_variaveis


#
# covars <- cbind(
#   db_ts$horas,
#   db_ts$divida,
#   db_ts$idade,
#   db_ts$propaganda,
#   db_ts$escolaridade
# )

auto.arima(db_ts$men, xreg = covars)
```

```
Series: db_ts$men
Regression with ARIMA(1,0,0) errors

Coefficients:
         ar1   intercept   horas   divida     idade  propaganda  escolaridade
      0.1968  -23753.966  2.0271  34.5286  342.9908      0.2046       -30.3841
s.e.  0.1000    2752.767  0.2204  20.1900   43.9319      0.0733        41.3101

sigma^2 = 8316739:  log likelihood = -1122.71
AIC=2261.43   AICc=2262.72   BIC=2283.73
```

Modelo sugerido é o c(1,0,0)

```r
# Ajuste o modelo ARIMA com covariáveis
modelo_completo = Arima(
  db_ts$men,
  order = c(1, 0, 0),
  xreg = covars,
)
```

```r
# Defina as variáveis independentes originais
nomes_variaveis <- c("horas", "divida", "idade", "propaganda", "escolaridade")

# Inicialize uma lista para armazenar os modelos ajustados para cada VI
modelos_vi <- list()

# Loop através das variáveis independentes
for (variavel in nomes_variaveis) {

  # Selecione a VI específica
  variavel_ts <- db_ts[, variavel, drop = FALSE]

  # Ajuste as ordens p, d, q para a VI específica
  ordens_vi <- c(1, 0, 0)  # p, d, q

  # Ajuste o modelo ARIMA para a VI específica
  modelo_vi <- Arima(
    variavel_ts,
    order = ordens_vi,
    include.mean = TRUE,
    transform.pars = TRUE,
    fixed = NULL,
    include.drift = FALSE,
    method = "ML",  # Mude conforme necessário
    optim.control = list(trace = FALSE, REPORT = 1),
    kappa = 1
  )

  # Adicione o modelo ao vetor de modelos
  modelos_vi[[variavel]] <- modelo_vi
}

# Agora, você tem modelos ajustados para cada VI na lista modelos_vi

# Combine os modelos ARIMA para as VI em uma única matriz
covars <- cbind(
  modelos_vi$horas$fitted,
  modelos_vi$divida$fitted,
  modelos_vi$idade$fitted,
```

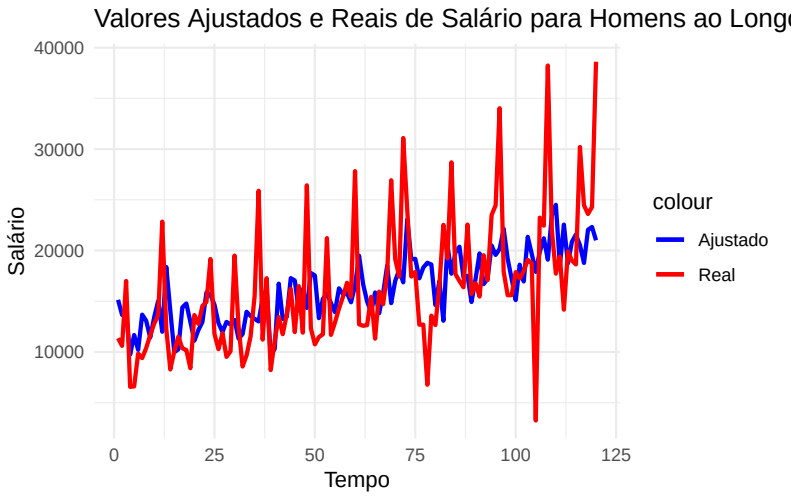
```
  modelos_vi$propaganda$fitted,
  modelos_vi$escolaridade$fitted
)

# Ajuste o modelo ARIMA principal com as covariáveis
modelo_completo <- Arima(
  db_ts$men,
  order = c(1, 0, 0),
  xreg = covars,
  seasonal = list(order = c(0, 0, 0)),  # Adapte conforme necessário
  include.mean = TRUE,
  transform.pars = TRUE,
  fixed = NULL,
  include.drift = FALSE,
  method = "ML",  # Mude conforme necessário
  optim.control = list(trace = FALSE, REPORT = 1),
  kappa = 1
)
```

### 9.9.2 Plot do modelo com VIs

```
# Criar um dataframe com as séries temporais
df_full <- data.frame(
  Tempo = seq_along(modelo_completo$fitted),
  Ajustado = modelo_completo$fitted,
  Real = modelo_completo$x
)

# Criar o gráfico com ggplot2
ggplot(df_full, aes(x = Tempo)) +
  geom_line(aes(y = Ajustado, color = "Ajustado"), size = 1) +
  geom_line(aes(y = Real, color = "Real"), size = 1) +
  labs(x = "Tempo", y = "Salário", title = "Valores Ajustados e Reais de Salário para Homens
  scale_color_manual(values = c("Ajustado" = "blue", "Real" = "red"), guide = "legend") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
Don't know how to automatically pick scale for object of type <ts>. Defaulting
to continuous.
```




### 9.9.3 AIC, BIC e RMSE

```
performance(modelo_completo)
```

```
# Indices of model performance

AIC      |     AICc |      BIC | R2.modelos_vi$horas$fitted | R2.modelos_vi$divida$fitted | R2
-----------------------------------------------------------------------------------------------
2421.052 | 6925.052 | 2443.352 |                      0.397 |                       0.213 |
```

### 9.9.4 Resultados

```
# Visualize o resumo do modelo
summary(modelo_completo)
```

```
Series: db_ts$men
Regression with ARIMA(1,0,0) errors
```

```
Coefficients:
          ar1  intercept  modelos_vi$horas$fitted  modelos_vi$divida$fitted
      -0.1727   34621.57                   0.7594                 -607.0271
s.e.   0.1639   40027.85                   1.3741                  316.5860
      modelos_vi$idade$fitted  modelos_vi$propaganda$fitted
                     371.3498                        0.1770
s.e.                 127.6716                        1.0677
      modelos_vi$escolaridade$fitted
                            135.8938
s.e.                         81.0383

sigma^2 = 31455262:  log likelihood = -1202.53
AIC=2421.05   AICc=2422.35   BIC=2443.35

Training set error measures:
                    ME    RMSE      MAE       MPE     MAPE     MASE        ACF1
Training set -5.05266 5442.46 3863.789 -11.92625 27.46778 0.826093 0.002906111
```

```
# db_ts$horas,       xreg1
# db_ts$divida  ,    xreg2
# db_ts$idade,       xreg3
# db_ts$propaganda,  xreg4
# db_ts$escolaridade xreg5
```

**Coeficientes e valores de p**

```
library(lmtest) # pacote para calcular os estimates e valores de p
```

```
Warning: package 'lmtest' was built under R version 4.3.2

Carregando pacotes exigidos: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
# Use a função coeftest para obter coeficientes e p-valores
test_coef <- coeftest(modelo_completo)

# Acesse os coeficientes estimados e os p-valores
coeficientes <- round(test_coef[, "Estimate"], 3)
p_valores <- round(test_coef[, "Pr(>|z|)"], 3)

# Crie uma nova coluna com asteriscos para valores de p significativos
test_coef$Significativo <- ifelse(p_valores < 0.05, "*", "")
```

```
Warning in test_coef$Significativo <- ifelse(p_valores < 0.05, "*", ""):
Realizando coerção de LHD para uma lista
```

```r
# Exiba os resultados
resultados <- data.frame(Coeficientes = coeficientes, p_valores = paste0(format(p_valores, d
print(resultados)
```

```
                               Coeficientes p_valores
ar1                                  -0.173     0.292
intercept                         34621.566     0.387
modelos_vi$horas$fitted               0.759     0.581
modelos_vi$divida$fitted           -607.027     0.055
modelos_vi$idade$fitted             371.350    0.004*
modelos_vi$propaganda$fitted          0.177     0.868
modelos_vi$escolaridade$fitted      135.894     0.094
```

### 9.9.5 Mulheres - Modelo com variáveis independentes para

**Verificar qual o melhor modelo utilizando as VIs no auto.arima**

```r
# Supondo que você tenha um dataframe 'db_ts' com as variáveis mencionadas

# Defina as variáveis independentes originais
nomes_variaveis <- c("horas", "divida", "idade", "propaganda", "escolaridade")
```

```r
# Crie a matriz de covariáveis
covars <- as.matrix(db_ts[, nomes_variaveis, drop = FALSE])

# Atribua os nomes diretamente à matriz de covariáveis
colnames(covars) <- nomes_variaveis


#
# covars <- cbind(
#   db_ts$horas,
#   db_ts$divida,
#   db_ts$idade,
#   db_ts$propaganda,
#   db_ts$escolaridade
# )

auto.arima(db_ts$women, xreg = covars)
```

```
Series: db_ts$women
Regression with ARIMA(0,0,1) errors

Coefficients:
         ma1   intercept   horas   divida     idade  propaganda  escolaridade
      0.3351  -37941.151  2.6828  90.5433    1.6294      0.9805      445.6786
s.e.  0.1096    6458.967  0.5145  52.4388  110.2176      0.1693      102.0171

sigma^2 = 49641429:  log likelihood = -1229.95
AIC=2475.89   AICc=2477.19   BIC=2498.19
```

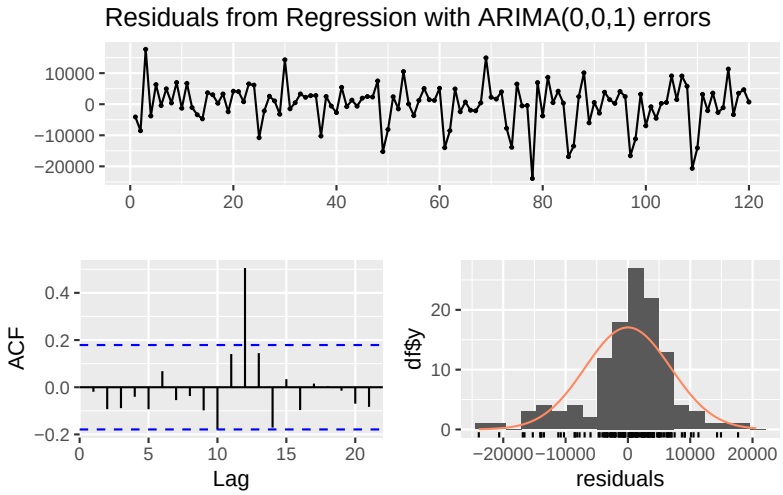
Modelo sugerido é o c(0,0,1)

```r
# Ajuste o modelo ARIMA com covariáveis
modelo_completo_women = Arima(
  db_ts$women,
  order = c(0, 0, 1),
  xreg = covars
)
```

```r
# Visualize o resumo do modelo
summary(modelo_completo_women)
```

```
Series: db_ts$women
Regression with ARIMA(0,0,1) errors

Coefficients:
         ma1   intercept   horas   divida     idade  propaganda  escolaridade
      0.3351  -37941.151  2.6828  90.5433    1.6294      0.9805       445.6786
s.e.  0.1096    6458.967  0.5145  52.4388  110.2176      0.1693       102.0171

sigma^2 = 49641429:  log likelihood = -1229.95
AIC=2475.89   AICc=2477.19   BIC=2498.19

Training set error measures:
                     ME     RMSE  MAE       MPE    MAPE      MASE        ACF1
Training set -0.1962005 6837.081 4889 -3.414187 14.6811 0.5114592 -0.01923702
```

```r
# db_ts$horas,       xreg1
# db_ts$divida  ,    xreg2
# db_ts$idade,       xreg3
# db_ts$propaganda,  xreg4
# db_ts$escolaridade xreg5

checkresiduals(modelo_completo_women)
```

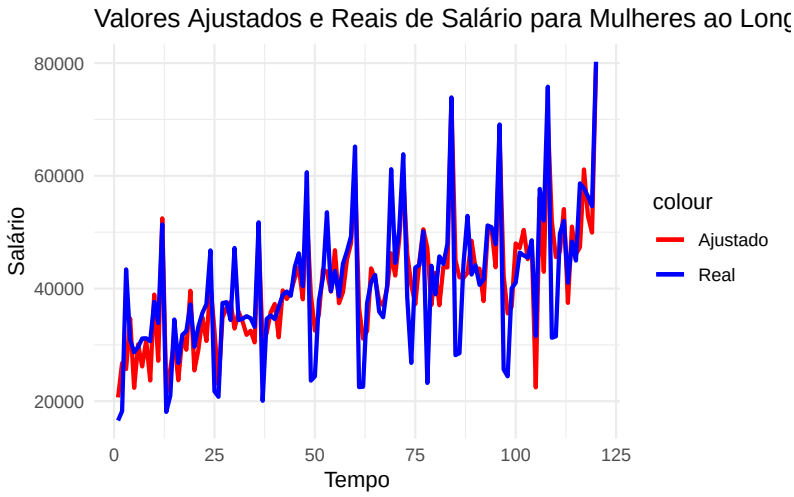



```
    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,0,1) errors
Q* = 10.048, df = 9, p-value = 0.3466

Model df: 1.   Total lags used: 10
```

### Plot do modelo com VIs

```r
# Criar um dataframe com as séries temporais
df_full_women <- data.frame(
  Tempo = seq_along(modelo_completo_women$fitted),
  Ajustado = modelo_completo_women$fitted,
  Real = modelo_completo_women$x
)

# Criar o gráfico com ggplot2
ggplot(df_full_women, aes(x = Tempo)) +
  geom_line(aes(y = Ajustado, color = "Ajustado"), size = 1) +
  geom_line(aes(y = Real, color = "Real"), size = 1) +
  labs(x = "Tempo", y = "Salário", title = "Valores Ajustados e Reais de Salário para Mulher
  scale_color_manual(values = c("Ajustado" = "red", "Real" = "blue"), guide = "legend") +
  theme_minimal()
```

```
Don't know how to automatically pick scale for object of type <ts>. Defaulting
to continuous.
```




### Coeficientes e valores de p

```r
library(lmtest) # pacote para calcular os estimates e valores de p


# Use a função coeftest para obter coeficientes e p-valores
test_coef_women <- coeftest(modelo_completo_women)

# Acesse os coeficientes estimados e os p-valores
coeficientes_women <- round(test_coef_women[, "Estimate"], 3)
p_valores_women <- round(test_coef_women[, "Pr(>|z|)"], 3)

# Crie uma nova coluna com asteriscos para valores de p significativos
test_coef_women$Significativo <- ifelse(p_valores_women < 0.05, "*", "")
```

```
Warning in test_coef_women$Significativo <- ifelse(p_valores_women < 0.05, :
Realizando coerção de LHD para uma lista
```

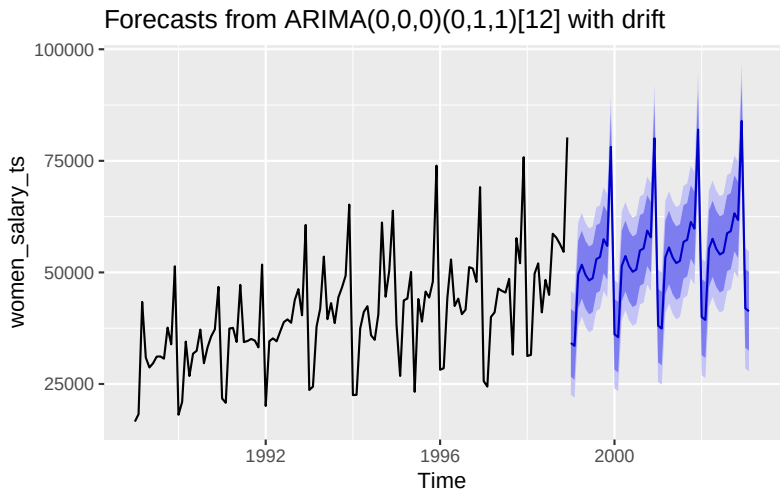
```r
# Exiba os resultados
resultados_women <- data.frame(Coeficientes = coeficientes_women, Pvalores = paste0(format(p
print(resultados_women)
```

```
             Coeficientes Pvalores
ma1                 0.335   0.002*
intercept      -37941.151   0.000*
horas               2.683   0.000*
divida             90.543    0.084
idade               1.629    0.988
propaganda          0.981   0.000*
escolaridade      445.679   0.000*
```

## 9.10 Forecast (previsões)

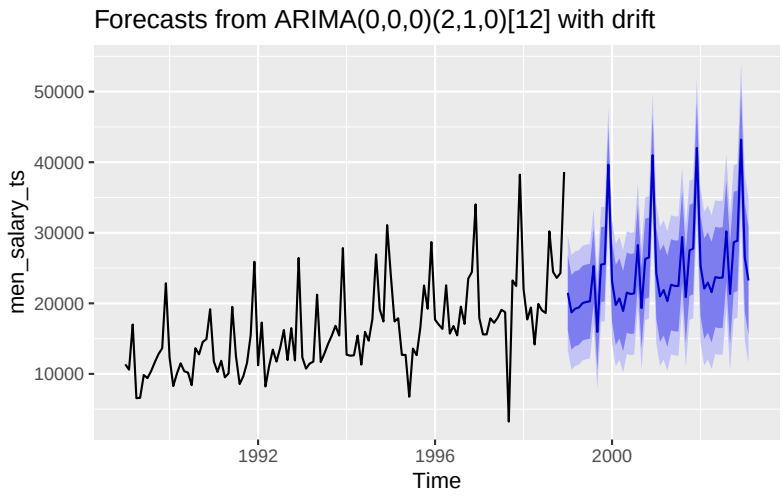
### 9.10.1 Mulheres - 50 anos

```r
women_salary_ts = ts(db_ts$women,
                frequency = 12,
                start = c(1989, 1))


fit_arima_women = auto.arima(women_salary_ts)

fcast_women = forecast(fit_arima_women, h=50)
autoplot(fcast_women)
```




### 9.10.2 Homens - 50 anos

```
men_salary_ts = ts(db_ts$men,
                   frequency = 12,
                   start = c(1989, 1))


fit_arima_men = auto.arima(men_salary_ts)

fcast_men = forecast(fit_arima_men, h=50)
autoplot(fcast_men)
```



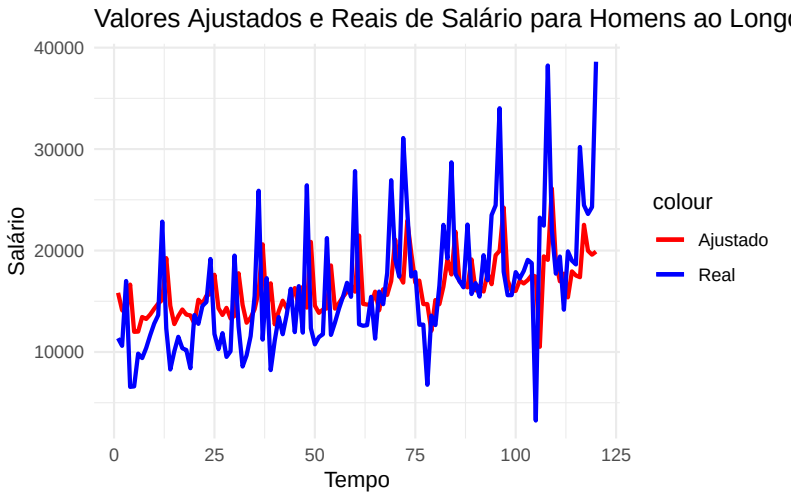

## 9.11 Extras

### 9.11.1 Mais gráficos

#### Plot 2 do modelo (1,0,0)

```
modelo_sal_men = Arima(db_ts$men, order = c(1,0,0))

# Criar um dataframe com as séries temporais
df_100 <- data.frame(
  Tempo = seq_along(modelo_sal_men$fitted),
  Ajustado = modelo_sal_men$fitted,
  Real = modelo_sal_men$x
)

# Criar o gráfico com ggplot2
ggplot(df_100, aes(x = Tempo)) +
  geom_line(aes(y = Ajustado, color = "Ajustado"), size = 1) +
  geom_line(aes(y = Real, color = "Real"), size = 1) +
  labs(x = "Tempo", y = "Salário", title = "Valores Ajustados e Reais de Salário para Homens
  scale_color_manual(values = c("Ajustado" = "red", "Real" = "blue"), guide = "legend") +
  theme_minimal()
```

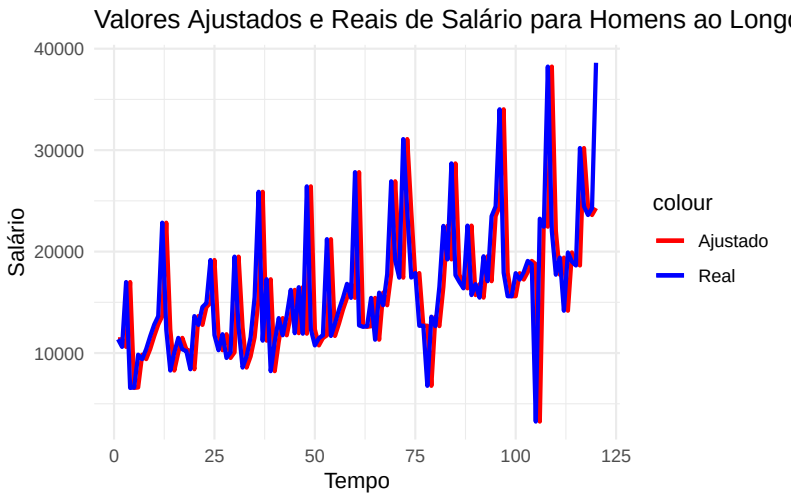
```
Don't know how to automatically pick scale for object of type <ts>. Defaulting
to continuous.
```




**Plot 2 do modelo (0,1,0)**

```r
# Criar um dataframe com as séries temporais
df_010 <- data.frame(
  Tempo = seq_along(modelo2_sal_men$fitted),
  Ajustado = modelo2_sal_men$fitted,
  Real = modelo2_sal_men$x
)

# Criar o gráfico com ggplot2
ggplot(df_010, aes(x = Tempo)) +
  geom_line(aes(y = Ajustado, color = "Ajustado"), size = 1) +
  geom_line(aes(y = Real, color = "Real"), size = 1) +
  labs(x = "Tempo", y = "Salário", title = "Valores Ajustados e Reais de Salário para Homens
  scale_color_manual(values = c("Ajustado" = "red", "Real" = "blue"), guide = "legend") +
  theme_minimal()
```

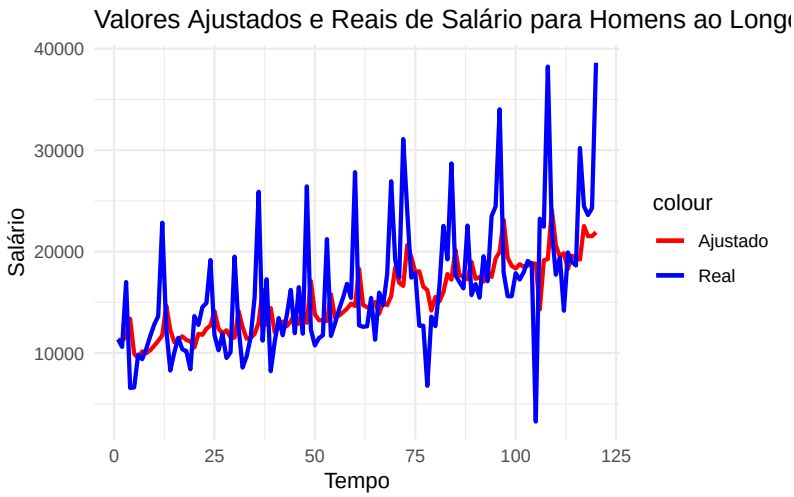
```
Don't know how to automatically pick scale for object of type <ts>. Defaulting
to continuous.
```




**Plot 2 do modelo (1,1,1)**

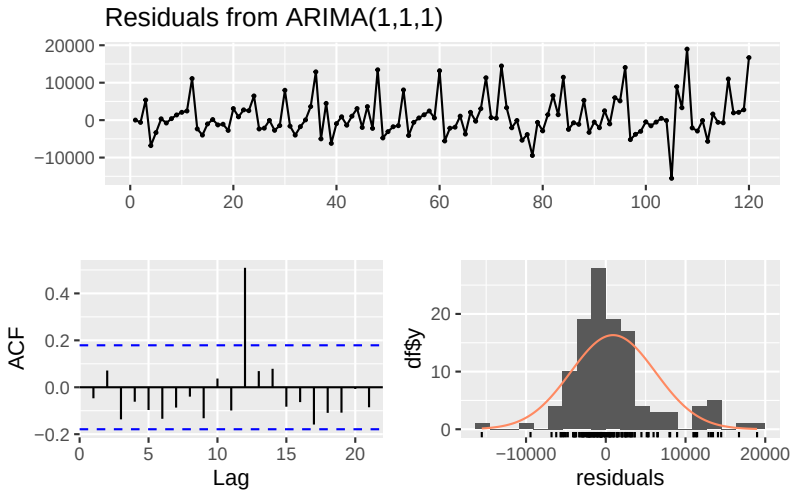
```
modelo_atuo_sal_men = Arima(db_ts$men, order = c(1,1,1))

# Criar um dataframe com as séries temporais
df_111 <- data.frame(
  Tempo = seq_along(modelo_atuo_sal_men$fitted),
  Ajustado = modelo_atuo_sal_men$fitted,
  Real = modelo_atuo_sal_men$x
)

# Criar o gráfico com ggplot2
ggplot(df_111, aes(x = Tempo)) +
  geom_line(aes(y = Ajustado, color = "Ajustado"), size = 1) +
  geom_line(aes(y = Real, color = "Real"), size = 1) +
  labs(x = "Tempo", y = "Salário", title = "Valores Ajustados e Reais de Salário para Homens
  scale_color_manual(values = c("Ajustado" = "red", "Real" = "blue"), guide = "legend") +
  theme_minimal()
```

```
Don't know how to automatically pick scale for object of type <ts>. Defaulting
to continuous.
```




## 9.12 Verificando resíduos

```
checkresiduals(modelo_auto_sal_men)
```




```
    Ljung-Box test

data:  Residuals from ARIMA(1,1,1)
```

```
Q* = 10.891, df = 8, p-value = 0.208

Model df: 2.   Total lags used: 10
```

```r
summary(modelo_auto_sal_men)
```

```
Series: db_ts$men
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.2036  -0.9139
s.e.  0.1002   0.0347

sigma^2 = 29737029:  log likelihood = -1192.43
AIC=2390.86   AICc=2391.07   BIC=2399.2

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE
Training set 915.6723 5384.571 3662.003 -5.571742 25.15003 0.7829504
                     ACF1
Training set -0.04692903
```

## 9.13 Lista 7 resolvida no SPSS

https://www.youtube.com/watch?v=qTQ1YDgyByE

## 9.14 Referências


https://facebook.github.io/prophet/docs/installation.html#r

https://rpubs.com/mpleo/timeseries_prophet

https://www.youtube.com/watch?v=ny3gRhfVsi4&t=10s

https://www.youtube.com/watch?v=Txuo9JQjnKE ótima ref em PT-BR

https://www.youtube.com/watch?v=RJzmHkGWCxs&list=PLEuzmtv9IuT_vg5oE0lQyZR-wgbVeGztt

## 9.15 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages effectsize
(version 0.8.6; Ben-Shachar MS et al., 2020), Rcpp (version 1.0.11;
Eddelbuettel D et al., 2023), tm (version 0.7.11; Feinerer I, Hornik K, 2023),
flexplot (version 0.20.5; Fife D, 2024), lubridate (version 1.9.3; Grolemund G,
Wickham H, 2011), rlang (version 1.1.1; Henry L, Wickham H, 2023), NLP (version
0.2.1; Hornik K, 2020), forecast (version 8.21.1; Hyndman R et al., 2023),
parameters (version 0.21.3; Lüdecke D et al., 2020), performance (version
0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D et al.,
2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6;
Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019),
modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7;
Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022),
tibble (version 3.2.1; Müller K, Wickham H, 2023), datawizard (version 0.9.0;
Patil I et al., 2022), foreign (version 0.8.85; R Core Team, 2023), prophet
(version 1.0; Taylor S, Letham B, 2021), rempsyc (version 0.1.6; Thériault R,
2023), tseries (version 0.10.55; Trapletti A, Hornik K, 2023), ggplot2 (version
3.4.4; Wickham H, 2016), forcats (version 1.0.0; Wickham H, 2023), stringr
(version 1.5.1; Wickham H, 2023), tidyverse (version 2.0.0; Wickham H et al.,
2019), dplyr (version 1.1.3; Wickham H et al., 2023), purrr (version 1.0.2;
Wickham H, Henry L, 2023), readr (version 2.1.4; Wickham H et al., 2023), tidyr
(version 1.3.0; Wickham H et al., 2023), zoo (version 1.8.12; Zeileis A,
Grothendieck G, 2005), lmtest (version 0.9.40; Zeileis A, Hothorn T, 2002) and
kableExtra (version 1.3.4; Zhu H, 2021).

References
----------
  - Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of
Effect Size Indices and Standardized Parameters." _Journal of Open Source
Software_, *5*(56), 2815. doi:10.21105/joss.02815
<https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
```

- Eddelbuettel D, Francois R, Allaire J, Ushey K, Kou Q, Russell N, Ucar I, Bates D, Chambers J (2023). _Rcpp: Seamless R and C++ Integration_. R package version 1.0.11, <https://CRAN.R-project.org/package=Rcpp>. Eddelbuettel D, François R (2011). "Rcpp: Seamless R and C++ Integration." _Journal of Statistical Software_, *40*(8), 1-18. doi:10.18637/jss.v040.i08 <https://doi.org/10.18637/jss.v040.i08>. Eddelbuettel D (2013). _Seamless R and C++ Integration with Rcpp_. Springer, New York. doi:10.1007/978-1-4614-6868-4 <https://doi.org/10.1007/978-1-4614-6868-4>, ISBN 978-1-4614-6867-7. Eddelbuettel D, Balamuta J (2018). "Extending R with C++: A Brief Introduction to Rcpp." _The American Statistician_, *72*(1), 28-36. doi:10.1080/00031305.2017.1375990 <https://doi.org/10.1080/00031305.2017.1375990>.
- Feinerer I, Hornik K (2023). _tm: Text Mining Package_. R package version 0.7-11, <https://CRAN.R-project.org/package=tm>. Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infrastructure in R." _Journal of Statistical Software_, *25*(5), 1-54. doi:10.18637/jss.v025.i05 <https://doi.org/10.18637/jss.v025.i05>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate." _Journal of Statistical Software_, *40*(3), 1-25. <https://www.jstatsoft.org/v40/i03/>.
- Henry L, Wickham H (2023). _rlang: Functions for Base Types and Core R and 'Tidyverse' Features_. R package version 1.1.1, <https://CRAN.R-project.org/package=rlang>.
- Hornik K (2020). _NLP: Natural Language Processing Infrastructure_. R package version 0.2-1, <https://CRAN.R-project.org/package=NLP>.
- Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2023). _forecast: Forecasting functions for time series and linear models_. R package version 8.21.1, <https://pkg.robjhyndman.com/forecast/>. Hyndman RJ, Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R." _Journal of Statistical Software_, *26*(3), 1-22. doi:10.18637/jss.v027.i03 <https://doi.org/10.18637/jss.v027.i03>.
- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open Source Software_, *5*(53), 2445. doi:10.21105/joss.02445 <https://doi.org/10.21105/joss.02445>.
- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139.

doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
  - Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
  - Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
  - Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
  - Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.
  - Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_. <https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version 3.2.1, <https://CRAN.R-project.org/package=tibble>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
  - R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria.

<https://www.R-project.org/>.
  - Taylor S, Letham B (2021). _prophet: Automatic Forecasting Procedure_. R package version 1.0, <https://CRAN.R-project.org/package=prophet>.
  - Thériault R (2023). "rempsyc: Convenience functions for psychology." _Journal of Open Source Software_, *8*(87), 5466. doi:10.21105/joss.05466 <https://doi.org/10.21105/joss.05466>, <https://doi.org/10.21105/joss.05466>.
  - Trapletti A, Hornik K (2023). _tseries: Time Series Analysis and Computational Finance_. R package version 0.10-55, <https://CRAN.R-project.org/package=tseries>.
  - Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
  - Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
  - Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
  - Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
  - Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
  - Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
  - Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.
  - Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
  - Zeileis A, Grothendieck G (2005). "zoo: S3 Infrastructure for Regular and Irregular Time Series." _Journal of Statistical Software_, *14*(6), 1-27. doi:10.18637/jss.v014.i06 <https://doi.org/10.18637/jss.v014.i06>.
  - Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." _R News_, *2*(3), 7-10. <https://CRAN.R-project.org/doc/Rnews/>.
  - Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4, <https://CRAN.R-project.org/package=kableExtra>.

# Part IV

# SEM

A modelagem de equação estrutural (SEM) é uma técnica estatística que permite examinar as relações entre múltiplas variáveis, tanto observadas quanto não observadas. Ela combina elementos da análise de regressão, da análise fatorial e da análise de caminhos para testar hipóteses sobre a estrutura causal de fenômenos complexos.

Dentro do SEM, vamos abordar apenas duas técnicas nos exercícios: a análise fatorial confirmatória (CFA) e a análise de caminhos (PA). A CFA é usada para verificar a validade de um modelo de mensuração, ou seja, se as variáveis observadas medem adequadamente os construtos latentes. A PA é usada para estimar os efeitos diretos, indiretos e totais entre as variáveis, tanto observadas quanto latentes. Ambas as técnicas podem ser combinadas em um modelo de equações estruturais completo, que inclui tanto a parte de mensuração quanto a parte estrutural.

A modelagem de equação estrutural é uma ferramenta poderosa e flexível para a pesquisa em diversas áreas do conhecimento, como psicologia, sociologia, educação, economia, administração, entre outras. Ela permite testar teorias, comparar modelos alternativos, avaliar a qualidade do ajuste, controlar variáveis de confusão, e explorar relações não lineares e interações. No entanto, ela também requer cuidados na especificação, estimação, avaliação e interpretação dos modelos, bem como na escolha dos dados e dos softwares adequados.

Nos três exercícios propostos, serão exploradas técnicas estatísticas avançadas. No primeiro, uma regressão linear será aplicada aos dados DADOSPATH.sav, investigando a relação entre idade, IMC, sociabilidade e o número de treinos. O segundo exercício envolverá uma Path Analysis no mesmo banco de dados, comparando resultados com a regressão linear. No terceiro, será realizada uma Análise Fatorial Confirmatória no banco Fatorial_escala.sav, examinando a estrutura do questionário de apego a amigos em fatores de Confiança e Alienacão, além de discutir a qualidade do modelo e possíveis limitações.

## Referências

https://repositorio.ufba.br/bitstream/ri/17684/1/ebook_SEM_2012.pdf

https://statplace.com.br/blog/modelagem-de-equacoes-estruturais/

# 10 Lista 8 - CFA e Path Analysis

```
library(foreign)
library(tidyverse)
library(lavaan)
library(semPlot)
library(performance)
library(easystats)
library(kableExtra)
```

## 10.1 a) Regressão linear

> **Exercício**
>
> Veja o banco de dados DADOSPATH.sav. Nele temos os dados de Idade, IMC, numero de treinos e sociabilidade (questionario) de um grupo de 94 pessoas. Faca um modelo de regressao linear tendo como variavel dependente o numero de Treinos e as demais variaveis como independentes.

```
original = read.spss("DADOS PATH.sav", to.data.frame=TRUE)
modelo_1 = lm(Treinos ~ Idade + IMC1 + Sociabilidade, data = original)
```

Modelo:

$$Y \sim \beta_0 + \beta_1 * idade + \beta_2 * IMC1 + \beta_3 * Sociabilidade + \epsilon$$

### 10.1.1 Resultados

```
kable(summary(modelo_1)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 78.0103771 | 33.1315541 | 2.3545644 | 0.0207191 |
| Idade | 1.9071028 | 0.5479748 | 3.4802749 | 0.0007745 |
| IMC1 | -2.8021799 | 1.1809989 | -2.3727202 | 0.0197865 |
| Sociabilidade | 0.5177685 | 0.5903261 | 0.8770889 | 0.3827736 |

Um modelo linear (estimado usando Mínimos Quadrados Ordinários - OLS) foi utilizado para prever a variável Treinos com base nas variáveis Idade, IMC1 e Sociabilidade. O modelo explica uma proporção estatisticamente significativa e moderada da variância ($R^2$ = 0,14, F(3, 90) = 5,06, p = 0,003, $R^2$ ajustado = 0,12). Dentro desse modelo: • O efeito da Idade é estatisticamente significativo e positivo (beta = 1,91, IC 95% [0,82, 3,00], t(90) = 3,48, p < 0,001; Beta padronizado = 0,35, IC 95% [0,15, 0,56]) • O efeito do IMC1 é estatisticamente significativo e negativo (beta = -2,80, IC 95% [-5,15, -0,46], t(90) = -2,37, p = 0,020; Beta padronizado = -0,24, IC 95% [-0,44, -0,04]) • O efeito da Sociabilidade é estatisticamente não significativo e positivo (beta = 0,52, IC 95% [-0,66, 1,69], t(90) = 0,88, p = 0,383; Beta padronizado = 0,09, IC 95% [-0,11, 0,28]) Parâmetros padronizados foram obtidos ajustando o modelo a uma versão padronizada do conjunto de dados. Intervalos de Confiança (ICs) de 95% e valores-p foram calculados usando uma aproximação da distribuição t de Wald.

## 10.2 b) Path Analysis

> **Exercício**
>
> Com base no mesmo banco acima faça uma Path Analysis e monte um diagrama no ~~AMOS~~ R. Compare os resultados com os dados encontrados na regressão linear.

```r
path_1 = "Treinos ~ Idade + IMC1 + Sociabilidade"


path_model_1 = sem(
  model = path_1,
  data = original,
)
```

### 10.2.1 Tabela com os resultados

Como sempre, podemos utilizar a função `summary()` para retornar um resumo com os resultados do modelo

```r
summary(path_model_1) # posso colocar o parametro fit.measures = TRUE para obter os valores
```

```
lavaan 0.6.16 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         4

  Number of observations                            94

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Treinos ~
    Idade             1.907    0.536    3.557    0.000
    IMC1             -2.802    1.156   -2.425    0.015
```

```
    Sociabilidade     0.518    0.578    0.896    0.370

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Treinos        2050.999  299.169    6.856    0.000
```

No caso da path analisys recomendamos utilizar a função `parameterEstimates()` do pacote `lavaan` para ter uma tabela mais direta com os resultados dos estimadores.

```
kable(parameterEstimates(path_model_1))
```

| lhs | op | rhs | est | se | z | pvalue | ci.lower | |
|---|---|---|---|---|---|---|---|---|
| Treinos | ~ | Idade | 1.9071028 | 0.5361890 | 3.5567736 | 0.0003754 | 0.8561917 | |
| Treinos | ~ | IMC1 | -2.8021799 | 1.1555981 | -2.4248741 | 0.0153137 | -5.0671106 | -0 |
| Treinos | ~ | Sociabilidade | 0.5177685 | 0.5776294 | 0.8963679 | 0.3700563 | -0.6143644 | |
| Treinos | ~~ | Treinos | 2050.9987436 | 299.1689143 | 6.8556546 | 0.0000000 | 1464.6384463 | 263 |
| Idade | ~~ | Idade | 82.5840878 | 0.0000000 | NA | NA | 82.5840878 | 8 |
| Idade | ~~ | IMC1 | 10.7872961 | 0.0000000 | NA | NA | 10.7872961 | 1 |
| Idade | ~~ | Sociabilidade | 3.7635808 | 0.0000000 | NA | NA | 3.7635808 | |
| IMC1 | ~~ | IMC1 | 17.7567863 | 0.0000000 | NA | NA | 17.7567863 | 1 |
| IMC1 | ~~ | Sociabilidade | 1.2498636 | 0.0000000 | NA | NA | 1.2498636 | |
| Sociabilidade | ~~ | Sociabilidade | 65.6008375 | 0.0000000 | NA | NA | 65.6008375 | 6 |

Os resultados foram os mesmos obtidos tanto pela path analysis quanto pela regressão linear simples.

### 10.2.2 Indices de qualidade do modelo

```
model_performance(path_model_1, metrics = c("Chi2", "Chi2_df", "NFI", "NNFI", "CFI", "RMSEA"
```

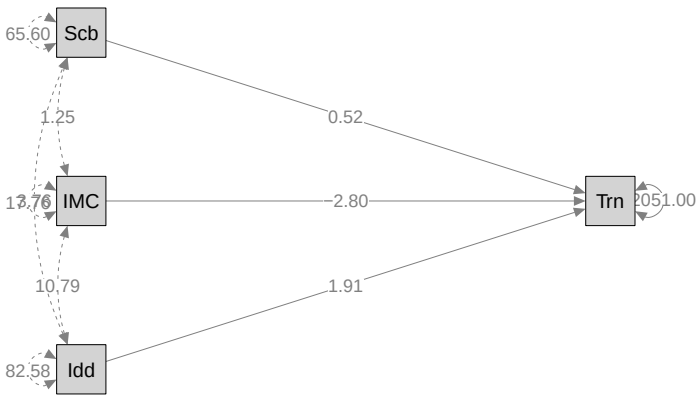
```
# Indices of model performance

Chi2(0) |   NFI |  NNFI |   CFI | RMSEA |     AIC |      BIC
-----------------------------------------------------------------
0.000   | 1.000 | 1.000 | 1.000 | 0.000 | 991.612 | 1001.785
```

```
AIC(path_model_1)
```

```
[1] 991.6122
```

### 10.2.3 Diagrama da path analysis

```
P <- semPaths(
          object = path_model_1,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```



## 10.3 c) CFA

> **Exercício**
>
> Veja o banco de dados Fatorial escala.sav. Faça uma Análise fatorial confirmatória (CFA) gerando os seguintes fatores com base no questionário de apego a amigos (IAA).

Segundo a teoria esperada, os fatores teriam o seguinte agrupamento: a. Confianca – Q13 Q14 Q15 b. Alienacao – Q1 Q2 Q3 Monte o diagrama e discuta a qualidade do modelo e suas limitações caso existam.

**Equação do Modelo 1:**

```
cfa_eq = " Alienação =~ IAa1 + IAa2 + IAa3 Confiança
=~ IAa13 + IAa14 + IAa15  "
```

**Análise Fatorial Confirmatória do modelo 1**

```
cfa_modelo = cfa(   model = cfa_eq,   data = dados_CFA,
std.lv = TRUE  )
```

```r
dados_CFA = read.spss("fatorial CFA.sav", to.data.frame=TRUE)
```

```r
cfa_eq = "
Alienação =~ IAa1 + IAa2 + IAa3
Confiança =~ IAa13 + IAa14 + IAa15

"
```

```r
cfa_modelo = cfa(
  model = cfa_eq,
  data = dados_CFA,
  std.lv = TRUE #If TRUE, the metric of each latent variable is determined by fixing their (

)
```

### 10.3.1 Resultados do modelo sem covariâncias entre os resíduos (modelo 1)

```
summary(cfa_modelo) # posso colocar no summary o parametro fit.measures = TRUE
```

```
lavaan 0.6.16 ended normally after 18 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        13

                                                  Used       Total
  Number of observations                           347         348

Model Test User Model:

  Test statistic                                39.166
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Alienação =~
    IAa1              0.578    0.069    8.408    0.000
    IAa2              0.842    0.069   12.135    0.000
    IAa3              0.423    0.051    8.278    0.000
  Confiança =~
    IAa13             0.580    0.044   13.279    0.000
    IAa14             0.660    0.052   12.652    0.000
    IAa15             0.586    0.053   11.105    0.000

Covariances:
                 Estimate  Std.Err  z-value  P(>|z|)
```

```
  Alienação ~~
    Confiança         0.865    0.051   16.807    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .IAa1              1.023    0.088   11.607    0.000
   .IAa2              0.676    0.089    7.627    0.000
   .IAa3              0.568    0.049   11.667    0.000
   .IAa13             0.316    0.036    8.898    0.000
   .IAa14             0.485    0.051    9.564    0.000
   .IAa15             0.564    0.052   10.767    0.000
    Alienação         1.000
    Confiança         1.000
```

```
kable(parameterEstimates(cfa_modelo))
```

| lhs | op | rhs | est | se | z | pvalue | ci.lower | ci.upper |
|---|---|---|---|---|---|---|---|---|
| Alienação | =~ | IAa1 | 0.5783085 | 0.0687768 | 8.408481 | 0 | 0.4435084 | 0.7131086 |
| Alienação | =~ | IAa2 | 0.8419500 | 0.0693835 | 12.134735 | 0 | 0.7059609 | 0.9779391 |
| Alienação | =~ | IAa3 | 0.4226729 | 0.0510624 | 8.277581 | 0 | 0.3225925 | 0.5227533 |
| Confiança | =~ | IAa13 | 0.5796173 | 0.0436497 | 13.278840 | 0 | 0.4940655 | 0.6651691 |
| Confiança | =~ | IAa14 | 0.6603480 | 0.0521921 | 12.652247 | 0 | 0.5580533 | 0.7626427 |
| Confiança | =~ | IAa15 | 0.5856667 | 0.0527386 | 11.105077 | 0 | 0.4823008 | 0.6890325 |
| IAa1 | ~~ | IAa1 | 1.0234312 | 0.0881741 | 11.606933 | 0 | 0.8506131 | 1.1962493 |
| IAa2 | ~~ | IAa2 | 0.6763156 | 0.0886784 | 7.626611 | 0 | 0.5025092 | 0.8501221 |
| IAa3 | ~~ | IAa3 | 0.5677370 | 0.0486609 | 11.667220 | 0 | 0.4723635 | 0.6631105 |
| IAa13 | ~~ | IAa13 | 0.3160382 | 0.0355192 | 8.897671 | 0 | 0.2464219 | 0.3856546 |
| IAa14 | ~~ | IAa14 | 0.4851673 | 0.0507280 | 9.564088 | 0 | 0.3857422 | 0.5845925 |
| IAa15 | ~~ | IAa15 | 0.5636709 | 0.0523500 | 10.767362 | 0 | 0.4610669 | 0.6662750 |
| Alienação | ~~ | Alienação | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |
| Confiança | ~~ | Confiança | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |
| Alienação | ~~ | Confiança | 0.8646695 | 0.0514471 | 16.806951 | 0 | 0.7638349 | 0.9655040 |

Os resultados da análise de equações estruturais indicam que o modelo ajustado apresenta um bom ajuste aos dados observados ( ² = 39,166, df = 8, p < 0,001). O modelo envolve duas variáveis latentes, “Alienação” e “Confiança”, e suas variáveis observadas.

Os coeficientes de carga (estimates) indicam que as perguntas associadas a “Alienação” (IAa1, IAa2, IAa3) e “Confiança”

(IAa13, IAa14, IAa15) têm influências positivas significativas em suas respectivas variáveis latentes.

Além disso, a covariância entre "Alienação" e "Confiança" é estatisticamente significativa (estimate = 0,865, p < 0,001), sugerindo uma relação entre essas duas dimensões.

Esses resultados fornecem evidências de que o modelo proposto é estatisticamente significativo.

### 10.3.2 Índices de qualidade do modelo 1

```
model_performance(cfa_modelo, metrics = c("Chi2", "Chi2_df", "NFI", "CFI", "RMSEA", "p_RMSEA
```

```
# Indices of model performance

Chi2(8) |   NFI |   CFI | RMSEA | p (RMSEA) |      AIC |      BIC |  NNFI
-------------------------------------------------------------------------
39.166  | 0.918 | 0.933 | 0.106 |     0.003 | 5402.476 | 5452.517 | 0.874
```

Os resultados dos índices de qualidade indicam que o modelo apresenta uma adequada qualidade de aderência aos dados observados, conforme evidenciado pelos índices de ajuste (NFI, CFI) acima de 0.9. Apenas o NNFI (ou TFI) está abaixo de 0.9, indicando um bom ajuste relativo.

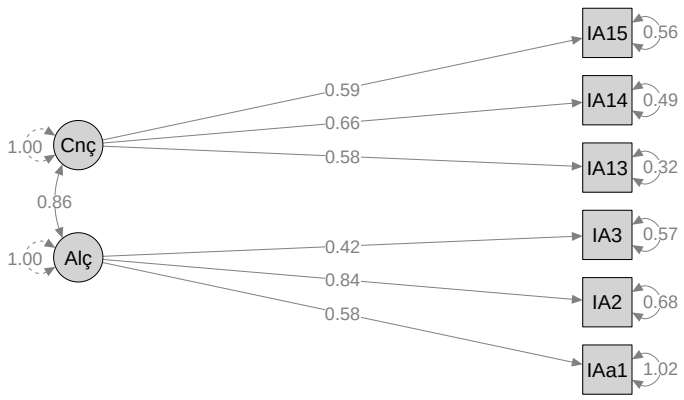
No entanto, o valor do RMSEA é alto (10%), indicando que o modelo pode ser aprimorado.

Os valores de AIC e BIC serão utilizados para efeito de comparação com os modelos a seguir.

### 10.3.3 Diagrama da CFA com o modelo 1

```
plot_CFA <- semPaths(
          object = cfa_modelo,
          what = "path",
          whatLabels = "par",
          style = "ram",
```

```
            layout = "tree",
            rotation = 2,
            sizeMan = 7,
            sizeLat = 7,
            color = "lightgray",
            edge.label.cex = 1.2,
            label.cex = 1.3
)
```




### 10.3.4 Verificar os índices de modificações do modelo 1

Os índices de modificação podem ser obtidos utilizando a função `modindices()`. Por padrão, os índices de modificação são impressos para cada parâmetro não livre (ou fixado como zero). Os índices de modificação são complementados pelos valores de mudança esperada nos parâmetros (EPC) (coluna epc). As últimas três colunas contêm os valores padronizados de EPC (sepc.lv: padronização apenas das variáveis latentes; sepc.all: padronização de todas as variáveis; sepc.nox: padronização de todas, exceto variáveis observadas exógenas).

```
kable(modificationindices(cfa_modelo, sort = TRUE, minimum.value = 5))
```

| | lhs | op | rhs | mi | epc | sepc.lv | sepc.all | sepc.nox |
|---|---|---|---|---|---|---|---|---|
| 34 | IAa13 | ~~ | IAa14 | 18.245523 | 0.1825252 | 0.1825252 | 0.4661302 | 0.4661302 |
| 18 | Alienação | =~ | IAa15 | 18.245517 | 0.9569991 | 0.9569991 | 1.0050447 | 1.0050447 |
| 29 | IAa2 | ~~ | IAa14 | 17.160433 | -0.2043919 | -0.2043919 | -0.3568151 | -0.3568151 |
| 30 | IAa2 | ~~ | IAa15 | 10.461405 | 0.1539148 | 0.1539148 | 0.2492832 | 0.2492832 |
| 20 | Confiança | =~ | IAa2 | 8.403845 | -1.6415789 | -1.6415789 | -1.3947817 | -1.3947817 |
| 23 | IAa1 | ~~ | IAa3 | 8.403843 | -0.1390871 | -0.1390871 | -0.1824669 | -0.1824669 |
| 35 | IAa13 | ~~ | IAa15 | 5.006171 | -0.0840609 | -0.0840609 | -0.1991642 | -0.1991642 |
| 17 | Alienação | =~ | IAa14 | 5.006165 | -0.5603084 | -0.5603084 | -0.5837728 | -0.5837728 |

### 10.3.5 Novo modelo com a covariância dos resíduos (modelo 2)

```
cfa_eq_2 = "
Alienação =~ IAa1 + IAa2 + IAa3
Confiança =~ IAa13 + IAa14 + IAa15

# Covariancia dos resíduos
IAa1 ~~ IAa3
IAa13 ~~ IAa14
IAa13 ~~ IAa15

"
```

```
cfa_modelo_2 = cfa(
  model = cfa_eq_2,
  data = dados_CFA,
  std.lv = TRUE
)
```

**Equação do Modelo 2:**

```
cfa_eq_2 = " Alienação =~ IAa1 + IAa2 + IAa3 Confiança
=~ IAa13 + IAa14 + IAa15  # Covariancia dos resíduos
IAa1 ~~ IAa3  IAa13 ~~ IAa14 IAa13 ~~ IAa15  "
```

**Análise Fatorial Confirmatória do modelo 2**

```
cfa_modelo_2 = cfa(   model = cfa_eq_2,   data =
dados_CFA,   std.lv = TRUE    )
```

### 10.3.6 Resultados modelo 2

```
summary(cfa_modelo_2) # posso colocar no summary o parametro fit.measures = TRUE
```

```
lavaan 0.6.16 ended normally after 24 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        16

                                                  Used       Total
  Number of observations                           347         348

Model Test User Model:

  Test statistic                                14.194
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.014

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Alienação =~
    IAa1              0.625    0.069    9.009    0.000
    IAa2              0.844    0.066   12.774    0.000
    IAa3              0.440    0.052    8.484    0.000
  Confiança =~
    IAa13             0.486    0.054    9.010    0.000
    IAa14             0.558    0.057    9.744    0.000
    IAa15             0.627    0.058   10.741    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .IAa1 ~~
   .IAa3             -0.131    0.047   -2.812    0.005
```

```
 .IAa13 ~~
   .IAa14            0.156    0.042    3.700    0.000
   .IAa15            0.005    0.037    0.123    0.902
  Alienação ~~
    Confiança        0.945    0.063   15.041    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .IAa1              0.968    0.088   11.049    0.000
   .IAa2              0.674    0.081    8.299    0.000
   .IAa3              0.553    0.049   11.303    0.000
   .IAa13             0.416    0.048    8.717    0.000
   .IAa14             0.609    0.059   10.323    0.000
   .IAa15             0.514    0.060    8.497    0.000
    Alienação         1.000
    Confiança         1.000
```

```
kable(parameterEstimates(cfa_modelo_2))
```

| lhs | op | rhs | est | se | z | pvalue | ci.lower | ci.upper |
|---|---|---|---|---|---|---|---|---|
| Alienação | =~ | IAa1 | 0.6246375 | 0.0693348 | 9.0090043 | 0.0000000 | 0.4887438 | 0.7605313 |
| Alienação | =~ | IAa2 | 0.8436173 | 0.0660435 | 12.7736683 | 0.0000000 | 0.7141745 | 0.9730601 |
| Alienação | =~ | IAa3 | 0.4398298 | 0.0518412 | 8.4841676 | 0.0000000 | 0.3382228 | 0.5414367 |
| Confiança | =~ | IAa13 | 0.4855333 | 0.0538885 | 9.0099663 | 0.0000000 | 0.3799138 | 0.5911527 |
| Confiança | =~ | IAa14 | 0.5583551 | 0.0573047 | 9.7436136 | 0.0000000 | 0.4460399 | 0.6706703 |
| Confiança | =~ | IAa15 | 0.6267943 | 0.0583571 | 10.7406769 | 0.0000000 | 0.5124166 | 0.7411721 |
| IAa1 | ~~ | IAa3 | -0.1314725 | 0.0467540 | -2.8120052 | 0.0049234 | -0.2231086 | -0.0398363 |
| IAa13 | ~~ | IAa14 | 0.1558199 | 0.0421126 | 3.7000779 | 0.0002155 | 0.0732807 | 0.2383591 |
| IAa13 | ~~ | IAa15 | 0.0045594 | 0.0370060 | 0.1232056 | 0.9019443 | -0.0679712 | 0.0770899 |
| IAa1 | ~~ | IAa1 | 0.9676998 | 0.0875801 | 11.0493149 | 0.0000000 | 0.7960460 | 1.1393536 |
| IAa2 | ~~ | IAa2 | 0.6735053 | 0.0811578 | 8.2987178 | 0.0000000 | 0.5144390 | 0.8325716 |
| IAa3 | ~~ | IAa3 | 0.5529392 | 0.0489190 | 11.3031626 | 0.0000000 | 0.4570597 | 0.6488186 |
| IAa13 | ~~ | IAa13 | 0.4162519 | 0.0477495 | 8.7174157 | 0.0000000 | 0.3226647 | 0.5098391 |
| IAa14 | ~~ | IAa14 | 0.6094664 | 0.0590382 | 10.3232593 | 0.0000000 | 0.4937537 | 0.7251790 |
| IAa15 | ~~ | IAa15 | 0.5138053 | 0.0604718 | 8.4966149 | 0.0000000 | 0.3952828 | 0.6323277 |
| Alienação | ~~ | Alienação | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |
| Confiança | ~~ | Confiança | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |
| Alienação | ~~ | Confiança | 0.9447949 | 0.0628146 | 15.0410074 | 0.0000000 | 0.8216805 | 1.0679092 |

Os resultados da análise indicam que o modelo apresenta um ra-

zoável ajuste aos dados observados, conforme evidenciado pelos índices de ajuste, embora o teste qui-quadrado seja estatisticamente significativo ( $^2 = 14.194$, df $= 5$, p $= 0.014$), indicando diferenças entre o modelo e os dados.

As cargas fatoriais para os indicadores associados às variáveis latentes "Alienação" e "Confiança" são todas estatisticamente significativas (p $< 0.001$), indicando que esses indicadores têm uma relação com suas respectivas variáveis latentes.

### 10.3.7 Índices de qualidade do modelo 2

```
model_performance(cfa_modelo_2, metrics = c("Chi2", "Chi2_df", "NFI", "CFI", "RMSEA", "p_RMS
```

```
# Indices of model performance

Chi2(5) |   NFI |   CFI | RMSEA | p (RMSEA) |      AIC |      BIC |  NNFI
-----------------------------------------------------------------------------
14.194  | 0.970 | 0.980 | 0.073 |     0.165 | 5383.504 | 5445.093 | 0.940
```

Os resultados dos índices de qualidade indicam que o modelo apresenta uma adequada qualidade de aderência aos dados observados, conforme evidenciado pelos índices de ajuste (NFI, CFI e NNFI) acima de 0.9.

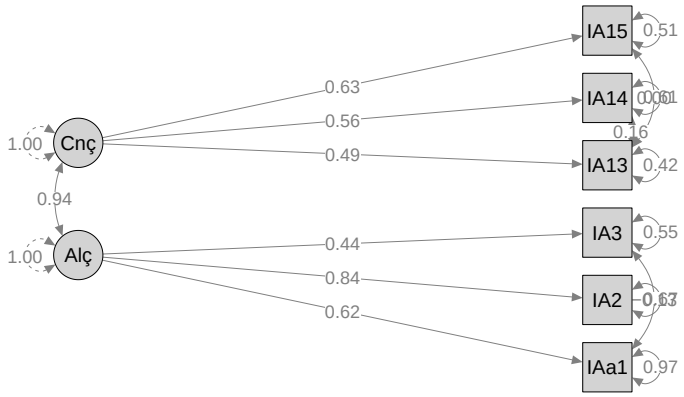
No entanto, o valor do RMSEA é moderado (7%), indicando que o modelo pode ser aprimorado.

Os valores de AIC e BIC serão utilizados para efeito de comparação com os modelos a seguir.

### 10.3.8 Diagrama do modelo 2

```
plot_CFA <- semPaths(
          object = cfa_modelo_2,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
```

```
            rotation = 2,
            sizeMan = 7,
            sizeLat = 7,
            color = "lightgray",
            edge.label.cex = 1.2,
            label.cex = 1.3
)
```




### 10.3.9 Comparação entre os modelos

```
compare_performance(cfa_modelo, cfa_modelo_2,
                    metrics = c("NFI", "CFI", "RMSEA", "p_RMSEA", "AIC", "BIC", "NNFI"),
                    rank = TRUE, verbose = F)
```

```
# Comparison of Model Performance Indices

Name          |  Model |   NFI |   CFI | RMSEA | p (RMSEA) |  NNFI | AIC weights | BIC weights
-------------------------------------------------------------------------------------------------
cfa_modelo_2 | lavaan | 0.970 | 0.980 | 0.073 |     0.165 | 0.940 |       1.000 |       0.976
cfa_modelo   | lavaan | 0.918 | 0.933 | 0.106 |     0.003 | 0.874 |    7.59e-05 |       0.024
```

O modelo_2 demonstra superioridade em relação ao modelo_1
com base nos critérios de ajuste avaliados.

## 10.4 Complementar: Modelo com apenas um fator latente (modelo 3)

```
cfa_eq_3 = "
F1 =~ IAa1 + IAa2 + IAa3 + IAa13 + IAa14 + IAa15
"
```

```
cfa_modelo_3 = cfa(
  model = cfa_eq_3,
  data = dados_CFA,
  std.lv = TRUE
)
```

**Equação do modelo 3:**

```
cfa_eq_3 = " F1 =~ IAa1 + IAa2 + IAa3 + IAa13 +
IAa14 + IAa15 "
```

**Análise Fatorial Confirmatória do modelo 2**

```
cfa_modelo_3 = cfa(   model = cfa_eq_3,   data =
dados_CFA,   std.lv = TRUE    )
```

### 10.4.1 Resultados do modelo 3

```
kable(parameterEstimates(cfa_modelo_3))
```

| lhs | op | rhs | est | se | z | pvalue | ci.lower | ci.upper |
|---|---|---|---|---|---|---|---|---|
| F1 | =~ | IAa1 | 0.5580340 | 0.0664300 | 8.400335 | 0 | 0.4278336 | 0.6882343 |
| F1 | =~ | IAa2 | 0.7651182 | 0.0637750 | 11.997154 | 0 | 0.6401216 | 0.8901149 |
| F1 | =~ | IAa3 | 0.4044799 | 0.0493864 | 8.190103 | 0 | 0.3076843 | 0.5012755 |
| F1 | =~ | IAa13 | 0.5564023 | 0.0432259 | 12.871965 | 0 | 0.4716811 | 0.6411235 |
| F1 | =~ | IAa14 | 0.6400430 | 0.0517363 | 12.371268 | 0 | 0.5386419 | 0.7414442 |
| F1 | =~ | IAa15 | 0.5959511 | 0.0519930 | 11.462144 | 0 | 0.4940467 | 0.6978555 |
| IAa1 | ~~ | IAa1 | 1.0464704 | 0.0865635 | 12.089050 | 0 | 0.8768091 | 1.2161317 |
| IAa2 | ~~ | IAa2 | 0.7997900 | 0.0764096 | 10.467135 | 0 | 0.6500299 | 0.9495502 |
| IAa3 | ~~ | IAa3 | 0.5827853 | 0.0479619 | 12.151008 | 0 | 0.4887817 | 0.6767889 |
| IAa13 | ~~ | IAa13 | 0.3424110 | 0.0348598 | 9.822517 | 0 | 0.2740870 | 0.4107349 |
| IAa14 | ~~ | IAa14 | 0.5115718 | 0.0501136 | 10.208248 | 0 | 0.4133510 | 0.6097926 |
| IAa15 | ~~ | IAa15 | 0.5515190 | 0.0510740 | 10.798424 | 0 | 0.4514157 | 0.6516222 |
| F1 | ~~ | F1 | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |

O modelo de uma única variável latente "F1" apresenta um ajuste geral adequado aos dados, conforme indicado pelo teste qui-quadrado significativo ( $^2 = 45.034$, df = 9, p = 0.000).

As cargas fatoriais dos indicadores para "F1" são todas estatisticamente significativas ($p < 0.001$), indicando que essas variáveis observadas têm uma relação com a variável latente "F1".

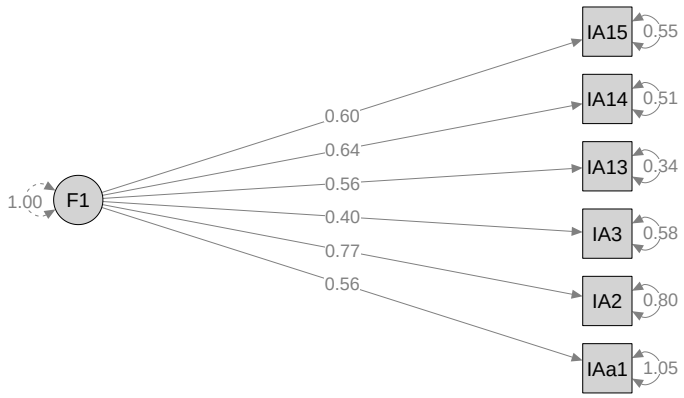
### 10.4.2 Índices de qualidade do modelo 3

```
model_performance(cfa_modelo_3, metrics = c("Chi2", "Chi2_df", "NFI", "CFI", "RMSEA", "p_RMS
```

```
# Indices of model performance

Chi2(9) |   NFI |   CFI | RMSEA | p (RMSEA) |      AIC |      BIC |  NNFI
-------------------------------------------------------------------------
45.034  | 0.906 | 0.922 | 0.107 |     0.001 | 5406.344 | 5452.536 | 0.870
```

### 10.4.3 Diagrama do modelo 3

```
plot_CFA <- semPaths(
          object = cfa_modelo_3,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```




### 10.4.4 Índices de modificação para o modelo

```
kable(modificationindices(cfa_modelo_3, standardized = FALSE, minimum.value = 5))
```

| | lhs | op | rhs | mi | epc |
|---|---|---|---|---|---|
| 16 | IAa1 | ~~ | IAa13 | 6.419170 | -0.1034145 |
| 19 | IAa2 | ~~ | IAa3 | 5.260435 | 0.1017547 |
| 21 | IAa2 | ~~ | IAa14 | 19.900038 | -0.2224777 |
| 22 | IAa2 | ~~ | IAa15 | 6.284104 | 0.1223301 |
| 26 | IAa13 | ~~ | IAa14 | 24.013115 | 0.1712177 |

## 10.5 Modelo 4 com covariância entre os resíduos

```
cfa_eq_4 = "
F1 =~ IAa1 + IAa2 + IAa3 + IAa13 + IAa14 + IAa15

#Covariância dos resíduos
IAa2    ~~  IAa14
IAa13   ~~  IAa14
"
```

**Análise Fatorial Confirmatória do modelo 4**

```
cfa_modelo_4 = cfa(
  model = cfa_eq_4,
  data = dados_CFA,
  std.lv = TRUE
)
```

### 10.5.1 Resultados do modelo 4

```
kable(parameterEstimates(cfa_modelo_4))
```

| lhs | op | rhs | est | se | z | pvalue | ci.lower | ci.upper |
|---|---|---|---|---|---|---|---|---|
| F1 | =~ | IAa1 | 0.5755782 | 0.0658579 | 8.739696 | 0.0000000 | 0.4464990 | 0.7046573 |
| F1 | =~ | IAa2 | 0.8812882 | 0.0659778 | 13.357336 | 0.0000000 | 0.7519740 | 1.0106024 |
| F1 | =~ | IAa3 | 0.4106200 | 0.0490163 | 8.377216 | 0.0000000 | 0.3145499 | 0.5066902 |
| F1 | =~ | IAa13 | 0.4773285 | 0.0461558 | 10.341675 | 0.0000000 | 0.3868648 | 0.5677923 |
| F1 | =~ | IAa14 | 0.6235655 | 0.0604044 | 10.323181 | 0.0000000 | 0.5051751 | 0.7419560 |
| F1 | =~ | IAa15 | 0.5905868 | 0.0525289 | 11.243084 | 0.0000000 | 0.4876320 | 0.6935415 |
| IAa2 | ~~ | IAa14 | -0.1604716 | 0.0490176 | -3.273757 | 0.0010613 | -0.2565442 | -0.0643989 |
| IAa13 | ~~ | IAa14 | 0.1249518 | 0.0402391 | 3.105235 | 0.0019013 | 0.0460846 | 0.2038189 |
| IAa1 | ~~ | IAa1 | 1.0265817 | 0.0852570 | 12.041031 | 0.0000000 | 0.8594811 | 1.1936823 |
| IAa2 | ~~ | IAa2 | 0.6085266 | 0.0820552 | 7.416062 | 0.0000000 | 0.4477013 | 0.7693519 |
| IAa3 | ~~ | IAa3 | 0.5777805 | 0.0475147 | 12.160028 | 0.0000000 | 0.4846534 | 0.6709077 |
| IAa13 | ~~ | IAa13 | 0.4241519 | 0.0395633 | 10.720852 | 0.0000000 | 0.3466093 | 0.5016945 |
| IAa14 | ~~ | IAa14 | 0.5298461 | 0.0642606 | 8.245273 | 0.0000000 | 0.4038976 | 0.6557945 |
| IAa15 | ~~ | IAa15 | 0.5578837 | 0.0519546 | 10.737912 | 0.0000000 | 0.4560546 | 0.6597128 |
| F1 | ~~ | F1 | 1.0000000 | 0.0000000 | NA | NA | 1.0000000 | 1.0000000 |

Os resultados do modelo sugerem que o ajuste do modelo aos dados é razoável, conforme indicado pelo teste qui-quadrado ( $^2$ = 12.216, df = 7, p = 0.094). O modelo envolve uma única variável latente "F1," e por seis variáveis observadas (IAa1, IAa2, IAa3, IAa13, IAa14, IAa15). As cargas fatoriais associadas a cada indicador são todas estatisticamente significativas (p < 0.001), indicando uma relação entre esses indicadores e a variável latente "F1". As variâncias dos indicadores também são significativas, sugerindo que cada indicador contribui para a variabilidade total da variável latente "F1".

Além disso, há duas covariâncias estimadas entre os indicadores: uma entre IAa2 e IAa14, e outra entre IAa13 e IAa14. Essas covariâncias indicam associações adicionais entre os indicadores além daquelas explicadas pelas relações com a variável latente "F1".

### 10.5.2 Índices de qualidade do modelo 4

```
model_performance(cfa_modelo_4, metrics = c("Chi2", "Chi2_df", "NFI", "CFI", "RMSEA", "p_RMS
```

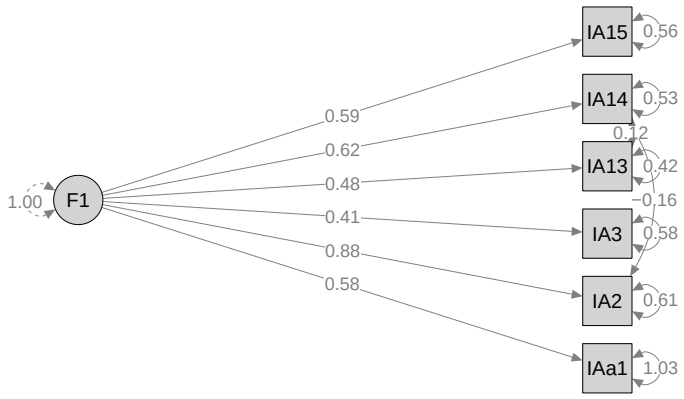
```
# Indices of model performance
```

```
Chi2(7) |   NFI |   CFI | RMSEA | p (RMSEA) |      AIC |      BIC |  NNFI
-------------------------------------------------------------------------
12.216  | 0.974 | 0.989 | 0.046 |     0.498 | 5377.526 | 5431.416 | 0.976
```

O modelo apresenta índices NFI (0.974), CFI (0.989) e NNFI (0.976) próximos de 1, indicando um bom ajuste. O RMSEA (0.046) é baixo, sugerindo uma adequada aproximação do modelo aos dados.

### 10.5.3 Diagrama do modelo 4

```
plot_CFA <- semPaths(
          object = cfa_modelo_4,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```




### 10.5.4 Comparação entre os modelos

```
compare_performance(cfa_modelo, cfa_modelo_2, cfa_modelo_3, cfa_modelo_4,
                    metrics = c("NFI", "CFI", "RMSEA", "p_RMSEA", "AIC", "BIC", "NNFI"),
                    rank = TRUE, verbose = F)
```

```
# Comparison of Model Performance Indices

Name         |  Model |   NFI |   CFI | RMSEA | p (RMSEA) |  NNFI | AIC weights | BIC weights
-----------------------------------------------------------------------------------------------
cfa_modelo_4 | lavaan | 0.974 | 0.989 | 0.046 |     0.498 | 0.976 |       0.952 |       0.999
cfa_modelo_2 | lavaan | 0.970 | 0.980 | 0.073 |     0.165 | 0.940 |       0.048 |       0.001
cfa_modelo   | lavaan | 0.918 | 0.933 | 0.106 |     0.003 | 0.874 |    3.64e-06 |    2.62e-05
cfa_modelo_3 | lavaan | 0.906 | 0.922 | 0.107 |     0.001 | 0.870 |    5.26e-07 |    2.59e-05
```

O modelo_4 demonstra superioridade em relação aos demais modelos com base nos critérios de ajuste avaliados.

```
# Links de referência

# https://rdrr.io/cran/performance/man/model_performance.lavaan.html
# https://methodenlehre.github.io/SGSCLM-R-course/cfa-and-sem-with-lavaan.html#structural-eq
```

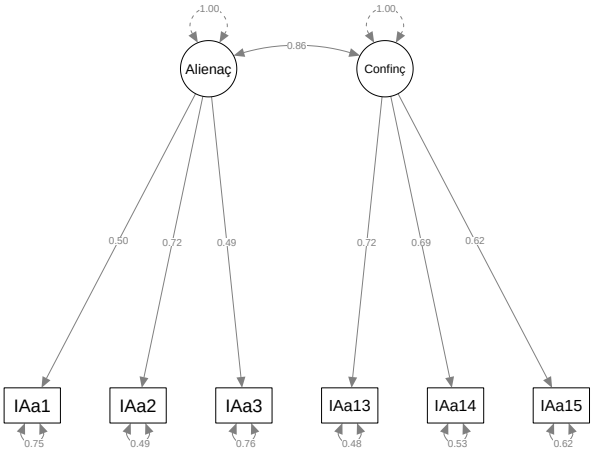
## 10.6 Lista 8 resolvida no SPSS

https://www.youtube.com/watch?v=f_fXWuCGssQ

## 10.7 Extras!

### 10.7.1 Mais gráficos

```
semPaths(cfa_modelo, "std", weighted = FALSE, nCharNodes = 7, shapeMan = "rectangle",
         sizeMan = 8, sizeMan2 = 5)
```




## 10.8 Referências

https://www.jstatsoft.org/article/view/v048i02

https://lavaan.ugent.be/tutorial/inspect.html

## 10.9 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages effectsize
(version 0.8.6; Ben-Shachar MS et al., 2020), semPlot (version 1.1.6; Epskamp
S, 2022), lubridate (version 1.9.3; Grolemund G, Wickham H, 2011), parameters
(version 0.21.3; Lüdecke D et al., 2020), performance (version 0.10.8; Lüdecke
D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see
(version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et
al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased
(version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et
al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), tibble
(version 3.2.1; Müller K, Wickham H, 2023), datawizard (version 0.9.0; Patil I
et al., 2022), foreign (version 0.8.85; R Core Team, 2023), lavaan (version
0.6.16; Rosseel Y, 2012), ggplot2 (version 3.4.4; Wickham H, 2016), forcats
(version 1.0.0; Wickham H, 2023), stringr (version 1.5.1; Wickham H, 2023),
tidyverse (version 2.0.0; Wickham H et al., 2019), dplyr (version 1.1.3;
Wickham H et al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), readr
(version 2.1.4; Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et
al., 2023) and kableExtra (version 1.3.4; Zhu H, 2021).

References
----------
  - Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of
Effect Size Indices and Standardized Parameters." _Journal of Open Source
Software_, *5*(56), 2815. doi:10.21105/joss.02815
<https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
  - Epskamp S (2022). _semPlot: Path Diagrams and Visual Analysis of Various SEM
Packages' Output_. R package version 1.1.6,
<https://CRAN.R-project.org/package=semPlot>.
  - Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate."
_Journal of Statistical Software_, *40*(3), 1-25.
<https://www.jstatsoft.org/v40/i03/>.
  - Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing
and Exploring the Parameters of Statistical Models using R." _Journal of Open
Source Software_, *5*(53), 2445. doi:10.21105/joss.02445
<https://doi.org/10.21105/joss.02445>.
```

- Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021). "performance: An R Package for Assessment, Comparison and Testing of Statistical Models." _Journal of Open Source Software_, *6*(60), 3139. doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
- Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_. R package, <https://easystats.github.io/easystats/>.
- Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021). "see: An R Package for Visualizing Statistical Models." _Journal of Open Source Software_, *6*(64), 3393. doi:10.21105/joss.03393 <https://doi.org/10.21105/joss.03393>.
- Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to Access Information from Model Objects in R." _Journal of Open Source Software_, *4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
- Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework." _Journal of Open Source Software_, *4*(40), 1541. doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>, <https://joss.theoj.org/papers/10.21105/joss.01541>.
- Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of Model-Based Predictions, Contrasts and Means." _CRAN_. <https://github.com/easystats/modelbased>.
- Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023). "Automated Results Reporting as a Practical Tool to Improve Reproducibility and Methodological Best Practices Adoption." _CRAN_. <https://easystats.github.io/report/>.
- Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022). "correlation: Methods for Correlation Analysis." Version 0.8.3, <https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306 <https://doi.org/10.21105/joss.02306>, <https://joss.theoj.org/papers/10.21105/joss.02306>.
- Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version 3.2.1, <https://CRAN.R-project.org/package=tibble>.
- Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022). "datawizard: An R Package for Easy Data Preparation and Statistical Transformations." _Journal of Open Source Software_, *7*(78), 4684. doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.
- R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85,

<https://CRAN.R-project.org/package=foreign>.
  - R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
  - Rosseel Y (2012). "lavaan: An R Package for Structural Equation Modeling." _Journal of Statistical Software_, *48*(2), 1-36. doi:10.18637/jss.v048.i02 <https://doi.org/10.18637/jss.v048.i02>.
  - Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
  - Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
  - Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
  - Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
  - Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.
  - Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.
  - Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.
  - Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.
  - Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4, <https://CRAN.R-project.org/package=kableExtra>.

# 11 Lista 8.1 - Moderação e Mediação

Veja o banco de dados DADOSPATH.sav. Nele temos as variáveis Idade, IMC, Sociabilidade (medida por um questionário) e número de treinos realizados em uma academia. Temos uma hipótese teórica de que SOCIABILIDADE é uma variável preditora (X) do número de TREINOS (Y) que a pessoa realiza na academia. No entanto, o IMC (M) é uma variável mediadora desse efeito, não apenas moderadora. Ou seja, a relação entre sociabilidade e treinos só aparece na presença do IMC ou quando ele é conhecido.

```
library(foreign)
library(tidyverse)
library(lavaan)
library(semPlot)
library(performance)
library(easystats)
library(semPlot)
library(semTools)
library(flexplot)
library(mediation)
library(kableExtra)
```

## 11.1 a) Modelo causal teórico

> **Exercício**
>
> Verifique esse modelo causal teórico e veja se ele faz sentido, utilizando um modelo SEM com mediação. Avalie os efeitos diretos e indiretos e decida se esse modelo teórico faz sentido, utilizando o AMOS e o Process.

Resolução do exercício foi baseada no vídeo "Simple Mediation using lavaan package of R" https://www.youtube.com/watch?v=nfQOCy9xMnk

```
original = read.spss("DADOS PATH.sav", to.data.frame=TRUE)
glimpse(original)
```

```
Rows: 94
Columns: 4
$ Idade         <dbl> 57, 41, 29, 26, 33, 37, 26, 44, 31, 36, 30, 55, 43, 27, ~
$ IMC1          <dbl> 28.46122, 32.62609, 26.99050, 19.03602, 28.76650, 24.686~
$ Treinos       <dbl> 108, 144, 48, 102, 123, 63, 39, 105, 6, 48, 144, 144, 11~
$ Sociabilidade <dbl> 14, 34, 17, 20, 26, 19, 15, 32, 27, 19, 28, 9, 30, 29, 1~
```

```
modelo_1 = "Treinos ~ c_*Sociabilidade + b*IMC1
            IMC1 ~ a*Sociabilidade
            Indireto := a*b
            Total_direto_C := a*b + c_"
```

```
fit_1 = sem(modelo_1, original, se = "bootstrap", bootstrap = 500)
```

```
#summary(fit_1) # parâmetros adicionais summary(fit_1, fit.measures = TRUE, standardized = T

kable(parameterEstimates(fit_1))
```

| lhs | op | rhs | label | est | se | z | pvalue |
|---|---|---|---|---|---|---|---|
| Treinos | ~ | Sociabilidade | c_ | 0.6052243 | 0.6540394 | 0.9253637 | 0.3547768 |
| Treinos | ~ | IMC1 | b | -1.6497656 | 1.0250511 | -1.6094471 | 0.1075186 |
| IMC1 | ~ | Sociabilidade | a | 0.0190526 | 0.0512185 | 0.3719857 | 0.7099035 |
| Treinos | ~~ | Treinos | | 2327.0247317 | 184.1922135 | 12.6336759 | 0.0000000 |
| IMC1 | ~~ | IMC1 | | 17.7329732 | 3.0881294 | 5.7423025 | 0.0000000 |
| Sociabilidade | ~~ | Sociabilidade | | 65.6008375 | 0.0000000 | NA | NA |
| Indireto | := | a*b | Indireto | -0.0314322 | 0.0956155 | -0.3287358 | 0.7423554 |
| Total_direto_C | := | a*b+c_ | Total_direto_C | 0.5737921 | 0.6492666 | 0.8837542 | 0.3768289 |

### 11.1.1 Resultados

1. **Regressões:**

- O coeficiente estimado para a relação entre Sociabilidae e Treinos é 0.605, mas não é estatisticamente significativo (p = 0.360).
- O coeficiente estimado para a relação entre IMC e Treinos é -1.650, indicando uma relação negativa. No entanto, esse coeficiente também não é estatisticamente significativo (p = 0.100).
- O coeficiente estimado para a relação entre Sociabilidae e IMC é 0.019 e não é estatisticamente significativo (p = 0.693).

2. **Parâmetros Definidos:**

- O efeito indireto é estimado como -0.031, mas não é estatisticamente significativo (p = 0.717). Isso sugere que a variável IMC não medeia significativamente a relação entre Sociabilidae e Treinos.
- O efeito direto da Sociabilidade no Treino é estimado como 0.574 e também não é estatisticamente significativo (p = 0.386).

Com base nos resultados, podemos concluir que o modelo teórico não se sustenta, pois não há evidência estatística significativa para sugerir relações entre as variáveis Sociabilidae , IMC e Treinos.

## 11.2 b) Mediação vs Regressões lineares

> **Exercício**
>
> Compare os dados encontrados com aqueles realizados por um conjunto de regressões lineares (OLS). Fazer esta análise de mediação por regressão linear e utilizando o AMOS+Process é a mesma coisa? Coloque também o diagrama gerado aqui.

### 11.2.1 Valor de "c"

```
soc_treinos = lm(Treinos ~ Sociabilidade, data = original) #valor de c
kable(summary(soc_treinos)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 69.4395522 | 14.1638541 | 4.9025888 | 0.0000041 |
| Sociabilidade | 0.5737921 | 0.6273496 | 0.9146289 | 0.3627775 |

### 11.2.2 Valor de a

```
soc_imc = lm(IMC1 ~ Sociabilidade, data = original)
kable(summary(soc_imc)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 25.3971491 | 1.2238100 | 20.7525264 | 0.0000000 |
| Sociabilidade | 0.0190526 | 0.0542054 | 0.3514884 | 0.7260257 |

### 11.2.3 valor de b e de c'

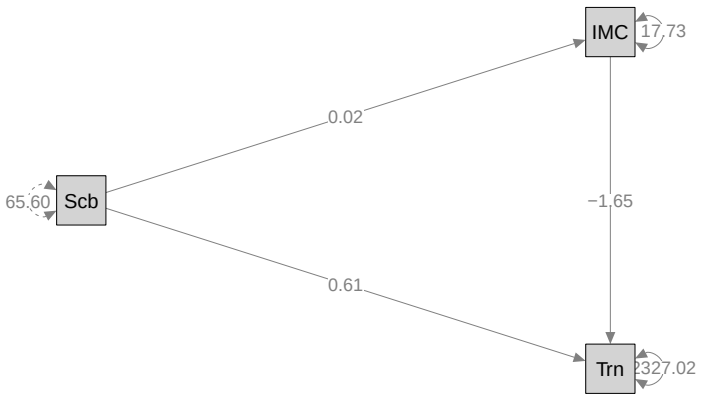
```
soc_E_imc_treinos = lm(Treinos ~ IMC1 + Sociabilidade, data = original)
kable(summary(soc_E_imc_treinos)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 111.3388945 | 33.5981763 | 3.3138374 | 0.0013220 |
| IMC1 | -1.6497656 | 1.2008507 | -1.3738307 | 0.1728690 |
| Sociabilidade | 0.6052243 | 0.6247647 | 0.9687235 | 0.3352508 |

Os resultados são diferentes. As mediações apenas por regressão linear não apresentam o resultado do efeito indireto, mostrado no resultado do exercício anterior

### 11.2.4 Diagrama do modelo

```
diagrama_1 <- semPaths(
          object = fit_1,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```



## 11.3 Modelo 2 (Opcional 1)

Refaça o modelo tendo a variável Idade como mediador.

```
modelo_2 = "Treinos ~ c_*Sociabilidade + b*Idade
            Idade ~ a*Sociabilidade
            Indireto := a*b
            Total_direto_C := a*b + c_"
```

```
fit_2 = sem(modelo_2, original, se = "bootstrap", bootstrap = 500)

kable(parameterEstimates(fit_2)) # parâmetros adicionais summary(fit_1, fit.measures = TRUE,
```

| lhs | op | rhs | label | est | se | z | pvalue |
|---|---|---|---|---|---|---|---|
| Treinos | ~ | Sociabilidade | c_ | 0.4852942 | 0.6154592 | 0.7885075 | 0.4303999 |
| Treinos | ~ | Idade | b | 1.5425565 | 0.4990713 | 3.0908540 | 0.0019958 |
| Idade | ~ | Sociabilidade | a | 0.0573709 | 0.1112607 | 0.5156440 | 0.6061030 |
| Treinos | ~~ | Treinos | | 2179.2955768 | 214.4060865 | 10.1643363 | 0.0000000 |
| Idade | ~~ | Idade | | 82.3681677 | 10.4675791 | 7.8688842 | 0.0000000 |
| Sociabilidade | ~~ | Sociabilidade | | 65.6008375 | 0.0000000 | NA | NA |
| Indireto | := | a*b | Indireto | 0.0884979 | 0.1736466 | 0.5096438 | 0.6103010 |
| Total_direto_C | := | a*b+c_ | Total_direto_C | 0.5737921 | 0.6701349 | 0.8562336 | 0.3918686 |

### 11.3.1 Resultados

1. **Regressões:**

   - A relação estimada entre `Sociabilidade` e `Treinos` é 0.485, mas não é estatisticamente significativa (p = 0.384).
   - A relação estimada entre `Idade` e `Treinos` é 1.543, indicando uma relação positiva e significativa (p = 0.001).
   - A relação estimada entre `Sociabldd` e `Idade` é 0.057 e não é estatisticamente significativa (p = 0.592).

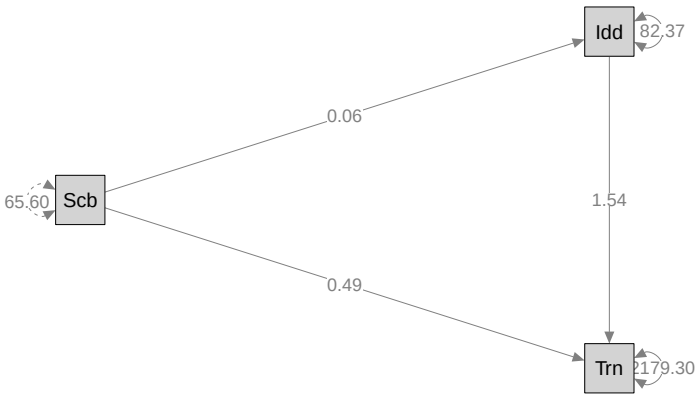
2. **Parâmetros Definidos:**

- O efeito `Indireto` é estimado como 0.088, mas não é estatisticamente significativo (p = 0.605). Isso sugere que a variável `Idade` não medeia significativamente a relação entre `Sociabilidade` e `Treinos`.
- O efeito total direto da Sociabilidae nos Treinos é estimado como 0.574 e não é estatisticamente significativo (p = 0.345).

Os resultados sugerem que a variável `Idade` está significativamente relacionada à variável `Treinos`, enquanto a variável `Sociabilidade` não tem uma relação significativa com `Treinos`. O efeito indireto através de `Idade` não é estatisticamente significativo, e o efeito total direto também não é significativo.

### 11.3.2 Diagrama do modelo 2

```
diagrama_2 <- semPaths(
          object = fit_2,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```




```
mediation_model_2 = lm(Idade ~ Sociabilidade, data = original)
kable(summary(mediation_model_2)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 32.8228531 | 2.6375635 | 12.4443843 | 0.0000000 |
| Sociabilidade | 0.0573709 | 0.1168237 | 0.4910896 | 0.6245325 |

```
# library(flexplot)
# visualize(mediation_model_2) análise gráfica do modelo
```

```
full_model_2 = lm(Treinos ~ Idade + Sociabilidade, data = original)
kable(summary(full_model_2)$coef)
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 18.8084463 | 22.3454098 | 0.8417141 | 0.4021546 |
| Idade | 1.5425565 | 0.5392097 | 2.8607731 | 0.0052418 |
| Sociabilidade | 0.4852942 | 0.6049941 | 0.8021469 | 0.4245578 |

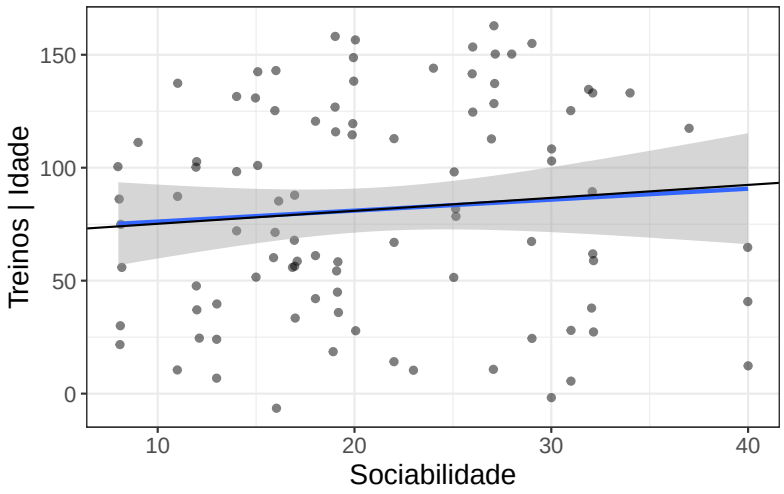
```
#visualize(full_model_2) análise gráfica do modelo
```

```
results_2 = mediate(mediation_model_2, full_model_2,
                    treat = "Sociabilidade",
                    mediator = "Idade",
```

```
                  boot = TRUE,
                  sims = 500)

summary(results_2)
```

```
Causal Mediation Analysis

Nonparametric Bootstrap Confidence Intervals with the Percentile Method

               Estimate 95% CI Lower 95% CI Upper p-value
ACME             0.0885      -0.2453         0.48    0.66
ADE              0.4853      -0.6971         1.63    0.43
Total Effect     0.5738      -0.7406         1.88    0.41
Prop. Mediated   0.1542      -1.6558         1.96    0.58

Sample Size Used: 94


Simulations: 500
```

```
mediate_plot(Treinos ~ Idade  + Sociabilidade  , data = original)
```



## 11.4 Modelo 3 (Opcional 2)

Testando outros modelos, foi possível observar que o efeito do IMC sobre o treinamento é mediado pela Idade

```
modelo_3 = "Treinos ~ c_*IMC1 + b*Idade
            Idade ~ a*IMC1
            Indireto := a*b
            Total_direto_C := a*b + c_"
```

```
fit_3 = sem(modelo_3, original, se = "bootstrap", bootstrap = 500) #demora um tempo para exe
```

```
kable(parameterEstimates(fit_3)) # parâmetros adicionais summary(fit_1, fit.measures = TRUE,
```

| lhs | op | rhs | label | est | se | z | pvalue | |
|---|---|---|---|---|---|---|---|---|
| Treinos | ~ | IMC1 | c_ | -2.7781642 | 0.9924376 | -2.799334 | 0.0051208 | - |
| Treinos | ~ | Idade | b | 1.9275620 | 0.4755354 | 4.053456 | 0.0000505 | |
| Idade | ~ | IMC1 | a | 0.6075027 | 0.2314968 | 2.624238 | 0.0086843 | |
| Treinos | ~~ | Treinos | | 2068.5298836 | 200.0458615 | 10.340278 | 0.0000000 | 162 |
| Idade | ~~ | Idade | | 76.0307760 | 10.2233210 | 7.436994 | 0.0000000 | 5 |
| IMC1 | ~~ | IMC1 | | 17.7567863 | 0.0000000 | NA | NA | 1 |
| Indireto | := | a*b | Indireto | 1.1709992 | 0.5288095 | 2.214406 | 0.0268008 | |
| Total_direto_C | := | a*b+c_ | Total_direto_C | -1.6071651 | 1.0688523 | -1.503636 | 0.1326750 | - |

### 11.4.1 Resultados

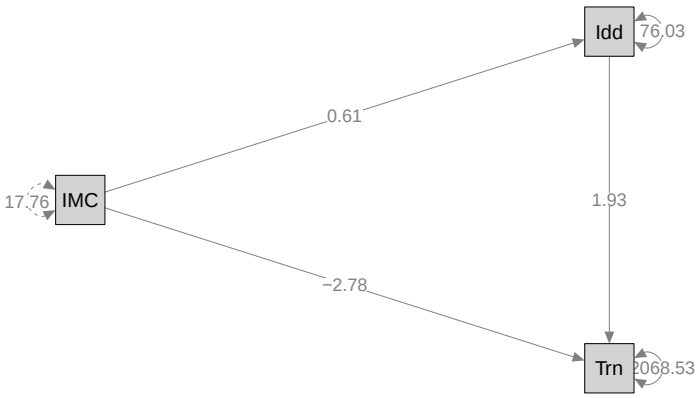
1. **Regressões:**
    - A relação estimada entre `IMC1` (Índice de Massa Corporal) e `Treinos` é -2.778, indicando uma relação negativa e significativa (p = 0.006).
    - A relação estimada entre `Idade` e `Treinos` é 1.928, indicando uma relação positiva e significativa (p = 0.000).
    - A relação estimada entre `IMC` e `Idade` é 0.608 e é estatisticamente significativa (p = 0.005).
2. **Parâmetros Definidos:**

- O efeito `Indireto` é estimado como 1.171 e é estatisticamente significativo (p = 0.030). Isso sugere que a variável `Idade` medeia significativamente a relação entre `IMC` e `Treinos`.
- O efeito total direto de IMC nos Treinos é estimado como -1.607, mas não é estatisticamente significativo (p = 0.114).

Os resultados indicam que a variável `IMC` está significativamente relacionada negativamente à variável `Treinos`. A variável `Idade` atua como mediadora nessa relação. O efeito indireto é estimado como 1.171 (p = 0.030), indicando que a inclusão de `Idade` no modelo altera a relação entre `IMC1` e `Treinos`, tornando-a mais negativa do que a relação direta.

## 11.5 Diagrama do modelo 3

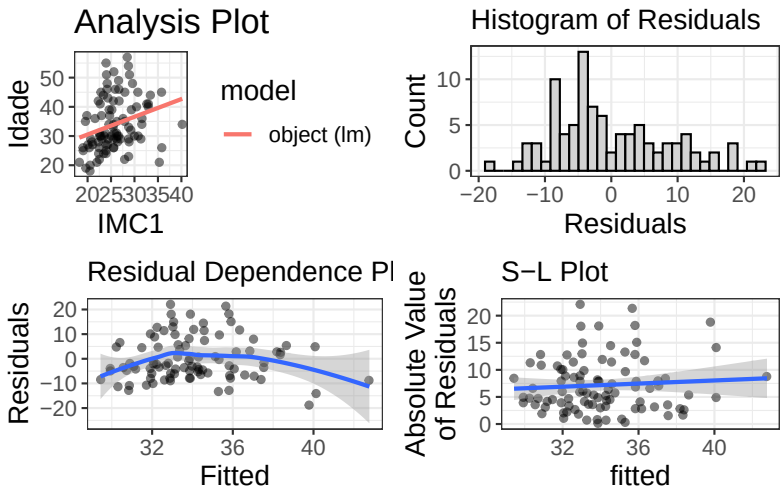
```
diagrama_3 <- semPaths(
          object = fit_3,
          what = "path",
          whatLabels = "par",
          style = "ram",
          layout = "tree",
          rotation = 2,
          sizeMan = 7,
          sizeLat = 7,
          color = "lightgray",
          edge.label.cex = 1.2,
          label.cex = 1.3
)
```




```
mediation_model_3 = lm(Idade ~ IMC1, data = original)
summary(mediation_model_3)$coef
```

```
              Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 18.3591515   5.639405 3.255512 0.001584908
IMC1         0.6075027   0.215734 2.815980 0.005949584
```

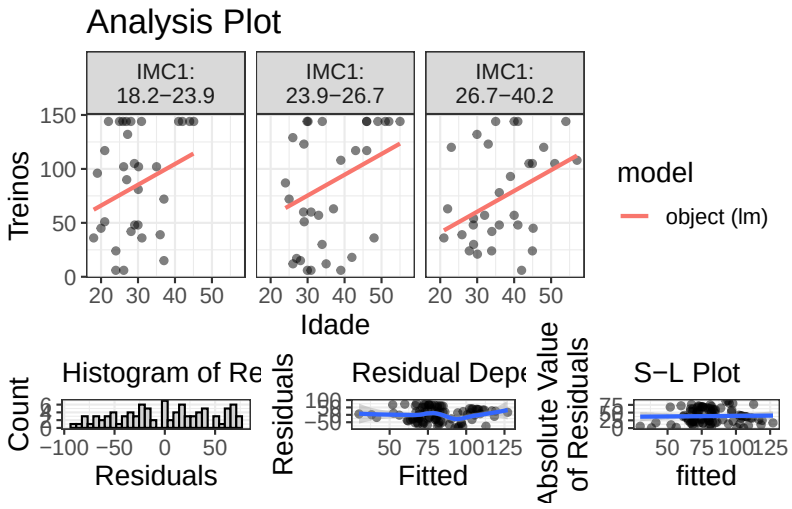
```
visualize(mediation_model_3)
```



```r
full_model_3 = lm(Treinos ~ Idade + IMC1, data = original)
summary(full_model_3)$coef
```

```
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 87.606237 31.2333809  2.804891 0.0061538896
Idade        1.927562  0.5467836  3.525274 0.0006645242
IMC1        -2.778164  1.1791838 -2.356006 0.0206193649
```

```r
visualize(full_model_3)
```

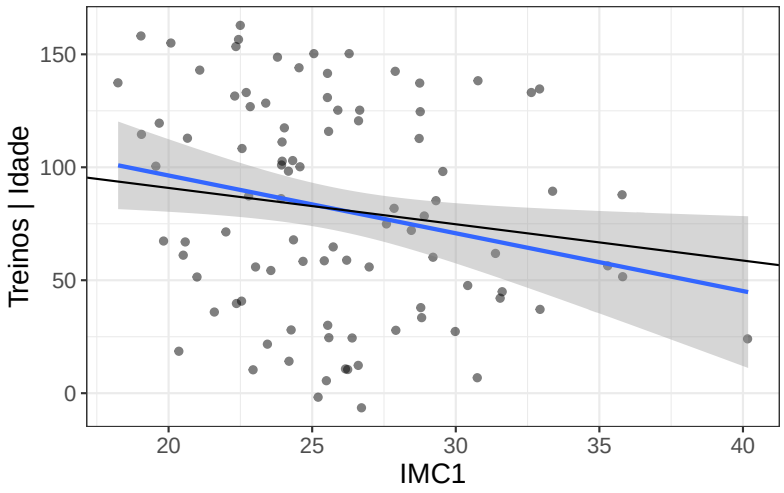



```r
results_3 = mediate(mediation_model_3, full_model_3,
                    treat = "IMC1",
                    mediator = "Idade",
                    boot = TRUE,
                    sims = 500)

summary(results_3)
```

```
Causal Mediation Analysis

Nonparametric Bootstrap Confidence Intervals with the Percentile Method
```

```
               Estimate 95% CI Lower 95% CI Upper p-value
ACME              1.171        0.325         2.38  <2e-16 ***
ADE              -2.778       -4.753        -0.99   0.012 *
Total Effect     -1.607       -3.677         0.54   0.124
Prop. Mediated   -0.729       -8.856         3.85   0.124
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Sample Size Used: 94


Simulations: 500
```

```
mediate_plot(Treinos ~ Idade  + IMC1  , data = original)
```




## 11.6 Lista 8.1 resolvida no SPSS

https://www.youtube.com/watch?v=NstttDePtcM

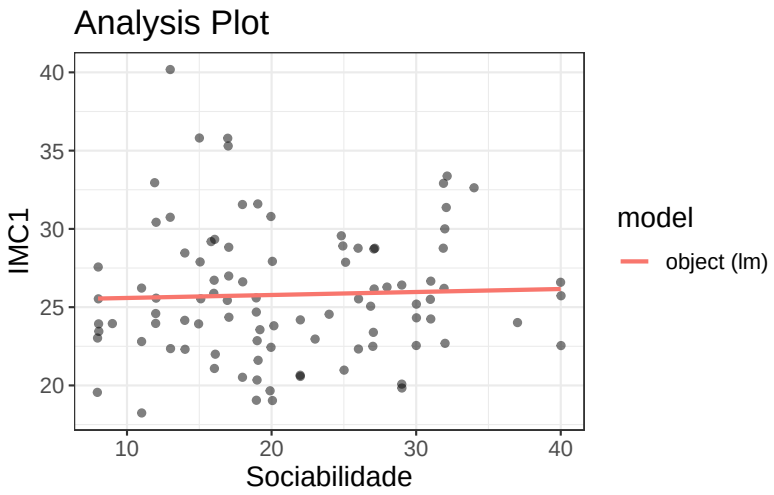
## 11.7 Extras!

Outro tipo de resolução baseada no vídeo do Dustin Fife (How to do a mediation analysis in R...with visuals!)

```
# Mediação com visualização
library(mediation)
library(flexplot)


mediation_model = lm(IMC1 ~ Sociabilidade, data = original)
summary(mediation_model)
```

```
Call:
lm(formula = IMC1 ~ Sociabilidade, data = original)

Residuals:
    Min      1Q  Median      3Q     Max
-7.3645 -2.8931 -0.4598  2.7881 14.5354

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   25.39715    1.22381  20.753   <2e-16 ***
Sociabilidade  0.01905    0.05421   0.351    0.726
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.257 on 92 degrees of freedom
Multiple R-squared:  0.001341,  Adjusted R-squared:  -0.009514
F-statistic: 0.1235 on 1 and 92 DF,  p-value: 0.726
```

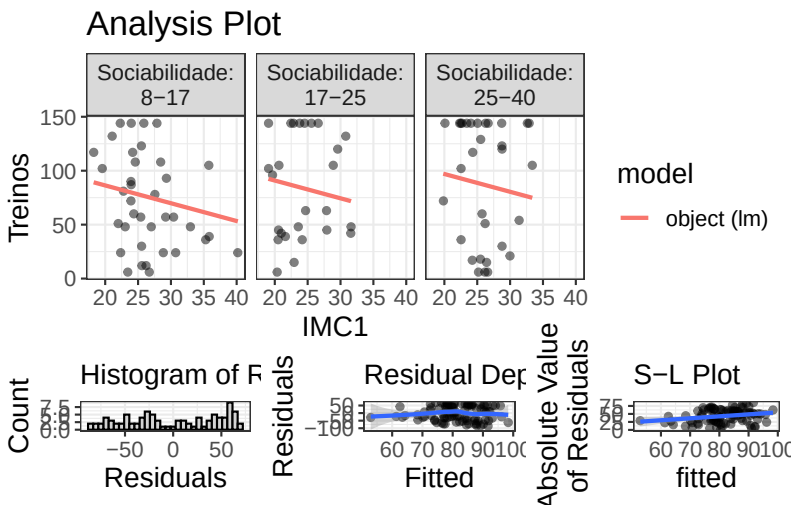
```
visualize(mediation_model, plot = "model")
```




```
full_model = lm(Treinos ~ IMC1 + Sociabilidade, data = original)
summary(full_model)
```

```
Call:
lm(formula = Treinos ~ IMC1 + Sociabilidade, data = original)

Residuals:
    Min      1Q  Median      3Q     Max
-85.683 -42.165   2.807  47.623  69.596

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  111.3389    33.5982   3.314  0.00132 **
IMC1          -1.6498     1.2009  -1.374  0.17287
Sociabilidade  0.6052     0.6248   0.969  0.33525
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49.03 on 91 degrees of freedom
Multiple R-squared:  0.02915,   Adjusted R-squared:  0.00781
F-statistic: 1.366 on 2 and 91 DF,  p-value: 0.2603
```

```r
visualize(full_model)
```



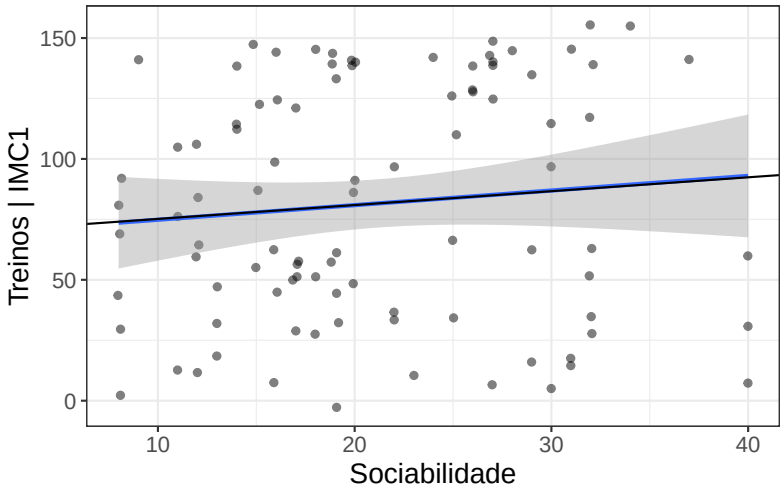

```r
results = mediate(mediation_model, full_model,
                  treat = "Sociabilidade",
                  mediator = "IMC1",
                  boot = TRUE,
                  sims = 500)
```

```r
summary(results)
```

```
Causal Mediation Analysis

Nonparametric Bootstrap Confidence Intervals with the Percentile Method

               Estimate 95% CI Lower 95% CI Upper p-value
ACME            -0.0314      -0.2155         0.20    0.79
ADE              0.6052      -0.6084         2.05    0.36
Total Effect     0.5738      -0.5997         2.05    0.36
Prop. Mediated  -0.0548      -0.8745         0.88    0.93

Sample Size Used: 94
```

```
Simulations: 500
```

```
mediate_plot(Treinos ~ IMC1 +  Sociabilidade, data = original) # Ordem em que aparece as var
```




## 11.8 Referências

https://www.youtube.com/watch?v=_4Fu8SZID2k

## 11.9 Versões dos pacotes

```
report(sessionInfo())
```

```
Analyses were conducted using the R Statistical language (version 4.3.1; R Core
Team, 2023) on Windows 11 x64 (build 22621), using the packages Matrix (version
1.6.0; Bates D et al., 2023), effectsize (version 0.8.6; Ben-Shachar MS et al.,
2020), semPlot (version 1.1.6; Epskamp S, 2022), flexplot (version 0.20.5; Fife
D, 2024), mvtnorm (version 1.2.3; Genz A, Bretz F, 2009), lubridate (version
1.9.3; Grolemund G, Wickham H, 2011), semTools (version 0.5.6; Jorgensen TD et
al., 2022), parameters (version 0.21.3; Lüdecke D et al., 2020), performance
```

(version 0.10.8; Lüdecke D et al., 2021), easystats (version 0.6.0; Lüdecke D et al., 2022), see (version 0.8.1; Lüdecke D et al., 2021), insight (version 0.19.6; Lüdecke D et al., 2019), bayestestR (version 0.13.1; Makowski D et al., 2019), modelbased (version 0.8.6; Makowski D et al., 2020), report (version 0.5.7; Makowski D et al., 2023), correlation (version 0.8.4; Makowski D et al., 2022), tibble (version 3.2.1; Müller K, Wickham H, 2023), datawizard (version 0.9.0; Patil I et al., 2022), foreign (version 0.8.85; R Core Team, 2023), lavaan (version 0.6.16; Rosseel Y, 2012), mediation (version 4.5.0; Tingley D et al., 2014), MASS (version 7.3.60; Venables WN, Ripley BD, 2002), ggplot2 (version 3.4.4; Wickham H, 2016), forcats (version 1.0.0; Wickham H, 2023), stringr (version 1.5.1; Wickham H, 2023), tidyverse (version 2.0.0; Wickham H et al., 2019), dplyr (version 1.1.3; Wickham H et al., 2023), purrr (version 1.0.2; Wickham H, Henry L, 2023), readr (version 2.1.4; Wickham H et al., 2023), tidyr (version 1.3.0; Wickham H et al., 2023), sandwich (version 3.1.0; Zeileis A et al., 2020) and kableExtra (version 1.3.4; Zhu H, 2021).

References
----------


- Bates D, Maechler M, Jagan M (2023). _Matrix: Sparse and Dense Matrix Classes and Methods_. R package version 1.6-0, <https://CRAN.R-project.org/package=Matrix>.
- Ben-Shachar MS, Lüdecke D, Makowski D (2020). "effectsize: Estimation of Effect Size Indices and Standardized Parameters." _Journal of Open Source Software_, *5*(56), 2815. doi:10.21105/joss.02815 <https://doi.org/10.21105/joss.02815>, <https://doi.org/10.21105/joss.02815>.
- Epskamp S (2022). _semPlot: Path Diagrams and Visual Analysis of Various SEM Packages' Output_. R package version 1.1.6, <https://CRAN.R-project.org/package=semPlot>.
- Fife D (2024). _flexplot: Graphically Based Data Analysis Using 'flexplot'_. R package version 0.20.5.
- Genz A, Bretz F (2009). _Computation of Multivariate Normal and t Probabilities_, series Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.
- Grolemund G, Wickham H (2011). "Dates and Times Made Easy with lubridate." _Journal of Statistical Software_, *40*(3), 1-25. <https://www.jstatsoft.org/v40/i03/>.
- Jorgensen TD, Pornprasertmanit S, Schoemann AM, Rosseel Y (2022). _\texttt{semTools}: Useful tools for structural equation modeling_. R package version 0.5-6, <https://CRAN.R-project.org/package=semTools>.
- Lüdecke D, Ben-Shachar M, Patil I, Makowski D (2020). "Extracting, Computing and Exploring the Parameters of Statistical Models using R." _Journal of Open

Source Software_, *5*(53), 2445. doi:10.21105/joss.02445
<https://doi.org/10.21105/joss.02445>.
  - Lüdecke D, Ben-Shachar M, Patil I, Waggoner P, Makowski D (2021).
"performance: An R Package for Assessment, Comparison and Testing of
Statistical Models." _Journal of Open Source Software_, *6*(60), 3139.
doi:10.21105/joss.03139 <https://doi.org/10.21105/joss.03139>.
  - Lüdecke D, Ben-Shachar M, Patil I, Wiernik B, Makowski D (2022). "easystats:
Framework for Easy Statistical Modeling, Visualization, and Reporting." _CRAN_.
R package, <https://easystats.github.io/easystats/>.
  - Lüdecke D, Patil I, Ben-Shachar M, Wiernik B, Waggoner P, Makowski D (2021).
"see: An R Package for Visualizing Statistical Models." _Journal of Open Source
Software_, *6*(64), 3393. doi:10.21105/joss.03393
<https://doi.org/10.21105/joss.03393>.
  - Lüdecke D, Waggoner P, Makowski D (2019). "insight: A Unified Interface to
Access Information from Model Objects in R." _Journal of Open Source Software_,
*4*(38), 1412. doi:10.21105/joss.01412 <https://doi.org/10.21105/joss.01412>.
  - Makowski D, Ben-Shachar M, Lüdecke D (2019). "bayestestR: Describing Effects
and their Uncertainty, Existence and Significance within the Bayesian
Framework." _Journal of Open Source Software_, *4*(40), 1541.
doi:10.21105/joss.01541 <https://doi.org/10.21105/joss.01541>,
<https://joss.theoj.org/papers/10.21105/joss.01541>.
  - Makowski D, Ben-Shachar M, Patil I, Lüdecke D (2020). "Estimation of
Model-Based Predictions, Contrasts and Means." _CRAN_.
<https://github.com/easystats/modelbased>.
  - Makowski D, Lüdecke D, Patil I, Thériault R, Ben-Shachar M, Wiernik B (2023).
"Automated Results Reporting as a Practical Tool to Improve Reproducibility and
Methodological Best Practices Adoption." _CRAN_.
<https://easystats.github.io/report/>.
  - Makowski D, Wiernik B, Patil I, Lüdecke D, Ben-Shachar M (2022).
"correlation: Methods for Correlation Analysis." Version 0.8.3,
<https://CRAN.R-project.org/package=correlation>. Makowski D, Ben-Shachar M,
Patil I, Lüdecke D (2020). "Methods and Algorithms for Correlation Analysis in
R." _Journal of Open Source Software_, *5*(51), 2306. doi:10.21105/joss.02306
<https://doi.org/10.21105/joss.02306>,
<https://joss.theoj.org/papers/10.21105/joss.02306>.
  - Müller K, Wickham H (2023). _tibble: Simple Data Frames_. R package version
3.2.1, <https://CRAN.R-project.org/package=tibble>.
  - Patil I, Makowski D, Ben-Shachar M, Wiernik B, Bacher E, Lüdecke D (2022).
"datawizard: An R Package for Easy Data Preparation and Statistical
Transformations." _Journal of Open Source Software_, *7*(78), 4684.
doi:10.21105/joss.04684 <https://doi.org/10.21105/joss.04684>.

- R Core Team (2023). _foreign: Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ..._. R package version 0.8-85, <https://CRAN.R-project.org/package=foreign>.
- R Core Team (2023). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Rosseel Y (2012). "lavaan: An R Package for Structural Equation Modeling." _Journal of Statistical Software_, *48*(2), 1-36. doi:10.18637/jss.v048.i02 <https://doi.org/10.18637/jss.v048.i02>.
- Tingley D, Yamamoto T, Hirose K, Keele L, Imai K (2014). "mediation: R Package for Causal Mediation Analysis." _Journal of Statistical Software_, *59*(5), 1-38. <http://www.jstatsoft.org/v59/i05/>. Imai K, Keele L, Yamamoto T (2010). "Identification, Inference, and Sensitivity Analysis for Causal Mediation Effects." _Statistical Science_, *25*(1), 51-71. <http://imai.princeton.edu/research/mediation.html>. Imai K, Keele L, Tingley D (2010). "A General Approach to Causal Mediation Analysis." _Psychological Methods_, *15*(4), 309-334. <http://imai.princeton.edu/research/BaronKenny.html>. Imai K, Keele L, Tingley D, Yamamoto T (2011). "Unpacking the Black Box of Causality: Learning about Causal Mechanisms from Experimental and Observational Studies." _American Political Science Review_, *105*(4), 765-789. <http://imai.princeton.edu/research/mediationP.html>. Imai K, Yamamoto T (2013). "Identification and Sensitivity Analysis for Multiple Causal Mechanisms: Revisiting Evidence from Framing Experiments." _Political Analysis_, *21*(2), 141-171. <http://imai.princeton.edu/research/medsens.html>. Imai K, Keele L, Tingley D, Yamamoto T (2010). "Causal Mediation Analysis Using R." In Vinod HD (ed.), _Advances in Social Science Research Using R_. Springer-Verlag, New York.
- Venables WN, Ripley BD (2002). _Modern Applied Statistics with S_, Fourth edition. Springer, New York. ISBN 0-387-95457-0, <https://www.stats.ox.ac.uk/pub/MASS4/>.
- Wickham H (2016). _ggplot2: Elegant Graphics for Data Analysis_. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.
- Wickham H (2023). _forcats: Tools for Working with Categorical Variables (Factors)_. R package version 1.0.0, <https://CRAN.R-project.org/package=forcats>.
- Wickham H (2023). _stringr: Simple, Consistent Wrappers for Common String Operations_. R package version 1.5.1, <https://CRAN.R-project.org/package=stringr>.
- Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G,

Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.

- Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.3, <https://CRAN.R-project.org/package=dplyr>.

- Wickham H, Henry L (2023). _purrr: Functional Programming Tools_. R package version 1.0.2, <https://CRAN.R-project.org/package=purrr>.

- Wickham H, Hester J, Bryan J (2023). _readr: Read Rectangular Text Data_. R package version 2.1.4, <https://CRAN.R-project.org/package=readr>.

- Wickham H, Vaughan D, Girlich M (2023). _tidyr: Tidy Messy Data_. R package version 1.3.0, <https://CRAN.R-project.org/package=tidyr>.

- Zeileis A, Köll S, Graham N (2020). "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." _Journal of Statistical Software_, *95*(1), 1-36. doi:10.18637/jss.v095.i01 <https://doi.org/10.18637/jss.v095.i01>. Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." _Journal of Statistical Software_, *11*(10), 1-17. doi:10.18637/jss.v011.i10 <https://doi.org/10.18637/jss.v011.i10>. Zeileis A (2006). "Object-Oriented Computation of Sandwich Estimators." _Journal of Statistical Software_, *16*(9), 1-16. doi:10.18637/jss.v016.i09 <https://doi.org/10.18637/jss.v016.i09>.

- Zhu H (2021). _kableExtra: Construct Complex Table with 'kable' and Pipe Syntax_. R package version 1.3.4, <https://CRAN.R-project.org/package=kableExtra>.