

# Consistência Interna dos Dados usando R

Cinco maneiras de calcular a consistência interna

*Erivando Sena (Com adaptações do original de Simon Jackson - BLOGR)*

27/05/2019

*O objetivo desta versão adaptada utilizando R Notebook é reapresentar à análise realizada no idioma Português Brasileiro para que mais pessoas possam obter acesso ao conteúdo publicado originalmente em inglês no blog de Simon Jackson.*

Vamos obter psicometria e aprender uma série de maneiras de calcular a consistência interna de um teste ou questionário em R. Estaremos cobrindo:

1. Correlação média entre itens.
2. Correlação média total do item.
3. Alfa de Cronbach.
4. Confiabilidade dividida. *(ajustada usando a fórmula da profecia de Spearman-Brown)*
5. Confiabilidade composta.

Se você não conhece nada disso, aqui estão alguns recursos para você se atualizar:

- Consistência interna (Wikipédia).
- Alfa de Cronbach (Wikipédia).
- Uso do Coeficiente Alfa de Cronbach em Avaliações (Infoteca-e/Embrapa).
- Alfa de Cronbach (Blog Sonia Vieira).

## Os pacotes

Acrescentei este trecho de código R a fim de facilitar na instalação dos pacotes utilizados nesta análise.

```
# Verificar e instalar o(s) pacote(s) a utilizar na analise.
pacotes_analise <- c("tidyverse", "corrr", "psych", "lavaan")
# Necessarios apenas para RMarkdown.
pacotes_padrao <- c("installr", "rmarkdown", "tinytex", "prettydoc")
if (length(setdiff(c(pacotes_padrao, pacotes_analise), rownames(installed.packages())) > 0) {
  install.packages(setdiff(c(pacotes_padrao, pacotes_analise), rownames(installed.packages())))
  # RMarkdown precisa de Pandoc e MiKTeX instalados. https://miktex.org/2.9/setup.
  install.pandoc()
}
```

## Os dados

Para este notebook, usaremos dados sobre uma medida de personalidade Big 5 que está disponível gratuitamente nos Testes de Personalidade. Você pode fazer o download dos dados AQUI ou se preferir, à execução do código a seguir manipulará o download e salvará os dados em um objeto chamado `dados`:

```
# Setando codificacao para representar qualquer caractere universal padrao Unicode.
options(encoding="UTF-8")

# Setando o local da pasta de trabalho.
setwd("E:/Outras Analises R/Calculo da consistencia interna em R")

arq_temp <- tempfile()
download.file("http://personality-testing.info/_rawdata/BIG5.zip", arq_temp, mode="wb")
dados <- read.table(unz(arq_temp, "BIG5/data.csv"), header = TRUE, sep="\t")
unlink(arq_temp); rm(arq_temp)
```

No momento em que o post original foi escrito, este conjunto de dados continha dados para 19719 pessoas, começando com algumas informações demográficas e, em seguida, suas respostas em 50 itens: 10 para cada dimensão do Big 5. Isso é um pouco demais, então vamos reduzi-lo para trabalhar nos primeiros 500 participantes e nos itens de extroversão (E1 para E10):

```
dados <- dados[1:500, paste0("E", 1:10)]
str(dados)
```

```
## 'data.frame':    500 obs. of  10 variables:
## $ E1 : int  4 2 5 2 3 1 5 4 3 1 ...
## $ E2 : int  2 2 1 5 1 5 1 3 1 4 ...
## $ E3 : int  5 3 1 2 3 2 5 5 5 2 ...
## $ E4 : int  2 3 4 4 3 4 1 3 1 5 ...
## $ E5 : int  5 3 5 3 3 1 5 5 5 2 ...
## $ E6 : int  1 3 1 4 1 3 1 1 1 4 ...
## $ E7 : int  4 1 1 3 3 2 5 4 5 1 ...
## $ E8 : int  3 5 5 4 1 4 4 3 2 4 ...
## $ E9 : int  5 1 5 4 3 1 4 4 5 1 ...
## $ E10: int  1 5 1 5 5 5 1 3 3 5 ...
```

Aqui está uma lista dos itens de extroversão que as pessoas estão classificando de 1 = *Discordo* a 5 = *Concordo*:

- E1 Eu vivo em festa.
- E2 Eu não falo muito.
- E3 Eu me sinto confortável em torno das pessoas.
- E4 Eu me mantenho em segundo plano.
- E5 Eu inicio conversas.
- E6 Eu tenho pouco a falar.
- E7 Eu falo com muitas pessoas diferentes em festas.
- E8 Não gosto de chamar atenção para mim mesmo.
- E9 Eu não me importo de ser o centro das atenções.
- E10 Eu não converso com estranhos.

Você pode ver que há cinco itens que precisam ser marcados inversamente (E2, E4, E6, E8, E10). Como as classificações variam de 1 a 5, podemos fazer o seguinte:

```
dados[, paste0("E", c(2, 4, 6, 8, 10))] <- 6 - dados[, paste0("E", c(2, 4, 6, 8, 10))]
```

Agora temos um quadro de dados (data frame) de respostas com cada coluna sendo um item (pontuado na direção correta) e cada linha sendo um participante. Vamos iniciar!

## 1. Correlação média entre itens

A correlação entre itens é qualquer lugar fácil para começar. Para calcular essa estatística, precisamos das correlações entre todos os itens e depois calculá-las. Vamos usar o pacote *corr* de *Simon Jackson* (*Obrigado! Simon*) para obter estas correlações da seguinte forma (sem viés aqui!):

```
library(tidyverse)
library(corr)
```

```
dados %>% correlate()
```

```
## # A tibble: 10 x 11
##   rowname    E1    E2    E3    E4    E5    E6    E7    E8    E9
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 E1      NA    0.453 0.500 0.524 0.538 0.366 0.636 0.450 0.528
## 2 E2      0.453 NA    0.479 0.555 0.592 0.569 0.473 0.379 0.396
```

```
## 3 E3      0.500 0.479 NA      0.493 0.616 0.330 0.568 0.418 0.475
## 4 E4      0.524 0.555 0.493 NA      0.512 0.471 0.500 0.451 0.463
## 5 E5      0.538 0.592 0.616 0.512 NA      0.500 0.621 0.385 0.485
## 6 E6      0.366 0.569 0.330 0.471 0.500 NA      0.373 0.331 0.328
## 7 E7      0.636 0.473 0.568 0.500 0.621 0.373 NA      0.403 0.528
## 8 E8      0.450 0.379 0.418 0.451 0.385 0.331 0.403 NA      0.598
## 9 E9      0.528 0.396 0.475 0.463 0.485 0.328 0.528 0.598 NA
## 10 E10     0.491 0.449 0.500 0.523 0.553 0.414 0.518 0.429 0.457
## # ... with 1 more variable: E10 <dbl>
```

Como a diagonal já está definida NA, podemos obter a correlação média de cada item com todos os outros calculando as médias de cada coluna (excluindo a coluna rowname):

```
inter_item <- dados %>% correlate() %>% select(-rowname) %>% colMeans(na.rm = TRUE)
inter_item

##      E1      E2      E3      E4      E5      E6      E7
## 0.4983678 0.4827948 0.4866458 0.4991764 0.5334524 0.4090418 0.5133091
##      E8      E9      E10
## 0.4270190 0.4731896 0.4814496
```

Além disso, note que `select()` vem do pacote **dplyr**, que é importado quando você usa o **corrr**. Podemos ver isso E5 e E7 estamos mais fortemente correlacionados com os outros itens em média do que E8. No entanto, a maioria dos itens se correlaciona com os outros em um intervalo razoavelmente restrito em torno de 0,4 a 0,5. Para obter a média geral da correlação entre itens, calculamos a `mean()` (média) dos valores:

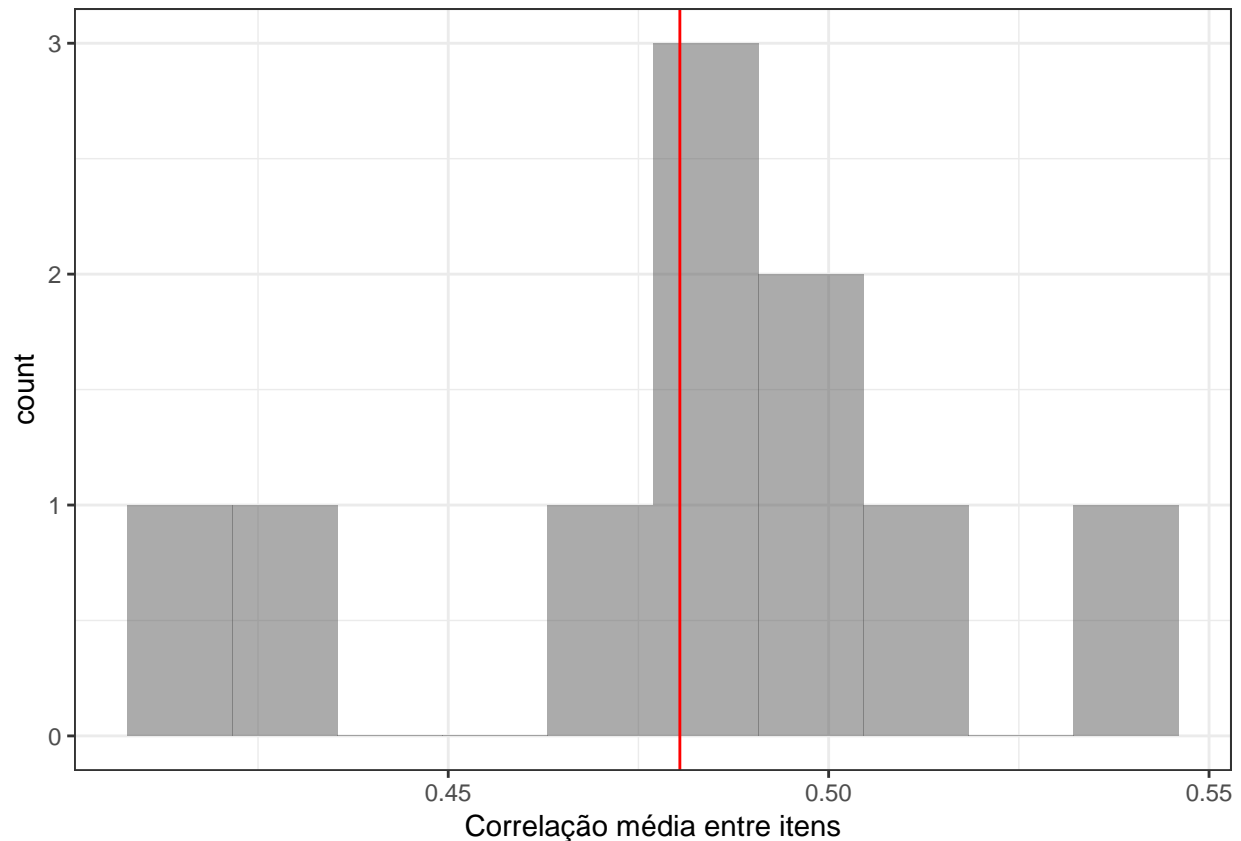
```
mean(inter_item)
```

```
## [1] 0.4804446
```

No entanto, com esses valores, podemos explorar uma série de atributos sobre os relacionamentos entre os itens. Por exemplo, podemos visualizá-los em um histograma e destacar a média da seguinte forma:

```
library(ggplot2)

data.frame(inter_item) %>%
  ggplot(aes(x = inter_item)) +
  geom_histogram(bins = 10, alpha = .5) +
  geom_vline(xintercept = mean(inter_item), color = "red") +
  xlab("Correlação média entre itens") +
  theme_bw()
```



## 2. Correlação média total do item

Podemos investigar a correlação média item-total de maneira semelhante às correlações entre itens. A primeira coisa que precisamos fazer é calcular a pontuação total. Digamos que a pontuação de uma pessoa seja a média de suas respostas para todos os dez itens:

```
dados$score <- rowMeans(dados)
head(dados)
```

```
##   E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 score
## 1  4  4  5  4  5  5  4  3  5   5   4.4
## 2  2  4  3  3  3  3  1  1  1   1   2.2
## 3  5  5  1  2  5  5  1  1  5   5   3.5
## 4  2  1  2  2  3  2  3  2  4   1   2.2
## 5  3  5  3  3  3  5  3  5  3   1   3.4
## 6  1  1  2  2  1  3  2  2  1   1   1.6
```

Agora, vamos `correlate()` novamente, mas desta vez `focus()` sobre as correlações do score com os itens:

```
item_total <- dados %>% correlate() %>% focus(score)
item_total
```

```
## # A tibble: 10 x 2
##   rowname score
##   <chr>    <dbl>
## 1 E1      0.752
## 2 E2      0.730
## 3 E3      0.735
```

```
## 4 E4      0.749
## 5 E5      0.795
## 6 E6      0.637
## 7 E7      0.777
## 8 E8      0.664
## 9 E9      0.727
## 10 E10    0.731
```

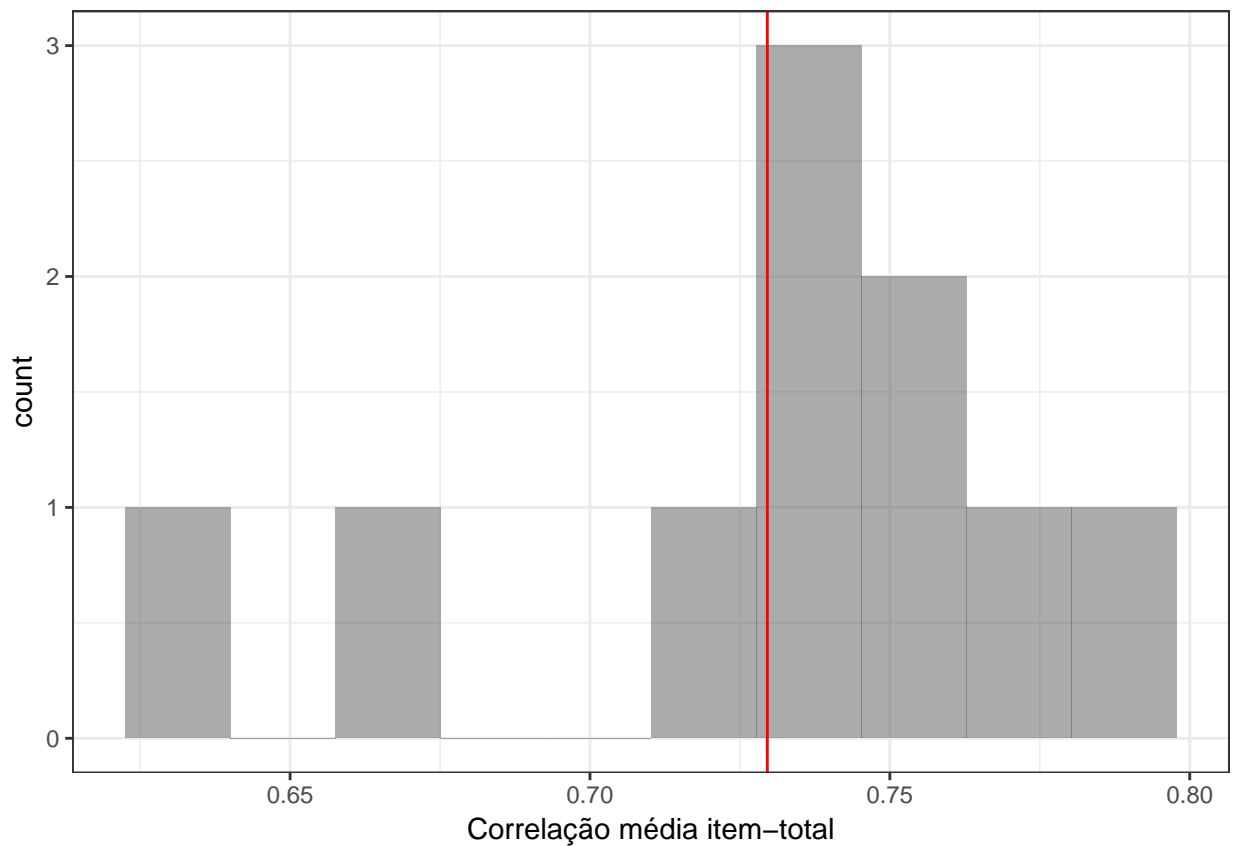
Mais uma vez, podemos calcular sua média como:

```
mean(item_total$score)
```

```
## [1] 0.7295695
```

E podemos traçar os resultados:

```
item_total %>%
  ggplot(aes(x = score)) +
  geom_histogram(bins = 10, alpha = .5) +
  geom_vline(xintercept = mean(item_total$score), color = "red") +
  xlab("Correlação média item-total") +
  theme_bw()
```



### 3. Alfa de Cronbach

O Alfa de Cronbach é uma das medidas de consistência interna mais amplamente relacionadas. Embora seja possível implementar a matemática por trás disso, sou preguiçoso e gosto de usar a função `alpha()` do pacote **psych**. Essa função usa um **data frame** (quadro de dados) ou uma matriz de dados na estrutura

que estamos usando: cada coluna é um item de teste/questionário, cada linha é uma pessoa. Vamos testá-lo abaixo. Note que `alpha()` também é uma função do pacote **ggplot2**, e isso cria um conflito. Para especificar que queremos `alpha()` do pacote psicológico, usaremos o nome do pacote **psych** implicitamente à função como segue `psych::alpha()`.

```
library(psych)

dados$score <- NULL # Excluir a coluna de pontuacao que fizemos anteriormente.
psych::alpha(dados, check.keys=TRUE)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = dados, check.keys = TRUE)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##         0.9         0.9         0.9         0.48 9.2 0.0065    3 0.98    0.49
##
##   lower alpha upper      95% confidence boundaries
## 0.89 0.9 0.91
##
## Reliability if an item is dropped:
##   raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## E1      0.89      0.89      0.89      0.48 8.2  0.0073 0.0064 0.48
## E2      0.89      0.89      0.89      0.48 8.3  0.0072 0.0065 0.49
## E3      0.89      0.89      0.89      0.48 8.3  0.0073 0.0062 0.48
## E4      0.89      0.89      0.89      0.48 8.2  0.0073 0.0074 0.48
## E5      0.89      0.89      0.89      0.47 7.9  0.0075 0.0056 0.47
## E6      0.90      0.90      0.90      0.50 8.9  0.0068 0.0044 0.50
## E7      0.89      0.89      0.89      0.47 8.1  0.0075 0.0056 0.48
## E8      0.90      0.90      0.90      0.49 8.8  0.0069 0.0056 0.50
## E9      0.89      0.89      0.89      0.48 8.4  0.0071 0.0063 0.49
## E10     0.89      0.89      0.89      0.48 8.3  0.0072 0.0073 0.48
##
## Item statistics
##      n raw.r std.r r.cor r.drop mean  sd
## E1  500 0.75 0.75 0.72 0.68 2.6 1.3
## E2  500 0.73 0.73 0.70 0.66 3.2 1.3
## E3  500 0.74 0.74 0.70 0.67 3.3 1.3
## E4  500 0.75 0.75 0.72 0.68 2.8 1.3
## E5  500 0.79 0.80 0.78 0.74 3.3 1.3
## E6  500 0.64 0.64 0.59 0.55 3.6 1.3
## E7  500 0.78 0.77 0.75 0.70 2.8 1.5
## E8  500 0.66 0.66 0.61 0.58 2.7 1.3
## E9  500 0.73 0.72 0.68 0.64 3.1 1.5
## E10 500 0.73 0.73 0.69 0.66 2.3 1.3
##
## Non missing response frequency for each item
##      1 2 3 4 5 miss
## E1 0.28 0.23 0.23 0.16 0.10 0
## E2 0.12 0.21 0.22 0.21 0.23 0
## E3 0.09 0.20 0.25 0.22 0.24 0
## E4 0.19 0.26 0.23 0.21 0.12 0
## E5 0.11 0.21 0.20 0.23 0.25 0
## E6 0.09 0.15 0.17 0.28 0.30 0
## E7 0.28 0.20 0.17 0.16 0.20 0
```

```
## E8  0.23 0.27 0.19 0.19 0.12    0
## E9  0.19 0.21 0.17 0.19 0.25    0
## E10 0.36 0.27 0.12 0.15 0.09    0
```

Esta função geralmente fornece um intervalo de saída, o que nos interessa de `std.alpha` é o “alpha padronizado baseado nas correlações”. Observe também que obtemos “a correlação intertemporal média”, `average_re` várias versões da “correlação de cada item com a pontuação total”, como `raw.r`, cujos valores correspondem aos nossos cálculos anteriores. Se você quiser acessar o próprio valor alfa, faça o seguinte:

```
psych::alpha(dados, check.keys=TRUE)$total$std.alpha
```

```
## [1] 0.9024126
```

#### 4. Confiabilidade dividida (*ajustada usando a fórmula da profecia de Spearman-Brown*)

Há momentos em que não podemos calcular a consistência interna usando respostas de itens. Por exemplo, costumo trabalhar com uma variável de tomada de decisão chamada imprudência. Essa variável é calculada depois que as pessoas respondem a perguntas (por exemplo, “Qual é o rio mais longo na Ásia”) e em seguida, decidem se devem ou não apostar que a resposta está correta. A imprudência é calculada como a proporção de respostas incorretas nas quais uma pessoa aposta.

Se você pensar sobre isso, não é possível calcular a consistência interna para essa variável usando qualquer uma das medidas acima. A razão para isso é que os itens que contribuem para as pontuações de imprudência de duas pessoas podem ser completamente diferentes. Uma pessoa poderia dar respostas incorretas nas questões de 1 a 5 (assim essas questões são calculadas), enquanto outra pessoa pode responder incorretamente às questões de 6 a 10. Assim, calcular imprudência para muitos indivíduos não é tão simples quanto somar os itens. Em vez disso, precisamos de um pool de itens para extrair diferentes combinações de perguntas para cada pessoa.

Para superar esse tipo de problema, um método apropriado para calcular a consistência interna é usar uma confiabilidade dividida. Isso implica dividir seus itens de teste pela metade (por exemplo, em ímpar e par) e calcular sua variável para cada pessoa com cada metade. Por exemplo, normalmente calculo imprudência para cada participante de itens ímpares e depois de itens pares. Essas pontuações são então correlacionadas e ajustadas usando a fórmula profecia/predição de Spearman-Brown (por exemplo, sobre esse assunto, veja algumas publicações científicas de Simon aqui ou aqui).

Semelhante ao alfa de Cronbach, um valor mais próximo de um e maior que zero indica maior consistência interna.

Ainda podemos calcular a confiabilidade da metade para variáveis que não têm esse problema! Então, vamos fazer isso com os nossos dados de extroversão da seguinte forma:

```
# Calculando a pontuacao total.
score_e <- rowMeans(dados[, c(TRUE, FALSE)]) # Com itens pares.
score_o <- rowMeans(dados[, c(FALSE, TRUE)]) # Com itens impares.

# Correlação de pontuações de itens pares e impares.
r <- cor(score_e, score_o)
r
```

```
## [1] 0.7681034
```

```
# Ajuste com a formula da profecia de Spearman-Brown.
(2 * r) / (1 + r)
```

```
## [1] 0.8688445
```

Assim, neste caso, a abordagem de confiabilidade da metade dividida produz uma estimativa de consistência interna de 0,87.

## 5. Confiabilidade composta

O método final para calcular a consistência interna que cobriremos é a confiabilidade composta. Sempre que possível, minha preferência pessoal é usar essa abordagem. Embora não seja perfeito, ele cuida de muitas suposições inadequadas que medem como o Alfa de Cronbach. Se as especificidades lhe interessarem, sugiro ler este post.

A confiabilidade composta é baseada nas cargas fatoriais em uma análise fatorial confirmatória (AFC). No caso de uma escala unidimensional, definimos um CFA de um fator e em seguida, usamos as cargas fatoriais para calcular nossa estimativa de consistência interna.

Não vou entrar em detalhes, mas podemos interpretar uma pontuação de confiabilidade composta similarmente a qualquer outra métrica aqui abordada (mais próximo de uma indica melhor consistência interna). Nós vamos caber nosso modelo CFA usando o pacote lavaan da seguinte forma:

```
# Definir o modelo.
items <- paste(names(dados), collapse = "+")
model <- paste("extraversion", items, sep = "=~")
model

## [1] "extraversion=~E1+E2+E3+E4+E5+E6+E7+E8+E9+E10"

# Ajusta o modelo.
fit <- lavaan::cfa(model, data = dados)
```

Existem várias maneiras de obter a confiabilidade composta deste modelo. Vamos extrair as cargas fatoriais padronizadas e trabalhar com elas:

```
library(lavaan)

sl <- lavaan::standardizedSolution(fit)
sl <- sl$est.std[sl$op == "=~"]
names(sl) <- names(dados)
sl # Estas sao as cargas fatoriais padronizadas para cada item.

##          E1          E2          E3          E4          E5          E6          E7
## 0.7266721 0.6903172 0.7154705 0.7118762 0.7841427 0.5791526 0.7589250
##          E8          E9          E10
## 0.5962855 0.6726232 0.6940120
```

Em seguida, obtemos a confiabilidade composta por meio do seguinte:

```
# Compute a variancia residual de cada item.
re <- 1 - sl^2

# Compute a confiabilidade composta.
sum(sl)^2 / (sum(sl)^2 + sum(re))

## [1] 0.9029523
```

Então você tem isso. A confiabilidade composta para o fator de extroversão é de **0,90**. Um aspecto atraente da confiabilidade composta é que podemos calcular isso para vários fatores no mesmo modelo. Por exemplo, digamos que incluímos todos os itens de personalidade em um CFA com cinco fatores, poderíamos fazer os cálculos acima separadamente para cada fator e obter sua confiabilidade composta.

Apenas para finalizar, vou mencionar que você pode usar as cargas fatoriais padronizadas para visualizar mais informações, como fizemos anteriormente com as correlações. Essa parte fica para você praticar!

Obrigado por ler e espero que esta análise em versão R Notebook tenha sido útil para você. Se precisar entrar em contato comigo, envie e-mail para [erivandosena@gmail.com](mailto:erivandosena@gmail.com).



Se você quiser o código que produziu esta publicação, confira no meu repositório do GitHub.

## REFERÊNCIAS

BLOGR. **Five ways to calculate internal consistency**. Disponível em: <<https://drsimonj.svbtile.com/how-to-calculate-internal-consistency>>. Acesso em 27 mai. 2019.