CFA_GPTsupport

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

```
library(quarto)
```

```
Warning: Paket 'quarto' wurde unter R Version 4.4.1 erstellt
```

```
#tinytex::install_tinytex()
```

Running Code

Tip 1: Getting Help from ChatGPT with Prompt Engineering

You can use the following prompts to prepare GPT for working with CFA. Use GPT4 if you can. Still all GTPs have a lot more trouble with R than with other programming languages. Be extra critical when taking reading the information it gives you. Keep your instructions as small and precise as possible. You may copy/paste the following prompt text into your preferred LLM:

Comprehensive Prompt for Configural Frequency Analysis (CFA) Using the `confreq` Package in R:

"I want to train you to consistently give correct answers about Configural Frequency Analysis (CFA) and the R package `confreq`. It is especially important that you do not mix up Configural Frequency Analysis (CFA) with Confirmatory Factor Analysis, which is also often referred to as CFA. For information about the R package `confreq`, you should only ever use information from this CRAN page: https://rdrr.io/cran/confreq/.

When providing information, please follow these instructions:

1. Always assume that 'CFA' refers to Configural Frequency Analysis unless explicitly stated otherwise. Tailor your responses to a beginner's expertise level.
2. There are no specific key points to emphasize. However, if any part of your response has a small likelihood of being correct, clearly state that you are not sure, rather than making assumptions or providing potentially incorrect information.
3. Always link directly to the official CRAN documentation for the `confreq` package https://rdrr.io/cran/confreq/ whenever relevant.
4. Use the following code as a typical example of performing Configural Frequency Analysis with data saved in a CSV file on the computer:

library(confreq)

cfa <- read.table("BeispielCFA.csv", sep=";", header=TRUE, quote=" ")

Data to patterned frequencies

cfa_new <- dat2fre(fre2dat(cfa))

(cfa_new)

Start the CFA function, results stored in "resd1"

resd1 <- CFA(cfa_new, form="~ Seizures + Birth + Intelligence")

Tabulated printout

summary(resd1, sorton="pat.")

5. Rely on the most recent stable version of the `confreq` package unless otherwise directed.

Please provide clear, step-by-step explanations following these instructions to ensure accurate and reliable guidance on performing Configural Frequency Analysis using the `confreq` package in R."

Tip 2: Use custom GPTs

Open GPT3 or GPT4/GPT4o. Go to "discover GPTs" on the left side bar. Type into the search bar: "R and R Studio". Chose R and R Studio by Widenex or R and R Studio Tutor by Jose Calvo. Klick to open and paste the prompts from Tip 1.

Tip 3: Alternative AIs

GPT is currently one of the best AIs for R. But it still a lot more trouble with R than with other programming languages such as Java. But often it is useful. One problem is that you often don't know where GPT got the information from. We tried to avoid this by giving GPT the CRAN link to confreq. Still it might use other sources too. One way to avoid this is by using **perplexity.ai**. This gives you the links to all the information it used. This way you can see where the AI might have gotten certain code from (for example from github, stack overflow or reddit). This can be very helpful in determining if the information can be trusted and might help you fix errors.

Example

```
library(confreq)
```

```
Warning: Paket 'confreq' wurde unter R Version 4.4.1 erstellt
```

```
Lade nötiges Paket: gmp
```

```
Warning: Paket 'gmp' wurde unter R Version 4.4.1 erstellt
```

```
Attache Paket: 'gmp'
```

```
Die folgenden Objekte sind maskiert von 'package:base':

    %*%, apply, crossprod, matrix, tcrossprod
```

```
Lade nötiges Paket: grid
```

```
Lade nötiges Paket: vcd
```

```
Warning: Paket 'vcd' wurde unter R Version 4.4.1 erstellt
```

```
cfa <- read.table("BeispielCFA.csv", sep=";", header=TRUE, quote="\"")
```

```r
#data to patterned frequencies
cfa_new <- dat2fre(fre2dat(cfa))
```

```
 Number of categories for each variable
 estimated from data are:
 Seizures Birth Intelligence
 2 2 2
 --> 8 different configurations
```

```r
cfa_new
```

```
  Seizures Birth Intelligence freq
1        1     1            1   27
2        1     1            2    4
3        1     2            1   12
4        1     2            2    1
5        2     1            1    4
6        2     1            2    1
7        2     2            1    2
8        2     2            2    7
```

```r
#to start the function CFA the results are written into the object "resd1"
resd1 <- CFA(cfa_new, form="~ Seizures + Birth + Intelligence")
```

```r
#tabulated printout
summary(resd1, sorton="pat.")
```

```
function Call:
 -------------
 Formula: ~ Seizures + Birth + Intelligence
 Variables: Seizures Birth Intelligence
 Categories: 2 2 2

 results of global tests:
 ------------------------
 pearson Chi-square test:
       Chi df         pChi alpha
1 35.43167  4 3.787218e-07  0.05
```

```
 likelihood ratio test:
       Chi df         pChi alpha
1 21.62194  4 0.0002383096  0.05

 Information Criteria:
     loglik      AIC      BIC
1 -24.02564 56.05129 56.36905

 results of local tests:
 -----------------------
 Type (+) / Antitype (-) based on: z.pChi ;
 with Bonferroni adjusted alpha: 0.00625


   pat. obs.   exp. Type df  z.Chi z.pChi
1 1 1 1   27 21.189    .  1  1.262  0.103
2 1 1 2    4  6.121    .  1 -0.857  0.196
3 1 2 1   12 12.949    .  1 -0.264  0.396
4 1 2 2    1  3.741    .  1 -1.417  0.078
5 2 1 1    4  6.742    .  1 -1.056  0.145
6 2 1 2    1  1.948    .  1 -0.679  0.249
7 2 2 1    2  4.120    .  1 -1.044  0.148
8 2 2 2    7  1.190    +  1  5.325  0.000
```