Package 'PROscorer'

May 16, 2017

Type Package

Title Functions to Score Commonly-Used Patient-Reported Outcome (PRO) Measures and Other Psychometric Instruments

Version 0.0.1

Description An extensible repository of accurate, up-to-date functions to score commonly used patient-reported outcome (PRO), quality of life (QOL), and other psychometric and psychological measures. 'PROscorer', together with the 'PROscorerTools' package, is a system to facilitate the incorporation of PRO measures into research studies and clinical settings in a scientifically rigorous and reproducible manner. These packages and their vignettes are intended to help establish and promote ``best practices'' to improve the planning, scoring, and reporting of PRO-like measures in research. The 'PROscorer' ``Instrument Descriptions'' vignette contains descriptions of each instrument scored by 'PROscorer', complete with references. These instrument descriptions are suitable for inclusion in formal study protocol documents, grant proposals, and manuscript Method sections. Each 'PROscorer' function is composed of helper functions from the 'PROscorerTools' package, and users are encouraged to contribute new functions to 'PROscorer'. More scoring functions are currently in development and will be added in future updates.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL http://github.com/raybaser/PROscorer

BugReports http://github.com/raybaser/PROscorer/issues

RoxygenNote 6.0.1

Imports PROscorerTools

Suggests knitr, rmarkdown, testthat, covr

VignetteBuilder knitr

NeedsCompilation no

Author Ray Baser [aut, cre]

Maintainer Ray Baser <ray.stats@gmail.com>

Repository CRAN

Date/Publication 2017-05-16 04:37:41 UTC

R topics documented:

`fsfi` *Score the Female Sexual Function Index (FSFI)*

Description

Scores the Female Sexual Function Index (FSFI)

Usage

```
fsfi(df, iprefix = "fsfi", keepNvalid = FALSE)
```

Arguments

`df`	A data frame containing responses to the 19 FSFI items, and possibly other variables.
`iprefix`	Item number prefix. Quote the letter(s) preceding the FSFI item numbers as they are named in your data frame. If this argument is omitted, the function will assume that your items are named `"fsfi1"`, `"fsfi2"`, etc.
`keepNvalid`	Logical, whether to return variables containing the number of valid, non-missing items on each scale for each respondent should be returned in the data frame with the scale scores. The default is `FALSE`. Set to `TRUE` to return these variables, which will be named `"scalename_N"` (e.g., `fsfi_pain_N`). Most users should omit this argument entirely. This argument might be removed from future versions of the package, so please let me know if you think this argument useful and would rather it remain a part of the function.

Details

This function returns the 6 subscale scores and the FSFI Total score (Rosen et al., 2000), as well as an indicator variable flagging respondents with FSFI Total scores suggestive of clinically significant levels of sexual dysfunction (i.e., `fsfi_tot <= 26.55`; Wiegel et al., 2005).

The FSFI is intended to measure the sexual function of recently sexually active women (Rosen et al., 2000), and strong evidence suggests it may not be a valid measure of sexual function in women with little or no recent sexual activity (e.g., see Baser et al., 2012).

As such the fsfi function also returns two variables (fsfi_nzero15 and fsfi_sexactive01) that can be used to evaluate whether respondents have been sufficiently sexually active for the FSFI to be a valid assessment of their sexual function. These variables are based on the fact that 15 of the 19 FSFI items have a response option of "no sexual activity" or "did not attempt intercourse", which corresponds to an item score of 0. Specifically, the fsfi_nzero15 variable contains the number of items with responses of 0 or NA (out of those 15 items that have a response option indicating "no sexual activity"). Missing responses (i.e., NA) are included in this count because respondents with no relevant sexual activity often skip these items. The fsfi_sexactive01 variable is a rough indicator that a respondent was sufficiently sexually active for the FSFI to be a valid assessment of their sexual function. It is a dummy variable that is 1 when fsfi_nzero15 <= 7 (i.e., when the respondent said "no sexual activity" to 7 or fewer of the 15 items with that option), and 0 otherwise. See Baser et al. (2012) for more details on how this cutoff was chosen.

Value

A data frame with the following variables is returned:

- **fsfi_des** - FSFI Desire subscale (range 1.2 - 6)
- **fsfi_arous** - FSFI Arousal subscale (range 0 - 6)
- **fsfi_lub** - FSFI Lubrication subscale (range 0 - 6)
- **fsfi_org** - FSFI Orgasm subscale (range 0 - 6)
- **fsfi_sat** - FSFI Satisfaction subscale (range 0.8 - 6)
- **fsfi_pain** - FSFI Pain subscale (range 0 - 6)
- **fsfi_tot** - FSFI Total score (range 2 - 36)
- **fsfi_dys01** - Indicator of FSFI sexual dysfunction (i.e., of fsfi_tot <= 26.55); 0 = No Dysfunction, 1 = Dysfunction
- **fsfi_nzero15** - There are 15 FSFI items that have a response option of 0 ("No sexual activity"). This is the number of those items with responses of 0 or NA (See Details).
- **fsfi_sexactive01** - For the FSFI scores to be valid estimates of sexual functioning, respondents need to have been sexually active during the 4 week recall period. This variable indicates whether their sexual activity levels were high enough for their FSFI scores to be valid. Specifically, it is an indicator that fsfi_nzero15 <= 7 (See Details).

Optionally, the data frame can additionally have variables containing the number of valid item responses on each scale for each respondent (if keepNvalid = TRUE, but this option might be removed in future package updates).

How Missing Data is Handled

The FSFI authors do not indicate how to handle missing item data when calculating the FSFI scores. This is unfortunate because women frequently skip items they feel are not relevant to them (e.g., the items asking about satisfaction with "your partner" are often skipped by non-partnered women), leading to an unexpectedly large number of missing subscale and FSFI total scores. To minimize excessive missing values for the FSFI subscale and Total scores, the fsfi function handles missing items similarly to the scoring methods for many other PROs. Specifically, the fsfi function will calculate the 6 subscale scores as long as at least half of the items on the given subscale have valid, non-missing item responses. More concretely, each subscale must have at least 2 non-missing

responses, except for Desire, which has only 2 items and requires only 1 non-missing response. The `fsfi` function will calculate the FSFI Total Score for a respondent as long as it was able to calculate at least 5 out of the 6 subscale scores. Scores calculated in the presence of missing items are pro-rated so that their theoretical minimum and maximum values are identical those from scores calculated from complete data.

These methods of handling missing item responses were chosen to balance the reality that respondents often skip some items with the need to maintain the validity of the scores. However, I know of no directly applicable empirical study that supports these choices, and I encourage more research into how missing responses affect the psychometrics of this and other instruments.

Note

The six FSFI subscale scores are scaled to have a maximum score of 6.0. The subscale scores are summed to calculate the FSFI Total score, which has a maximum score of 36. Because 4 items have no response option scored `0` (2 items from Desire subscale and 2 from Satisfaction subscale), the minimum possible score for the Desire subscale, the Satisfaction subscale, and the FSFI Total score is greater than zero.

References

Rosen, R, Brown, C, Heiman, J, Leiblum, S, Meston, C, Shabsigh, R, et al. (2000). The Female Sexual Function Index (FSFI): a multidimensional self-report instrument for the assessment of female sexual function. *Journal of Sex & Marital Therapy, 26*(2), 191-208.

Wiegel, M, Meston, C, & Rosen, R. (2005). The Female Sexual Function Index (FSFI): Cross-Validation and Development of Clinical Cutoff Scores. *Journal of Sex & Marital Therapy, 31*(1), 1-20.

Baser, RE, Li, Y, & Carter, J. (2012). Psychometric validation of the female sexual function index (FSFI) in cancer survivors. *Cancer, 118*(18), 4606-4618.

Examples

```
# Creating data frame of fake FSFI responses
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 19, values = 0:5,
                                    prefix = 'f')
dat1 <- PROscorerTools::makeFakeData(n = 10, nitems = 4, values = 1:5)
names(dat1) <- c('f1', 'f2', 'f15', 'f16')
dat[c(1, 2, 15, 16)] <- dat1
# Scoring the fake FSFI responses
fsfi(dat, 'f')
```

`narcc` *Score the Cognitive Causation (CC) and Negative Affect in Risk (NAR) scales*

Description

Scores the Cognitive Causation (CC) and Negative Affect in Risk (NAR) scales, two scales measuring intuitive elements of cancer risk perception (see references).

Usage

```
narcc(df, items = NULL, whichScale, minmax = c(0, 3), okmiss = 0.5,
  keepNvalid = FALSE)
```

Arguments

df	A data frame containing responses to the CC and/or NAR items, and possibly other variables.
items	(optional) A character vector with the CC or NAR item names, or a numeric vector indicating the column numbers of the CC or NAR items in df. If items is omitted, then narcc will assume that df contains **ONLY** the items to be scored (either CC or NAR items) and no non-scored variables.
whichScale	(required) Either "CC" or "NAR", the scale you wish to score.
minmax	A vector of 2 integers with format c(itemMin, itemMax), indicating the minimum and maximum possible item responses. The default value is c(0, 3), and assumes that the item responses are coded from 0 to 3. If, instead, your item responses are coded from 1 to 4, then enter c(1, 4) for this argument.
okmiss	(optional) The maximum proportion of items on whichScale that a respondent is allowed to have missing and still have their non-missing items scored (and prorated). If the proportion of missing items for a respondent is greater than okmiss, then the respondent will be assigned a value of NA for their scale score. The default value is 0.50, and this generally should not be changed.
keepNvalid	(optional) Logical value indicating whether a variable containing the number of valid, non-missing items for each respondent should be returned in a data frame with the scale score. The default is FALSE. Set to TRUE to return this variable, which will be named "whichScale_N" (with whatever name you gave to the whichScale argument). Most users should omit this argument entirely. This argument might be removed from future versions of the package, so please let me know if you think this argument useful and would rather it remain a part of the function.

Details

The CC scale originally contained 10 items (Hay et al., 2014). Later, evidence that 3 of the items might be measurement non-invariant across important subgroups led to the recommendation to omit these 3 items and score a 7-item version of the CC scale (Baser et al., 2016). When whichScale = "CC" the narcc function will accept and score either 7 or 10 CC items, although the 7-item version is recommended. The NAR scale has 6 items, and the narcc function will accept only 6 NAR items when whichScale = "NAR".

If you want to score both the CC and NAR scales, then you need to run the narcc function twice, once for CC and again for NAR.

Value

A data frame containing a variable containing the scored scale, named either "CC" or "NAR". Scores are scales to have range 0 to 100.

Optionally, the data frame can additionally have a variable containing the number of valid item responses on the scale for each respondent (if `keepNvalid = TRUE`, but this option might be removed in future package updates).

Note

The `narcc` function assumes that your item data are numerically coded from 0 to 3 (i.e., with 0 = "Strongly Disagree" and 3 = "Strongly Agree"). However, your item data might instead be coded from 1 to 4. If this is the case, you MUST let the `narcc` function know this by using the `minmax` argument, specifically, `minmax = c(1, 4)`.

References

Hay, JL, Baser, R, Weinstein, ND, Li, Y, Primavera, L, & Kemeny, MM. (2014). Examining intuitive risk perceptions for cancer in diverse populations. *Health, Risk & Society, 16*(3), 227-242.

Baser, RE, Li, Y, Brennessel, D, Kemeny, MM, & Hay, JL. (2016). Measurement Invariance of Intuitive Cancer Risk Perceptions Across Diverse Populations: The Cognitive Causation and Negative Affect in Risk Scales. *Journal of Health Psychology*; In Submission.

Examples

```
# Make fake data for the example
nardat <- PROscorerTools::makeFakeData(nitems = 6, values = 0:3,
                                       propmiss = 0.40, prefix = "nar")
ccdat <- PROscorerTools::makeFakeData(nitems = 7, values = 0:3,
                                      propmiss = 0.40, prefix = "cc",
                                      id = TRUE)

# The nardat data frame contains ONLY NAR items, so can omit "items" argument
narcc(nardat, whichScale = "NAR")

# The ccdat data frame contains an "ID" variable, so need to use "items" arg
names(ccdat)

# The "items" argument can be either:
#     (1) the numeric vector indexing the location of the items in df, or
#     (2) a character vector of the item names
narcc(ccdat, items = 2:8, whichScale = "CC")

cc_names <- c("cc1", "cc2", "cc3", "cc4", "cc5", "cc6", "cc7")
narcc(ccdat, items = cc_names, whichScale = "CC")
```

`PROscorer` *PROscorer*

Description

An open-source repository of functions to score specific Patient-Reported Outcome (PRO), Quality of Life (QoL), and other psychometric measures commonly used in research.

Details

The **PROscorer** package is an extensible repository of functions to score specific PRO, QoL, and other psychometric measures and questionnaire-based instruments commonly used in research. It is intended to promote best practices for scoring PRO-like instruments, to standardize scoring procedures for PRO measures across studies, and to improve the reproducibility of research with PRO-like instruments by providing accurate, up-to-date, and well-documented PRO scoring functions that can easily be integrated into scientifically reproducible workflows.

Additionally, **PROscorer** is accompanied by a package vignette that contains detailed descriptions of each instrument scored by **PROscorer**, complete with references. Importantly, the instrument summaries are written according to a set of standards that ensure they meet "best practice" guidelines for descriptions of PRO-like measures in formal research protocols and in reports of research results featuring such measures. This means that, with little or no editing, a given instrument summary can be copied and pasted directly into protocols, grant proposals, and manuscripts. In addition to improving the measure descriptions in research documents, this saves the study investigators considerable time and effort.

The Problem

The scientific rigor and reproducibility of research involving PRO, QoL, and similar measures is lagging behind other research areas. Three major reasons for these shortcomings are (1) measurement error introduced by faulty scoring procedures, (2) inconsistent application of scoring instructions across different studies using the same PRO measures, and (3) inadequate, incomplete, and/or inaccurate descriptions of PRO-like measures in research protocols and in published results of studies that incorporate such measures.

Scoring procedures represent a major source of error in research studies that rely upon PRO and similar measures. These errors typically go unnoticed, hidden, and/or ignored, eroding the scientific integrity of the research and hindering progress in the numerous scientific fields that conduct studies that use these measures.

Similarly, inconsistent application of PRO scoring procedures and variation in scoring across studies makes study results less likely to replicate and slows the accumulation of reliable scientific data from the PRO measure.

Inadequate, incomplete, and/or inaccurate descriptions of PRO-like measures in research documents can cause confusion and introduce errors, oversights, and other mistakes at multiple stages in the research process.

The Proposed Solution

The **PROscorer** package provides the framework for addressing these problems with research involving PRO-like measures. The lofty goal of the **PROscorer** package is to eliminate these serious deficiencies in PRO-based research by serving as the gold-standard open-source repository of scoring syntax and instrument descriptions for PRO-like measures commonly used in research and clinical settings.

The features of the **PROscorer** package and supporting infrastructure were carefully planned with this ambitious goal in mind.

- **PROscorer** serves as the repository of scoring functions for specific, commonly-used PRO measures (e.g., the EORTC QLQ-C30).

- All scoring functions in **PROscorer** are written using simpler functions from a separate package, **PROscorerTools**. The **PROscorerTools** package provides the well-tested, reusable infrastructure for the **PROscorer** functions. This makes it easy to write new scoring functions to add to **PROscorer**, and decreases the chance of errors and other bugs.
- Advanced users can use **PROscorerTools** to write new scoring functions for their favorite PRO-like measures and submit them for inclusion in future **PROscorer** updates. As of this initial release, the system for writing and submitting new functions is immature. A vignette will be included in future updates with a guide for writing new functions and submitting them for review on GitHub.

Functions to score additional PRO measures are currently under development and will be included in future releases.

Please Provide Feedback

The **PROscorer** and **PROscorerTools** packages are still in their initial versions. As such, some details and other conventions are still being hammered out, particularly in **PROscorerTools** (e.g., function naming conventions, argument-checking functions, etc.). However, any changes to the **PROscorer** functions are expected to be internal and have little or no impact on end-users.

I put a lot of thought into the **PROscorer** and **PROscorerTools** packages, and I have tested them as an end-user as well as developer. However, I cannot anticipate the needs of all users, and I would like your feedback on your experience using the package(s). Please let me know if you found **PROscorer** and/or **PROscorerTools** helpful. And please tell me how I can improve their usability, and definitely report any bugs or other unexpected behaviors you encounter. Make feature requests and bug reports here: https://github.com/raybaser/PROscorerTools/issues

---

qlq_c30 *Score the EORTC QLQ-C30 Quality of Life Questionnaire*

Description

Scores the European Organization for Research and Treatment of Cancer (EORTC) QLQ-C30 Quality of Life Questionnaire (version 3.0).

Usage

```
qlq_c30(df, iprefix = NULL, items = NULL, keepNvalid = FALSE)
```

Arguments

`df`	A data frame containing responses to the 30 QLQ-C30 items, and possibly other variables.
`iprefix`	Quoted item number prefix. Quote the letter(s) preceding the item numbers as they are named in your data frame. For example, use `iprefix = "q"` if your items are named `"q1"`, `"q2"`, etc. Use either this `iprefix` argument or the `items` argument (but NOT BOTH) to tell the function which variables in `df` are the QLQ-C30 items. See Details for more information.

`items` A character vector with the QLQ-C30 item names, or a numeric vector indicating the column numbers of the QLQ-C30 items in `df`. Use either this `items` argument or the `iprefix` argument (but NOT BOTH) to tell the function which variables in `df` are the QLQ-C30 items. If `items` is omitted, then `qlq_c30` will assume that `df` contains **ONLY** the QLQ-C30 items and no other variables. See Details for more information.

`keepNvalid` Logical, whether to return variables containing the number of valid, non-missing items on each scale for each respondent should be returned in the data frame with the scale scores. The default is `FALSE`. Set to `TRUE` to return these variables, which will be named `"scalename_N"` (e.g., `QL_N`). Most users should omit this argument entirely. This argument might be removed from future versions of the package, so please let me know if you think this argument useful and would rather it remain a part of the function.

Details

This function returns a total of 16 different scores from the EORTC QLQ-C30 (Aaronson et al., 1993), including the new QLQ-C30 Summary Score (Giesinger et al, 2016). Scores are calculated according to the official scoring algorithms in the EORTC QLQ-C30 Scoring Manual (Fayers et al, 2001).

In addition to the name of your data frame containing the QLQ-C30 item responses (`df`), you need to tell the function how to find the variables that correspond to the QLQ-C30 items in `df`. You can do this in 1 of 3 different ways:

1. The easiest way is to use the `iprefix` argument. This assumes that your items are named using a consistent prefix, followed by the item number (e.g., 'q1', 'q2', 'q3', etc.). In this case, you could use `iprefix = 'q'`, and the function will know to look for items named 'q1' to 'q30' in your data (`df`). Note that this method will **NOT** work if your items are numbered with leading zeros for single digit item numbers (e.g., 'q01', 'q02', etc.).
2. The second way is to manually provide the item names or locations using the `items` argument. For example, if your first 10 variables in `df` contain demographics, followed by the 30 QLQ-C30 items **in order** starting with the 11th variable, then you could use `items = 11:40`.
3. The last way only applies if your data frame (`df`) contains **ONLY** the 30 variables corresponding to the 30 QLQ-C30 items, in order, with no other non-QLQ-C30 variables. In this case, you can just use the `df` argument and omit `iprefix` and `items`.

You can use EITHER the `iprefix` or `items` argument, or NEITHER of them (in the case of #3 above). **But you cannot use both.**

Value

A data frame with all of the QLQ-C30 scores is returned. All scores are scaled to range from 0-100, even scores based on single items. Be aware that these single-item scales still have only 4 possible values, even though they are transformed to range from 0-100. The scale name and number of items are listed below.

Global health status/QoL

- **QL** - Global health status/QoL (revised) (from 2 items)

Functional Scales (higher is better functioning)

- **PF** - Physical functioning (from 5 items)
- **RF** - Role functioning (from 2 items)
- **EF** - Emotional functioning (from 4 items)
- **CF** - Cognitive functioning (from 2 items)
- **SF** - Social functioning (from 2 items)

Symptom Scales (higher is more symptoms, worse functioning)

- **FA** - Fatigue (from 3 items)
- **NV** - Nausea and Vomiting (from 2 items)
- **PA** - Pain (from 2 items)

Single-Item Symptom Scores (higher is more symptoms, worse functioning)

- **DY** - Dyspnoea
- **SL** - Insomnia
- **AP** - Appetite Loss
- **CO** - Constipation
- **DI** - Diarrhoea
- **FI** - Financial Difficulties

QLQ-C30 Summary Score (higher is better functioning, fewer symptoms)

- **C30SUMMARY** - QLQ-C30 Summary Score, composed by taking mean of all scores except for QL (Global health status/QoL) and FI (Financial Difficulties)

Optionally, the data frame can additionally have variables containing the number of valid item responses on each scale for each respondent (if `keepNvalid = TRUE`, but this option might be removed in future package updates).

How Missing Data is Handled

The `qlq_c30` function will calculate the scale scores as long as at least half of the items on the given scale have valid, non-missing item responses. The `qlq_c30` function will calculate the QLQ-C30 Summary Score (`C30SUMMARY`) for a respondent only if all 13 scales contributing to that score are non-missing. Scores calculated in the presence of missing items are pro-rated so that their theoretical minimum and maximum values are identical to those from scores calculated from complete data.

Note

This function follows the scoring algorithm in the official EORTC QLQ-C30 Scoring Manual (Fayers et al, 2001) exactly, with two exceptions.

- **QLQ-C30 Summary Score** - The QLQ-C30 Summary Score `C30SUMMARY` was developed after the EORTC QLQ-C30 Scoring Manual was published. This summary scale was scored according to instructions on the EORTC website (http://groups.eortc.be/qol/manuals).

- **Scale Score Names** - The QLQ-C30 Scoring Manual names the Global Health Status/QoL scale, the Physical Functioning scale, and the Role Functioning scale 'QL2', 'PF2', and 'RF2', respectively, to indicate that these are revised versions of these scales. However, this clashes with the naming convention that many statisticians use for longitudinal assessments (e.g., where 'QL2' would be used to indicate the second 'QL' assessment). As such, this function drops the '2' suffix from these scale names.

References

Aaronson NK, Ahmedzai S, Bergman B, Bullinger M, Cull A, Duez NJ, Filiberti A, Flechtner H, Fleishman SB, Haes JCJM de, Kaasa S, Klee M, Osoba D, Razavi D, Rofe PB, Schraub S, Sneeuw K, Sullivan M, Takeda F (1993). The European Organization for Research and Treatment of Cancer QLQ-C30: A Quality-of-Life Instrument for Use in International Clinical Trials in Oncology. *JNCI J Natl Cancer Inst 85*:365-376.

Fayers PM, Aaronson NK, Bjordal K, Groenvold M, Curran D, Bottomley A, on behalf of the EORTC Quality of Life Group. *The EORTC QLQ-C30 Scoring Manual (3rd Edition)*. Published by: European Organisation for Research and Treatment of Cancer, Brussels 2001.

Giesinger JM, Kieffer JM, Fayers PM, Groenvold M, Petersen MA, Scott NW, Sprangers MAG, Velikova G, Aaronson NK (2016). Replication and validation of higher order models demonstrated that a summary score for the EORTC QLQ-C30 is robust. *Journal of Clinical Epidemiology 69*:79-88.

Examples

```
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 30, values = 1:4)
qlq_c30(dat, 'q')
```

Index