

# Package ‘PROscorer’

July 21, 2025

**Type** Package

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

**Version** 0.0.4

**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 for scoring PRO and 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

**URL** <https://github.com/raybaser/PROscorer>

**BugReports** <https://github.com/raybaser/PROscorer/issues>

**Imports** PROscorerTools

**Suggests** knitr, rmarkdown, testthat, covr

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Language** en-US

**NeedsCompilation** no

**Author** Ray Baser [aut, cre]

**Maintainer** Ray Baser <ray.stats@gmail.com>  
**Repository** CRAN  
**Date/Publication** 2023-10-16 22:20:05 UTC

**Contents**

fsfi . . . . .	2
narcc . . . . .	5
PROscorer . . . . .	7
qlq_blm30 . . . . .	9
qlq_c30 . . . . .	11
qlq_cr38 . . . . .	14
qlq_cx24 . . . . .	16
qlq_pan26 . . . . .	18
<b>Index</b>	<b>22</b>

---

fsfi	<i>Score the Female Sexual Function Index (FSFI)</i>
------	--

---

**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`  $\leq$  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`  $\leq$  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`  $\leq$  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`  $\leq$  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	<i>Score the Cognitive Causation (CC) and Negative Affect in Risk (NAR) scales</i>
-------	--

---

## 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 <code>items</code> is omitted, then <code>narcc</code> will assume that df contains <b>ONLY</b> 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 <code>c(itemMin, itemMax)</code> , indicating the minimum and maximum possible item responses. The default value is <code>c(0, 3)</code> , 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 <code>c(1, 4)</code> 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 <code>okmiss</code> , 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>

### Author(s)

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



**See Also**

Useful links:

- <https://github.com/raybaser/PROscorer>
- Report bugs at <https://github.com/raybaser/PROscorer/issues>

---

qlq\_blm30

---

*Score the EORTC QLQ-BLM30 Quality of Life Questionnaire*


---

**Description**

Scores the European Organization for Research and Treatment of Cancer (EORTC) QLQ-BLM30 Muscle-Invasive Bladder Cancer Module.

**Usage**

```
qlq_blm30(df, items = NULL, keepNvalid = FALSE)
```

**Arguments**

<code>df</code>	A data frame containing responses to the 30 QLQ-BLM30 items, and possibly other variables.
<code>items</code>	A character vector with the QLQ-BLM30 item names, or a numeric vector indicating the column numbers of the QLQ-BLM30 items in <code>df</code> . If <code>items</code> is omitted, then <code>qlq_blm30</code> will assume that <code>df</code> contains <b>ONLY</b> the QLQ-BLM30 items and no other variables. See Details for more information.
<code>keepNvalid</code>	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 7 different scores from the EORTC QLQ-BLM30. Scores are calculated according to the official scoring algorithms from the EORTC.

In addition to the name of your data frame containing the QLQ-BLM30 item responses (`df`), you need to tell the function how to find the variables that correspond to the QLQ-BLM30 items in `df`. You can do this in 1 of 2 ways:

1. The first 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-BLM30 items **in order** starting with the 11th variable, then you could use `items = 11:40`.
2. The second way only applies if your data frame (`df`) contains **ONLY** the 30 variables corresponding to the 30 QLQ-BLM30 items, in order, with no other non-QLQ-BLM30 variables. In this case, you can just use the `df` argument and omit `items`.

## Value

A data frame with all 7 of the QLQ-BLM30 scores is returned. Of the 7 scores, there are 5 multi-item Symptom Scales, 1 single-item Symptom Scale (Catheter Use Problems), and 1 multi-item Functional Scale (Sexual Functioning) (see below). 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 names and numbers of items are listed below.

### Symptom Scales (higher is more symptoms, worse functioning)

- **BLM\_US** - Urinary Symptoms (from 7 items)
- **BLM\_UP** - Urostomy Problems (from 6 items)
- **BLM\_FU** - Future Perspective (from 3 items)
- **BLM\_BAF** - Abdominal Bloating and Flatulence (from 2 items)
- **BLM\_BI** - Body Image (from 3 items)
- **BLM\_CU** - Catheter Use Problems (from 1 item)

### Functional Scales (higher is better functioning)

- **BLM\_SX** - Sexual Functioning (from 2, 4, 6, or 7 items, depending on the respondent)

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

For all scale scores **except for the Sexual Functioning scale**, the `qlq_blm30` 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. 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.

The Sexual Functioning scale is handled differently because it has several conditional questions that may not be applicable to a given patient (i.e., some questions apply only to men or only to women, and some questions apply only to sexually active respondents). As a result, the Sexual Functioning scale may have 2, 4, 6, or 7 questions applicable for a given respondent. Because the minimum number of applicable items is 2, the `qlq_blm30` function will calculate the Sexual Functioning scale score as long as at least 1 of the items has a valid, non-missing response. Note that this can result in the `qlq_blm30` function scoring the Sexual Functioning scale for some respondents when they should technically be assigned a missing value for the scale.

## References

The QLQ-BLM30 validation study does not appear to be publicly available.

Rammant E, Fox L, Beyer K, et al. The current use of the EORTC QLQ-NMIBC24 and QLQ-BLM30 questionnaires for the assessment of health-related quality of life in bladder cancer patients: a systematic review. *Qual Life Res*. Published online January 17, 2023. doi:10.1007/s11136-022-03335-4

## Examples

```
## Not run:
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 30, prefix = "blm", values = 1:4)
qlq_blm30(dat, items = 1:30)

## End(Not run)
```

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 <code>iprefix = "q"</code> if your items are named "q1", "q2", etc. Use either this <code>iprefix</code> argument or the <code>items</code> argument (but NOT BOTH) to tell the function which variables in <code>df</code> 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 <code>df</code> . Use either this <code>items</code> argument or the <code>iprefix</code> argument (but NOT BOTH) to tell the function which variables in <code>df</code> are the QLQ-C30 items. If both <code>items</code> and <code>iprefix</code> are omitted, then <code>qlq_c30</code> will assume that <code>df</code> contains <b>ONLY</b> 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 names and numbers 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, Rofo 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

```
## Not run:
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 30, values = 1:4)
qlq_c30(dat, 'q')

## End(Not run)
```

---

qlq\_cr38

---

*Score the EORTC QLQ-CR38 Quality of Life Questionnaire*


---

## Description

Scores the European Organization for Research and Treatment of Cancer (EORTC) QLQ-CR38 Colorectal Cancer Module. (Experimental: This function was written quickly... please hand score 1 or 2 patients and check for accuracy)

## Usage

```
qlq_cr38(df, items = NULL, keepNvalid = FALSE)
```

## Arguments

df	A data frame containing responses to the 38 QLQ-CR38 items, and possibly other variables.
items	A character vector with the QLQ-CR38 item names, or a numeric vector indicating the column numbers of the QLQ-CR38 items in df. If items is omitted, then qlq_CR38 will assume that df contains <b>ONLY</b> the QLQ-CR38 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 12 different scores from the EORTC QLQ-CR38. The scoring instructions for the QLQ-CR38 are no longer available from the EORTC, and the QLQ-CR38 seems to have been superseded by the QLQ-CR29. This function calculates the QLQ-CR38 scores based on an old SPSS syntax file from the EORTC, and from Sprangers et al. (1999).

In addition to the name of your data frame containing the QLQ-CR38 item responses (df), you need to tell the function how to find the variables that correspond to the QLQ-CR38 items in df. You can do this in 1 of 2 ways:

1. The first 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 38 QLQ-CR38 items **in order** starting with the 11th variable, then you could use `items = 11:48`.
2. The second way only applies if your data frame (df) contains **ONLY** the 38 variables corresponding to the 38 QLQ-CR38 items, in order, with no other non-QLQ-CR38 variables. In this case, you can just use the `df` argument and omit `items`.

## Value

A data frame with all 12 of the QLQ-CR38 scores is returned. Of the 12 scores, 4 are Functional Scales and 8 are Symptom Scales (see below). Of the 8 Symptom Scales, 1 is based on a single item and 7 are multi-item scales. Of the 4 Functional Scales, 2 are multi-item and 2 are based on single items. 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 names and numbers of items are listed below.

## How Missing Data is Handled

The `qlq_cr38` 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. 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

Some of the QLQ-CR38 items/scales are not applicable to all patients. There are 2 questions for men only, 2 questions for women only, 7 questions only for patients WITHOUT a stoma (colostomy bag), and 7 questions only for patients WITH a stoma. Patients will therefore have missing values for some of the scale scores.

### Functional Scales (higher is better functioning)

- **CR\_BI** - Body Image (3 items)
- **CR\_SX** - Sexual Functioning (2 items)
- **CR\_SE** - Sexual Enjoyment (1 item)
- **CR\_FU** - Future Perspective (1 item)

### Symptom Scales (higher is more symptoms, worse functioning)

- **CR\_MI** - Micturition Problems (3 items)

- **CR\_GI** - Gastrointestinal Tract Symptoms (5 items)
- **CR\_CT** - Chemotherapy Side-Effects (3 items)
- **CR\_DF** - Problems with Defecation (Only for Pts WITHOUT a stoma) (7 items)
- **CR\_STO** - Stoma-Related Problems (Only for Pts w/stoma) (7 items)
- **CR\_MSX** - Male Sexual Problems (2 items)
- **CR\_FSX** - Female Sexual Problems (2 items)
- **CR\_WL** - Weight Loss (1 item)

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).

References

Sprangers, M. A. G., te Velde, A., & Aaronson, N. K. (1999). The construction and testing of the EORTC colorectal cancer-specific quality of life questionnaire module (QLQ-CR38). *European Journal of Cancer*, 35(2), 238–247. [https://doi.org/10.1016/S0959-8049\(98\)00357-8](https://doi.org/10.1016/S0959-8049(98)00357-8)

Examples

```
## Not run:
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 38, prefix = "cr", values = 1:4)
qlq_cr38(dat, items = 1:38)

## End(Not run)
```

---

qlq_cx24	<i>Score the EORTC QLQ-CX24 Quality of Life Questionnaire</i>
----------	---

---

Description

Scores the European Organization for Research and Treatment of Cancer (EORTC) QLQ-CX24 Cervical Cancer Module.

Usage

```
qlq_cx24(df, items = NULL, keepNvalid = FALSE)
```

Arguments

df	A data frame containing responses to the 24 QLQ-CX24 items, and possibly other variables.
items	A character vector with the QLQ-CX24 item names, or a numeric vector indicating the column numbers of the QLQ-CX24 items in df. If items is omitted, then qlq_cx24 will assume that df contains <b>ONLY</b> the QLQ-CX24 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 9 different scores from the EORTC QLQ-CX24. Scores are calculated according to the official scoring algorithms from the EORTC.

In addition to the name of your data frame containing the QLQ-CX24 item responses (df), you need to tell the function how to find the variables that correspond to the QLQ-CX24 items in df. You can do this in 1 of 2 ways:

1. The first 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 24 QLQ-CX24 items **in order** starting with the 11th variable, then you could use `items = 11:34`.
2. The second way only applies if your data frame (df) contains **ONLY** the 24 variables corresponding to the 24 QLQ-CX24 items, in order, with no other non-QLQ-CX24 variables. In this case, you can just use the `df` argument and omit `items`.

## Value

A data frame with all 9 of the QLQ-CX24 scores is returned. Of the 9 scores, 7 are Symptom Scales and 2 are Functional Scales (see below). Of the 9 scores, only 3 are based on multiple items (the first 3 Symptom Scales), and the other 6 scores are each based on a single item. 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 names and numbers of items are listed below.

### Symptom Scales (higher is more symptoms, worse functioning)

- **CX\_SE** - Symptom Experience (from 11 items)
- **CX\_BI** - Body Image (from 3 items)
- **CX\_SV** - Sexual/Vaginal Functioning (from 4 items)
- **CX\_LY** - Lymphoedema (from 1 item)
- **CX\_PN** - Peripheral Neuropathy (from 1 item)
- **CX\_MS** - Menopausal Symptoms (from 1 item)
- **CX\_SXW** - Sexual Worry (from 1 item)

### Functional Scales (higher is better functioning)

- **CX\_SXA** - Sexual Activity (from 1 item)
- **CX\_SXE** - Sexual Enjoyment (from 1 item)

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_cx24` 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. 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

Note that the QLQ-CX24 instructs respondents to answer the last 5 questions on the instrument **only** if they have been sexually active during the past 4 weeks. These 5 items include all 4 items on the Sexual/Vaginal Functioning symptom scale and the 1 item on the Sexual Enjoyment functional scale. Because of this, when the QLQ-CX24 is scored correctly, **it is normal for a large number of respondents to have missing values for the Sexual/Vaginal Functioning and Sexual Enjoyment scores**. This is due to those respondents skipping those questions because they were not sexually active during the 4 weeks before completing the questionnaire.

### References

Greimel E, Kuljanic Vlasic K, Waldenstrom AC et al. on behalf of the EORTC Quality of Life Group. The European Organization for Research and Treatment of Cancer (EORTC) Quality-of-Life questionnaire cervical cancer module - EORTC QLQ-CX24. *Cancer* 107 (8): 1812-1822, 2006.

### Examples

```
## Not run:
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 24, prefix = "cx", values = 1:4)
qlq_cx24(dat, items = 1:24)

## End(Not run)
```

---

qlq\_pan26

*Score the EORTC QLQ-PAN26 Quality of Life Questionnaire*

---

### Description

Scores the European Organization for Research and Treatment of Cancer (EORTC) QLQ-PAN26 Pancreatic Cancer Module. (Experimental: This function was written quickly... please hand score 1 or 2 patients and check for accuracy)

### Usage

```
qlq_pan26(df, items = NULL, keepNvalid = FALSE)
```

## Arguments

<code>df</code>	A data frame containing responses to the 26 QLQ-PAN26 items, and possibly other variables.
<code>items</code>	A character vector with the QLQ-PAN26 item names, or a numeric vector indicating the column numbers of the QLQ-PAN26 items in <code>df</code> . If <code>items</code> is omitted, then <code>qlq_PAN26</code> will assume that <code>df</code> contains <b>ONLY</b> the QLQ-PAN26 items and no other variables. See Details for more information.
<code>keepNvalid</code>	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 17 different scores from the EORTC QLQ-PAN26. Scores are calculated according to the official scoring algorithms from the EORTC. At the time this scoring function was written (April 2022), the QLQ-PAN26 had completed Phase 3 testing; however, the official scoring instructions from the EORTC warned that this scaling structure is still preliminary and may change in the future.

In addition to the name of your data frame containing the QLQ-PAN26 item responses (`df`), you need to tell the function how to find the variables that correspond to the QLQ-PAN26 items in `df`. You can do this in 1 of 2 ways:

1. The first 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 26 QLQ-PAN26 items **in order** starting with the 11th variable, then you could use `items = 11:36`.
2. The second way only applies if your data frame (`df`) contains **ONLY** the 26 variables corresponding to the 26 QLQ-PAN26 items, in order, with no other non-QLQ-PAN26 variables. In this case, you can just use the `df` argument and omit `items`.

## Value

A data frame with all 17 of the QLQ-PAN26 scores is returned. Of the 17 scores, 15 are Symptom Scales and 2 are Functional Scales (see below). Of the 15 Symptom Scales, 10 are based on a single item each. 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 names and numbers of items are listed below.

### Symptom Scales (higher is more symptoms, worse functioning)

- **PAN\_PP** - Pancreatic pain (from 4 items)
- **PAN\_BF** - Bloating (from 1 item)
- **PAN\_DS** - Digestive symptoms (from 2 items)
- **PAN\_TA** - Taste (from 1 item)

- **PAN\_ID** - Indigestion (from 1 item)
- **PAN\_FL** - Flatulence (from 1 item)
- **PAN\_WL** - Weight loss (from 1 item)
- **PAN\_WE** - Weakness arms and legs (from 1 item)
- **PAN\_DM** - Dry mouth (from 1 item)
- **PAN\_LI** - Hepatic symptoms (from 2 items)
- **PAN\_BO** - Altered bowel habit (from 2 items)
- **PAN\_BI** - Body image (from 2 item)
- **PAN\_SE** - Troubled with side-effects(from 1 item)
- **PAN\_FU** - Future Worries (from 1 item)
- **PAN\_PL** - Planning of activities (from 1 item)

#### **Functional Scales (higher is better functioning)**

- **PAN\_SA** - Satisfaction with health care (from 2 items)
- **PAN\_SX** - Sexuality (from 2 items)

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_PAN26` 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. 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**

**As of April 15, 2022, there is an error in the official PAN26 scoring instructions from the EORTC.** The first page of the official PAN26 scoring instructions from the EORTC indicates that the two items comprising the SA (Satisfaction with health care) functional scale should be reverse scored. In the instructions, these two items are numbered as 53 and 54. However, the second page of the instructions indicates items 55 and 56 should be reversed (see bullet, "(1) Raw score" in the "Principle for scoring" subsection). Items 55 and 56 comprise the SX (Sexuality) functional scale. **The items that need to be reversed are the SX items (55 and 56), NOT the SA items (not 53 and 54).** This function correctly reverses the SX items instead of the SA items.

#### **References**

Fitzsimmons D, Johnson CD, George S, et al. Development of a disease specific quality of life (QoL) questionnaire module to supplement the EORTC core cancer QoL questionnaire, the QLQ-PAN26 in patients with pancreatic cancer. *Eur. J. Cancer* 35: 939-941, 1999.

**Examples**

```
## Not run:  
dat <- PROscorerTools::makeFakeData(n = 10, nitems = 26, prefix = "pan", values = 1:4)  
qlq_pan26(dat, items = 1:26)  
  
## End(Not run)
```

# Index

fsfi, [2](#)

narcc, [5](#)

PROscorer, [7](#)

PROscorer-package (PROscorer), [7](#)

qlq\_blm30, [9](#)

qlq\_c30, [11](#)

qlq\_cr38, [14](#)

qlq\_cx24, [16](#)

qlq\_pan26, [18](#)