# Package ‘JDCruncheR’

May 22, 2024

**Type** Package

**Title** Interface Between the 'JDemetra+' Cruncher and R, and Quality Report Generator

**Version** 0.2.4

**Description** Tool for generating quality reports from cruncher outputs (and calculating series scores). The latest version of the cruncher can be downloaded here: <https://github.com/jdemetra/jwsacruncher/releases>.

**URL** https://github.com/InseeFr/JDCruncheR

**BugReports** https://github.com/InseeFr/JDCruncheR/issues

**Imports** XLConnect (>= 1.0.0)

**Suggests** knitr, kableExtra, rmarkdown

**License** EUPL

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Tanguy Barthelemy [aut, cre, art],
Alain Quartier-la-Tente [aut] (<https://orcid.org/0000-0001-7890-3857>),
Institut national de la statistique et des études économiques [cph]
(https://www.insee.fr/),
Anna Smyk [aut]

**Maintainer** Tanguy Barthelemy <tanguy.barthelemy@insee.fr>

**Repository** CRAN

**Date/Publication** 2024-05-22 12:20:13 UTC

## R topics documented:

## `add_indicator` *Adding an indicator in QR_matrix objects*

### Description

Function to add indicators in `QR_matrix` objects.

### Usage

```
add_indicator(x, indicator, variable_name, ...)
```

### Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix` object |
| `indicator` | a `vector` or a `data.frame` (cf. details). |
| `variable_name` | a string containing the name of the variables to add. |
| `...` | other parameters of the function `merge`. |

### Details

The function `add_indicator()` adds the chosen indicator to the values matrix of a quality report. Therefore, because said indicator isn't added in the modalities matrix, it cannot be used to calculate a score (except for weighting). Before using the added variable for score calculation, it will have to be coded with the function `recode_indicator_num`.

The new indicator can be a `vector` or a `data.frame`. In both cases, its format must allow for pairing:

- a `vector`'s elements must be named and these names must match those of the quality report (variable "series");
- a `data.frame` must contain a "series" column that matches with the quality report's series.

## Value

This function returns the same object, enhanced with the chosen indicator. So if the input x is a QR_matrix, an object of class QR_matrix is returned. If the input x is a mQR_matrix, an object of class mQR_matrix is returned.

## See Also

Traduction française

Other var QR_matrix manipulation: QR_var_manipulation, recode_indicator_num()

---

compute_score *Score calculation*

---

## Description

To calculate a score for each series from a quality report

## Usage

```
## S3 method for class 'QR_matrix'
compute_score(
  x,
 score_pond = c(qs_residual_sa_on_sa = 30, f_residual_sa_on_sa = 30, qs_residual_sa_on_i
   = 20, f_residual_sa_on_i = 20, f_residual_td_on_sa = 30, f_residual_td_on_i = 20,
   oos_mean = 15, oos_mse = 10, residuals_independency = 15, residuals_homoskedasticity
    = 5, residuals_skewness = 5, m7 = 5, q_m2 = 5),
  modalities = c("Good", "Uncertain", "", "Bad", "Severe"),
  normalize_score_value,
  na.rm = FALSE,
  n_contrib_score,
  conditional_indicator,
  ...
)
```

## Arguments

| | |
|---|---|
| x | a QR_matrix or mQR_matrix object. |
| score_pond | the formula used to calculate the series score. |
| modalities | modalities ordered by importance in the score calculation (cf. details). |
| normalize_score_value | integer indicating the reference value for weights normalisation. If missing, weights will not be normalised. |
| na.rm | logical indicating whether missing values must be ignored when calculating the score. |

`n_contrib_score`
: integer indicating the number of variables to create in the quality report's values matrix to store the `n_contrib_score` greatest contributions to the score (cf. details). If not specified, no variable is created.

`conditional_indicator`
: a `list` containing 3-elements sub-lists: "indicator", "conditions" and "condition_modalities". To reduce down to 1 the weight of chosen indicators depending on other variables' values (cf. details).

`...`
: other unused parameters.

## Details

The function `compute_score` calculates a score from the modalities of a quality report: to each modality corresponds a weight that depends on the parameter `modalities`. The default parameter is `c("Good", "Uncertain", "Bad","Severe")`, and the associated weights are respectively 0, 1, 2 and 3.

The score calculation is based on the `score_pond` parameter, which is a named integer vector containing the weights to apply to the (modalities matrix) variables. For example, with `score_pond = c(qs_residual_sa_on_sa = 10, f_residual_td_on_sa = 5)`, the score will be based on the variables qs_residual_sa_on_sa and f_residual_td_on_sa. The qs_residual_sa_on_sa grades will be multiplied by 10 and the f_residual_td_on_sa grades, by 5. To ignore the missing values when calculating a score, use the parameter `na.rm = TRUE`.

The parameter `normalize_score_value` can be used to normalise the scores. For example, to have all scores between 0 and 20, specify `normalize_score_value = 20`.

When using parameter `n_contrib_score`, `n_contrib_score` new variables are added to the quality report's values matrix. These new variables store the names of the variables that contribute the most to the series score. For example, `n_contrib_score = 3` will add to the values matrix the three variables that contribute the most to the score. The new variables' names are *i*_highest_score, with *i* being the rank in terms of contribution to the score (1_highest_score contains the name of the greatest contributor, 2_highest_score the second greatest, etc). Only the variables that have a non-zero contribution to the score are taken into account: if a series score is 0, all *i*_highest_score variables will be empty. And if a series score is positive only because of the m7 statistic, 1_highest_score will have a value of "m7" for this series and the other *i*_highest_score will be empty.

Some indicators are only relevant under certain conditions. For example, the homoscedasticity test is only valid when the residuals are independant, and the normality tests, only when the residuals are both independant and homoscedastic. In these cases, the parameter `conditional_indicator` can be of use since it reduces the weight of some variables down to 1 when some conditions are met. `conditional_indicator` is a `list` of 3-elements sub-lists:

- "indicator": the variable whose weight will be conditionally changed
- "conditions": the variables used to define the conditions
- "conditions_modalities": modalities that must be verified to induce the weight change For example, `conditional_indicator = list(list(indicator = "residuals_skewness", conditions = c("residuals_independency", "residuals_homoskedasticity"), conditions_modalities = c("Bad","Severe")))`, reduces down to 1 the weight of the variable "residuals_skewness" when the modalities of the independancy test ("residuals_independency") or the homoscedasticity test ("residuals_homoskedasticity") are "Bad" or "Severe".

## Value

a QR_matrix or mQR_matrix object.

## See Also

Traduction française

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute the score
QR <- compute_score(QR, n_contrib_score = 2)
print(QR)

# Extract the modalities matrix:
QR$modalities$score
```

---

`export_xlsx` *Exporting QR_matrix or mQR_matrix objects in an Excel file*

---

## Description

Exporting QR_matrix or mQR_matrix objects in an Excel file

## Usage

```
export_xlsx(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix` object. |
| `...` | other parameters of the function `export_xlsx.QR_matrix`. |

## Value

If `x` is a `mQR_matrix`, the function returns invisibly (via `invisible(x)`) the same `mQR_matrix` object as `x`. Else if `x` is a `QR_matrix`, the function returns invisibly (via `invisible(x)`) a workbook object created by `XLConnect::loadWorkbook()` for further manipulation.

## See Also

Other QR_matrix functions: `export_xlsx.QR_matrix()`, `export_xlsx.mQR_matrix()`, `extract_QR()`, `rbind.QR_matrix()`, `sort()`, `weighted_score()`

---

`export_xlsx.mQR_matrix` *Exporting mQR_matrix objects in Excel files*

---

## Description

To export several quality reports in Excel files

## Usage

```
## S3 method for class 'mQR_matrix'
export_xlsx(
  x,
  export_dir = "./",
  layout_file = c("ByComponent", "ByQRMatrix"),
  file_extension = c(".xls", ".xlsx"),
  layout = c("all", "modalities", "values", "combined"),
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | a `mQR_matrix` object to export. |
| `export_dir` | export directory. |
| `layout_file` | export parameter. By default, (`layout_file = "ByComponent"`) and an Excel file is exported for each part of the quality report matrix (modalities and values matrices). To group both modalities and values reports/sheets into a single Excel file, use the option `layout_file = "ByQRMatrix"`. |
| `file_extension` | possible values are `".xls"` and `".xlsx"`. |
| `layout` | elements of the report to export: see `export_xlsx.QR_matrix` . |
| `...` | other parameters of the function `export_xlsx.QR_matrix`. |

## Value

Returns invisibly (via `invisible(x)`) the same `mQR_matrix` object as `x`.

## See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.QR_matrix()`, `extract_QR()`, `rbind.QR_matrix()`, `sort()`, `weighted_score()`

## export_xlsx.QR_matrix *Exporting QR_matrix objects in an Excel file*

### Description

To export a quality report in an Excel file.

### Usage

```
## S3 method for class 'QR_matrix'
export_xlsx(
  x,
  layout = c("all", "modalities", "values", "combined"),
  create = TRUE,
  clear_sheet = TRUE,
  auto_format = TRUE,
  file_name,
  sheet_names,
  ...
)
```

### Arguments

| | |
|---|---|
| `x` | a `QR_matrix` object. |
| `layout` | the components of the report to export. By default, `layout = "all"`: the matrices modalities (`"modalities"`) and values (`"values"`) are exported in separate files. To export them in a single file (in two sheets), use `layout = "combined"`. |
| `create` | logical indicating whether to create an Excel file if it doesn't exist yet (`create = TRUE` by default) |
| `clear_sheet` | logical indicating whether to clear the Excel sheets before the export (`clear_sheet = TRUE` by default). |
| `auto_format` | logical indicating whether to format the output (`auto_format = TRUE` by default). |
| `file_name` | optional argument to choose the path and name of the file to export. If not specified, an *export.xls* will be created in the working directory. |
| `sheet_names` | names of the exported Excel sheets. If not specified, the sheets will be named after the exported components. If specified, existing sheets with these names will be overwritten. |
| `...` | other unused parameters. |

### Value

Returns invisibly (via `invisible(x)`) a workbook object created by `XLConnect::loadWorkbook()` for further manipulation.

## See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.mQR_matrix()`, `extract_QR()`, `rbind.QR_matrix()`, `sort()`, `weighted_score()`

---

`extract_QR` *Extraction of a quality report*

---

## Description

To extract a quality report from the csv file containing the diagnostics matrix.

## Usage

```
extract_QR(matrix_output_file, sep = ";", dec = ",")
```

## Arguments

| | |
|---|---|
| `matrix_output_file` | the csv file containing the diagnostics matrix. |
| `sep` | the separator used in the csv file (by default, `sep = ";"`) |
| `dec` | the decimal separator used in the csv file (by default, `dec = ","`) |

## Details

This function generates a quality report from a csv file containing diagnostics (usually from the file *demetra_m.csv*). The *demetra_m.csv* file can be generated by launching the cruncher (functions `cruncher` or `cruncher_and_param`) with the default export parameters, having used the default option `csv_layout = "vtable"` to format the output tables of the functions `cruncher_and_param` and `create_param_file` when creating the parameters file.

This function returns a `QR_matrix` object, which is a list of 3 objects:

- `modalities`, a `data.frame` containing several indicators and their categorical quality (Good, Uncertain, Bad, Severe).
- `values`, a `data.frame` containing the same indicators and the values that lead to their quality category (i.e.: p-values, statistics, etc.) as well as additional variables that don't have a modality/quality (series frequency and arima model).
- `score_formula` that will store the formula used to calculate the score (when relevant). Its initial value is `NULL`.

## Value

a `QR_matrix` object.

## See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.QR_matrix()`, `export_xlsx.mQR_matrix()`, `rbind.QR_matrix()`, `sort()`, `weighted_score()`

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

print(QR)

# Extract the modalities matrix:
QR$modalities
# Or:
QR[["modalities"]]
```

---

`extract_score` *Score extraction*

---

## Description

To extract score variables from `QR_matrix` or `mQR_matrix` objects.

## Usage

```
extract_score(
  x,
  format_output = c("data.frame", "vector"),
  weighted_score = FALSE
)
```

## Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix`. |
| `format_output` | string of characters indicating the output format: either a `data.frame` or a `vector`. |
| `weighted_score` | logical indicating whether to extract the weighted score (if previously calculated) or the unweighted one. By default, the unweighted score is extracted. |

## Details

For `QR_matrix` objects, the output is a vector or the object `NULL` if no score was previously calculated. For `mQR_matrix` objects, it is a list of scores (`NULL` elements or vectors).

## Value

`extract_score()` returns a data.frame with two column: the series name and their score.

## See Also

Traduction française

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute the score
QR1 <- compute_score(QR, n_contrib_score = 2)
mQR <- mQR_matrix(QR, compute_score(QR))

# Extract score
extract_score(QR1)
extract_score(mQR)
```

---

`print.QR_matrix` *Printing QR_matrix and mQR_matrix objects*

---

## Description

To print information on a QR_matrix or mQR_matrix object.

## Usage

```
## S3 method for class 'QR_matrix'
print(x, print_variables = TRUE, print_score_formula = TRUE, ...)

## S3 method for class 'mQR_matrix'
print(x, score_statistics = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | a `mQR_matrix` or `mQR_matrix` object. |
| `print_variables` | logical indicating whether to print the indicators' name (including additionnal variables). |
| `print_score_formula` | logical indicating whether to print the formula with which the score was calculated (when calculated). |
| `...` | other unused arguments. |
| `score_statistics` | logical indicating whether to print the statistics in the `mQR_matrix` scores (when calculated). |

## Value

the `print` method prints a `mQR_matrix` or `mQR_matrix` object and returns it invisibly (via `invisible(x)`).

## See Also

Traduction française

---

`QR_matrix` *Quality report objects*

---

## Description

`mQR_matrix()` and `QR_matrix()` are creating one (or several) quality report. The function `is.QR_matrix()` and `is.mQR_matrix()` are functions to test whether an object is a quality report or a list of quality reports.

## Usage

```
QR_matrix(modalities = NULL, values = NULL, score_formula = NULL)

mQR_matrix(x = list(), ...)

is.QR_matrix(x)

is.mQR_matrix(x)
```

## Arguments

| | |
|---|---|
| `modalities` | a `data.frame` containing the output variables' modalities (Good, Bad, etc.) |
| `values` | a `data.frame` containing the output variables' values (test p-values, test statistics, etc.) Therefore, the values data frame can contain more variables than the data frame `modalities`. |

| | |
|---|---|
| `score_formula` | the formula used to calculate the series score (if defined). |
| `x` | a `QR_matrix` object, a `mQR_matrix` object or a list of `QR_matrix` objects. |
| `...` | objects of the same type as `x`. |

### Details

A`QR_matrix` object is a list of three items:

- `modalities`, a `data.frame` containing a set of categorical variables (by default: Good, Uncertain, Bad, Severe).
- `values`, a `data.frame` containing the values corresponding to the `modalities` indicators (i.e. p-values, statistics, etc.), as well as variables for which a modality cannot be defined (e.g. the series frequency, the ARIMA model, etc).
- `score_formula` contains the formula used to calculate the series score (once the calculus is done).

### Value

`QR_matrix()` creates and returns a `QR_matrix` object. `mQR_matrix()` creates and returns a `mQR_matrix` object (ie. a list of `QR_matrix` objects). `is.QR_matrix()` and `is.mQR_matrix()` return Boolean values (`TRUE` or `FALSE`).

### See Also

Traduction française

---

## `QR_var_manipulation` *Editing the indicators list*

---

### Description

Functions to remove indicators (`remove_indicators()`) or retrain some indicators only (`retain_indicators()`) from `QR_matrix` or `mQR_matrix` objects. The series names (column "series") cannot be removed.

### Usage

```
remove_indicators(x, ...)

retain_indicators(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix` object. |
| `...` | names of the variable to remove (or keep) |

## Value

`remove_indicators()` returns the same object `x` reduced by the flags and variables used as arguments ... So if the input `x` is a QR_matrix, an object of class QR_matrix is returned. If the input `x` is a mQR_matrix, an object of class mQR_matrix is returned.

## See Also

Traduction française

Other var QR_matrix manipulation: `add_indicator()`, `recode_indicator_num()`

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute the score
QR <- compute_score(QR, n_contrib_score = 2)

# Retain indicators
retain_indicators(QR, "score", "m7") # retaining "score" and "m7"
retain_indicators(QR, c("score", "m7")) # Same

# Remove indicators
QR <- retain_indicators(QR, "score") # removing "score"

extract_score(QR) # is NULL because we removed the score indicator
```

---

`rbind.QR_matrix` *Combining QR_matrix objects*

---

## Description

Function to combine multiple `QR_matrix` objects: line by line, both for the `modalities` and the `values` table.

## Usage

```
## S3 method for class 'QR_matrix'
rbind(..., check_formula = TRUE)
```

## Arguments

| | |
|---|---|
| `...` | `QR_matrix` objects to combine. |
| `check_formula` | logical indicating whether to check the score formulas' coherency. By default, `check_formula = TRUE`: an error is returned if the scores were calculated with different formulas. If `check_formula = FALSE`, no check is performed and the `score_formula` of the output is `NULL`. |

## Value

`rbind.QR_matrix()` returns a `QR_matrix` object.

## See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.QR_matrix()`, `export_xlsx.mQR_matrix()`, `extract_QR()`, `sort()`, `weighted_score()`

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute differents scores
QR1 <- compute_score(QR, score_pond = c(m7 = 2, q = 3, qs_residual_sa_on_sa = 5))
QR2 <- compute_score(QR, score_pond = c(m7 = 2, qs_residual_sa_on_sa = 5))

# Merge two quality report
try(rbind(QR1, QR2)) # Une erreur est renvoyée
rbind(QR1, QR2, check_formula = FALSE)
```

---

## `recode_indicator_num` *Converting "values variables" into "modalities variables"*

---

## Description

To transform variables from the values matrix into categorical variables that can be added into the modalities matrix.

### Usage

```
recode_indicator_num(
  x,
  variable_name,
  breaks = c(0, 0.01, 0.05, 0.1, 1),
  labels = c("Good", "Uncertain", "Bad", "Severe"),
  ...
)
```

### Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix` object. |
| `variable_name` | a vector of strings containing the names of the variables to convert. |
| `breaks` | see function `cut`. |
| `labels` | see function `cut`. |
| `...` | other parameters of the `cut` function. |

### Value

The function `recode_indicator_num()` returns the same object, enhanced with the chosen indicator. So if the input `x` is a QR_matrix, an object of class QR_matrix is returned. If the input `x` is a mQR_matrix, an object of class mQR_matrix is returned.

### See Also

Traduction française

Other var QR_matrix manipulation: `QR_var_manipulation`, `add_indicator()`

---

## `sort` *QR_matrix and mQR_matrix sorting*

---

### Description

To sort the quality reports on one or several variables

### Usage

```
## S3 method for class 'QR_matrix'
sort(x, decreasing = FALSE, sort_variables = "score", ...)

## S3 method for class 'mQR_matrix'
sort(x, decreasing = FALSE, sort_variables = "score", ...)
```

### Arguments

| | |
|---|---|
| `x` | a `QR_matrix` or `mQR_matrix` object |
| `decreasing` | logical indicating whether the quality reports must be sorted in ascending or decreasing order. By default, the sorting is done in ascending order. |
| `sort_variables` | They must be present in the modalities table. |
| `...` | other parameters of the function `order` (unused for now) |

### Value

the input with sorted quality reports

### See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.QR_matrix()`, `export_xlsx.mQR_matrix()`, `extract_QR()`, `rbind.QR_matrix()`, `weighted_score()`

### Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute the score
QR <- compute_score(QR, n_contrib_score = 2)
print(QR$modalities$score)

# Sort the scores
QR <- sort(QR, sort_variables = "score") # Pour trier par ordre croissant sur le score
print(QR$modalities$score)
```

## `weighted_score` *Weighted score calculation*

### Description

Function to weight a pre-calculated score

### Usage

```
weighted_score(x, pond = 1)
```

## Arguments

| | |
|---|---|
| x | a `QR_matrix` or `mQR_matrix` object |
| pond | the weights to use. Can be an integer, a vector of integers, the name of one of the quality report variables or a list of weights for the `mQR_matrix` objects. |

## Value

the input with an additionnal weighted score

## See Also

Traduction française

Other QR_matrix functions: `export_xlsx()`, `export_xlsx.QR_matrix()`, `export_xlsx.mQR_matrix()`, `extract_QR()`, `rbind.QR_matrix()`, `sort()`

## Examples

```
# Path of matrix demetra_m
demetra_path <- file.path(
    system.file("extdata", package = "JDCruncheR"),
    "WS/ws_ipi/Output/SAProcessing-1",
    "demetra_m.csv"
)

# Extract the quality report from the demetra_m file
QR <- extract_QR(demetra_path)

# Compute the score
QR <- compute_score(QR, n_contrib_score = 2)

# Weighted score
QR <- weighted_score(QR, 2)
print(QR)

# Extract the weighted score
QR$modalities$score_pond
```

# Index