Package 'qat'

July 2, 2014

Type Package

Title Quality Assurance Toolkit

Version 0.72

Date 2013-06-13

Author Andre Duesterhus

Maintainer Andre Duesterhus <andue@uni-bonn.de>

Description This package delivers some functions to provide a scientific quality assurance of meteorological data.

Depends R (>= 2.6.1), ncdf, gplots, XML, gdata, moments, boot, fields

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-06-19 00:49:39

R topics documented:

qat-package *Quality Assurance Toolkit*

Description

This package helps to provide a quality assurance on data.

Details

Package:	qat
Type:	Package
Version:	0.72
Date:	2013-06-13
License:	GPL-2

Author(s)

Andre Duesterhus
Maintainer: Andre Duesterhus <andue@uni-bonn.de>

Examples

```
library("qat")
# define testvector
testvector<-rnorm(200)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=200)
minlim <- seq(-1,-3,length.out=200)
uproc <- seq(1,3,length.out=200)
downroc <- seq(3,1,length.out=200)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
```

```
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

`qat_add_algorithm` *Algorithm of a check*

Description

For each check in the workflow it is possible to add a algorithm of the test. This will be saved into the XML result file under agolgorithm. This function adds a new or replace an existing algorithm.

Usage

```
qat_add_algorithm(workflowlist, listelem, algorithm_text)
```

Arguments

`workflowlist`	A workflowlist like it will be created by qat_config_read_workflow
`listelem`	Number of check, where the algorithm should be added.
`algorithm_text`	Text of the algorithm.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

`qat_config_read_workflow`

Examples

```
library("qat")
## read in workflow from systemfiles
# filename_in <- system.file("extdata/workflowexample.xml", package="qat")
# workflowlist <- qat_config_read_workflow(filename_in)
## add some more informations for the workflow
# workflowlist <- qat_add_algorithm(workflowlist, 1, "Algorithm information")
# filename_out <- "myworkflow_result.xml"
## write edited workflow in current directory
# qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_all_algorithms *Adds all algorithms to a workflow*

Description

For each check in the workflow it is possible to add a algorithm of the test. This will be saved into the XML result file under agolgorithm. This function adds for each test the known algorithm-information.

Usage

```
qat_add_all_algorithms(workflowlist)
```

Arguments

workflowlist	A workflowlist like it will be created by qat_config_read_workflow

Details

This function use the informatio, which is stored in the system file qat_basetools.xml.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
# filename_in <- system.file("extdata/workflowexample.xml", package="qat")
# workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
# workflowlist <- qat_add_all_descriptions(workflowlist)
# workflowlist <- qat_add_all_algorithms(workflowlist)

# filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
# qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_all_descriptions *Adds all descriptions to a workflow*

Description

For each check in the workflow it is possible to add a description of the test. This will be saved into the XML result file under the description. This function adds for each test the known description-information.

Usage

```
qat_add_all_descriptions(workflowlist)
```

Arguments

workflowlist	A workflowlist like it will be created by qat_config_read_workflow

Details

This function use the informatio, which is stored in the system file qat_basetools.xml.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_comment *Comment on result*

Description

For each check in the workflow it is possible to add a comment on the result of the test. This will be saved into the XML result file under the tag result/comment_on_result. This function adds a new or replace an existing comment.

Usage

```
qat_add_comment(workflowlist, listelem, comment_text)
```

Arguments

workflowlist	A workflowlist like it will be created by qat_config_read_workflow
listelem	Number of check, which should be commented.
comment_text	Text of the comment

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_description *Description of a check*

Description

For each check in the workflow it is possible to add a description of the test. This will be saved into the XML result file under the description. This function adds a new or replace an existing description.

Usage

```
qat_add_description(workflowlist, listelem, description_text)
```

Arguments

workflowlist	A workflowlist like it will be created by qat_config_read_workflow
listelem	Number of check, where the description should be added.
description_text	Text of the description.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_description(workflowlist, 1, "How the test works...")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_add_resultfile *Resultfile of a check*

Description

For each check in the workflow it is possible to add one or more result files of the test. This will be saved into the XML result file under result_file. This function adds a new resultfile.

Usage

```
qat_add_resultfile(workflowlist, listelem, resultfile_text)
```

Arguments

workflowlist	A workflowlist like it will be created by qat_config_read_workflow.
listelem	Number of check, where the resultfile should be added.
resultfile_text	Text of the resultfile.

Value

Give back the edited workflowlist.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_resultfile(workflowlist, 1, "filename.png")
filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

qat_analyse_block_distribution_1d *Perform a block distribution check*

Description

The measurement vector will be splitted into blocks, and on every block some statistical parameters will be calculated.

Usage

```
qat_analyse_block_distribution_1d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Length of the blocks

Details

The measurement vector will be splitted into blocks, with the length of the given blocksize parameter. After this some statistical parameters will be calculated for every block. As a result a list will be given back, with these parameters, where every entry got a length of the length of the measurement vector divided by the blocksize, which is rounded down to the next integer.

Value

It returns a list with the following entries:

first_moment	First moment of the measurement vector
second_moment	Second moment of the measurement vector
third_moment	Third moment of the measurement vector
fourth_moment	Fourth moment of the measurement vector
standard_deviation	Standard deviation of the measurement vector
skewness	Skewness of the measurement vector
kurtosis	Kurtosis of the measurement vector
median	Median of the measurement vector
p5_quantile	5 percent quantile of the measurement vector
p95_quantile	95 percent quantile of the measurement vector
p25_quantile	25 percent quantile of the measurement vector
p75_quantile	75 percent quantile of the measurement vector
blocksize	Length of the used blocks

Author(s)

Andre Duesterhus

See Also

qat_plot_block_distribution_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_block_distribution_1d(vec, 50)
```

---

qat_analyse_block_distribution_2d

Perform a block distribution check

Description

The measurement vector will be splitted into blocks in the direction of the first dimension. After this on every block some statistical parameters will be calculated.

Usage

```
qat_analyse_block_distribution_2d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Length of the blocks

Details

The measurement vector will be splitted into blocks for each element of the second dimension, with the length of the given blocksize parameter. After this some statistical parameters will be calculated for each block. As a result a list will be given back, with these parameters, where every entry got the dimension of the measurement vector, where the first dimension is divided by the blocksize, which is rounded down to the next integer.

Value

It returns a list with the following entries:

first_moment	First moment of the measurement vector
second_moment	Second moment of the measurement vector
third_moment	Third moment of the measurement vector
fourth_moment	Fourth moment of the measurement vector

standard_deviation	Standard deviation of the measurement vector
skewness	Skewness of the measurement vector
kurtosis	Kurtosis of the measurement vector
median	Median of the measurement vector
p5_quantile	5 percent quantile of the measurement vector
p95_quantile	95 percent quantile of the measurement vector
p25_quantile	25 percent quantile of the measurement vector
p75_quantile	75 percent quantile of the measurement vector
blocksize	Length of the used blocks

Author(s)

Andre Duesterhus

See Also

qat_analyse_block_distribution_1d, qat_plot_block_distribution_2d

Examples

```
vec <- array(rnorm(500),c(25,20))
result <- qat_analyse_block_distribution_2d(vec, 5)
```

---

qat_analyse_boot_distribution_1d

Perform a bootstrapped distribution check

Description

The measurement vector will be bootstrapped and statistical parameters will be determined.

Usage

```
qat_analyse_boot_distribution_1d(measurement_vector, bootruns)
```

Arguments

measurement_vector	The measurement vector, which should be tested
bootruns	Number of bootstrap runs, which should be performed

Details

The measurement vector will be bottstrapped with the number of runs, which is given by the parameter bootruns. From each runs, some statistical parameters will be calculated and given back in the resultlist.

Value

It returns a list with the following entries:

`first_moment`	First moments of the bootstrapped measurement vector
`second_moment`	Second moments of the bootstrapped measurement vector
`third_moment`	Third moments of the bootstrapped measurement vector
`fourth_moment`	Fourth moments of the bootstrapped measurement vector
`standard_deviation`	Standard deviations of the bootstrapped measurement vector
`skewness`	Skewness of the bootstrapped measurement vector
`kurtosis`	Kurtosis of the bootstrapped measurement vector
`median`	Medians of the bootstrapped measurement vector
`p5_quantile`	5 percent quantiles of the bootstrapped measurement vector
`p95_quantile`	95 percent quantiles of the bootstrapped measurement vector
`p25_quantile`	25 percent quantiles of the bootstrapped measurement vector
`p75_quantile`	75 percent quantiles of the bootstrapped measurement vector

Author(s)

Andre Duesterhus

See Also

`qat_plot_boot_distribution_1d`

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_boot_distribution_1d(vec, 1000)
```

---

qat_analyse_boot_distribution_2d

Perform a bootstrapped distribution check

Description

The measurement vector will be bootstrapped and statistical parameters will be determined.

Usage

```
qat_analyse_boot_distribution_2d(measurement_vector, bootruns)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
bootruns	Number of bootstrap runs, which should be performed

Details

The measurement vector will be bottstrapped in direction of the first dimension with the number of runs, which is given by the parameter bootruns. From each runs, some statistical parameters will be calculated and given back in the resultlist.

Value

It returns a list with the following entries:

first_moment	First moments of the bootstrapped measurement vector
second_moment	Second moments of the bootstrapped measurement vector
third_moment	Third moments of the bootstrapped measurement vector
fourth_moment	Fourth moments of the bootstrapped measurement vector
standard_deviation	Standard deviations of the bootstrapped measurement vector
skewness	Skewness of the bootstrapped measurement vector
kurtosis	Kurtosis of the bootstrapped measurement vector
median	Medians of the bootstrapped measurement vector
p5_quantile	5 percent quantiles of the bootstrapped measurement vector
p95_quantile	95 percent quantiles of the bootstrapped measurement vector
p25_quantile	25 percent quantiles of the bootstrapped measurement vector
p75_quantile	75 percent quantiles of the bootstrapped measurement vector

Author(s)

Andre Duesterhus

See Also

qat_analyse_boot_distribution_1d, qat_plot_boot_distribution_1d

Examples

```
vec <- array(rnorm(100),c(25,20))
result <- qat_analyse_boot_distribution_2d(vec, 50)
```

qat_analyse_distribution_1d *Perform a distribution check*

Description

This check makes a histogram and gives back some statistical parameters of the given measurement vector.

Usage

```
qat_analyse_distribution_1d(measurement_vector, numofbars)
```

Arguments

measurement_vector	The measurement vector, which should be tested
numofbars	Numbers of bars of the histogram plot

Details

From a given measurement vector a histogram will be performed. The number of bars of this will be given by the parameter numofbars. Additionally some statistical parameters, like the first moments and some quantiles will be calculated.

Value

It returns a list with the following entries:

first_moment	First moment of the measurement vector
second_moment	Second moment of the measurement vector
third_moment	Third moment of the measurement vector
fourth_moment	Fourth moment of the measurement vector
standard_deviation	Standard deviation of the measurement vector
skewness	Skewness of the measurement vector
kurtosis	Kurtosis of the measurement vector
median	Median of the measurement vector
p5_quantile	5 percent quantile of the measurement vector
p95_quantile	95 percent quantile of the measurement vector
p25_quantile	25 percent quantile of the measurement vector
p75_quantile	75 percent quantile of the measurement vector
numofbars	Number of bars of the histogram
...	Elements of the histogram

Author(s)

Andre Duesterhus

See Also

qat_plot_distribution_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_distribution_1d(vec, 15)
```

---

qat_analyse_distribution_2d

Perform a distribution check

Description

This check makes a histogram and gives back some statistical parameters of the given measurement vector.

Usage

```
qat_analyse_distribution_2d(measurement_vector, numofbars)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
numofbars	Numbers of bars of the histogram plot

Details

From a given measurement vector (2d array) a histogram will be performed. The number of bars of this will be given by the parameter numofbars. Additionally some statistical parameters, like the first moments and some quantiles will be calculated.

Value

It returns a list with the following entries:

first_moment	First moment of the measurement vector
second_moment	Second moment of the measurement vector
third_moment	Third moment of the measurement vector
fourth_moment	Fourth moment of the measurement vector
standard_deviation	Standard deviation of the measurement vector

skewness	Skewness of the measurement vector
kurtosis	Kurtosis of the measurement vector
median	Median of the measurement vector
p5_quantile	5 percent quantile of the measurement vector
p95_quantile	95 percent quantile of the measurement vector
p25_quantile	25 percent quantile of the measurement vector
p75_quantile	75 percent quantile of the measurement vector
numofbars	Number of bars of the histogram
...	Elements of the histogram

Author(s)

Andre Duesterhus

See Also

qat_analyse_distribution_1d, qat_plot_distribution_1d

Examples

```
vec <- array(rnorm(500),c(25,20))
result <- qat_analyse_distribution_2d(vec, 10)
```

---

qat_analyse_histogram_test_1d

Perform a histogram test with a given metric

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using a given metric.

Usage

```
qat_analyse_histogram_test_1d(measurement_vector,
co_measurement_vector=measurement_vector, metric="EMD", blocksize=100, numofbars=65,
factorofbar=100)
```

Arguments

measurement_vector	The measurement vector, which should be tested.
co_measurement_vector	An optional second measurement vector, which is compared to the first. The default is the first measurement vector.
metric	Metric of the comparison. Details see below.
blocksize	Number of elements, which should be used for each block.
numofbars	Number of bins of the histogram.
factorofbar	Correction factor for non-value bins.

Details

The field will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison one of the following five options are usable: EMD: Earth Mover's Distance (default); KLD: Kullback-Leibler Distance; JSD: Jenson-Shannon Distance; RMS: Root Mean Square; MS: Mean Square. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- vec[51:100, ] + 2
result <- qat_analyse_histogram_test_2d(vec, metric="EMD", blocksize=4, numofbars=65)
qat_plot_histogram_test(result$field, "test_emd_2d", result$blocksize, result$numofbars,
"emd", result$runs)
```

`qat_analyse_histogram_test_2d`

Perform a histogram test with a given metric

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using a given metric.

Usage

```
qat_analyse_histogram_test_2d(measurement_vector, co_measurement_vector=
measurement_vector, metric="EMD", blocksize=100, numofbars=65, factorofbar=100)
```

Arguments

`measurement_vector`	The measurement vector (2d array), which should be tested.
`co_measurement_vector`	An optional second measurement vector (2d array), which is compared to the frst. The default is the first measurement vector.
`metric`	Metric of the comparison. Details see below.
`blocksize`	Number of elements in the first dimension, which should be used for each block.
`numofbars`	Number of bins of the histogram.
`factorofbar`	Correction factor for non-value bins.

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison one of the following five options are usable: EMD: Earth Mover's Distance (default); KLD: Kullback-Leibler Distance; JSD: Jenson-Shannon Distance; RMS: Root Mean Square; MS: Mean Square. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

`field`	Result matrix of the comparison.
`blocksize`	Size of blocks in the first dimension.
`numofbars`	Number of bins of the used histograms.
`metric`	Used metric in the comparisons.
`runs`	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_1d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- vec[51:100, ] + 2
result <- qat_analyse_histogram_test_2d(vec, metric="EMD", blocksize=4, numofbars=65)
qat_plot_histogram_test(result$field, "test_emd_2d", result$blocksize, result$numofbars,
"emd", result$runs)
```

---

qat_analyse_histogram_test_emd_1d

Perform a histogram test with the metric EMD

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Earth Movers Distance.

Usage

```
qat_analyse_histogram_test_emd_1d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The vector will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Earth Movers Distance is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_emd_2d, qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_jsd_1
qat_analyse_histogram_test_rms_1d, qat_analyse_histogram_test_ms_1d

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
result <- qat_analyse_histogram_test_emd_1d(vec, 50, 65)
qat_plot_histogram_test(result$field, "test_emd_1d", result$blocksize,
result$numofbars, "emd", result$runs)
```

qat_analyse_histogram_test_emd_2d

Perform a histogram test with the metric EMD

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Earth Movers Distance.

Usage

```
qat_analyse_histogram_test_emd_2d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Earth Movers Distance is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_emd_1d, qat_analyse_histogram_test_kld_2d, qat_analyse_histogram_test_jsd_2
qat_analyse_histogram_test_rms_2d, qat_analyse_histogram_test_ms_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- vec[51:100, ] + 1
result <- qat_analyse_histogram_test_emd_2d(vec, 4, 65)
qat_plot_histogram_test(result$field, "test_emd_2d", result$blocksize,
result$numofbars, "emd", result$runs)
```

---

qat_analyse_histogram_test_jsd_1d

Perform a histogram test with the metric JSD

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Jenson-Shannon Divergence.

Usage

```
qat_analyse_histogram_test_jsd_1d(measurement_vector, blocksize, numofbars,
factorofbar)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram
factorofbar	Correction factor for non-value bins

Details

The vector will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Jenson-Shannon Divergence is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
factorofbar	Correction factor used for the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_jsd_2d, qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_rms_1
qat_analyse_histogram_test_ms_1d, qat_analyse_histogram_test_emd_1d

Examples

```
vec <- c(rnorm(1000), round(rnorm(1000)))
result <- qat_analyse_histogram_test_jsd_1d(vec, 50, 65, 100)
qat_plot_histogram_test(result$field, "test_jsd_1d", result$blocksize, result$numofbars,
result$factorofbar, "jsd", result$runs)
```

---

qat_analyse_histogram_test_jsd_2d

Perform a histogram test with the metric JSD

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Jenson-Shannon Divergence.

Usage

```
qat_analyse_histogram_test_jsd_2d(measurement_vector, blocksize, numofbars,
factorofbar)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram
factorofbar	Correction factor for non-value bins

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Jenson-Shannon Divergence is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
factorofbar	Correction factor used for the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_jsd_1d, qat_analyse_histogram_test_kld_2d, qat_analyse_histogram_test_rms_2
qat_analyse_histogram_test_ms_2d, qat_analyse_histogram_test_emd_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- round(vec[51:100, ])
result <- qat_analyse_histogram_test_jsd_2d(vec, 4, 65, 100)
qat_plot_histogram_test(result$field, "test_jsd_2d", result$blocksize,
result$numofbars, result$factorofbar, "jsd", result$runs)
```

qat_analyse_histogram_test_kld_1d *Perform a histogram test with the metric KLD*

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Kullback-Leibler Divergence.

Usage

```
qat_analyse_histogram_test_kld_1d(measurement_vector, blocksize =
floor(length(measurement_vector)/20), numofbars = 65, factorofbar = 100)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram
factorofbar	Correction factor for non-value bins

Details

The vector will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Kullback-Leibler Divergence is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
factorofbar	Correction factor used for the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_kld_2d, qat_analyse_histogram_test_jsd_1d, qat_analyse_histogram_test_rms_1
qat_analyse_histogram_test_ms_1d, qat_analyse_histogram_test_emd_1d

Examples

```
vec <- c(rnorm(1000), round(rnorm(1000)))
result <- qat_analyse_histogram_test_kld_1d(vec, 50, 65, 100)
qat_plot_histogram_test(result$field, "test_kld_1d", result$blocksize,
result$numofbars, result$factorofbar, "kld", result$runs)
```

---

qat_analyse_histogram_test_kld_2d

Perform a histogram test with the metric KLD

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Kullback-Leibler Divergence.

Usage

```
qat_analyse_histogram_test_kld_2d(measurement_vector, blocksize =
floor(length(measurement_vector)/20), numofbars = 65, factorofbar = 100)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram
factorofbar	Correction factor for non-value bins

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Kullback-Leibler Divergence is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

`field`	Result matrix of the comparison.
`blocksize`	Size of blocks in the first dimension.
`numofbars`	Number of bins of the used histograms.
`factorofbar`	Correction factor used for the used histograms.
`metric`	Used metric in the comparisons.
`runs`	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_jsd_2d, qat_analyse_histogram_test_rms_2
qat_analyse_histogram_test_ms_2d, qat_analyse_histogram_test_emd_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- round(vec[51:100, ])
result <- qat_analyse_histogram_test_kld_2d(vec, 4, 65, 100)
qat_plot_histogram_test(result$field, "test_kld_2d", result$blocksize,
result$numofbars, result$factorofbar, "kld", result$runs)
```

---

qat_analyse_histogram_test_ms_1d

Perform a histogram test with the metric MS

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Mean Square.

Usage

```
qat_analyse_histogram_test_ms_1d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The vector will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Mean Square is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_ms_2d, qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_jsd_1d
qat_analyse_histogram_test_rms_1d, qat_analyse_histogram_test_emd_1d

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+2)
result <- qat_analyse_histogram_test_ms_1d(vec, 50, 65)
qat_plot_histogram_test(result$field, "test_ms_1d", result$blocksize,
result$numofbars, "ms", result$runs)
```

qat_analyse_histogram_test_ms_2d *Perform a histogram test with the metric MS*

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Mean Square.

Usage

```
qat_analyse_histogram_test_ms_2d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Mean Square is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_ms_1d, qat_analyse_histogram_test_kld_2d, qat_analyse_histogram_test_jsd_2d
qat_analyse_histogram_test_rms_2d, qat_analyse_histogram_test_emd_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- vec[51:100, ] + 2
result <- qat_analyse_histogram_test_ms_2d(vec, 4, 65)
qat_plot_histogram_test(result$field, "test_ms_2d", result$blocksize,
result$numofbars, "ms", result$runs)
```

---

qat_analyse_histogram_test_rms_1d

Perform a histogram test with the metric RMS

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Root-Mean Square.

Usage

```
qat_analyse_histogram_test_rms_1d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The vector will be divided into blocks, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Root-Mean Square is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_rms_2d, qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_jsd_1
qat_analyse_histogram_test_ms_1d, qat_analyse_histogram_test_emd_1d

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+2)
result <- qat_analyse_histogram_test_rms_1d(vec, 50, 65)
qat_plot_histogram_test(result$field, "test_rms_1d", result$blocksize,
result$numofbars, "rms", result$runs)
```

---

qat_analyse_histogram_test_rms_2d

Perform a histogram test with the metric RMS

Description

This check divides the data into blocks, estimates their probability density functions by histograms and compares them by using the Root-Mean Square.

Usage

```
qat_analyse_histogram_test_rms_2d(measurement_vector, blocksize, numofbars)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Number of elements in the first dimension, which should be used for each block
numofbars	Number of bins of the histogram

Details

The field will be divided into blocks in the first dimension, with a length given by the parameter blocksize. From these blocks histograms are computed and afterwards compared. As a metric for the comparison the Root-Mean Square is used. As a result a field is generated, which includes the results of the comparison between every combination of blocks.

Value

It returns a list with the following entries:

field	Result matrix of the comparison.
blocksize	Size of blocks in the first dimension.
numofbars	Number of bins of the used histograms.
metric	Used metric in the comparisons.
runs	Number of blocks, which are compared.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_rms_1d, qat_analyse_histogram_test_kld_2d, qat_analyse_histogram_test_jsd_2
qat_analyse_histogram_test_ms_2d, qat_analyse_histogram_test_emd_2d

Examples

```
vec <- array(rnorm(1000), c(100, 20))
vec[51:100, ] <- vec[51:100, ] + 2
result <- qat_analyse_histogram_test_rms_2d(vec, 4, 65)
qat_plot_histogram_test(result$field, "test_rms_2d", result$blocksize,
result$numofbars, "rms", result$runs)
```

---

qat_analyse_lim_rule_dynamic_1d

Perform a dynamic lim-rule-check

Description

This check tests data on whether it exceeds a dynamic threshold.

Usage

```
qat_analyse_lim_rule_dynamic_1d(measurement_vector, min_vector = NULL,
max_vector = NULL, min_vector_name = NULL, max_vector_name = NULL,
min_vector_identifier = NULL, max_vector_identifier = NULL)
```

Arguments

`measurement_vector`
: The measurement vector, which should be tested

`min_vector`
: A vector which consists of the minimum threshold values, with the same dimension like the measurement vector

`max_vector`
: A vector which consists of the maximum threshold values, with the same dimension like the measurement vector

`min_vector_name`
: A name or title of the minimum vector, which will be given back in the result

`max_vector_name`
: A name or title of the maximum vector, which will be given back in the result

`max_vector_identifier`
: The identifier of the maximum vector

`min_vector_identifier`
: The identifier of the minimum vector

Details

This tests tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the minimum or maximum-vector are considered as not existing. There is no checking, if the maximum-vector is greater than the minimum-vector.

Value

It returns a list with the following entries:

`flagvector`
: A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend.

`min_vector`
: Give back the given min_vector

`max_vector`
: Give back the given max_vector

`min_vector_name`
: Give back the given min_vector_name

`max_vector_name`
: Give back the given max_vector_name

Warning

There is no checking, if the maximum-vector is greater than the minimum-vector.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_lim_rule_dynamic_1d, qat_call_lim_rule, qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_s

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- qat_analyse_lim_rule_dynamic_1d(vec, min_vector, max_vector,
min_vector_name="minimum vector", max_vector_name="maximum vector")
```

---

qat_analyse_lim_rule_dynamic_2d

Perform a dynamic lim-rule-check

Description

This check tests data on whether it exceeds a dynamic threshold.

Usage

```
qat_analyse_lim_rule_dynamic_2d(measurement_vector, min_vector = NULL,
max_vector = NULL, min_vector_name = NULL, max_vector_name = NULL,
min_vector_identifier = NULL, max_vector_identifier = NULL)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
min_vector	A 2d array which consists of the minimum threshold values, with the same dimensions like the measurement vector
max_vector	A 2d array which consists of the maximum threshold values, with the same dimensions like the measurement vector
min_vector_name	A name or title of the minimum vector, which will be given back in the result
max_vector_name	A name or title of the maximum vector, which will be given back in the result
max_vector_identifier	The identifier of the maximum vector
min_vector_identifier	The identifier of the minimum vector

Details

This tests tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector (2d array) and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the minimum or maximum-vector are considered as not existing. There is no checking, if the maximum-vector is greater than the minimum-vector.

Value

It returns a list with the following entries:

`flagvector`	A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend.
`min_vector`	Give back the given min_vector
`max_vector`	Give back the given max_vector
`min_vector_name`	Give back the given min_vector_name
`max_vector_name`	Give back the given max_vector_name

Warning

There is no checking, if the maximum-vector is greater than the minimum-vector.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

`qat_analyse_lim_rule_dynamic_1d`, `qat_plot_lim_rule_dynamic_2d`, `qat_call_lim_rule`, `qat_analyse_lim_rule_static_2d`, `qat_analyse_lim_rule_sigma_2d`

Examples

```
vec <- array(rnorm(100),c(5,20))
min_vector<-array(rnorm(100)-2,c(5,20))
max_vector<-array(rnorm(100)+2,c(5,20))
result <- qat_analyse_lim_rule_dynamic_2d(vec, min_vector, max_vector,
min_vector_name="minimum vector", max_vector_name="maximum vector")
```

qat_analyse_lim_rule_sigma_1d
Perform a sigma lim-rule-check

Description

This check tests data on whether it exceeds a threshold formed by multiple standard derviations away from the mean.

Usage

```
qat_analyse_lim_rule_sigma_1d(measurement_vector, sigma_factor)
```

Arguments

measurement_vector	The measurement vector, which should be tested
sigma_factor	Multiplier of standard derivation, which determin the maximum allowed deviation from the mean

Details

First the mean and the standard derivation of the measurement vector will be calculated. After this the limits will be determined by

$$lim_{\pm} = \mu \pm f\sigma,$$

where f is the given sigma factor.

Value

It returns a list with the following entries:

flagvector	A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happend.
sigma_factor	Give back the given sigma_factor
meanofvector	Give back the calculated mean of the measurement vector
sdofvector	Give back the calculated standard deviation of the measurement vector

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_lim_rule_sigma_1d, qat_call_lim_rule, qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_dyn

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_sigma_1d(vec, 2)
```

---

qat_analyse_lim_rule_sigma_2d

Perform a sigma lim-rule-check

Description

This check tests data on whether it exceeds a threshold formed by multiple standard derviations away from the mean.

Usage

```
qat_analyse_lim_rule_sigma_2d(measurement_vector, sigma_factor)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
sigma_factor	Multiplier of standard derivation, which determin the maximum allowed deviation from the mean

Details

First the mean and the standard derivation of the measurement vector will be calculated. After this the limits will be determined by

$$lim_{\pm} = \mu \pm f\sigma,$$

where f is the given sigma factor.

Value

It returns a list with the following entries:

flagvector	A vector of length of measurement vector. For every element of the measurement vector the flagvector (2d array) contains a -1, if its exceeding its dedicated minimum vector element, a 1, if its exceeding its dedicated maximum vector element, or a 0, when no exceeding has happenned.
sigma_factor	Give back the given sigma_factor
meanofvector	Give back the calculated mean of the measurement vector
sdofvector	Give back the calculated standard deviation of the measurement vector

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_analyse_lim_rule_sigma_1d, qat_plot_lim_rule_sigma_2d, qat_call_lim_rule, qat_analyse_lim_rule_stat
qat_analyse_lim_rule_dynamic_2d

Examples

```
vec <- array(rnorm(100),c(5,20))
result <- qat_analyse_lim_rule_sigma_2d(vec, 2)
```

---

qat_analyse_lim_rule_static_1d

Perform a static lim-rule-check

Description

This check tests data on whether it exceeds a static threshold.

Usage

```
qat_analyse_lim_rule_static_1d(measurement_vector, min_value, max_value)
```

Arguments

measurement_vector	The measurement vector, which should be tested
min_value	The minimum threshold
max_value	The maximum threshold

Details

This check tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend.

Value

It returns a list with the following entries:

flagvector	A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend.
min_value	Give back the given min_value
max_value	Give back the given max_value

Warning

There is no checking, if the maximum-value is greater than the minimum-value.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_lim_rule_static_1d, qat_call_lim_rule, qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_s

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_static_1d(vec, -2,2)
```

---

qat_analyse_lim_rule_static_2d

Perform a static lim-rule-check

Description

This check tests data on whether it exceeds a static threshold.

Usage

```
qat_analyse_lim_rule_static_2d(measurement_vector, min_value, max_value)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
min_value	The minimum threshold
max_value	The maximum threshold

Details

This check tests every element, on whether it exceeds the minimum or maximum threshold. The result will be given back as a list, which contains the result of the test as a flagvector (2d array) and its parameters. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend.

Value

It returns a list with the following entries:

`flagvector`	A vector of length of measurement vector. For every element of the measurement vector the flagvector contains a -1, if its exceeding the minimum value, a 1, if its exceeding the maximum value, or a 0, when no exceeding has happend.
`min_value`	Give back the given min_value
`max_value`	Give back the given max_value

Warning

There is no checking, if the maximum-value is greater than the minimum-value.

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_analyse_lim_rule_static_1d, qat_plot_lim_rule_static_2d, qat_call_lim_rule, qat_analyse_lim_rule_dy
qat_analyse_lim_rule_sigma_2d

Examples

```
vec <- array(rnorm(100),c(5,20))
result <- qat_analyse_lim_rule_static_2d(vec, -2,2)
```

qat_analyse_noc_rule_1d *Perform a noc-rule-check*

Description

This check tests data on whether it changes after a given amount of values.

Usage

```
qat_analyse_noc_rule_1d(measurement_vector, max_return_elements)
```

Arguments

measurement_vector	The measurement vector, which should be tested
max_return_elements	Number of coherent elements, which are allowed to have no change between the single values, without indicate an error

Details

This check tests the given measurement vector from the beginning to the end, on how much values in a row got the same value. If the number of values, which is defined by max_return_elements prior to the actual element got the same value as the actual element, the resulting flagvector will be set to 1 on the actual position. Else it will be set to 0.

Value

It returns a list with the following entries:

flagvector	flagvektor with the dimension of measurement vector, where a 0 indicates no error and a 1 that there is a repetition error
max_return_elements	Give back the given max_return_elements

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_noc_rule_1d, qat_call_noc_rule

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- qat_analyse_noc_rule_1d(vec, 1)
```

---

qat_analyse_noc_rule_2d

Perform a noc-rule-check

Description

This check tests data on whether it changes after a given amount of values.

Usage

```
qat_analyse_noc_rule_2d(measurement_vector, max_return_elements)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
max_return_elements	Number of coherent elements, which are allowed to have no change between the single values, without indicate an error

Details

This check tests the given measurement vector (2d array) in direction of the first dimension, on how much values in a row gut the same value. If the number of values, which is defined by max_return_elements prior to the actual element got the same value as the actual element, the resulting flagvector will be set to 1 on the actual position. Else it will be set to 0.

Value

It returns a list with the following entries:

flagvector	flagvektor (2d array) with the dimension of measurement vector, where a 0 indicates no error and a 1 that there is a repetition error
max_return_elements	Give back the given max_return_elements

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_noc_rule_1d, qat_call_noc_rule

Examples

```
vec <- array(c(1,1,1,2,2),c(5,20))
result <- qat_analyse_noc_rule_2d(vec, 2)
```

---

qat_analyse_roc_rule_dynamic_1d
Perform a dynamic roc-rule-check

Description

This check tests data on whether the change between two consecutive data points exceeds a dynamic threshold.

Usage

```
qat_analyse_roc_rule_dynamic_1d(measurement_vector, max_upward_vector = NULL,
max_downward_vector = NULL, upward_vector_name = NULL, downward_vector_name = NULL,
upward_vector_identifier = NULL, downward_vector_identifier = NULL)
```

Arguments

measurement_vector
: The measurement vector, which should be tested

max_upward_vector
: A vector which consists of the threshold values for upward changes, with the same dimension like the measurement vector

max_downward_vector
: A vector which consists of the threshold values for downward changes, with the same dimension like the measurement vector and have to be positive definite

upward_vector_name
: A name or title of the upward vector, which will be given back in the result

downward_vector_name
: A name or title of the downward vector, which will be given back in the result

upward_vector_identifier
: The identifier of the upward vector

downward_vector_identifier
: The identifier of the downward vector

Details

This check tests two consecutive elements, on wether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happend. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the upward or downward-vector are considered as not existing.

Value

It returns a list with the following entries:

`flagvector` A vector of length of measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happend.

`max_upward_vector`
Give back the given max_upward_vector

`max_downward_vector`
Give back the given max_downward_vector

`upward_vector_name`
Give back the given upward_vector_name

`downward_vector_name`
Give back the given downward_vector_name

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_roc_rule_dynamic_1d, qat_call_roc_rule, qat_analyse_roc_rule_static_1d

Examples

```
vec <- rnorm(100)
min_vector<-seq(1,2,length.out=100)
max_vector<-seq(1,2,length.out=100)
result <- qat_analyse_roc_rule_dynamic_1d(vec, min_vector, max_vector,
upward_vector_name="upward vector", downward_vector_name="downward vector")
```

`qat_analyse_roc_rule_dynamic_2d` *Perform a dynamic roc-rule-check*

Description

This check tests data on whether the change between two consecutive data points exceeds a dynamic threshold.

Usage

```
qat_analyse_roc_rule_dynamic_2d(measurement_vector, max_upward_vector = NULL,
max_downward_vector = NULL, upward_vector_name = NULL, downward_vector_name = NULL,
upward_vector_identifier = NULL, downward_vector_identifier = NULL)
```

Arguments

`measurement_vector`
: The measurement vector (2d array), which should be tested

`max_upward_vector`
: A vector (2d array) which consists of the threshold values for upward changes, with the same dimensions like the measurement vector

`max_downward_vector`
: A vector (2d array) which consists of the threshold values for downward changes, with the same dimension like the measurement vector and have to be positive definite

`upward_vector_name`
: A name or title of the upward vector, which will be given back in the result

`downward_vector_name`
: A name or title of the downward vector, which will be given back in the result

`upward_vector_identifier`
: The identifier of the upward vector

`downward_vector_identifier`
: The identifier of the downward vector

Details

This check tests two consecutive elements (in the direction of the first dimension), on wether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happend. NaN-values in the measurement vector will be given back as a 0 in the flagvector, NaN-values in the upward or downward-vector are considered as not existing.

Value

It returns a list with the following entries:

`flagvector` A 2d array with the dimensions of the measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding its dedicated downward vector element, a 1, if its exceeding its dedicated upward vector element, or a 0, when no exceeding has happend.

`max_upward_vector`
Give back the given max_upward_vector

`max_downward_vector`
Give back the given max_downward_vector

`upward_vector_name`
Give back the given upward_vector_name

`downward_vector_name`
Give back the given downward_vector_name

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

`qat_analyse_roc_rule_dynamic_1d`, `qat_plot_roc_rule_dynamic_2d`, `qat_call_roc_rule`, `qat_analyse_roc_rule_static_2d`

Examples

```
vec <- array(rnorm(100),c(5,20))
min_vector<-array(rnorm(100)+2,c(5,20))
max_vector<-array(rnorm(100)+2,c(5,20))
result <- qat_analyse_roc_rule_dynamic_2d(vec, min_vector, max_vector, upward_vector_name=
"upward vector", downward_vector_name="downward vector")
```

---

qat_analyse_roc_rule_static_1d

Perform a static roc-rule-check

Description

This check tests data on whether the change between two consecutive data points exceeds a static threshold.

Usage

```
qat_analyse_roc_rule_static_1d(measurement_vector, max_upward_value,
max_downward_value)
```

Arguments

measurement_vector	The measurement vector, which should be tested
max_upward_value	The upward threshold
max_downward_value	The downward threshold, which should be positive definite

Details

This check tests two consecutive elements, on wether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happend.

Value

It returns a list with the following entries:

flagvector	A vector of length of measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happend.
max_upward_value	Give back the given max_upward_value
max_downward_value	Give back the given max_downward_value

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_plot_roc_rule_static_1d, qat_call_roc_rule, qat_analyse_roc_rule_dynamic_1d

Examples

```
vec <- rnorm(100)
result <- qat_analyse_roc_rule_static_1d(vec, 2,2)
```

`qat_analyse_roc_rule_static_2d`

Perform a static roc-rule-check

Description

This check tests data on whether the change between two consecutive data points exceeds a static threshold.

Usage

```
qat_analyse_roc_rule_static_2d(measurement_vector, max_upward_value,
max_downward_value)
```

Arguments

`measurement_vector`
: The measurement vector (2d array), which should be tested

`max_upward_value`
: The upward threshold

`max_downward_value`
: The downward threshold, which should be positive definite

Details

This check tests two consecutive elements (in the direction of the first dimension), on wether the change of values between those two exceeds the upward or downward threshold. The result will be given back as a list, which contains the result of the test as a flagvector and its parameters. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happend.

Value

It returns a list with the following entries:

`flagvector`
: A vector with the dimensions of the measurement vector. For every change between two elements of the measurement vector the flagvector contains a -1, if its exceeding the downward value, a 1, if its exceeding the upward value, or a 0, when no exceeding has happend.

`max_upward_value`
: Give back the given max_upward_value

`max_downward_value`
: Give back the given max_downward_value

Author(s)

Andre Duesterhus

References

Meek, D.W., Hatfield, J.L. (1994) Data quality checking for single station meteorological databases, _Agricultural and Forest Meteorology_, *69* (1-2), 85-109.

See Also

qat_analyse_roc_rule_static_1d, qat_plot_roc_rule_static_2d, qat_call_roc_rule, qat_analyse_roc_rule_dy

Examples

```
vec <- array(rnorm(100),c(5,20))
result <- qat_analyse_roc_rule_static_2d(vec, 2,2)
```

---

qat_analyse_set_addup_1d

Addup values of a vector

Description

This function adds up sucessive values of a given vector

Usage

```
qat_analyse_set_addup_1d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements, which should be added up

Details

Starting with the first element the measurement vector will be splitted up into blocks of the size of the parameter block size. In a second step the elements of these blocks will be summed up. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_mean_1d, qat_analyse_set_nans_1d

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_addup_1d(vec, 3)
```

---

qat_analyse_set_addup_2d

Addup values of a vector

Description

This function adds up successive values of a given vector

Usage

```
qat_analyse_set_addup_2d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements, which should be added up

Details

Starting with the first element the measurement vector will be split up into blocks of the size of the parameter block size. In a second step the elements of these blocks will be summed up. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_mean_2d, qat_analyse_set_nans_2d

Examples

```
vec <- array(rnorm(144), c(12,12))
result <- qat_analyse_set_addup_2d(vec, 3)
```

qat_analyse_set_mean_1d *Mean of values of a vector*

Description

This function makes a mean of sucessive values of a given vector.

Usage

```
qat_analyse_set_mean_1d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements, which should be added up

Details

Starting with the first element the measurement vector will be splitted up into blocks of the size of the parameter block size. In a second step a mean will be formed with the elements of these blocks. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_mean_1d(vec, 3)
```

qat_analyse_set_mean_2d *Mean of values of a vector*

Description

This function makes a mean of successive values of a given vector.

Usage

```
qat_analyse_set_mean_2d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector, which should be tested
blocksize	Number of elements, which should be added up

Details

Starting with the first element the measurement vector will be split up into blocks of the size of the parameter block size. In a second step a mean will be formed with the elements of these blocks. If the last block haven't the size of block size, this block will be ignored.

Value

Give back a list, which includes the vector with the results of the blocks.

Author(s)

Andre Duesterhus

Examples

```
vec <- array(rnorm(144), c(12,12))
result <- qat_analyse_set_mean_2d(vec, 3)
```

qat_analyse_set_nans_1d *Set given values of a vector to NaN*

Description

This function set a specified value of a vector to NaN.

Usage

```
qat_analyse_set_nans_1d(measurement_vector, nan_value)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_value	Value, which should be replaced by NaN

Details

In the given measurement vector, the value, which is specified by nan_value, will be replaced by NaN.

Value

Retrun a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_1d, qat_analyse_set_mean_1d

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_1d(vec, 4)
```

qat_analyse_set_nans_2d *Set given values of a vector to NaN*

Description

This function set a specified value of a vector to NaN.

Usage

```
qat_analyse_set_nans_2d(measurement_vector, nan_value)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_value	Value, which should be replaced by NaN

Details

In the given measurement vector, the value, which is specified by nan_value, will be replaced by NaN.

Value

Retrun a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_2d, qat_analyse_set_mean_2d

Examples

```
vec <- array(c(1,2,3,4,5,4,3,2,1), c(3,3))
result <- qat_analyse_set_nans_2d(vec, 4)
```

qat_analyse_set_nans_above_1d *Set values above threshold to NaN*

Description

This function set a values of a vector above a given value to NaN.

Usage

```
qat_analyse_set_nans_above_1d(measurement_vector, nan_above)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_above	Value, above the values should be replaced by NaN

Details

In the given measurement vector, the values, which are above nan_above, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_1d, qat_analyse_set_mean_1d, qat_analyse_set_nans_1d, qat_analyse_set_nans_below_

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_above_1d(vec, 4)
```

qat_analyse_set_nans_above_2d *Set values above threshold to NaN*

Description

This function set a values of a vector above a given value to NaN.

Usage

```
qat_analyse_set_nans_above_2d(measurement_vector, nan_above)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_above	Value, above the values should be replaced by NaN

Details

In the given measurement vector, the values, which are above nan_above, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_2d, qat_analyse_set_mean_2d, qat_analyse_set_nans_2d, qat_analyse_set_nans_below_

Examples

```
vec <- array(c(1,2,3,4,5,4,3,2,1), c(3,3))
result <- qat_analyse_set_nans_above_2d(vec, 4)
```

qat_analyse_set_nans_below_1d *Set values below threshold to NaN*

Description

This function set a values of a vector below a given value to NaN.

Usage

```
qat_analyse_set_nans_below_1d(measurement_vector, nan_below)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_below	Value, below the values should be replaced by NaN

Details

In the given measurement vector, the values, which are below nan_below, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_1d, qat_analyse_set_mean_1d, qat_analyse_set_nans_1d, qat_analyse_set_nans_above_

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
result <- qat_analyse_set_nans_below_1d(vec, 4)
```

qat_analyse_set_nans_below_2d *Set values below threshold to NaN*

Description

This function set a values of a vector below a given value to NaN.

Usage

```
qat_analyse_set_nans_below_2d(measurement_vector, nan_below)
```

Arguments

measurement_vector	The measurement vector, which should be worked on
nan_below	Value, below the values should be replaced by NaN

Details

In the given measurement vector, the values, which are below nan_below, will be replaced by NaN.

Value

Return a list, which includes the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_2d, qat_analyse_set_mean_2d, qat_analyse_set_nans_2d, qat_analyse_set_nans_above_

Examples

```
vec <- array(c(1,2,3,4,5,4,3,2,1), c(3,3))
result <- qat_analyse_set_nans_below_2d(vec, 4)
```

`qat_analyse_slide_distribution_1d` *Perform a slide distribution check*

Description

The measurement vector will be scanned stepwise by a sliding window, and on every step some statistical parameters will be calculated.

Usage

```
qat_analyse_slide_distribution_1d(measurement_vector, blocksize)
```

Arguments

`measurement_vector`	The measurement vector, which should be tested
`blocksize`	Length of the sliding window

Details

The measurement vector will be scanned stepwise by a sliding window, which got a length of the given parameter blocksize. At every step some statistical parameters will be calculated for the actual window. As a result a list will be given back, with these parameters, where every entry got a length of the length of the measurement vector minus the blocksize plus one.

Value

It returns a list with the following entries:

`first_moment`	First moment of the measurement vector
`second_moment`	Second moment of the measurement vector
`third_moment`	Third moment of the measurement vector
`fourth_moment`	Fourth moment of the measurement vector
`standard_deviation`	Standard deviation of the measurement vector
`skewness`	Skewness of the measurement vector
`kurtosis`	Kurtosis of the measurement vector
`median`	Median of the measurement vector
`p5_quantile`	5 percent quantile of the measurement vector
`p95_quantile`	95 percent quantile of the measurement vector
`p25_quantile`	25 percent quantile of the measurement vector
`p75_quantile`	75 percent quantile of the measurement vector
`blocksize`	Length of the used blocks

Author(s)

Andre Duesterhus

See Also

qat_plot_slide_distribution_1d

Examples

```
vec <- rnorm(100)
result <- qat_analyse_slide_distribution_1d(vec, 10)
```

---

qat_analyse_slide_distribution_2d

Perform a slide distribution check

Description

The measurement vector will be scanned stepwise by a sliding window, and on every step some statistical parameters will be calculated.

Usage

```
qat_analyse_slide_distribution_2d(measurement_vector, blocksize)
```

Arguments

measurement_vector	The measurement vector (2d array), which should be tested
blocksize	Length of the sliding window

Details

The measurement vector will be scanned stepwise by a sliding window for each element of the second dimension, which got a length of the given parameter blocksize. At every step some statistical parameters will be calculated for the actual window. As a result a list will be given back, with these parameters, where every entry got the same dimension like the measurement vector, where the first dimension is reduced by the blocksize plus one.

Value

It returns a list with the following entries:

first_moment	First moment of the measurement vector
second_moment	Second moment of the measurement vector
third_moment	Third moment of the measurement vector
fourth_moment	Fourth moment of the measurement vector

`standard_deviation`
: Standard deviation of the measurement vector

`skewness`
: Skewness of the measurement vector

`kurtosis`
: Kurtosis of the measurement vector

`median`
: Median of the measurement vector

`p5_quantile`
: 5 percent quantile of the measurement vector

`p95_quantile`
: 95 percent quantile of the measurement vector

`p25_quantile`
: 25 percent quantile of the measurement vector

`p75_quantile`
: 75 percent quantile of the measurement vector

`blocksize`
: Length of the used blocks

Author(s)

Andre Duesterhus

See Also

`qat_analyse_slide_distribution_1d`, `qat_plot_slide_distribution_2d`

Examples

```
vec <- array(rnorm(100),c(25,20))
result <- qat_analyse_slide_distribution_2d(vec, 5)
```

---

qat_analyse_trimmed_distribution_1d

Perform a trimmed distribution check

Description

The measurement vector will be trimmed at each side stepwise and at every step some statistical parameters will be calculated.

Usage

```
qat_analyse_trimmed_distribution_1d(measurement_vector)
```

Arguments

`measurement_vector`
: The measurement vector, which should be tested

Details

The measurement vector will be trimmed at each side stepwise, with a step of 1 percent. At each step some statistical parameters will be calculated. As a result a list will be given back, with these parameters, where every entry got a length of 50.

Value

It returns a list with the following entries:

`first_moment`	First moment of the measurement vector
`second_moment`	Second moment of the measurement vector
`third_moment`	Third moment of the measurement vector
`fourth_moment`	Fourth moment of the measurement vector
`standard_deviation`	Standard deviation of the measurement vector
`skewness`	Skewness of the measurement vector
`kurtosis`	Kurtosis of the measurement vector
`median`	Median of the measurement vector
`p5_quantile`	5 percent quantile of the measurement vector
`p95_quantile`	95 percent quantile of the measurement vector
`p25_quantile`	25 percent quantile of the measurement vector
`p75_quantile`	75 percent quantile of the measurement vector

Author(s)

Andre Duesterhus

See Also

`qat_plot_trimmed_distribution_1d`

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_trimmed_distribution_1d(vec)
```

---

qat_analyse_trimmed_distribution_2d

Perform a trimmed distribution check

Description

The measurement vector (2d array) will be handled separately for every element in the direction of the second dimension. Each vector will be trimmed stepwise at each side and at every step some statistical parameters will be calculated.

Usage

```
qat_analyse_trimmed_distribution_2d(measurement_vector)
```

Arguments

`measurement_vector`
: The measurement vector, which should be tested

Details

The measurement vector will be trimmed, in direction of the first dimension, at each side stepwise, with a step of 1 percent. At each step some statistical parameters will be calculated. As a result a list will be given back, with these parameters, where every entry got the first dimension of 50 and as the second the second dimension of the measurement vector.

Value

It returns a list with the following entries:

`first_moment`
: First moment of the measurement vector

`second_moment`
: Second moment of the measurement vector

`third_moment`
: Third moment of the measurement vector

`fourth_moment`
: Fourth moment of the measurement vector

`standard_deviation`
: Standard deviation of the measurement vector

`skewness`
: Skewness of the measurement vector

`kurtosis`
: Kurtosis of the measurement vector

`median`
: Median of the measurement vector

`p5_quantile`
: 5 percent quantile of the measurement vector

`p95_quantile`
: 95 percent quantile of the measurement vector

`p25_quantile`
: 25 percent quantile of the measurement vector

`p75_quantile`
: 75 percent quantile of the measurement vector

Author(s)

Andre Duesterhus

See Also

`qat_analyse_trimmed_distribution_1d`, `qat_plot_trimmed_distribution_2d`

Examples

```
vec <- array(rnorm(100),c(25,20))
result <- qat_analyse_trimmed_distribution_2d(vec)
```

qat_call_block_distribution *Perform a block distribution check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_block_distribution_1d.

Usage

```
qat_call_block_distribution(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_block_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_slide_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
result <- qat_call_block_distribution(vec, workflowlist_part)
```

---

qat_call_boot_distribution

Perform a bootstrapped distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_boot_distribution_1d.

Usage

```
qat_call_boot_distribution(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_boot_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_boot_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
result <- qat_call_boot_distribution(vec, workflowlist_part)
```

---

qat_call_distribution *Perform a distribution check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_distribution_1d.

Usage

```
qat_call_distribution(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector

lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
result <- qat_call_distribution(vec, workflowlist_part)
```

---

qat_call_histogram_test

Perform a LIM Rule Check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_histogram_test_xxx_xd.

Usage

```
qat_call_histogram_test(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_static_1d and qat_analyse_lim_rule_sigma_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_histogram_test_kld_1d, qat_analyse_histogram_test_jsd_1d, qat_analyse_histogram_test_rms_1
qat_analyse_histogram_test_ms_1d, qat_analyse_histogram_test_emd_1d, qat_analyse_histogram_test_kld_2d
qat_analyse_histogram_test_jsd_2d, qat_analyse_histogram_test_rms_2d, qat_analyse_histogram_test_ms_2d
qat_analyse_histogram_test_emd_2d

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
workflowlist_part <- list(blocksize=50, numofbars=65, metric="emd")
resultlist <- qat_call_histogram_test(vec, workflowlist_part, element=1)
savelist <- qat_call_save_histogram_test(resultlist[[2]])
```

---

qat_call_lim_rule *Perform a LIM Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_static_1d and qat_analyse_lim_rule_sigma_1d.

Usage

```
qat_call_lim_rule(measurement_vector, workflowlist_part, element = -999, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_static_1d and qat_analyse_lim_rule_sigma_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_dynamic_1d, qat_analyse_lim_rule_sigma_1d, qat_plot_lim_rule_dynamic_1d, qat_plot_lim_rule_static_1d, qat_plot_lim_rule_sigma_1d

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",
maximum_vector="vec2",minimum_vector_name="minimum vector",maximum_vector_name="maximum vector",
sigma_factor=2)
result <- qat_call_lim_rule(vec, workflowlist_part, vec1=min_vector, vec2=max_vector)
```

---

qat_call_noc_rule *Perform a NOC Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_noc_rule_1d.

Usage

```
qat_call_noc_rule(measurement_vector, workflowlist_part, element = -999, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector

lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_noc_rule_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_plot_noc_rule_1d

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
result <- qat_call_noc_rule(vec, workflowlist_part)
```

---

qat_call_plot_block_distribution

Plot a result of a block distribution check

Description

A result of qat_analyse_block_distribution_1d will be plotted.

Usage

```
qat_call_plot_block_distribution(resultlist_part, measurement_vector = NULL, time=NULL,
height= NULL, lat=NULL, lon=NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_block_distribution_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_block_distribution_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_block_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
resultlist <- qat_call_block_distribution(vec, workflowlist_part, element=1)
# this example produce the files exampleplot_1_blockdist_1.png, exampleplot_1_blockdist_2.png
# and exampleplot_1_blockdist_3.png in the current directory
qat_call_plot_block_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

qat_call_plot_boot_distribution *Plot a result of a bootstrapped distribution check*

Description

A result of qat_analyse_boot_distribution_1d will be plotted.

Usage

```
qat_call_plot_boot_distribution(resultlist_part, measurement_vector = NULL,
time = NULL, height = NULL, lat = NULL, lon = NULL, measurement_name = "",
directoryname = "", basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_boot_distribution_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_boot_distribution_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_boot_distribution_1d

Examples

```
vec <- rnorm(500)
workflowlist_part <- list(bootruns=100)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
# this example produce the file exampleplot_1_bootdist.png in the current directory
qat_call_plot_boot_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

---

qat_call_plot_distribution

Plot a result of a distribution check

Description

A result of qat_analyse_distribution_1d will be plotted.

Usage

```
qat_call_plot_distribution(resultlist_part, measurement_vector = NULL, time = NULL,
height = NULL, lat = NULL, lon = NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_distribution_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_distribution_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
resultlist <- qat_call_distribution(vec, workflowlist_part, element=1)
# this example produce a file exampleplot_1_dist.png in the current directory
qat_call_plot_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

---

qat_call_plot_histogram_test

Plot a result of a histogram test

Description

A result of qat_analyse_histogram_test_xxx_xd will be plotted.

Usage

```
qat_call_plot_histogram_test(resultlist_part, measurement_vector = NULL, time = NULL,
height = NULL, lat = NULL, lon = NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_histogram_test_xxx_xd.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_histogram_test_xxx_xd. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
workflowlist_part <- list(blocksize=50, numofbars=65, metric="emd")
resultlist <- qat_call_histogram_test(vec, workflowlist_part, element=1)
# this example produce the file exampleplot_1_histogramtest_emd.png in the current
# directory
qat_call_plot_histogram_test(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

qat_call_plot_lim_rule

Plot a result of a LIM rule check

Description

A result of qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_sigma_1d or qat_analyse_lim_rule_dynamics_1
will be plotted.

Usage

```
qat_call_plot_lim_rule(resultlist_part, measurement_vector = NULL, time = NULL,
height = NULL, lat = NULL, lon = NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_sigma_1d or qat_analyse_lim_rule_dynamics_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator in the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_lim_rule_static_1d, qat_analyse_lim_rule_sigma_1d or qat_analyse_lim_rule_dynamics_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_lim_rule_dynamic_1d, qat_plot_lim_rule_static_1d, qat_plot_lim_rule_sigma_1d

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",
maximum_vector="vec2",minimum_vector_name="minimum vector",
maximum_vector_name="maximum vector", sigma_factor=2)
resultlist <- qat_call_lim_rule(vec, workflowlist_part, element=1, vec1=min_vector,
vec2=max_vector)
# this example produce the files exampleplot_1_lim_sigma.png, exampleplot_1_lim_static.png
# and exampleplot_1_lim_dynamic.png in the current directory
for (ii in 2:4) {
qat_call_plot_lim_rule(resultlist[[ii]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
}
```

---

qat_call_plot_noc_rule

Plot a result of a NOC rule check

Description

A result of qat_analyse_noc_rule_1d will be plotted.

Usage

```
qat_call_plot_noc_rule(resultlist_part, measurement_vector = NULL, time = NULL,
height = NULL, lat = NULL, lon = NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_noc_rule_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector

lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_noc_rule_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_noc_rule_1d

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
resultlist <- qat_call_noc_rule(vec, workflowlist_part,element=1)
# this example produce a file exampleplot_1_noc.png in the current directory
qat_call_plot_noc_rule(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

---

qat_call_plot_roc_rule

Plot a result of a ROC rule check

Description

A result of qat_analyse_roc_rule_static_1d or qat_analyse_roc_rule_dynamics_1d will be plotted.

Usage

```
qat_call_plot_roc_rule(resultlist_part, measurement_vector = NULL, time = NULL,
height = NULL, lat = NULL, lon = NULL, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_roc_rule_static_1d or qat_analyse_roc_rule_dynamics_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator in the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_roc_rule_static_1d or qat_analyse_roc_rule_dynamics_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_roc_rule_dynamic_1d, qat_plot_roc_rule_static_1d

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=100)
upward_vector<-seq(1,2,length.out=100)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",
upward_vector="vec2",downward_vector_name="downward vector",
upward_vector_name="upward vector")
resultlist <- qat_call_roc_rule(vec, workflowlist_part, element=1, vec1=downward_vector,
vec2=upward_vector)
# this example produce the files exampleplot_1_roc_static.png and
# exampleplot_1_roc_dynamic.png in the current directory
for (ii in 2:3) {
qat_call_plot_roc_rule(resultlist[[ii]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
}
```

---

qat_call_plot_slide_distribution

Plot a result of a slide distribution check

Description

A result of qat_analyse_slide_distribution_1d will be plotted.

Usage

```
qat_call_plot_slide_distribution(resultlist_part, measurement_vector = NULL,
time = NULL, height = NULL, lat = NULL, lon = NULL, measurement_name = "",
directoryname = "", basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_slide_distribution_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_slide_distribution_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_slide_distribution_1d

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
resultlist <- qat_call_slide_distribution(vec, workflowlist_part, element=1)
# this example produce the files exampleplot_1_slidedist_1.png, exampleplot_1_slidedist_2.png
# and exampleplot_1_slidedist_3.png in the current directory
qat_call_plot_slide_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

---

qat_call_plot_trimmed_distribution

Plot a result of a trimmed distribution check

Description

A result of qat_analyse_trimmed_distribution_1d will be plotted.

Usage

```
qat_call_plot_trimmed_distribution(resultlist_part, measurement_vector = NULL,
time = NULL, height = NULL, lat = NULL, lon = NULL, measurement_name = "",
directoryname = "", basename = "", plotstyle = NULL)
```

Arguments

resultlist_part	A list with the result, which is directly or indirectly produced by qat_analyse_trimmed_distribution_1d.
measurement_vector	The measurement vector, which was tested
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
measurement_name	Name of the data, which will be used as an indicator on the plot
directoryname	Definition of the directory, where the plot should be stored
basename	Basic name of the resulting file
plotstyle	A list with a qat color scheme

Details

A plot will be produced, which base on the resulting list of qat_analyse_trimmed_distribution_1d. The measurement_name will be used as a title of the plot and the plotstyle list define the colors of the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory. As a filename the basename with additional information will be used (number of test and a label, which indicate which test was performed).

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_trimmed_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list()
resultlist <- qat_call_trimmed_distribution(vec, workflowlist_part, element=1)
# this example produce a file exampleplot_1_trimmeddist.png in the current directory
qat_call_plot_trimmed_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
```

qat_call_roc_rule *Perform a ROC Rule Check*

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part.
The possible called functions are qat_analyse_roc_rule_dynamic_1d and qat_analyse_roc_rule_static_1d.

Usage

```
qat_call_roc_rule(measurement_vector, workflowlist_part, element = -999, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part.
The possible called functions are qat_analyse_roc_rule_dynamic_1d and qat_analyse_roc_rule_static_1d.
As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_roc_rule_static_1d, qat_analyse_roc_rule_dynamic_1d, qat_plot_roc_rule_dynamic_1d, qat_plot_roc_rule_static_1d

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=1000)
upward_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",
upward_vector="vec2",downward_vector_name="downward vector", upward_vector_name="upward vector")
result <- qat_call_roc_rule(vec, workflowlist_part,vec1=downward_vector,vec2=upward_vector)
```

---

qat_call_save_block_distribution

Produce a savelist-entry for a Block Distribution Test

Description

This function calls qat_save_block_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_block_distribution(resultlist_part, element = -999, dim_mv=1,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3

vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_block_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_block_distribution_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(blocksize=50)
resultlist <- qat_call_block_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_block_distribution(resultlist[[2]])
```

---

qat_call_save_boot_distribution

Produce a savelist-entry for a Boot Distribution Test

Description

This function calls qat_save_boot_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_boot_distribution(resultlist_part, element = -999, dim_mv=1,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_boot_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_boot_distribution_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_boot_distribution(resultlist[[2]])
```

qat_call_save_distribution
Produce a savelist-entry for a Distribution Test

Description

This function calls qat_save_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_distribution(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_distribution_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(numofbars=15)
resultlist <- qat_call_distribution(vec, workflowlist_part, element=1)
qat_call_plot_distribution(resultlist[[2]], measurement_vector=vec,
measurement_name="Result of Check", basename="exampleplot")
savelist <- qat_call_save_distribution(resultlist[[2]])
```

---

qat_call_save_histogram_test

Produce a savelist-entry for a Histogram Test

Description

This function calls qat_save_histogram_test. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_histogram_test(resultlist_part, element = -999, dim_mv = 1,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3

vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are qat_save_histogram_test. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_histogram_test, qat_run_workflow_save

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
workflowlist_part <- list(blocksize=50, numofbars=65, metric="emd")
resultlist <- qat_call_histogram_test(vec, workflowlist_part, element=1)
savelist <- qat_call_save_histogram_test(resultlist[[2]])
```

---

qat_call_save_lim_rule

Produce a savelist-entry for a LIM-RULE Test

Description

This function calls qat_save_lim_rule_static_1d, qat_save_lim_rule_sigma_1d or qat_save_lim_rule_dynamic_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_lim_rule(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are qat_save_lim_rule_static_1d, qat_save_lim_rule_sigma_1d or qat_save_lim_rule_dynamic_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_lim_rule_static_1d, qat_save_lim_rule_sigma_1d, qat_save_lim_rule_dynamic_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(minimum_value=-2, maximum_value=2,minimum_vector="vec1",
maximum_vector="vec2",minimum_vector_name="minimum vector",
maximum_vector_name="maximum vector", sigma_factor=2)
```

```
resultlist <- qat_call_lim_rule(vec, workflowlist_part, element=1, vec1=min_vector,
vec2=max_vector)
savelist <- list()
savelistcounter <- 1
for (ii in 2:4) {
savelist <- qat_call_save_lim_rule(resultlist[[ii]], savelist=savelist,
savelistcounter=savelistcounter)
if (length(which(names(savelist)=="element"))==0) {
savelistcounter<-length(savelist)
} else {
savelistcounter<-1
}
}
```

`qat_call_save_noc_rule` *Produce a savelist-entry for a NOC RULE Test*

Description

This function calls qat_save_noc_rule_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_noc_rule(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

`resultlist_part`	A list with the results of the check
`element`	Element-identifier for the result, which will be given back in the savelist
`dim_mv`	Dimension of the measurement vector.
`time`	A vector of time elements with the length of the measurement vector
`height`	A vector of height elements with the length of the measurement vector
`lat`	A vector of latitude elements with the length of the measurement vector
`lon`	A vector of longitude elements with the length of the measurement vector
`vec1`	An additional vector, which is named as vec1
`vec2`	An additional vector, which is named as vec2
`vec3`	An additional vector, which is named as vec3
`vec4`	An additional vector, which is named as vec4
`baseunit`	The unit of the original measurement vector
`savelist`	A list with save elements
`savelistcounter`	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_noc_rule_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_noc_rule_1d, qat_run_workflow_save

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
workflowlist_part <- list(max_return_elements=1)
resultlist <- qat_call_noc_rule(vec, workflowlist_part,element=1)
savelist <- qat_call_save_noc_rule(resultlist[[2]])
```

---

qat_call_save_roc_rule

Produce a savelist-entry for a ROC-Rule Test

Description

This function calls qat_save_roc_rule_static_1d or qat_save_roc_rule_dynamic_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_roc_rule(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector

height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are qat_save_roc_rule_static_1d and qat_save_roc_rule_static_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_roc_rule_static_1d, qat_save_roc_rule_dynamic_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(100)
downward_vector<-seq(1,2,length.out=1000)
upward_vector<-seq(1,2,length.out=1000)
workflowlist_part <- list(downward_value=2, upward_value=2,downward_vector="vec1",
upward_vector="vec2",downward_vector_name="downward vector",
upward_vector_name="upward vector")
resultlist <- qat_call_roc_rule(vec, workflowlist_part, element=1,
vec1=downward_vector, vec2=upward_vector)
savelist <- list()
savelistcounter <- 1
for (ii in 2:3) {
savelist <- qat_call_save_roc_rule(resultlist[[ii]], savelist=savelist,
savelistcounter=savelistcounter)
if (length(which(names(savelist)=="element"))==0) {
savelistcounter<-length(savelist)
```

```
} else {
savelistcounter<-1
}
}
```

---

qat_call_save_set_addup

Produce a savelist-entry for a set Addup

Description

This function calls qat_save_set_addup_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_addup(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_set_addup_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_set_addup_1d, qat_run_workflow_save

Examples

```
## still to come
```

---

qat_call_save_set_mean

Produce a savelist-entry for a Set Mean

Description

This function calls qat_save_set_mean_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_mean(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3

vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_set_mean_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_set_mean_1d, qat_run_workflow_save

Examples

```
## still to come
```

---

qat_call_save_set_nans

Produce a savelist-entry for a set NAN

Description

This function calls qat_save_set_nans_1d, qat_save_set_nans_above_1d or qat_save_set_nans_below_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_set_nans(resultlist_part, element = -999, dim_mv=1, time = NULL,
height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL,
vec4 = NULL, baseunit = NULL, savelist = list(), savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called functions are qat_save_set_nans_1d, qat_save_set_nans_above_1d or qat_save_set_nans_below_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_set_nans_1d, qat_save_set_nans_above_1d, qat_save_set_nans_below_1d, qat_run_workflow_save

Examples

```
## still to come
```

qat_call_save_slide_distribution
Produce a savelist-entry for a Slide Distribution Test

Description

This function calls qat_save_slide_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_slide_distribution(resultlist_part, element = -999, dim_mv=1,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, baseunit = NULL, savelist = list(),
savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_slide_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_slide_distribution_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
resultlist <- qat_call_slide_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_slide_distribution(resultlist[[2]])
```

---

qat_call_save_trimmed_distribution

Produce a savelist-entry for a Trimmed Distribution Test

Description

This function calls qat_save_trimmed_distribution_1d. As a result a part of a savelist is constructed, which can be used to construct a netCDF file.

Usage

```
qat_call_save_trimmed_distribution(resultlist_part, element = -999, dim_mv=1,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, baseunit = NULL, savelist = list(),
savelistcounter = 1)
```

Arguments

resultlist_part	A list with the results of the check
element	Element-identifier for the result, which will be given back in the savelist
dim_mv	Dimension of the measurement vector.
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4

baseunit	The unit of the original measurement vector
savelist	A list with save elements
savelistcounter	Numbers of elements of the savelist

Details

This function calls the described saving-function, which transform the resultlist elements to a savinglist element. The possible called function is qat_save_trimmed_distribution_1d. As a result the given savelist will get an additional entry.

Value

The given savelist will be returned, with included results of the functions which may be called in this function.

Author(s)

Andre Duesterhus

See Also

qat_save_trimmed_distribution_1d, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list(bootruns=1000)
resultlist <- qat_call_boot_distribution(vec, workflowlist_part, element=1)
savelist <- qat_call_save_boot_distribution(resultlist[[2]])
```

---

qat_call_set_addup *Addup values of a vector*

Description

This function adds up sucessive values of a given vector

Usage

```
qat_call_set_addup(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described method, which are defined by the parameters in the workflowlist_part. The possible called function is qat_analyse_set_addup_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the vector with the results of the block.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_addup_1d

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(blocksize=3)
result <- qat_call_set_addup(vec, workflowlist_part)
```

qat_call_set_mean *Mean of values of a vector*

Description

This function make a mean of sucessive values of a given vector.

Usage

```
qat_call_set_mean(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described method, which are defined by the parameters in the workflowlist_part. The possible called function is qat_analyse_set_mean_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the vector with the results of the block.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_mean_1d

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(blocksize=3)
result <- qat_call_set_mean(vec, workflowlist_part)
```

---

qat_call_set_nans *Set given values of a vector to NaN*

Description

This function set a specified value of a vector to NaN.

Usage

```
qat_call_set_nans(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described method, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_set_nans_1d, qat_analyse_set_nans_above_1d and qat_analyse_set_nans_below_1d. As a result the function will give back a list, which include the corrected measurement vector.

Value

Give back a list, which include the measurement vector with the replaced values.

Author(s)

Andre Duesterhus

See Also

qat_analyse_set_nans_1d

Examples

```
vec <- c(1,2,3,4,5,4,3,2,1)
workflowlist_part <- list(nan_value=4)
result <- qat_call_set_nans(vec, workflowlist_part)
```

---

qat_call_slide_distribution

Perform a slide distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_slide_distribution_1d.

Usage

```
qat_call_slide_distribution(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector

height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_slide_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_slide_distribution_1d

Examples

```
vec <- rnorm(100)
workflowlist_part <- list(blocksize=10)
result <- qat_call_slide_distribution(vec, workflowlist_part)
```

---

qat_call_trimmed_distribution

Perform a trimmed distribution check

Description

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_trimmed_distribution_1d.

Usage

```
qat_call_trimmed_distribution(measurement_vector, workflowlist_part, element = -999,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, resultlist = list(), resultlistcounter = 1)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist_part	A list with the parameters of the check
element	Element-identifier for the result, which will be given back in the resultlist
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
resultlist	A list with results of tests
resultlistcounter	Number of elements of the resultlist

Details

This function calls the described tests, which are defined by the parameters in the workflowlist_part. The possible called functions are qat_analyse_trimmed_distribution_1d. As a result the resultlist will get additional entries, which are defined by the tests, which may called by this function.

Value

The given resultlist will be returned, with included results of the functions which may called in this function.

Author(s)

Andre Duesterhus

See Also

qat_analyse_trimmed_distribution_1d

Examples

```
vec <- rnorm(1000)
workflowlist_part <- list()
result <- qat_call_trimmed_distribution(vec, workflowlist_part)
```

qat_config_read_workflow *Read an XML workflow*

Description

This functions read a XML-workflow-file.

Usage

```
qat_config_read_workflow(filename)
```

Arguments

filename	Path and filename of the xml-file, in which the workflow is defined

Details

This functions read a file, which got a XML-workflow in it. This will be transformed to a workflowlist, which may be processed by qat_run_workflow_check.

Value

A workflowlist, which consists of the tests and its parameters, which should be performed.

Author(s)

Andre Duesterhus

See Also

qat_run_workflow_check

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
```

qat_config_write_workflow *Write a result*

Description

A workflowlist will be stored into a XML-file.

Usage

```
qat_config_write_workflow(workflowlist, name = "", description = "",
author = "", date = "", sample_time_start = "", sample_time_stop = "",
sample_place = "", config_filename = "", output_filename = "")
```

Arguments

workflowlist	A workflowlist, which may be loaded by qat_config_read_workflow and used for tests.
name	Name of the tests, which were performed with this workflowlist
description	Description of the workflowlist
author	Author who used the workflowlist for a test.
date	Date of the test.
sample_time_start	Start time of the sample, which was tested
sample_time_stop	End time of the sample, which was tested
sample_place	Location of the sample, which was tested
config_filename	A filename of the configuration file, which was read in at qat_config_read_workflow.
output_filename	Filename, where the result should be stored.

Details

The workflow will be stored at the location of output_filename. As additional information in the header of this file, the other arguments will be used.

Value

The information, which was stored, will be given back.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow

Examples

```
library("qat")
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

---

qat_data_close_ncdf *Close an open netCDF-file*

Description

An open netCDF file will be closed.

Usage

```
qat_data_close_ncdf(obj)
```

Arguments

obj	An open netCDF object.

Value

None.

Author(s)

Andre Duesterhus

See Also

qat_data_read_ncdf, qat_data_nameofvars_ncdf, qat_data_numofvars_ncdf, qat_data_varcontent_ncdf

Examples

```
#still to come
```

qat_data_nameofvars_ncdf *Variable names of ncdf object*

Description

Give back the names of the variables in a netCDF-object.

Usage

```
qat_data_nameofvars_ncdf(obj)
```

Arguments

obj	A netcdf object, which will be produced by qat_data_read_ncdf.

Details

The names of the variables, which are stored in the ncdf-object will be given back as a list.

Value

List of names.

Author(s)

Andre Duesterhus

See Also

qat_data_read_ncdf, qat_data_numofvars_ncdf, qat_data_varcontent_ncdf

Examples

```
#still to come
```

qat_data_numofvars_ncdf *Nomber of Variables of ncdf object*

Description

Give back the number of the variables in a netCDF-object.

Usage

```
qat_data_numofvars_ncdf(obj)
```

Arguments

obj	A netcdf object, which will be produced by qat_data_read_ncdf.

Details

The number of variables, which are stored in the ncdf-obect will be given back.

Value

Number of variables.

Author(s)

Andre Duesterhus

See Also

qat_data_read_ncdf, qat_data_nameofvars_ncdf qat_data_varcontent_ncdf

Examples

```
#still to come
```

qat_data_read_ncdf *Read in netCDF-file*

Description

A netCDF file will be read in and a ncdf-object will be given back.

Usage

```
qat_data_read_ncdf(filename)
```

Arguments

filename	Path and filename of the netCDF-file, which should be read in.

Value

A ncdf-Object, with the content of the file.

Author(s)

Andre Duesterhus

See Also

qat_data_nameofvars_ncdf, qat_data_numofvars_ncdf, qat_data_varcontent_ncdf

Examples

```
#still to come
```

qat_data_varcontent_ncdf *Content of a variable*

Description

Give back the content of a specified variable of an ncdf-object.

Usage

```
qat_data_varcontent_ncdf(obj, numofvar)
```

Arguments

obj	A netcdf object, which will be produced by qat_data_read_ncdf.
numofvar	Number of variable, which content should be delivered.

Details

The content of the variable, which is specified by its number in numofvars will be given back.

Value

The content of the variable.

Author(s)

Andre Duesterhus

See Also

qat_data_read_ncdf, qat_data_nameofvars_ncdf qat_data_numofvars_ncdf

Examples

```
#still to come
```

---

qat_measure_histogram_difference

Perform a comparison of two datasets by means of its histograms with a given metric

---

Description

This function compares two datasets by calculating their histograms and compares them by a given metric.

Usage

```
qat_measure_histogram_difference(data1, data2, metric="EMD", breakvector=NULL,
numofbars=65, factorofbar=100)
```

Arguments

data1	The first dataset.
data2	The second dataset.
metric	Metric of the comparison. Details see below.
breakvector	Breakvector for the histograms. When not given (NULL), an equidistant breakvector between the minimum and maximum of the two datasets with the given number of bars will be generated.
numofbars	Number of bins of the histogram, when no breakvector is given.
factorofbar	Correction factor for non-value bins.

Details

For both datasets the histograms are computed and compared by means of a given metric. As a metric for the comparison one of the following five options are usable: EMD: Earth Mover's Distance (default); KLD: Kullback-Leibler Distance; JSD: Jenson-Shannon Distance; RMS: Root Mean Square; MS: Mean Square. As a result the distance between the two histograms calculated by the metric is given.

Author(s)

Andre Duesterhus

References

Duesterhus, A., Hense, A. (2012) Advanced Information Criterion for Environmental Data Quality Assurance, _Advances in Science and Research_, *8*, 99-104.

See Also

qat_analyse_histogram_test_1d, qat_analyse_histogram_test_2d

Examples

```
vec1 <- array(rnorm(1000), c(100, 20))
vec2 <- vec1 + 1
result <- qat_measure_histogram_difference(vec1, vec2, metric="EMD", numofbars=65)
```

---

qat_plot_block_distribution_1d

Plot a block distribution check result

Description

A plot of the result of a block distribution check will be produced.

Usage

```
qat_plot_block_distribution_1d(resultlist, filename, blocksize = -1,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_block_distribution_1d
filename	Name of the file without extension.
blocksize	Length of the blocks
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_block_distribution_1d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_block_distribution_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_block_distribution_1d(vec, 50)
# this example produce a file exampleplot_blockdist.png in the current directory
qat_plot_block_distribution_1d(result$stat, "exampleplot_blockdist",
blocksize=result$blocksize, measurement_name="Result of Check")
```

---

qat_plot_block_distribution_2d

Plot a block distribution check result

Description

A plot of the result of a block distribution check will be produced.

Usage

```
qat_plot_block_distribution_2d(resultlist, filename, blocksize = -1,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_block_distribution_2d
filename	Name of the file without extension.
blocksize	Length of the blocks
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_block_distribution_2d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_block_distribution_1d

Examples

```
vec <- array(rnorm(500),c(25,20))
result <- qat_analyse_block_distribution_2d(vec, 5)
# this example produce a file exampleplot_blockdist.png in the current directory
qat_plot_block_distribution_2d(result$stat, "exampleplot_blockdist",
blocksize=result$blocksize, measurement_name="Result of Check")
```

---

qat_plot_boot_distribution_1d

Plot a bootstrapped distribution check result

Description

A plot of the result of a booted distribution check will be produced.

Usage

```
qat_plot_boot_distribution_1d(resultlist_stat, filename, bootruns = -1,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist_stat	List of results from qat_analyse_boot_distribution_1d
filename	Name of the file without extension.
bootruns	Number of bootstrap runs used in the test.

`measurement_name`
: Name of the measurement.

`directoryname`
: Directory, where the resulted file should be stored.

`plotstyle`
: A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting vectors of qat_analyse_boot_distribution_1d. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

`qat_analyse_boot_distribution_1d`

Examples

```
vec <- rnorm(500)
result <- qat_analyse_boot_distribution_1d(vec, 100)
# this example produce a file exampleplot_bootdist.png in the current directory
qat_plot_boot_distribution_1d(result$stat, "exampleplot_bootdist",
bootruns=result$bootruns, measurement_name="Result of Check")
```

---

qat_plot_distribution_1d

Plot a distribution check result

Description

A plot of the result of a distribution check will be produced.

Usage

```
qat_plot_distribution_1d(resultlist_hist, filename, resultlist_stat,
numofbars = -1, measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist_hist	Result of a hist function.
filename	Name of the file without extension.
resultlist_stat	List of statistical parameters.
numofbars	Numbers of bars of the histogram plot.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_distribution_1d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_distribution_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_distribution_1d(vec, 15)
# this example produce a file exampleplot_dist.png in the current directory
qat_plot_distribution_1d(result$hist, "exampleplot_dist", result$stat,
numofbars=result$numofbars, measurement_name="Result of Check")
```

qat_plot_histogram_test *Plot a histogram test result*

Description

A plot of the result of a histogram test will be produced.

Usage

```
qat_plot_histogram_test(resultfield, filename, blocksize = -1, numofbars = -1,
factorofbar = -1, metric = NULL, runs = NULL, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

resultfield	The resulting matrix of qat_analyse_histogram_test_xxx_xd
filename	Name of the file without extension.
blocksize	Length of a block.
numofbars	Number of bins of the histograms.
factorofbar	Correction factor for non-value bins.
metric	Metric used for the comparison of the histograms.
runs	Number of used blocks.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting field of qat_analyse_histogram_test_xxx_xd. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
result <- qat_analyse_histogram_test_emd_1d(vec, 50, 65)
qat_plot_histogram_test(result$field, "test_emd_1d", result$blocksize,
result$numofbars, "emd", result$runs)
```

`qat_plot_lim_rule_dynamic_1d`
Plot a dynamic LIM rule result

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_dynamic_1d(flagvector, filename, measurement_vector = NULL,
min_vector = NULL, max_vector = NULL, min_vector_name = NULL, max_vector_name = NULL,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

`flagvector`	The resulting flagvector of qat_analyse_lim_rule_dynamic_1d
`filename`	Name of the file without extension.
`measurement_vector`	The measurement vector, which should be plotted
`min_vector`	The vector with the minimum values.
`max_vector`	The vector with the maximum values.
`min_vector_name`	Name of the vector of the minimum values.
`max_vector_name`	Name of the vector of the minimal values.
`measurement_name`	Name of the measurement.
`directoryname`	Directory, where the resulted file should be stored.
`plotstyle`	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_dynamic_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_lim_rule_dynamic_1d,qat_plot_lim_rule_static_1d, qat_plot_lim_rule_sigma_1d

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- qat_analyse_lim_rule_dynamic_1d(vec, min_vector, max_vector,
min_vector_name="minimum vector", max_vector_name="maximum vector")
# this example produce a file exampleplot_lim_dyn.png in the current directory
qat_plot_lim_rule_dynamic_1d(result$flagvector, "exampleplot_lim_dyn",
measurement_vector=vec, min_vector=result$min_vector, max_vector=result$max_vector,
min_vector_name=result$min_vector_name, max_vector_name=result$max_vector_name,
measurement_name="Result of Check")
```

---

qat_plot_lim_rule_dynamic_2d

Plot a dynamic LIM rule result

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_dynamic_2d(flagvector, filename, measurement_vector = NULL,
min_vector = NULL, max_vector = NULL, min_vector_name = NULL, max_vector_name = NULL,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_lim_rule_dynamic_2d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
min_vector	The vector (2d array) with the minimum values.
max_vector	The vector (2d array) with the maximum values.

`min_vector_name`
: Name of the vector of the minimum values.

`max_vector_name`
: Name of the vector of the minimal values.

`measurement_name`
: Name of the measurement.

`directoryname`
: Directory, where the resulted file should be stored.

`plotstyle`
: A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_dynamic_2d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_lim_rule_dynamic_1d, qat_analyse_lim_rule_dynamic_2d, qat_plot_lim_rule_static_2d, qat_plot_lim_rule_sigma_2d

Examples

```
vec <- array(rnorm(500),c(25,20))
min_vector <- array(rnorm(500)-2, c(25,20))
max_vector <- array(rnorm(500)+2, c(25,20))
result <- qat_analyse_lim_rule_dynamic_2d(vec, min_vector, max_vector,
min_vector_name="minimum vector", max_vector_name="maximum vector")
# this example produce a file exampleplot_lim_dyn.png in the current directory
qat_plot_lim_rule_dynamic_2d(result$flagvector, "exampleplot_lim_dyn",
measurement_vector=vec, min_vector=result$min_vector, max_vector=result$max_vector,
min_vector_name=result$min_vector_name, max_vector_name=result$max_vector_name,
measurement_name="Result of Check")
```

qat_plot_lim_rule_sigma_1d *Plot a sigma LIM rule result*

Description

A plot of the result of a dynamic lim rule check will be produced.

Usage

```
qat_plot_lim_rule_sigma_1d(flagvector, filename, measurement_vector = NULL,
sigma_factor = NULL, meanofvector = NaN, sdofvector = NULL, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_lim_rule_sigma_1d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
sigma_factor	The sigma factor, which was used, when the test were performed.
meanofvector	The mean of the measurement vector
sdofvector	The standard deviation of the measurement vector
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_sigma_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value

Author(s)

Andre Duesterhus

See Also

qat_analyse_lim_rule_static_1d,qat_plot_lim_rule_dynamic_1d, qat_plot_lim_rule_sigma_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_sigma_1d(vec, 2)
# this example produce a file exampleplot_lim_sig.png in the current directory
qat_plot_lim_rule_sigma_1d(result$flagvector, "exampleplot_lim_sig", measurement_vector=vec,
sigma_factor=result$sigma_factor, meanofvector=result$meanofvector, sdofvector=result$sdofvector,
measurement_name="Result of Check")
```

---

qat_plot_lim_rule_sigma_2d

Plot a sigma LIM rule result

Description

A plot of the result of a dynamic lim rule check will be produced.

Usage

```
qat_plot_lim_rule_sigma_2d(flagvector, filename, measurement_vector = NULL,
sigma_factor = NULL, meanofvector = NaN, sdofvector = NULL, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

`flagvector`	The resulting flagvector of qat_analyse_lim_rule_sigma_2d
`filename`	Name of the file without extension.
`measurement_vector`	The measurement vector, which should be plotted
`sigma_factor`	The sigma factor, which was used, when the test were performed.
`meanofvector`	The mean of the measurement vector
`sdofvector`	The standard deviation of the measurement vector
`measurement_name`	Name of the measurement.
`directoryname`	Directory, where the resulted file should be stored.
`plotstyle`	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_sigma_1d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value

Author(s)

Andre Duesterhus

See Also

qat_plot_lim_rule_static_1d, qat_analyse_lim_rule_static_2d, qat_plot_lim_rule_dynamic_2d, qat_plot_lim_rule_sigma_2d

Examples

```
vec <- array(rnorm(500), c(25,20))
result <- qat_analyse_lim_rule_sigma_2d(vec, 2)
# this example produce a file exampleplot_lim_sig.png in the current directory
qat_plot_lim_rule_sigma_2d(result$flagvector, "exampleplot_lim_sig",
measurement_vector=vec, sigma_factor=result$sigma_factor,
meanofvector=result$meanofvector, sdofvector=result$sdofvector,
measurement_name="Result of Check")
```

---

qat_plot_lim_rule_static_1d

Plot a static lim rule result

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_static_1d(flagvector, filename, measurement_vector = NULL,
min_value = NULL, max_value = NULL, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_lim_rule_static_1d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
min_value	The used minimum value of the test.
max_value	The used maximum value of the test.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_static_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_lim_rule_static_1d,qat_plot_lim_rule_dynamic_1d, qat_plot_lim_rule_sigma_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_lim_rule_static_1d(vec, -2,2)
# this example produce a file exampleplot_lim_sta.png in the current directory
qat_plot_lim_rule_static_1d(result$flagvector, "exampleplot_lim_sta",
measurement_vector=vec, min_value=result$min_value, max_value=result$max_value,
measurement_name="Testresult")
```

---

qat_plot_lim_rule_static_2d

Plot a static lim rule result

Description

A plot of the result of a dynamic LIM rule check will be produced.

Usage

```
qat_plot_lim_rule_static_2d(flagvector, filename, measurement_vector = NULL,
min_value = NULL, max_value = NULL, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_lim_rule_static_2d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
min_value	The used minimum value of the test.
max_value	The used maximum value of the test.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_lim_rule_static_2d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_lim_rule_static_1d, qat_analyse_lim_rule_static_2d, qat_plot_lim_rule_dynamic_2d, qat_plot_lim_rule_sigma_2d

Examples

```
vec <- array(rnorm(500),c(25,20))
result <- qat_analyse_lim_rule_static_2d(vec, -2, 2)
# this example produce a file exampleplot_lim_sta.png in the current directory
qat_plot_lim_rule_static_2d(result$flagvector, "exampleplot_lim_sta",
measurement_vector=vec, min_value=result$min_value, max_value=result$max_value,
measurement_name="Testresult")
```

qat_plot_noc_rule_1d *Plot a NOC rule result*

Description

A plot of the result of a NOC rule check will be produced.

Usage

```
qat_plot_noc_rule_1d(flagvector, filename, measurement_vector = NULL,
max_return_elements = 0, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_noc_rule_1d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_return_elements	The number of maximum reruning elements, which was used in the test.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_noc_rule_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_noc_rule_1d

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- qat_analyse_noc_rule_1d(vec, 1)
# this example produce a file exampleplot_noc.png in the current directory
qat_plot_noc_rule_1d(result$flagvector, "exampleplot_noc", measurement_vector=vec,
max_return_elements=result$max_return_elements, measurement_name="Result of Check")
```

---

qat_plot_noc_rule_2d *Plot a NOC rule result*

Description

A plot of the result of a NOC rule check will be produced.

Usage

```
qat_plot_noc_rule_2d(flagvector, filename, measurement_vector = NULL,
max_return_elements = 0, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_noc_rule_2d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_return_elements	The number of maximum reruning elements, which was used in the test.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_noc_rule_2d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_noc_rule_1d, qat_analyse_noc_rule_2d

Examples

```
vec <- array(c(1,1,1,2,2), c(25,20))
result <- qat_analyse_noc_rule_2d(vec, 1)
# this example produce a file exampleplot_noc.png in the current directory
qat_plot_noc_rule_2d(result$flagvector, "exampleplot_noc", measurement_vector=vec,
max_return_elements=result$max_return_elements, measurement_name="Result of Check")
```

---

qat_plot_roc_rule_dynamic_1d

Plot a dynamic ROC rule result

Description

A plot of the result of a dynamic ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_dynamic_1d(flagvector, filename, measurement_vector = NULL,
max_upward_vector = NULL, max_downward_vector = NULL, upward_vector_name = NULL,
downward_vector_name = NULL, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_roc_rule_dynamic_1d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_upward_vector	The vector with the upward values.
max_downward_vector	The vector with the downward values.
upward_vector_name	Name of the vector of the upward values.
downward_vector_name	Name of the vector of the downward values.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_roc_rule_dynamic_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_roc_rule_dynamic_1d,qat_plot_roc_rule_static_1d

Examples

```
vec <- rnorm(100)
min_vector<-seq(1,2,length.out=100)
max_vector<-seq(1,2,length.out=100)
result <- qat_analyse_roc_rule_dynamic_1d(vec, min_vector, max_vector,
upward_vector_name="upward vector", downward_vector_name="downward vector")
# this example produce a file exampleplot_roc_dyn.png in the current directory
qat_plot_roc_rule_dynamic_1d(result$flagvector, "exampleplot_roc_dyn",
measurement_vector=vec, max_upward_vector=result$max_upward_vector,
max_downward_vector=result$max_downward_vector, upward_vector_name=result$upward_vector_name,
downward_vector_name=result$downward_vector_name, measurement_name="Result of Check")
```

---

qat_plot_roc_rule_dynamic_2d

Plot a dynamic ROC rule result

Description

A plot of the result of a dynamic ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_dynamic_2d(flagvector, filename, measurement_vector = NULL,
max_upward_vector = NULL, max_downward_vector = NULL, upward_vector_name = NULL,
downward_vector_name = NULL, measurement_name = "", directoryname = "",
plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_roc_rule_dynamic_2d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_upward_vector	The vector (2d array) with the upward values.
max_downward_vector	The vector (2d array) with the downward values.
upward_vector_name	Name of the vector of the upward values.
downward_vector_name	Name of the vector of the downward values.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_roc_rule_dynamic_2d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_roc_rule_dynamic_1d, qat_analyse_roc_rule_dynamic_2d, qat_plot_roc_rule_static_2d

Examples

```
vec <- array(rnorm(500), c(25,20))
min_vector <- array(rnorm(500)+2, c(25,20))
max_vector <- array(rnorm(500)+2, c(25,20))
result <- qat_analyse_roc_rule_dynamic_2d(vec, min_vector, max_vector,
upward_vector_name="upward vector", downward_vector_name="downward vector")
# this example produce a file exampleplot_roc_dyn.png in the current directory
qat_plot_roc_rule_dynamic_2d(result$flagvector, "exampleplot_roc_dyn",
measurement_vector=vec, max_upward_vector=result$max_upward_vector,
```

```
max_downward_vector=result$max_downward_vector, upward_vector_name=result$upward_vector_name,
downward_vector_name=result$downward_vector_name, measurement_name="Result of Check")
```

---

qat_plot_roc_rule_static_1d
Plot a static ROC rule result

Description

A plot of the result of a static ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_static_1d(flagvector, filename, measurement_vector = NULL,
max_upward_value = 0, max_downward_value = 0, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_roc_rule_static_1d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_upward_value	The used maximum upward value.
max_downward_value	The used maximum downward value.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_roc_rule_static_1d. With additional information on the parameters, which were used while performing the test, this function will produce a more detailed plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_roc_rule_static_1d,qat_plot_roc_rule_dynamic_1d

Examples

```
vec <- rnorm(100)
result <- qat_analyse_roc_rule_static_1d(vec, 2,2)
# this example produce a file exampleplot_roc_sta.png in the current directory
qat_plot_roc_rule_static_1d(result$flagvector, "exampleplot_roc_sta",
measurement_vector=vec, max_upward_value=result$max_upward_value,
max_downward_value=result$max_downward_value, measurement_name="Result of Check")
```

---

qat_plot_roc_rule_static_2d

Plot a static ROC rule result

Description

A plot of the result of a static ROC rule check will be produced.

Usage

```
qat_plot_roc_rule_static_2d(flagvector, filename, measurement_vector = NULL,
max_upward_value = 0, max_downward_value = 0, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

flagvector	The resulting flagvector of qat_analyse_roc_rule_static_2d
filename	Name of the file without extension.
measurement_vector	The measurement vector, which should be plotted
max_upward_value	The used maximum upward value.
max_downward_value	The used maximum downward value.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_roc_rule_static_2d. Additional information on the parameters, which were used while performing the test, will be added to the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_plot_roc_rule_static_1d, qat_analyse_roc_rule_static_2d, qat_plot_roc_rule_dynamic_2d

Examples

```
vec <- array(rnorm(500), c(25,20))
result <- qat_analyse_roc_rule_static_2d(vec, 2,2)
# this example produce a file exampleplot_roc_sta.png in the current directory
qat_plot_roc_rule_static_2d(result$flagvector, "exampleplot_roc_sta",
measurement_vector=vec, max_upward_value=result$max_upward_value,
max_downward_value=result$max_downward_value, measurement_name="Result of Check")
```

---

qat_plot_slide_distribution_1d

Plot a slide distribution check result

Description

A plot of the result of a slide distribution check will be produced.

Usage

```
qat_plot_slide_distribution_1d(resultlist, filename, blocksize = -1,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_slide_distribution_1d
filename	Name of the file without extension.
blocksize	Length of the blocks
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_slide_distribution_1d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_slide_distribution_1d

Examples

```
vec <- rnorm(100)
result <- qat_analyse_slide_distribution_1d(vec, 10)
# this example produce a file exampleplot_slidedist.png in the current directory
qat_plot_slide_distribution_1d(result$stat, "exampleplot_slidedist",
blocksize=result$blocksize, measurement_name="Result of Check")
```

---

qat_plot_slide_distribution_2d

Plot a slide distribution check result

Description

A plot of the result of a slide distribution check will be produced.

Usage

```
qat_plot_slide_distribution_2d(resultlist, filename, blocksize = -1,
measurement_name = "", directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_slide_distribution_2d
filename	Name of the file without extension.
blocksize	Length of the blocks
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_slide_distribution_2d. Additional information on the parameters, which were used while performing the test, will be included into the plot. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_slide_distribution_1d

Examples

```
vec <- array(rnorm(100),c(25,20))
result <- qat_analyse_slide_distribution_2d(vec, 5)
# this example produce a file exampleplot_slidedist.png in the current directory
qat_plot_slide_distribution_2d(result$stat, "exampleplot_slidedist",
blocksize=result$blocksize, measurement_name="Result of Check")
```

---

qat_plot_trimmed_distribution_1d

Plot a trimmed distribution check result

Description

A plot of the result of a trimmed distribution check will be produced.

Usage

```
qat_plot_trimmed_distribution_1d(resultlist, filename, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_trimmed_distribution_1d
filename	Name of the file without extension.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_trimmed_distribution_1d. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_trimmed_distribution_1d

Examples

```
vec <- rnorm(1000)
result <- qat_analyse_trimmed_distribution_1d(vec)
# this example produce a file exampleplot_trimmeddist.png in the current directory
qat_plot_trimmed_distribution_1d(result$stat, "exampleplot_trimmeddist",
measurement_name="Result of Check")
```

---

qat_plot_trimmed_distribution_2d

Plot a trimmed distribution check result

Description

A plot of the result of a trimmed distribution check will be produced.

Usage

```
qat_plot_trimmed_distribution_2d(resultlist, filename, measurement_name = "",
directoryname = "", plotstyle = NULL)
```

Arguments

resultlist	List of results from qat_analyse_trimmed_distribution_2d
filename	Name of the file without extension.
measurement_name	Name of the measurement.
directoryname	Directory, where the resulted file should be stored.
plotstyle	A list with a qat color scheme.

Details

A plot will be produced, which base on the resulting flagvector of qat_analyse_trimmed_distribution_2d. When no plotstyle is defined the standard-colorscheme will be used. The resulting plot will be stored in the folder, which is defined by directory under the given filename, with the extension png.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_analyse_trimmed_distribution_2d

Examples

```
vec <- array(rnorm(100),c(25,20))
result <- qat_analyse_trimmed_distribution_2d(vec)
# this example produce a file exampleplot_trimmeddist.png in the current directory
qat_plot_trimmed_distribution_2d(result$stat, "exampleplot_trimmeddist",
measurement_name="Result of Check")
```

---

qat_read_parameter *Informations on a method*

Description

This functions delivers informations of methods, which are stored under the given filename.

Usage

```
qat_read_parameter(filename, methodname)
```

Arguments

filename	Filename of the file with the descriptions of the methods
methodname	Name of the method, where informations are required.

Details

This functions delivers informations of methods, which are stored under the given filename. For this the methodname will be used as a search parameter. The informations will be given back as a list.

Value

A list with the following elements:

name	Name of the method, may be corrected to standard name.
analysis_function	Name of the analysis function, which should be called for this method
plot_function	Name of the plot function, which should be called for this method
manipulation_function	Name of the manipulation function, which should be called for this method
description	Description of the method
algorithm	Algorithm of the method

Author(s)

Andre Duesterhus

Examples

```
#still to come
```

---

qat_run_workflow_check *Perform a workflow of checks*

Description

This function performs a workflow of checks by a given workflowlist on a given vector.

Usage

```
qat_run_workflow_check(measurement_vector, workflowlist, time = NULL, height = NULL,
lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL, vec4 = NULL)
```

Arguments

measurement_vector	The measurement vector, which should be tested
workflowlist	The workflowlist, which should be performed.
time	A time vector of the measurment_vector
height	A height vector of the measurment_vector
lat	A latitude vector of the measurment_vector
lon	A longitude vector of the measurment_vector
vec1	A potential additional vector
vec2	A potential additional vector
vec3	A potential additional vector
vec4	A potential additional vector

Details

This function performs a workflow of checks by a given workflowlist on a given measurement vector. Additional vectors can be used in the tests.

Value

A resultlist, with the results of the performed tests will be given back.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow, qat_run_workflow_plot

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

---

qat_run_workflow_plot *Produce plots of a workflow*

Description

This function produces plots of the results, which were produced by a workflow.

Usage

```
qat_run_workflow_plot(resultlist, measurement_name = "", directoryname = "",
basename = "", plotstyle = NULL)
```

Arguments

resultlist	The results, which are produced by qat_run_workflow_check
measurement_name	The measurement vector, which is used at the tests
directoryname	Directory, where the resulting plots should be stored
basename	Basic name of the filename
plotstyle	A list with a qat color scheme.

Details

The resultlist contains the parameters and results of the tests. From this the plots will be constructed and stored in the given directory. As filename the basename is used, with further extensions to indicate the tests. When no plotstyle is defined the standard-colorscheme will be used.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_run_workflow_check

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test")
# add some more informations for the workflow
```

```
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")

filename_out <- "myworkflow_result.xml"
# write edited workflow in current directory
qat_config_write_workflow(workflowlist, output_filename=filename_out)
```

`qat_run_workflow_save` *Performing a workflow of constructing saving elements by a given resultlist*

Description

This function performs a workflow of constructing a savelist by a given resultlist.

Usage

```
qat_run_workflow_save(resultlist, baseunit = "", time = NULL, height = NULL,
lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL, vec3 = NULL, vec4 = NULL)
```

Arguments

`resultlist`	Resultlist with results of checks
`baseunit`	Unit of the original measurement vector
`time`	A time vector of the measurment_vector
`height`	A height vector of the measurment_vector
`lat`	A latitude vector of the measurment_vector
`lon`	A longitude vector of the measurment_vector
`vec1`	A potential additional vector
`vec2`	A potential additional vector
`vec3`	A potential additional vector
`vec4`	A potential additional vector

Details

This function performs a workflow of constructing a savelist by a given resultlist. This can be used to build netCDF-files by the function qat_save_result_ncdf.

Value

A savelist, with the results of the performed tests will be given back.

Author(s)

Andre Duesterhus

See Also

qat_config_read_workflow, qat_run_workflow_check, qat_run_workflow_plot

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce the savelist
savelist <- qat_run_workflow_save(rlist)
filename_out <- "myresults"
# write netCDF-file of the results in current directory
## Not run:
qat_save_result_ncdf(testvector, savelist=savelist, filename_out,
workflowlist=workflowlist ,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
## End(Not run)
```

---

qat_save_block_distribution_1d

Produce a savelist from a resultlist for a Block Distribution Test

---

Description

This function takes the results, produced by qat_analyse_block_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_block_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_block_distribution, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_block_distribution_1d(vec, 50))
savelist <- qat_save_block_distribution_1d(result)
```

---

qat_save_block_distribution_2d

Produce a savelist from a resultlist for a Block Distribution Test

---

Description

This function takes the results, produced by qat_analyse_block_distribution_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_block_distribution_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_block_distribution, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_block_distribution_2d(vec, 5))
savelist <- qat_save_block_distribution_2d(result)
```

---

qat_save_boot_distribution_1d

Produce a savelist from a resultlist for a Boot Distribution Test

---

Description

This function takes the results, produced by qat_analyse_boot_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_boot_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_boot_distribution, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_boot_distribution_1d(vec, 1000))
savelist <- qat_save_boot_distribution_1d(result)
```

---

qat_save_boot_distribution_2d

Produce a savelist from a resultlist for a Boot Distribution Test

---

Description

This function takes the results, produced by qat_analyse_boot_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_boot_distribution_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_boot_distribution, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_boot_distribution_2d(vec, 10))
savelist <- qat_save_boot_distribution_2d(result)
```

qat_save_distribution_1d *Produce a savelist from a resultlist for a Distribution Test*

Description

This function takes the results, produced by qat_analyse_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_distribution, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_distribution_1d(vec, 15))
savelist <- qat_save_distribution_1d(result)
```

qat_save_histogram_test *Produce a savelist from a resultlist for a Histogram Test*

Description

This function takes the results, produced by qat_analyse_histogram_test_xxx_xd and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_histogram_test(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_histogram_test, qat_run_workflow_save

Examples

```
vec <- c(rnorm(1000), rnorm(1000)+1)
workflowlist_part <- list(blocksize=50, numofbars=65, metric="emd")
resultlist <- qat_call_histogram_test(vec, workflowlist_part, element=1)
savelist <- qat_save_histogram_test(resultlist[[2]])
```

qat_save_lim_rule_dynamic_1d *Produce a savelist from a resultlist for a LIM Rule Dynamic Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_dynamic_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_dynamic_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
min_vector<-seq(-1,-2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- list(result=qat_analyse_lim_rule_dynamic_1d(vec, min_vector,
max_vector, min_vector_name="minimum vector", max_vector_name="maximum vector"))
savelist <- qat_save_lim_rule_dynamic_1d(result)
```

qat_save_lim_rule_dynamic_2d *Produce a savelist from a resultlist for a LIM Rule Dynamic Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_dynamic_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_dynamic_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
min_vector<-array(seq(-1,-2,length.out=1000), c(10, 100))
max_vector<-array(seq(1,2,length.out=1000), c(10, 100))
result <- list(result=qat_analyse_lim_rule_dynamic_2d(vec, min_vector,
max_vector, min_vector_name="minimum vector", max_vector_name="maximum vector"))
savelist <- qat_save_lim_rule_dynamic_2d(result)
```

qat_save_lim_rule_sigma_1d *Produce a savelist from a resultlist for a LIM Rule Sigma Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_sigma_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_sigma_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_lim_rule_sigma_1d(vec, 2))
savelist <- qat_save_lim_rule_sigma_1d(result)
```

qat_save_lim_rule_sigma_2d *Produce a savelist from a resultlist for a LIM Rule Sigma Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_sigma_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_sigma_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_lim_rule_sigma_2d(vec, 2))
savelist <- qat_save_lim_rule_sigma_2d(result)
```

qat_save_lim_rule_static_1d *Produce a savelist from a resultlist for a LIM Rule Static Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_static_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_static_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_lim_rule_static_1d(vec, -2,2))
savelist <- qat_save_lim_rule_static_1d(result)
```

qat_save_lim_rule_static_2d *Produce a savelist from a resultlist for a LIM Rule Static Test*

Description

This function takes the results, produced by qat_analyse_lim_rule_static_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_lim_rule_static_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_lim_rule, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_lim_rule_static_2d(vec, -2,2))
savelist <- qat_save_lim_rule_static_2d(result)
```

`qat_save_noc_rule_1d` *Produce a savelist from a resultlist for a NOC Rule Test*

Description

This function takes the results, produced by qat_analyse_noc_rule_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_noc_rule_1d(resultlist_part, baseunit = "")
```

Arguments

`resultlist_part`	A list with the results of the check
`baseunit`	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

`qat_call_save_noc_rule`, `qat_run_workflow_save`

Examples

```
vec <- c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,1)
result <- list(result=qat_analyse_noc_rule_1d(vec, 1))
savelist <- qat_save_noc_rule_1d(result)
```

qat_save_noc_rule_2d *Produce a savelist from a resultlist for a NOC Rule Test*

Description

This function takes the results, produced by qat_analyse_noc_rule_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_noc_rule_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_noc_rule, qat_run_workflow_save

Examples

```
vec <- array(c(1,2,3,4,4,4,5,5,4,3,NaN,3,2,11), c(5,3))
result <- list(result=qat_analyse_noc_rule_2d(vec, 1))
savelist <- qat_save_noc_rule_2d(result)
```

qat_save_result_ncdf *Writing a savelist to a netCDF-file*

Description

A savelist, which is constructed by the function qat_run_workflow_save will be written to a given filename in netCDF format. Additional needed informations are the workflowlist, which constructed the savelist.

Usage

```
qat_save_result_ncdf(measurement_vector, savelist, filename, workflowlist = NULL,
time = NULL, height = NULL, lat = NULL, lon = NULL, vec1 = NULL, vec2 = NULL,
vec3 = NULL, vec4 = NULL, store_mes_vec = TRUE, baseunit = "unitless",
addunits = c("minutes", "metres", "degrees", "degrees", "unitless",
"unitless", "unitless", "unitless"), directoryname = "", nan_value = -999,
variable_name = "", transformationonvariable = "", authorname = "",
original_filename = "", data_level = "", workflow_filename = "")
```

Arguments

measurement_vector	The measurement vector, which was tested
savelist	The resulted savelist
filename	The name of the file, which should be written
workflowlist	The used workflowlist for the tests
time	A vector of time elements with the length of the measurement vector
height	A vector of height elements with the length of the measurement vector
lat	A vector of latitude elements with the length of the measurement vector
lon	A vector of longitude elements with the length of the measurement vector
vec1	An additional vector, which is named as vec1
vec2	An additional vector, which is named as vec2
vec3	An additional vector, which is named as vec3
vec4	An additional vector, which is named as vec4
store_mes_vec	A bolean variable if the measurement vector should also be stored
baseunit	Unit of the measurement vector
addunits	Vector of units for the other vectors
directoryname	Directory, where the resulting file should be stored
nan_value	Fill value for NaN in vectors
variable_name	Name of the original variable
transformationonvariable	Information on transformation of the original variable

`authorname` Name of the author who performed the tests

`original_filename` Filename, where the original data was stored

`data_level` Data level of the original variable

`workflow_filename` Filename of the workflow

Details

The savelist, which is a result of the function qat_run_workflow_save, which transformed the resultlist of qat_run_workflow_check to a here usable formate, delivers all necessary information to construct a netCDF-file. The workflowlist is needed, because further informations, like algorithms, descriptions and comments on results are simpler to edit in this list. This can be also saved by qat_config_write_workflow to a XML-format. The netCDF-format used here is the QAD-convention. This allows to store the modifications of a tests and also the results into one file.

Value

No return value.

Author(s)

Andre Duesterhus

See Also

qat_run_workflow_save

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce the savelist
savelist <- qat_run_workflow_save(rlist)
filename_out <- "myresults"
# write netCDF-file of the results in current directory
## Not run:
qat_save_result_ncdf(testvector, savelist=savelist, filename_out,
workflowlist=workflowlist,vec1=maxlim, vec2=minlim, vec3=uproc, vec4=downroc)
```

```
## End(Not run)
```

---

qat_save_roc_rule_dynamic_1d *Produce a savelist from a resultlist for a ROC Rule Dynamic Test*

Description

This function takes the results, produced by qat_analyse_roc_rule_dynamic_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_dynamic_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_roc_rule, qat_run_workflow_save

Examples

```
vec <- rnorm(100)
min_vector<-seq(1,2,length.out=1000)
max_vector<-seq(1,2,length.out=1000)
result <- list(result=qat_analyse_roc_rule_dynamic_1d(vec, min_vector,
max_vector, upward_vector_name="upward vector",
downward_vector_name="downward vector"))
savelist <- qat_save_roc_rule_dynamic_1d(result)
```

qat_save_roc_rule_dynamic_2d *Produce a savelist from a resultlist for a ROC Rule Dynamic Test*

Description

This function takes the results, produced by qat_analyse_roc_rule_dynamic_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_dynamic_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_roc_rule, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
min_vector<-array(seq(-1,-2,length.out=1000), c(10, 100))
max_vector<-array(seq(1,2,length.out=1000), c(10, 100))
result <- list(result=qat_analyse_roc_rule_dynamic_2d(vec, min_vector,
max_vector, upward_vector_name="upward vector", downward_vector_name="downward vector"))
savelist <- qat_save_roc_rule_dynamic_2d(result)
```

qat_save_roc_rule_static_1d *Produce a savelist from a resultlist for a ROC Rule Static Test*

Description

This function takes the results, produced by qat_analyse_roc_rule_static_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_static_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_roc_rule, qat_run_workflow_save

Examples

```
vec <- rnorm(100)
result <- list(result=qat_analyse_roc_rule_static_1d(vec, 2,2))
savelist <- qat_save_roc_rule_static_1d(result)
```

qat_save_roc_rule_static_2d *Produce a savelist from a resultlist for a ROC Rule Static Test*

Description

This function takes the results, produced by qat_analyse_roc_rule_static_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_roc_rule_static_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_roc_rule, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_roc_rule_static_2d(vec, 2,2))
savelist <- qat_save_roc_rule_static_2d(result)
```

qat_save_set_addup_1d *Produce a savelist from a resultlist for a Set Addup*

Description

This function takes the results, produced by qat_analyse_set_addup_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_addup_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_set_addup

Examples

```
## still to come
```

qat_save_set_mean_1d *Produce a savelist from a resultlist for a Set Mean*

Description

This function takes the results, produced by qat_analyse_set_mean_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_mean_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_set_mean

Examples

```
## still to come
```

qat_save_set_nans_1d *Produce a savelist from a resultlist for a Set NAN*

Description

This function takes the results, produced by qat_analyse_set_nan_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_set_nans

Examples

```
## still to come
```

qat_save_set_nans_above_1d *Produce a savelist from a resultlist for a Set NAN above*

Description

This function takes the results, produced by qat_analyse_set_nan_above_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_above_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_set_nans

Examples

```
## still to come
```

qat_save_set_nans_below_1d *Produce a savelist from a resultlist for a Set NAN below*

Description

This function takes the results, produced by qat_analyse_set_nan_below_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_set_nans_below_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultlist and transfer the content to a newly organized list. This consists mainly of a text, which is use in the parameter description for a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_set_nans

Examples

```
## still to come
```

qat_save_slide_distribution_1d *Produce a savelist from a resultlist for a Slide Distribution Test*

Description

This function takes the results, produced by qat_analyse_slide_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_slide_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_slide_distribution, qat_run_workflow_save

Examples

```
vec <- rnorm(100)
result <- list(result=qat_analyse_slide_distribution_1d(vec, 10))
savelist <- qat_save_slide_distribution_1d(result)
```

qat_save_slide_distribution_2d *Produce a savelist from a resultlist for a Slide Distribution Test*

Description

This function takes the results, produced by qat_analyse_slide_distribution_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_slide_distribution_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. This also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_slide_distribution, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_slide_distribution_2d(vec, 5))
savelist <- qat_save_slide_distribution_2d(result)
```

qat_save_trimmed_distribution_1d *Produce a savelist from a resultlist for a Trimmed Distribution Test*

Description

This function takes the results, produced by qat_analyse_trimmed_distribution_1d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_trimmed_distribution_1d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_trimmed_distribution, qat_run_workflow_save

Examples

```
vec <- rnorm(1000)
result <- list(result=qat_analyse_trimmed_distribution_1d(vec))
savelist <- qat_save_trimmed_distribution_1d(result)
```

qat_save_trimmed_distribution_2d *Produce a savelist from a resultlist for a Trimmed Distribution Test*

Description

This function takes the results, produced by qat_analyse_trimmed_distribution_2d and construct a savelist, which may be used to produce a netCDF output.

Usage

```
qat_save_trimmed_distribution_2d(resultlist_part, baseunit = "")
```

Arguments

resultlist_part	A list with the results of the check
baseunit	The unit of the original measurement vector

Details

This function takes the resultslist and transfer the content to a newly organized list. Ths also consists of more information, which help to generate an output like a netCDF-file.

Value

Returning a savelist with the content of the resultlist.

Author(s)

Andre Duesterhus

See Also

qat_call_save_trimmed_distribution, qat_run_workflow_save

Examples

```
vec <- array(rnorm(1000), c(10, 100))
result <- list(result=qat_analyse_trimmed_distribution_2d(vec))
savelist <- qat_save_trimmed_distribution_2d(result)
```

qat_style_plot *Produce a plotstylelist*

Description

Loads a plotstyle, when a filename is given. When not a standard plotstyle will be given back

Usage

```
qat_style_plot(filename = "")
```

Arguments

filename	Filename of a plotstyle-XML

Details

A plotstyle is a possibility to include a colorsheme in every plot, which is produced by the qat-package. With given filename a certain plotstyle will be loaded. Without a standard sheme will be used.

Value

A list with the inforamtion of the colorsheme.

Author(s)

Andre Duesterhus

See Also

qat_run_workflow_plot

Examples

```
library("qat")
# define testvector
testvector<-rnorm(500)
# read in workflow from systemfiles
filename_in <- system.file("extdata/workflowexample.xml", package="qat")
workflowlist <- qat_config_read_workflow(filename_in)
# define some additional vectors
maxlim <- seq(3,1,length.out=500)
minlim <- seq(-1,-3,length.out=500)
uproc <- seq(1,3,length.out=500)
downroc <- seq(3,1,length.out=500)
# load plotstyle
filename_ps <- system.file("extdata/plotstyle1.xml", package="qat")
ps<-qat_style_plot(filename_ps)
```

```
# run the workflow on the testvector
rlist <- qat_run_workflow_check(testvector,workflowlist,vec1=maxlim, vec2=minlim,
vec3=uproc, vec4=downroc)
# produce some plots of the result in teh current directory with new plotstyle
qat_run_workflow_plot(rlist, measurement_name="Test", basename="test", plotstyle=ps)
# add some more informations for the workflow
workflowlist <- qat_add_all_descriptions(workflowlist)
workflowlist <- qat_add_all_algorithms(workflowlist)
workflowlist <- qat_add_comment(workflowlist, 1, "No problems")
```

Index