

Package: **piamValidation** (via r-universe)

October 15, 2026

Type Package

Title Validation Tools for PIK-PIAM

Version 0.19.0

Date 2026-10-07

Description The `piamValidation` package provides validation tools for the Potsdam Integrated Assessment Modelling environment.

License LGPL-3

URL <https://github.com/pik-piam/piamValidation>,
<https://pik-piam.github.io/piamValidation>

Imports devtools, dplyr (>= 1.1.1), DT, ggplot2, ggthemes, htmltools, knitr, `piamInterfaces`, `piamutils`, plotly, quitte (>= 0.3123.0), readr, readxl, tibble, tidy

Suggests gridExtra, htmlwidgets, kableExtra, madrat, magclass, testthat (>= 3.0.0), remind2, rmarkdown, rprojroot, scales, scenarioevaluationcriteria

Remotes scenarioevaluationcriteria=github::IAMconsortium/scenario-evaluation-criteria

Encoding UTF-8

Depends R (>= 2.10)

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website tidyverse/tidytemplate

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev git make libharfbuzz-dev libgit2-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://pik-piam.r-universe.dev>

Date/Publication 2026-10-08 13:13:17 UTC

RemoteUrl <https://github.com/pik-piam/piamValidation>

RemoteRef HEAD

RemoteSha 0437cce197b45d02d9b5abfd96380f6e63603fab

Contents

piamValidation-package	2
appendTooltips	3
average_2020	3
checkRefData	4
checkUnits	4
combineData	5
convertSCIcriteria	5
getConfig	6
getThresholdBands	6
importScenarioData	7
linePlotThresholds	7
listConfigs	8
listReports	8
refineRefData	9
showInputSummary	9
validateScenarios	10
validationHeatmap	10
validationPass	11
validationReport	11
Index	13

piamValidation-package
<i>Validation Tools for PIK-PIAM</i>

Description

The piamValidation package provides validation tools for the Potsdam Integrated Assessment Modelling environment.

Author(s)

Maintainer: Pascal Weigmann <pascal.weigmann@pik-potsdam.de> ([ORCID](#)) (affiliation: Potsdam Institute for Climate Impact Research)

Authors:

- Pascal Weigmann <pascal.weigmann@pik-potsdam.de> ([ORCID](#)) (affiliation: Potsdam Institute for Climate Impact Research)
- Oliver Richters ([ORCID](#)) (affiliation: Potsdam Institute for Climate Impact Research)
- Fabrice Lécuyer ([ORCID](#)) (affiliation: Potsdam Institute for Climate Impact Research)
- Johannes Koch ([ORCID](#)) (affiliation: Potsdam Institute for Climate Impact Research)

See Also

Useful links:

- <https://github.com/pik-piam/piamValidation>
- <https://pik-piam.github.io/piamValidation>

appendTooltips	<i>construct tooltips for interactive plots</i>
----------------	---

Description

construct tooltips for interactive plots

Usage

```
appendTooltips(df)
```

Arguments

df	data.frame as returned from ‘validateScenarios()’
----	---

average_2020	<i>Average 2020 to smoothen Covid shock in historical data</i>
--------------	--

Description

Adds a new model for each model in reference data with smoothed 2020 period and name "<model>_smoothed".

Usage

```
average_2020(hist)
```

Arguments

hist	reference data as used in “validateScenarios()”
------	---

checkRefData	<i>Warn if reference data required by the config is missing from input data</i>
--------------	---

Description

Compares the reference models of all historical checks in the config against the models found in the reference data provided by the user (scenario "historical") and warns about missing ones. If reference data files shipped with the package contain the missing models, they are suggested.

Usage

```
checkRefData(cfg, hist)
```

Arguments

cfg	processed config as used in “validateScenarios()”
hist	historical/reference data as used in “validateScenarios()”

checkUnits	<i>Check variable consistency</i>
------------	-----------------------------------

Description

Test whether unit of on row of config and data for this variable match.

Usage

```
checkUnits(data, cfgRow)
```

Arguments

data	scenario or reference data for one variable
cfgRow	one row of a config file containing the same variable as the data object

combineData	<i>Combine scenario and reference data with thresholds</i>
-------------	--

Description

for one row of cfg: filter and merge relevant scenario data with cfg results in one df that contains scenario data, reference data and thresholds

Usage

```
combineData(scenData, cfgRow, histData = NULL)
```

Arguments

scenData	scenario data for one variable
cfgRow	one row of a config file
histData	reference data

convertSCIcriteria	<i>Import SCI criteria and convert to config</i>
--------------------	--

Description

Use installed "scenario-evaluation-criteria" package to convert threshold criteria from Scenario Compass release to a piamValidation config and a reference data .rds file

Usage

```
convertSCIcriteria(version = "2026.8.3")
```

Arguments

version	criteria version that should be converted to config
---------	---

getConfig	<i>import a config shipped with the package</i>
-----------	---

Description

get a validation config file either from "inst/config" (.csv) or by providing a full path (.csv or .xlsx) or a tibble with the necessary columns see README or vignette for rules on how to fill the config

Usage

```
getConfig(config)
```

Arguments

config	name of validation config from "inst/config"
--------	--

getThresholdBands	<i>takes the output of "validateScenarios()" and computes threshold bands</i>
-------------------	---

Description

converts the thresholds attached to each data point by "validateScenarios()" into absolute values per variable, unit, region and period. The resulting bands can be plotted as colored background areas behind scenario data, see "linePlotThresholds()" or the "thresholds" argument of "mip::createLinePlots()".

Usage

```
getThresholdBands(validData, warnMultiple = TRUE)
```

Arguments

validData	data.frame as returned by "validateScenarios()"
warnMultiple	if TRUE, warn in case multiple sets of thresholds are found for the same variable, unit, region and period, e.g. from checks using multiple metrics or reference scenarios; plotting such bands can lead to overlaps

Details

For checks using the metrics "relative" and "difference", thresholds are converted into absolute values using the reference values of the respective check. Thresholds which are not defined in the config appear as -Inf/Inf.

Value

data.frame with columns variable, unit, region, period, min_red, min_yel, max_yel and max_red

importScenarioData	<i>import IAM data for validation</i>
--------------------	---------------------------------------

Description

import IAM data for validation

Usage

```
importScenarioData(scenarioPath, variables = NULL)
```

Arguments

scenarioPath	one or multiple paths to .mif, .csv, .rds or .xlsx file(s) or a data.frame containing scenario data in IAM format
variables	optional character vector of variable names (may contain "*" wildcards as in the config); if given, all other variables are dropped while reading so that only the data needed for the validation is held in memory

linePlotThresholds	<i>takes the output of "validateScenarios()" creates a line plot</i>
--------------------	--

Description

takes the output of "validateScenarios()" creates a line plot

Usage

```
linePlotThresholds(
  valiData,
  scenData = NULL,
  refData = NULL,
  xlim = c(2010, 2030),
  interactive = TRUE
)
```

Arguments

valiData	data to be plotted, as returned by “validateScenarios()” and after filtering for one variable and one region.
scenData	hand over additional scenario data to be plotted alongside the validation data. Will use the same variable and region, otherwise all available data.
refData	hand over additional reference data to be plotted alongside the validation data. Will use the same variable and region, otherwise all available data.
xlim	set limits for the x axis
interactive	decide if an interactive ggploty object should be returned

listConfigs	<i>List available configs</i>
-------------	-------------------------------

Description

List all validation configuration files that are delivered with the package and can be directly imported with “getConfig()” or used in “validateScenarios()” and “validationReport()”.

Usage

```
listConfigs()
```

Value

character vector of config names

listReports	<i>List available reports</i>
-------------	-------------------------------

Description

List all validation report templates (.Rmd) that are delivered with the package and can be used as “report” in “validationReport()”.

Usage

```
listReports()
```

Value

character vector of report names

refineRefData	<i>evaluate the content of the "ref_<type>" column and filter reference data accordingly. cases: - mode chosen - range - mean - no mode chosen - use mean if multiple references</i>
---------------	--

Description

returns df without variable, unit and <type> columns (see below) returns df with ref_value_min/max, ref_model, ref_scenario, ref_period

Usage

```
refineRefData(ref_data, cfgRow, ref_type = "ref_model")
```

Arguments

ref_data	pre-filtered reference data
cfgRow	row of validation config used for this data slice
ref_type	historical, model, scenario, period

showInputSummary	<i>Print a summary of the input data</i>
------------------	--

Description

Print multiple metrics of all data objects given to “validateScenarios()” which might be helpful to spot and fix data inconsistencies.

Usage

```
showInputSummary(scen, hist, config)
```

Arguments

scen	scenario data as used in “validateScenarios()”
hist	historical/reference data as used in “validateScenarios()”
config	config as used in “validateScenarios()”

validateScenarios	<i>performs the validation checks from a config on a scenario data set</i>
-------------------	--

Description

performs the validation checks from a config on a scenario data set

Usage

```
validateScenarios(
  dataPath,
  config,
  outputFile = NULL,
  extraColors = TRUE,
  giveSummary = FALSE
)
```

Arguments

dataPath	one or multiple path(s) to scenario data in .mif or .csv format, in case of historic comparison, also path to reference data
config	select config from inst/config or give a full path to a config file on your computer
outputFile	give name of output file in case results should be exported; include file extension
extraColors	if TRUE, use cyan and blue for violation of min thresholds instead of using the same colors as for max thresholds (yel and red)
giveSummary	print a summary of input data via “showInputSummary()” which allows spotting data inconsistencies

validationHeatmap	<i>takes the output of "validateScenarios()" and plots heat maps per variable</i>
-------------------	---

Description

takes the output of "validateScenarios()" and plots heat maps per variable

Usage

```
validationHeatmap(
  valiData,
  main_dim = "variable",
  x_plot = NULL,
  y_plot = NULL,
  x_facet = NULL,
  y_facet = NULL,
  interactive = TRUE
)
```

Arguments

validData	data to be plotted, as returned by “validateScenarios()” (and “appendTooltips()” if interactive), plus optional filtering. Needs to have at least one dimension with only one unique element.
main_dim	out of the 5-dim df, 1 dim has to contain only one element, this is the main dimension of the plot, default: variable
x_plot	choose dimension to display on x-axis of plot, if any is NULL, arrangement is chosen automatically based on data dimensions
y_plot	choose dimension to display on y-axis of plot
x_facet	choose dimension to display on x-dim of facets
y_facet	choose dimension to display on x-dim of facets
interactive	return plots as interactive plotly plots by default

validationPass	<i>returns information on whether scenarios passed critical validation checks</i>
----------------	---

Description

returns information on whether scenarios passed critical validation checks

Usage

```
validationPass(data, yellowFail = FALSE)
```

Arguments

data	data.frame as returned from “validateScenarios()”
yellowFail	if set to TRUE a yellow check result of a critical variable will lead to the scenario not passing as validated

validationReport	<i>perform validateScenarios and create an .html report using .Rmd templates</i>
------------------	--

Description

perform validateScenarios and create an .html report using .Rmd templates

Usage

```
validationReport(  
  dataPath,  
  config,  
  report = "default",  
  outputDir = "output",  
  extraColors = TRUE,  
  giveSummary = FALSE  
)
```

Arguments

dataPath	one or multiple path(s) to scenario data in .mif or .csv format
config	name a config from inst/config ("validationConfig_<name>.csv") or give a full path to a separate configuration file
report	name a .Rmd from inst/markdown ("validationReport_<name>.Rmd") to be rendered or give a full path to a separate .Rmd file
outputDir	choose a directory to save validation reports to
extraColors	if TRUE, use cyan and blue for violation of min thresholds instead of using the same colors as for max thresholds (yel and red)
giveSummary	print a summary of input data via "showInputSummary()" which allows spotting data inconsistencies

Index

[appendTooltips](#), [3](#)
[average_2020](#), [3](#)

[checkRefData](#), [4](#)
[checkUnits](#), [4](#)
[combineData](#), [5](#)
[convertSCICriteria](#), [5](#)

[getConfig](#), [6](#)
[getThresholdBands](#), [6](#)

[importScenarioData](#), [7](#)

[linePlotThresholds](#), [7](#)
[listConfigs](#), [8](#)
[listReports](#), [8](#)

[piamValidation](#)
 ([piamValidation-package](#)), [2](#)
[piamValidation-package](#), [2](#)

[refineRefData](#), [9](#)

[showInputSummary](#), [9](#)

[validateScenarios](#), [10](#)
[validationHeatmap](#), [10](#)
[validationPass](#), [11](#)
[validationReport](#), [11](#)