

# Package ‘mipplot’

May 8, 2026

**Type** Package

**Title** An Open-Source Tool for Visualization of Climate Mitigation Scenarios

**Version** 0.3.1

**Maintainer** Akimitsu Inoue <inoue.akimitsu@chino-js.com>

**Description** Generic functions to produce area/bar/box/line plots of data following IAMC (Integrated Assessment Modeling Consortium) submission format.

**Imports** ggplot2, stringr, tidyr, shinyWidgets, data.table, readxl, shiny.i18n (>= 0.2.0), showtextdb, shinyalert, readr, showtext, shiny (>= 1.5.0), reshape, rlang, dplyr, reshape2

**Depends** R (>= 3.5.0), stats, utils, graphics, grDevices,

**License** MIT + file LICENSE

**Suggests** testthat, knitr, rmarkdown, tidyverse

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Diego Silva Herran [aut],  
Jiayang WANG [aut],  
Masahiro SUGIYAMA [aut],  
Hiroto SHIRAKI [aut],  
Akimitsu Inoue [ctr, cre]

**Repository** CRAN

**Date/Publication** 2021-05-18 07:20:02 UTC

## Contents

add_credit_to_list_of_plot . . . . .	3
add_credit_to_plot . . . . .	3

ar5_db_sample_data . . . . .	4
ar5_db_sample_rule_table . . . . .	4
change_data_types_of_iamc_dataframe . . . . .	5
check_column_availability . . . . .	5
check_format_of_iamc_dataframe . . . . .	6
correct_format_of_iamc_dataframe . . . . .	6
generate_code_to_plot_area . . . . .	7
generate_code_to_plot_bar . . . . .	7
generate_code_to_plot_line . . . . .	8
get_model_name_list . . . . .	9
get_scenario_name_list . . . . .	9
get_string_expression_of_vector_of_strings . . . . .	10
get_variable_group_name_list . . . . .	10
get_variable_name_list_in_variable_group . . . . .	11
mipplot . . . . .	11
mipplot_additivity_check . . . . .	12
mipplot_additivity_check_bar . . . . .	12
mipplot_area . . . . .	13
mipplot_autofill_color . . . . .	15
mipplot_bar . . . . .	15
mipplot_box . . . . .	17
mipplot_default_color_palette . . . . .	18
mipplot_generate_color_mapper . . . . .	18
mipplot_interactive_additivity_check_bar . . . . .	19
mipplot_interactive_area . . . . .	20
mipplot_interactive_bar . . . . .	20
mipplot_interactive_line . . . . .	21
mipplot_interactive_plot_line . . . . .	22
mipplot_line . . . . .	22
mipplot_point . . . . .	24
mipplot_print_pdf . . . . .	25
mipplot_read_iamc . . . . .	26
mipplot_read_ruletab . . . . .	27
mipplot_return_table . . . . .	27
mipplot_var_submission . . . . .	28
read_iamc_xlsx . . . . .	28
split_variable_into_positive_and_negative_parts . . . . .	29
sr15_sample_conversion_rule_table . . . . .	30
sr15_sample_data . . . . .	30

---

`add_credit_to_list_of_plot`  
*Add credit text to plots*

---

**Description**

Add credit text to a list of ggplot2 plot objects

**Usage**

```
add_credit_to_list_of_plot(list_of_plot)
```

**Arguments**

`list_of_plot` list of ggplot2 plot objects

**Value**

list of modified ggplot2 plot objects

---

`add_credit_to_plot` *Add credit text to a plot*

---

**Description**

Add credit text and project URL to a ggplot2 plot object

**Usage**

```
add_credit_to_plot(plot_object)
```

**Arguments**

`plot_object` ggplot2 plot object

**Value**

modified ggplot2 plot object

---

ar5\_db\_sample\_data      *Sample Dataset*

---

**Description**

A sample dataset of IAMC format

**Usage**

ar5\_db\_sample\_data

**Format**

A tibble data.table with 25240 rows and 7 variables:

**model** model, categorical

**scenario** scenario, categorical

**region** region, ASIA, OECD90 or World

**variable** the name of simulated variable that changes over time

**unit** unit of a variable

**period** year

**value** the value of a variable

**Source**

<https://tntcat.iiasa.ac.at/AR5DB/dsd?Action=htmlpage&page=about>

---

ar5\_db\_sample\_rule\_table  
*Sample Rule Table*

---

**Description**

A sample rule table

**Usage**

ar5\_db\_sample\_rule\_table

**Format**

A data frame of additivity rule.

**Rule\_ID** rule id

**Left\_side** name of left-side variable

**Right\_side** name of right-side variable

**Color\_code** hex color code

---

*change\_data\_types\_of\_iamc\_dataframe*  
*change column data type in data-set*

---

**Description**

change column data type in data-set to be able to be treated as an IAMC data-set.

**Usage**

`change_data_types_of_iamc_dataframe(iamc_data)`

**Arguments**

`iamc_data` data frame which has columns 'model', 'scenario', 'region', 'variable', 'period', 'unit'

**Value**

converted data-frame.

---

*check\_column\_availability*  
*check if the dataset has required fields of IAMC dataset*

---

**Description**

if dataset has all required fields, then returns TRUE

**Usage**

`check_column_availability(iamc_data)`

**Arguments**

`iamc_data` IAMC data frame

**Value**

boolean flag

check\_format\_of\_iamc\_dataframe

*check if the format of given data is valid as an IAMC dataset.*

---

**Description**

check if the format of given data is valid as an IAMC dataset.

**Usage**

```
check_format_of_iamc_dataframe(iamc_data)
```

**Arguments**

iamc\_data      IAMC dataset in dataframe format

**Value**

TRUE if it is valid

---

correct\_format\_of\_iamc\_dataframe

*correct data format of given IAMC data table*

---

**Description**

Dataset in IAMC format rule is not rigid. This function corrects data types of columns in the dataset. If necessary columns is missing, it throws exception. Output object of this function is as follows:

type: data.table columns: model: factor scenario: factor region: factor variable: factor unit: factor period: double value: double

**Usage**

```
correct_format_of_iamc_dataframe(iamc_data)
```

**Arguments**

iamc\_data      IAMC dataset described above

**Value**

modified dataframe

---

generate\_code\_to\_plot\_area  
*generate code to reproduce area plot*

---

### Description

This function is called in the `mipplot_interactive_area()` and provides R code to reproduce the currently drawn plot. This function cannot be used out of reactive expression in Shiny.

### Usage

```
generate_code_to_plot_area(  
  input,  
  name_of_input_data_variable,  
  name_of_input_rule_table_variable  
)
```

### Arguments

`input` This is the same as the `input` argument in the `shiny:ui()`.  
`name_of_input_data_variable` A string such as "ar5\_sample\_data".  
`name_of_input_rule_table_variable` A string such as "ar5\_sample\_rule".

### Value

A string representing the R code for rerun.

---

generate\_code\_to\_plot\_bar  
*generate code to reproduce bar plot*

---

### Description

This function is called in the `mipplot_interactive_bar()` and provides R code to reproduce the currently drawn plot. This function cannot be used out of reactive expression in Shiny.

### Usage

```
generate_code_to_plot_bar(  
  input,  
  name_of_input_data_variable,  
  name_of_input_rule_table_variable  
)
```

**Arguments**

`input` This is the same as the input argument in the `shiny::ui()`.

`name_of_input_data_variable`  
A string such as "ar5\_sample\_data".

`name_of_input_rule_table_variable`  
A string such as "ar5\_sample\_rule".

**Value**

A string representing the R code for rerun.

---

`generate_code_to_plot_line`  
*generate code to reproduce line plot*

---

**Description**

from 'input' argument generally used in reactive context in Shiny, this function generates R code to reproduce current plot. This function could not used out of reactive expression in Shiny.

**Usage**

```
generate_code_to_plot_line(input, name_of_iamc_data_variable = "D")
```

**Arguments**

`input` it is same as the argument of `shiny::ui()` this function accesses following attributes: - model - period - variable - scenario - region

`name_of_iamc_data_variable`  
name of IAMC data variable

**Value**

R code

---

get\_model\_name\_list     *Get name list of models in IAMC formatted data frame*

---

**Description**

select name of models from the column "model" then make unique it. output is character vector such as, c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4" )

**Usage**

```
get_model_name_list(D)
```

**Arguments**

D                     A quitte format dataframe of IAMC data to produce graph.

**Value**

A list of strings representing model names

---

get\_scenario\_name\_list  
*Get name list of scenarios in IAMC formatted data frame*

---

**Description**

select name of scenarios from the column "scenario" then make unique it. output is character vector such as, c("EMF27-450-Conv", "EMF27-450-FullTech", "EMF27-450-NoCCS", "EMF27-450-NucOff")

**Usage**

```
get_scenario_name_list(D)
```

**Arguments**

D                     A quitte format dataframe of IAMC data to produce graph.

**Value**

A list of strings representing scenario names

---

```
get_string_expression_of_vector_of_strings
```

*Get expression of vector of string in string format*

---

**Description**

To evaluate expression, get string of expression

**Usage**

```
get_string_expression_of_vector_of_strings(vector_of_strings)
```

**Arguments**

```
vector_of_strings
```

vector of strings, such as c("A", "B")

**Value**

An R code representing character vector

---

```
get_variable_group_name_list
```

*Get variable-group-name list*

---

**Description**

variable-group is a combination of one LHS and one or more RHS. this function outputs the list of names of variable-group in given rule-table. the format of return value is "LHS|RHS1,RHS2,RHS3,...".

**Usage**

```
get_variable_group_name_list(rule_table)
```

**Arguments**

```
rule_table
```

A rule table

**Value**

variable group name

**Examples**

```
get_variable_group_name_list(ar5_db_sample_rule_table)
```

---

`get_variable_name_list_in_variable_group`  
*Get variable name list in given variable-group*

---

**Description**

Scan rule-table and extract variable names in given variable-group.

**Usage**

`get_variable_name_list_in_variable_group(group_name)`

**Arguments**

`group_name`      `variable-group-name`

**Value**

A list of strings representing variable names

**Examples**

```
get_variable_name_list_in_variable_group(  
  "Final Energy|Industry,Residential and Commercial,Transportation")
```

---

`mipplot`                      *mipplot*

---

**Description**

Package contains generic functions to produce area/bar/box/line plots of data following IAMC submission format.

---

`mipplot_additivity_check`*check additivity of rules and data*

---

**Description**

This function is used for debugging a rule table and data-set. An input is a rule table and a data-set, the outputs are some area plots showing the divergence between the left-side variable and the sum of the right-side variables.

**Usage**

```
mipplot_additivity_check(D, R, max_n_plots = Inf, plot_all = FALSE)
```

**Arguments**

<code>D</code>	A dataframe of IAMC data in tibble format to produce area plots.
<code>R</code>	A dataframe of data aggregation rules (meta data).
<code>max_n_plots</code>	The maximum number of output plots.
<code>plot_all</code>	set FALSE to plot only inconsistent combinations

**Value**

A list of area plots.

**Examples**

```
if (interactive()) {  
  mipplot_additivity_check(  
    ar5_db_sample_data, ar5_db_sample_rule_table, max_n_plots = 10)  
}
```

---

`mipplot_additivity_check_bar`*Additivity check using bar plot*

---

**Description**

This function is used for debugging a rule table and data-set. An input is a rule table and a data-set, the outputs are some bar plots showing the divergence between the left-side variable and the sum of the right-side variables.

**Usage**

```
mipplot_additivity_check_bar(
  D,
  R,
  target_scenarios,
  target_rule_ids = 4,
  show_all_scenarios = FALSE,
  show_all_rule_ids = FALSE,
  debug = FALSE
)
```

**Arguments**

D                    A dataframe of IAMC data in tibble format to produce area plots.

R                    A dataframe of data aggregation rules (meta data).

target\_scenarios    A character vector of scenario names

target\_rule\_ids     A list of rule id.

show\_all\_scenarios    Set TRUE to show all scenarios.

show\_all\_rule\_ids    Set TRUE to show all rules.

debug                Set TRUE if show intermediate dataframe using View function.

**Value**

A list of bar plots.

**Examples**

```
mipplot_additivity_check_bar(
  ar5_db_sample_data, ar5_db_sample_rule_table,
  target_scenarios = c("EMF27-450-Conv", "EMF27-Base-NucOff"))
```

---

mipplot\_area

*Area plot from IAMC data*

---

**Description**

Area plots using right-hand-side values of target additivity rule. The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions (two in this version).

**Usage**

```
mipplot_area(
  D,
  R,
  region = levels(D$region),
  scenario = levels(D$scenario),
  facet_x = NULL,
  facet_y = NULL,
  PRINT_OUT = FALSE,
  DEBUG = TRUE,
  fontsize = 20,
  color_code_specify = TRUE,
  one_hundred_percent_stacked = FALSE,
  axis_year_text_angle = 0,
  language = "en"
)
```

**Arguments**

D	A dataframe of IAMC data in tibble format to produce area plots.
R	A dataframe of data aggregation rules (meta data).
region	A list of regions.
scenario	A list of scenario.
facet_x	facet_x
facet_y	facet_y
PRINT_OUT	set TRUE to generate PDF file.
DEBUG	set TRUE to show debug messages.
fontsize	font size of text.
color_code_specify	set FALSE if you apply default color palette.
one_hundred_percent_stacked	set TRUE if you want a graph of 100% stacked, set this to TRUE.
axis_year_text_angle	text angle of x axis
language	A string of language. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

A list of area plots.

**Examples**

```
library(dplyr)
data_subset <- ar5_db_sample_data %>%
  filter(variable == "Emissions|CO2|Land Use") %>%
```

```

filter(model %in% c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4")) %>%
filter(2005 <= period) %>%
filter(period <= 2100)
mipplot_area(data_subset, ar5_db_sample_rule_table,
region = c("ASIA"),
scenario = c("EMF27-450-Conv"),
one_hundred_percent_stacked = FALSE,
axis_year_text_angle = 0,
language = 'en')

```

---

mipplot\_autofill\_color

*Complementation of color scheme*


---

### Description

fill colors automatically

### Usage

```
mipplot_autofill_color(rule_table_without_colors)
```

### Arguments

rule\_table\_without\_colors

Incomplete color specification rule table. It doesn't contain "Color\_code" column.

### Value

Complete color specification rule table. It is containing "Color\_code" column. However, if color complementation can not be performed automatically, the return value is an incomplete color specification.

---

mipplot\_bar

*Bar plot from IAMC data*


---

### Description

Bar plots using right-hand-side values of target additivity rule. The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions.

**Usage**

```
mipplot_bar(
  D,
  R,
  region = levels(D$region),
  xby = "scenario",
  target_year = levels(as.factor(D$period)),
  facet_x = NULL,
  facet_y = NULL,
  PRINT_OUT = FALSE,
  DEBUG = TRUE,
  fontsize = 20,
  color_code_specify = TRUE,
  one_hundred_percent_stacked = FALSE,
  axis_scenario_text_angle = 0,
  language = "en"
)
```

**Arguments**

D	A dataframe of IAMC data in tibble format to produce plots.
R	A dataframe of data aggregation rules (meta data).
region	A list of region.
xby	name of axis. the default setting is "scenario".
target_year	target year.
facet_x	facet_x
facet_y	facet_y
PRINT_OUT	set TRUE to generate A PDF file.
DEBUG	set TRUE to show debug messages.
fontsize	size of font in the output plot.
color_code_specify	set FALSE if you apply default color palette.
one_hundred_percent_stacked	set TRUE if you want a graph of 100% stacked, set this to TRUE.
axis_scenario_text_angle	text angle of x axis
language	A string of language. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

A list of bar plots.

**Examples**

```

library(dplyr)
data_subset <- ar5_db_sample_data %>%
  filter(variable == "Emissions|CO2|Land Use") %>%
  filter(model %in% c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4")) %>%
  filter(scenario %in% c("EMF27-450-Conv", "EMF27-450-FullTech"))

mipplot_bar(data_subset, ar5_db_sample_rule_table,
  region = c("ASIA"),
  target_year = 2005,
  one_hundred_percent_stacked = FALSE,
  axis_scenario_text_angle = 0,
  language = 'en')

```

---

mipplot\_box

*Box plot from IAMC data*


---

**Description**

The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions

**Usage**

```

mipplot_box(
  D,
  region = levels(D$region),
  variable = levels(D$variable),
  target_year = levels(as.factor(D$period)),
  PRINT_OUT = FALSE,
  DEBUG = TRUE,
  language = "en"
)

```

**Arguments**

D	A dataframe of IAMC data in tibble format to produce plots.
region	A list of regions.
variable	A list of variables.
target_year	target year.
PRINT_OUT	set TRUE to generate PDF file.
DEBUG	set TRUE to show debug messages.
language	A string of language. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

A list of box plots.

**Examples**

```
library(dplyr)
data_subset <- ar5_db_sample_data %>%
  filter(variable == "Emissions|CO2|Land Use") %>%
  filter(model %in% c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4")) %>%
  filter(period == 2100) %>% filter(region == "OECD90")
mipplot_box(data_subset)
```

---

mipplot\_default\_color\_palette

*Default color palette.*

---

**Description**

Default color palette.

**Usage**

```
mipplot_default_color_palette
```

**Format**

A default color palette object, which maps variable name (such as "Land Use") to hex color code.

---

mipplot\_generate\_color\_mapper

*Manual coloring*

---

**Description**

Generate mapper from name of variable to name of color

**Usage**

```
mipplot_generate_color_mapper(raw_table, category_separator = "\\|")
```

**Arguments**

`raw_table` rule table which includes "Color\_code" column.

`category_separator`

regular expression for separating right-hand-side variable name into categories.  
For example: separator should be "|" for "Secondary Energy|Electricity|Coal"

**Value**

named list of named string vectors. for example,

```
result = list( "Emissions|CO2" = c( "Fossil Fuels and Industry" = "#17202a", "Land Use" = "#008000",
...), "Emissions|CO2|Fossil Fuels and Industry" = c( "Energy Demand" = "#444444", ... ),...
```

---

mipplot\_interactive\_additivity\_check\_bar

*A function to launch interactive plot using Shiny*

---

**Description**

A function to launch interactive plot for additivity check.

**Usage**

```
mipplot_interactive_additivity_check_bar(D, R, debug = FALSE)
```

**Arguments**

D	A quitte format dataframe of IAMC data to produce graph.
R	A table with additivity rules.
debug	Set TRUE if table view is required.

**Value**

No return value, called for side effects

**Examples**

```
if (interactive()) {
  mipplot_interactive_additivity_check_bar(ar5_db_sample_data, ar5_db_sample_rule_table)
}
```

---

`mipplot_interactive_area`*A function to launch interactive plotting session on Shiny*

---

**Description**

Provides gui to set plotting parameter for area plot.

**Usage**

```
mipplot_interactive_area(D, R, language = "en")
```

**Arguments**

D	A dataframe of IAMC data in tibble format to produce area plots.
R	A dataframe of data aggregation rules (meta data).
language	A string of language for initial plot. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

No return value, called for side effects

**Examples**

```
if (interactive()) {  
  mipplot_interactive_area(ar5_db_sample_data, ar5_db_sample_rule_table)  
}
```

---

`mipplot_interactive_bar`*A function to launch interactive plot using Shiny*

---

**Description**

A function to launch interactive bar plot using right-hand-side values of target additivity rule. The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions

**Usage**

```
mipplot_interactive_bar(D, R, language = "en")
```

**Arguments**

D	A quitte format dataframe of IAMC data to produce graph.
R	A table with additivity rules.
language	A string of language for initial plot. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

No return value, called for side effects

**Examples**

```
if (interactive()) {  
  mipplot_interactive_bar(ar5_db_sample_data, ar5_db_sample_rule_table)  
}
```

---

mipplot\_interactive\_line

*A function to launch interactive plot using Shiny*

---

**Description**

A function to launch interactive line plot. The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions

**Usage**

```
mipplot_interactive_line(D, language = "en")
```

**Arguments**

D	A quitte format dataframe of IAMC data to produce graph.
language	A string of language for initial plot. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".

**Value**

No return value, called for side effects

**Examples**

```
if (interactive()) {  
  mipplot_interactive_line(ar5_db_sample_data)  
}
```

---

 mipplot\_interactive\_plot\_line

*A function to launch interactive plot using Shiny*


---

### Description

A function to launch interactive plot using Shiny

### Usage

```
mipplot_interactive_plot_line(D, R)
```

### Arguments

D                    A quitte format dataframe of IAMC data to produce graph.  
 R                    A table with additivity rules.

### Value

No return value, called for side effects

### Examples

```
if (interactive()) {
  mipplot_interactive_plot_line(ar5_db_sample_data, ar5_db_sample_rule_table)
}
```

---

 mipplot\_line

*Line plot from IAMC data*


---

### Description

The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions

### Usage

```
mipplot_line(
  D,
  region = levels(D$region),
  variable = levels(D$variable),
  colorby = "scenario",
  linetypeby = "model",
  shapeby = "model",
  scenario = levels(D$scenario),
```

```

facet_x = NULL,
facet_y = NULL,
legend = TRUE,
PRINT_OUT = FALSE,
DEBUG = TRUE,
axis_year_text_angle = 0,
language = "en",
max_scenarios = 15,
max_models = 15
)

```

### Arguments

D	A dataframe of IAMC data in tibble format to produce plots.
region	A list of regions.
variable	A list of variables.
colorby	an axis for color setting.
linetypeby	an axis for line type setting.
shapeby	an axis for shape setting.
scenario	A list of scenarios.
facet_x	facet_x
facet_y	facet_y
legend	set TRUE to plot legend. default is TRUE.
PRINT_OUT	set TRUE to generate PDF files.
DEBUG	set TRUE to show debug messages.
axis_year_text_angle	text angle of x axis
language	A string of language. Possible values are "en", "jp", "es", "zh-cn", "zh-tw". The default value is "en".
max_scenarios	Maximum number of scenarios to be shown. If legend is FALSE, this option is .
max_models	Maximum number of models to be shown. If legend is FALSE, this option is

### Value

A list of line plots.

### Examples

```

library(dplyr)
data_subset <- ar5_db_sample_data %>%
  filter( model %in% c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4") ) %>%
  filter(2005 <= period) %>%
  filter(period <= 2100)
mipplot_line(
  data_subset,

```

```

variable = c("Emissions|CO2"),
scenario = c("EMF27-450-Conv", "EMF27-450-FullTech", "EMF27-450-NoCCS"),
region = c("ASIA"),
legend = TRUE,
axis_year_text_angle = 0,
language = 'en')

```

---

mipplot\_point

*Point plot from IAMC data*


---

### Description

The function arguments include the input dataframe, labels for the plot/axes/legend, and faceting dimensions

### Usage

```

mipplot_point(
  D,
  region = levels(D$region),
  variable = levels(D$variable),
  target_year = levels(as.factor(D$period)),
  colorby = "model",
  shapeby = "model",
  xby = "scenario",
  facetby = NULL,
  facet_x = NULL,
  facet_y = NULL,
  fontsize = 20,
  PRINT_OUT = FALSE,
  DEBUG = TRUE
)

```

### Arguments

D	A dataframe of IAMC data in tibble format to produce plots.
region	A list of regions.
variable	A list of variables.
target_year	A list of target years.
colorby	An axis for color setting.
shapeby	An axis for shape setting.
xby	An axis for x locating setting.
facetby	facetby.
facet_x	facet_x.

facet_y	facet_y.
fontsize	font size.
PRINT_OUT	set TRUE to generate PDF image.
DEBUG	set TRUE to show debug messages.

**Value**

A list of point plots.

**Examples**

```
library(dplyr)
data_subset <- ar5_db_sample_data %>%
  filter(variable == "Emissions|CO2|Land Use") %>%
  filter(model %in% c("AIM-Enduse 12.1", "GCAM 3.0", "IMAGE 2.4")) %>%
  filter(period == 2100) %>% filter(region == "OECD90")
mipplot_point(data_subset)
```

---

mipplot\_print\_pdf      *Print list of plots to pdf file*

---

**Description**

This function plots a ggplot plots to PDF file.

**Usage**

```
mipplot_print_pdf(
  p_list1,
  filelabel = "",
  filename = tryCatch(file.choose(new = TRUE), error = function(e) { NA })
)
```

**Arguments**

p_list1	A list of ggplot plot.
filelabel	A string of prefix of output filename.
filename	A string of filename. If it is given, filelabel is ignored.

**Value**

No return value, called for side effects

## Examples

```
if (interactive()) {  
  p <- mipplot_area(ar5_db_sample_data, ar5_db_sample_rule_table,  
                  region = "World", scenario = "EMF27-450-FullTech")  
  mipplot_print_pdf(p)  
}
```

---

mipplot_read_iamc	<i>Read IAMC scenario input data.</i>
-------------------	---------------------------------------

---

## Description

Read scenario input data (in IAMC format) as tibble format dataframe.

## Usage

```
mipplot_read_iamc(  
  filename = NULL,  
  sep = ",",  
  interactive = FALSE,  
  DEBUG = TRUE  
)
```

## Arguments

filename	Path to a file containing scenario data in IAMC format.
sep	A character indicating the separator used in the input file.
interactive	open a dialog for selecting file if interactive=TRUE.
DEBUG	experimental.

## Value

A dataframe in tibble format ("model, scenario, variable, unit, period, value")

## Examples

```
## Not run:  
mipplot_read_iamc("filename")  
  
## End(Not run)
```

---

mipplot\_read\_ruletab *Read file of rule table without ID number*

---

**Description**

Read table of additivity rule and adds column with id number.

**Usage**

```
mipplot_read_ruletab(R_without_id)
```

**Arguments**

R\_without\_id Path to a file containing data of additivity rule.

**Value**

A dataframe of additivity rule ("ID, Left\_side, Right\_side")

**Examples**

```
## Not run:  
mipplot_read_ruletab("filename")  
  
## End(Not run)
```

---

mipplot\_return\_table *Mutated table of SR15 Data*

---

**Description**

Mutated Table using filtered variable from the rule table The function arguments include the input dataframes: The SR15 dataset and the Rule Table and returns a mutated table with variable, value, model, scenario, region, period

**Usage**

```
mipplot_return_table(D, R)
```

**Arguments**

D A dataframe of IAMC data in tibble format to produce mutated table  
R A dataframe of data aggregation rules

**Value**

Mutated Table of model,scenario,region,variable,unit,period,value

**Examples**

```
mipplot_return_table(sr15_sample_data, sr15_sample_conversion_rule_table)
```

---

```
mipplot_var_submission
      variable SUBMISSION CHECK
```

---

**Description**

Verify whether data of variables included in list template have been submitted.

**Usage**

```
mipplot_var_submission(D, V, na_name = "N/A")
```

**Arguments**

D	input data table
V	list of variables
na_name	string for N/A

**Value**

A dataframe representing variable availabilities.

---

```
read_iamc_xlsx      Read IAMC scenario input data in Excel format
```

---

**Description**

Read scenario input data (in IAMC format) as tibble format dataframe from Excel

**Usage**

```
read_iamc_xlsx(file_path, sheet = 2)
```

**Arguments**

file_path	Path to a file containing scenario data in IAMC format.
sheet	the index of sheet which contains records.

**Value**

A dataframe in tibble format ("model, scenario, variable, unit, period, value")

### Examples

```
## Not run:  
read_iamc_xlsx("filename", sheet = 2)  
  
## End(Not run)
```

---

```
split_variable_into_positive_and_negative_parts  
    Split variable into positive and negative parts
```

---

### Description

Generally, the range of the input value of stacked chart is greater than or equal to zero. This function splits variable into positive and negative parts in order to include negative values to stacked chart.

### Usage

```
split_variable_into_positive_and_negative_parts(  
  df_all,  
  domain_column_name,  
  variable_column_name,  
  value_column_name,  
  variable_name_converter = function(x) { paste(x, "_negative", sep = "") },  
  increment_of_domain_in_interpolation = 0.1  
)
```

### Arguments

df_all	input data frame
domain_column_name	domain column name, such as year
variable_column_name	variable column name, such as 'coal'
value_column_name	value column name, such as 'val'
variable_name_converter	function which convert original variable name into its negative part name
increment_of_domain_in_interpolation	step size for interpolation

### Value

modified data frame

---

sr15\_sample\_conversion\_rule\_table  
*Sample Conversion Rule Table*

---

**Description**

A sample conversion rule table for mipplot\_return\_table.

**Usage**

sr15\_sample\_conversion\_rule\_table

**Format**

An object of class data.frame with 37 rows and 6 columns.

---

sr15\_sample\_data      *Sample Dataset*

---

**Description**

A sample dataset of IAMC format consist of a subset of IPCC special report (Global Warming of 1.5°C, 2018).

**Usage**

sr15\_sample\_data

**Format**

A tibble data.table with 396425 rows and 7 variables:

**model** model, categorical

**scenario** scenario, categorical

**region** region, ASIA, OECD90 or World

**variable** the name of simulated variable that changes over time

**unit** unit of a variable

**period** year

**value** the value of a variable

**Source**

<https://data.ene.iiasa.ac.at/iamc-1.5c-explorer/>

# Index

## \* datasets

- ar5\_db\_sample\_data, 4
- ar5\_db\_sample\_rule\_table, 4
- mipplot\_default\_color\_palette, 18
- sr15\_sample\_conversion\_rule\_table, 30
- sr15\_sample\_data, 30
  
- add\_credit\_to\_list\_of\_plot, 3
- add\_credit\_to\_plot, 3
- ar5\_db\_sample\_data, 4
- ar5\_db\_sample\_rule\_table, 4
  
- change\_data\_types\_of\_iamc\_dataframe, 5
- check\_column\_availability, 5
- check\_format\_of\_iamc\_dataframe, 6
- correct\_format\_of\_iamc\_dataframe, 6
  
- generate\_code\_to\_plot\_area, 7
- generate\_code\_to\_plot\_bar, 7
- generate\_code\_to\_plot\_line, 8
- get\_model\_name\_list, 9
- get\_scenario\_name\_list, 9
- get\_string\_expression\_of\_vector\_of\_strings, 10
- get\_variable\_group\_name\_list, 10
- get\_variable\_name\_list\_in\_variable\_group, 11
  
- mipplot, 11
- mipplot\_additivity\_check, 12
- mipplot\_additivity\_check\_bar, 12
- mipplot\_area, 13
- mipplot\_autofill\_color, 15
- mipplot\_bar, 15
- mipplot\_box, 17
- mipplot\_default\_color\_palette, 18
- mipplot\_generate\_color\_mapper, 18
- mipplot\_interactive\_additivity\_check\_bar, 19
  
- mipplot\_interactive\_area, 20
- mipplot\_interactive\_bar, 20
- mipplot\_interactive\_line, 21
- mipplot\_interactive\_plot\_line, 22
- mipplot\_line, 22
- mipplot\_point, 24
- mipplot\_print\_pdf, 25
- mipplot\_read\_iamc, 26
- mipplot\_read\_rulatab, 27
- mipplot\_return\_table, 27
- mipplot\_var\_submission, 28
  
- read\_iamc\_excel, 28
  
- split\_variable\_into\_positive\_and\_negative\_parts, 29
- sr15\_sample\_conversion\_rule\_table, 30
- sr15\_sample\_data, 30