Package ‘kidney.epi’

March 4, 2020

Title Kidney Functions: Clinical and Epidemiological
Version 1.2.0
Maintainer Boris Bikbov <boris@bikbov.ru>
Description Contains kidney care oriented functions.
   Current version contains functions for calculation of:
   - Kidney Donor Risk Index and Kidney Donor Profile Index for kidney trans-
   - Estimated glomerular filtration rate by CKD-EPI, MDRD and other equations.
   Citation: Bikbov B. R open source programming code for calculation of the Kidney Donor Pro-
Depends R (>= 3.4.0)
License LGPL (>= 2)
URL http://kidneyepidemiology.org/r/
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Boris Bikbov [aut, cre] (<https://orcid.org/0000-0002-1925-7506>)
Repository CRAN
Date/Publication 2020-03-04 16:10:06 UTC

R topics documented:

   egfr.ckdepi ................................................................. 2
   egfr.mdrd4 ................................................................. 3
   egfr.schwartz ............................................................ 5
   kidney.epi ............................................................... 6
Description

CKD-EPI

Usage

```r
egfr.ckdepi(creatinine, age, sex, ethnicity,
    creatinine_units = "micromol/L",
    label_afroamerican = c("Afroamerican"),
    label_sex_male = c("Male", 1),
    label_sex_female = c("Female", 0))
```

Arguments

- **creatinine**: Numeric vector. Serum creatinine, could be expressed in "micromol/L", "mmol/L" or "mg/dL". Units of measurement should be defined in variable creatinine_units (if not defined explicitly by user, the default value is "micromol/L").
- **age**: Numeric vector. Age, in years.
- **sex**: Vector. The value of variable refers to the parameters label_sex_male and label_sex_female.
- **ethnicity**: Vector. Ethnicity, specify in case of African-American patients. The value of variable refers to the parameter label_afroamerican.
egfr.mdrd4

creatinine_units
Character string. Units in which serum creatinine is expressed. Could be one of the following: "micromol/L", "mmol/L", or "mg/dL".

label_afroamerican
List. Label(s) for Afroamerican ethnicity.

label_sex_male
List. Label(s) for definition(s) of male sex.

label_sex_female
List. Label(s) for definition(s) of female sex.

Details

Programming: Boris Bikbov <boris@bikbov.ru>. Citation: Bikbov B. R open source programming code for calculation of the Kidney Donor Profile Index and Kidney Donor Risk Index. Kidney Diseases, 2018. DOI: 10.1159/000492427 (citation for the whole package)

Value
numeric eGFR expressed in ml/min/1.73m<sup>2</sup>.

Examples

egfr.ckdepi (creatinine = 1.4, age = 60, sex = "Male", ethnicity = "White",
creatinine_units = "mg/dl")

egfr.mdrd4

Calculate estimated glomerular filtration rate (eGFR) by different equations

Description
MDRD

Usage

egfr.mdrd4(creatinine, age, sex, ethnicity, 
creatinine_units = "micromol/l", creatinine_method = "non-IDMS",
label_afroamerican = c("Afroamerican"), label_sex_male = c("Male", 1), label_sex_female = c("Female", 0))
Arguments

creatinine Numeric vector. Serum creatinine, could be expressed in "micromol/L", "mmol/L" or "mg/dL". Units of measurement should be defined in variable creatinine_units (if not defined explicitly by user, the default value is "micromol/L").

age Numeric vector. Age, in years.

sex Vector. The value of variable refers to the parameters label_sex_male and label_sex_female.


creatinine_units Character string. Units in which serum creatinine is expressed. Could be one of the following: "micromol/L", "mmol/L" or "mg/dL".

creatinine_method Character string. Creatinine standardisation method in a laboratory. Could be either "IDMS" or "non-IDMS". If not explicitly defined by user, the default assumption is "non-IDMS".

label_afroamerican List. Label(s) for Afroamerican ethnicity.

label_sex_male List. Label(s) for definition(s) of male sex.

label_sex_female List. Label(s) for definition(s) of female sex.

Details


Programming: Boris Bikbov <boris@bikbov.ru>. Citation: Bikbov B. R open source programming code for calculation of the Kidney Donor Profile Index and Kidney Donor Risk Index. Kidney Diseases, 2018. DOI: 10.1159/000492427 (citation for the whole package)

Value

decimal eGFR expressed in ml/min/1.73m<sup>2</sup>.

Examples

egfr.mdrd4 (creatinine = 1.4, age = 60, sex = "Male", ethnicity = "White", creatinine_units = "mg/dl")
Schwartz (for children only)

Usage

```r
egrf.schwartz(creatinine, age, sex, height_cm = 0, height_ft = 0, height_inch = 0, creatinine_units = "micromol/l", equation_type = "classic", label_sex_male = c("Male", 1), label_sex_female = c("Female", 0))
```

Arguments

- **creatinine**: Numeric vector. Serum creatinine, could be expressed in "micromol/L", "mmol/L" or "mg/dL". Units of measurement should be defined in variable creatinine_units (if not defined explicitly by user, the default value is "micromol/L").
- **age**: Numeric vector. Age, in years. Age does not accounted in Schwartz equation, but used in the function to check whether Schwartz equation could be applied to a given patient.
- **sex**: Vector. The value of variable refers to the parameters label_sex_male and label_sex_female. Required only in case of quadratic Schwartz equation.
- **height_cm**: Numeric vector. Could be defined either as height_cm if is measured in cm, or as height_ft and height_inch if is measured in feet and inches. If the parameter height_cm is greater than 0, the function uses cm, otherwise - feet and inches.
- **height_ft**: see height_cm
- **height_inch**: see height_cm
- **creatinine_units**: Character string. Units in which serum creatinine is expressed. Could be one of the following: "micromol/L", "mmol/L" or "mg/dL".
- **equation_type**: Character string. Define whether to calculate eGFR either by classic Schwartz or quadratic Schwartz equation. Could be one of the following: "classic", "quadratic". If not explicitly defined by user, the default assumption is "classic".
- **label_sex_male**: List. Label(s) for definition(s) of male sex.
- **label_sex_female**: List. Label(s) for definition(s) of female sex.
Details


Programming: Boris Bikbov <boris@bikbov.ru>. Citation: Bikbov B. R open source programming code for calculation of the Kidney Donor Profile Index and Kidney Donor Risk Index. Kidney Diseases, 2018. DOI: 10.1159/000492427 (citation for the whole package)

Value

numeric eGFR expressed in ml/min/1.73m<sup>2</sup>.

Examples

```r
egfr.schwartz (creatinine = 1.4, age = 10, height_cm = 90, sex = "Male", creatinine_units = "mg/dl")

egfr.schwartz (creatinine = 1.4, age = 10, height_cm = 90, sex = "Male", creatinine_units = "mg/dl", equation_type = "quadratic")
```

kidney.epi  

*Kidney functions for R: clinical and epidemiological*

Description

Package contains different functions for use in the field of kidney disease and general epidemiology. Current version contains functions for calculation of estimated GFR by different equations, and calculation of KDPI and KDRI for kidney transplant donors. More functions are coming soon.

ktx  

*Sample dataset with kidney transplant patients.*

Description

A dataset containing 10 records for kidney transplant patients, including information for deceased donors.

Usage

```
ktx
```
Format

A data frame with 10 rows and 12 variables:

- **ptid**: patient identifier
- **rec.age**: age of the recipient, in years
- **don.age**: age of the donor, in years
- **don.height**: height of the donor, in cm
- **don.weight**: weight of the donor, in kg
- **don.ethnicity**: ethnicity of the donor
- **don.hypertension**: history of hypertension for the donor
- **don.diabetes**: history of diabetes for the donor
- **don.causeofdeath**: cause of death for the donor
- **don.creatinine**: serum creatinine of the donor, in mg/dL
- **don.hcv**: hepatitis c virus status of the donor
- **don.dcdstatus**: donation after circulatory death status of the donor
- **don.sex**: sex of the donor

Source

Generation from different patients’ records

**ktx.kdpi.optn**

Calculate KDRI and KDPI for deceased kidney donor

Description

Calculate KDRI and KDPI for deceased kidney donor

Usage

```r
ktx.kdpi.optn(age, height_cm = 0, height_ft = 0, height_inch = 0,
weight_kg = 0, weight_lb = 0, ethnicity, hypertension, diabetes,
causeofdeath, creatinine, hcv, dcdstatus,
c creatinine_units = "micromol/l", return_output_type = "KDPI",
mapping_values_year = "latest",
label_afroamerican = c("Afroamerican"),
label_hypertension_positive = c("yes"),
label_hypertension_unknown = "NA",
label_diabetes_positive = c("yes"), label_diabetes_unknown = "NA",
label_causeofdeath = c("cva"), label_hcv_positive = c("positive"),
label_hcv_unknown = "NA", label_dcdstatus = c("yes"))
```
Arguments

**age**
Numeric vector. Age, in years.

**height_cm**
Numeric vector. Could be defined either as height_cm if measured in cm, or as height_ft and height_inch if measured in feet and inches. If the parameter height_cm is greater than 0, the function uses cm, otherwise - feet and inches.

**height_ft**
See height_cm

**height_inch**
See height_cm

**weight_kg**
Numeric vector. Could be defined either as weight_kg if measured in kg, or as weight_lb if measured in pounds. If the parameter weight_kg is greater than 0, the function uses kg, otherwise - pounds.

**weight_lb**
See weight_kg

**ethnicity**
Vector. Ethnicity, specify in case of African-American donors which have special coefficient in the regression equation. The value of variable refers to the parameter label_afroamerican.

**hypertension**
Vector. History of hypertension, specify in case of hypertensive donors which have special coefficient in the regression equation. The value of variable refers to the parameters label_hypertension_positive and label_hypertension_unknown.

**diabetes**
Vector. History of diabetes, specify in case of donors with diabetes which have special coefficient in the regression equation. The value of variable refers to the parameters label_diabetes_positive and label_diabetes_unknown.

**causeofdeath**
Vector. Cause of death, specify whether death was due to cerebrovascular disease, or other reasons.

**creatinine**
Numeric vector. Serum creatinine, could be expressed in "micromol/L", "mmol/L" or "mg/dL". Units of measurement should be defined in variable creatinine_units (if not defined explicitly by user, the default value is "micromol/L").

**hcv**
Vector. Hepatitis C virus status. The value of variable refers to the parameters label_hcv_positive and label_hcv_unknown.

**dcdstatus**
Vector. Donation after circulatory death status. Specify whether organ was from a donor after circulatory death or not. The value of variable refers to the parameter label_dcdstatus.

**creatinine_units**
Character string. Units in which serum creatinine is expressed. Could be one of the following: "micromol/L", "mmol/L" or "mg/dL".

**return_output_type**
Character string. Specify which calculated parameter to return from the function: "KDRI_Rao" - Raw Kidney Donor Risk Index, "KDRI_median" - scaled to the median Kidney Donor Risk Index, or "KDPI" - Kidney Donor Profile Index.

**mapping_values_year**
Numeric value or character string. Specify which year to take for the OPTN mapping table, as well as KDRI scaling factor and chances of hypertension and diabetes in case if they were unknown for donor.

By default the value is "latest", and the function takes the latest available OPTN mapping table and coefficients from the internal dataframes ktx.kdpi_mapping_table and ktx.kdpi_coefficients_table.
But if necessary, a user could define the exact year (i.e. mapping_values_year = 2015).
For a list of available years run the following: ktx.kdpi.optn.show.years().

- **label_afroamerican**: List. Label(s) for Afroamerican ethnicity.
- **label_hypertension_positive**: List. Label(s) for a positive history of hypertension.
- **label_hypertension_unknown**: List. Label(s) for donors with unknown history of hypertension.
- **label_diabetes_positive**: List. Label(s) for a positive history of diabetes.
- **label_diabetes_unknown**: List. Label(s) for donors with unknown history of diabetes.
- **label_causeofdeath**: List. Label(s) for a cause of death due to cerebrovascular/stroke.
- **label_hcv_positive**: List. Label(s) for a positive HCV status.
- **label_hcv_unknown**: List. Label(s) for an unknown, not done, indeterminate, or pending HCV status.
- **label_dcdstatus**: List. Label(s) for a donor after circulatory death status.

**Details**

Calculate Kidney Donor Risk Index (KDRI) and Kidney Donor Profile Index (KDPI) based on the algorithm of US Organ Procurement and Transplantation Network. The Kidney Donor Profile Index (KDPI) is a numerical measure that combines ten donor factors to summarize into a single number the quality of deceased donor kidneys relative to other recovered kidneys. *KDRI could be calculated only for a deceased donor!*

More reading:

- OPTN web-based calculator
- Guide to calculating and interpreting KDPI
- Latest data for mapping table, scaling factor, etc

Programming: Boris Bikbov <boris@bikbov.ru>.

Citation: Bikbov B. R open source programming code for calculation of the Kidney Donor Profile Index and Kidney Donor Risk Index. Kidney Diseases, 2018. DOI: 10.1159/000492427

**Value**

numeric One of the following values based on the return_output_type argument: Raw Kidney Donor Risk Index (KDRI), Scaled to the median Kidney Donor Risk Index (KDRI), or Kidney Donor Profile Index (KDPI).
service.check_obligatory_params

Examples

ktx.kdpi.optn (age = 60, height_cm = 168, weight_kg = 93, ethnicity = "White",
hypertension = "yes", diabetes = "no", causeofdeath = "roadinjury",
creatinine = 1.4, hcv = "negative", dcdstatus = "no",
creatinine_units = "mg/dl", return_output_type = "KDRI_Rao")
ktx.kdpi.optn (age = 30, height_cm = 176, weight_kg = 82, ethnicity = "White",
hypertension = "NA", diabetes = "no", causeofdeath = "roadinjury",
creatinine = 150, hcv = "negative", dcdstatus = "no", return_output_type = "KDPI")

ktx.kdpi.optn.show.years

Shows which years are available in the R package for the OPTN mapping table, KDRI scaling factor, etc.

Description

Shows which years are available in the R package for the OPTN mapping table, KDRI scaling factor, etc.

Usage

ktx.kdpi.optn.show.years()

Details

Service function which shows for user for which year(s) the OPTN mapping table, as well as KDRI scaling factor and chances of hypertension and diabetes in case if they were unknown for donor in the ktx.kdpi_mapping_table and ktx.kdpi_coefficients_table. This years could be used for the argument mapping_values_year of the ktx.kdpi.optn function.

This function has no arguments.

Value

numeric List of years which could be used for the argument mapping_values_year of the ktx.kdpi.optn function.

service.check_obligatory_params

Check whether all obligatory parameteres of a given function are present.

Description

Check whether all obligatory parameteres of a given function are present.
usage

\texttt{service.check_obligatory_params(fx_params, args, predefined_result = TRUE)}

\textbf{Arguments}

- \texttt{fx_params} List. List of parameters required by function.
- \texttt{args} List. Arguments transferred to the function upon user call.
- \texttt{predefined_result} Logical. Required only in case if other checks were performed in the main script and the result of this check has to be processed to the function. For example, if in the parent script I’ve checked the presence of height parameter, and it is absent (while is obligatory), I transfer this info in the "predefined_result = FALSE", so in the function the \texttt{fx_params_resulting} become False and will lead to stop().

\textbf{Details}

Check whether all obligatory parameteres of a given function are present.

Programming: Boris Bikbov <boris@bikbov.ru>.

\textbf{Value}

Character string. Returns a messages and stops function if any of the obligatory parameters are absent.

\textbf{Examples}

\begin{verbatim}
# could be run only inside function wich receives some parameters
# fx_params <- c("creatinine", "age", "ethnicity", "sex")
# args <- names(as.list(match.call())[-1])
# service.check_obligatory_params(fx_params, args)
\end{verbatim}

---

\texttt{service.check_params_numeric}

\textit{Check whether the following variables are numeric and stop function if at least one of them is not numeric}

\textbf{Description}

Check whether the following variables are numeric and stop function if at least one of them is not numeric

\textbf{Usage}

\texttt{service.check_params_numeric(...)}
service.check_param_arguments

Arguments

... Argument list. Argument list (arbitrary number of variables) with data to check.
nothing to return

Details

Check whether the following variables are numeric and stop function if at least one of them is not numeric. Service function that will not be exported to user.
Programming: Boris Bikbov <boris@bikbov.ru>.

service.check_param_arguments

Check the argument of a given parameter which set by user and stop function if the value set by user is not among the possible values of the argument

Description

Check the argument of a given parameter which set by user and stop function if the value set by user is not among the possible values of the argument

Usage

service.check_param_arguments(param2check, possible_params,
custom_message = "")

Arguments

param2check List, Character string, Number. Parameter used in a function.
possible_params List. List of possible values of the parameter arguments
custom_message Character string. Custom message to be output. If not defined, the standard output message is provided. nothing to return

Details

Check the argument of a given parameter which set by user and stop function if the value set by user is not among the possible values of the argument. Service function that will not be exported to user.
Programming: Boris Bikbov <boris@bikbov.ru>.
**service.check_param_number**

*Check number of parameters and stop function if it exceeds the expected number of parameters*

---

**Description**

Check number of parameters and stop function if it exceeds the expected number of parameters.

**Usage**

```python
service.check_param_number(param2check, acceptable_number = 1,
custom_message = "")
```

**Arguments**

- `param2check`: List, Character string, Number. Parameter used in a function.
- `acceptable_number`: Numeric. Acceptable number of arguments in the list param2check (by default is "1")
- `custom_message`: Character string. Custom message to be output. If not defined, the standard output message is provided. Nothing to return.

**Details**

Check number of parameters and stop function if it exceeds the expected number of parameters. Service function that will not be exported to user.

Programming: Boris Bikbov <boris@bikbov.ru>.

---

**service.check_plausibility.age**

*Service functions for data check on biological plausibility and biochemistry conversion which could be applied in any function of the package or externally*

---

**Description**

Check and modify if necessary the age values.

**Usage**

```python
service.check_plausibility.age(age)
```

**Arguments**

- `age`: Numeric. The value to be checked.
Details
Service function which check whether age is in biologically plausible boundaries, shows to user
warnings if any, and substitute unplausible values.

Value
numeric Vector with controlled values.

---

service.check_plausibility.creatinine

Check and modify if necessary the creatinine values.

Description
Check and modify if necessary the creatinine values.

Usage
service.check_plausibility.creatinine(creatinine)

Arguments
creatinine Numeric. The value to be checked.

Details
Service function which check whether creatinine is in biologically plausible boundaries, shows to
user warnings if any, and substitute unplausible values.

Value
numeric Vector with controlled values.

---

service.convert_creatinine

Convert creatinine values if necessary (depending on the measurement units).

Description
Convert creatinine values if necessary (depending on the measurement units).

Usage
service.convert_creatinine(creatinine, creatinine_units)
service.count_greater_threshold

Arguments

creatinine Numeric. The creatinine value from data set.
creatinine_units Character. Creatinine measurement units defined by user.

Details

Service function which check measurement units and convert creatinine values if necessary.

Value

numeric Vector with converted values.

Example

```r
myvals <- c(1, 8, -5, "oggi", NA)
myvals2 <- service.count_greater_threshold(myvals, 0)
myvals2 # 2
```
service.count.lowerequal_threshold

Count how many values are less or equal than the defined threshold.

Description

Count how many values are less or equal than the defined threshold.

Usage

service.count.lowerequal_threshold(x, threshold)

Arguments

x the vector to be checked.
threshold numeric the threshold to compare with.

Details

Count how many values are less or equal than the defined threshold.

Programming: Boris Bikbov <boris@bikbov.ru>.

Value

numeric returns number of numeric values less or equal to the threshold.

Examples

myvals <- c(1, 8, -5, "oggi", NA)
myvals2 <- service.count.lowerequal_threshold(myvals, 0)
myvals2 # 1

service.is.param_possible

Service functions for data check which could be applied in any function of the package or externally

Description

Service functions for data check which could be applied in any function of the package or externally

Usage

service.is.param_possible(param2check, possible_params)
**service.is_numeric**

**Arguments**

- **param2check**: Numeric value or character string. The single value to be verified.
- **possible_params**: Vector. The vector of values which contains all possible values.

**Details**

Verifies whether the single value is among the values of the vector. Function is useful to check whether the argument of the function defined by the user is among the possible arguments recognized inside the function.

Programming: Boris Bikbov <boris@bikbov.ru>.

**Value**

logic returns TRUE if argument param2check is found in possible values possible_params, and FALSE if it is not.

**Examples**

```r
go
possible_params = c("KDPI", "KDRI_Rao", "KDRI_median")
service.is.param_possible("KDZO", possible_params) # return FALSE
service.is.param_possible("KDPI", possible_params) # return TRUE
```

---

**service.is_numeric**

Check whether a vector is numeric.

**Description**

Check whether a vector is numeric.

**Usage**

```r
service.is_numeric(x)
```

**Arguments**

- **x**: the vector to be checked.

**Details**

Check whether a vector is numeric.

Programming: Boris Bikbov <boris@bikbov.ru>.

**Value**

logic whether vector x is numeric or not.
**service.output_message**

*Produce message for warning or cat*

**Description**

Produce message for warning or cat

**Usage**

```
service.output_message(x, custom_phrase, warning_type)
```

**Arguments**

- **x**  
  Numeric. The value to be checked (usually a counter of some variable).
- **custom_phrase**  
  Character string. Custom message to be inserted in the middle of standard message.
- **warning_type**  
  Character string. The type of message: warning (with substitution to NA) or cat (with leave as is).

**Details**

Produce message that is used by warning or cat in the ktx.kdpiOPTN function. Service function that will not be exported to user, and used only in the ktx.kdpiOPTN function.

Programming: Boris Bikbov <boris@bikbov.ru>.

**Value**

Character string. Returns a phrase.

---

**service.singular_or_plural**

*Form output message in singular or plural.*

**Description**

Form output message in singular or plural.

**Usage**

```
service.singular_or_plural(x, singular, plural)
```
service.strict_to_numeric_threshold_greater

Arguments

- **x**: Numeric. The value to be checked (usually a counter of some variable).
- **singular**: Character string. The value to be returned in case of singular form (usually a string, but could be any type).
- **plural**: Character string. The value to be returned in case of plural form (usually a string, but could be any type).

Details

Provide different output for constructing messages in singular or plural.

Programming: Boris Bikbov <boris@bikbov.ru>.

Value

Character string. Returns a value for constructing messages in singular or plural form.

Examples

```
service.singular_or_plural(1, "This value was", "These values were")  # "This value was"
service.singular_or_plural(99, "This value was", "These values were")  # "These values were"
```

---

service.strict_to_numeric_threshold_greater

*Select only numeric values lower than defined threshold*

Description

Select only numeric values lower than defined threshold

Usage

```
service.strict_to_numeric_threshold_greater(x, threshold)
```

Arguments

- **x**: the vector to be checked.
- **threshold**: numeric the threshold to compare with.

Details

Select only numeric values lower than defined threshold, and substitute other values with NA.

Programming: Boris Bikbov <boris@bikbov.ru>.

Value

numeric returns only numeric values lower than threshold.
service.strict_to_numeric_threshold_lower

Examples

myvals <- c(1, 8, -5, "oggi", NA)
# return to myvals2 only numeric values lower than threshold (3 in this case)
# susbstitute non-numeric or negative values with NA
myvals2 <- service.strict_to_numeric_threshold_greater(myvals, 3)
myvals2 # 1, NA, -5, NA, NA

service.strict_to_numeric_threshold_lower

Select only numeric values greater than defined threshold.

Description

Select only numeric values greater than defined threshold.

Usage

service.strict_to_numeric_threshold_lower(x, threshold)

Arguments

x the vector to be checked.
threshold numeric the threshold to compare with.

Details

Select only numeric values greater than defined threshold, and substitute other values with NA.
Programming: Boris Bikbov <boris@bikbov.ru>.

Value

numeric returns only numeric values greater than threshold.

Examples

myvals <- c(1, 8, -5, "oggi", NA)
# return to myvals2 only numeric values greater than defined threshold (0 in this case)
# and susbstitute non-numeric or negative values with NA
myvals2 <- service.strict_to_numeric_threshold_lower(myvals, 0)
myvals2 # 1, 8, NA, NA, NA
Index

+Topic datasets
  ktx, 6

  egfr.ckdepi, 2
  egfr.mdrd4, 3
  egfr.schwartz, 5

  kidney.epi, 6
  ktx, 6
  ktx.kdpi.optn, 7
  ktx.kdpi.optn.show.years, 10

  service.check_obligatory_params, 10
  service.check_param_arguments, 12
  service.check_param_number, 13
  service.check_params_numeric, 11
  service.check_plausibility.age, 13
  service.check_plausibility.creatinine, 14
  service.convert_creatinine, 14
  service.count_greater_threshold, 15
  service.count_lowerequal_threshold, 16
  service.is.param_possible, 16
  service.is_numeric, 17
  service.output_message, 18
  service.singular_or_plural, 18
  service.strict_to_numeric_threshold_greater, 19
  service.strict_to_numeric_threshold_lower, 20