Package 'ArrowDQAToolkit'

November 28, 2023

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description Assess intrinsic quality of an arrow Table.

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Depends stats, tidyverse, rlang, readxl, reticulate, data.table, arrow

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

acc_uni_outliers *Assess Accuracy - Univaritate Outliers*

Description

check for univariate outliers definition of outliers: based on method introduced by Tukey in 1977

- less than 1.5*IQR away from 1st quantile
- greater than 1.5*IQR away from the 3rd quantile

Usage

```
acc_uni_outliers(data, metadata, plot_result = FALSE)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

list of 2 items: result - result summary data.frame outliers - return outliers values for each variable

arrow_summary *Utility function - Custom summary function for arrow object*

Description

Utility function - Custom summary function for arrow object

Usage

```
arrow_summary(data, var, ...)
```

Arguments

data	• arrow data table to analyze, can be passed via pipe
var	• variables to generate summary
...	

com_crude_missing *Assess Completeness - Crude Missing*

Description

Assess Completeness - Crude Missing

Usage

```
com_crude_missing(
  data,
  item_metadata,
  cross_item_metadata,
  plot_result = FALSE
)
```

Arguments

data	• arrow data table
cross_item_metadata	• cross level metadata, expected data table object generated by prep_metadata function
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

2 data.frame for univariate and multivariate missingness result

com_qualified_missing *Assess Completeness - Qualified missing*

Description

Assess Completeness - Qualified missing

Usage

```
com_qualified_missing(data, metadata, plot_result = FALSE)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

result summary data.frame

con_contradiction *Assess Consistency - Value Contradiction Summarize how much data has contradicting values, and return the rows with contradicting values*

Description

Assess Consistency - Value Contradiction Summarize how much data has contradicting values, and return the rows with contradicting values

Usage

```
con_contradiction(data, metadata, plot_result = FALSE)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

list containing 2 items:

- result summary (data.frame)
- contradicted data (tibble)

con_label *Assess Consistency - Valid labels for categorical labels*

Description

Assess Consistency - Valid labels for categorical labels

Usage

```
con_label(data, metadata, path = NULL, plot_result = FALSE)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function
path	• path to item level metadata

Value

result summary for consistency assessment

con_range *Assess Consistency - Range of values*

Description

Assess Consistency - Range of values

Usage

```
con_range(data, metadata, plot_result = FALSE)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

result summary data.frame

intrinsic_eval *Perform all Intrinsic quality assessment*

Description

Perform all Intrinsic quality assessment

Usage

```
intrinsic_eval(metadata_path, data)
```

Arguments

metadata_path	• path to the data folder
data	• arrow data table

int_datatype *Assess Integrity - Datatype \n*

Description

Check whether all variables follow the pre-defined datatype, and cast the given data to the defined datatype

Usage

```
int_datatype(data, metadata, date_format = NULL)
```

Arguments

data	• arrow data table
metadata	• item level metadata, expected data table object generated by prep_metadata function

Value

data after casting to specified datatypes

int_duplicates *Assess Integrity - duplication*

Description

Check whether there are duplicate rows in pre-defined columns/variables

Usage

```
int_duplicates(
  data,
  check_all = TRUE,
  remove_dups = FALSE,
  cross_item_metadata = NULL,
  plot_result = FALSE
)
```

Arguments

data	• arrow data table
check_all	• whether to check duplicates over all variables
remove_dups	• whether to remove duplicates
cross_item_metadata	• cross item level metadata, expected data table object generated by prep_metadata function

Value

list of 2 items

- result: result summary data.frame
- duplicates: logical vector indicating whether each row is a duplicate

---

`mapping_filter_condition`
Utility function - Mapping value range operator to the built in function

---

Description

Utility function - Mapping value range operator to the built in function

Usage

```
mapping_filter_condition(
  data,
  var,
  datatype,
  lower_bound,
  higher_bound,
  greater,
  less
)
```

Arguments

`data`	• arrow data table
`var`	• variable for fitlering
`datatype`	• datatype of variable
`lower_bound`	• lowerbound value
`higher_bound`	• higherbound value
`greater`	• "(" or "["
`less`	• ")" or "]"

`prep_crossitem_metadata`
Prepare cross-item level metadata

Description

Read cross-item level metadata

Usage

```
prep_crossitem_metadata(path = NULL)
```

Arguments

path	• path to the metadata folder

Value

a list of 2 data.table:

- multivariates - define multivariate variables
- contradiction - contradiction rules

---

prep_item_metadata *Prepare item level metadata*

Description

Read item level metadata

Usage

```
prep_item_metadata(path = NULL)
```

Arguments

path	• path to the metadata folder

Value

a data.table for item level metadata

---

util_compute_duplicates *Utility to check each row whether they are duplicate or not*

Description

Utility to check each row whether they are duplicate or not

Usage

```
util_compute_duplicates(data, vars)
```

Arguments

data	• arrow data table
vars	• variables to check for duplicates

Value

logical vector indicating whether each row is a duplicate or not

util_compute_outliers *Utility function to compute outliers*

Description

Utility function to compute outliers

Usage

```
util_compute_outliers(data, var)
```

Arguments

data	
var	• variable to calculate outlier, expected to be a numeric variable

Value

arrow table containing outliers

util_export *Utility function - Export data*

Description

Save data that failed the checks as an excel file

Usage

```
util_export(data, outfile_name = "failed_check_data")
```

Arguments

data	• data returned by evaluation functionss
outfile_name	• name of the output file

util_graphing_percentage *Utility function - Graphing utility*

Description

graph result table using ggplot and plotly

Usage

```
util_graphing_percentage(
  data,
  label,
  title = NULL,
  percentage = percentage,
  fill = "#87ceeb"
)
```

Arguments

data	• the result dataframe generated by DQA functions
label	• the column used for labeling (usually varname column)
title	• title for the plot
percentage	• column containing percentage to graph
fill	• fill color for the graph

Value

histogram of the result

util_is_integer *Utility to check whether a vector only contains integer*

Description

Utility to check whether a vector only contains integer

Usage

```
util_is_integer(x, tol = .Machine$double.eps^0.5)
```

Arguments

x	• object to †est
tol	• precision of the detection. Values deviating more than tol from their closest integer value will not be deemed integer.

Value

boolean value of whether vector only contains integer