Package 'logib'

December 20, 2024

Type Package

Title Salary Analysis by the Swiss Federal Office for Gender Equality

Version 0.2.0

Description Implementation of the Swiss Confederation's standard analysis model for salary analyses <https://www.ebg.admin.ch/en/equal-pay-analysis-with-logib> in R. The analysis is run at company-level and the model is intended for medium-sized and large companies. It can technically be used with 50 or more employees (apprentices, trainees/interns and expats are not included in the analysis). Employees with at least 100 employees are required by the Gender Equality Act to conduct an equal pay analysis. This package allows users to run the equal salary analysis in R, providing additional transparency with respect to the methodology and simple automation possibilities.

License GPL (>= 3)

Depends R (>= 3.1)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports lubridate, readxl, stats, utils

Suggests testthat

URL https://github.com/admin-ebg/logib

BugReports https://github.com/admin-ebg/logib/issues

NeedsCompilation no

Author Marc Stöckli [aut, cre],
Jonathan Chassot [aut],
Jeremy Kolly [ctb],
Federal Office for Gender Equality of Switzerland [cph, fnd]

Maintainer Marc Stöckli <marc.stoeckli@ebg.admin.ch>

Repository CRAN

Date/Publication 2024-12-20 09:10:02 UTC

Contents

`all_column_names` *Column names*

Description

List of column names used in the code, from the datalist and exportfiles in all four languages (de, fr, it, en)

Usage

```
all_column_names
```

Format

An object of class `list` of length 3.

`analysis` *Run a Salary Analysis*

Description

Runs a salary analysis according to the Swiss standard analysis model

Usage

```
analysis(
  data,
  reference_month,
  reference_year,
  usual_weekly_hours = NULL,
  female_spec = "F",
  male_spec = "M",
  age_spec = NULL,
  entry_date_spec = NULL
)
```

Arguments

data	a data.frame of employees as produced by read_data
reference_month	an integer representing the reference month, i.e. the month for which we analyze the salaries
reference_year	an integer representing the reference year, i.e. the year for which we analyze the salaries
usual_weekly_hours	an optional numeric representing the usual weekly working hours (missing values in weekly_hours are replaced by usual_weekly_hours; if NULL, the missing values are not replaced)
female_spec	an optional string or numeric representing the way women are encoded in the data
male_spec	an optional string or numeric representing the way men are encoded in the data
age_spec	an optional string to specify the way age is encoded in the data (NULL will try to automatically infer the age format, "age" implies that the age is specified as the age of a person, "birthyear" implies that the age is specified as the year of birth of a person, and "birthdate" implies that the age is specified as the date of birth of a person)
entry_date_spec	an optional string to specify the way entry_date is encoded in the data (NULL will try to automatically infer the format, "years" implies that the entry_date is specified as the number of years for which the person has been in the company, "entry_year" implies that the entry_date is specified as the year of the entry date of the person, "entry_date" implies that the age is specified as the date of entry of the person)

Value

object of type analysis_model with the following elements

params:	The set of original parameters passed to the function
data_original:	The original data passed by the user in the data parameter
data_clean:	The cleaned up data which was used for the analysis
data_errors:	The list of errors which were found upon checking the data
results:	The result of the standard analysis model

Examples

```
results <- analysis(data = datalist_example, reference_month = 1,
   reference_year = 2019, usual_weekly_hours = 40, female_spec = "F",
   male_spec = "M", age_spec = "age")
```

build_custom_mapping *Build column name mappings*

Description

build_custom_mapping creates a vector of column name mappings for the user to read her or his custom dataframe

Usage

```
build_custom_mapping(data, language = "de", prompt_mapping = TRUE)
```

Arguments

data	the custom dataframe for which the user wants to build a custom mapping
language	a character string representing the language in which the columns will be displayed during the mapping prompt ("de" or "fr" or "it" or "en")
prompt_mapping	a boolean indicating whether the function prompts the user for the exact mapping of his dataframe or whether the columns are mapped automatically by order

Details

Builds a mapping from the custom column names of a given data.frame to the variable names used in the standard analysis model. If prompt_mapping is set to TRUE, the function prompts the mapping for each column of the data.frame. If prompt_mapping is set to FALSE, the mapping is built using the order of the columns of the given data.frame.

Value

A named vector of characters, where the name indicates the column name in the original data.frame and the value indicates the column name as used by the standard analysis model.

datalist_example *Example datalist*

Description

Fictional dataset containing the necessary information to run an equal pay analysis.

Usage

```
datalist_example
```

Format

A data frame with 318 rows and 23 variables:

personal_number personal number of the employee, alphanumeric,

age age, in years,

sex sex, 1 = male, 2 = female,

years_of_service years of service, in years,

training training code, 1-8,

professional_function function / job,

level_of_requirements level of requirements code, 1-4,

professional_position professional position / hierarchy code, 1-5,

activity_rate activity rate, in percent,

paid_hours paid hours, in hours,

basic_wage basic wage, in CHF,

allowances allowances, in CHF,

monthly_wage_13 13th monthly wage, in CHF,

special_payments special payments, in CHF,

weekly_hours weekly contractual hours, in hours,

annual_hours annual contractual hours, in hours,

population analysis population code, 1-5,

comments comments for the employee,

supplement1 additional remarks (1 of 5),

supplement2 additional remarks (2 of 5),

supplement3 additional remarks (3 of 5),

supplement4 additional remarks (4 of 5),

supplement5 additional remarks (5 of 5)

`download_datalist` *Download official Excel datalists*

Description

Downloads an empty version of the latest official Excel datalist in the specified language to the given `path`.

Usage

```
download_datalist(file, language = "de")
```

Arguments

file	a character string representing the file path to which the downloaded datalist will be saved.
language	a character string representing the language of the datalist to be download ("de" or "fr" or "it" or "en").

Value

None

---

download_example_datalist *Download official filled-in sample Excel datalists*

Description

Downloads a filled-in version of the latest official Excel datalist in the specified language to the given path.

Usage

```
download_example_datalist(file, language = "de")
```

Arguments

file	a character string representing the file path to which the downloaded datalist will be saved.
language	a character string representing the language of the datalist to be download ("de" or "fr" or "it" or "en").

Value

None

---

read_data *Create the dataframe object used for the standard analysis model*

Description

Reads either a custom dataframe object or an official Excel file (datalist or data export) and transforms it to a dataframe object which can be used for the standard analysis model

Usage

```
read_data(
  data_path = NULL,
  custom_data = NULL,
  prompt_mapping = TRUE,
  language = "de"
)
```

Arguments

data_path	a string indicating the path for an official Excel file, if this parameter is set to NULL, the function reads the dataframe object provided in the parameter custom_data instead
custom_data	a dataframe which was imported by the user beforehand, if this parameter is set to NULL, the function import the data from the path provided in the parameter data_path instead
prompt_mapping	a boolean indicating whether the function prompts the user for the exact mapping of his dataframe or whether the columns are mapped automatically by order. This parameter is only relevant when custom_data is not set to NULL
language	a character string representing the language in which the columns will be displayed during the mapping prompt ("de" or "fr" or "it" or "en"). This parameter is only relevant when custom_data is not set to NULL

Details

Exactly one of data_path or custom_data must be NULL.

Value

a dataframe which can be used to compute the standard analysis model

---

summary.analysis_model

Summary of the Salary Analysis

Description

Summary of an estimated salary analysis object of class analysis_model

Usage

```
## S3 method for class 'analysis_model'
summary(object, ...)
```

Arguments

`object`	estimated salary analysis object of class `analysis_model`
`...`	further arguments passed to or from other methods

Details

`summary.analysis_model` provides a short summary of the wage analysis according to the Standard Analysis Model. The summary describes the number of records used for the analysis, the Kennedy estimate of the wage difference under otherwise equal circumstances and the summary of the linear regression.

Value

Nothing

Examples

```
# Estimate standard analysis model
results <- analysis(data = datalist_example, reference_month = 1,
   reference_year = 2019, usual_weekly_hours = 40, female_spec = "F",
   male_spec = "M", age_spec = "age")

# Show summary of the salary analysis
summary(results)
```

`transform_data` *Transform a data.frame according to the requirements of the model*

Description

Transforms specific columns of a data.frame to match the requirements of the standard analysis model.

Usage

```
transform_data(
  data,
  reference_year,
  usual_weekly_hours,
  female_spec = "F",
  male_spec = "M",
  age_spec = NULL,
  entry_date_spec = NULL
)
```

Arguments

data	a dataframe object as produced by read_data which is to be transformed
reference_year	a number indicating the reference year of the analysis
usual_weekly_hours	an optional numeric representing the usual weekly working hours
female_spec	a string or number indicating the way females are specified in the dataset.
male_spec	a string or number indicating the way males are specified in the dataset
age_spec	a string indicating the age specification, can be one of NULL, "age", "birthyear", or "date_of_birth". If this parameter is set to NULL, the function automatically tries to infers the specification
entry_date_spec	a string indicating the entry_date specification, can be one of NULL, "years", "entry_year", or "entry_date". If this parameter is set to NULL, the function automatically tries to infers the specification

Index