

Package ‘priceR’

March 31, 2019

Type Package

Title Regular Expressions for Prices and Currencies

Version 0.1.0

Author Steve Condylis [aut, cre] (<<https://orcid.org/0000-0003-0599-844X>>)

Maintainer Steve Condylis <steve.condylis@gmail.com>

Description Functions to aid in the analysis of price and currency data by expediting data preprocessing. This includes extraction of relevant data (e.g. from text fields), conversion into numeric class, cleaning, and standardisation as appropriate.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports dplyr, gsubfn, stringr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-31 13:20:03 UTC

R topics documented:

extract_salary	2
priceR	4

Index	5
--------------	----------

extract_salary	<i>Extract numeric salary from text data</i>
----------------	--

Description

Extract numeric salary from text data. 'extract_salary' automatically converts weekly and hourly rates to amounts per annum.

Usage

```
extract_salary(salary_text, exclude_below, exclude_above, salary_range_handling,  
include_periodicity, hours_per_workday, days_per_workweek, working_weeks_per_year)
```

Arguments

salary_text	A character string, or vector of character strings.
exclude_below	A lower bound. Anything lower than this number will be replaced with NA.
exclude_above	An upper bound. Anything above this number will be replaced with NA.
salary_range_handling	A method of handling salary ranges. Defaults to returning an average of the range; can also be set to "max" or "min".
include_periodicity	Set to TRUE to return an additional column stating the detected periodicity in the character string.
hours_per_workday	Set assumed number of hours in the workday. Only affects annualisation of rates identified as Daily. Default is 8 hours.
days_per_workweek	Set assumed number of days per workweek. Only affects annualisation of rates identified as Daily. Default is 5 days.
working_weeks_per_year	Set assumed number of working weeks in the year. Only affects annualisation of rates identified as Daily or Weekly. Default is 48 weeks.

Examples

```
# Provide a salary string and 'extract_salary' and will extract the salary and return it  
extract_salary("$160,000 per annum")  
# 160000  
  
# If a range is present, the average will be taken by default  
extract_salary("$160,000 - $180000.00 per annum")  
# 170000
```

```

# Take the 'min' or 'max' of a salary range by setting salary_range_handling parameter accordingly
extract_salary("$160,000 - $180000.00 per annum", salary_range_handling = "min")
# 160000

# Extract salaries from character string(s)
annual_salaries <- c("$160,000 - $180000.00 per annum",
                    "$160000.00 - $180000.00 per annum",
                    "$145000 - $155000.00 per annum",
                    "$70000.00 - $90000 per annum",
                    "$70000.00 - $90000.00 per annum plus 15.4% super",
                    "$80000.00 per annum plus 15.4% super",
                    "60,000 - 80,000",
                    "$78,686 to $89,463 pa, plus 15.4% superannuation",
                    "80k - 100k")

extract_salary(annual_salaries)
# 170000 170000 150000 80000 53338 40008 70000 56055 90000

# Automatically detect, extract, and annualise daily rates
daily_rates <- c("$200 daily", "$400 - $600 per day", "Day rate negotiable dependent on experience")
extract_salary(daily_rates)
# 48000 120000 NA

# Automatically detect, extract, and annualise hourly rates
hourly_rates <- c("$80 - $100+ per hour", "APS6/EL1 hourly rate contract")
extract_salary(hourly_rates)
# 172800 6720
# Note 6720 is undesirable. Setting the lower and upper bounds sensibly avoids this

salaries <- c(annual_salaries, daily_rates, hourly_rates)

# Setting lower and upper bounds provides a catch-all to remove unrealistic results
# Out of bounds values will be converted to NA
extract_salary(salaries, exclude_below = 20000, exclude_above = 600000)
# 170000 170000 150000 80000 53338 40008 70000 56055 90000 48000 120000 NA 172800 NA

# extract_salary automatically annualises hourly and daily rates
# It does so by making assumptions about the number of working weeks in a year,
# days per workweek, and hours per workday
# And the assumed number of hours per workday can be changed from the default (8)
# The assumed number of workdays per workweek can be changed from the default (5)
# The assumed number of working weeks in year can be changed from the default (48)
# E.g.
extract_salary(salaries, hours_per_workday = 7, days_per_workweek = 4, working_weeks_per_year = 46)
# 170000 170000 150000 80000 53338 40008 70000 56055 90000 36800 92000 NA 115920 4508

```

```
# To see which salaries were detected as hourly or weekly, set include_periodicity to TRUE
extract_salary(salaries, include_periodicity = TRUE)
```

```
# salary periodicity
# 1 170000 Annual
# 2 170000 Annual
# 3 150000 Annual
# 4 80000 Annual
# 5 53338 Annual
# 6 40008 Annual
# 7 70000 Annual
# 8 56055 Annual
# 9 90000 Annual
# 10 48000 Daily
# 11 120000 Daily
# 12 NA Daily
# 13 172800 Hourly
# 14 6720 Hourly
```

priceR

priceR: Regular Expressions for Prices and Currencies

Description

Functions to aid in the analysis of price and currency data by expediting data preprocessing. This includes extraction of relevant data (e.g. from text fields), conversion into numeric class, cleaning, and standardisation as appropriate.

Details

It has the goal of providing a quick and practical way of extracting numeric price and currency data from text fields, thus allowing faster and easier analysis of ostensibly numeric data.

Author(s)

Steve Condylios <steve.condylios@gmail.com>

Index

`extract_salary`, [2](#)

`priceR`, [4](#)

`priceR-package (priceR)`, [4](#)