Package 'usincometaxes'

October 30, 2022

Title Calculate Federal and State Income Taxes in the United States

Version 0.5.4

Description Calculates federal and state income taxes in the United States. It acts as a wrapper to the NBER's TAXSIM 35 (<http://taxsim.nber.org/taxsim35/>) tax simulator. TAXSIM 35 conducts the calculations, while 'usincometaxes' prepares the data for TAXSIM 35, sends the data to TAXSIM 35's server or communicates with the Web Assembly file, retrieves the data, and places it into a data frame. All without the user worrying about this process.

License MIT + file LICENSE

URL https://github.com/shanejorr/usincometaxes, https://shaneorr.io/r/usincometaxes

BugReports https://github.com/shanejorr/usincometaxes/issues

Depends R (>= 3.1)

Imports datasets, vroom, httr, tibble, utils, V8, tidyselect

Suggests dplyr, ggplot2, knitr, rmarkdown, scales, testthat (>= 3.0.0), tidyr, tools

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

NeedsCompilation no

Author Shane Orr [aut, cre, cph], Thomas Wells [ctb]

Maintainer Shane Orr <shane.j.orr@gmail.com>

Repository CRAN

Date/Publication 2022-10-30 15:40:02 UTC

R topics documented:

`create_dataset_for_taxsim` *Convert a data frame to the TAXSIM 35 output.*

Description

This function takes a data set that is in the format required for `taxsim_calculate_taxes`, checks it to make sure it is in the proper format for TAXSIM 35, and then cleans so it can be sent to TAXSIM 35. This function is useful for troubleshooting. It is not needed to calculate taxes. The function is useful if you continue receiving unreasonable errors from `taxsim_calculate_taxes`. In such as case, you can run this function on your data set. You should then save the resulting data frame as a csv file. Then, upload the file to TAXSIM 35. If there are no errors with TAXSIM 35 then the issue lies in `taxsim_calculate_taxes`.

Usage

```
create_dataset_for_taxsim(.data)
```

Arguments

`.data`	Data frame containing the information that will be used to calculate taxes. This data set will be sent to TAXSIM. Data frame must have specified column names and data types.

Details

`create_dataset_for_taxsim` takes the same columns as column names as TAXSIM 35.

Value

A data frame that that can be manually uploaded to TAXSIM 35.

Examples

```
family_income <- data.frame(
    taxsimid = c(1, 2),
    state = c('North Carolina', 'NY'),
    year = c(2015, 2015),
    mstat = c('single', 'married, jointly'),
```

```
    pwages = c(10000, 100000),
    page = c(26, 36)
)

family_taxes <- create_dataset_for_taxsim(family_income)

# You can then write out the data frame as a csv file for uploading to TAXSIM 35
```

`get_state_soi` *Get state SOI from state name.*

Description

Converts state names or state abbreviations to numeric SOI codes, which are required for TAXSIM.

Usage

```
get_state_soi(state_column)
```

Arguments

`state_column`	Vectors containing the states to calculate taxes for. Generally, this is the state column from the data set that will be sent to TAXSIM.

Value

Named integer vector with each number between 1 and 51 representing the state's SOI. Names are the state's two letter abbreviation.

`taxpayer_finances` *Financial and household characteristics of 1,000 taxpayer units.*

Description

A data set containing financial and household characteristics of 1,000 taxpayer units. The data set was randomly generated and does not reflect real data. It is formatted and ready for use in the `usincometaxes` package.

Usage

```
taxpayer_finances
```

Format

A data frame with 1,000 rows and 16 variables. Variable definitions can be found in the following article: https://www.shaneorr.io/r/usincometaxes/articles/taxsim-input.html

Source

Created through random data generation.

`taxsim_calculate_taxes` *Calculate state and federal taxes using TASXSIM 35.*

Description

This function calculates state and federal income taxes using the TAXSIM 35 tax simulator. See http://taxsim.nber.org/taxsim35/ for more information on TAXSIM 35.

Usage

```
taxsim_calculate_taxes(
  .data,
  marginal_tax_rates = "Wages",
  return_all_information = FALSE,
  interface = "wasm"
)
```

Arguments

`.data`	Data frame containing the information that will be used to calculate taxes. This data set will be sent to TAXSIM. Data frame must have specified column names and data types.
`marginal_tax_rates`	Variable to use when calculating marginal tax rates. One of 'Wages', 'Long Term Capital Gains', 'Primary Wage Earner', or 'Secondary Wage Earner'. Default is 'Wages'.
`return_all_information`	Boolean (TRUE or FALSE). Whether to return all information from TAXSIM (TRUE), or only key information (FALSE). Returning all information returns 42 columns of output, while only returning key information returns 9 columns. It is faster to download results with only key information.
`interface`	String indicating which NBER TAXSIM interface to use. Should be one of: "wasm", "ssh", or "http". • "wasm" uses a compiled WebAssembly version of the TAXSIM app. Details about generating the wasm file can be found here: https://github.com/tmm1/taxsim.js • "ssh" uses SSH to connect to taxsimssh.nber.org. Your system must already have SSH installed. • "http" uses CURL to connect to https://taxsim.nber.org/uptest/webfile.cgi. 'http" does not work for dataset with more than 1,000 rows.

Value

The output data set contains all the information returned by TAXSIM 35, using the same column names. Descriptions of these columns can be found at the bottom of the page containing TAXSIM 35's documentation.

Formatting your data

In the input data set, `.data`, each column is a tax characteristic (year, filing status, income, etc.) and each row is a tax filing unit.

Columns should take the same names, and fulfill the same requirements, as those needed for TAXSIM 35. Potential columns, with there names and descriptions, can be found at: http://taxsim.nber.org/taxsim35/.

The following columns are required: `taxsimid`, `year`, `mstat`, and `state`.

There are two points where `taxsim_calculate_taxes` departs from TAXSIM 35.

1. For filing status, `mstat`, users can either enter the number allowed by TAXSIM 35 or one of the following descriptions:

- "single"
- "married, jointly"
- "married, separately"
- "dependent child"
- "head of household"

1. For `state`, users can either enter the SOI code, as required by TAXSIM 35, the two-letter state abbreviation, or the full name of the state.

It is OK if the input data set, `.data`, contains columns in addition to the ones that are used by TAXSIM 35.

Giving credit where it is due

The NBER's TAXSIM 35 tax simulator does all tax calculations. This package simply lets users interact with the tax simulator through R. Therefore, users should cite the TASXSIM 35 tax simulator when they use this package in their work:

Feenberg, Daniel Richard, and Elizabeth Coutts, An Introduction to the TAXSIM Model, Journal of Policy Analysis and Management vol 12 no 1, Winter 1993, pages 189-194.

Examples

```
## Not run:
family_income <- data.frame(
    taxsimid = c(1, 2),
    state = c('North Carolina', 'NY'),
    year = c(2015, 2015),
    mstat = c('single', 'married, jointly'),
    pwages = c(10000, 100000),
    page = c(26, 36)
```

```
)

family_taxes <- taxsim_calculate_taxes(family_income)

merge(family_income, family_taxes, by = 'taxsimid')

## End(Not run)
```

Index