# treasurycurves

*Release v1.0.2*

**Danny Fryer**

**Sep 22, 2022**

# CONTENTS

CHAPTER

ONE

# README

# TREASURY CURVES

Access and chart US Treasury Curve data

Full documentation can be found at treasury-curves.readthedocs.io

CHAPTER

TWO

# SELECTING AND PLOTTING DATA

Using the Developer API found at treasury.gov, one can find yield curves dating back to the 90's.

Access data for a particular date using `curves("2022-08-08")`. curves will find the nearest valid date (weekday) and returns a pandas dataframes of all curves available on that date or nearest to it. calling `curves()` will use the current date. Optionally specify `curves(allow_missing=True)` to keep years with missing data.

Alternatively, you can retrieve all data by calling `download()`.

Once you have the data, it is possible to plot out the yield curves with `plot(curves_data, start="2022-01-01", end="2022-05-03", num_years=1)`. Specify the number of years, or pick a start and end date to sample from, as the chart would otherwise become inundated with yield curve data.

Save your data in a spreadsheet using `export(curves_data, file_extension="csv")` and specify "csv" or "xlsx" to determine the file type.

CHAPTER

THREE

# USING TREASURYCURVES FROM THE COMMAND LINE

Install through pip:

`pip install treasury-curves`

Once installing, you can run the entrypoint:

`treasury --help`

Use the module directly:

`import treasury`

Working inside the repository

`python treasury.py --help`

```
usage: treasury.py [-h] [-a] [-s START] [-e END] [-d DATE] [-y YEARS] [-p] [-o OUTPUT]

treasurycurves - query and analyze US Treasury yield data

optional arguments:
  -h, --help            show this help message and exit
  -a, --allowna         Allow NaN values
  -s START, --start START
                        Year to start analysis
  -e END, --end END     Year to end analysis. If equal to start, analyze curves for the␣
↪year
  -d DATE, --date DATE  Date in YYYY-MM-DD to analyze
  -y YEARS, --years YEARS
                        Num years before end to analyze
  -p, --plot            Plot yield curves
  -o OUTPUT, --output OUTPUT
                        File extension to save data (csv or xlsx), leave empty to avoid␣
↪saving file
```

## 3.1 treasury module

treasury - carry out various analyses on US Treasury Data

`treasury.curves`(*date=None*, *allow_missing=False*)

get treasury curves for today or a specific date

**Parameters**

- **date** (*str*) – a datetime str in the format YYYY-MM-DD
- **allow_missing** (*bool*) – boolean flag that allows NaN values when True

`treasury.export`(*curves_data*, *file_extension='csv'*)

export curves data analysis to a desired output format

**Parameters**

- **curves_data** (*pandas.DataFrame*) – dataframe containing treasury curve data
- **file_extension** (*str*) – the file type to export. can be "csv" or "xlsx"

`treasury.plot`(*raw_curve_data*, *num_years=10*, *start_year=None*, *end_year=None*)

plot treasury curves over the past num_years. alternatively use start_year, end_year as a range. if start_year equals end_year, plot yearly data over the 12 months

**Parameters**

- **raw_curve_data** (*pandas.DataFrame*) – treasury curve data with index Year
- **num_years** (*int*) – number of years to plot, default is 10
- **start_year** (*int*) – first year to begin plotting data
- **end_year** (*int*) – last year to consider when plotting

`treasury.yearly_curves`(*year*, *allow_missing=False*)

get the yearly curve over months for different treasury durations

**Parameters**

- **year** (*int*) – the year to analyze
- **allow_missing** (*bool*) – boolean flag that allows NaN values when True

# PYTHON MODULE INDEX

## t

# INDEX