Tutorial

The function fred_api() allows users to conveniently download economic time series data from the FRED (Federal Reserve Economic Data) online database maintained by the Research Department at the Federal Reserve Bank of St. Louis. The data are accessed by connecting directly to the FRED data server (https://fred.stlouisfed.org).

The selected series are returned into a timetable. When multiple series are requested by the user, they are all merged together and returned into the same timetable. Merging returns the union (rather than the intersection) of the dates of the selected series. If a series is shorter than the rest of the requested series, the absent observations are set into missing (NaN). Similarly, series of different frequencies can also be requested in the same function call, and missing observations (that are due to the frequency mismatch) are treated in the same manner. The user can request the desired range of dates by setting the Name-Value pairs 'StartDate' and 'EndDate'. If no range is specified, the function retrieves all the observations available in FRED for the selected series.

```
%SYNTAX:
%[TT, Meta] = fred_api(fred_code, 'StartDate',startdate, 'EndDate',enddate);

%INPUTS:
%- 'fred_code': The FRED mnemonic(s) of the requested series.
%- 'StartDate': Specifies the series starting date. If the input is
%               omitted, the function returns the earliest starting date
%               available in FRED. The input should be specified as a
%               datetime e.g. datetime(2021,3,31) for 31st March 2021.
%- 'EndDate':   Specifies the series ending date. If the input is omitted,
%               the function returns the latest ending date available in
%               FRED. The input should be specified as a
%               datetime

%OUTPUTS:
%- 'TT':   A timetable of size T-by-N containing the timeseries, where N is
%          the number of the specified FRED series in 'fred_seriescode'.
%          The dates in the resulted timetable are in DD/MM/YYYY format.
%- 'Meta': A N-by-10 sized table containing metadata information for the
%          N requested variables. Specifically, the following information
%          is retruned in the table:
%           DateRange, Frequency, LastUpdated, Notes, Release,
%           SeasonalAdjustment, SeriesID, Source, Title, Units.
```

Examples

```
%-----------------------------%
% (A) Request a single series %
%-----------------------------%

%- Retrieve all the available dates
fred_code = 'CPIAUCSL';  %CPI Inflation (monthly)
[TT, Meta] = fred_api(fred_code);
[TT, Meta] = fred_api(fred_code, 'StartDate',NaT,'EndDate',NaT); %Equivalent to above
[TT, Meta] = fred_api(fred_code, 'StartDate',[],'EndDate',[]);   %Equivalent to above
```

```
[TT, Meta] = fred_api(fred_code, 'StartDate',[],'EndDate',datetime('now'));  %Equivalent to abc

%Print the last 5 rows
TT(end-4:end, :)
```

ans = 5×1 timetable

	Time	CPIAUCSL
1	01/03/2...	287.7080
2	01/04/2...	288.6630
3	01/05/2...	291.4740
4	01/06/2...	295.3280
5	01/07/2...	295.2710

```
%Print selected metadata
Meta(:, ["DateRange","Frequency","LastUpdated","Units","SeasonalAdjustment"])
```

ans = 1×5 table

...

	DateRange	Frequency	LastUpdated	Units
1	"1947-01-01 to 2022-07-01"	"Monthly"	"2022-08-10 7:38 ...	"Index 1982-1984=100"

```
%- Retrieve selected range (from startdate to enddate)
fred_code = 'CPIAUCSL';
startdate = datetime(2020,1,1);
enddate = datetime(2020,12,1);
[TT, Meta] = fred_api(fred_code, 'StartDate',startdate, 'EndDate',enddate);

%Print the entire table
TT(:,:)
```

ans = 12×1 timetable

	Time	CPIAUCSL
1	01/01/2...	258.6820
2	01/02/2...	259.0070
3	01/03/2...	258.1650
4	01/04/2...	256.0940
5	01/05/2...	255.9440
6	01/06/2...	257.2170
7	01/07/2...	258.5430
8	01/08/2...	259.5800
9	01/09/2...	260.1900
10	01/10/2...	260.3520
11	01/11/2...	260.7210
12	01/12/2...	261.5640

```
%Return the series as a numerical vector (instead of timetable)
TT{:,:}
```

```
ans = 12×1
  258.6820
  259.0070
  258.1650
  256.0940
  255.9440
  257.2170
  258.5430
  259.5800
  260.1900
  260.3520
      ⋮
```

```
%----------------------------%
% (B) Request multiple series %
%----------------------------%

%- Request SAME-FREQUENCY variables
fred_code = ["INDPRO","IPB50001N","UNRATE"]; %3 monthly series
[TT, Meta] = fred_api(fred_code, 'StartDate',datetime(2000,1,1), 'EndDate',datetime(2020,3,1));

%Print the last 5 rows
TT.Time.Format = 'dd-MMM-yyyy'; %Change the display format of dates
TT(end-4:end, :)
```

ans = 5×3 timetable

	Time	INDPRO	IPB50001N	UNRATE
1	01-Nov-2019	102.0979	101.5782	3.6000
2	01-Dec-2019	101.7632	101.7663	3.6000
3	01-Jan-2020	101.3030	101.2873	3.5000
4	01-Feb-2020	101.7038	102.0064	3.5000
5	01-Mar-2020	97.8746	98.0919	4.4000

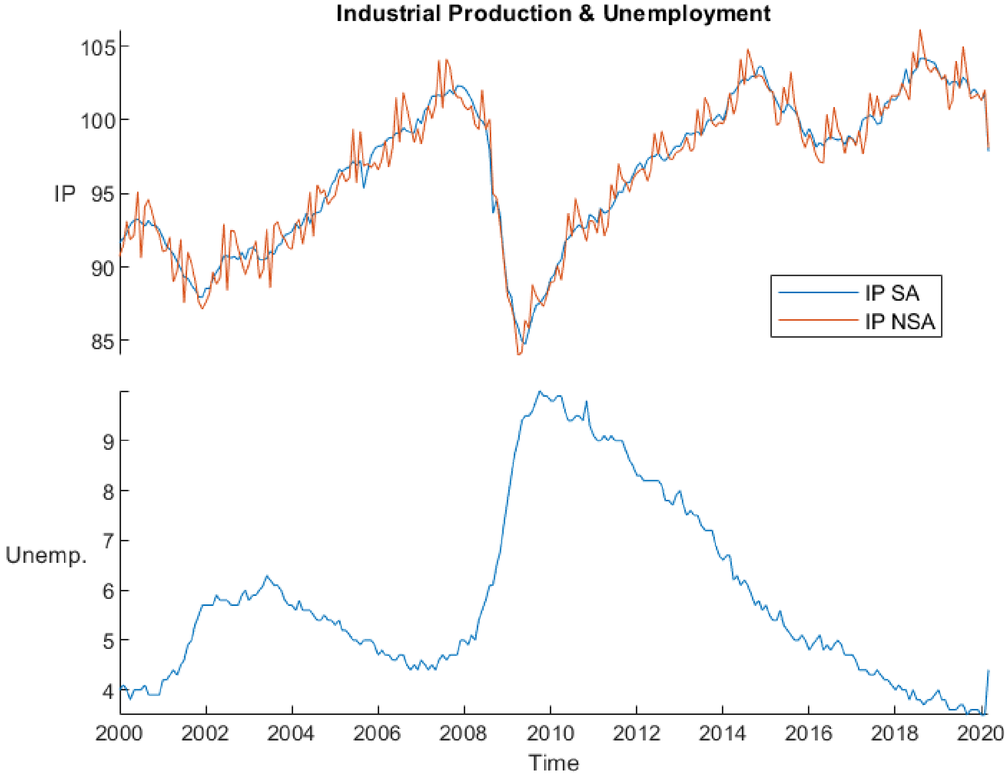
```
%Print selected metadata
Meta(:, ["Frequency","Units","SeasonalAdjustment"])
```

ans = 3×3 table

	Frequency	Units	SeasonalAdjustment
1	"Monthly"	"Index 2017=100"	"Seasonally Adjusted"
2	"Monthly"	"Index 2017=100"	"Not Seasonally Adjusted"
3	"Monthly"	"Percent"	"Seasonally Adjusted"

```
%Plot the series
p = stackedplot(TT,{["INDPRO" "IPB50001N"] "UNRATE"});
p.Title = 'Industrial Production & Unemployment';
p.DisplayLabels = {'IP','Unemp.'};
p.AxesProperties(1).LegendLabels = ["IP SA","IP NSA"];
```

```
p.AxesProperties(1).LegendLocation = 'SouthEast';
```



```
%- Request variables of DIFFERENT FREQUENCIES
fred_code = ["INDPRO","GDPC1"];  %1 monthly & 1 quarterly series
[TT, Meta] = fred_api(fred_code, 'StartDate',datetime(2000,1,1), 'EndDate',datetime(2022,6,1));

%Print the last 8 rows
TT(end-7:end, :)
```

```
ans = 8×2 timetable
```

	Time	INDPRO	GDPC1
1	01/11/2...	101.9610	NaN
2	01/12/2...	101.7587	NaN
3	01/01/2...	102.1460	1.9728e+04
4	01/02/2...	102.8987	NaN
5	01/03/2...	103.6218	NaN
6	01/04/2...	104.3022	1.9699e+04
7	01/05/2...	104.2103	NaN
8	01/06/2...	104.2229	NaN

```
%Print selected metadata
Meta(:, ["Frequency","Units","SeriesID"])
```

```
ans = 2×3 table
```

	Frequency	Units	SeriesID
1	"Monthly"	"Index 2017=100"	"INDPRO"
2	"Quarterly"	"Billions of Chained 2012 Do...	"GDPC1"

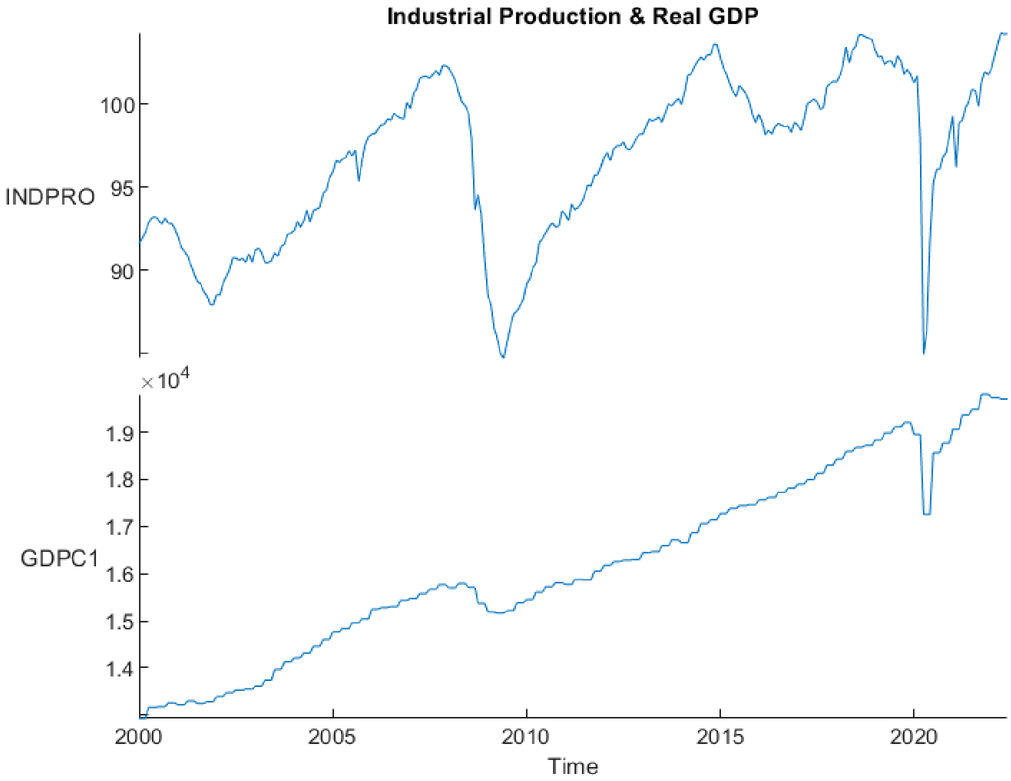
```
%Fill the quarterly series by repeating within-period values
TT(:,"GDPC1") = fillmissing(TT(:,"GDPC1"),'previous');

%Print the last 8 rows
TT(end-7:end, :)
```

```
ans = 8×2 timetable
```

	Time	INDPRO	GDPC1
1	01/11/2...	101.9610	1.9806e+04
2	01/12/2...	101.7587	1.9806e+04
3	01/01/2...	102.1460	1.9728e+04
4	01/02/2...	102.8987	1.9728e+04
5	01/03/2...	103.6218	1.9728e+04
6	01/04/2...	104.3022	1.9699e+04
7	01/05/2...	104.2103	1.9699e+04
8	01/06/2...	104.2229	1.9699e+04

```
%Plot the series in separate panels
p = stackedplot(TT);
p.Title = 'Industrial Production & Real GDP';
```



Author: Haris Karagiannakis