

Getting to Know the SAS® Data Access Engine: SASEFRED

Paul Newsom, University of South Carolina Aiken

ABSTRACT

Manually downloading data from the FRED® website and uploading it to SAS® can be slow. Fortunately, SAS has a tool to make this process faster. With the SASEFRED Data Access Engine, you can retrieve data from FRED directly into a SAS data set. Spend less time accessing and preparing your data and more time exploring and building models.

INTRODUCTION

SAS/ETS® software has a data access engine, SASEFRED, that uses the LIBNAME statement to access time series data in the FRED database. FRED brings data from multiple providers to one location, making it easier to use, search, display, save, share, and store your desired data. The Federal Reserve Bank of Saint Louis FRED database is free, and according to the FRED website, there are 822,000 United States and international time series from 114 sources (Federal Reserve Bank of Saint Louis, 2023). However, FRED does not provide tools to analyze data, and this is where SAS comes in. Using SASEFRED is easy and requires you to complete two tasks.

First, you must create a FRED® [user account](#) (Federal Reserve Bank of Saint Louis, 2023). Second, you must create an [application programming interface](#) (API) key (Federal Reserve Bank of Saint Louis, 2023). You can create as many API keys as you like. Figure 1 shows what it looks like after creating some API keys and the API key used to demonstrate SASEFRED by using SAS On Demand for Academics or SAS 9.4. Importantly, replace the API used in the demonstrations with the API you create. The demonstration API key will be deleted.

API Key	Application Description
a941fbd163371e27fe76727fa1b89e7c	Southeast SAS User Group API Key for DEMONSTRATION
	I want to download the entire daily history of 3-month Treasury bill rates from the secondary market

Figure 1. FRED API Keys

The purpose of this paper is to demonstrate the use of SASEFRED. To do this, it's important to understand the FRED database structure.

FRED API KEY IN A WEB BROWSER

To help you understand the FRED database structure, you can get extensive documentation and examples of using the FRED API in a web browser located on the FRED website (Federal Reserve Bank of Saint Louis, 2023). To begin, here are a few examples to increase your understanding.

First, to get a list of all data providers—FRED calls data providers, data *sources*—enter the following in your web browser:

- https://api.stlouisfed.org/fred/sources?api_key=a941fbd163371e27fe76727fa1b89e7c.

The output displays information about each data source, each with a unique *source id*. There are a lot of sources, and here are two:

- source id=11 (Dow Jones & Company), and
- source id=85 (S&P Dow Jones Indices LLC).

Each data source can have multiple *releases*.

Second, to illustrate how to retrieve all releases for a given source id—in this case, source id=85—enter the following in your web browser:

- https://api.stlouisfed.org/fred/source/releases?source_id=85&api_key=a941fbd163371e27fe76727fa1b89e7c.

Now, the output displays information about each release from source id=85. Source id=85 has three releases:

- release id=189 (Standard & Poors),
- release id=197 (Dow Jones Averages), and
- release id=199 (S&P Case-Shiller Home Price Indices).

Each release can have multiple *series*.

Third, to demonstrate how to retrieve all data series for a given release id—in this case, release id=197—enter the following in your web browser:

- https://api.stlouisfed.org/fred/release/series?release_id=197&api_key=a941fbd163371e27fe76727fa1b89e7c.

Now, the output displays information about each series from release id=197. And release id=197 (Dow Jones Averages) has four series:

- series id=DJCA (Dow Jones Composite Average),
- series id=DJIA (Dow Jones Industrial Average),
- series id= DJTA (Dow Jones Transportation Average), and
- series id=DJUA (Dow Jones Utility Average).

Knowing the series id is important, and each series id is unique.

Finally, to demonstrate how to retrieve observations from a series—in this case, series id=DJIA—enter the following in your web browser:

- https://api.stlouisfed.org/fred/series/observations?series_id=DJIA&api_key=a941fbd163371e27fe76727fa1b89e7c.

Figure 2 shows daily observations for the closing level of the Dow Jones Industrial Average (DJIA). Notice that the 2013-09-02 value is missing. This date is associated with the United States 2013 Labor Day—a federal holiday with closed stock markets.

```

▼<observations realtime_start="2023-08-29" realtime_end="2023-08-29" observation_start="1600-01-01" observat3
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-08-29" value="14840.95"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-08-30" value="14810.31"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-02" value="."/> ←
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-03" value="14833.96"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-04" value="14930.87"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-05" value="14937.48"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-06" value="14922.50"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-09" value="15063.12"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-10" value="15191.06"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-11" value="15326.60"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-12" value="15300.64"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-13" value="15063.12"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-16" value="15494.78"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-17" value="15529.73"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-18" value="15676.94"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-19" value="15063.55"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-20" value="15451.09"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-23" value="15401.38"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-24" value="15334.59"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-25" value="15273.06"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-26" value="15328.30"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-27" value="15258.24"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-09-30" value="15129.67"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-01" value="15191.70"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-02" value="15133.14"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-03" value="14996.48"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-04" value="15072.58"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-07" value="14936.24"/>
<observation realtime_start="2023-08-29" realtime_end="2023-08-29" date="2013-10-08" value="14776.53"/>

```

Figure 2. Observations from the DJIA series

Changing the series id to DJCA, DJTA, or DJUA produces similar reports of daily prices for each series, respectively. Now that you understand the FRED database structure better, let's explore SASEFRED.

SASEFRED: SAS REPORTS

To repeat the same steps in SAS, enter code similar to the following. All LIBNAME statements below produce reports in SAS instead of your web browser. The only changes you need to make are to the *physical-name* in the LIBNAME *libref* SASEFRED "*physical-name*" and to insert your APIKEY:

```

libname fred sasefred "/home/u49890908/FRED"
  url="https://api.stlouisfed.org/fred/sources?"
  APIKEY='a941fbd163371e27fe76727falb89e7c';

libname fred sasefred "/home/u49890908/FRED"
  url="https://api.stlouisfed.org/fred/source/releases?source_id=85"
  APIKEY='a941fbd163371e27fe76727falb89e7c';

libname fred sasefred "/home/u49890908/FRED"
  url="https://api.stlouisfed.org/fred/release/series?release_id=197"
  APIKEY='a941fbd163371e27fe76727falb89e7c';

libname fred sasefred "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727falb89e7c'
  idlist='DJIA';

```

Notice that the last SASEFRED data access engine step does not require the URL option. The main takeaway is that you need to be aware of the metadata for the variable *observation_value* because this is usually the variable of interest. In this case, *observation_value* is the daily close for the DJIA series. Figure 3 shows that *observation_value* is character, not numeric. FRED contains a lot of economic data, and you often want this variable to be *numeric*, not character.

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
2	observation_ORDINAL	Num	8		
5	observation_date	Num	8	IS8601DA10.	IS8601DA10.
4	observation_realtime_end	Num	8	IS8601DA10.	IS8601DA10.
3	observation_realtime_start	Num	8	IS8601DA10.	IS8601DA10.
6	observation_value	Char	8		
1	observations_ORDINAL	Num	8		

Figure 3. Alphabetic List of Variables and Attributes

CATEGORIES & TAGS

Another way to search for FRED data is using categories or tags. FRED is so large that it can be challenging to find a time series, so searching by categories or tags can save time. This is especially useful if you do not know the time series name or are having issues finding a particular series. FRED has eight main categories and 6,048 tags.

To demonstrate, go to FRED and select Category, Tag, or enter the name of a time series. Figure 4 shows the FRED search interface. Suppose you know the name of the Standard and Poor's 500 time series, SP500, but you're having trouble finding a similar time series. Entering SP500 into the search produces the results in Figure 5.

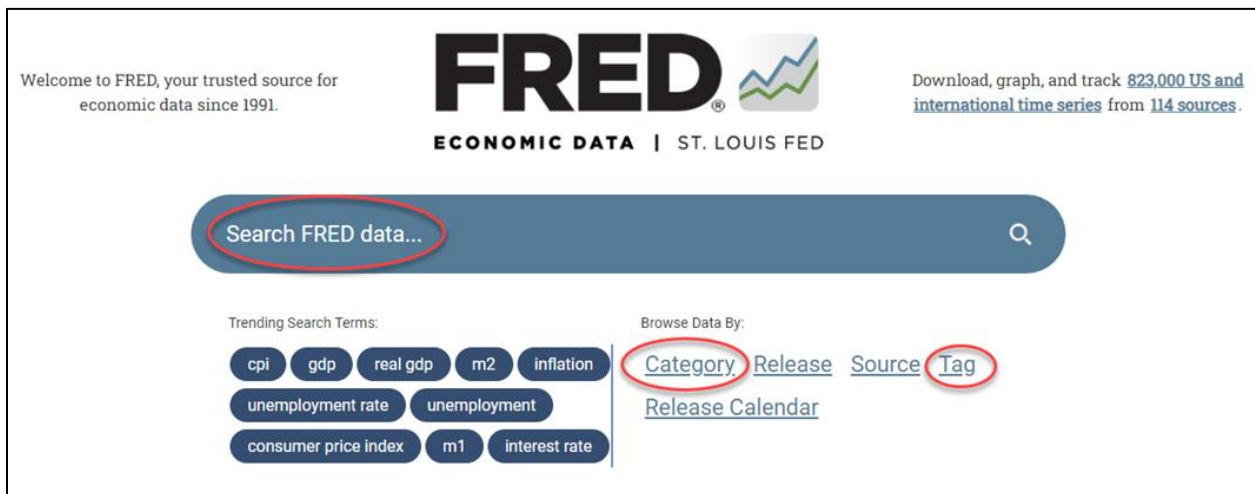


Figure 4. FRED Website

Observe the Related Categories and Tags in Figure 5. You can use these categories and tags to search for the data you want. For example, selecting the Stock Market Indexes category returns 42 stock market indexes in FRED, which includes DJCA, DJIA, DJTA, and DJUA.

Figure 5. Categories and Tags for the Time Series SP500

You can also use an API key and web browser to search for tag names. To demonstrate how to retrieve tags, enter the following in a web browser.

- https://api.stlouisfed.org/fred/tags?api_key=a941fbd163371e27fe76727fa1b89e7c

Figure 6 shows a partial listing of tags. Knowing the *tag name* can help you find the data you want.

```
<tags realtime_start="2023-09-18" realtime_end="2023-09-18" order_by="series_count" sort_order="desc" counts="5982" offset="0" limit="1000">
<tag name="nsa" group_id="seas" notes="Not Seasonally Adjusted" created="2012-02-27 10:18:19-06" popularity="99" series_count="728970"/>
<tag name="usa" group_id="geo" notes="United States of America" created="2012-02-27 10:18:19-06" popularity="100" series_count="658434"/>
<tag name="public domain: citation requested" group_id="cc" notes="" created="2018-12-17 23:33:13-06" popularity="99" series_count="612448"/>
<tag name="annual" group_id="freq" notes="" created="2012-02-27 10:18:19-06" popularity="91" series_count="490360"/>
<tag name="county" group_id="geot" notes="County or County Equivalent" created="2012-02-27 10:18:19-06" popularity="85" series_count="330832"/>
<tag name="nation" group_id="geot" notes="" created="2012-02-27 10:18:19-06" popularity="97" series_count="266004"/>
<tag name="census" group_id="src" notes="Census" created="2012-02-27 10:18:19-06" popularity="85" series_count="237880"/>
<tag name="monthly" group_id="freq" notes="" created="2012-02-27 10:18:19-06" popularity="92" series_count="214884"/>
<tag name="copyrighted: citation required" group_id="cc" notes="" created="2018-12-17 23:33:13-06" popularity="85" series_count="206142"/>
<tag name="bls" group_id="src" notes="Bureau of Labor Statistics" created="2012-02-27 10:18:19-06" popularity="86" series_count="174782"/>
<tag name="persons" group_id="gen" notes="" created="2012-02-27 10:18:19-06" popularity="78" series_count="126486"/>
<tag name="quarterly" group_id="freq" notes="" created="2012-02-27 10:18:19-06" popularity="83" series_count="109446"/>
<tag name="5-year" group_id="gen" notes="" created="2012-02-27 10:18:19-06" popularity="79" series_count="107748"/>
<tag name="employment" group_id="gen" notes="" created="2012-02-27 10:18:19-06" popularity="71" series_count="106204"/>
<tag name="acs" group_id="rls" notes="American Community Survey" created="2014-05-22 12:32:43-05" popularity="79" series_count="101902"/>
<tag name="sa" group_id="seas" notes="Seasonally Adjusted" created="2012-02-27 10:18:19-06" popularity="86" series_count="94116"/>
<tag name="realtor.com" group_id="src" notes="" created="2020-03-24 11:15:04-05" popularity="65" series_count="90324"/>
<tag name="state" group_id="geot" notes="State" created="2012-02-27 10:18:19-06" popularity="74" series_count="89908"/>
<tag name="population" group_id="gen" notes="" created="2012-02-27 10:18:19-06" popularity="80" series_count="88470"/>
<tag name="msa" group_id="geot" notes="Metropolitan Statistical Area" created="2012-02-27 10:18:19-06" popularity="72" series_count="87024"/>
<tag name="price" group_id="gen" notes="" created="2013-07-11 11:24:55-05" popularity="85" series_count="84560"/>
<tag name="saipr" group_id="rls" notes="Small Area Income & Poverty Estimates" created="2015-12-30 13:26:34-06" popularity="60" series_count="80736"/>
<tag name="bea" group_id="src" notes="Bureau of Economic Analysis" created="2012-02-27 10:18:19-06" popularity="78" series_count="78372"/>
<tag name="frb stl" group_id="src" notes="St. Louis Fed" created="2012-02-27 10:18:19-06" popularity="67" series_count="78142"/>
```

Figure 6. List of Tag Names

You can also use categories and tags in SASEFRED. First, to demonstrate searching FRED using categories, run the following LIBNAME step. The LIBNAME step returns the category_id, category_name, and category_parent_id for the SP500 series. Figure 7 shows the category_id=32255, category_name=Stock Market Indexes, and category_parent_id=46 for the SP500 series id.

```
libname fred sasefred "/home/u49890908/FRED"
URL="https://api.stlouisfed.org/fred/series/categories?series_id=SP500"
APIKEY='a941fbd163371e27fe76727fa1b89e7c';
```

Obs	categories_ORDINAL	category_ORDINAL	category_id	category_name	category_parent_id
1	1	1	32255	Stock Market Indexes	46

Figure 7. Category Name and Category Parent ID for Category ID=32255

Next, use the category_id=32255 to search FRED for time series in the same category. To do this, run the following LIBNAME step.

```
libname fred sasefred "/home/u49890908/FRED"
  URL="https://api.stlouisfed.org/fred/category/series?category_id=32255"
  APIKEY='a941fbd163371e27fe76727falb89e7c';
```

Figure 8 shows 46 stock market index time series where category_id=32255. Additional results, not shown, reveal the series id for all of the 46 time series. One of these stock market index time series might be what you want. The main takeaway is that knowing the category of a time series can help you find similar time series.

Obs	series_ORDINAL	series_realtime_start	series_realtime_end	series_order_by	series_sort_order	series_count	series_offset	series_limit
1	1	2023-09-19	2023-09-19	series_id	asc	46	0	1000

Figure 8. 46 Time Series where Category_ID=32255

SASEFRED can also use tags to search FRED. Suppose you want to search for all the time series with two tags: *stock market* and *indexes*. To do this, run the following LIBNAME step. Note that the semicolon after the tag *stock+market* separates the stock+market tag from the *indexes* tag, it isn't a SAS semicolon to end the statement.

```
libname fred sasefred "/home/u49890908/FRED"
  URL="https://api.stlouisfed.org/fred/tags/series?tag_names=stock+market;
  indexes" APIKEY='a941fbd163371e27fe76727falb89e7c';
```

Figure 9 shows there are 57 time series with the tags *stock market* and *indexes*. Additional results, not show, reveal the series id for all of the 57 time series.

Obs	series_ORDINAL	series_realtime_start	series_realtime_end	series_order_by	series_sort_order	series_count	series_offset	series_limit
1	1	2023-09-19	2023-09-19	series_id	asc	57	0	1000

Figure 9. 57 Time Series with two tags: stock market and indexes

Next, to demonstrate what happens when you continue adding tags, run the following LIBNAME step.

```
libname fred sasefred "/home/u49890908/FRED"
  URL="https://api.stlouisfed.org/fred/tags/series?tag_names=stock+market;
  indexes;financial" APIKEY='a941fbd163371e27fe76727falb89e7c';
```

The following section demonstrates how to use SASEFRED to create SAS datasets.

SASEFRED: SAS DATA SETS

SASEFRED can create SAS datasets from FRED time series so that you can perform your analysis and produce reports. To demonstrate what happens when you request data from multiple time series, you can access the daily price levels of the four Dow Jones series: DJIA, DJCA, DJUA, and DJTA:

```
libname FRED SASEFRED "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727falb89e7c'
  idlist='DJIA,DJCA,DJUA,DJTA'
  /*REPLACE is the default, can change to REUSE */
  automap=REPLACE
  /*Specify where XMLmap is stored. Default is XMLMAP=Fred.map */
  XMLMAP="/home/u49890908/FRED/dowl.map"
  /*Specify the name of both the XML file (downloaded) and the */
  /*SAS data set created when XML data are read into SAS. */
  outxml=dowl;
```

```
data dow1;
  set fred.dow1 (drop=realtime_start realtime_end);
run;
```

Figure 10 shows the work.dow1 data set. Notice that the variables DJIA, DJCA, DJUA, and DJTA are character, not numeric and that SASEFRED returns annual, not daily, price levels. Until now, the frequency of the query results for all Dow Jones series has been daily, and the *native* frequency for the Dow Jones series is daily. This means that these time series are reported daily. Other time series in FRED will be reported weekly, monthly, quarterly, annually, or some other frequency.

	date	DJIA	DJCA	DJUA	DJTA
1	2013-01-01
2	2014-01-01	16777.69	5933.69	550.07	8103.23
3	2015-01-01	17587.03	6207.11	583.17	8383.12
4	2016-01-01	17927.11	6289.60	657.01	7882.16
5	2017-01-01	21750.20	7475.10	716.87	9465.50
6	2018-01-01	25053.95	8360.20	707.38	10665.77
7	2019-01-01	26379.59	8738.40	807.54	10412.91
8	2020-01-01	26890.67	8848.42	832.43	10254.66
9	2021-01-01	34055.29	11325.30	897.46	14794.99
10	2022-01-01	32897.35	11118.60	973.65	14310.01
11	2023-01-01

Figure 10. WORK.DOW1 SAS data set.

What happened to the daily data for the four Dow Jones series? By default, SASEFRED resorts to annual frequency when multiple time series are specified in the IDLIST=option. To override the annual default frequency when multiple time series are specified in the IDLIST=, you need to add the FREQ=option. To get SASEFRED to return daily, not annual, data, and to get numeric, not character data, modify the code:

```
libname FRED SASEFRED "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727fa1b89e7c'
  idlist='DJIA,DJCA,DJUA,DJTA'
  automap=REPLACE
  XMLMAP="/home/u49890908/FRED/dow1.map"
  outxml=dow1
  freq='d';

data dow1;
  set fred.dow1 (drop=realtime_start realtime_end);
  newDJIA=input(DJIA, 12.2);
  newDJCA=input(DJCA, 12.2);
  newDJUA=input(DJUA, 12.2);
  newDJTA=input(DJTA, 12.2);
  drop DJIA DJCA DJUA DJTA;
  rename newDJIA=DJIA newDJCA=DJCA newDJUA=DJUA newDJTA=DJTA;
run;
```

A complete list of SASEFRED FREQ=options are found online in the SAS documentation (SAS, 2023). More importantly, Figure 11 shows that the work.dow1 data set now contains daily, numeric data for all of the time series.

Columns		Total rows: 2610 Total columns: 5				
<input checked="" type="checkbox"/> Select all		date	DJIA	DJCA	DJUA	DJTA
<input checked="" type="checkbox"/> 123 date	1	2013-09-02
<input checked="" type="checkbox"/> 123 DJIA	2	2013-09-03	14833.96	5022.09	472.9	6270.98
<input checked="" type="checkbox"/> 123 DJCA	3	2013-09-04	14930.87	5055.43	471.86	6343.95
<input checked="" type="checkbox"/> 123 DJUA	4	2013-09-05	14937.48	5062.68	470.31	6379.61
<input checked="" type="checkbox"/> 123 DJTA	5	2013-09-06	14922.5	5063.42	473.37	6371.11
	6	2013-09-09	15063.12	5114.42	475.61	6460.43
	7	2013-09-10	15191.06	5174.8	479.41	6585.26
	8	2013-09-11	15326.6	5189.98	474.24	6587.55
	9	2013-09-12	15300.64	5164.88	472.64	6516.44
	10	2013-09-13	15376.06	5187.53	476.89	6523.42
	11	2013-09-16	15494.78	5227.55	477.32	6597.15

Figure 11. WORK.DOW1 SAS data set with daily, numeric observations.

There are many options you can specify in SASEFRED, and to illustrate a few more take a look at the following code:

```
libname FRED SASEFRED "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727fa1b89e7c' idlist='DJIA,VXDCLS'
  automap=replace
  XMLMAP="/home/u49890908/FRED/dow3.map" outxml=dow3 freq='d' units='chg'
  start='2015-12-31' end='2016-09-04';
```

```
data dow3;
  set fred.dow3;
  chgDJIA=input(DJIA, 12.2);
  chgVXDCLS=input(VXDCLS, 12.2);
  drop DJIA VXDCLS;
run;
```

```
libname FRED SASEFRED "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727fa1b89e7c' idlist='DJIA,VXDCLS'
  automap=replace
  XMLMAP="/home/u49890908/FRED/dow4.map" outxml=dow4 freq='d' units='lin'
  start='2015-12-31' end='2016-09-04';
```

```
data dow4;
  set fred.dow4;
  linDJIA=input(DJIA, 12.2);
  linVXDCLS=input(VXDCLS, 12.2);
  drop DJIA VXDCLS;
run;
```

```
proc sql;
  create table merged as select
    dow3.date, dow3.chgDJIA, dow3.chgVXDCLS, dow4.linDJIA, dow4.linVXDCLS
  from work.dow3, work.dow4
  where dow3.date=dow4.date;
quit;
```

The first LIBNAME statement sets UNITS= to 'chg' and the second LIBNAME statement sets it equal to 'lin'. In this case, when 'chg' is specified, SASEFRED computes the change in the value of the time series from one day to the next and when 'lin' is specified, which is the default, SASEFRED does not

perform any transformations of the reported data in the FRED® database. SASEFRED can perform many transformations.

Two other options specified in the LIBNAME statement are START= and END=. These two options allow you to specify when you want the time series to start and end. The earliest start date is 1776-07-04 and the latest end date is 9999-12-31. If you don't specify any dates, SASEFRED grabs all observations for the time series you specify in the IDLIST.

Figure 12 shows the work.merged data set. You can verify that the chgDJIA and chgVXDCLS variables are the daily change in the linDJIA and linVSDCLS time series, respectively. In addition, observe that the start date is 2015-12-31.

Table: WORK.MERGED		View: Column names		Filter: (none)			
Columns		Total rows: 177		Total columns: 5			
<input checked="" type="checkbox"/>	Select all	date	chgDJIA	chgVXDCLS	linDJIA	linVXDCLS	
<input checked="" type="checkbox"/>	date	1	2015-12-31	-178.84	0.45	17425.03	17.01
<input checked="" type="checkbox"/>	chgDJIA	2	2016-01-01
<input checked="" type="checkbox"/>	chgVXDCLS	3	2016-01-04	.	.	17148.94	19.12
<input checked="" type="checkbox"/>	linDJIA	4	2016-01-05	9.72	-0.23	17158.66	18.89
<input checked="" type="checkbox"/>	linVXDCLS	5	2016-01-06	-252.15	0.73	16906.51	19.62
		6	2016-01-07	-392.41	3.25	16514.1	22.87
		7	2016-01-08	-167.65	2.18	16346.45	25.05
		8	2016-01-11	52.12	-2.33	16398.57	22.72
		9	2016-01-12	117.65	-1.2	16516.22	21.52
		10	2016-01-13	-364.81	1.95	16151.41	23.47

Figure 12. WORK.MERGED SAS data set with UNITS=, START=, and END= options

PROC TIMEDATA, or other DATA or PROC steps, can accomplish similar results to using UNITS=, START=, and END= in SASEFRED, and it may be the preferred method to filter, modify or transform your data.

SASEFRED: A SIMPLE APPLICATION

To demonstrate the time savings of using SASEFRED with SAS Studio, three time series, DJIA, VXDCLS, and SOFR are downloaded and placed in the fred.dow5 dataset. DJIA is the daily closing level of the Dow Jones Industrial Index, a popular stock price index. VXDCLS is the daily closing level of the volatility of the Dow Jones Industrial Index. And SOFR is the daily secured overnight financing rate, a newer short-term interest rate.

Next, the data is filtered and prepared for analysis in a DATA step. Subsequent code is created by using point-and-click *tasks* in SAS Studio. The results are not shown. The main takeaway is that you can easily access real-world data and immediately begin the process of exploring, preparing, and analyzing your data. This is especially helpful when teaching undergraduate or graduate students with little programming experience. In a 75-minute class session, students can complete the two steps required to use SASEFRED, explore the FRED database structure, learn how to search FRED using SASEFRED, create SAS datasets, and build time series models.

```
libname FRED SASEFRED "/home/u49890908/FRED"
  APIKEY='a941fbd163371e27fe76727fa1b89e7c' idlist='DJIA,VXDCLS,SOFR'
  automap=replace XMLMAP="/home/u49890908/FRED/dow5.map" outxml=dow4
  freq='d'
  units='lin' output=1 start='1776-07-04' end='9999-12-31';
```

```
data work.dow5;
  set fred.dow5 (drop=realtime_start realtime_end);
```

```

where date>'02APR2018'd;
newDJIA=input(DJIA, 12.2);
newVXDCLS=input(VXDCLS, 12.2);
newSOFR=input(SOFR, 12.2);
drop DJIA VXDCLS SOFR;
rename newDJIA=DJIA newVXDCLS=VXDCLS newSOFR=SOFR;
format date date9.;
run;

ods graphics on/imagemap=on;
title1 "Level of the Daily Dow Jones Industrial Average (DJIA) Index";

proc sgplot data=work.dow5;
  series x=date y=djia;
run;

title1;
title1 "Level of the Daily Secured Overnight Financing Rate (SOFR)
";
title2 "Level of the Daily CBOE Dow Jones Industrial Average (DJIA)
Volatility Index (VXDCLS)";

proc sgplot data=work.dow5;
  series x=date y=sofr;
  series x=date y=vxdcls;
run;

title;
title1 "Level of the Daily Dow Jones Industrial Average (DJIA) Index";
title2 "Level of the Daily CBOE DJIA Volatility Index (VXDCLS)      ";

proc sgplot data=dow5;
  series x=date y=djia/lineattrs=(pattern=41);
  series x=date y=vxdcls/y2axis lineattrs=(pattern=15);
run;

title;

/*****
/* Time Series Data Preparation */
*****/
ods noproctitle;

proc sort data=WORK.DOW5 out=Work.preProcessedData;
  by date;
run;

proc timedata data=Work.preProcessedData seasonality=7 out=WORK._tsoutput;
  id date interval=day setmissing=previous;
  var DJIA / accumulate=none transform=none;
  var VXDCLS / accumulate=none transform=none;
  var SOFR / accumulate=none transform=none;
run;

data work.tsPrep(rename=());
  set WORK._tsoutput;
run;

```

```

proc print data=work.tsPrep(obs=10);
  title "Subset of work.tsPrep";
run;

title;

proc delete data=Work.preProcessedData;
run;

proc delete data=WORK._tsoutput;
run;

/*****/
/*Time Series Exploration*/
/*****/
ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TSPREP out=Work.preProcessedData;
  by date;
run;

proc timeseries data=Work.preProcessedData seasonality=7 plots=(series
  histogram corr acf pacf wn);
  id date interval=day;
  var DJIA / accumulate=none transform=none dif=0 sdif=0;
  ods exclude ACFNORMPlot;
  ods exclude PACFNORMPlot;
  ods exclude WhiteNoiseLogProbabilityPlot;
run;

/* Unit root test analysis */
proc arima data=Work.preProcessedData plots=none;
  ods select StationarityTests;
  identify var=DJIA stationarity=(adf=3);
  run;
quit;

proc delete data=Work.preProcessedData;
run;

/*****/
/*Time Series Modeling and Forecasting */
/*****/
/*Radom Walk with NO Drift */
ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TSPREP out=Work.preProcessedData;
  by date;
run;

proc arima data=Work.preProcessedData plots=all out=work.out_random_nodrift;
  identify var=DJIA (1) outcov=work.outcov_random_nodrift;
  estimate noint method=CLS outest=work.outest_random_nodrift
  outstat=work.outstat_random_nodrift outmodel=work.outmodel_random_nodrift;

```

```

forecast lead=12 back=12 alpha=0.05 id=date interval=day;
outlier;
run;
quit;

proc delete data=Work.preProcessedData;
run;

/*Random Walk with Drift */
ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TSPREP out=Work.preProcessedData;
by date;
run;

proc arima data=Work.preProcessedData plots=all out=work.out_random_drift;
identify var=DJIA (1) outcov=work.outcov_random_drift;
estimate method=CLS outest=work.outest_random_drift
outstat=work.outstat_random_drift outmodel=work.outmodel_random_drift;
forecast lead=12 back=12 alpha=0.05 id=date interval=day;
outlier;
run;
quit;

proc delete data=Work.preProcessedData;
run;

/*ARIMA: AR(1) */
ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TSPREP out=Work.preProcessedData;
by date;
run;

proc arima data=Work.preProcessedData plots=all out=work.out_AR1;
identify var=DJIA outcov=work.outcov_AR1;
estimate p=(1) method=ML outest=work.outest_AR1 outstat=work.outstat_AR1
outmodel=work.outmodel_AR1;
forecast lead=12 back=12 alpha=0.05 id=date interval=day;
outlier;
run;
quit;

proc delete data=Work.preProcessedData;
run;

/*ARIMAX: AR(1) VXDCLS */
ods noproctitle;
ods graphics / imagemap=on;

proc sort data=WORK.TSPREP out=Work.preProcessedData;
by date;
run;

```

```

proc arima data=Work.preProcessedData plots(only)=(series(corr crosscorr)
  residual(corr normal) forecast(forecast) );
  identify var=DJIA crosscorr=(VXDCLS);
  estimate p=(1) input=(VXDCLS) method=ML;
  forecast lead=12 back=12 alpha=0.05 id=date interval=day printall;
  outlier;
run;
quit;

proc delete data=Work.preProcessedData;
run;

```

CONCLUSION

The SASEFRED interface engine reads time series data in FRED directly into SAS. Using SASEFRED saves you time because the manual downloading of time series data from FRED and subsequent manual uploading of the data to SAS is eliminated. SAS has other data access engines that read time series data directly from different databases. Table 1 shows a list. Some of these databases require a subscription, however, FRED is a free database.

Table 1. List of SAS Data Access Engines.

Data Access Engine	Database
SASECRESP	Center for Research in Security Prices (CRSP)
SASEFAME	Fame
SASEFRED	Saint Louis Federal Reserve (FRED)
SASEHAVR	Haver Analytics DLX
SASENOAA	National Oceanic and Atmospheric Administration (NOAA)
SASOECD	Organisation for Economic Co-operation and Development (OECD)
SASEQUAN	Quandl
SASERAIN	World Weather Online
SASEWBGO	World Bank Group Open
SASEXCCM	CRSP/Compustat Merged (CCM)
SASEXFSD	FactSet OnDemand

REFERENCES

- Federal Reserve Bank of Saint Louis. (2023, September 4). *API Keys*. Retrieved from FRED Economic Data: https://fred.stlouisfed.org/docs/api/api_key.html
- Federal Reserve Bank of Saint Louis. (2023, August 29). *FRED*. Retrieved August 29, 2023, from Saint Louis Federal Reserve: <https://fred.stlouisfed.org/>
- Federal Reserve Bank of Saint Louis. (2023, September 5). *FRED API*. Retrieved September 5, 2023, from FRED Economic Data: <https://fred.stlouisfed.org/docs/api/fred/>
- Federal Reserve Bank of Saint Louis. (2023, September 4). *User Account*. Retrieved from FRED Economic Data: <https://fredaccount.stlouisfed.org/login/secure/>
- SAS. (2023, September 4). *SAS Viya Platform Programming Documentation*. Retrieved September 4, 2023, from SAS Help Center: https://documentation.sas.com/doc/en/pgmsascdc/v_042/etsug/etsug_sasefred_syntax01.htm#etsug.sasefred.fred-units-opt

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Paul Newsom
University of South Carolina Aiken
1-803-641-3229
pauln@usca.edu