



Next: [Missing values](#) **Up:** [Working with time-series data](#) **Previous:** [Avoiding the lag\(\) function](#)

Rolling regressions without macros

[Boehmer, Broussard, and Kallunki \(2002\)](#) recommend using macros to run rolling regressions. While macros make impossible tasks possible, they aren't particularly efficient. I describe here a macro-independent way of running rolling regressions, and doing similar tasks.

The task is to run rolling 24-month regressions of monthly stock excess returns on the Fama-French-Carhart factors. That is, at the end of each month, I need to:

- Get the last 24 months of excess returns for each stock, checking that there are at least, say, 12 months of data available.
- Get the Fama-French-Carhart factors.
- Run rolling regressions

I first create a dataset containing "rankdates", which are the date identifiers for the rolling regression. A rankdate of 31Dec2001, for instance, uses data from 31Jan2000 to 31Dec2001, inclusive.

To do this, I first get the first and last date for each permno, and then create a complete list by filling in the in-between dates. I do this because if I were to just use the permno-date pairs available in MSF, if a return is missing for a particular month, then that month will not be a rankdate, even though it might be perfectly valid to make that date a rankdate.

```
data firstandlastdates;
  set crsp.msf(keep=permno date);
  by permno; /*MSF is always sorted by permno date*/
  retain firstdate;
  date=intnx('month', date, 1)-1;
  if first.permno then firstdate=date;
  if last.permno then do;
    lastdate=date;
    output;
  end;
run;

data permnosrankdates(rename=(date=rankdate));
  set firstandlastdates;
  date=firstdate;
  do while(date<=lastdate);
    output;
    date=intnx('month', date+1, 1)-1;
  end;
run;
```

For each rankdate, I then get the list of the 24 dates from which that rankdate will use data.

```
data permnosrankdates;
  set permnosrankdates;
  date=rankdate;
  i=1;
  do while(i<=24);
    output;
```

```

        date=intnx('month', date, 0)-1;
        i=i+1;
    end;
run;

```

permnosrankdates is a dataset that looks, in part, like this:

Obs	PERMNO	rankdate	date	i
1	10000	19851231	31DEC1985	1
2	10000	19851231	30NOV1985	2
3	10000	19851231	31OCT1985	3
4	10000	19851231	30SEP1985	4
5	10000	19851231	31AUG1985	5
6	10000	19851231	31JUL1985	6
7	10000	19851231	30JUN1985	7
8	10000	19851231	31MAY1985	8
9	10000	19851231	30APR1985	9
10	10000	19851231	31MAR1985	10
11	10000	19851231	28FEB1985	11
12	10000	19851231	31JAN1985	12
13	10000	19851231	31DEC1984	13
14	10000	19851231	30NOV1984	14
15	10000	19851231	31OCT1984	15
16	10000	19851231	30SEP1984	16
17	10000	19851231	31AUG1984	17
18	10000	19851231	31JUL1984	18
19	10000	19851231	30JUN1984	19
20	10000	19851231	31MAY1984	20
21	10000	19851231	30APR1984	21
22	10000	19851231	31MAR1984	22
23	10000	19851231	29FEB1984	23
24	10000	19851231	31JAN1984	24
25	10000	19860131	31JAN1986	1
26	10000	19860131	31DEC1985	2
27	10000	19860131	30NOV1985	3
28	10000	19860131	31OCT1985	4
29	10000	19860131	30SEP1985	5
30	10000	19860131	31AUG1985	6

We don't need to keep i, but I kept it for clarity.

Once we have this, all we need to do is merge it with the factors and the returns:

```

data ff;
    set ff.factors_monthly(keep=date rf smb hml umd mktrf);
    date=intnx('month', date, 1)-1;
run;

```

```

proc sort data=permnosrankdates;
    by date permno;

```

```

data permnosrankdates;
    merge permnosrankdates(in=a) ff(in=b);
    by date;
    if a and b;
run;

```

```

data msf;
    set crsp.msf(keep=permno date ret);
    where ret is not missing;
    date=intnx('month', date, 1)-1;
run;

```

```

proc sort data=msf;
    by date permno;
run;

```

```
/*permnosrankdates is already sorted*/
```

```
data permnosrankdates;
  merge permnosrankdates(in=a) msf(in=b);
  by date permno;
  if a and b;
run;
```

Notice that I merged by date, not rankdate.

And now all that remains is to calculate excess returns and run the regressions:

```
data permnosrankdates;
  set permnosrankdates;
  exret=ret-rf;
run;

proc sort data=permnosrankdates;
  by permno rankdate;

proc reg data=permnosrankdates outest=est edf;
  by permno rankdate;
  model exret=mktrf smb hml umd;
run;
```

Notice I run proc reg by rankdate, not date.

Running proc print on EST yields the following output:

Obs	PERMNO	rankdate	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	mktrf
1	10000	19860228	MODEL1	PARMS	exret	.	-0.2624	0.000
2	10000	19860331	MODEL1	PARMS	exret	.	1.9027	-32.219
3	10000	19860430	MODEL1	PARMS	exret	.	1.4644	-31.110
4	10000	19860531	MODEL1	PARMS	exret	.	4.3923	2.102
5	10000	19860630	MODEL1	PARMS	exret	.	-14.0666	-191.074
6	10000	19860731	MODEL1	PARMS	exret	0.48716	-0.1872	-4.129
7	10000	19860831	MODEL1	PARMS	exret	0.34479	-0.1559	-2.784
8	10000	19860930	MODEL1	PARMS	exret	0.29117	-0.1216	-0.492
9	10000	19861031	MODEL1	PARMS	exret	0.25216	-0.1214	-0.499
10	10000	19861130	MODEL1	PARMS	exret	0.25809	-0.0767	-1.790
11	10000	19861231	MODEL1	PARMS	exret	0.27266	-0.1228	-0.709
12	10000	19870131	MODEL1	PARMS	exret	0.25251	-0.1217	-0.665
13	10000	19870228	MODEL1	PARMS	exret	0.23828	-0.1090	-0.784
14	10000	19870331	MODEL1	PARMS	exret	0.24214	-0.1270	-1.872
15	10000	19870430	MODEL1	PARMS	exret	0.22978	-0.1254	-1.942
16	10000	19870531	MODEL1	PARMS	exret	0.22052	-0.1194	-1.983
17	10000	19870630	MODEL1	PARMS	exret	0.22052	-0.1194	-1.983

Obs	smb	hml	umd	exret	_IN_	_P_	_EDF_	_RSQ_
1	0.000	0.00	0.000	-1	0	1	0	.
2	0.000	0.00	0.000	-1	1	2	0	1.00000
3	-71.326	0.00	0.000	-1	2	3	0	1.00000
4	340.707	477.88	0.000	-1	3	4	0	1.00000
5	-884.257	-1322.51	506.315	-1	4	5	0	1.00000
6	-13.672	-16.38	7.106	-1	4	5	1	0.05020
7	-8.529	-10.71	5.470	-1	4	5	2	0.54571
8	7.327	3.24	4.754	-1	4	5	3	0.51848
9	7.303	3.22	4.751	-1	4	5	4	0.52996
10	3.312	-0.84	3.657	-1	4	5	5	0.42145
11	4.391	0.97	3.748	-1	4	5	6	0.30088
12	4.378	0.91	3.725	-1	4	5	7	0.30569
13	4.342	-0.04	3.353	-1	4	5	8	0.31433
14	0.837	-3.20	2.548	-1	4	5	9	0.26778
15	0.669	-3.35	2.531	-1	4	5	10	0.27495

16	0.604	-3.41	2.454	-1	4	5	11	0.27148
17	0.604	-3.41	2.454	-1	4	5	11	0.27148

We can tell how many observations we have for each date by looking at the `_EDF_`: the error degrees of freedom. We asked for this by including `edf` in the `proc model` statement. If we want to keep only regressions for which there were 12 observations or more, we keep those where `_EDF_ >= 7`.

EST is the rolling regression output dataset.

This technique has wide application: rolling regressions, obviously, but also rolling standard deviations, moving averages and whatnot.



Next: [Missing values](#) **Up:** [Working with time-series data](#) **Previous:** [Avoiding the lag\(\) function](#)

Andre de Souza 2012-11-19