混频回归与Stata应用：midasreg

王群勇（南开大学数量经济研究所，教授、博士生导师，QunyongWang@outlook.com)

内容

引言

经典的模型中因变量和自变量的观测频率是相同的，但实践中往往需要对不同频率的数据进行回归，比如通过月度数据来实现对季度数据的及时预测。

常见做法：

- 对高频数据进行简单加总或平均为低频数据，这种做法对高频数据不同期的权重约束为相同，不能有效地对低频数据进行预测或解释。
- 低频数据按照某些插值法转换为高频数据。

混频回归（Midas）则是通过对高频变量自动赋权实现对低频变量回归或预测的一种方法，在宏观经济、金融市场预测中得到越来越多的应用。

混频回归模型

Ghysels (2004, 2006), Andreou(2013), Armesto (2010):

$$y_t = x_t\beta + B(L^{1/m};\theta)x_{t-h}^{(m)}\lambda + \epsilon_t.$$

y_t 低频因变量，x_t为低频自变量。$x_t^{(m)}$为高频自变量，m为高频数据的抽样频率。比如，y为季度数据，$x^{(m)}$为月度数据，那么$m = 3$。

$$B(L^{1/m};\theta) = \sum_{k=0}^{K} \alpha_k L^{k/m}; L^{s/m}x_{t-h}^{(m)} = x_{t-h-s/m}^{(m)}.$$

$$B(L^{1/m};\theta)x_{t-h}^{(m)}=\sum_{k=0}^{K}\alpha_k\,L^{k/m}x_{t-h}^{(m)}=\sum_{k=0}^{K}\alpha_k\,x_{t-h-k/m}^{(m)}.$$

U-MIDAS and STEP

α_k无约束，每个滞后项都有不同的系数（Foroni, 2015; Ghysels,2018）。

step：$x_t^{(m)}$的每s个滞后项具有相同的系数。

- 比如，季度对月度数据的混频回归，1、2、3月具有共同的系数，4、5、6月具有共同的系数，依此类推。
- 月度对日度数据的混频回归，每个月的天数不同，可以约束一个固定的步长，比如10.

Almon多项式法

假定系数满足多项式：$\alpha_k=\sum_{j=0}^{p}a_j\,k^j$，

$$\begin{aligned}\alpha_0 &= a_0\\ \alpha_1 &= a_0+a_1+a_2+\cdots+a_p\\ \alpha_2 &= a_0+2a_1+2^2a_2+\cdots+2^pa_p\\ \cdots &= \cdots\\ \alpha_K &= a_0+Ka_1+K^2a_2+\cdots+K^pa_p\end{aligned}$$

Almon多项式法

midas回归可以写为

$$\begin{aligned}y_t &= x_t\beta+\sum_{k=0}^{K}\left(x_{t-k/m}^{(m)}\sum_{j=0}^{p}k^j\,a_j\right)+\epsilon_t\\ &= x_t\beta+\sum_{j=0}^{p}m_{j,t}\,a_j+\epsilon_t\end{aligned},$$

$$m_{j,t}=\sum_{k=0}^{K}k^j\,x_{t-k/m}^{(m)}.$$

回归系数为个数为p而不是k。

normalized exponential Almon weighting

$$\alpha_k = \frac{\exp(k\theta_1 + k^2\theta_2)}{\sum_{j=0}^{K} \exp(j\theta_1 + j^2\theta_2)}$$

$$\begin{aligned} y_t &= x_t\beta + \sum_{k=0}^{K} x_{t-k/s}^{(m)} \left(\frac{\exp(k\theta_1 + k^2\theta_2)}{\sum_{j=0}^{K} \exp(j\theta_1 + j^2\theta_2)} \right) \lambda + \epsilon_t \\ &= x_t\beta + \sum_{j=0}^{k} m_{j,t}\, \lambda + \epsilon_t \end{aligned},$$

$$m_{j,t} = \frac{\exp(k\theta_1 + k^2\theta_2)}{\sum_{j=0}^{K} \exp(j\theta_1 + j^2\theta_2)} x_{t-j/m}^{(m)}.$$

每个高频变量有三个参数，$\lambda, \theta_1, \theta_2$. 如果$\theta_1 = \theta_2 = 0$，即退化为简单平均。

Beta weighting

$$\begin{aligned} y_t &= x_t\beta + \sum_{k=0}^{K} x_{t-k/m}^{(m)} \left(\frac{\omega_k^{\theta_1-1}(1-\omega_k)^{\theta_2-1}}{\sum_{j=0}^{K} \omega_k^{\theta_1-1}(1-\omega_k)^{\theta_2-1}} + \theta_3 \right) \lambda + \epsilon_t \\ &= x_t\beta + \sum_{j=0}^{K} m_{j,t}\, \lambda + \epsilon_t. \end{aligned}$$

$$m_{j,t} = \left(\frac{\omega_k^{\theta_1-1}(1-\omega_k)^{\theta_2-1}}{\sum_{j=0}^{K} \omega_k^{\theta_1-1}(1-\omega_k)^{\theta_2-1}} + \theta_3 \right) x_{t-j/m}^{(m)},$$

$$\omega_i = \begin{cases} \delta, & i = 0 \\ i/K, & i = 1,2,\ldots,K-1 \\ 1-\delta, & i = K \end{cases}$$

其中，δ设定为非常小的数值（程序中设置为$2.22e^{-16}$）。

Beta weighting

Beta函数可以表现为多种形状，递增、递减、水平、U型、倒U型，取决于$\theta_1, \theta_2, \theta_3$.

特殊情况：

- $\theta_1 = \theta_2 = 1$，简单平均。
- $\theta_1 = 1, \theta_2 > 1$，：递减；
- $\theta_1 = 1, \theta_2 < 1$，：递增；
- $\theta_3 = 0$：$\alpha_0 = 0$，$\alpha_K = 0$.

估计方法

NLS

Bayesian estimation

maximum likelihood estimation

nonparametric estimation

AR动态项

$$y_t = \rho y_{t-1} + x_t\beta + B(L^{1/m};\theta)(1-\rho L)x_{t-h}^{(m)}\lambda + \epsilon_t.$$

权重函数的选择

混频回归中权重的函数形式是事先设定的，函数中的参数需要估计。

权重函数的选择：

- 信息准则
- cross-validation：最低预测误差

内容

midasreg功能概览

可以回归多种不同频率的数据（年度-季度（月度等）、季度-月度、月度-日度等等）

多种赋权方法，包括step、U-midas、Almon PDL、normalized exponential Almon、normalized Beta等加权函数

可以根据信息准则、滚动回归和递归回归预测误差等方法自动选择模型

该指令兼容标准的estat、predict等指令。

语法

`midasreg` *depvar* [*if*] [*in*] , `hframe(`*frame*`)` `hvars(`*varlist*`)` [`hlag(`*numlist*`)` `hweight(`*method*`)` `swhich(`*integer*`)` `lagselect(`*spec*`)` `genlink(`*linkname*`)` `ar` `noconstant` `nolog` *options*]

`hframe(`*frame*`)` 高频数据的frame名称。 midasreg默认当前的工作frame为低频数据。

`hvars(`*varlist*`)` 高频变量

`hlag(`*numlist*`)` 高频变量的滞后阶数

语法

`hweight(`*method*, [*spec*]`)`设定赋权方法，包括

- `step` (step weighting)：日度数据默认为10，其它数据默认为观测频数（比如季度对月度回归，那么step=3）。比如，`hweight(step,5)`.
- `umidas` (individual coefficient)
- `pdl` (Almon polynomial distributed lags)：默认为3阶多项式。比如，`hweight(pdl,4)`
- `exp` (Almon exponential coefficient)：
- `beta` (beta weighting)。比如，`hweight(beta, (1 . .))`约束$\theta_1=1$, `hweight(beta, (1 . 0))`约束$\theta_1=1$，$\theta_3=0$。`hweight(beta, (1 1 .))`约束$\theta_1=\theta_2=1$.

语法

`lagselect(`*spec*`)`

first lag, *maximum lag numlis*, *method*, [, *length*]

比如，`lagselect(1, 6/12, ic)` 根据信息准则进行选择，备选滞后项为1/6, 1/7, ..., 1/12.

`lagselect(1, 6/12, rolling, 100)` 根据滚动回归RMSE进行选择，窗口长度为100。

语法

swhich(*integer*)
设置匹配的季节。比如，季度对月度回归，swhich(3)表示利用每个季度的第3个月份进行回归。默认值为0，swhich(0)表示用每个季节的最后一个观测值进行估计。

genlink(*linkname*) 设置低频frame和高频frame的链接名称。

ar: 估计MIDAS-AR模型.

内容

混频回归

midasreg：Stata混频回归程序

案例

结论

季度-月度

```
. frame reset

. use usq, clear

. tsset
        time variable:  qdate, 1947q1 to 2020q1
                delta:  1 quarter

. frame rename default flow
```

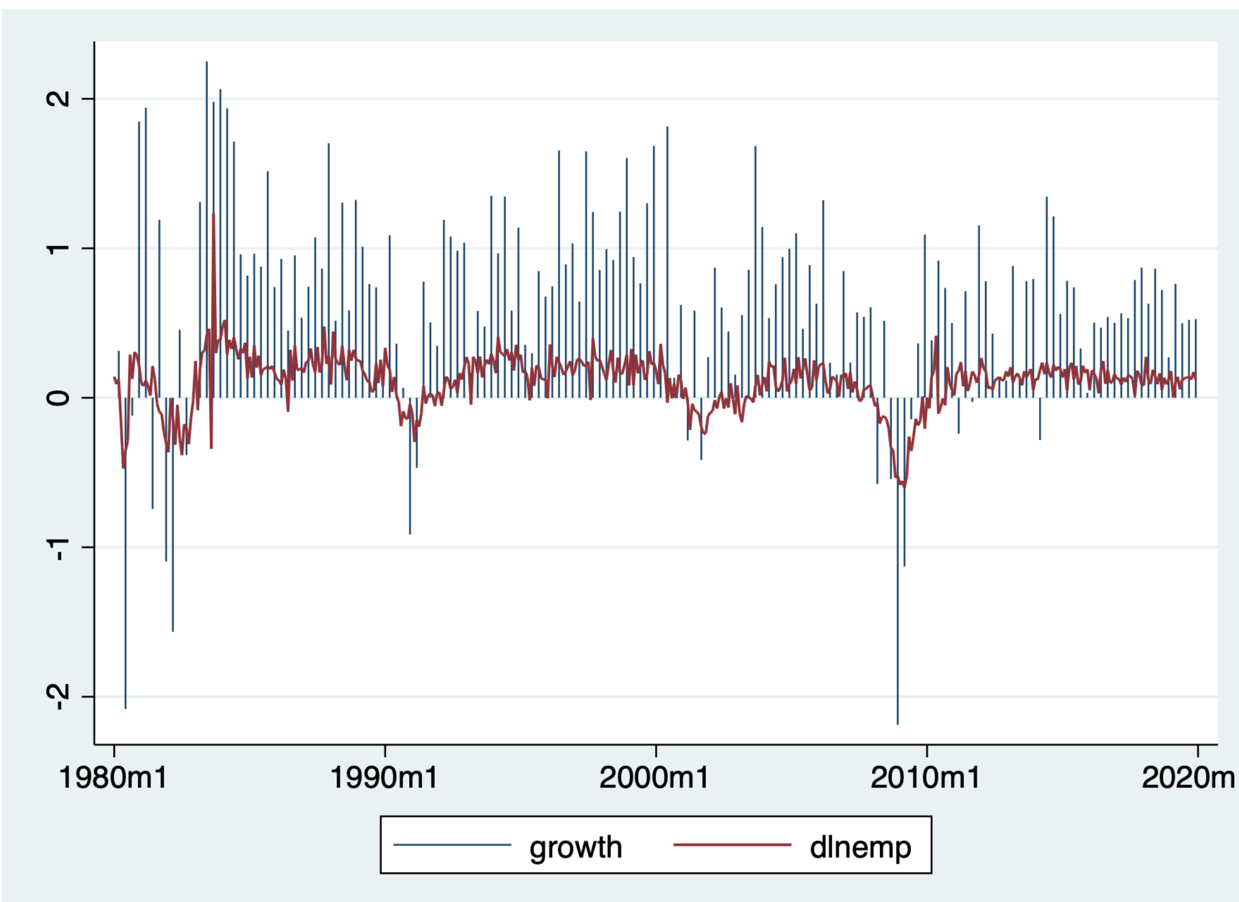
季度-月度

```
. frame create fhigh

. frame change fhigh
```

```
. use usm, clear

. tsset
        time variable:  mdate, 1919m1 to 2020m5
                delta:  1 month

. frame dir
  fhigh  1217 x 10; usm.dta
  flow   293 x 6; usq.dta
```

季度-月度

```
. frame change flow

. midasplot growth if tin(1980q1,2019q4), hvar(dlnemp)
```



季度-月度

```
. frame change flow

. midasreg growth L.growth, hvar(dlnemp) hlag(1/6) hweight(step)

Weight           =             step              Number of obs    =         291
Low frequency    =                q              R-squared        =      0.5092
High frequency   =                m              Adj R-squared    =      0.5041
log-likelihood   =        -289.2290              Root MSE         =      0.6583
AIC              =         586.4580              DW               =      2.1460
BIC              =         601.1513              From             =      1947q3
HQIC             =         592.3442              To               =      2020q1
```

growth	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Low						
growth						
L1.	.0274032	.0589419	0.46	0.642	-.0881208	.1429271
_cons	.3988614	.0503128	7.93	0.000	.3002501	.4974726
High						
dlnemp_Step1	1.240293	.084082	14.75	0.000	1.075496	1.405091
dlnemp_Step2	-.4305778	.0912213	-4.72	0.000	-.6093682	-.2517873

```
. estat lagcoef
```

	Lag1	Lag2	Lag3	Lag4	Lag5	Lag6
dlnemp	1.240293	1.240293	1.240293	-.4305778	-.4305778	-.4305778

季度-月度

$dlnemp_{t-1/3}, dlnemp_{t-2/3}, dlnemp_{t-1}$ 的系数为1.2403

$dlnemp_{t-1-1/3}, dlnemp_{t-1-2/3}, dlnemp_{t-2}$的系数为 -1.4306

midasreg生成低频frame和高频frame的链接，默认名称为`midaslink`。链接可以将低频变量复制到高频数据的frame中，其中包括低频数据中的日期变量。

季度-月度

默认方法为Almon PDL法：

```
. midasreg growth L.growth, hvar(dlnemp) hlag(1/6)

Weight              =              pdl          Number of obs    =          291
Low frequency       =                q          R-squared        =       0.5370
High frequency      =                m          Adj R-squared    =       0.5288
log-likelihood      =        -280.7410          Root MSE         =       0.6350
AIC                 =         573.4820          DW               =       2.1402
BIC                 =         595.5220          From             =       1947q3
HQIC                =         582.3113          To               =       2020q1
```

growth	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Low						
growth						
L1.	.0552539	.0577501	0.96	0.339	-.0579342	.1684419
_cons	.3929575	.0491353	8.00	0.000	.296654	.4892609
High						
dlnemp_PDL0	1.168311	.1461796	7.99	0.000	.8818043	1.454818
dlnemp_PDL1	1.128427	.2966547	3.80	0.000	.5469941	1.709859
dlnemp_PDL2	-.8469424	.1601267	-5.29	0.000	-1.160785	-.5331
dlnemp_PDL3	.1109222	.0222431	4.99	0.000	.0673265	.1545179

```
. estat lagcoef
```

	Lag1	Lag2	Lag3	Lag4	Lag5	Lag6
dlnemp	1.168311	1.560717	.9247721	-.0739917	-.7700407	-.4978418

季度-月度

```
. midasreg growth L.growth, hvar(dlnemp) hlag(1/6) hweight(exp)
Iteration 0:   f(p) = -410.27425  (not concave)
Iteration 1:   f(p) = -326.61472  (not concave)
Iteration 2:   f(p) = -317.53566
Iteration 3:   f(p) = -302.38784
Iteration 4:   f(p) = -299.95699
Iteration 5:   f(p) = -299.63817
```

```
Iteration 6:   f(p) = -299.63203
Iteration 7:   f(p) = -299.63201

Weight              =            exp          Number of obs   =        291
Low frequency       =              q          R-squared       =     0.4728
High frequency      =              m          Adj R-squared   =     0.4635
log-likelihood      =      -299.6320          Root MSE        =     0.6775
AIC                 =       609.2640          DW              =     1.6259
BIC                 =       627.6306          From            =     1947q3
HQIC                =       616.6218          To              =     2020q1
------------------------------------------------------------------------------
      growth |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Low          |
      growth |
         L1. |   -.092836   .0556763    -1.67   0.095    -.2019596    .0162875
             |
       _cons |   .3987914   .0524296     7.61   0.000     .2960312    .5015517
-------------+----------------------------------------------------------------
High         |
dlnemp_theta1|   .7590752   .2869594     2.65   0.008     .1966451    1.321505
dlnemp_theta2|  -.4977317   .1351623    -3.68   0.000    -.7626449   -.2328185
dlnemp_lambda|    18.8855   1.412575    13.37   0.000      16.1169    21.65409
-------------+----------------------------------------------------------------
Variance     |
     lnsigma |  -.3892753   .0414513    -9.39   0.000    -.4705184   -.3080322
------------------------------------------------------------------------------

. estat lagcoef

             |       Lag1        Lag2        Lag3        Lag4        Lag5        Lag6
-------------+-------------------------------------------------------------------------
      dlnemp |   6.212527    8.068046    3.872079    .6867454    .0450115    .0010903
```

季度-月度

```
. midasreg growth L.growth, hvar(dlnemp dlnip) hlag(1/6) hweight(pdl)

Weight              =            pdl          Number of obs   =        291
Low frequency       =              q          R-squared       =     0.6281
High frequency      =              m          Adj R-squared   =     0.6162
log-likelihood      =      -248.8628          Root MSE        =     0.5691
AIC                 =       517.7255          DW              =     2.0365
BIC                 =       554.4588          From            =     1947q3
HQIC                =       532.4410          To              =     2020q1
------------------------------------------------------------------------------
```

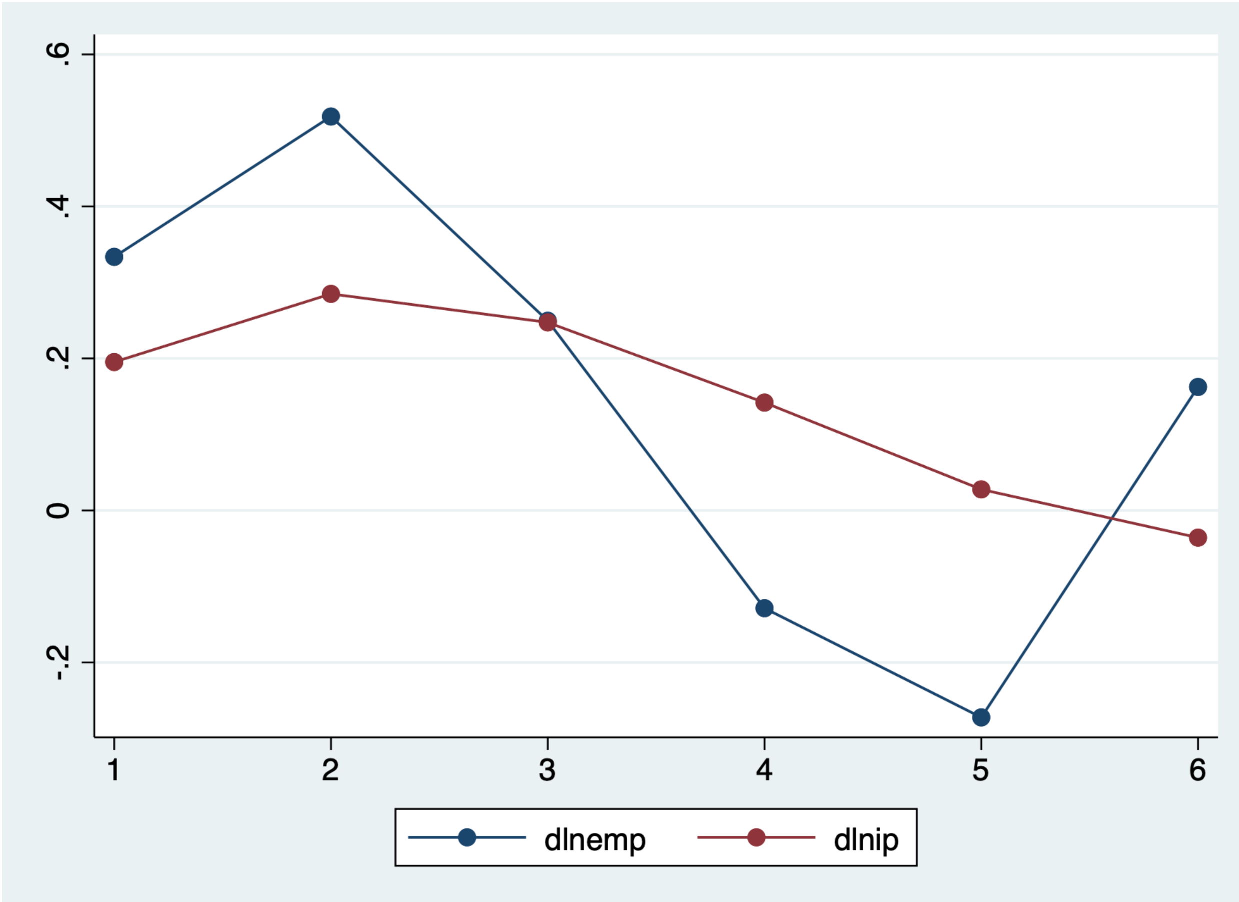
growth	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Low						
growth						
L1.	-.0611198	.0594779	-1.03	0.304	-.1776942	.0554547
_cons	.4876619	.0495333	9.85	0.000	.3905783	.5847454
High						
dlnemp_PDL0	.3336255	.2039422	1.64	0.102	-.0660938	.7333448
dlnemp_PDL1	.5259734	.3510672	1.50	0.134	-.1621056	1.214052
dlnemp_PDL2	-.3986482	.1783656	-2.24	0.025	-.7482384	-.049058
dlnemp_PDL3	.0573218	.0238161	2.41	0.016	.0106431	.1040005
dlnip_PDL0	.1952605	.0534325	3.65	0.000	.0905348	.2999861
dlnip_PDL1	.1731124	.0993305	1.74	0.081	-.0215718	.3677967
dlnip_PDL2	-.0932891	.0496416	-1.88	0.060	-.1905849	.0040067
dlnip_PDL3	.0098849	.0065704	1.50	0.132	-.0029928	.0227625

```
. estat lagcoef
```

	Lag1	Lag2	Lag3	Lag4	Lag5	Lag6
dlnemp	.3336255	.5182725	.2495537	-.1286004	-.272259	.1625083
dlnip	.1952605	.2849687	.2474079	.1418875	.0277168	-.035795

季度-月度

```
estat lagplot
```



季度-月度

```
. qui midasreg growth L.growth, hvar(dlnemp dlnip) hlag(1/6) hweight(pdl)
swhich(2)

. tsappend, add(1)

. capture drop xb

. predict xb if tin(2020q2, 2020q2), xb
Linear prediction

. list xb if tin(2020q1, 2020q2)

     +-----------+
     |        xb |
     |-----------|
293. |         . |
294. |  -9.29716 |
     +-----------+
```

季度-月度

```
. midasreg growth L.growth, hvars(dlnemp) hweight(pdl) lagselect(1, 6/10, "ic")
```

lag	logl	aic	bic	hqic	N	rmse
6	-277.3018	566.6036	588.6021	575.4183	289	.6316519
7	-279.1216	570.2433	592.2418	579.058	289	.635642
8	-283.6477	579.2955	601.294	588.1102	289	.6456753
9	-279.1426	570.2852	592.263	579.0926	288	.6378237
10	-276.6253	565.2505	587.2283	574.0579	288	.6322729

季度-月度

```
. midasreg growth L.growth, hvars(dlnemp) hweight(step) lagselect(1, 6/15,
"rolling", 120)
```

	lag	rmse
r1	6	.6802348
r2	7	.6764846
r3	8	.66101
r4	9	.6530716
r5	10	.6555312
r6	11	.6568462
r7	12	.651029
r8	13	.6513945
r9	14	.650822
r10	15	.6526567

季度-月度

```
. frame change flow

. matrix wmat = J(5,4,.)

. local r=1

. foreach w in step umidas pdl exp beta {
  2. if "`w'"=="beta" qui midasreg growth L.growth, hvar(dlnemp) hlag(1/12)
hweight(`w',(1 . .))
  3. else qui midasreg growth L.growth, hvar(dlnemp) hlag(1/12) hweight(`w')
  4. matrix wmat[`r',1] = e(aic)
  5. matrix wmat[`r',2] = e(bic)
  6. matrix wmat[`r',3] = e(hqic)
  7. matrix wmat[`r',4] = e(r2a)
  8. local ++r
  9. }

. matrix colnames wmat = aic bic hqic r2a

. matrix rownames wmat = step umidas pdl exp beta
```

季度-月度

```
. matlist wmat

             |       aic        bic       hqic        r2a
-------------+--------------------------------------------
        step |   559.633    581.673   568.4623   .5508333
      umidas |  561.6664   613.0929   582.2681   .5594227
         pdl |    573.94   595.9799   582.7693   .5280965
         exp |  609.2641   627.6307   616.6219   .4635164
        beta |  609.7307   631.7706     618.56   .4644568
```

季度-月度

```
. midasreg growth, hvar(dlnemp) hlag(1/12) hweight(step) ar
Iteration 0:   f(p) = -275.67904
Iteration 1:   f(p) = -273.88477
Iteration 2:   f(p) =  -273.8615
Iteration 3:   f(p) =  -273.8615
```

```
Weight             =           step          Number of obs     =        291
Low frequency      =              q          R-squared         =     0.5461
High frequency     =              m          Adj R-squared     =     0.5365
log-likelihood     =      -273.8615          Root MSE          =     0.6200
AIC                =       559.7230          DW                =     1.8039
BIC                =       581.7629          From              =     1947q3
HQIC               =       568.5523          To                =     2020q1
```

growth	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Low						
_cons	.5394313	.0554337	9.73	0.000	.4307832	.6480795
High						
dlnemp_Step1	1.223584	.0794717	15.40	0.000	1.067822	1.379346
dlnemp_Step2	-.2382278	.1057195	-2.25	0.024	-.4454343	-.0310213
dlnemp_Step3	-.04271	.1042566	-0.41	0.682	-.2470491	.1616292
dlnemp_Step4	-.3064493	.0767596	-3.99	0.000	-.4568953	-.1560033
AR						
alpha	-.1735674	.1223173	-1.42	0.156	-.4133049	.0661701
Variance						
lnsigma	-.4778337	.0414513	-11.53	0.000	-.5590768	-.3965906

Note: rho=2*invlogit(alpha)-1; alpha=logit(rho/2+0.5)

季度-月度

```
. nlcom 2*invlogit(_b[AR:alpha])-1

       _nl_1:  2*invlogit(_b[AR:alpha])-1
```

growth	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_nl_1	-.0865665	.0607003	-1.43	0.154	-.2055369	.032404

月度-日度

```
. frame reset

. use inf, clear

. frame rename default flow

. frame create fhigh

. frame change fhigh

. use dff, clear

. frame change fhigh

. frame dir
  fhigh  24079 x 3; dff.dta
  flow   880 x 6; inf.dta
```

月度-日度

```
. frame change flow

. midasreg inf L(1/2).inf, hvars(rate) hweight(pdl) lagselect(10, 20/30, "ic")
```

lag	logl	aic	bic	hqic	N	rmse
20	45.32334	-76.64668	-43.94245	-64.07574	790	.2284793
21	45.54079	-77.08158	-44.37735	-64.51063	790	.2284164
22	45.22059	-76.44118	-43.73695	-63.87024	790	.228509
23	46.31941	-78.63881	-45.93458	-66.06787	790	.2281914
24	46.65315	-79.30629	-46.60206	-66.73535	790	.228095
25	45.20624	-76.41248	-43.70825	-63.84154	790	.2285131
26	45.39014	-76.78029	-44.07606	-64.20935	790	.2284599
27	45.61395	-77.2279	-44.52367	-64.65696	790	.2283952
28	45.47567	-76.95134	-44.24711	-64.3804	790	.2284352
29	45.18555	-76.37109	-43.66686	-63.80015	790	.2285191
30	45.14523	-76.29045	-43.58622	-63.71951	790	.2285308

月度-日度

```
. midasreg inf L(1/2).inf, hvars(rate) hweight(pdl) hlag(10/56)

Weight            =              pdl          Number of obs   =          789
Low frequency     =                m          R-squared       =       0.4697
High frequency    =                d          Adj R-squared   =       0.4656
log-likelihood    =          54.2279          Root MSE        =       0.2259
AIC               =         -94.4559          DW              =       2.0202
BIC               =         -61.7605          From            =       1954m8
HQIC              =         -81.8876          To              =       2020m4

------------------------------------------------------------------------------
         inf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Low          |
         inf |
         L1. |   .4140089   .0353673    11.71   0.000     .3446903    .4833275
         L2. |    .068718   .0354223     1.94   0.052    -.0007085    .1381445
             |
       _cons |   .0305118   .0135641     2.25   0.024     .0039266    .0570969
-------------+----------------------------------------------------------------
High         |
   rate_PDL0 |   .0225777   .0062312     3.62   0.000     .0103648    .0347907
   rate_PDL1 |  -.0056913   .0014387    -3.96   0.000    -.0085111   -.0028715
   rate_PDL2 |   .0003202   .0000755     4.24   0.000     .0001722    .0004681
   rate_PDL3 |  -4.80e-06   1.09e-06    -4.42   0.000    -6.93e-06   -2.67e-06
------------------------------------------------------------------------------
```

不规则日度数据

```
. use daily, clear

. list daten bai nor in 5/8

     +----------------------------------+
     |     daten         bai        nor |
     |----------------------------------|
  5. | 04dec2014    .3359717    .066735 |
  6. | 05dec2014   -.1052678   .0460042 |
  7. | 08dec2014    .4580529   .0256586 |
  8. | 09dec2014   -.0206548    .018157 |
     +----------------------------------+
```

不规则日度数据

```
. frame reset

. use monthly

. tsset mdate, monthly
       time variable:  mdate, 2014m11 to 2019m12
               delta:  1 month

. frame rename default flow
```

不规则日度数据

```
. frame create fhigh

. frame change fhigh

. use daily

. bcal create midascal , from(daten) gen(bcdate) replace

  Business calendar midascal (format %tbmidascal):

    purpose:

      range:  28nov2014  31dec2019
                 20055      21914    in %td units
                     0       1204    in %tbmidascal units

     center:  28nov2014
                 20055               in %td units
                     0               in %tbmidascal units

    omitted:        655              days
                  128.6              approx. days/year

   included:      1,205              days
                  236.6              approx. days/year

  Notes:

    business calendar file midascal.stbcal saved
```

```
    variable bcdate created; it contains business dates in %tbmidascal format

. tsset bcdate
        time variable:  bcdate, 28nov2014 to 31dec2019
                delta:  1 day
```

谢 谢！

更多计量方法与Stata应用，请扫码关注公众号StataPLUS:

或咨询：StataPLUS@outlook.com