Bayesian Time Series in Stata

David Schenck

Senior Econometrician
Stata

Swiss Stata Conference
November 18, 2022

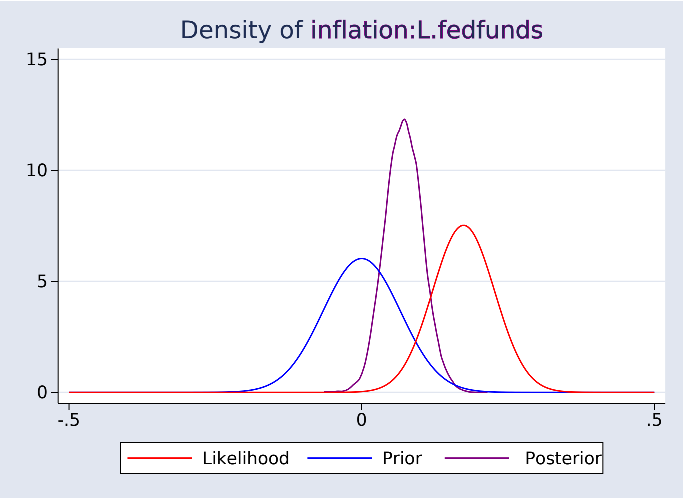
Bayesian econometrics in Stata

- Bayesian estimation has already been available for many cross-section econometric models
- Linear regression
  - `bayes: regress`
- Probit regression
  - `bayes: probit`
- Ordered probit regression
  - `bayes: oprobit`
- Poisson regression
  - `bayes: poisson`
- ...and more
- Arbitrary Bayesian models
  - `bayesmh`

Bayesian econometrics in Stata 17

- Stata 17 introduces Bayesian estimation of a variety of time–series and panel–data econometric models
  - Multivariate time-series:
    - `bayes: var`
    - `bayes: dsge`
    - `bayes: dsgenl`
    - `bayesirf`
    - `bayesfcast`
- Plan for the talk:
  - Review of Stata's `bayes:` environment
  - Bayesian VAR models with `bayes: var`
  - Bayesian DSGE models with `bayes: dsge`

Bayesian basics



Bayesian basics in Stata

- `regress y x`
- `bayes:` `regress y x`
- `bayes` *, prior_opts bayes_opts* `: regress y x` *, options*
  - `prior_opts` control aspects of the prior distributions
  - `bayes_opts` control aspects of the MCMC process
  - `options` control aspects of the likelihood model

Bayesian basics in Stata

```
. sysuse auto
(1978 automobile data)

. regress price mpg weight

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(2, 71)        =     14.74
       Model |   186321280         2  93160639.9   Prob > F        =    0.0000
    Residual |   448744116        71  6320339.67   R-squared       =    0.2934
-------------+----------------------------------   Adj R-squared   =    0.2735
       Total |   635065396        73  8699525.97   Root MSE        =      2514

------------------------------------------------------------------------------
       price | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         mpg |  -49.51222   86.15604    -0.57   0.567    -221.3025     122.278
      weight |   1.746559   .6413538     2.72   0.008      .467736    3.025382
       _cons |   1946.069    3597.05     0.54   0.590    -5226.245    9118.382
------------------------------------------------------------------------------
```

Bayesian basics in Stata

```
. sysuse auto
(1978 automobile data)
. bayes, rseed(18) : regress price mpg weight

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  price ~ regress(xb_price,{sigma2})
Priors:
  {price:mpg weight _cons} ~ normal(0,10000)                              (1)
                  {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_price.
Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         74
                                                 Acceptance rate  =      .3558
                                                 Efficiency:  min =     .06611
                                                              avg =       .104
Log marginal-likelihood = -696.94328                          max =      .1874

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
price        |
         mpg |  -4.05281   26.92524   .939398  -3.578713  -59.57364   48.81809
      weight |  2.067784   .1965986   .006932    2.06595   1.691988   2.447179
       _cons | -.6431985   99.66187   3.87603  -2.295516  -198.9983   199.0333
-------------+----------------------------------------------------------------
      sigma2 |   6398701    1066214   24632.4    6276103    4650364    8791535
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
Note: Adaptation tolerance is not met in at least one of the blocks.
```

Bayesian basics in Stata

```
. bayes, rseed(18) mcmcsize(20000) burnin(10000) : regress price mpg weight

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  price ~ regress(xb_price,{sigma2})
Priors:
  {price:mpg weight _cons} ~ normal(0,10000)                              (1)
                  {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_price.
Bayesian linear regression                       MCMC iterations  =     30,000
Random-walk Metropolis-Hastings sampling         Burn-in          =     10,000
                                                 MCMC sample size =     20,000
                                                 Number of obs    =         74
                                                 Acceptance rate  =      .2895
                                                 Efficiency:  min =     .06562
                                                              avg =      .1045
Log marginal-likelihood = -696.89977                          max =      .1973

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
price        |
         mpg | -5.163804   27.88902   .680289  -5.762984  -59.56813   50.21124
      weight |  2.070849    .197542   .005232   2.065953   1.678947    2.45389
       _cons |  .4716185   98.25035     2.712   1.604825  -191.8309   195.8815
-------------+----------------------------------------------------------------
      sigma2 |   6411526    1089118   17339.9    6306473    4585308    8847979
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

Bayesian postestimation in Stata

- General Bayesian postestimation features
  - `bayesstats grubin`
  - `bayesstats ppvalues`
  - `bayesstats summary`
  - `bayesstats ess`
  - `bayespredict`
  - `bayesgraph`

Bayesian postestimation in Stata

```
. bayesstats summary ({price:mpg})
Posterior summary statistics                      MCMC sample size =     20,000
      expr1 : {price:mpg}
```

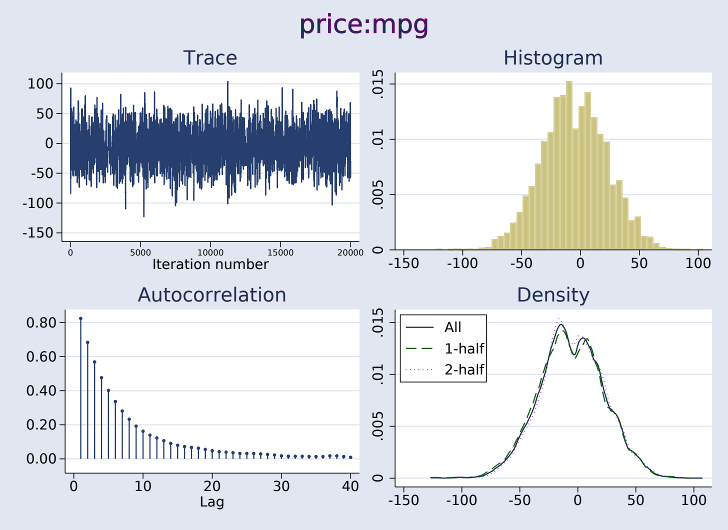
	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
expr1	-5.163804	27.88902	.680289	-5.762984	-59.56813	50.21124

```
.
.
.
.
. bayesstats summary ({price:mpg} + {price:weight})
Posterior summary statistics                      MCMC sample size =     20,000
      expr1 : {price:mpg} + {price:weight}
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
expr1	-3.092955	27.71758	.675873	-3.665984	-57.20129	51.94755

Bayesian postestimation in Stata

```
. bayesgraph diagnostics {price:mpg}
```



`bayes: var`

The autoregressive model

- An autoregression expresses a variable as a function of its own lags

$$y_t = a_0 + a_1 y_{t-1} + \cdots + a_p y_{t-p} + \mathbf{c}\mathbf{x}_t + u_t \qquad u_t \sim N(0, \sigma^2)$$

- y_t is the dependent variable
- u_t is a disturbance term with variance σ^2
- $(a_0, \ldots, a_p)$ are coefficients
- Model can include exogenous variables $\mathbf{x}_t$ with coefficients $\mathbf{c}$
- Choice of the maximum p lag

The vector autoregression model

- A VAR expresses a collection of variables as functions of their lags

$$\mathbf{y}_t = \mathbf{a} + \mathbf{A}_1\mathbf{y}_{t-1} + \cdots + \mathbf{A}_p\mathbf{y}_{t-p} + \mathbf{C}\mathbf{x}_t + \mathbf{u}_t \qquad \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma})$$

- $\mathbf{y}_t$ is a vector of k variables
- $\mathbf{u}_t$ is a vector of k disturbance terms with $k \times k$ covariance matrix $\mathbf{\Sigma}$
- $(\mathbf{A}_1, \ldots, \mathbf{A}_p)$ are $k \times k$ matrices of parameters
- Model can include exogenous variables $\mathbf{x}_t$ with coefficients $\mathbf{C}$
- Choice of k variables, p lags

VAR estimation

- Flexible setup with minimal structure
- But: many parameters to estimate ($k^2 p$ slope coefficients, k constant terms, $k(k+1)/2$ elements of $\boldsymbol{\Sigma}$)
- The large number of parameters to be estimated can lead to imprecise estimates

The Bayesian VAR

- Likelihood model: observables $\mathbf{y}_t$ follow

$$\mathbf{y}_t = \mathbf{a} + \mathbf{A}_1 \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{C}\mathbf{x}_t + \mathbf{u}_t, \qquad \mathbf{u}_t \sim N(\mathbf{0}, \mathbf{\Sigma})$$

- Prior for coefficients $\boldsymbol{\beta} = vec(\mathbf{A}_1, \ldots, \mathbf{A}_p, \mathbf{C})$ is multivariate normal

$$\boldsymbol{\beta} \sim N(\boldsymbol{\beta}_0, \mathbf{\Omega})$$

- Prior for covariance matrix $\mathbf{\Sigma}$ is either inverse Wishart or Jeffreys

Bayesian VAR priors I

- Look at a two-variable VAR with two lags for simplicity:

$$y_t = a_{11} y_{t-1} + a_{12} p_{t-1} + b_{11} y_{t-2} + b_{12} p_{t-2} + u_{1t}$$
$$p_t = a_{21} y_{t-1} + a_{22} p_{t-1} + b_{21} y_{t-2} + b_{22} p_{t-2} + u_{2t}$$

- Three types of slope coefficients:
  1. First autoregressive lag (red)
  2. Other autoregressive lags (blue)
  3. Cross-lags (black)
- Priors are specified for each type of coefficient

Bayesian VAR priors II

- Priors are normally distributed
- Prior means:
  - First autoregressive lag: prior mean 1
  - Further autoregressive lags: prior mean 0
  - All cross-lags: prior mean 0
- Prior variances:

$$\text{Autoregressive lags} = \left(\frac{\lambda_1}{\ell^{\lambda_3}}\right)^2 \qquad \text{Cross lags} = \frac{\sigma_i^2}{\sigma_j^2}\left(\frac{\lambda_1\lambda_2}{\ell^{\lambda_3}}\right)^2$$

- Interpretation:
  - λ_1: autoregressive lag tightness (default: 0.1)
  - λ_2: cross-lag tightness (default: 0.5)
  - λ_3: lag attenuation (default: 1)

Bayesian VAR priors III

- The upshot: the prior model is a random walk, with prior variances that are tighter around 0 as the lag length increases

$$
\begin{aligned}
y_t &= a_{11} y_{t-1} + a_{12} p_{t-1} + b_{11} y_{t-2} + b_{12} p_{t-2} + u_{1t} \\
p_t &= a_{21} y_{t-1} + a_{22} p_{t-1} + b_{21} y_{t-2} + b_{22} p_{t-2} + u_{2t}
\end{aligned}
$$

Bayesian VAR priors III

- The upshot: the prior model is a random walk, with prior variances that are tighter around 0 as the lag length increases

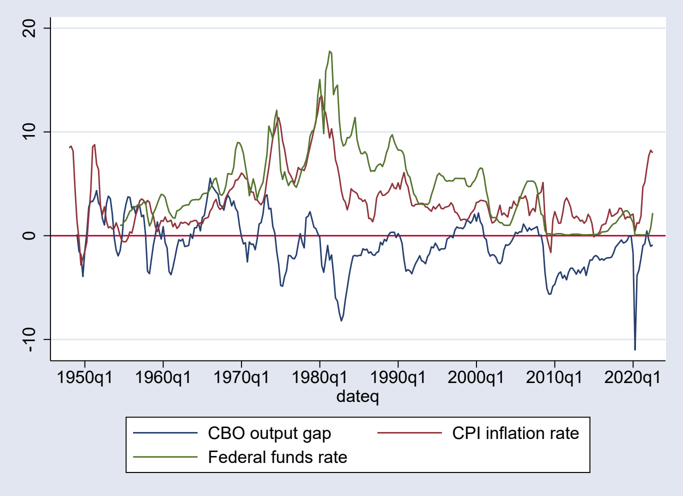
$$
\begin{aligned}
y_t &= a_{11} y_{t-1} + a_{12} p_{t-1} + b_{11} y_{t-2} + b_{12} p_{t-2} + u_{1t} \\
p_t &= a_{21} y_{t-1} + a_{22} p_{t-1} + b_{21} y_{t-2} + b_{22} p_{t-2} + u_{2t}
\end{aligned}
$$

US macro data

```
. use usmacro.dta

. describe

Contains data from usmacro.dta
 Observations:           304
    Variables:             5                  17 Nov 2022 16:47
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
fedfunds        float   %9.0g                 Federal funds rate
year            float   %9.0g
dateq           float   %tq
ogap            float   %9.0g                 CBO output gap
inflation       float   %9.0g                 CPI inflation rate
-------------------------------------------------------------------------------
Sorted by: dateq
```

US macro data

```
. tsline ogap inflation fedfunds, yline(0)
```



bayes: var

```
. bayes, rseed(17) : var inflation ogap fedfunds, lags(1/4)

Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  inflation
       ogap
   fedfunds ~ mvnormal(3,xb_inflation,xb_ogap,xb_fedfunds,{Sigma,m})

Priors:
  {inflation:L(1 2 3 4).inflation}                                         (1)
       {inflation:L(1 2 3 4).ogap}                                         (1)
   {inflation:L(1 2 3 4).fedfunds}                                         (1)
                 {inflation:_cons}                                         (1)
       {ogap:L(1 2 3 4).inflation}                                         (2)
            {ogap:L(1 2 3 4).ogap}                                         (2)
        {ogap:L(1 2 3 4).fedfunds}                                         (2)
                      {ogap:_cons}                                         (2)
   {fedfunds:L(1 2 3 4).inflation}                                         (3)
        {fedfunds:L(1 2 3 4).ogap}                                         (3)
    {fedfunds:L(1 2 3 4).fedfunds}                                         (3)
                  {fedfunds:_cons} ~ varconjugate(3,4,1,_b0,{Sigma,m},_Phi0)
                                     (3)
                         {Sigma,m} ~ iwishart(3,5,_Sigma0)
------------------------------------------------------------------------------
```

bayes: var

```
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_inflation.
(2) Parameters are elements of the linear form xb_ogap.
(3) Parameters are elements of the linear form xb_fedfunds.

Bayesian vector autoregression                   MCMC iterations  =     12,500
Gibbs sampling                                   Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Sample: 1955q3 thru 2022q3                       Number of obs    =        269
                                                 Acceptance rate  =          1
                                                 Efficiency:  min =      .9668
                                                              avg =      .9968
Log marginal-likelihood =  -1006.706                          max =          1
```

bayes: var

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
inflation						
inflation						
L1.	1.068163	.036918	.000369	1.06795	.9964609	1.14132
L2.	-.0599346	.0360274	.00036	-.0603735	-.1291826	.0114401
L3.	-.0310739	.0253892	.000254	-.0309561	-.0810876	.0188703
L4.	-.0299672	.019333	.000193	-.030066	-.0678256	.0082601
ogap						
L1.	.0404599	.026625	.000264	.0403996	-.0124314	.092381
L2.	.0074879	.0236302	.000236	.0074819	-.0394403	.0546182
L3.	-.0015834	.0169899	.00017	-.0014802	-.0348289	.031621
L4.	-.011837	.0131239	.00013	-.0117894	-.0375505	.0134571
fedfunds						
L1.	.0893568	.0333011	.000333	.0893335	.0238991	.1544197
L2.	-.0344558	.0316572	.000317	-.0345512	-.0964539	.0279689
L3.	-.0205046	.022671	.000224	-.0205149	-.065278	.0240842
L4.	-.0306231	.0172882	.000175	-.0309885	-.0644584	.0031282
_cons	.2230265	.0706231	.000706	.2236592	.0840805	.3619018

bayes: var

ogap						
inflation						
L1.	-.0808699	.0598309	.000608	-.0809985	-.1977924	.0367735
L2.	.0163327	.0586755	.000587	.0162722	-.0983593	.1312959
L3.	-.0021872	.0414088	.000414	-.0019286	-.0842381	.0784458
L4.	.0019039	.0317501	.000318	.0017431	-.0608111	.0631296
ogap						
L1.	.9233403	.0426501	.000427	.9233905	.8409345	1.007648
L2.	-.0030224	.0377026	.000377	-.0029346	-.0769842	.0701085
L3.	-.0193673	.0277016	.000277	-.0191945	-.0728326	.0353898
L4.	-.0121582	.0213866	.000218	-.0120346	-.0545398	.0289959
fedfunds						
L1.	.0525289	.053672	.000537	.0520162	-.0519092	.1585224
L2.	-.0621234	.051396	.000514	-.0619072	-.1628952	.0378323
L3.	.0036568	.0365474	.000365	.0034682	-.0665011	.0764284
L4.	-.0017984	.0280812	.000284	-.0021882	-.056955	.0537327
_cons	.1515797	.1144606	.001129	.1516304	-.074704	.3774874

bayes: var

fedfunds						
inflation						
L1.	.0147784	.0471859	.000472	.0152286	-.0766183	.1074257
L2.	.0423197	.0464882	.000465	.042183	-.0477244	.1330823
L3.	.009068	.0320992	.000326	.0089749	-.0541776	.0720777
L4.	.000943	.0246132	.000246	.0008175	-.046718	.0503128
ogap						
L1.	.1218024	.0333336	.000333	.1221063	.0566122	.1880058
L2.	-.0240608	.0293329	.000293	-.0242413	-.0826579	.0334335
L3.	-.0212177	.0214086	.000214	-.0211516	-.0627405	.0210806
L4.	-.0154994	.0167963	.000166	-.0155045	-.0481395	.0172744
fedfunds						
L1.	1.012207	.0425085	.000425	1.012351	.9282363	1.096246
L2.	-.091243	.0399355	.000393	-.0917477	-.1680828	-.0123084
L3.	.0130398	.0287797	.000283	.0132559	-.0428932	.0693724
L4.	-.0022905	.0221361	.000221	-.0024244	-.0460434	.0416874
_cons	.1424249	.0889396	.000872	.143151	-.0341906	.3129186

bayes: var

Sigma_1_1	.4037014	.0350521	.000351	.4016238	.341854	.4800336
Sigma_2_1	.1558131	.0412935	.0004	.1546991	.0786388	.2402095
Sigma_3_1	.1220397	.0322698	.000318	.1207185	.0613803	.1894467
Sigma_2_2	1.072202	.091977	.00092	1.06718	.9077038	1.265749
Sigma_3_2	.2782432	.0536041	.000536	.2764703	.1784471	.3876833
Sigma_3_3	.6520289	.0560112	.00056	.649563	.551443	.7731006

VAR postestimation

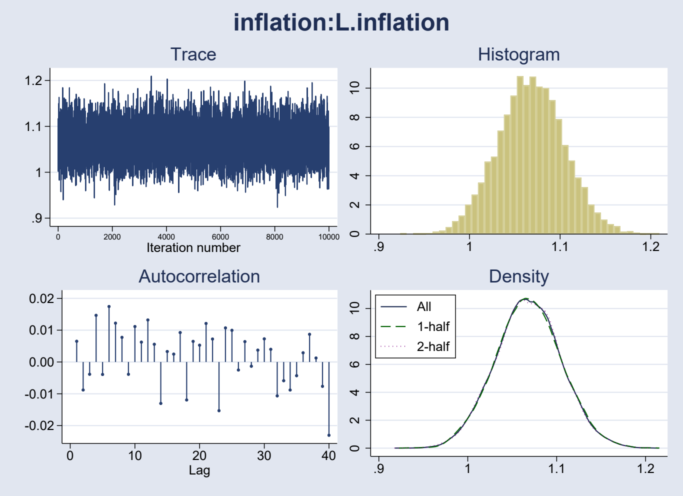
- VAR coefficients are not the only object of interest
- We are usually also interested in
  - VAR stability
  - forecasting
  - impulse response analysis
- ... which are functions of the VAR coefficients
- We can also use standard Bayesian postestimation features.

Bayesian VAR postestimation

- Specialized postestimation for Bayesian VARs:
  - `bayesvarstable`
  - `bayesfcast`
  - `bayesirf`
- General Bayesian postestimation features
  - `bayesstats grubin`
  - `bayesstats ppvalues`
  - `bayesstats summary`
  - `bayesstats ess`
  - `bayespredict`
  - `bayesgraph`

Posterior parameter diagnostics

```
. bayesgraph diagnostics {inflation:L.inflation}
```

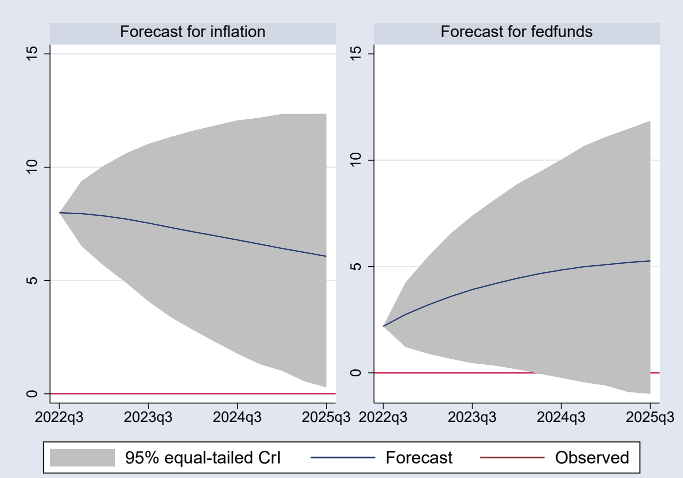


VAR forecasting

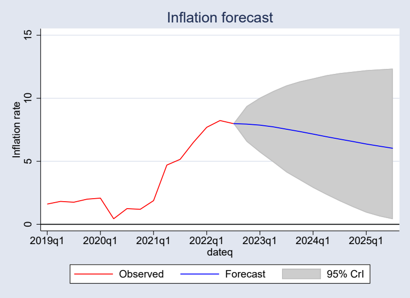
- Compute a forecast $(\widehat{\mathbf{y}}^s_{T+1}, \dots, \widehat{\mathbf{y}}^s_{T+h})$ using each of the MCMC draws
- . . . to arrive at the whole posterior distribution of forecasts
- Stata: `bayesfcast compute` and `bayesfcast graph`

bayesfcast

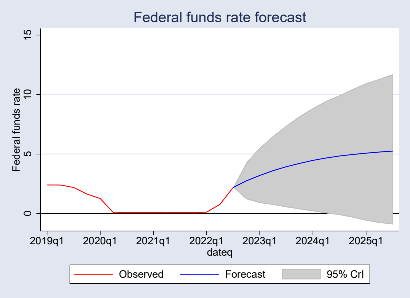
```
. bayes, saving(bayesvar.dta, replace)
. bayesfcast compute bf_, step(12)
. bayesfcast graph  bf_inflation bf_fedfunds, observed yline(0) legend(rows(1))
```



bayesfcast



Inflation forecast
Inflation rate
0
5
10
15
2019q1
2020q1
2021q1
2022q1
2023q1
2024q1
2025q1
dateq
Observed
Forecast
95% CrI



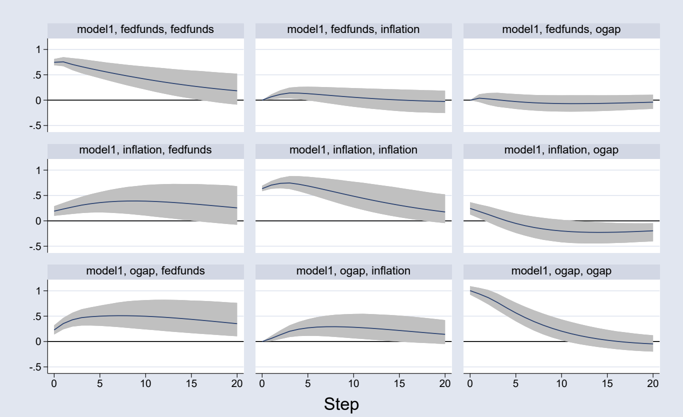
Federal funds rate forecast
Federal funds rate
0
5
10
15
2019q1
2020q1
2021q1
2022q1
2023q1
2024q1
2025q1
dateq
Observed
Forecast
95% CrI

Impulse response functions

- Impulse response functions trace out how a shock to one equation affects model variables
- `bayesirf` is a suite of commands for creating, managing, and analyzing impulse response functions (patterned after `irf`)
  - `bayesirf set`
  - `bayesirf create`
  - `bayesirf table`
  - `bayesirf graph`
  - ...among others
- `bayesirf create` builds a collection of IRF results, including
  - simple IRF (`irf`)
  - orthogonalized IRF (`oirf`)
  - cumulative IRF (`cirf`, `coirf`)
  - forecast error variance decomposition (`fevd`)
  - dynamic multiplier (`dm`) in the presence of exogenous variables

bayesirf

```
. bayesirf set bvarirf.irf, replace
(file bvarirf.irf created)
(file bvarirf.irf now active)

. bayesirf create model1, step(20)
(file bvarirf.irf updated)

. bayesirf graph oirf, yline(0, lcolor(black))
```



bayes: var options

- Controlling the prior: `minnconjprior()`
  - `mean(#)`
  - `selftight(#)` (default: 0.1)
  - `crosstight(#)` (default: 0.5)
  - `lagdecay(#)` (default: 1)
- Examples:

```
. webuse usmacro
. matrix arvec = (0, 0, 0)
. bayes, minnconjprior(mean(arvec)) : var inflation ogap fedfunds
. bayes, minnconjprior(selftight(0.3)) : var inflation ogap fedfunds
. bayes, minnconjprior(crosstight(1)) : var inflation ogap fedfunds
. bayes, minnconjprior(lagdecay(0.5)) : var inflation ogap fedfunds
```

`bayes: dsge`

Dynamic Stochastic General Equilibrium models

- Vector autoregression models have a minimum of structure
  - choose variables and lag length, and perhaps order
- Dynamic stochastic general equilibrium models have lots of structure
  - n variables in n equations
  - Equations can feature lags and leads
  - Some components are latent (unobserved)
  - Equations are motivated by economic theory
- DSGE models are solved into state-space form and estimated based on the likelihood of the state-space solution

A simple DSGE model

- A model with 3 control variables, driven by 2 state variables
- Equations:

$$
\begin{aligned}
x_t &= E_t x_{t+1} - \sigma(r_t - E_t p_{t+1} - z_t) \\
p_t &= \beta E_t p_{t+1} + \kappa x_t \\
r_t &= \frac{1}{\psi} p_t + w_t \\
z_{t+1} &= \rho_z z_t + \epsilon_{t+1} \\
w_{t+1} &= \rho_w w_t + e_{t+1}
\end{aligned}
$$

- x_t is the output gap, p_t is inflation, r_t is the interest rate
- z_t and w_t are driving state variables
- e_t and ϵ_t are shocks

Priors for DSGE model parameters

- VARs tend to have many parameters
- DSGEs tend to have fewer parameters, and parameters tend to have immediate theoretical interpretation
- Many DSGE model parameters have natural bounds that provide useful priors
  - With AR(1) state variables, autoregressive parameters must lie in (-1,1) for stability
  - Many parameters represent shares or rates that must lie in (0,1)
  - Beta distributions lie in (0,1) and give extra weight to specific parts of that interval, making them a popular choice

bayes: dsge

```
. webuse usmacro2

. bayes, prior({beta},  beta(95, 5))   prior({kappa}, beta(30,70))     ///
>        prior({sigma}, beta(10,90))   prior({psi},   beta(67,33))     ///
>        prior({rhow},  beta(10, 10))  prior({rhoz},  beta(35,15))     ///
>        rseed(17) dots burnin(5000) mcmcsize(30000):                  ///
>        dsge (x   = F.x - {sigma}*(r - F.p - z)    , unobserved )     ///
>             (p   = {beta}*F.p + {kappa}*x                     )     ///
>             (r   = (1/{psi})*p + w                            )     ///
>             (F.z = {rhoz}*z              , state              )     ///
>             (F.w = {rhow}*w              , state              )
note: initial parameter vector set to means of priors.

Burn-in 5000 aaaaaaaaa1000aaaaaaaaa2000aaaa.....3000.........4000.........5000
> done
Simulation 30000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000........
> .11000.........12000.........13000.........14000.........15000.........16000.
> ........17000.........18000.........19000.........20000.........21000........
> .22000.........23000.........24000.........25000.........26000.........27000.
> ........28000.........29000.........30000 done
```

bayes: dsge

```
Model summary
--------------------------------------------------------------------------------
Likelihood:
  p r ~ dsgell({sigma},{beta},{kappa},{psi},{rhoz},{rhow},{sd(e.z)},{sd(e.w)})

Priors:
            {sigma} ~ beta(10,90)
             {beta} ~ beta(95,5)
            {kappa} ~ beta(30,70)
              {psi} ~ beta(67,33)
             {rhoz} ~ beta(35,15)
             {rhow} ~ beta(10,10)
  {sd(e.z) sd(e.w)} ~ igamma(.01,.01)
--------------------------------------------------------------------------------
```

bayes: dsge

```
Bayesian linear DSGE model                          MCMC iterations  =     35,000
Random-walk Metropolis-Hastings sampling            Burn-in          =      5,000
                                                    MCMC sample size =     30,000
Sample: 1955q1 thru 2015q4                          Number of obs    =        244
                                                    Acceptance rate  =      .1741
                                                    Efficiency:  min =    .005331
                                                                 avg =     .01032
Log marginal-likelihood = -787.73905                             max =     .01974
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
sigma	.1443227	.0292876	.001318	.1432416	.088498	.2043906
beta	.9547238	.0203592	.001276	.9576523	.9077723	.9848212
kappa	.3419745	.0457376	.003318	.3415295	.2517864	.4357346
psi	.6527897	.043041	.003403	.6529768	.5686343	.7351054
rhoz	.9078086	.0157278	.00091	.9080455	.8749843	.9369648
rhow	.7546737	.0269813	.001109	.7541327	.7017534	.8085894
sd(e.z)	.6048148	.0950875	.005623	.5951787	.4475909	.8237128
sd(e.w)	1.955303	.1265823	.008904	1.948905	1.734296	2.230285

Bayesian DSGE postestimation

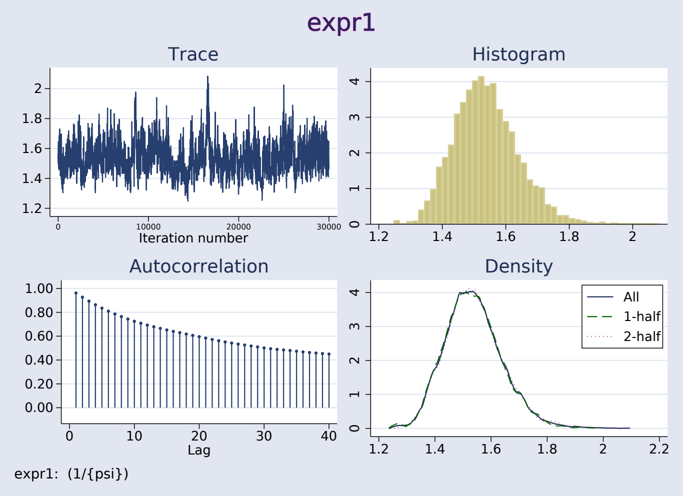
- Specialized postestimation for Bayesian DSGEs:
  - `bayesirf`
- General Bayesian postestimation features
  - `bayesstats grubin`
  - `bayesstats ppvalues`
  - `bayesstats summary`
  - `bayesstats ess`
  - `bayesgraph`

Posterior parameter diagnostic plots

```
. bayesstats summary {psi}

Posterior summary statistics                      MCMC sample size =     30,000

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
         psi |  .6527897    .043041   .003403   .6529768   .5686343   .7351054
------------------------------------------------------------------------------

.
.
.
. bayesstats summary (1/{psi})

Posterior summary statistics                      MCMC sample size =     30,000

       expr1 : 1/{psi}

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
       expr1 |  1.538681   .1035836   .008064   1.531448   1.360349     1.7586
------------------------------------------------------------------------------
```

Posterior parameter diagnostic plots

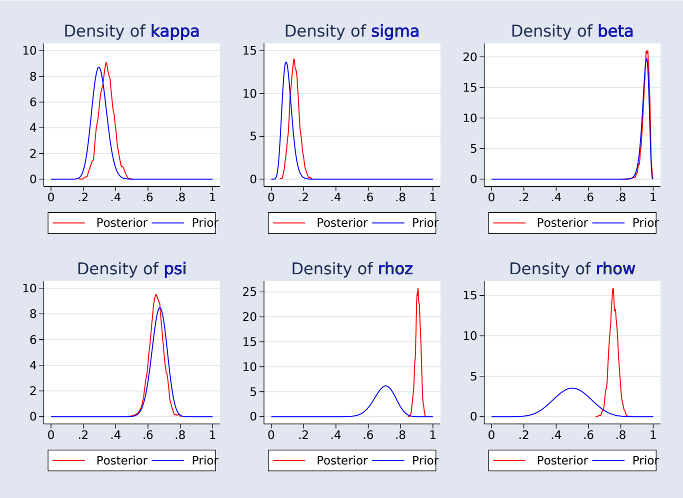
```
. bayesgraph diagnostics (1/{psi})
```



Posterior parameter distribution plots

```
. bayesgraph kdensity {kappa}, lcolor(red)         ///
>         addplot(function Prior=betaden(30,70,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(kappa) nodraw

.
. bayesgraph kdensity {sigma}, lcolor(red)         ///
>         addplot(function Prior=betaden(10,90,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(sigma) nodraw

.
. bayesgraph kdensity {beta}, lcolor(red)         ///
>         addplot(function Prior=betaden(95,5,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(beta) nodraw

.
. bayesgraph kdensity {psi}, lcolor(red)           ///
>         addplot(function Prior=betaden(67,33,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(psi) nodraw

.
. bayesgraph kdensity {rhoz}, lcolor(red)          ///
>         addplot(function Prior=betaden(35,15,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(rhoz) nodraw

.
. bayesgraph kdensity {rhow}, lcolor(red)          ///
>         addplot(function Prior=betaden(10,10,x), ///
>         legend(on label(1 "Posterior")) lcolor(blue)) name(rhow) nodraw

.
. graph combine kappa sigma beta psi rhoz rhow
```

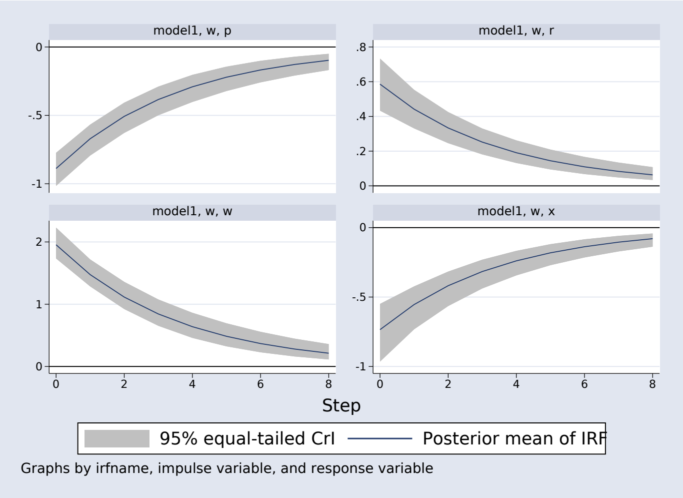
Posterior parameter distribution plots



Impulse response functions

```
. bayesirf set bdsgeirf.irf, replace
(file bdsgeirf.irf created)
(file bdsgeirf.irf now active)

. bayesirf create model1
(file bdsgeirf.irf updated)

. bayesirf graph irf, impulse(w) response(p x r w)    ///
>         byopts(yrescale) yline(0, lcolor(black))
```

Impulse response functions



Recap

- Stata 17 introduces Bayesian estimation of a variety of time–series and panel–data econometric models
  - Multivariate time-series:
    - `bayes: var`
    - `bayes: dsge`
    - `bayes: dsgenl`
    - `bayesirf`
    - `bayesfcast`

Thank you!