Bayesian Analysis

Isabel Cañette

Associate Director of Technical Services and
Principal Mathematician & Statistician
StataCorp LLC

2022 Colombian Stata Conference
6-8 September 2022

Bayesian vs classical statistics

In classical statistical analysis, we assume fixed unknown parameters, a dataset generated with a distribution based on them, and we use the data to construct an estimate of those underlying parameters.

In Bayesian statistic, parameters are considered random, according to a distribution, and our aim is to use previous knowledge of this distribution to estimate an updated version of it conditional on the observed data.

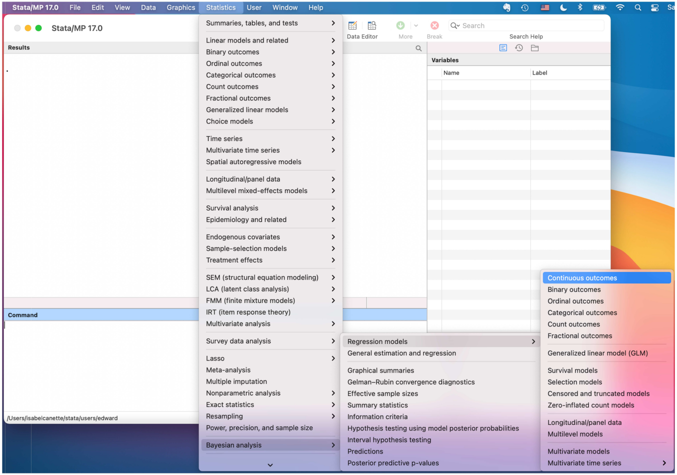
Stata commands for Bayesian estimation

- `bayes:` prefix provides a simple way to fit bayesian regression models. For example:

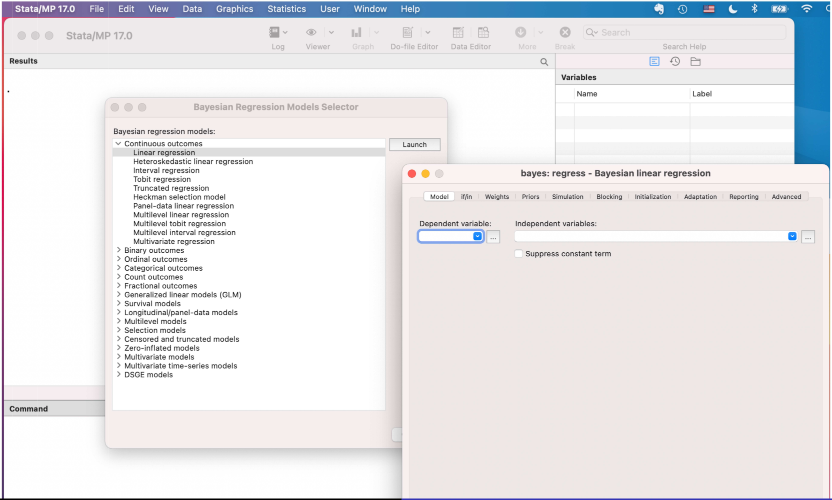
  ```
  . bayes: regress y x1 x2
  ```

  It supports a wide range of commands including regressions for continous, binary, ordinal, categorical, count or fractional outcomes, survival analysis, sample selection, panel data, multilevel, time series, and dynamic stochastic general equilibrium models. Type `help bayes estimation` to see the complete list.
- `bayesmh` allows us to fit customized Bayesian regressions by choosing among a set of available prior and likelihood functions, or with evaluators provided by the user. It can be used for linear and/or non-linear, one-level or multilevel, and one or multiple-equations models.

Stata Graphical User Interface



Stata Graphical User Interface



Stata's Bayesian suite consists of the following commands

Command	*Description*
Estimation	
`bayes:`	Bayesian regression models using the `bayes` prefix
`bayesmh`	General Bayesian models using MH
`bayesmh` *evaluators*	User-defined Bayesian models using MH
Postestimation	
`bayesgraph`	Graphical convergence diagnostics
`bayesstats ess`	Effective sample sizes and more
`bayesstats grubin`	Gelman–Rubin convergence diagnostics
`bayesstats summary`	Summary statistics
`bayesstats ic`	Information criteria and Bayes factors
`bayestest model`	Model posterior probabilities
`bayestest interval`	Interval hypothesis testing
`bayespredict`	Bayesian predictions (available only after `bayesmh`)
`bayesstats ppvalues`	Bayesian predictive *p*-values (available only after `bayesmh`)

Bayes' Theorem:

$$p(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{m(y)}$$

- Assume that we know $\pi(\theta)$ ("prior")
- We have already assumed that we know $f(Y|\theta) = L(y;\theta)$
- We observe the data, Y

Bayes' theorem tell us that we can obtain the "posterior" distribution of the parameter, $p(\theta|y)$

$$p(\theta|y) \propto L(y;\theta) \times \pi(\theta)$$

In theory, we don't need the constant because densities integrate to 1. In practice, we won't need the constant to simulate a sample for $p(\theta|y)$.

Example: weight of sugar packets.

Let's assume we have a random sample $y_1, \ldots y_{70} \sim N(\mu, \sigma^2)$ and we are interested in estimating the mean, μ. This can be estimated as the constant of a regression without covariates.

```
. use sugar, clear
(Weights of Domino sugar packets, Triola, Elementary Statistics.)

. regress weight , noheader
```

weight	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
_cons	3.586043	.0088481	405.29	0.000	3.568391	3.603694

The Bayesian version would be:

```
. bayes, rseed(3876): regress weight ,vsquish

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  weight ~ regress({weight:_cons},{sigma2})
Priors:
  {weight:_cons} ~ normal(0,10000)
        {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------

Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         70
                                                 Acceptance rate  =      .4382
                                                 Efficiency:  min =      .1988
                                                              avg =      .2231
Log marginal-likelihood =  66.950733                          max =      .2475
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
weight						
_cons	3.586146	.009181	.000185	3.586458	3.567973	3.604502
sigma2	.0059266	.0010415	.000023	.0057973	.0042246	.0083358

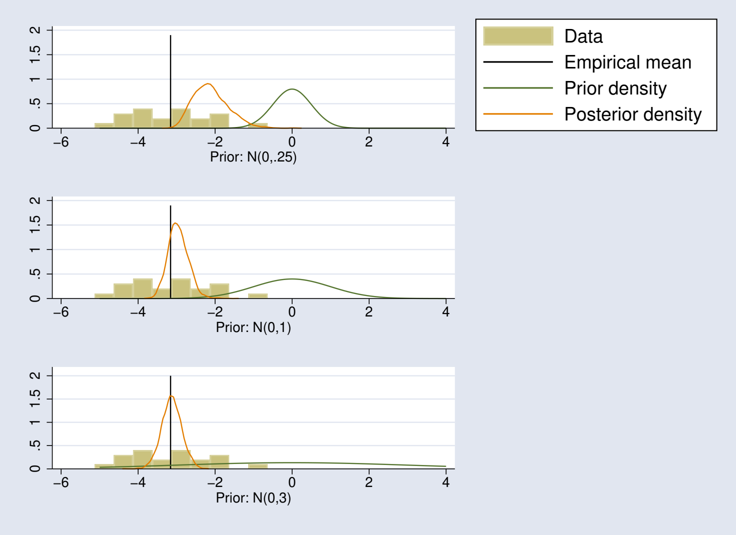
Note: Default priors are used for model parameters.

Model summary:

```
. bayes, rseed(3876): regress weight ,vsquish notable

Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  weight ~ regress({weight:_cons},{sigma2})

Priors:
  {weight:_cons} ~ normal(0,10000)
        {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------

Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         70
                                                 Acceptance rate  =      .4382
                                                 Efficiency:  min =      .1988
                                                              avg =      .2231
Log marginal-likelihood =  66.950733                          max =      .2475
```

Why are we using this prior by default?

The less "informative" the prior, the more we rely on the data.



The table

```
. bayes, rseed(3876) : regress weight ,vsquish noheader

Burn-in ...
Simulation ...
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
weight						
_cons	3.586146	.009181	.000185	3.586458	3.567973	3.604502
sigma2	.0059266	.0010415	.000023	.0057973	.0042246	.0083358

- Mean, median and std. dev. are estimates of the mean, the median and the standard deviation of the posterior distribution.
- A 95% credibility interval is interpreted as an interval such us the probability of the parameter being there is 0.95.

How is this density estimated? Because there is, in most cases, not a closed form for the posterior distribution, this is estimated via simulation (i.e., generating a large random sample of this distribution rather than having a functional form).
We use MCMC, i.e. create an ergodic Markov Chain whose limit (stationary) distribution is theoretically proven to be the posterior we are looking for.

Stata implements two methods: Gibbs sampling and Metropolis-Hastings algorithm.

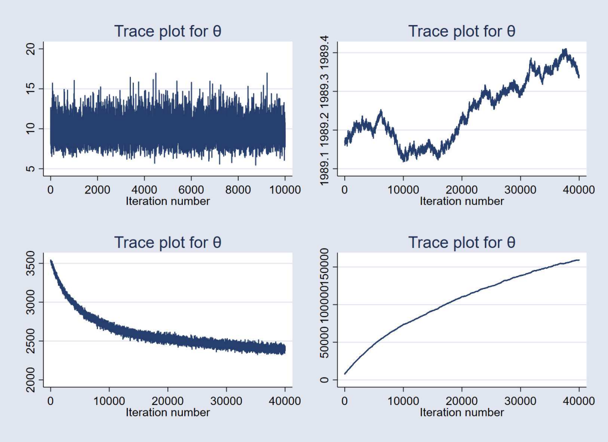
Metropolis-Hasting algorithm.
We choose a "proposal" distribution $q(.)$ (unrelated with our prior or our posterior, we actually use a Gaussian distribution) and start with θ_0 in the domain of the posterior p. Then, for each iteration t:

- Generate a proposal state $\theta_* \sim q(.|\theta)$
- Compute the acceptance probability

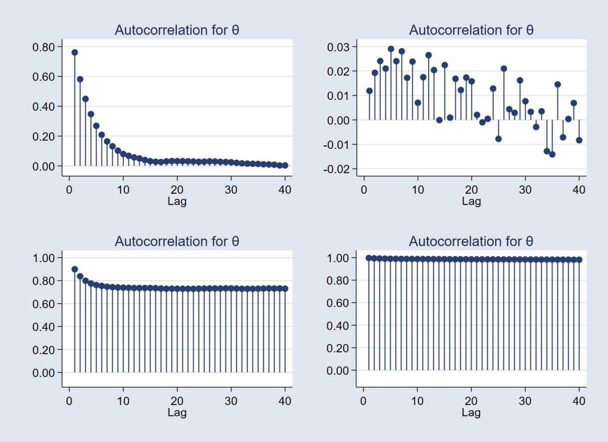
$$r(\theta_*|\theta_{t-1}) = \frac{p(\theta_*|y)}{p(\theta_{t-1}|y)}$$

- We accept θ_* with probability $r(\theta_*|\theta_{t-1})$ (or with probability 1 if $r(\theta_*|\theta_{t-1}) > 1$).
- Accepting means $\theta_t = \theta_*$; otherwise $\theta_t = \theta_{t-1}$

Trace plots: converge is achieved if the simulated values reach stationarity.



Autocorrelation plots: we expect the correlation to be negligible after a few lags. High autocorrelations imply low efficiency, so reaching stationarity will take more iterations than for more efficient problems.



Example: bikes rentals vs weather

```
. use bikes, clear
(Bike sharing dataset, Hadi Fanaee-T)

. describe

Contains data from bikes.dta
 Observations:           731                  Bike sharing dataset, H. Fanaee-T
    Variables:             4                  4 Sep 2022 16:08
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
precip          byte    %15.0g     preclab    Precipitation
ntemp           float   %9.0g                 Normalized Temperature (Celsius)
count100        float   %9.0g                 Hundreds of bikes rented
temp            float   %9.0g                 Temperature (Celsius)
--------------------------------------------------------------------------------
```

We fit a Bayesian linear model to the rental counts ($\times 0.01$) vs temperature and indicators of levels of precipitations (we set a seed for reproducibility).

```
. bayes, rseed(1357): regress count100 temp i.precip
```

```
Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  count100 ~ regress(xb_count100,{sigma2})
Priors:
  {count100:temp i.precip _cons} ~ normal(0,10000)                          (1)
                        {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_count100.
Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        731
                                                 Acceptance rate  =      .3475
                                                 Efficiency:  min =       .051
                                                              avg =     .09622
Log marginal-likelihood = -3008.9227                          max =      .2236
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
count100						
temp	1.347828	.0626141	.002614	1.346385	1.233055	1.469331
precip						
Mist	-5.802201	1.160169	.047753	-5.785858	-8.105655	-3.573989
Light rain/snow	-25.8168	3.281888	.109326	-25.84465	-32.31738	-19.29535
_cons	27.13917	1.198321	.053062	27.12183	24.85218	29.45991
sigma2	206.305	10.80463	.228511	206.0586	185.9393	228.7717

The header is:

```
. bayes, rseed(1357) nomodelsummary:regress count100 temp i.precip, vsquish

Burn-in ...
Simulation ...
Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        731
                                                 Acceptance rate  =      .3758
                                                 Efficiency:  min =     .02825
                                                              avg =     .07818
Log marginal-likelihood = -3013.5765                          max =      .2101
```

- Marginal log-likelihood $m(y) = p(Y = y|(\theta \sim M))$ $= \int (p(y|\theta, M)p(\theta|M)\, d\theta$. (i.e., integrate $p(y|\theta)$ over the distribution M of θ), evaluated at the observed data y.
- MCMC iterations - total number of iterations
- Burn-in - discarded iteration to eliminate influence of the initial values
- MCMC sample size - iterations used for estimation
- Acceptance rate - fraction of proposal values accepted. We expected it to be neither too small nor too large - optimal value for multivariate posteriors and proposal: 0.234; for univariate posteriors:0.45
- Efficiency - Indicator of the mixing quality of the chain

The table:

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
count100						
temp	.0135365	.0006187	.000034	.0135331	.0122963	.01472
precip						
Mist	-5.747835	1.143186	.042002	-5.774701	-7.882743	-3.509883
Light rain/snow	-25.65604	3.173647	.149708	-25.59269	-31.96759	-19.44203
_cons	27.00479	1.169644	.069591	27.02894	24.73269	29.2549
sigma2	206.1646	10.93242	.238485	205.9812	185.4657	228.1383

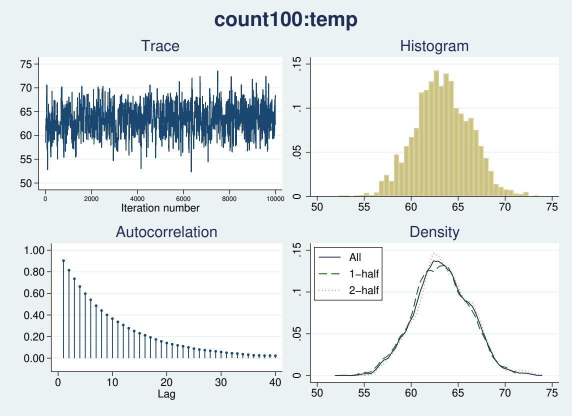
Note: Default priors are used for model parameters.

- Mean ($\hat{\theta}$), median and standard deviation ($\hat{s}$) are the mean, the median and the standard deviation of the posterior sample. Estimate respectively the mean ($E(\theta_t)$), the median and the standard deviation $\sqrt{Var(\theta_t)}$ of the posterior distribution.
- A 95% credibility interval - an interval such us the probability of the parameter being there is 0.95.
- MCSE - Monte Carlo standard error - an indicator of the precision of the sample posterior mean ("Mean" in the table). MCSE($\hat{\theta}$)=$\hat{s}/\sqrt{\text{ESS}}$

Convergence is attained when the chain achieves stationarity (and therefore the sample is drawn from the posterior distribution).

- Inspecting mixing and time trends within the chains of individual parameters
  - `bayesgraph diagnostics, trace, ac, histogram, kdensity`
  - `bayesgraph csum`
  - `bayesstats ess`
- Inspecting multiple chains for each parameter
  - `bayesgraph diagnostics, trace, ac, histogram, kdensity`
  - `bayesgraph rubin`

bayesgraph diagnostics - it needs to be run for each parameter



- The trace doesn't show convergence problems
- Correlation becomes negligible after 20 lags
- Density estimates with first and second half look similar

```
. bayesstats ess

Efficiency summaries        MCMC sample size =     10,000
                            Efficiency:  min =     .02825
                                         avg =     .07818
                                         max =      .2101
```

	ESS	Corr. time	Efficiency
count100			
temp	334.69	29.88	0.0335
precip			
Mist	740.79	13.50	0.0741
Light rain/snow	449.39	22.25	0.0449
_cons	282.49	35.40	0.0282
sigma2	2101.42	4.76	0.2101

- ESS -Effective sample size - Number of i.i.d observations that would contain the same information as in our MCMC sample.
- Corr time - T/ESS - Number of iterations where autocorrelation becomes negligible (T=MCMC sample size).
- Efficiency - ESS/T - Indicator of the the mixing quality of the MCMC procedure. The higher the better.
  - Efficiencies over 10% are considered good for MH.
  - Efficiencies under 1% would be a source of concern.

See Methods and Formulas section in manual entry for [BAYES] `bayesstats ess` for details.

Bayesian predictions: `bayespredict`

We started with a prior distribution, $\pi(\theta)$, and updated that prior with the information in our dataset, y, obtaining the posterior distribution, $p(\theta)$.

Now we can consider that the data we have already observed are fixed, and the actual distribution of θ is our posterior p. Under this assumption, we can predict the distribution of future outcomes, y^{new}.

Assuming that $\theta \sim p$, and we can use this (posterior) distribution and the likelihood ($f(y|\theta)$) to compute the predictive posterior distribution for a new value y^{new} of Y:

$$p(y^{new}) = \int f(y|\theta)p(\theta)\, d\theta.$$

We can see it as:

$$p(y^{new}|y^{obs}) = \int f(y|\theta)p(\theta|y^{obs})\, d\theta.$$

To obtain predictions, first we fit our model with `bayesmh`.

```
bayesmh count100 temp i.precip, ///
  likelihood(normal({sigma2}))  ///
  prior({count100:}, normal(0, 10000)) ///
  prior({sigma2}, igamma(.01, .01)) rseed(2476) ///
  saving(bikespost, replace)
```

We saved the simulated values for the posterior distribution of the parameters in a new file (`bikepost`). This file will be needed to perform predictions.

```
Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  count100 ~ normal(xb_count100,{sigma2})
Priors:
  {count100:temp i.precip _cons} ~ normal(0,10000)                        (1)
                         {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_count100.
Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        731
                                                 Acceptance rate  =      .1968
                                                 Efficiency:  min =     .02171
                                                              avg =     .03907
Log marginal-likelihood = -3009.1647                          max =     .05898
```

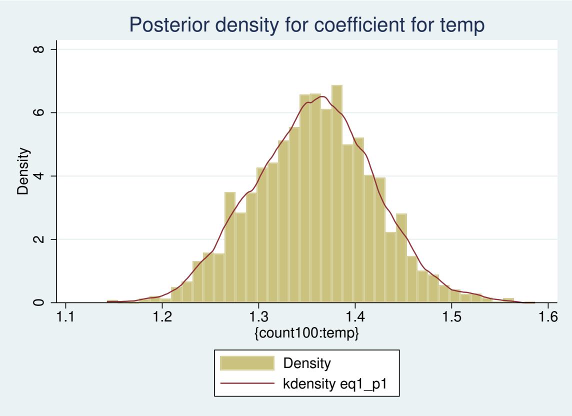
	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
count100						
temp	1.357601	.0627935	.002586	1.358048	1.237237	1.480288
precip						
Mist	-5.637735	1.121772	.058148	-5.604706	-7.831796	-3.528415
Light rain/snow	-25.61719	3.111537	.156311	-25.54275	-31.87835	-19.8676
_cons	26.90482	1.178267	.060584	26.87755	24.60462	29.20782
sigma2	204.306	10.741	.72906	203.9492	184.8968	226.2071

```
. estimates store bmh_bikes // store estimates
. use bikespost, clear
. describe
Contains data from bikespost.dta
 Observations:         2,737
    Variables:            11                  31 Aug 2022 16:21
-----------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-----------------------------------------------------------------------
_chain          int     %8.0g                 Chain identifier
_index          int     %8.0g                 Iteration number
_loglikelihood  double  %10.0g                Log likelihood
_logposterior   double  %10.0g                Log posterior
eq1_p1          double  %10.0g                {count100:temp}
eq1_p2          double  %10.0g                {count100:1b.precip}
eq1_p3          double  %10.0g                {count100:2.precip}
eq1_p4          double  %10.0g                {count100:3.precip}
eq1_p5          double  %10.0g                {count100:_cons}
eq0_p1          double  %10.0g                {sigma2}
_frequency      int     %8.0g                 Frequency weight
-----------------------------------------------------------------------
```

Variables containing the simulated values are named `eqj_pi`, where `j` is the equation and `i` distinguishes the parameters. `_freq` contains the frequency.

Those simulated values can be used to plot the posterior density, as we did with `bayesgraph kdensity`.

```
. histogram eq1_p1 [fw=_freq], addplot(kdensity eq1_p1 [fw=_freq]) ///
>   title("Posterior density for coefficient for temp")
(bin=40, start=52.637679, width=.58229688)
```



Out of sample predictions: `bayespredict`

Now, let's assume that the weather forecast for tomorrow is no precipitacions (`precip=1`) and a temperature of 20°Celsius (`temp=20`); given this weather, how do we predict the number of bikes to be rented?

```
use bikes, clear
estimates restore bmh_bikes

local N1 = _N + 1
set obs `N1´
replace precip = 1 in `N1´
replace temp = 20 in `N1´
global N1 = `N1´

* _ysim represents the outcome
bayespredict {_ysim} if _n == `N1´, rseed(1357) saving(ypred, replace)
```

We can use `bayesstats summary` to display statistics for the prediction.

```
. bayesstats summary _ysim_$N1 using ypred

Posterior summary statistics                      MCMC sample size =     10,000
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
_ysim1_732	54.20513	14.37852	.143785	54.23792	26.1864	82.50352

There is 95% probability of renting between 2618 and 8250 bikes.

Final remarks:

- Bayesian analysis can be used to answer questions about unknown parameters in terms of probability statements, using prior information on such probability.
- Stata provides a suite of commands for Bayesian estimation, diagnostics, visualization and prediction. Today we have just described a few of them. Please see the [Bayes] manual for a complete reference.