Bayesian Analysis using Stata

Bill Rising

StataCorp LP

2016 Italian Stata Users Group Meeting
Rome
17 November 2016

Goals

- Learn a little about Bayesian analysis
- Learn the core of how Bayesian analysis are implemented in Stata 14

Uncertainty as Probability

- In the frequentist world, probabilities are long-run proportions of repeated identical experiments
  - In some ways, this means we never know any probabilities of any events
- In the Bayesian world, probabilities are an expression of uncertainty
  - The advantage of the Bayesian viewpoint is that it allows talking about probabilities for events which cannot be repeated
    - What is the chance of a tropical storm hitting Spain this year?
    - What is the chance that Italy takes the 2018 World Cup?
  - The disadvantage is that these probabilities become subjective

Bayesian Analysis

- Uncertainty about parameters is expressed via a prior distribution $p(\boldsymbol{\theta})$
  - The prior distribution is necessarily subjective
  - If there is little knowledge about possible values, vague or non-informative priors get used
- The dataset $\mathbf{y}$ is used to update these priors into posterior distributions via Bayes rule

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})}$$

  - $p(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood
  - $p(\mathbf{y})$ is the marginal density of the data

$$p(\mathbf{y}) = \int_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

  - This last integral has been the bugaboo

Advantages and Disadvantages of Bayesian Analysis

- Advantages
  - Theoretically should allow updating knowledge with past experience
  - Can speak directly about probabilities instead of applying long-run proportions to a single event
    - Think of confidence intervals: have long-run chance of catching the parameter value, but know nothing about the current estimate
  - Can choose among multiple competing hypotheses instead of just two
- Disadvantages
  - Could be worried about subjectivity

Why Has Bayesian Analysis Become More Popular

- Computational speed allows rapid and good approximations of the marginal density of the data
  - Before computational horsepower could be used, only a small set of models could be estimated
- All the magic comes from Markov Chain Monte Carlo (MCMC) methods
  - These sample points from the not-fully-specified density in such a way that if left running forever, the density of simulation points would equal the target density

Implementation in Stata 14

- In Stata 14, the estimation portion of Baysian analysis is implented by the `bayesmh` command
  - `mh` for Metropolis-Hastings
- We will see how this works, both via point-and-click and syntactically
- We will look at some diagnostics and other post-estimation tools

A Simple Story

- We'll work with a very simple dataset measuring counts
- Here is our simulated story:
  - We've heard that the Rome airport (Fiumicino - Leonardo da Vinci) has large numbers of gate changes
  - We've the number of gate changes for a small sample of flights
  - We would like to get some concept of the rate of gate changes
- We'll simulate a dataset as zero-inflated Poisson with an overall arrival rate of 1.2. . .
  - . . . but in reality some flights never get gate changes

    ```
    . do rome
    ```
  - As the name suggests, this is mixture model of 0's and Poisson, called the zero-inflated poisson
- Let's see the mean count for this simulation

  ```
  . sum y
  ```

Starting a Bayesian Analysis: the Prior

- We would now like to do a Bayesian investigation of the rate of gate changes
  - Suppose we are also interested whether the rate of changes is over one per flight
  - We will start with the (erroneous) assumption that the gate changes are purely Poisson
- To start out, we need to specify a prior distribution
- How would this possibly be done?
  - We could try to use a vague prior which has very little information in it
  - We could try to elicit the opinions of experts
- We'll start with a vague prior

The Flat Vague Prior

- Vague priors are only vaguely defined: they ought to cover all remotely plausible values without favoring any values
- We will choose a flat prior, meaning that all possible values of our parameter have the same "probability"
  - The word probability is in quotes, because we want a probability density proportional to 1 over an infinite range
  - Because such a density is not really a density, it is called an improper prior
  - Because this means that we need a probability density proportional to 1 over the interval 0 to ∞, this is an improper prior
    - Clearly, like continuous-time white noise, this is impossible but helpful
  - Improper priors should typically be avoided, but this will help the exposition here

Specifying our Model: the Interface

- We will start by using the point-and-click interface
- There are two ways to access this
  - Either select **Statistics** > **Bayesian analysis** > **Estimation**
  - Or type `db bayesmh` in the command window
- We will choose what we would like to do now, and then come back to the full range of possible models

Choosing the Likelihood Model—Poisson Distribution

- We can start simple by simply modeling a Poisson distribution
- Click on the dropdown under *Syntax*, and select *Univariate distributions*
- Under *Dependent variable*, choose `y`
- Under ***Distribution***, choose *Poisson distribution*
- Click on the **Create** button
- Type `lambda` for the *Parameter name*, and click on **OK**

Specifying the Prior Distribution

- Click on the **Create** button near the ***Priors of model parameters***
- Choose *{lambda}* from the picklist under ***Parameters specification***
- Click on *Flat prior (with a density of 1)*
- Click on the **OK** button
- Click on the **Submit** button

Making Our Computations Reproducible

- We should set a random seed for this MCMC
  - This will make sure that we can reproduce our result in the future
- Click on the ***Simulation*** tab
- We'll put 7434 as the random seed
  - This is an arbitrary non-negative integer

Preparing for Later Comparisons

- We would like to use an `estimates store` command, so we need to save the simulations as a dataset
  - This is because the posterior distribution is needed for computations
- Click on the ***Reporting*** tab
- Check the *Save simulation results as a dataset* checkbox
- Give the name `poisson` for the file
- Check the *Overwrite file if it already exists* checkbox

Computing the Posterior

- We are done specifying this simple model, so click the **Submit** button
- The command gets issued

```
. bayesmh y, likelihood(dpoisson({lambda})) ///
      prior({lambda}, flat) rseed(7434) ///
      saving(poisson, replace)
```

- Stata races through the MCMC simulation to estimate the posterior distribution
- Stata reports the results; we will store them

```
. estimates store poisson
```

General Notes about the Output

- At the top, you see `Burn In ...` followed by `Simulation ...` as notifications
  - These show progress in very computationally intensive models
- We see the two elements we need to specify for any Bayesian analysis: the Likelihood model and the Prior distribution
- There is information about how the MCMC sampling was done
- There is information about summary statistics of the posterior distribution
  - Recall that we are not specifically trying to estimate mean values; we are finding a posterior distrbution for the rate

Output Specifics: MCMC

- By default, Stata uses a burn-in of 2,500 iterations
  - This is used to tune the adaptive model and to give time for the simulation to reach the main part of the posterior distribution
- By default, Stata runs the MCMC chain for 10,000 iterations
- The acceptance rate is the rate that new picks from the distribution are accepted
- The efficiency is relative to independent samples from the posterior distribution

Output Specifics: Regression Table

- The mean of our posterior distribution for the arrival rate is 1.183
- The standard deviation of the posterior distribution is 0.101
- The MCSE of 0.0022 is the standard error of estimation of the mean due to our using MCMC to find the posterior distribution
  - How much the posterior mean would vary from run to run if we used different random seeds
- The median is the median of the posterior distribution
- The probability that the arrival rate is between 0.996 and and 1.391 is 95%
  - Note this is not a trapping probability for unknown future samples

Starting with Postestimation

- We can see what postestimation commands are available by typing

```
. db postest
```

- Now click on the disclosure control next to *Bayesian analysis*
- This shows a list of things which can be done

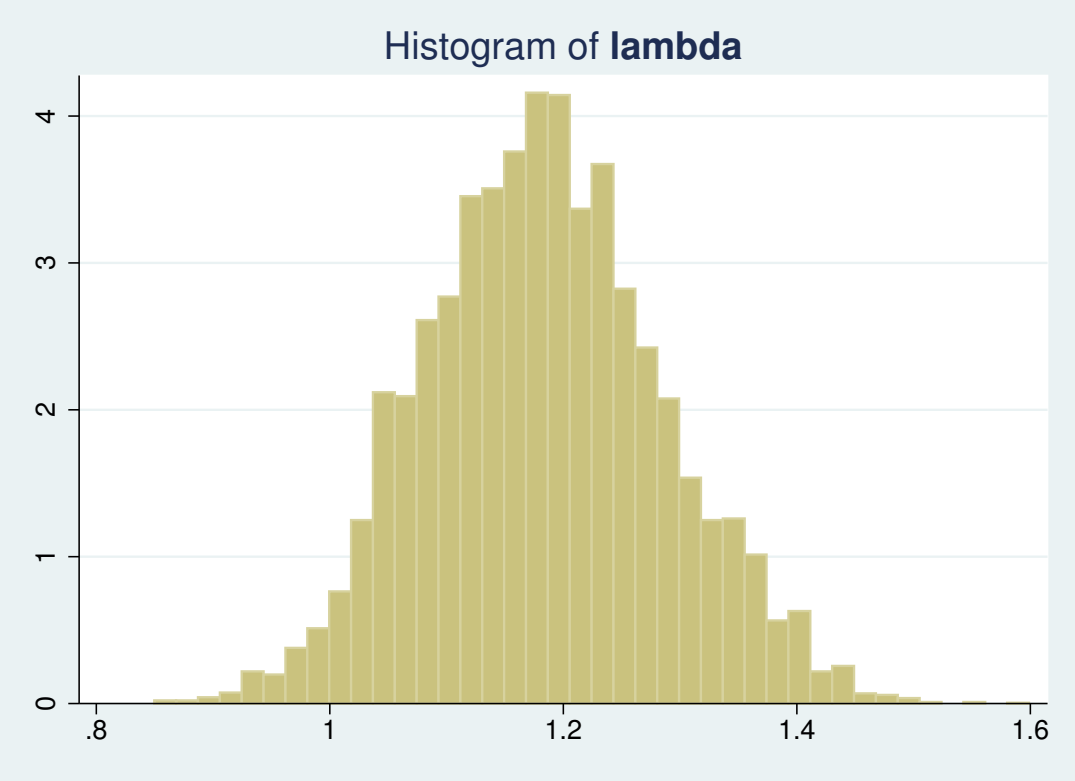
Investigating the Posterior

- We can draw a picture of the posterior distribution in a couple of ways
- Double click on the *Graphical summaries and convergence diagnostics* item
- To make a histogram, select the *Histograms* graph type
- To make life simple select the *Graphs for all model parameters* radio button
- Click on the **Submit** button

Histogram of the Posterior

- Here is the histogram version of the posterior distribution for the log of the rates

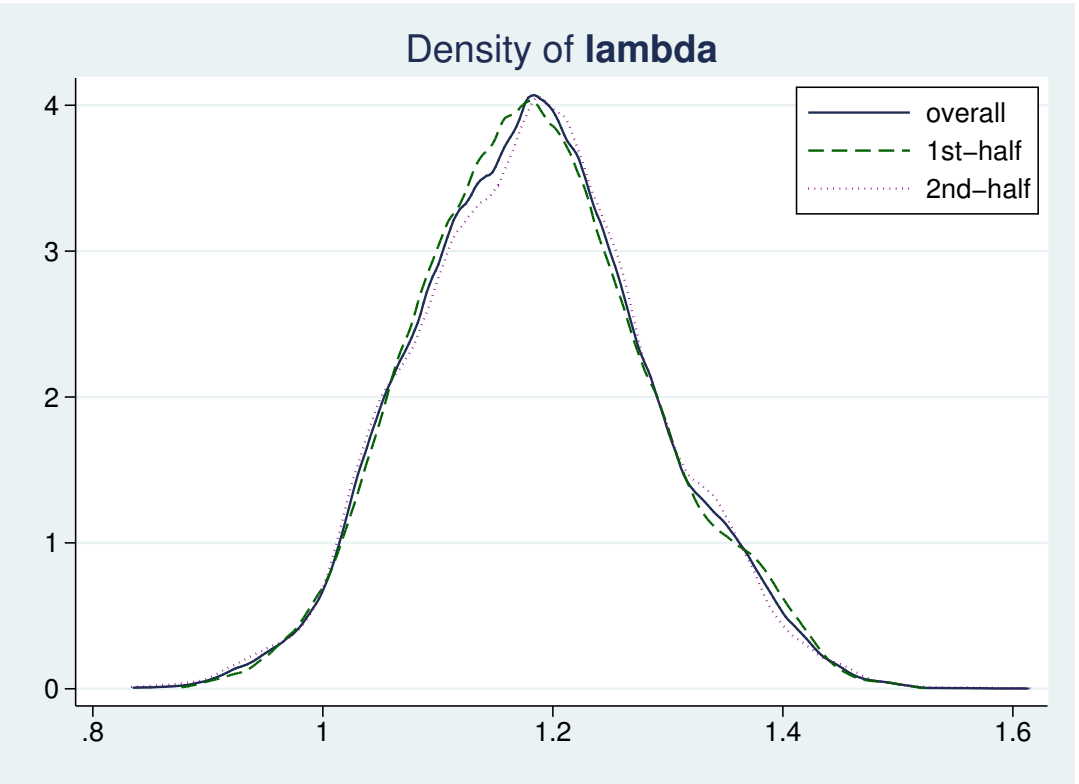
```
. bayesgraph histogram _all
```



Density Plot of the Posterior

- To get a density plot, select the *Density plots* graph type
- Click on the **Submit** button

```
. bayesgraph kdensity _all
```



Finding the Probability the Rate is Larger than 1

- Navigate back to the *Postestimation Selector* dialog box
- Double-click on the *Interval hypothesis testing* menu item
- Choose *{lambda}* parameter from the *Test model parameter* list
- Enter 1 as the *Lower* bound and leave . as the *Upper* bound
- Click the **Submit** button

```
. bayestest interval ({lambda}, lower(1))
```

- We can read off the probability as 0.973
  - This is a true probability
  - It is a subjective probability based on our flat prior

How MCMC Can Break

- There are multiple ways that MCMC can give bad answers
  - It can mix poorly, meaning either that
    - New candidate points for the simulation get rejected too often
    - The jumps are too small to cover the distribution
  - It can have bad initial values
    - These should be irrelevant because of the long burn-in sequence
    - But... if there is poor mixing this might not be the case
    - This leads to what is called 'drift'

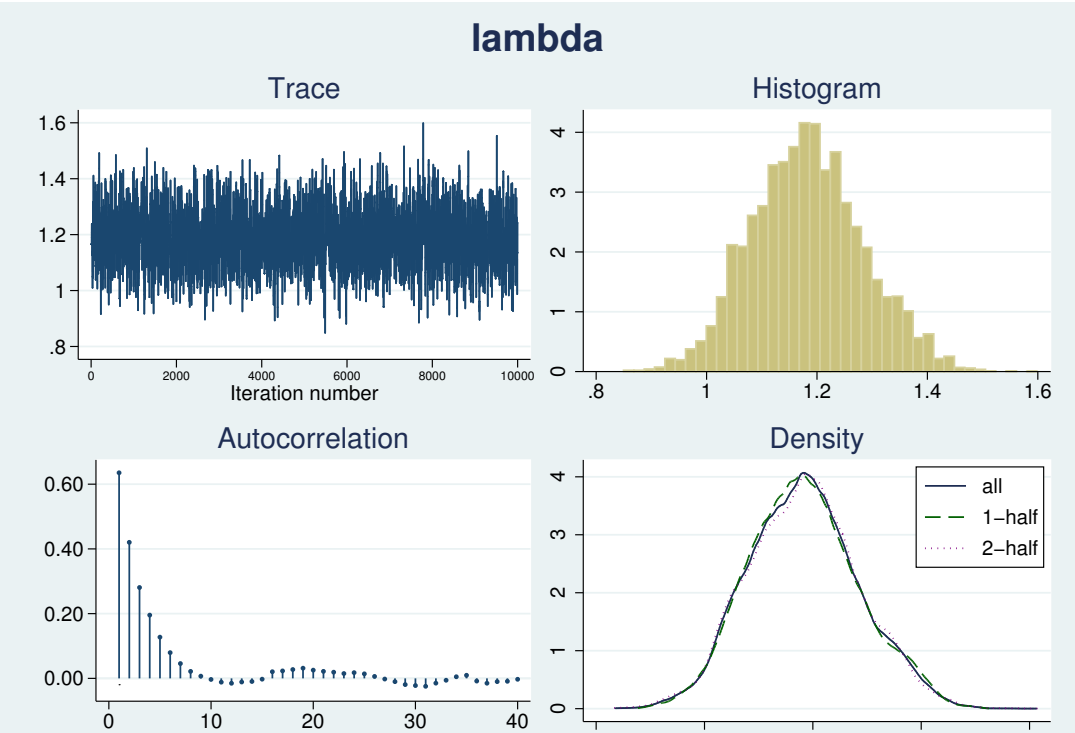
Useful Visual Diagnostics

- Go back to the *Postestimation Selector* dialog box
- Select the *Graphical summaries and convergence diagnostics* item
- Click on the **Launch** button

MCMC Diagnostics

- There is a simple tool for looking at the standard diagnostics all at once
- Select *Multiple diagnostics in compact form* in the *bayesgraph* dialog, and press **Submit**

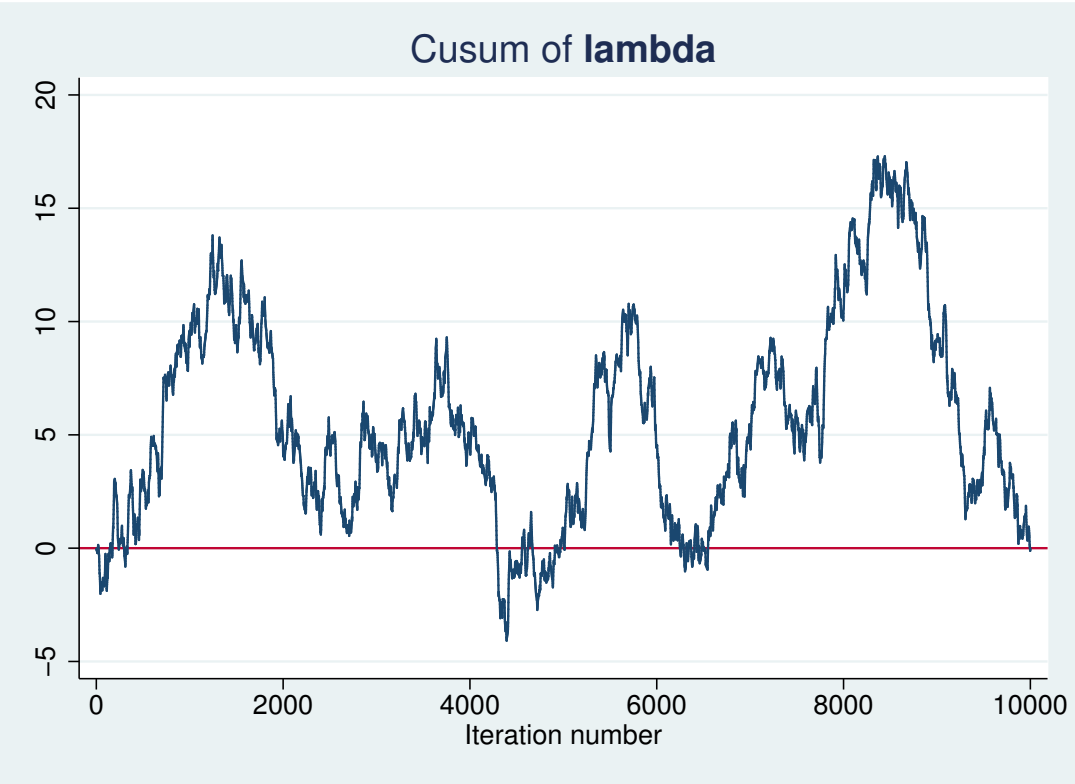
```
. bayesgraph diagnostics _all
```



Looking for Drift

- The cusum (short for cumulative sum) plot is used to look for small step size and drift
- Select *Cumulative sum plots* and press **Submit**

```
. bayesgraph cusum _all
```



Simple Diagnostic Conclusion

- Everything looks fine because there is no sign of bad mixing or drift
- We, as the dataset gods, know that we should be investigating other models
    - This will happen later

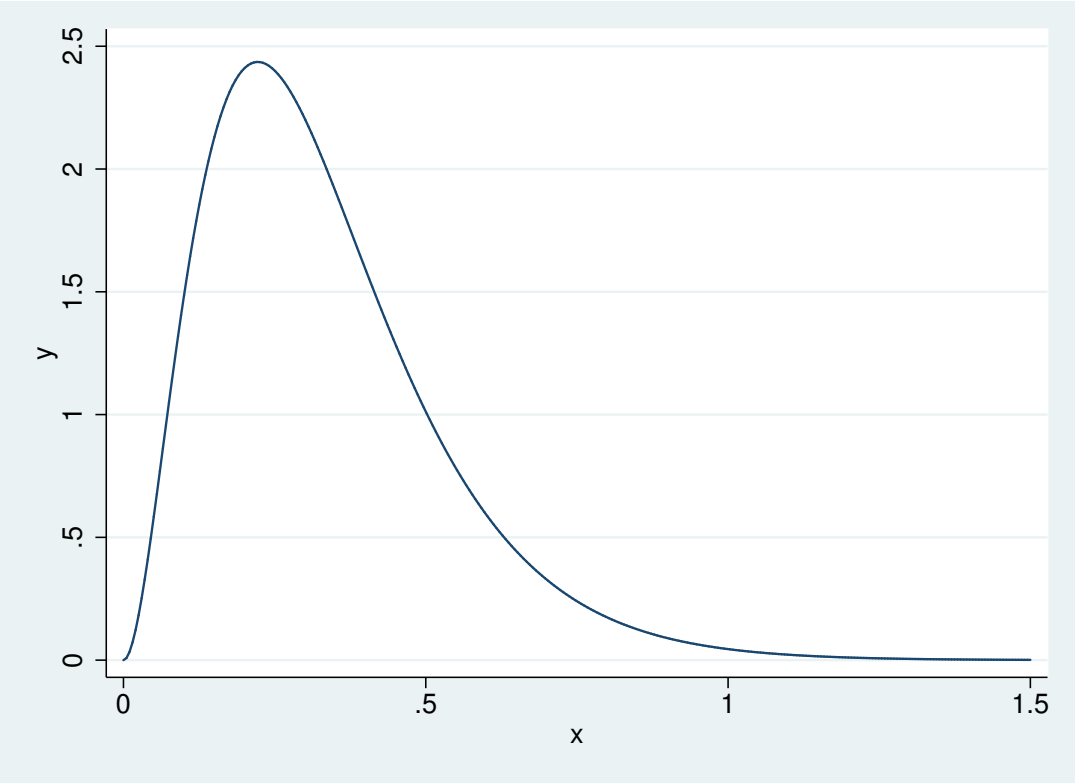
Playing with Different Priors

- Suppose we talk to people from other cities about their airports
- They all agree that gate changes should happen about 1 of every 3 flights, with little chance of averaging more than 1 gate change per flight
    - Thus, they are completely incorrect about Rome
- Based on this, a good prior would be a Gamma(3, 1/9)

Aside: Graph of the Prior

- Here is a graph of the Gamma(3, 1/9) distribution

```
. twoway function y = gammaden(3,1/9,0,x), range(0 1.5)
```



Specifying a New Prior

- Type `db bayesmh` to get our dialog box back
- Select the *Prior 1* prior
- Click on the **Edit** button
- Choose *Gamma distribution*
- Enter 3 as the *Shape* and 1/9 as the *Scale*
- Click on the `OK` button to dismiss the subdialog

Changing the Seed and Saving the Data

- Go to the ***Simulation*** tab
- Change the random seed to some other number, say 9983
- Click on the ***Reporting*** tab and save the data as `poisgamma`
- Click on the `Submit` button to run the analysis

```
. bayesmh y, likelihood(dpoisson({lambda})) ///
      prior({lambda}, gamma(3,1/9)) rseed(9983) ///
      saving(poisgamma, replace)
```

- Let's store the model for later

```
. est store poisgamma
```

What Happened?

- We can see that the mean of the posterior distribution is smaller
  - We should, however, be encouraged that the mean is only somewhat smaller despite the very-different prior
- We can compute our probability that the rate truly is larger than 1

```
. bayestest interval ({lambda}, lower(1))
```

  - It has been reduced to 0.902
  - This is not necessarily bad, considering how incorrect the prior was

Generalizing to Poisson Regression

- The above example is a bit unrealistic because it does not allow covariates
  - When would you really model simple counts?
- Let's do something less restrictive by modeling this as Poisson regression with no predictors
- Go back to the *bayesmh* dialog
- Click on the **Reset** button to empty the dialog box

Choosing the Likelihood Model–Part 2

- Now we would like a univariate linear model
- Clicking the drop-down menu for the *Dependent variable* and choose y
- We have no independent variables
- Choose *Poisson regression* as the ***Likelihood model***
- We can leave the *Exposure variable* blank

Specifying the Prior

- Click on the **Create...** button for the ***Priors of model parameters***
- From the *Parameters specification* dropdown, choose `{y:_cons}`
  - This is because we are modeling only the constant term without any covariates
- We will choose the *Flat prior* item
- Click **OK** to dismiss the subdialog

Making Our Computations Reproducible

- We should set a random seed for this MCMC
- Click on the ***Simulation*** tab
- We'll put 7434 as the random seed, again
- Go to the ***Reporting*** tab and save the simulation data as `poisreg`

Computing the Posterior

- We are already done specifying this simple model, so click the **Submit** button
- The command gets issued

```
. bayesmh y, likelihood(poisson) ///
      prior({y:_cons}, flat) rseed(7434) ///
      saving(poisreg, replace)
```

- Again, Stata runs through the MCMC simulations to find the posterior distribution
- Notice, however, that the parameter reported is the natural logarithm of the rate of gate changes
- It would be nice if we could report this in natural units (rate) rather than model units (log rate)
- Let's store this model, also

```
. est store poisreg
```

Changing to Natural Parameters

- Double-click on the *Summary statistics for model parameters and their functions*
- Select the *Summary statistics for model parameters and their functions* radio button
- Click on the **Create...** button
- Select the *Summary statistics for functions of model parameters, log likelihood and log posterior* radio button
- Click on the **Create...** button
- Type `exp(`, double-click on *{y:_cons}*, type `)`, and click on the **OK** button
- Click on the **OK** button in the parent dialog
- Click on the **Submit** button

The bayesstat summary Results

- The results for expr1 are now directly comparable with the results from the preceding example
  - Notice that simply exponentiating the results from the original output works only for the median of the posterior distribution
    - This is because exponentiating a mean is not the mean of the exponentiated values

Aside: When is Flat really Flat?

- The mean of the posterior distribution here is slightly smaller than that for the Poisson model
- For the Poisson model, "flat" meant that all values of lambda have the same prior probability
- For Poisson regression the "flat" prior for this model is that all values of log-lambda have the same prior probability
    - This is equivalent to the prior distribution in the Poisson model being proportional to $1/x$ for $0 < x < \infty$
- Here this makes very little difference; for very small samples it could be more important
- Lesson: the problem with flat priors is that they depend on parameterization
- Priors which avoid this are called Jeffreys priors

Specifying Our Own Likelihood

- What if we wanted a likelihood which is not one of the 10 built-in likelihoods?
- We can specify this by using the `likelihood()` option with the `llf()` suboption
- We just need an example to show this...

Beyond Poisson

- For a Poisson distribution, the mean and the variance are the same
- Suppose we take a closer look at the counts in our sample

```
. summarize y
```

- We can see that the sample mean of 1.175 is a ways away from the sample variance of 1.843
- Suppose we now try to use a zero-inflated Poisson model
- Zero-inflated Poisson distributions are not a part of Stata's suite, so we need to do some math

Writing Our New Likelihood Model

- The zero-inflated Poisson model assumes a mixture of
  - Always being zero (with probability π), and
  - A Poisson distribution with rate λ, with probability $1 - \pi$
- From this, the probability distribution becomes

$$\Pr y = 0 = \pi + (1 - \pi)\mathrm{e}^{-\lambda}$$

$$\Pr y = k = (1 - \pi)\frac{\lambda^k \mathrm{e}^{-\lambda}}{k!}; \qquad y = 1, 2 \ldots$$

- This means that there are now two parameters
  - π, which is between 0 and 1
  - λ, which is greater than zero

Substitutable Expressions

- The way we tell Stata to use the log-likelihood function is by using a *substitutable expression*
- We just need to replace
  - Symbols with the variables that represent them
  - Coefficient names to replace parameters
- Stata needs the log-likelihood, so we will have to take the logs of the above distribution:

```
cond(y==0, ln({pi}+(1-{pi})*exp(-{lambda})), ///
        ln(1-{pi}) + y*ln({lambda}) - lngamma(y+1) ///
        - {lambda})))
```

Working from Do-files

- Now the commands are becoming complicated enough that typing or clicking as we go will be unhelpful
- Let's open up a project file for this talk

```
. projman bayes
```

Finally: Analyzing the Zero-Inflated Poisson

- We can run our analysis with this do-file

```
. do zipois
```

  - The `saving()` option has been added because we will need it if we would like to compare this model to another later
  - We stored the model for later comparisons

Zero-Inflated Poisson Notes

- Notice the `note: invalid initial state` warning under `Burn in ...`:
  - This happened here because Stata started λ at 0, which is not a valid rate
  - This should only worry us if the efficiencies are low or if the chain did not converge
- Just as before, we can look at the diagnostics (not shown)
  - Now there are two sets of plots because there are two parameters
- Here is the probability that the rate of gate changes is over 1

```
. bayestest interval ((1-{pi})*{lambda}, lower(1))
```

Extending the Chain

- If we would like to get an effective sample size which is close to what we had for the poisson model, we need to extend the chain
- The `mcmcsize(25000)` option does this

```
. do zipois2
```

Comparing Competing Models

- We would like to see whether we should be using a Poisson model or a zero-inflated Poisson model
- This is done using the `bayestest model` command
- Being Bayesians, we assign prior probabilities to each of the models, and then compute their posterior probabilities given our data
- We have no reason to think one model is better than the other so we'll use the default of equally likely

```
. bayestest model poisson zipois2
```

- We now think that there is a 92.9% chance that the zero-inflated Poisson is the correct model
    - Of the two presented

Aside: Bayesian Hypothesis Testing

- One wonderful part of the Bayesian world is that more than two models may be compared
- One must take care that hypotheses are plausible
  - No point values for continuous variables, for example, unless they are 0 values for something that might not exist
- Sometimes it makes sense to have prior distributions which are not evenly distributed
  - There can be a decision-theoretic reason for this, for example different costs associated with falsely conclusions
- This is far more flexible than the typical us-versus-them hypothesis testing

Information Criteria

- We can also compare models using the deviance information criterion (DIC) and Bayes factors

```
. bayesstats ic poisson zipois2
```

- The smaller DIC for the `zipois2` model says that it should do a better job producing a similar dataset
- The log(BF) column gives the log of odds that the `zipois2` model is true
  - Here: `ln(0.0707/0.9293)`
- The Bayes factor will always give the same subjective result as assuming equal prior probabilities for models

Linear Regression

- All we've been doing is looking at a dataset of counts

```
. save rome_plus, replace
```

- Now let's try playing with linear regressions
- Open up the `autometric` dataset

```
. use autometric
```

  - Made for all countries except the US, Liberia, and Myanmar

Modeling Energy Usage

- We'd like to measure energy usage of these cars
- Perhaps: regressing `lp100km` on `weight`, `displacement` and `foreign`
- Let's go back to the dialog box for teaching purposes
  - Reset the dialog box by clicking the big **R** button

Filling in the Dialog Box

- This will take a little effort, but specify
  - `{var}` as the variance for the likelihood
  - Normals with large variances for the coefficients
  - Jeffries prior for the prior of `{var}`
  - A random seed of 142857
- Click on `OK` to submit and close

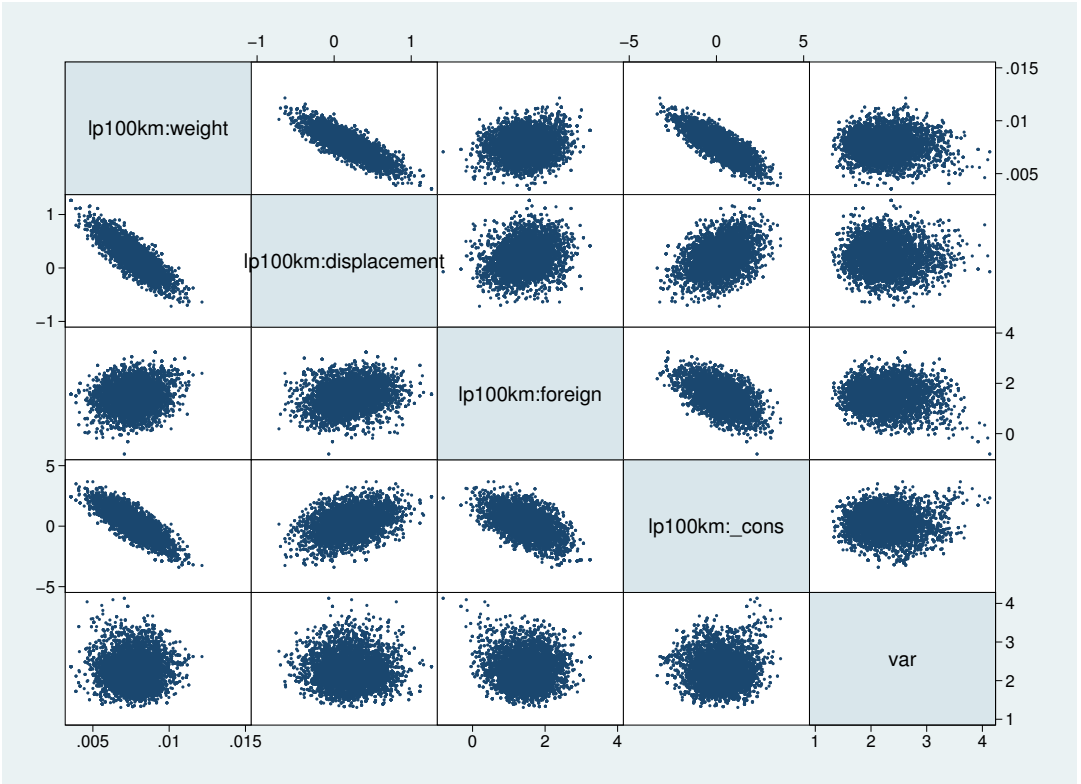
```
. do reg
```

- The model converges, but not at all efficiently

Looking at the Problem

- Draw a graph matrix to see the problems

```
. bayesgraph matrix _all
```



Partial Fix Number 1

- If we mean center the weight and the displacement, we'll get rid of some of the correlation between their simulated values and those of the intercept

```
. sum weight displacement
```

- While we're at it, let's make weight no so big

```
. gen wt1300 = (weight-1300)/1000
. gen displacement3 = displacement - 3
```

- Now let's see what happened

```
. do regcent
```

Partial Fix Number 2

- We've chosen very special prior distributions for our model
  - Normal priors for a normal regression are semi conjugate
  - This means that they produce normal posterior distributions
    - This means we know the posterior distrobution explicity
- So... we can use Gibbs sampling here
  - This is a special case of Metropolis-Hastings which exploits knowledge of the closed form
- As a side effect, we will estimate each of the predictors separately
  - The default is to estimate them all at once

Result of Gibbs Sampling

- Here is our Gibbs sampler

```
. do reggibbs
```

- This has helped a bunch with everything except the correlated predictors
- So: collinearity is a problem here, too!
- Our only solution is to run the chain much longer

```
. do reggibbs2
```

What We Have Seen

- Use of part of the GUI for Bayesian analysis in Stata
- Specification of a non-standard likelihood
- Specification of priors
- Basic Bayesian estimation
- Basic Baysian model comparison
- Gibbs samplers
- Centering

What We Have Not Seen

- Complex models
  - There are many many examples in the manuals
- Writing our own evaluators
  - If you have a likelihood function which is not the sum of the likelihoods for each of the observations, you can write a specially-formed evaluator program
    - This is similar in kind to the `ml` command

Conclusion

- We've just touched on what can be done
- I hope this has been somewhat informative