Title

> **bayesstats ppvalues** — Bayesian predictive p-values and other predictive summaries

Description

bayesstats ppvalues performs posterior predictive checking of the goodness of fit of a Bayesian model. It computes posterior predictive p-values (PPPs) for functions of replicated outcomes produced by bayespredict. PPPs measure the agreement between replicated and observed data. PPPs close to 0 or 1 indicate lack of model fit. The command also reports other summary statistics related to posterior predictive checking.

Quick start

Posterior predictive summaries of replicated outcomes

Bayesian predictions for all outcome variables after fitting a two-equation Bayesian model using bayesmh

        bayespredict {_ysim1} {_ysim2}, saving(prdata)

Posterior predictive summaries for the first replicated outcome

        bayesstats ppvalues {_ysim} using prdata

Posterior predictive summaries for the simulated residuals of the first outcome

        bayesstats ppvalues {_resid} using prdata

Posterior predictive summaries for both replicated outcomes

        bayesstats ppvalues {_ysim1} {_ysim2} using prdata

Posterior predictive summaries for the first observation of the second replicated outcome squared

        bayesstats ppvalues ({_ysim2[1]}^2) using prdata

Posterior predictive summaries for test statistics of replicated outcomes

Posterior predictive summaries for the maximum and minimum across observations of the second replicated outcome

        bayesstats ppvalues (y2max:@max({_ysim2})) (y2min:@min({_ysim2}))   ///
            using prdata

Posterior predictive summaries for the maximum and minimum across observations of the residuals for the first outcome variable

        bayesstats ppvalues (rmax:@max({_resid1})) (rmin:@min({_resid1}))   ///
            using prdata

Menu

Statistics > Bayesian analysis > Posterior predictive p-values

Syntax

Posterior predictive summaries for replicated outcomes, residuals, and more

> bayesstats <u>ppvalues</u> *yspec* [*yspec* ...] using *predfile* [, *options*]

Posterior predictive summaries for expressions of replicated outcomes, residuals, and more

> bayesstats <u>ppvalues</u> (*yexprspec*) [(*yexprspec*) ...] using *predfile* [, *options*]

Posterior predictive summaries for Mata functions of replicated outcomes, residuals, and more

> bayesstats <u>ppvalues</u> (*funcspec*) [(*funcspec*) ...] using *predfile* [, *options*]

Full syntax

> bayesstats <u>ppvalues</u> *predspec* [*predspec* ...] using *predfile* [, *options*]

predfile is the name of the dataset created by bayespredict that contains prediction results.

yspec is {*ysimspec* | *residspec* | *label*}.

ysimspec is {_ysim#} or {_ysim#[*numlist*]}, where {_ysim#} refers to all observations of the #th replicated outcome and {_ysim#[*numlist*]} refers to the selected observations, *numlist*, of the #th replicated outcome. {_ysim} is a synonym for {_ysim1}.

residspec is {_resid#} or {_resid#[*numlist*]}, where {_resid#} refers to all residuals of the #th replicated outcome and {_resid#[*numlist*]} refers to the selected residuals, *numlist*, of the #th replicated outcome. {_resid} is a synonym for {_resid1}.

label is the name of the function simulated using bayespredict.

With large datasets, specifications {_ysim#} and {_resid#} may use a lot of time and memory and should be avoided. See *Generating and saving simulated outcomes* in [BAYES] **bayespredict**.

yexprspec is [*exprlabel*:] *yexpr*, where *exprlabel* is a valid Stata name and *yexpr* is a scalar expression that may contain individual observations of simulated outcomes, {_ysim#[#]}; individual expected outcome values, {_mu#[#]}; individual simulated residuals, {_resid#[#]}; and other scalar predictions, {*label*}.

funcspec is [*label*:] @*func*(*arg1* [, *arg2*]), where *label* is a valid Stata name; *func* is an official or user-defined Mata function that operates on column vectors and returns a real scalar; and *arg1* and *arg2* are one of {_ysim[#]}, {_resid[#]}, or {_mu[#]}. *arg2* is primarily for use with user-defined Mata functions; see *Defining test statistics using Mata functions* in [BAYES] **bayespredict**.

predspec is one of *yspec*, (*yexprspec*), or (*funcspec*). See *Different ways of specifying predictions and their functions* in [BAYES] **Bayesian postestimation**.

options	Description
* ch-a-i-n-s(_all | *numlist*)	specify which chains to use for computation; default is chains(_all)
* sepchains	compute results separately for each chain
<u>nolegend</u>	suppress table legend

*Options chains() and sepchains are relevant only when option nchains() is used with bayesmh.

collect is allowed; see [U] **11.1.10 Prefix commands**.

Options

chains(_all | *numlist*) specifies which chains from the MCMC sample to use for computation. The default is chains(_all) or to use all simulated chains. Using multiple chains, provided the chains have converged, generally improves MCMC summary statistics. Option chains() is relevant only when option nchains() is specified with bayesmh.

sepchains specifies that the results be computed separately for each chain. The default is to compute results using all chains as determined by option chains(). Option sepchains is relevant only when option nchains() is specified with bayesmh.

nolegend suppresses the display of the table legend, which identifies the rows of the table with the expressions they represent.

Remarks and examples   **stata.com**

Remarks are presented under the following headings:

> *Posterior predictive checks*
> *PPPs*
> *Nonlinear effect of labor and capital on companies' output*

Posterior predictive checks

Posterior predictive checks, or model checks, are graphical and quantitative methods for comparing observed and replicated outcomes to assess goodness of fit of a Bayesian model. See Box (1980), Zellner (1975), West (1986), Gelman, Meng, and Stern (1996), and Gelman and Rubin (1992) for historical remarks and more in-depth discussions.

Replicated outcomes are outcome values that are simulated from the posterior predictive distribution using the observed covariate data; see *Overview of Bayesian predictions* and *Methods and formulas* in [BAYES] **bayespredict**. The distribution of replicated outcomes or its various summaries are compared with those of the observed outcomes. If they are similar, the Bayesian model is considered to fit the observed data well.

One of the graphical model checks uses quantile–quantile plots to compare observed and replicated residuals. These plots reveal misspecifications of the error distribution of a model. Histograms are commonly used to compare the distributions of the observed and replicated outcomes. More formally, the so-called PPPs, which we describe in the next section, are used to quantify the discrepancy between the summaries of the observed and replicated data.

PPPs

The notion of a PPP was introduced by Rubin (1984) as a Bayesian version of the classical p-value. The role of p-values in classical hypothesis testing is to quantify the discrepancy between the observed sample and population quantities. Test statistics, which are scalar functions of a sample, are commonly used as discrepancy measures. The p-value is defined as the probability to obtain a value of the test statistic as or more extreme than its observed value if the null hypothesis is true. This probability is computed with respect to the sampling distribution of the test statistic.

In a Bayesian setting, the discrepancy between the model and the observed data is measured by test quantities, which are scalar functions of a sample and model parameters. A test statistic is a special case of a test quantity that depends only on the sample. The distribution of a test quantity is defined with respect to the posterior predictive distribution of the replicated data and posterior distribution of model parameters. A PPP (or a Bayesian p-value or a Bayesian predictive p-value) is then defined as the probability that a test quantity for the replicated data could be as or more extreme than for the observed data. You can think of a PPP as a classical p-value averaged over the posterior distribution (Meng 1994). For more information about PPPs, see Tsui and Weerahandi (1989), Gelman, Meng, and Stern (1996), and Gelman et al. (2014), among others.

One of the advantages of PPPs over their classical counterparts is that they automatically handle nuisance parameters by averaging over the posterior distribution of all model parameters. In contrast, classical p-values are conditional on fixed model parameters, typically MLEs. Also, PPPs are not defined conditional on the null hypothesis being true and can be viewed simply as probabilities of model misfit. Values of PPPs close to zero or one indicate lack of fit. For a well-fitting model, the PPP should, ideally, be close to 0.5, although values between 0.05 and 0.95 are often considered acceptable in the literature (Gelman et al. 2014, 150; Congdon 2010, sec. 2.5.2).

One criticism of PPPs is that their distribution under the correct model specification is generally not uniform (for example, Bayarri and Berger [2000] and Robins, van der Vaart, and Ventura [2000]). The distribution tends to be more concentrated around 0.5 when the model is correct. Gelman (2013) argues that this property may be desirable in some cases and discusses the cases when it is not desirable. The author concludes that although it is difficult to provide general recommendations for how best to interpret PPPs, he suggests that they are still useful in practice to discover systematic discrepancies between the observed data and the fitted model.

When you check model fit, it is important to consider different test quantities that describe various aspects of the distribution of the replicated data. Certain distinctive aspects of the assumed model distribution such as symmetry and weight of the tails are commonly used as test quantities. For example, for assumed normal errors, it is appropriate to test the skewness and kurtosis of replicated residuals and compare them with the skewness and kurtosis of a normal distribution. When you use test quantities, Gelman (2013) suggests to use "caution in interpreting diagnostics that strongly depend on parameters or latent data". In addition to test quantities, you can use PPPs to compare individual observations, that is, compare the sample of replicated outcomes for a particular observation with the corresponding observed outcome value.

Nonlinear effect of labor and capital on companies' output

In this example, we show an application of PPPs to assess goodness of fit of a Bayesian model. We adapt an example described in Koop (2003, sec. 5.9) about the effect of labor and capital on companies' production. The dataset, `coutput.dta`, includes data for 123 companies with variables `output`, `labor`, and `capital`. The variables are scaled.

```
. use https://www.stata-press.com/data/r18/coutput
(Company output data)

. describe

Contains data from https://www.stata-press.com/data/r18/coutput.dta
 Observations:             123                  Company output data
    Variables:               3                  22 Feb 2023 13:24
                                                (_dta has notes)
─────────────────────────────────────────────────────────────────────────
Variable       Storage   Display    Value
    name          type    format    label      Variable label
─────────────────────────────────────────────────────────────────────────
output          float    %9.0g                 Output
labor           float    %9.0g                 Labor
capital         float    %9.0g                 Capital
─────────────────────────────────────────────────────────────────────────

Sorted by:
```

Koop (2003) proposes the following nonlinear model for describing companies' output:

$$\texttt{output}_i = \alpha + (\beta_1 \texttt{labor}_i^\lambda + \beta_2 \texttt{capital}_i^\lambda)^{1/\lambda} + \epsilon_i, \ \epsilon_i \sim_{\text{i.i.d.}} N(0, \sigma^2)$$

A nonlinear model ($\lambda > 1$) is expected to provide a better fit for the data than the linear model ($\lambda = 1$). We explore this by using posterior predictive checks.

Without concrete prior knowledge about the parameters α, β_1, β_2, and λ, we specify weakly informative priors for them. We use the $N(0, 100)$ prior for the coefficients, which is noninformative because the variables are scaled to be in the $(0, 2)$ range. We apply exponential prior, $\exp(1)$, for λ because λ is a positive parameter with 1 being a highly probable value for it. Below is the full model specification using `bayesmh`:

```
. bayesmh output =
> ({alpha}+({beta1}*labor^{lambda}+{beta2}*capital^{lambda})^(1/{lambda})),
> likelihood(normal({sig2}))
> prior({alpha beta1 beta2}, normal(0,100))
> prior({lambda}, exp(1)) prior({sig2}, igamma(0.1,0.1))
> init({alpha beta1 beta2 lambda} 1)
> saving(coutput_mcmc) mcmcsize(5000) rseed(16)
Burn-in ...
Simulation ...

Model summary
─────────────────────────────────────────────────────────────────────────
Likelihood:
  output ~ normal(<expr1>,{sig2})
Priors:
                {sig2} ~ igamma(0.1,0.1)
  {alpha beta1 beta2} ~ normal(0,100)
              {lambda} ~ exponential(1)
Expression:
  expr1 : {alpha}+({beta1}*labor^{lambda}+{beta2}*capital^{lambda})^
> (1/{lambda})
─────────────────────────────────────────────────────────────────────────
```

```
Bayesian normal regression                      MCMC iterations  =        7,500
Random-walk Metropolis-Hastings sampling        Burn-in          =        2,500
                                                MCMC sample size =        5,000
                                                Number of obs    =          123
                                                Acceptance rate  =        .2176
                                                Efficiency:  min =      .02226
                                                             avg =      .03045
Log marginal-likelihood =  6.9478788                         max =      .03524
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
alpha	1.028072	.0549225	.004364	1.028156	.9300813	1.137604
beta1	.6838483	.0990207	.007467	.6736414	.5037749	.8903975
beta2	.9578192	.140161	.013285	.9413369	.7197326	1.25467
lambda	1.270644	.2435384	.020624	1.255252	.8222015	1.78508
sig2	.0390367	.0052749	.000397	.038495	.0301147	.0503593

file **coutput_mcmc.dta** saved.

We generated an MCMC sample of size 5,000 with an average efficiency of about 3%. bayesmh estimated the posterior mean of λ, 1.3, to be larger than 1, which implies that labor and capital do have a nonlinear effect on companies' output.

Model assumptions can be assessed through residual analysis. We follow Koop (2003) and use PPPs to compare various aspects of the distribution of residuals simulated from the fitted model and observed residuals. By construction, the distribution of the simulated residuals is $N(0, \sigma^2)$.

▷ Example 1: PPPs for simple test statistics

One simple check is to compare the means and variances of the simulated residuals, $\mathbf{r}^{\text{sim}}$, with those of the observed residuals, $\mathbf{r}^{\text{obs}}$. Let

$$T_1(\mathbf{y}) = \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \text{ and } T_2(\mathbf{y}) = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2$$

denote the mean and variance test statistics. We want to compare $T_1(\mathbf{r}^{\text{sim}})$ with $T_1(\mathbf{r}^{\text{obs}})$ and $T_2(\mathbf{r}^{\text{sim}})$ with $T_2(\mathbf{r}^{\text{obs}})$.

We first use bayespredict to generate MCMC samples of means and variances of simulated and observed residuals.

```
. bayespredict (mean:@mean({_resid})) (var:@variance({_resid})),
> saving(coutput_pred) rseed(16)

Computing predictions ...

file coutput_pred.dta saved.
file coutput_pred.ster saved.
```

We used built-in Mata functions mean() and variance() to compute the means and variances; the Mata function specification is designated with @. We specified {_resid} as the argument to these functions to compute the means and variances of the simulated residuals. We labeled the resulting means as mean and variances as var; we can use these labels later within bayesstats ppvalues to refer to these prediction results. And we saved the simulated results in the prediction dataset, coutput_pred.dta. As we discussed in *Prediction dataset* of [BAYES] **bayespredict**, the generated prediction dataset includes, among other variables, the mean variable containing 5,000 means of simulated residuals, $\{T_1(\mathbf{r}^{\text{sim},1}), T_1(\mathbf{r}^{\text{sim},2}), \ldots, T_1(\mathbf{r}^{\text{sim},5000})\}$, where $\mathbf{r}^{\text{sim},t}$ is the column vector

containing 123 residuals simulated from the fitted model using the tth set of MCMC estimates of model parameters. (We saved our MCMC estimates of model parameters in `couput_mcmc.dta` with `bayesmh`.) Additionally, `bayespredict` generated 5,000 means of the observed residuals, $\{T_1(\mathbf{r}^{\text{obs},1}), T_1(\mathbf{r}^{\text{obs},2}), \ldots, T_1(\mathbf{r}^{\text{obs},5000})\}$, and stored them in the `_obs_mean` variable in the prediction dataset. Similarly, `bayespredict` generated variances of simulated and observed residuals and saved them in variables `var` and `_obs_var` in the prediction dataset. See [BAYES] **bayespredict** for details.

We can now access the simulated means and variances within `bayesstats ppvalues`. For example, we specify {mean} to compute PPPs for the mean test statistic. We also specify the prediction dataset, `coutput_pred.dta`, containing the simulated means in the required `using` specification.

```
. bayesstats ppvalues {mean} using coutput_pred
Posterior predictive summary    MCMC sample size =      5,000
```

T	Mean	Std. dev.	E(T_obs)	P(T>=T_obs)
mean	-.00007	.0177143	.0000147	.4978

```
Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
```

T and T_obs denote the test statistics computed using the replicated data and observed data, respectively. In our example, T is $T_1(\mathbf{r}^{\text{sim}})$ and T_obs is $T_1(\mathbf{r}^{\text{obs}})$. The posterior mean estimate, Mean, of $T_1(\mathbf{r}^{\text{sim}})$ from the MCMC sample of means of simulated residuals is -0.00007. The posterior mean estimate, E(T_obs), of $T_1(\mathbf{r}^{\text{obs}})$ from the MCMC sample of means of observed residuals is 0.0000147. Both are close to zero. The estimated PPP is about 0.5, which indicates a perfect agreement between the two means. This p-value represents the proportion of times the mean of simulated residuals was greater than or equal to the mean of the observed residuals in the MCMC sample.

The PPP can be also visualized using a histogram with the reference line at the observed mean, which is essentially 0 in our example.

```
. bayesgraph histogram {mean} using coutput_pred, color(%50) xline(0)
```


Histogram of **mean**
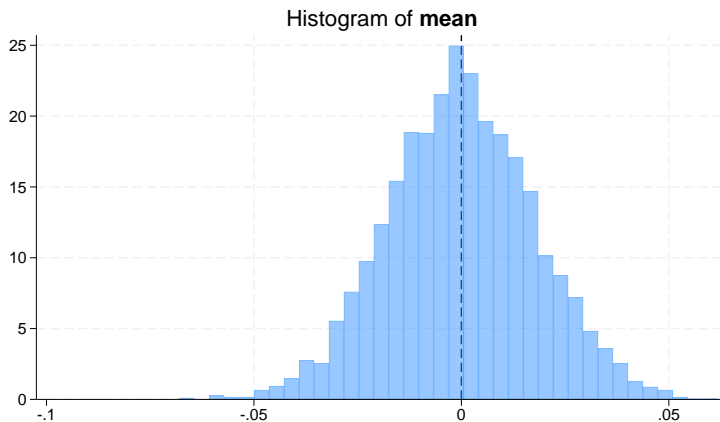
The PPP is the area of the histogram to the right of the observed value, which is about 50% in our example.

As with the mean, we can compare the variances.

```
. bayesstats ppvalues {var} using coutput_pred
Posterior predictive summary   MCMC sample size =      5,000
```

T	Mean	Std. dev.	E(T_obs)	P(T>=T_obs)
var	.038952	.0073444	.03694	.5762

```
Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
```

Posterior mean estimates of the variances of the simulated and observed residuals are similar and are close to the error variance {sig2} of 0.39, as estimated by bayesmh. The estimated PPP is 0.58 and again indicates very good agreement between the variances.

It is not surprising that the means and variances of simulated and observed residuals are in such good agreement. This tends to be true for many models in which the parameters directly model means and variances. In example 3, we explore other discrepancy measures such as skewness and kurtosis.

◁

▷ Example 2: Specifying Mata functions directly with bayesstats ppvalues

In example 1, we computed the mean and variance test statistics with bayespredict. Such specification is preferable with large datasets because it does not save a typically large sample of replicated outcomes. With moderate-sized datasets, you may save the replicated outcomes first and compute the functions within bayesstats ppvalues.

For example, here we simulate replicated outcomes using bayespredict. We replace the earlier coutput_pred.dta with new results.

```
. bayespredict {_ysim}, saving(coutput_pred, replace) rseed(16)
Computing predictions ...
file coutput_pred.dta saved.
file coutput_pred.ster saved.
```

In this case, the generated prediction dataset contains 5,000 MCMC replicates for each observation of our outcome output. That is, the dataset has 123 variables and 5,000 observations (and other auxiliary variables). We can now compute PPPs for any function of the replicated outcomes or their residuals.

We use the same specification of Mata functions with bayesstats ppvalues as we did with bayespredict in example 1.

```
. bayesstats ppvalues (mean:@mean({_resid})) (var:@variance({_resid}))
> using coutput_pred
Posterior predictive summary   MCMC sample size =      5,000
```

T	Mean	Std. dev.	E(T_obs)	P(T>=T_obs)
mean	-.00007	.0177143	.0000147	.4978
var	.038952	.0073444	.03694	.5762

```
Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
```

We obtain identical results to example 1. Notice that we can combine various specifications in one call to bayesstats ppvalues.

bayesstats ppvalues used the replicated outcomes from `coutput_pred.dta` to compute the simulated residuals, which resulted in an intermediate sample of 5,000 MCMC residuals for each of the 123 observations. It then produced yet another intermediate sample of 5,000 means of the residuals over 123 observations. Finally, it used the sample of 5,000 means to compute the posterior predictive summaries as displayed in the output table. The command performed the same computations for the variances, `var`.

Using `bayespredict` to save the entire sample of replicated outcomes, whenever feasible, is convenient because you can explore various discrepancy measures without having to predefine them. However, there are two other advantages of the earlier specification, in addition to speed and storage efficiency. When you compute functions using `bayespredict`, you can specify expressions of these functions with `bayesstats ppvalues` (or other Bayesian postestimation commands). Also, you can compute your own functions within Stata programs and specify them with `bayespredict`, whereas the use of Stata programs is not allowed within `bayesstats ppvalues` and other Bayesian postestimation commands. But you can define your own Mata functions and use them with `bayesstats ppvalues`, as we demonstrate in the next example.

◁

▷ Example 3: PPPs for user-defined test statistics

Continuing with example 2, we explore other discrepancy measures for the simulated and observed residuals. Given that we expect our residuals to be normally distributed when the model fits the data, we can explore their skewness and kurtosis.

Skewness and kurtosis are related to the third and fourth moments of a distribution. The skewness statistic measures the symmetry of a distribution about its mean. The kurtosis statistic measures the weight of the tails of a distribution. A normal distribution has skewness of 0 and kurtosis of 3.

There are no built-in Mata functions to compute these measures, so we need to define our own.

```
. mata:
                                    ──────── mata (type end to exit) ────
: real scalar skew(real colvector vresid) {
>         return (sqrt(length(vresid))*sum(vresid:^3)/(sum(vresid:^2)^1.5))
> }
: real scalar kurtosis(real colvector vresid) {
>         return (length(vresid)*sum(vresid:^4)/(sum(vresid:^2)^2) - 3)
> }
: end
```

Mata function `skew()` computes sample skewness, and `kurtosis()` computes sample kurtosis, but it subtracts 3 from the formula so that the kurtosis of a normally distributed sample is 0. Both functions accept a column vector of residuals as an argument and calculate and return the overall test statistic as a scalar.

We can now use these two functions to compute PPPs for skewness and kurtosis of residuals.

```
. bayesstats ppvalues (sy:@skew({_resid})) (ky:@kurtosis({_resid}))
> using coutput_pred

Posterior predictive summary    MCMC sample size =      5,000
```

T	Mean	Std. dev.	E(T_obs)	P(T>=T_obs)
sy	.0014651	.3420932	.1763123	.3464
ky	-.0368386	.423227	-.3171961	.7304

```
Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
```

The posterior mean estimates for the skewness and kurtosis of the observed residuals are not as close to zero as their counterparts simulated from the model. Nevertheless, according to the estimated PPPs of 0.35 for skewness and of 0.73 for kurtosis, the observed discrepancies can be explained by sampling variation. For instance, 35% of simulated skewnesses are greater than or equal to the observed skewnesses.

A PPP close to 0 or 1 indicates model misfit. Although there are no definitive recommendations, some literature suggests that PPPs less than 0.05 or larger than 0.95 be considered indicative of lack of fit (Gelman et al. 2014). However, it is important to consider PPPs in the context of your research question, such as whether the observed discrepancy is practically meaningful.

To visualize once again the PPP, we can plot the histogram of the simulated skewness with the reference line at the expected observed value of 0.18.

```
. bayesgraph histogram (sy:@skew({_resid})) using coutput_pred, xline(0.18)
> color(%50)
```
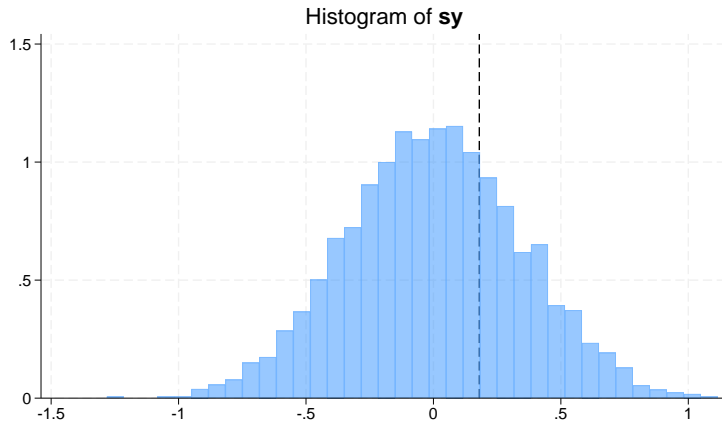

Histogram of **sy**

About 35% of the histogram area is on the right of the mean observed skewness of 0.18.

◁

In conclusion, our residual analysis revealed good agreement between the simulated and observed residuals with respect to several test statistics. Therefore, there do not appear to be any violations of the normality assumption for the error terms in the model.

We want to emphasize the importance of the choice of test statistics when assessing model fit. You should avoid using sufficient statistics such as sample mean and variance, which are usually well behaved because they are often directly modeled by parameters. Instead, you should focus on statistics that measure more specific distribution properties such as quantiles, skewness, kurtosis, maximum and minimum, and more.

Stored results

bayesstats ppvalues stores the following in r():

Scalars
    r(mcmcsize)   MCMC sample size used in the computation
    r(nchains)   number of chains used in the computation

Macros
    r(names)   names of model parameters and expressions
    r(expr_#)   #th expression
    r(exprnames)   expression labels
    r(chains)   chains used in the computation, if chains() is specified

Matrices
    r(summary)   matrix with predictive statistics for parameters in r(names)
    r(summary_chain#) matrix summary for chain #, if sepchains is specified

Methods and formulas

See *Methods and formulas* of [BAYES] **bayespredict** for general definitions and for formulas related to replicated outcomes, $\mathbf{y}^{\mathrm{rep}}$.

Let $T_q(\mathbf{y}, \boldsymbol{\theta})$ be a test quantity. The PPP, $q(T_q)$, is defined as the probability that $T_q(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta})$ is greater than or equal to the observed $T(\mathbf{y}^{\mathrm{obs}}, \boldsymbol{\theta})$ (Rubin 1984, Gelman et al. 2014). Specifically,

$$
\begin{aligned}
q(T_q) &= \Pr\left\{T_q(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta}) \geq T_q(\mathbf{y}^{\mathrm{obs}}, \boldsymbol{\theta})|\mathbf{y}^{\mathrm{obs}}, X^{\mathrm{obs}})\right\} \\
&= \int \int 1_{T_q(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta}) \geq T_q(\mathbf{y}^{\mathrm{obs}}, \boldsymbol{\theta})} p(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta}|\mathbf{y}^{\mathrm{obs}}, X^{\mathrm{obs}}) d\mathbf{y}^{\mathrm{rep}} d\boldsymbol{\theta} \\
&= \int \int 1_{T_q(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta}) \geq T_q(\mathbf{y}^{\mathrm{obs}}, \boldsymbol{\theta})} p(\mathbf{y}^{\mathrm{rep}}|\boldsymbol{\theta}, X^{\mathrm{obs}}) p(\boldsymbol{\theta}|\mathbf{y}^{\mathrm{obs}}, X^{\mathrm{obs}}) d\mathbf{y}^{\mathrm{rep}} d\boldsymbol{\theta}
\end{aligned}
$$

and $1_{(A)}$ is an indicator function of A being true.

In practice, the joint posterior distribution $p(\mathbf{y}^{\mathrm{rep}}, \boldsymbol{\theta}|\mathbf{y}^{\mathrm{obs}}, X^{\mathrm{obs}})$ is not available. Instead, we have a simulated sample $\left\{(\mathbf{y}^{\mathrm{rep},1}, \boldsymbol{\theta}^1), (\mathbf{y}^{\mathrm{rep},2}, \boldsymbol{\theta}^2), \ldots, (\mathbf{y}^{\mathrm{rep},\mathrm{T}}, \boldsymbol{\theta}^T)\right\}$, where T is the MCMC sample size. Then $q(T_q)$ is estimated as

$$
\widehat{q}(T_q) = \frac{1}{T} \sum_{t=1}^{T} 1_{T_q(\mathbf{y}^{\mathrm{rep},t}, \boldsymbol{\theta}^t) \geq T_q(\mathbf{y}^{\mathrm{obs}}, \boldsymbol{\theta}^t)}
$$

bayesstats ppvalues reports $\widehat{q}(T_q)$ in the output table and labels it P(T>=T_obs). bayesstats ppvalues also reports the average observed test quantity, E(T_obs),

$$\widehat{E}\left\{T_q(\mathbf{y}^{\text{obs}}, \boldsymbol{\theta})\right\} = \frac{1}{T}\sum_{t=1}^{T} T_q(\mathbf{y}^{\text{obs}}, \boldsymbol{\theta}^t)$$

and the sample mean and standard deviation of the sample of replicated test quantities, $\left\{T_q(\mathbf{y}^{\text{rep},1}, \boldsymbol{\theta}^1), T_q(\mathbf{y}^{\text{rep},2}, \boldsymbol{\theta}^2), \ldots, T_q(\mathbf{y}^{\text{rep,T}}, \boldsymbol{\theta}^T)\right\}$.

For a special case of test statistics, $T_q(\mathbf{y}, \boldsymbol{\theta}) = T_s(\mathbf{y})$, the above formulas simplify correspondingly.

> M. J. Bayarri (1956–2014) was born in Valencia, Spain. She received a bachelor's, master's, and doctorate degree in mathematics, all from the University of Valencia. She began as an assistant professor and then became a full professor at her alma mater.
>
> Bayarri won a Fulbright fellowship to attend Carnegie Mellon University in 1984, which marked the beginning of routine visits to the United States. She became a visiting professor at Purdue University, an adjunct professor at Duke University, and leader of the research program at the Statistical and Applied Mathematical Sciences Institute (SAMSI). She coauthored books on Bayesian statistics and biostatistics, and coauthored numerous research articles, including some award-winning papers. Her main areas of research included selection models, weighted distributions, and Bayesian analysis of queuing systems.
>
> Aside from her published contributions, she held multiple leadership roles. For example, Bayarri served as President of the International Society for Bayesian Analysis (ISBA) and as the principal investigator of Biostatnet, a network of biostatistical researchers. Her critical skills shined as Coordinating Editor of the *Journal of Statistical Planning and Inference* and as an award-winning food critic. In 1997, she was elected as a fellow of the American Statistical Association, and in 2008, she was elected as a fellow of the Institute of Mathematical Statistics.

References

Bayarri, M. J., and J. O. Berger. 2000. *P* values for composite null models. *Journal of the American Statistical Association* 95: 1127–1142. https://doi.org/10.2307/2669749.

Box, G. E. P. 1980. Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society, Series A* 143: 383–404. https://doi.org/10.2307/2982063.

Congdon, P. D. 2010. *Applied Bayesian Hierarchical Methods.* Boca Raton, FL: CRC Press.

Gelman, A. 2013. Two simple examples for understanding posterior p-values whose distributions are far from uniform. *Electronic Journal of Statistics* 7: 2595–2602. https://doi.org/10.1214/13-EJS854.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. 2014. *Bayesian Data Analysis.* 3rd ed. Boca Raton, FL: Chapman and Hall/CRC.

Gelman, A., X.-L. Meng, and H. S. Stern. 1996. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sanica* 6: 733–760.

Gelman, A., and D. B. Rubin. 1992. Inference from iterative simulation using multiple sequences. *Statistical Science* 7: 457–472. https://doi.org/10.1214/ss/1177011136.

Koop, G. 2003. *Bayesian Econometrics.* Chichester, UK: Wiley.

Meng, X.-L. 1994. Multiple-imputation inferences with uncongenial sources of input (with discussion). *Statistical Science* 9: 538–573. https://doi.org/10.1214/ss/1177010269.

Robins, J. M., A. W. van der Vaart, and V. Ventura. 2000. Asymptotic distribution of *p* values in composite null models. *Journal of the American Statistical Association* 95: 1143–1156. https://doi.org/10.1080/01621459.2000.10474310.

Rubin, D. B. 1984. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics* 12: 1151–1172. https://doi.org/10.1214/aos/1176346785.

Tsui, K.-W., and S. Weerahandi. 1989. Generalized *p*-values in significance testing of hypotheses in the presence of nuisance parameters. *Journal of the American Statistical Association* 84: 602–607. https://doi.org/10.2307/2289949.

West, M. 1986. Bayesian model monitoring. *Journal of the Royal Statistical Society, Series B* 48: 70–78. https://doi.org/10.1111/j.2517-6161.1986.tb01391.x.

Zellner, A. 1975. Bayesian analysis of regression error terms. *Journal of the American Statistical Association* 70: 138–144. https://doi.org/10.2307/2285389.

Also see

[BAYES] **bayesmh** — Bayesian models using Metropolis–Hastings algorithm

[BAYES] **bayespredict** — Bayesian predictions

[BAYES] **Bayesian postestimation** — Postestimation tools for bayesmh and the bayes prefix