Model Diagnostic

The BF class can compute model diagnostics for a given model.

Lets consider the following model for a linear regression:

$$Y_i \sim \text{Normal}(\alpha + \beta X_i, \sigma)$$

$$\alpha \sim \text{Normal}(0, 1)$$

$$\beta \sim \text{Normal}(0, 1)$$

$$\sigma \sim \text{Uniform}(0, 50)$$

```
from BayesForge import bf
import jax.numpy as jnp

# setup platform-----------------------------------------------
m = bf(platform="cpu")

# import data -------------------------------------------------
data_path = m.load.howell1(only_path=True)
m.data(data_path, sep=";")
m.df = m.df[m.df.age > 18]
m.scale(data=["weight"])


# define model ------------------------------------------------
def model(weight, height):
    a = m.dist.normal(178, 20, name="a")
    b = m.dist.log_normal(0, 1, name="b")
```

```
    s = m.dist.uniform(0, 50, name="s")
    m.dist.normal(a + b * weight, s, obs=height, shape=(weight.shape[0],))


# Run sampler ----------------------------------------------------
m.fit(model, num_samples=500, num_chains=4)
m.summary()
```

```
bf v 0.0.48 package loaded
```

```
E0526 16:18:16.948334 1066239 cuda_dnn.cc:523] Loaded runtime CuDNN library: 9.1.0 but source
E0526 16:18:16.950493 1066239 cuda_dnn.cc:523] Loaded runtime CuDNN library: 9.1.0 but source
```

```
jax.local_device_count 32
```

```
  0%|          | 0/1500 [00:00<?, ?it/s]
```

```
  0%|          | 0/1500 [00:00<?, ?it/s]
```

```
  0%|          | 0/1500 [00:00<?, ?it/s]
```

```
  0%|          | 0/1500 [00:00<?, ?it/s]
```

	mean	sd	hdi_5.5%	hdi_94.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
a	154.65	0.27	154.25	155.12	0.01	0.0	1873.35	1329.85	1.0
b	5.81	0.28	5.39	6.28	0.01	0.0	1849.54	1519.87	1.0
s	5.14	0.20	4.83	5.48	0.00	0.0	1982.99	1415.96	1.0

List of all available diagnostics

For additional documentation check the diagnostics API reference

Predictions from model based on specific data value

```
m.sample() # Predictions from model base on data in data_on_model
m.sample(data=dict(weight=jnp.array([0.4])), remove_obs=False)# Predictions from a given valu
```

```
/home/sosa/work/BF/BayesForge/Main/main.py:674: UserWarning:

Sample's batch dimension size 2000 is different from the provided 1 num_samples argument. Def
```

```
{'x': Array([[159.21693302],
        [163.9983087 ],
        [157.6438639 ],
        ...,
        [162.61740508],
        [148.05320942],
        [161.35796436]], dtype=float64)}
```

Forest plot of estimated values

```
m.diag.forest()
```

```
Unable to display output for mime type(s): text/html
```

```
Unable to display output for mime type(s): text/html
```

Density plots of the posterior distribution

```
m.diag.density()
```

```
Unable to display output for mime type(s): text/html
```

Posterior distribution plots

```
m.diag.posterior()
```

```
Unable to display output for mime type(s): text/html
```

Trace plots for MCMC chains

```
m.diag.plot_trace()
```

```
Unable to display output for mime type(s): text/html
```

Pairwise plots of the posterior distribution

```
m.diag.pair()
```

```
Unable to display output for mime type(s): text/html
```

Plot autocorrelation of MCMC chains

```
m.diag.autocor()
```

```
Unable to display output for mime type(s): text/html
```

Create rank plots for MCMC chains

```
m.diag.rank()
```

```
Unable to display output for mime type(s): text/html
```

Evolution of effective sample size across iterations

```
m.diag.plot_ess()
```

```
Unable to display output for mime type(s): text/html
```

Pareto-smoothed

```
m.diag.loo()
```

```
Computed from 2000 posterior samples and 346 observations (log scale).

         Estimate    SE
elpd_loo     -1058.44  14.67
p_loo            3.14        -

All Pareto k estimates OK (k < 0.70).
```

Widely applicable information criterion

```
m.diag.WAIC()
```

```
Computed from 2000 posterior samples and 346 observations (log scale).

         Estimate    SE
elpd_waic    -1058.43  14.67
p_waic           3.13        -
```