Tutorial 7

Statistical Computation and Analysis

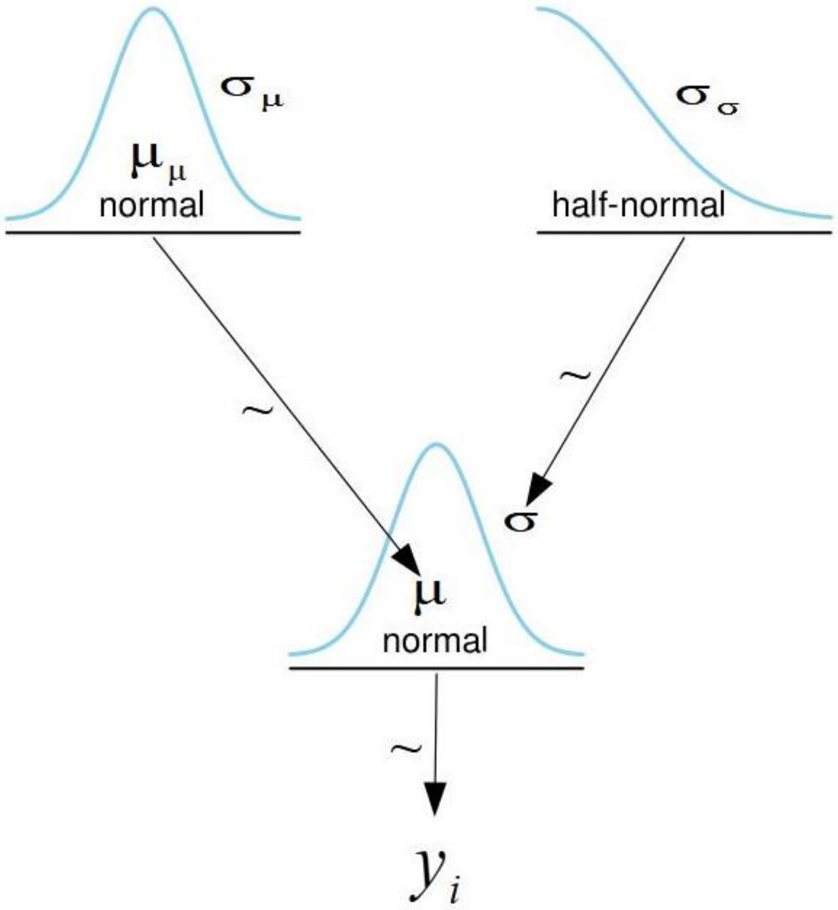
Spring 2025

Tutorial Outline

- Simple linear regression
- Bayesian p-value
- Bayesian workflow
- Data Transformation
- Heteroskedsticity

Simple Linear Regression

- We have learned about normal models.



$$y_i \sim N(\mu, \sigma)$$
$$\mu \sim N(\mu_\mu, \sigma_\mu)$$
$$\sigma \sim HalfNorm(\sigma_\sigma)$$

Simple Linear Regression

- Now we'll look at a case in which the mean depends on another variable.
  - Average height as a function of age.

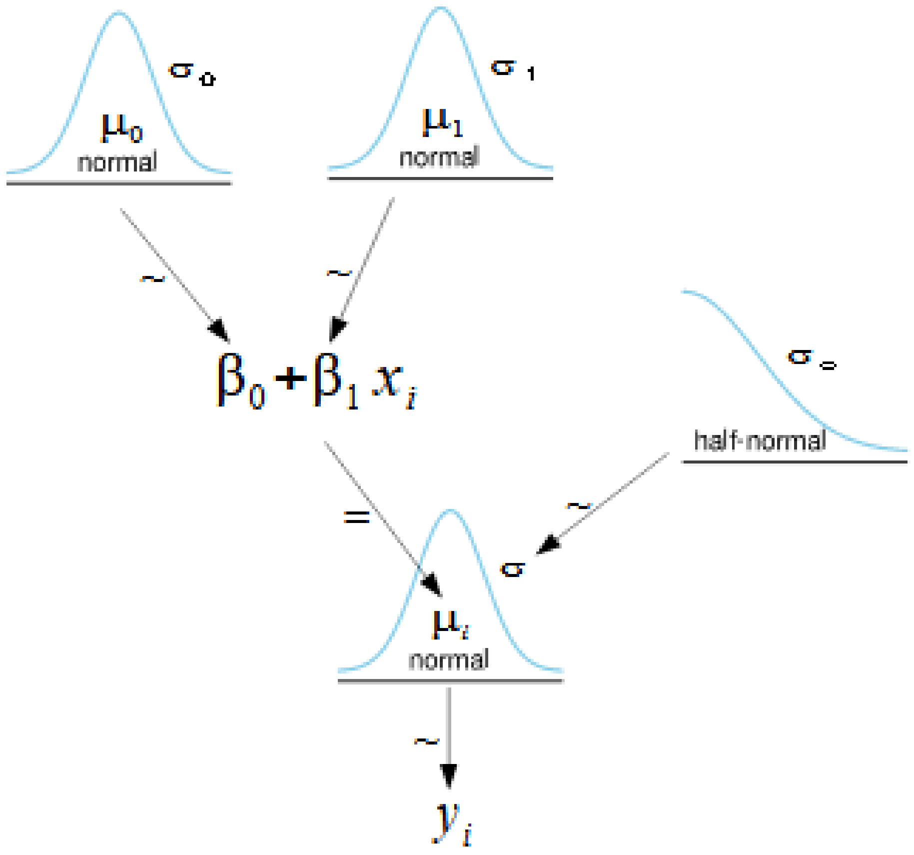
$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \beta_0 + \beta_1 x_i$$

$$\beta_0 \sim Prior0(\theta_0)$$

$$\beta_1 \sim Prior1(\theta_1)$$

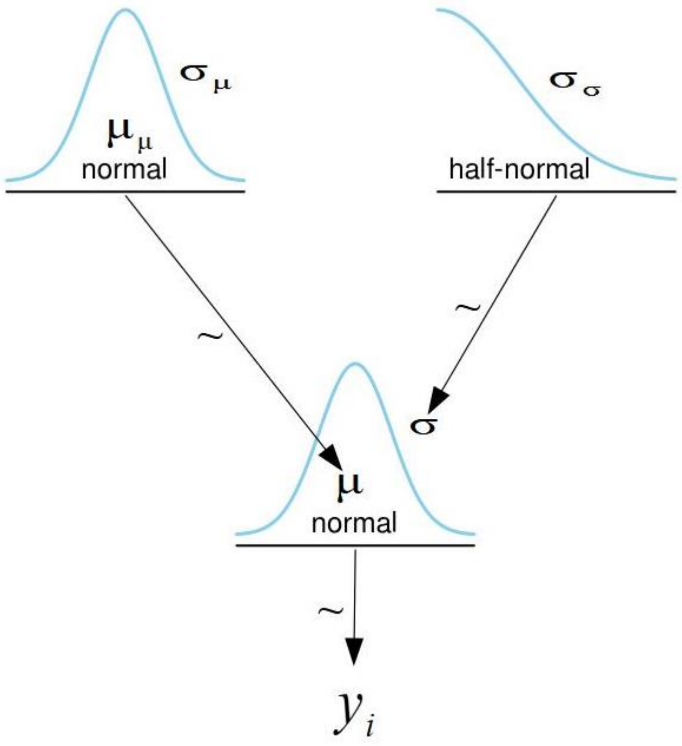
$$\sigma \sim Prior2(\theta_\sigma)$$



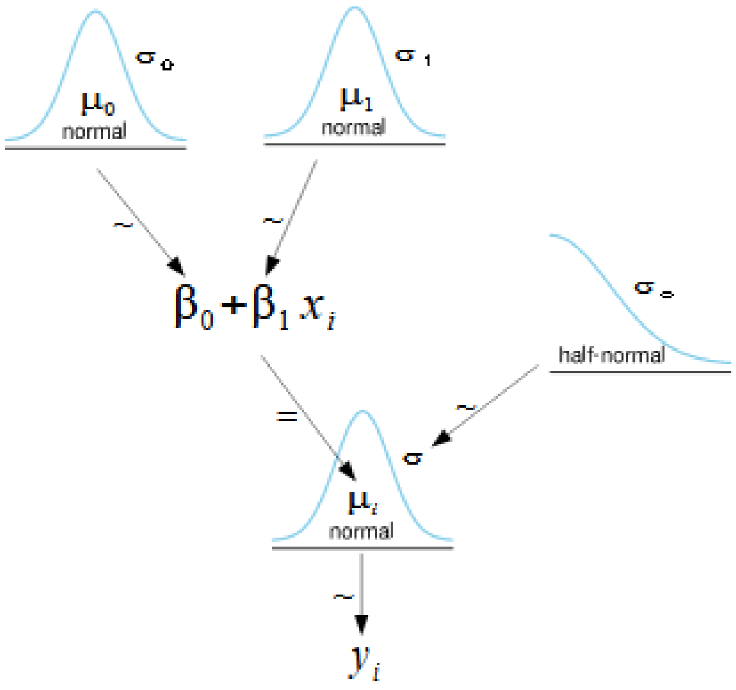
Simple Linear Regression

- Compare:

$$y_i \sim N(\mu, \sigma)$$
$$\mu \sim N(\mu_\mu, \sigma_\mu)$$
$$\sigma \sim HalfNorm(\sigma_\sigma)$$



$$y_i \sim N(\mu_i, \sigma)$$
$$\mu_i = \beta_0 + \beta_1 x_i$$
$$\beta_0 \sim N(\mu_0, \sigma_0)$$
$$\beta_1 \sim N(\mu_1, \sigma_1)$$
$$\sigma \sim HalfNorm(\sigma_\sigma)$$



Simple Linear Regression

- The main idea of linear regression is to extend the normal model by adding a predictor variable, x, to the estimation of the mean, μ.
- The meaning of the model is that there is a **linear** relationship between x and y.
- The relationship **not deterministic** because of the noise term σ**.**
- The intercept tells us the value of y when x=0.
- The slope tells us the change in y per unit change in x.

Simple Linear Regression

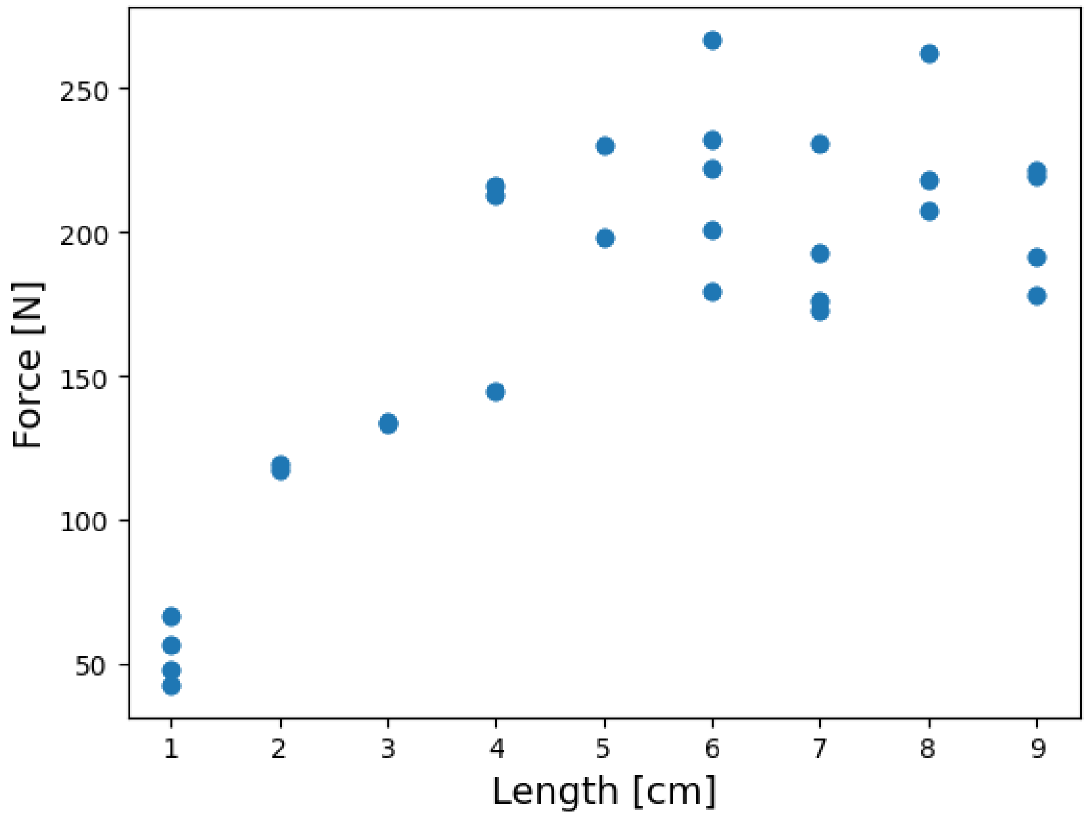
- How do we write this model in our code?

```
coords = {"data": np.arange(len(x))}
with pm.Model(coords=coords) as model_lb:
    β0 = pm.Normal("β0", mu=0, sigma=100)
    β1 = pm.Normal("β1", mu=0, sigma=10)
    σ = pm.HalfNormal("σ", 10)
    μ = pm.Deterministic("μ", β0 + β1 * x, dims="data")
    y_pred = pm.Normal("y_pred", mu=μ, sigma=σ, observed=y, dims="data")
)
```

- We use a deterministic variable

Simple Linear Regression

- A group of researches checked the connection between the muscle length and the generated force.



Simple Linear Regression

- We can see an increase in generated force for increased muscle length.
- Let’s model it using simple linear regression.

```
coords = {"data": np.arange(len(data))}
with pm.Model(coords=coords) as model_slr:
    b0 = pm.Normal("b0", mu=50, sigma=50)
    b1 = pm.Normal("b1", mu=0, sigma=50)
    sig = pm.HalfNormal("sig", 10)
    mu = pm.Deterministic("mu", b0 + b1 * data.Length, dims="data")
    y_pred = pm.Normal("y_pred", mu=mu, sigma=sig, observed=data.Force, dims="data")

    idata_slr = pm.sample(1000, chains = 4)
```

Simple Linear Regression

- Look at our inference object:

```
idata_slr
```

arviz.InferenceData

▼ posterior

xarray.Dataset

▶ Dimensions: (**chain**: 4, **draw**: 1000, **data**: 21)

▼ Coordinates:

chain	(chain)	int64	0 1 2 3
draw	(draw)	int64	0 1 2 3 4 5 ... 995 996 997 998 999
data	(data)	int64	0 1 2 3 4 5 6 ... 15 16 17 18 19 20

▼ Data variables:

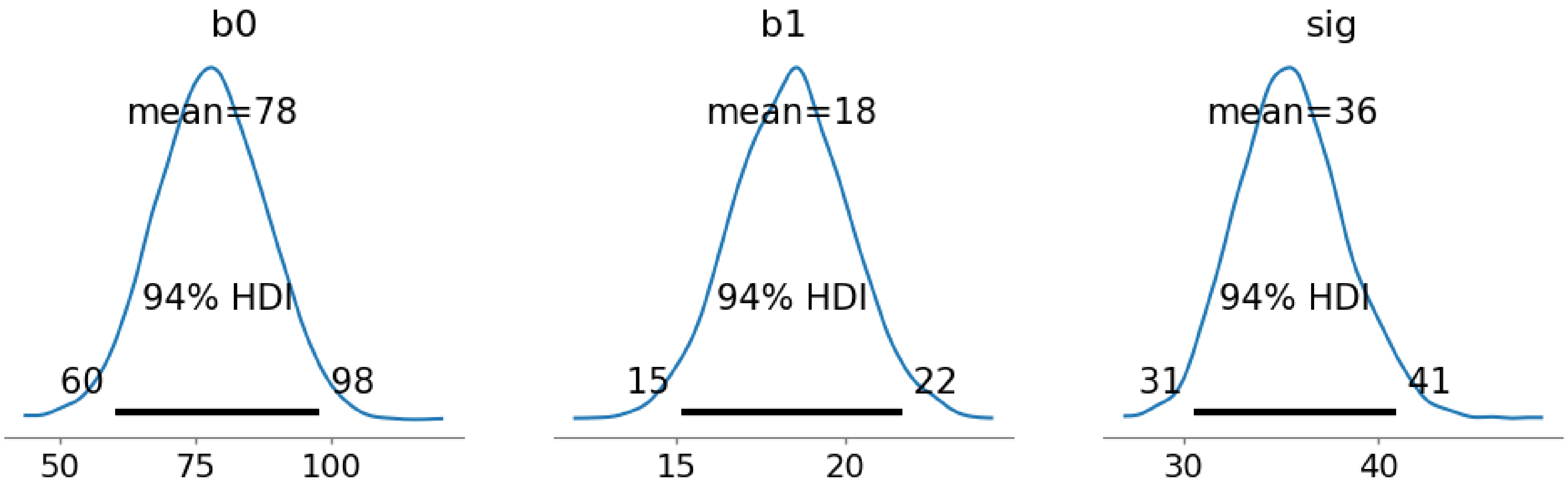
b0	(chain, draw)	float64	97.98 92.05 72.96 ... 55.83 55.83
b1	(chain, draw)	float64	18.68 17.28 19.54 ... 22.86 22.86
mu	(chain, draw, data)	float64	116.7 116.7 116.7 ... 238.7 261.6
sig	(chain, draw)	float64	33.3 33.38 30.41 ... 23.68 23.68

▶ Indexes: (3)

▶ Attributes: (6)

Simple Linear Regression

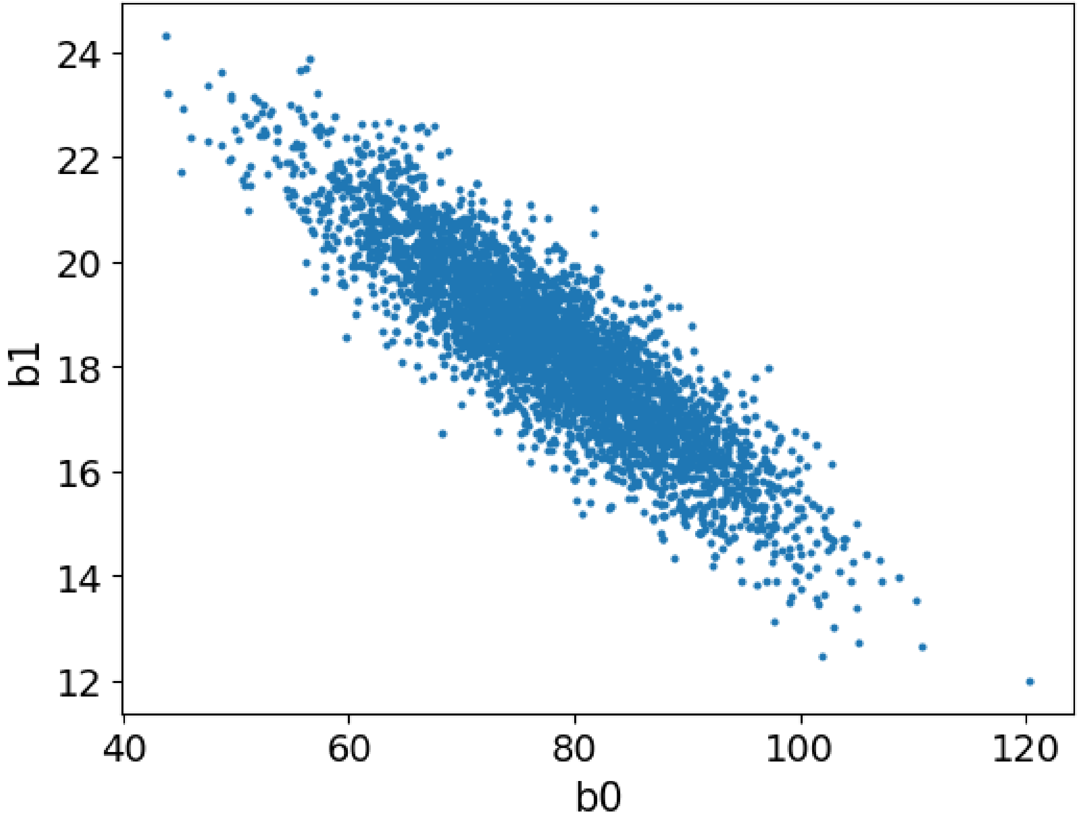
- Look at the posteriors for each of our parameters:



- If we take the mean of each distribution: $\mu = 64 + 22x$
  - But there are distributions for the intercept and the slope, so we can also take other values.

Simple Linear Regression

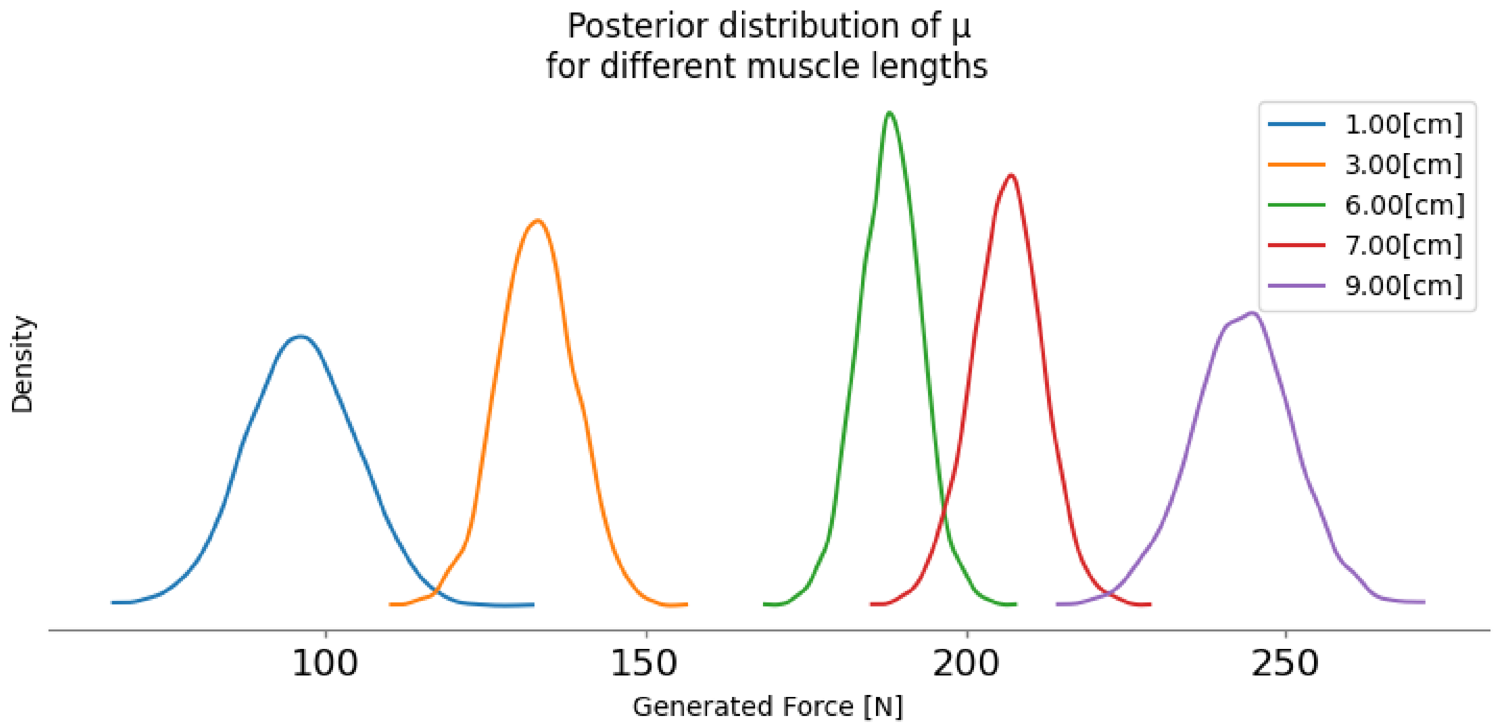
- The samples are from the joint distribution and the parameters by be correlated.




```python
az.plot_pair(idata_slr, var_names=['b0', 'b1'])
```

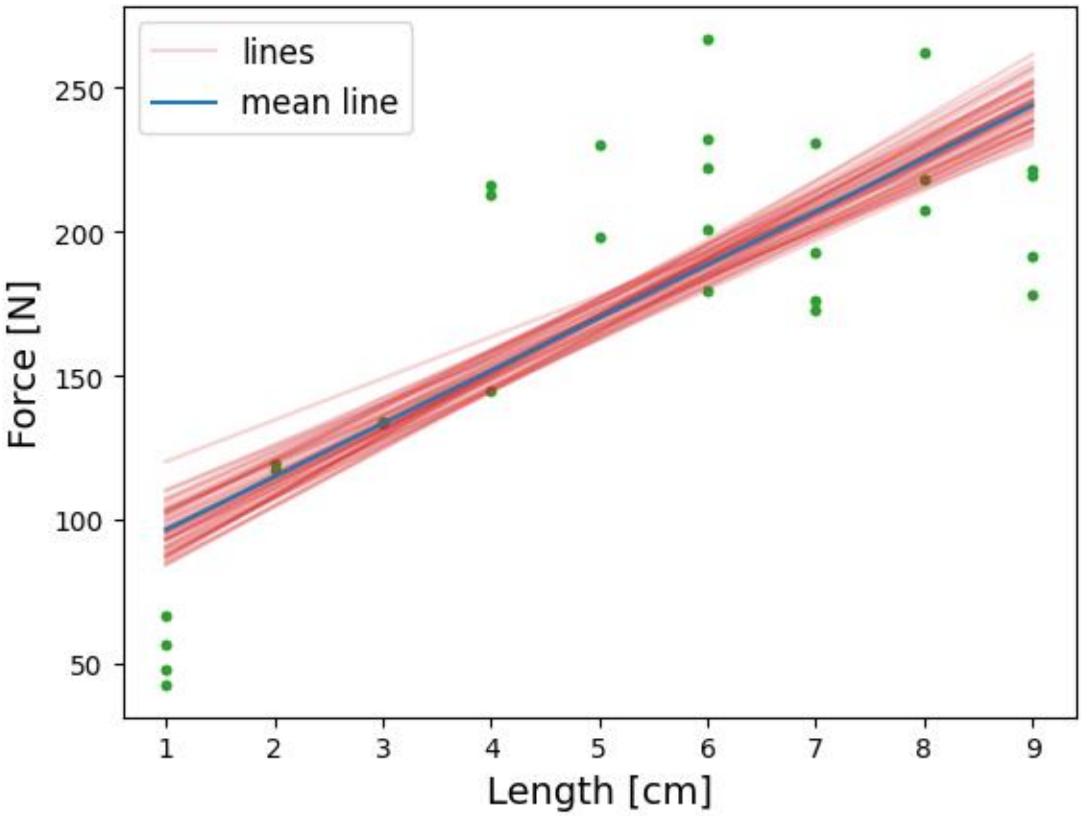
Simple Linear Regression

- We have a distribution of μ for each value of x (muscle length).
- Some examples:



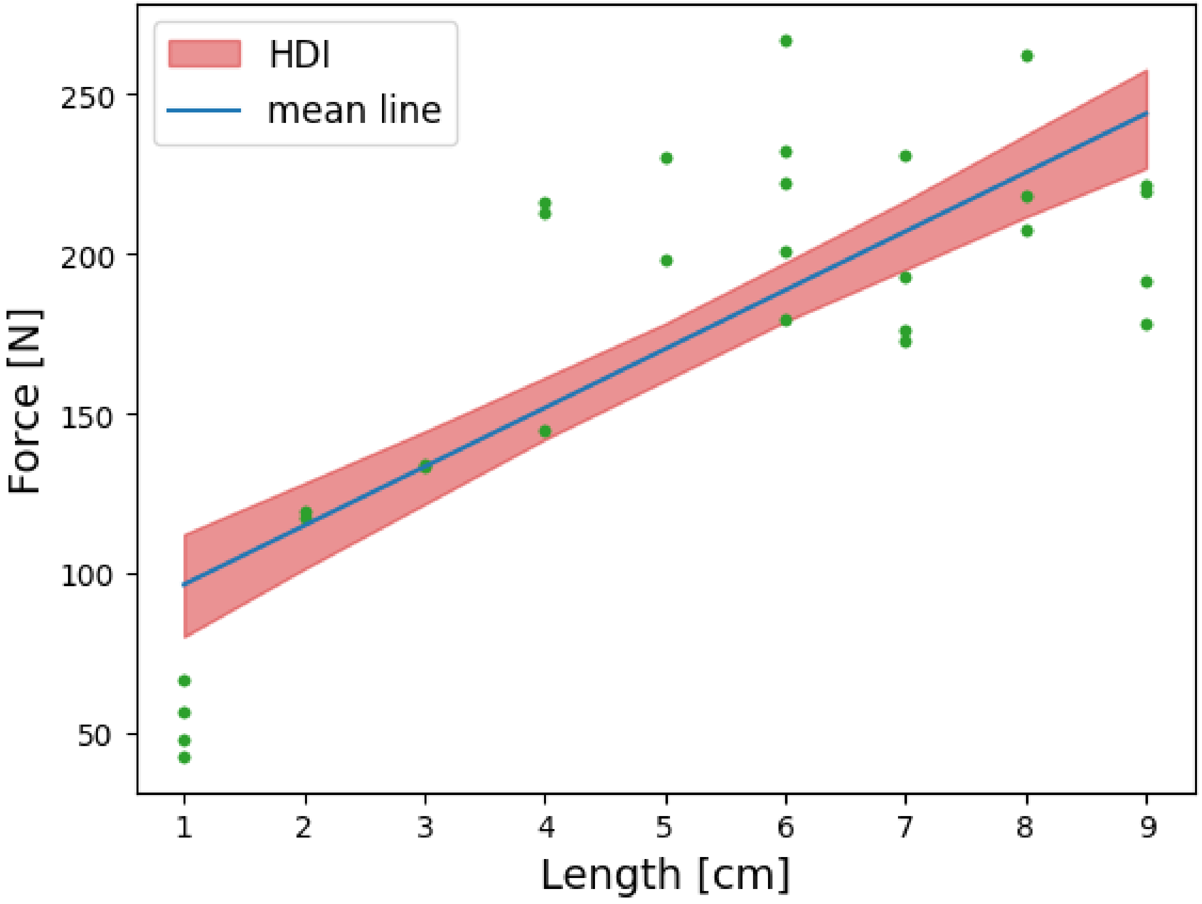
Simple Linear Regression

- Let's plot some possible regression lines using samples from the posterior.
  - This demonstrates the uncertainty we have regarding the values of the parameters.



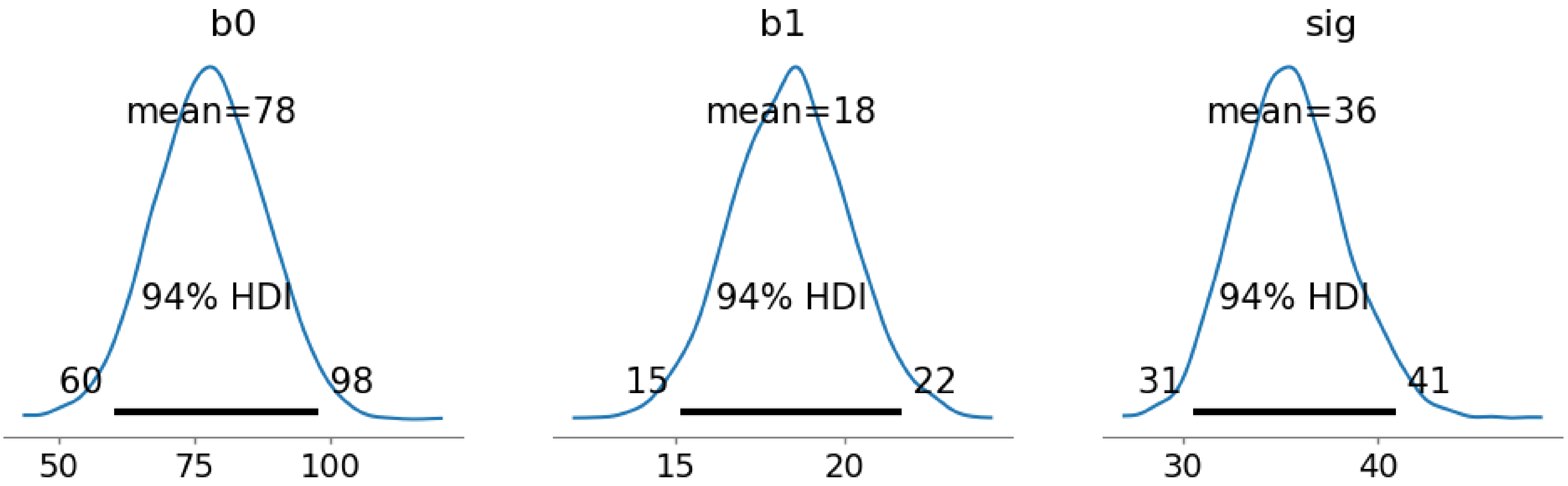
Simple Linear Regression

- We can also look at the HDI for the regression line.



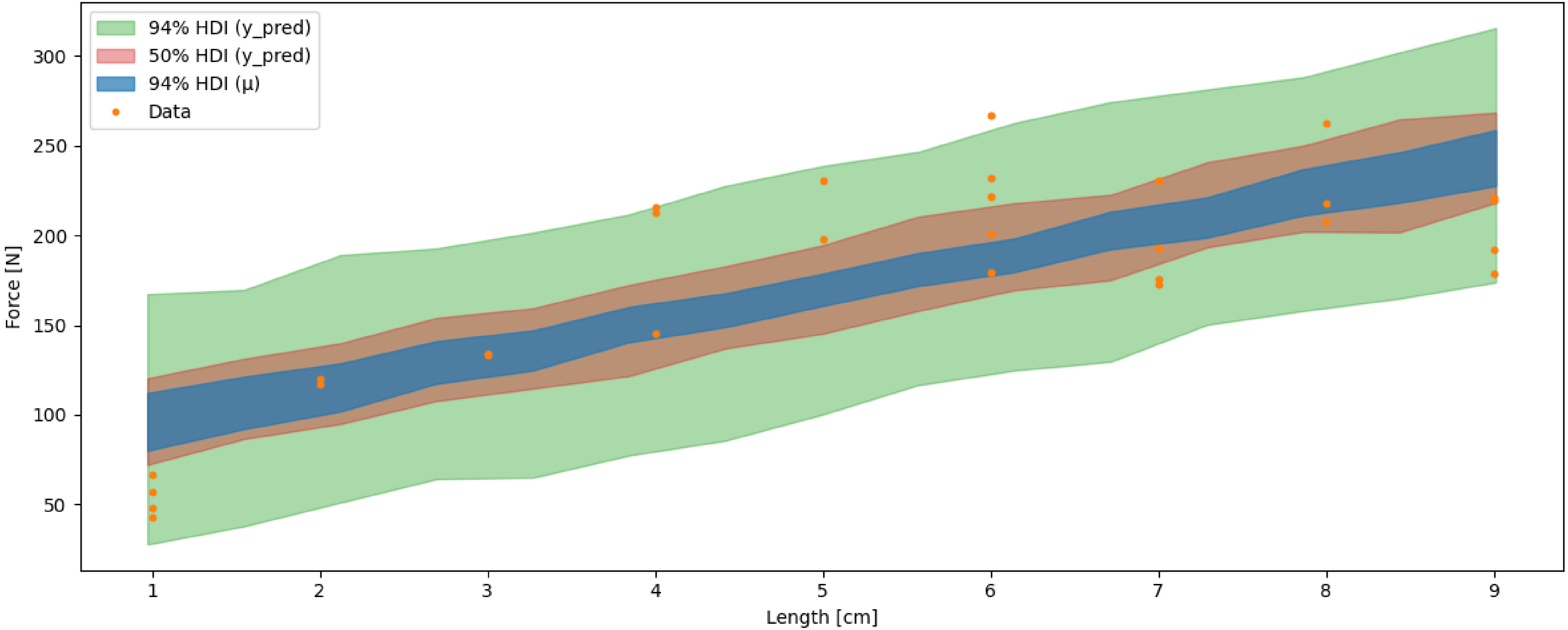
Simple Linear Regression

- There are a lot of possible regression lines that can make sense given the analysis.



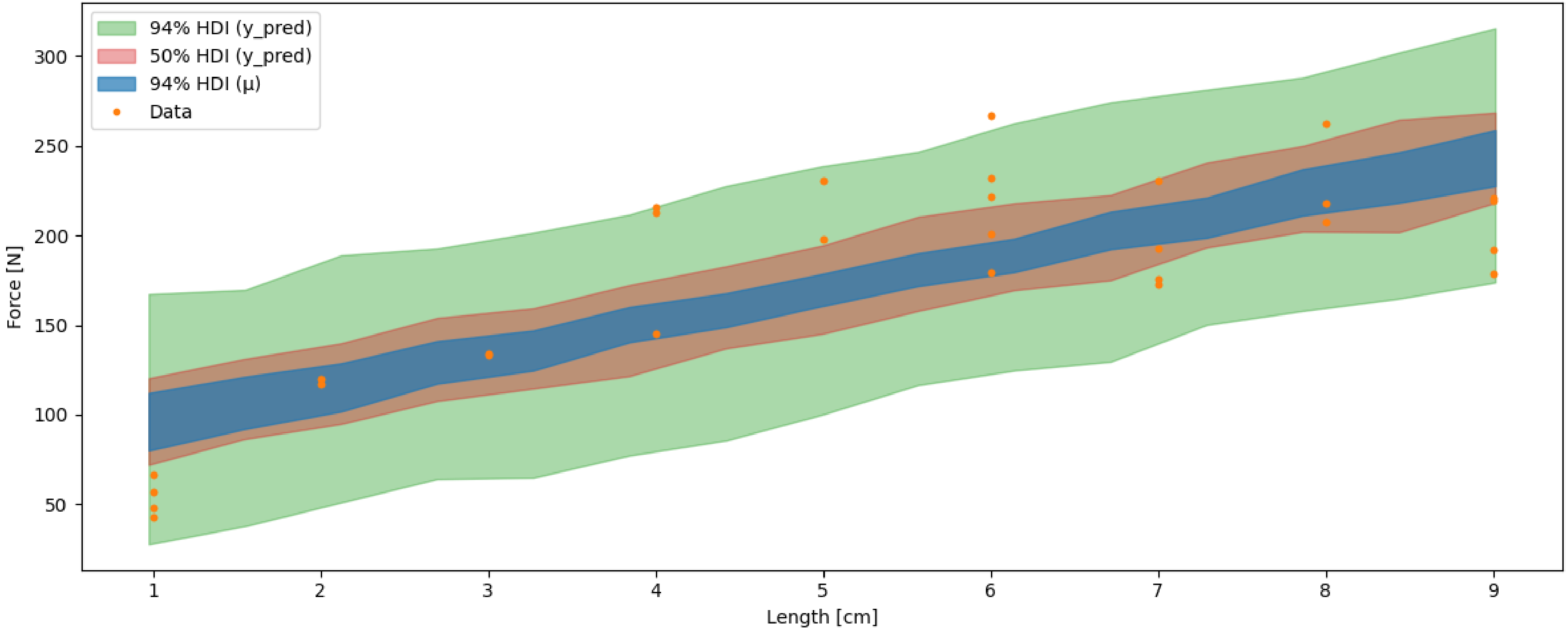
Simple Linear Regression

- Posterior predictive sampling:
  - Sample from the posterior distribution of the parameters
  - Sample from the likelihood given these posterior samples



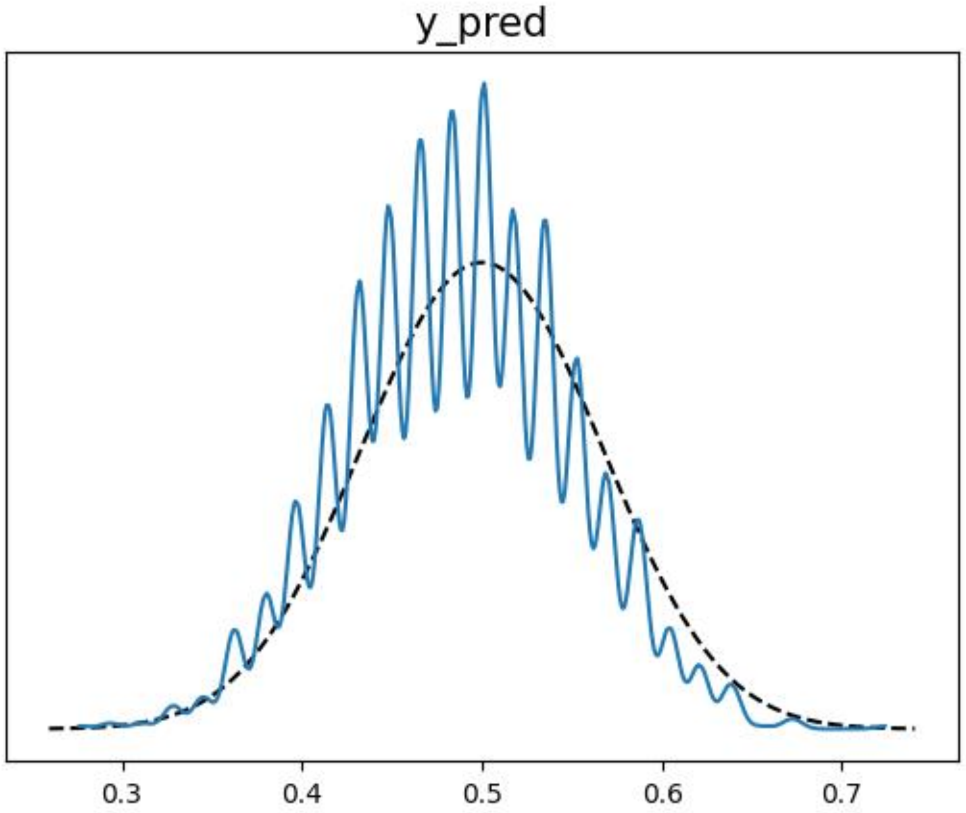
Simple Linear Regression

- Posterior predictive sampling:
  - Visualizes the uncertainty in both posterior mean and posterior predictive.
  - 50% of the data should in the 50% posterior predictive HDI
  - Posterior predictive should not have empty areas



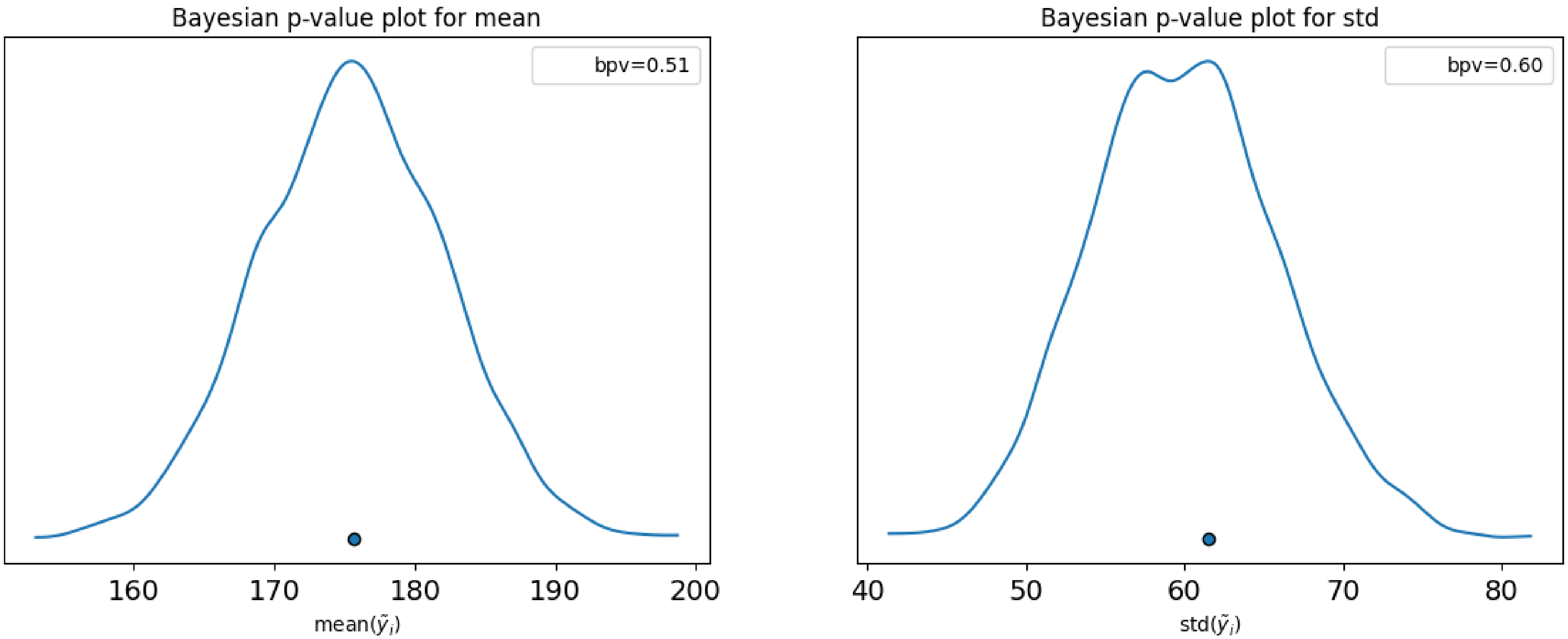
Bayesian p-value

- What percentage of posterior predictive values are less than actual data values?
  - We expect that it should be around half.
- We get a distribution over posterior predictive sets.

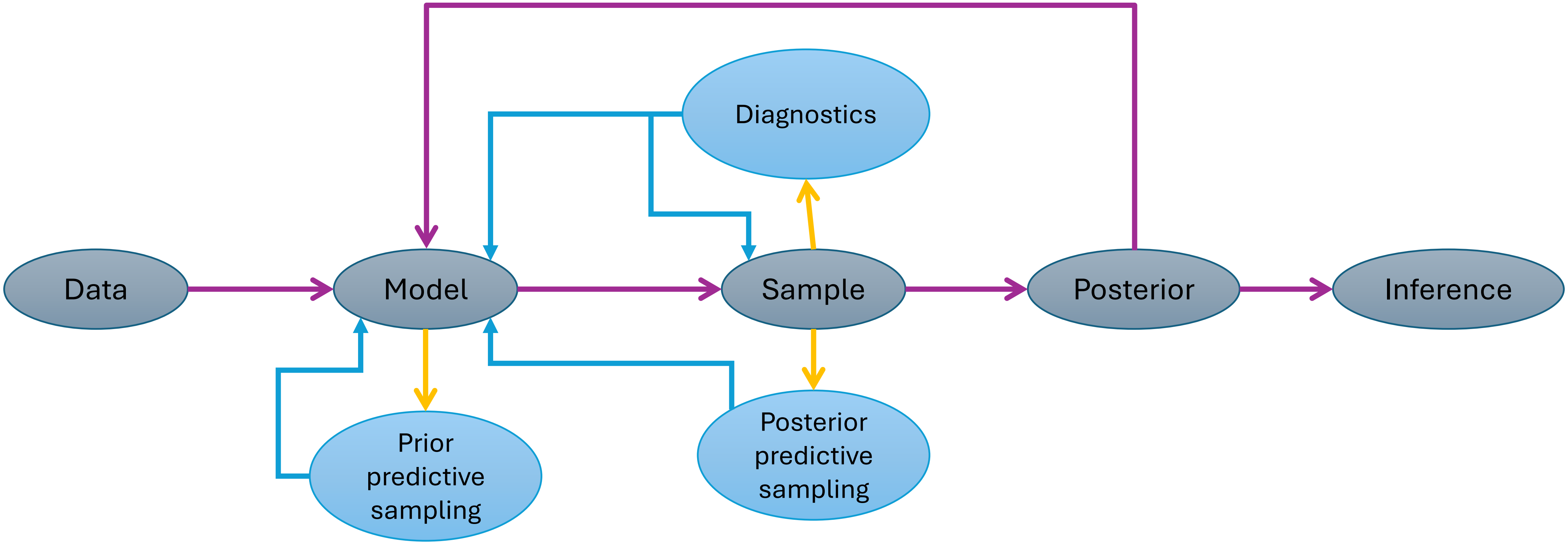


Bayesian p-value

- Instead of comparing value by value, we can compare for chosen statistics, such as the mean and the standard deviation.
  - The dot is the value for our observed data.
  - The distributions are those of the statistic for each generated dataset.



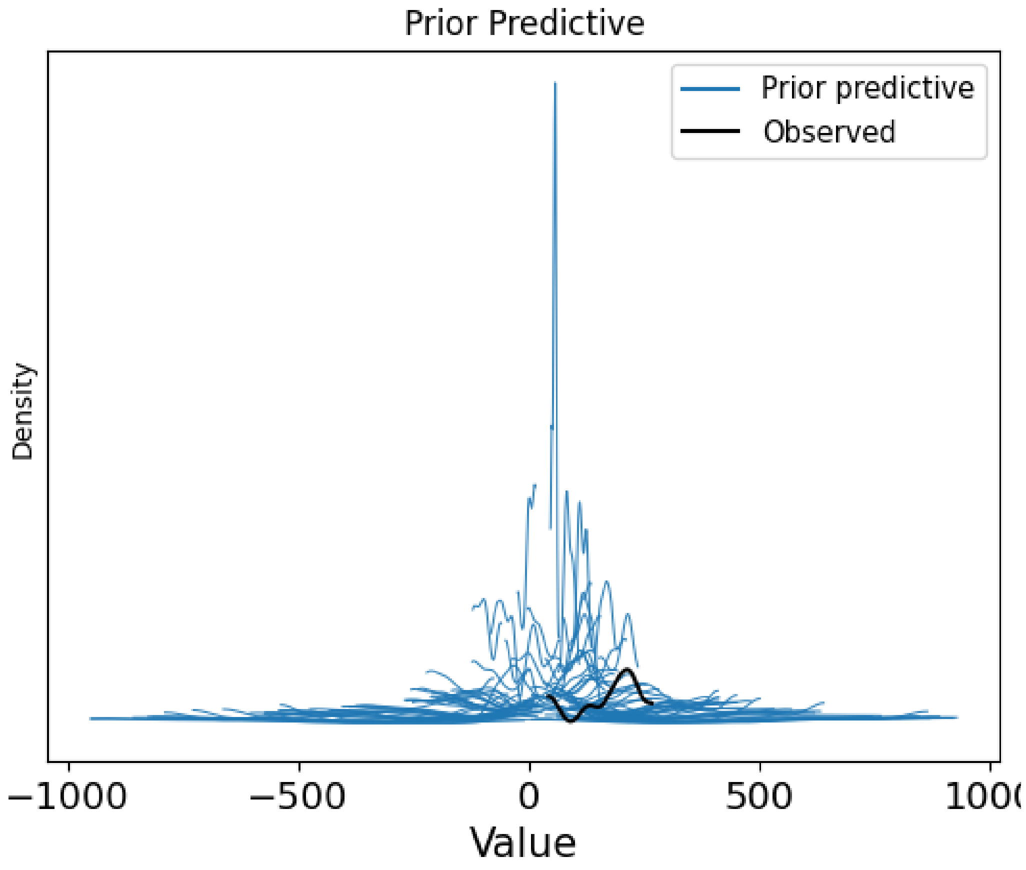
Bayesian Workflow

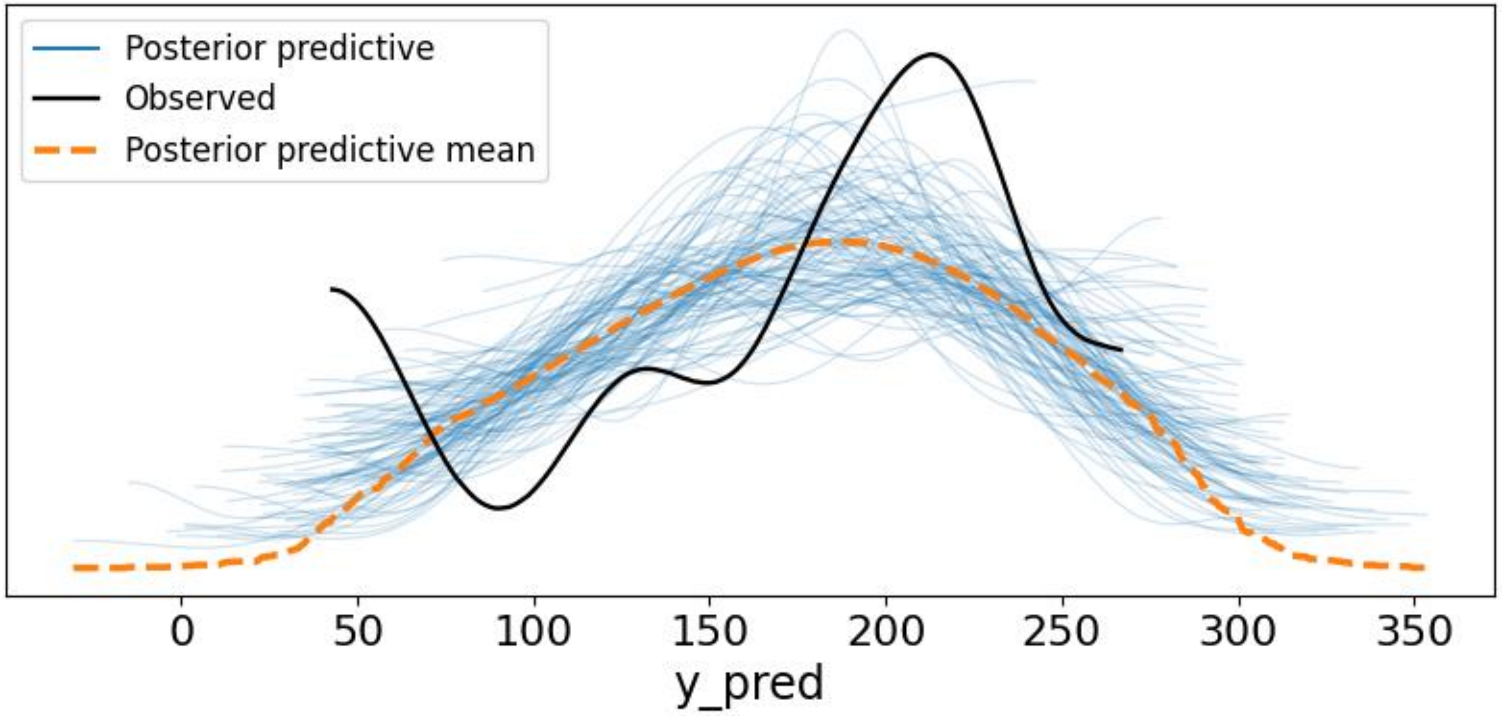


The steps in the Bayesian workflow

Step	Test	Solutions
Data collection	Data validation procedures	Improve methodology Develop protocols Document processes
Propose model	Generate graph Take sample	Test dimensions Debug
Sample prior predictive	Prior predictive plots	Simplify model Change priors
Sample posterior	Trace plots Rhat ESS MCSE Divergences	Improve sampling Change initialization Reparameterize model Simplify model Change priors
Sample posterior predictive	Posterior predictive plots Bayesian p values	Change model

Prior and Posterior Predictive





Data transformation

- We can compute a transformation on our data.

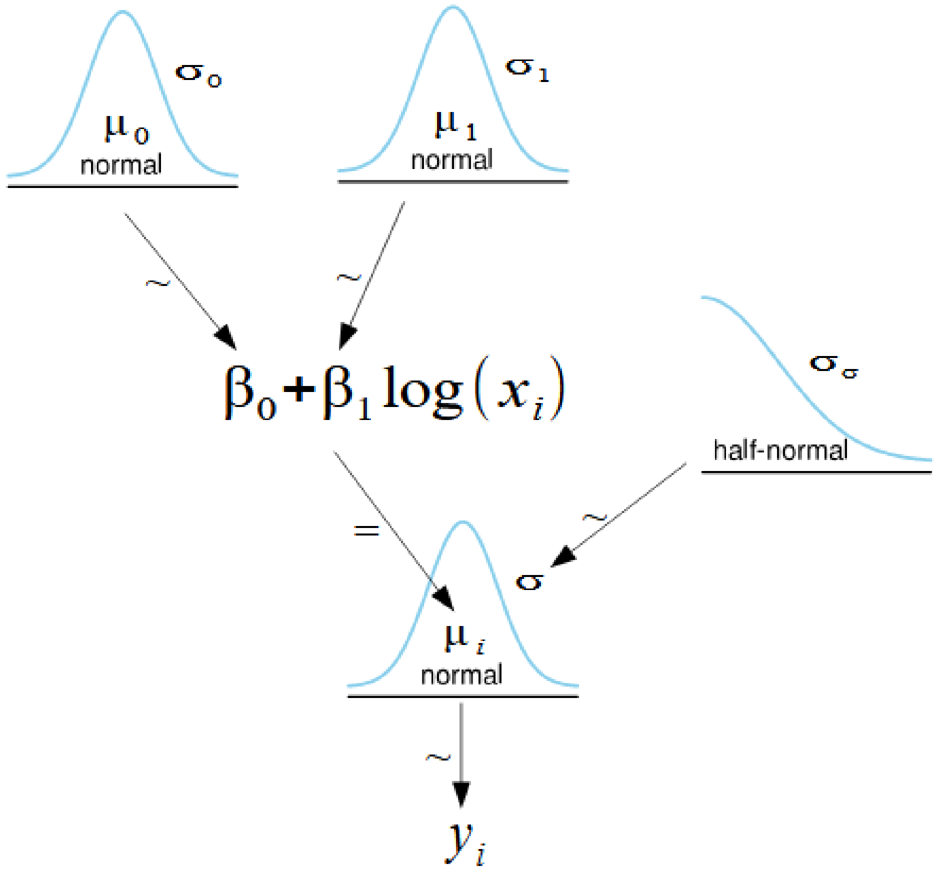
$$y_i \sim N(\mu_i, \sigma)$$

$$\mu_i = \beta_0 + \beta_1 \log(x_i)$$

$$\beta_0 \sim N(\mu_0, \sigma_0)$$

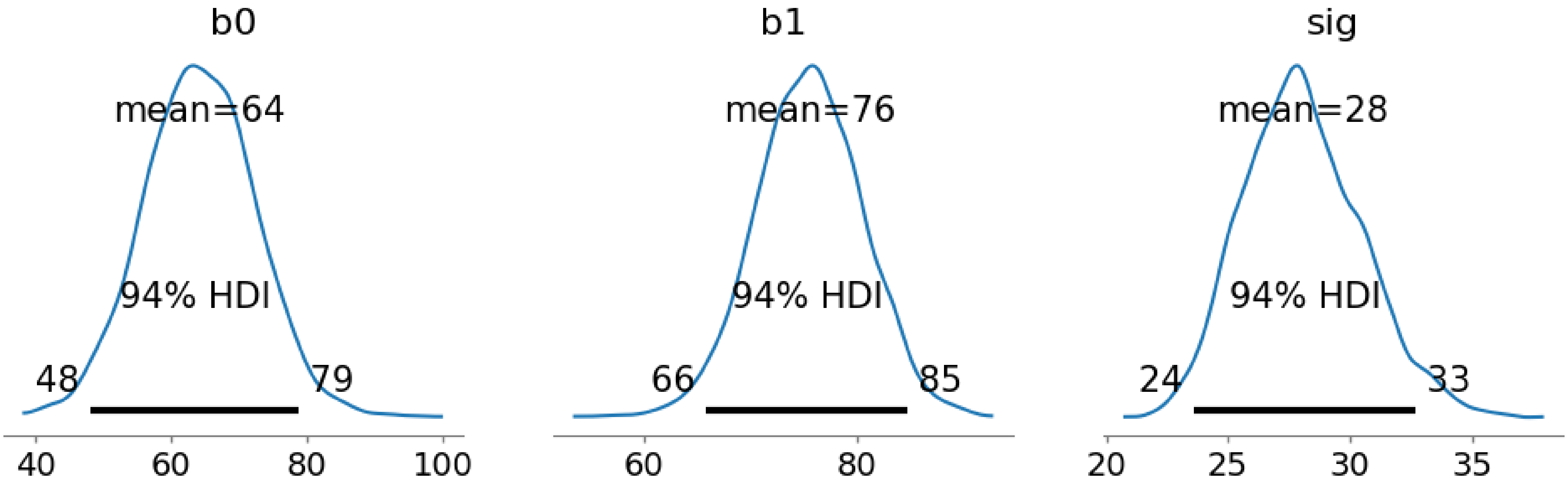
$$\beta_1 \sim N(\mu_1, \sigma_1)$$

$$\sigma \sim HalfNorm(\sigma_\sigma)$$



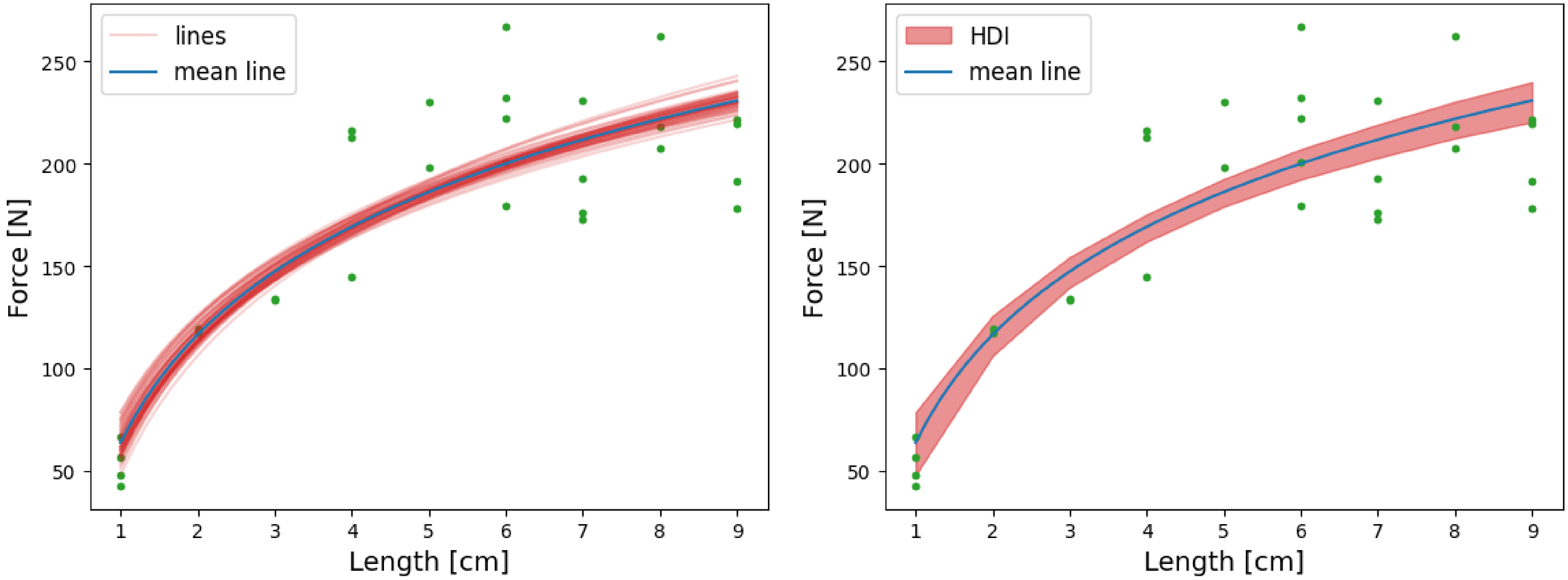
Data transformation

- Look at the posteriors for each of our parameters:



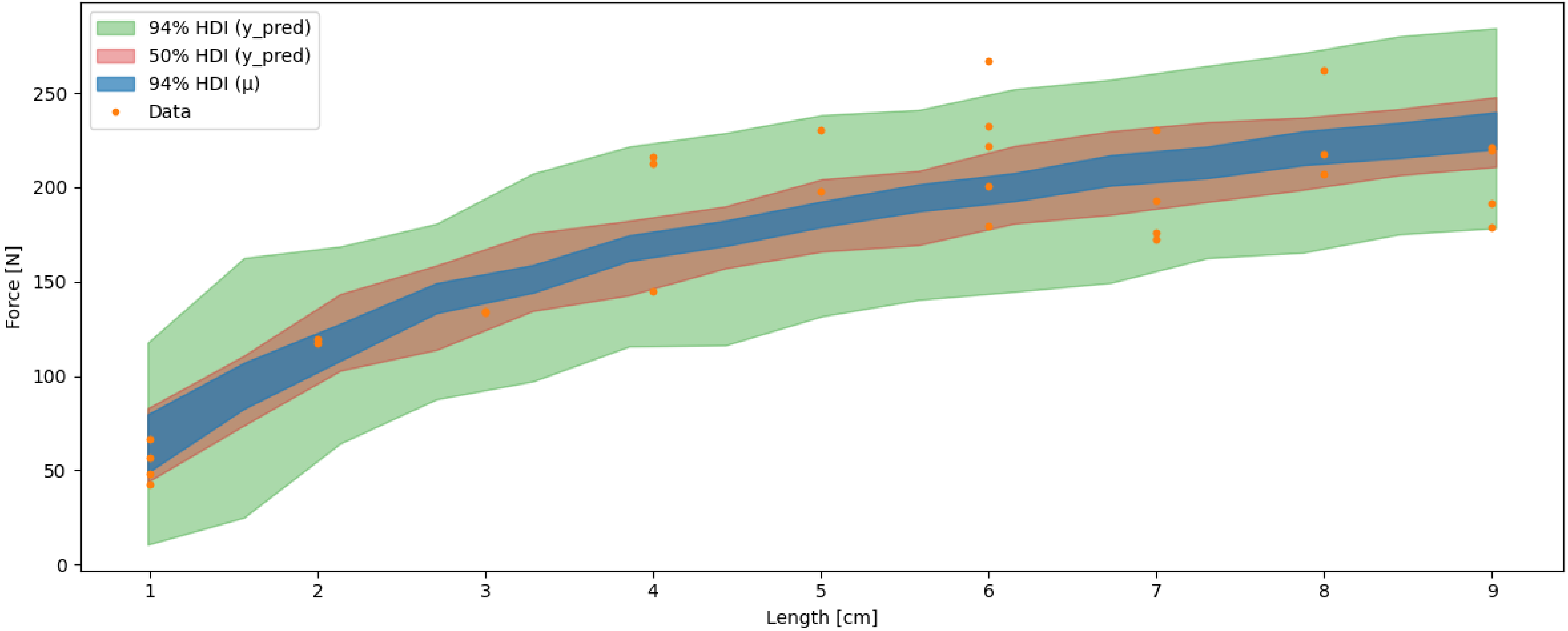
Data transformation

- Possible regression lines using samples from the posterior + HDI:



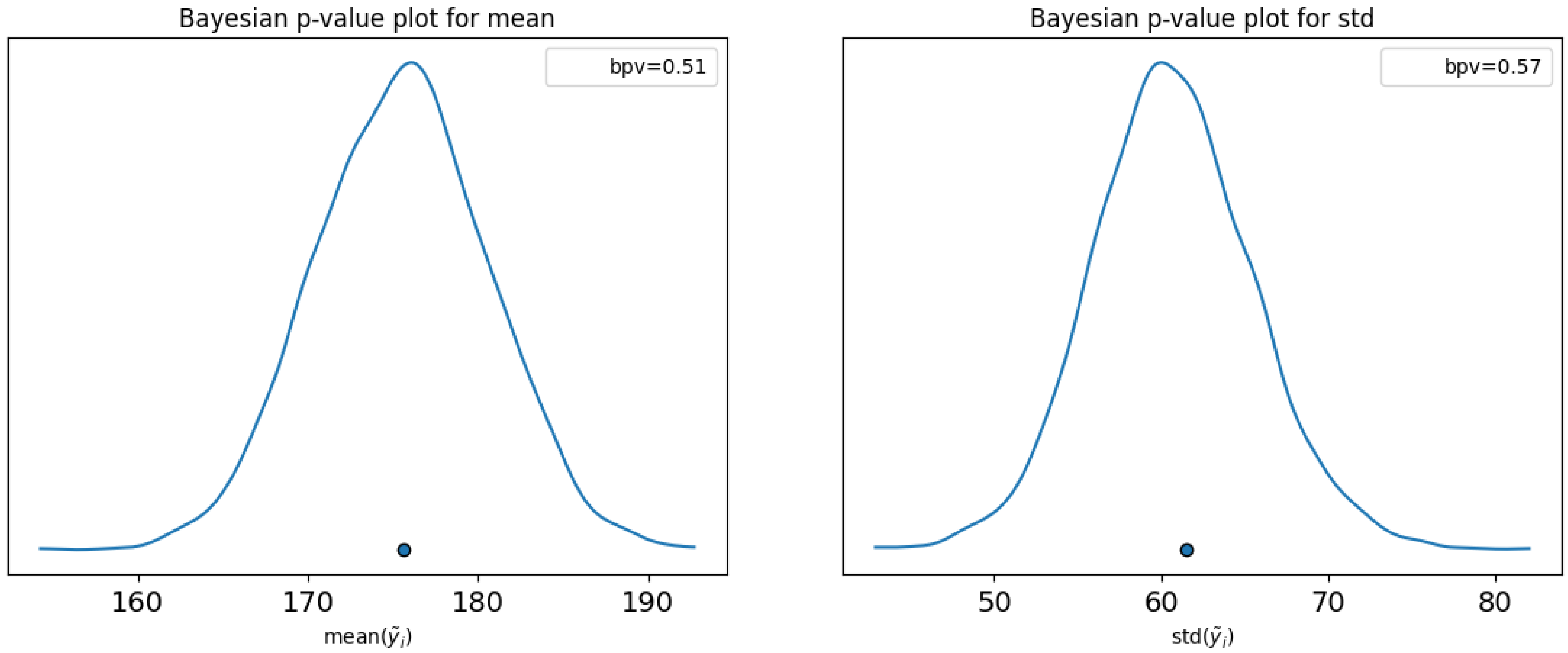
Data transformation

- Posterior predictive sampling:



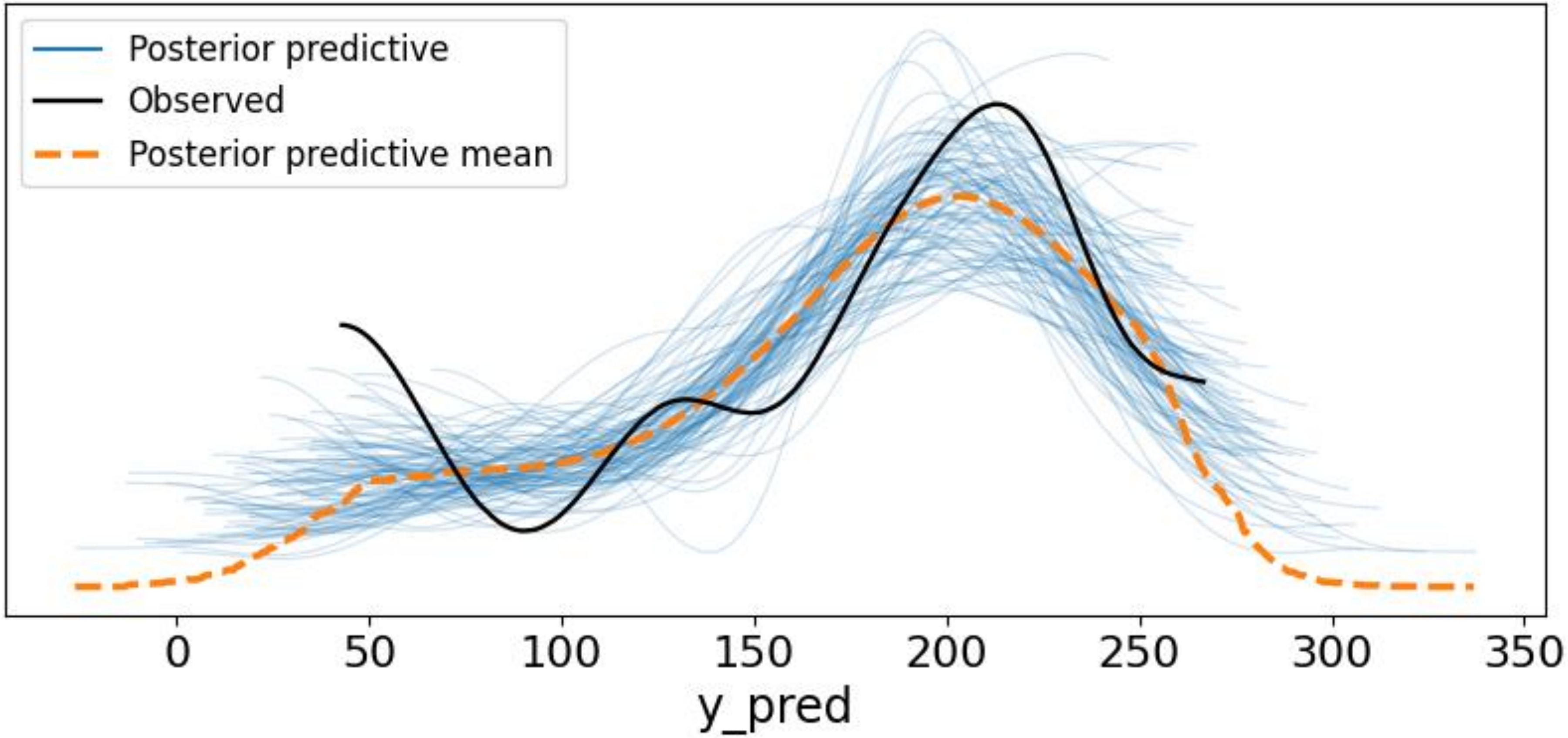
Data transformation

- Bayesian p-value:



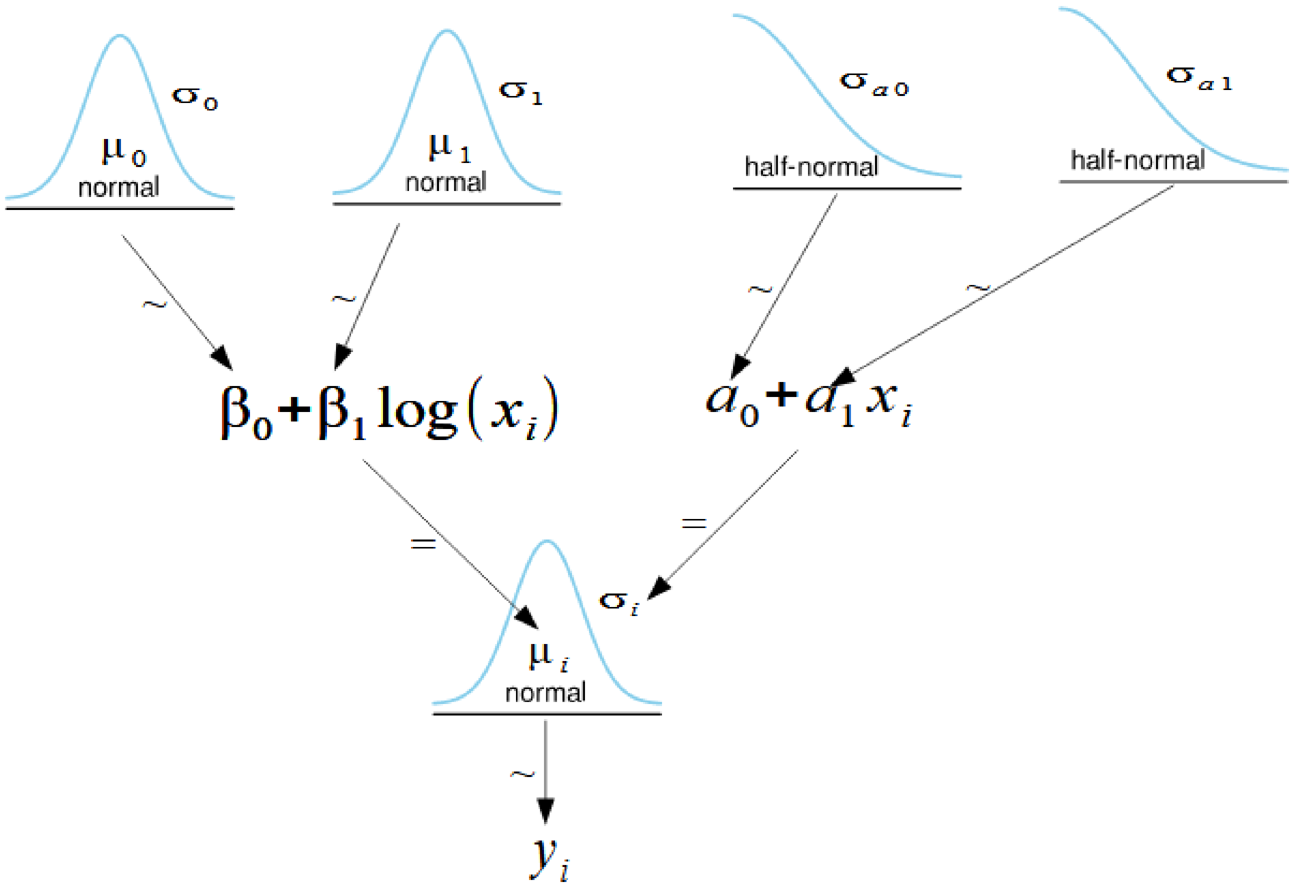
Data transformation

- Posterior predictive:

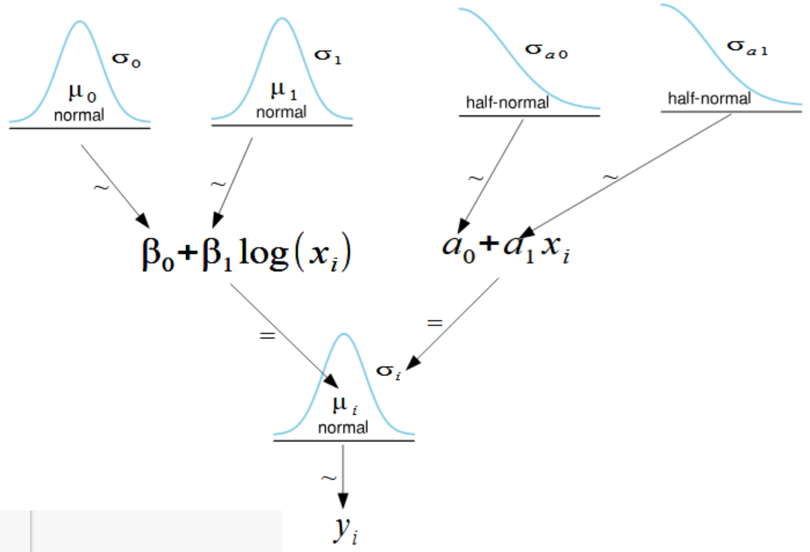


Heteroskedsticity

- The variance also depends on the independent variable.



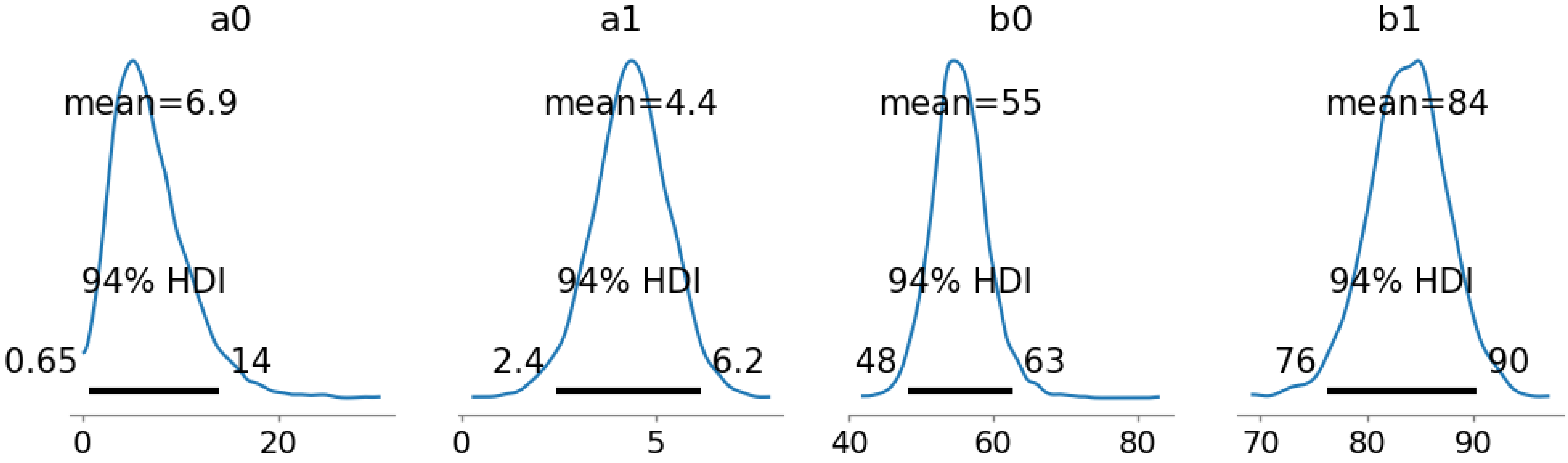
Heteroskedsticity




```
coords = {"length": range(len(data.Length))}
with pm.Model(coords=coords) as model_vv:
    x_shared = pm.Data("x_shared", data.Length, dims=["length"])
    b0 = pm.Normal("b0", mu=50, sigma=50)
    b1 = pm.Normal("b1", mu=0, sigma=50)

    a0 = pm.HalfNormal("a0", sigma=20)
    a1 = pm.HalfNormal("a1", sigma=20)

    mu = pm.Deterministic("mu", b0 + b1 * np.log(x_shared), dims="length")
    sig = pm.Deterministic("sig", a0 + a1 * x_shared, dims="length")

    y_pred = pm.Normal("y_pred", mu=mu, sigma=sig, observed=data.Force, dims="length")

    idata_vv = pm.sample(1000, chains = 4, target_accept = 0.95)
```

Heteroskedsticity

- We can look at the posterior distributions for our parameters.



Heteroskedsticity

- And our final result: