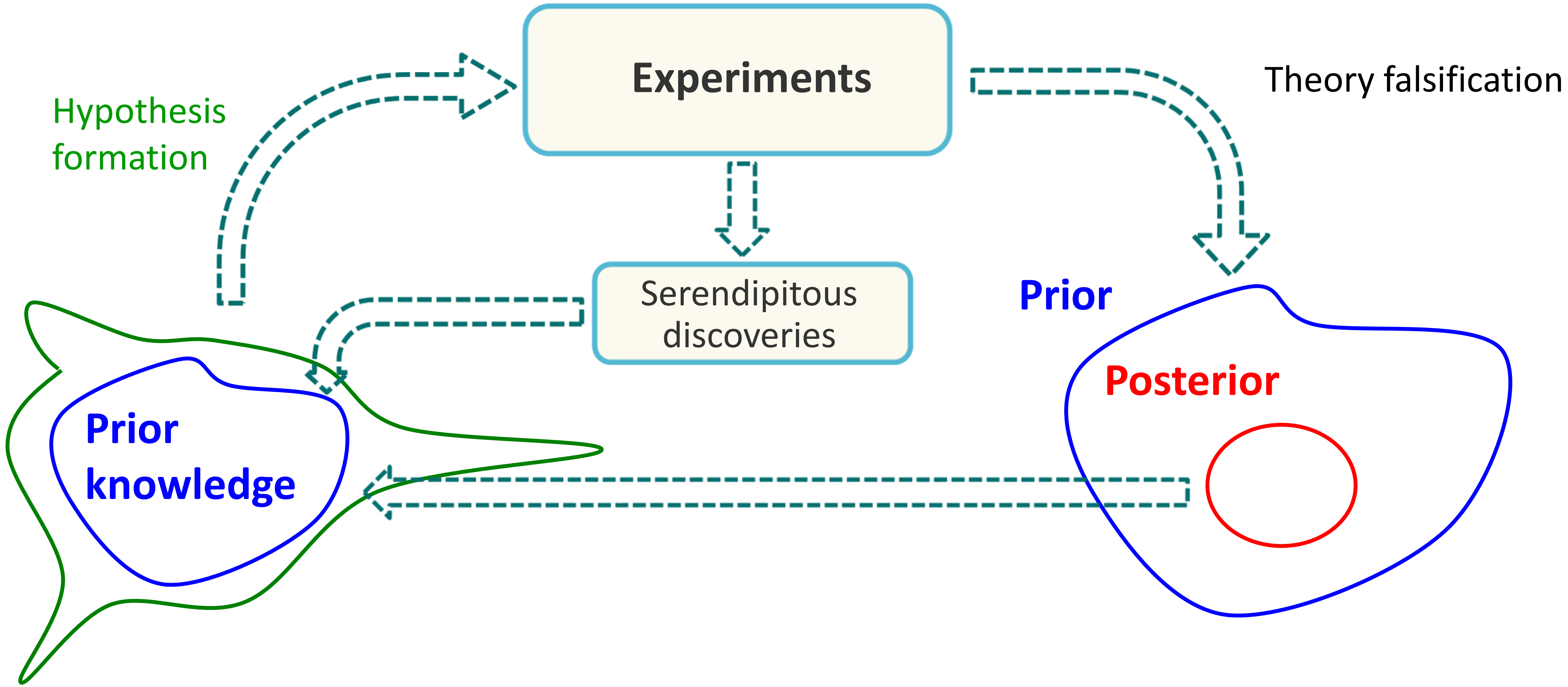
THE UNIVERSITY of TENNESSEE KNOXVILLE

# Lecture 27: Bayesian Solutions for (Linear) Real World Problems

Instructor: Sergei V. Kalinin

# Where do Bayesian methods fit into R&D?




**Rewards:** Why are we doing science/R&D?
**Policies:** Exploration-exploitation balance
**Instrument development:** New tools create new opportunities
**Hypothesis making:** Extrapolation into the unknown

# How does it work?

**Forward model:**
Theory

**Hypothesis driven science:**
What we want to learn

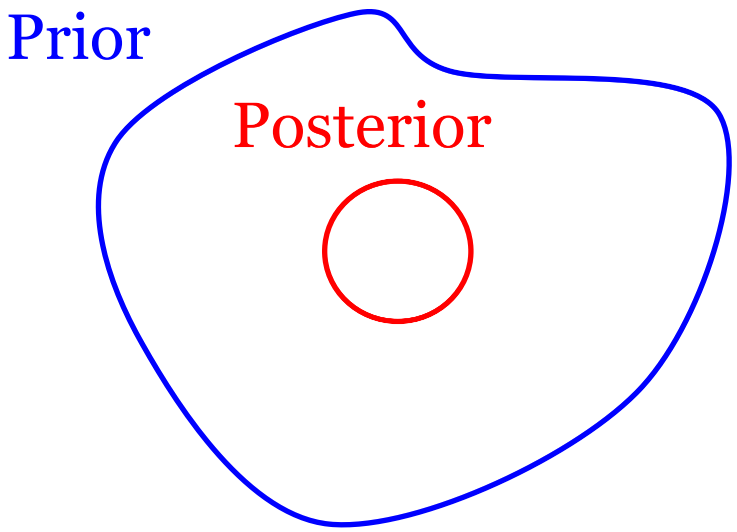
**Domain expertise**

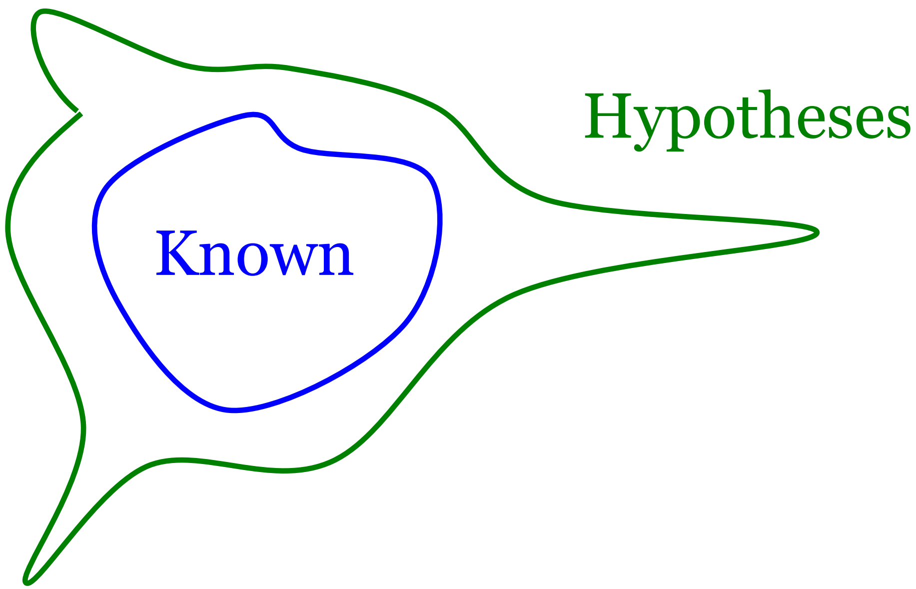
$$P(Theory|Data) = \frac{P(Data\ |Theory)\,P(Theory)}{P(Data)}$$

**High Performance Computing**

**Experimentalists know the priors. Albeit they do not know that they know it, or how to convert them to algorithmic form**

# What Bayesian Methods (and ML) cannot do?




**Refinement:**
Can be defined as probabilistic model

**Hypothesis formation:**
How can we do it?

# How does it work?

Human scientist

Hypothesis making

Model of the process (priors, model)

Bayesian Inference

Inferred parameters of the model (posteriors)

Prediction

Simulated data

Physical process (generates data with certain distribution)

Experiment

Observed Data

Posterior predictive check

# When Bayes is easy

1. **Beta distribution is conjugate to the Binomial likelihood**: This means that if you have a Binomial likelihood (e.g., flipping coins) and a Beta-distributed prior on the probability of heads, the resulting posterior distribution after observing some data will also be a Beta distribution.

2. **Gamma distribution is conjugate to the Poisson likelihood**: If you're observing the number of events occurring in fixed intervals of time or space (modeled by a Poisson distribution) and have a Gamma-distributed prior on the rate parameter, the posterior will also be Gamma-distributed.

3. **Normal distribution is conjugate to itself**: If both the likelihood and the prior are normally distributed, then the posterior will also be normally distributed.

Used to be biggest problem for Bayesian analysis – we **could** only solve simplest problems, or force real world problems into oversimplified frameworks

# Probability of data is generally intractable

- Prior information $p(\theta)$ on parameters $\theta$
- Likelihood of data given parameter values $f(x|\theta)$

**Problem for physics/theory**

$$p(\theta|x) = \frac{\boxed{f(x|\theta)}\, p(\theta)}{\boxed{f(x)}}$$

**Problem for computation**

$$f(x) = \int_{-\infty}^{\infty} f(x|\theta)p(\theta)d\theta$$

To use Bayesian methods, we need to be able to evaluate the denominator, which is the integral of the numerator over the parameter space. In general, this integral is very hard to evaluate.

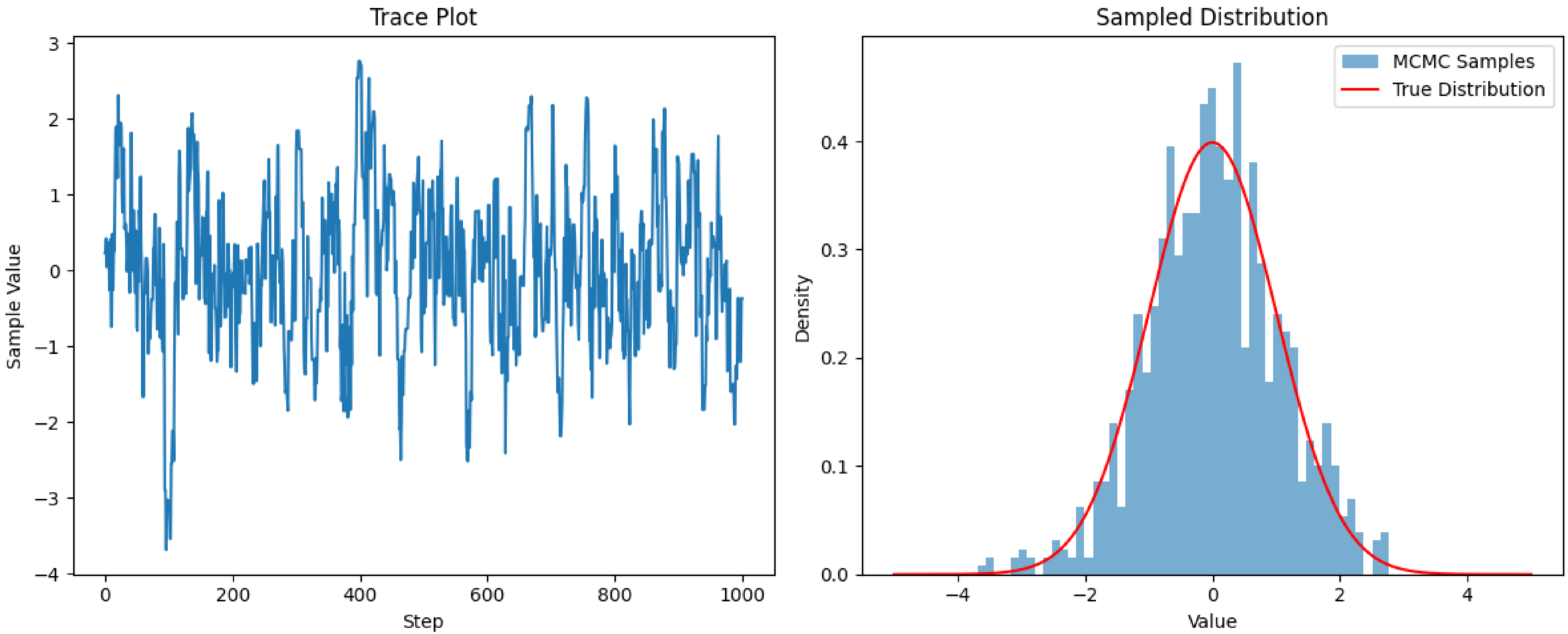
# Metropolis-Hastings algorithm

We don't need to evaluate any integral, *we just sample from the distribution many times* (e.g., 50K times) and find (estimate) the posterior mean, middle 95%, etc., from that.

- Initial value $\theta^{(0)}$ to start the Markov Chain
- Propose new value $\theta'$
- Accepted value:

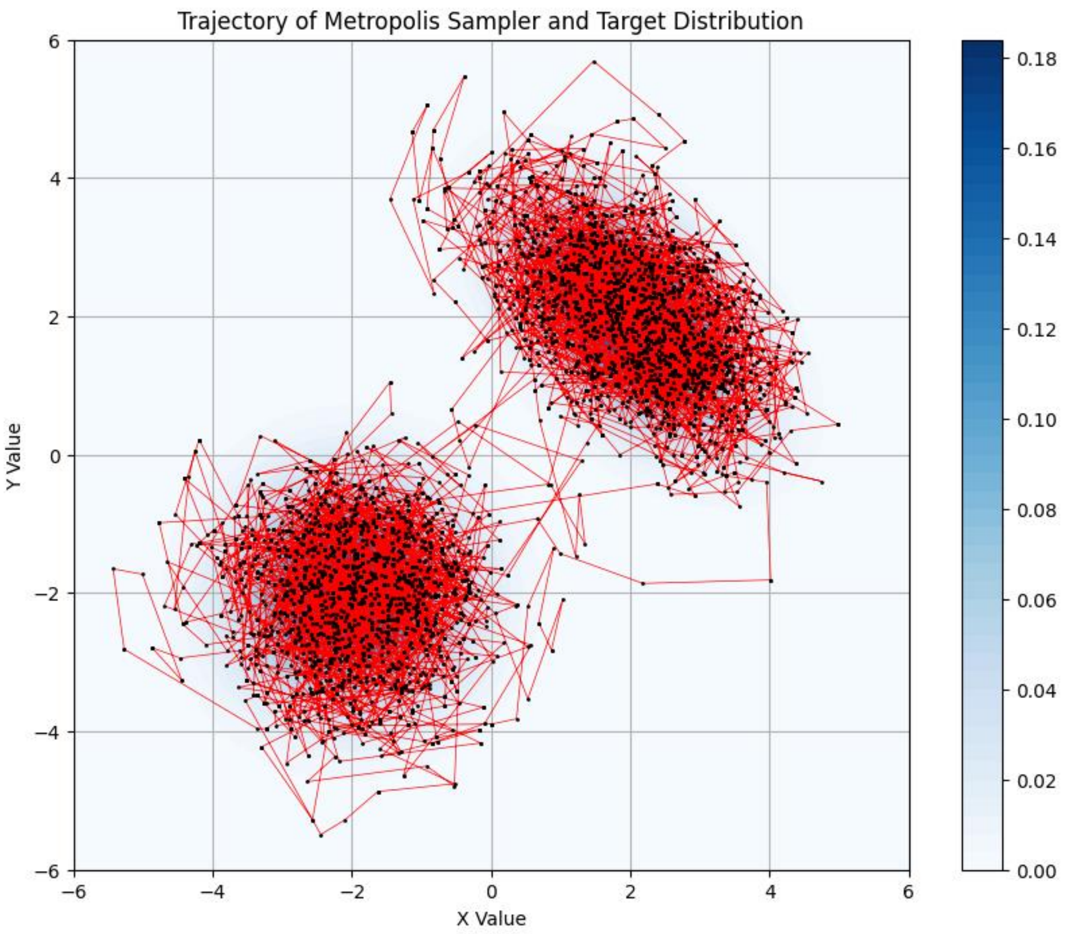
$$\theta^{(1)} = \begin{cases} \theta' \text{ with probability } \alpha \\ \theta^{(0)} \text{ with probability } 1-\alpha \end{cases} \qquad \text{where } \alpha = \min\left(1, \frac{\pi(\theta')}{\pi(\theta^{(0)})}\right)$$

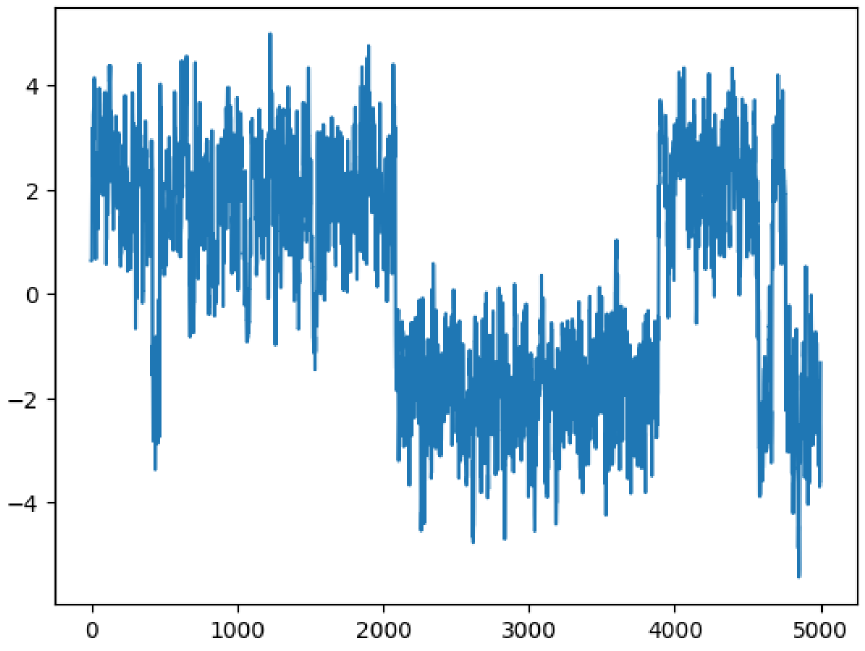
# Sampling 1D Gaussian




We can calculate any function of the posterior by summing over the trace

# Sampling 2D Gaussian



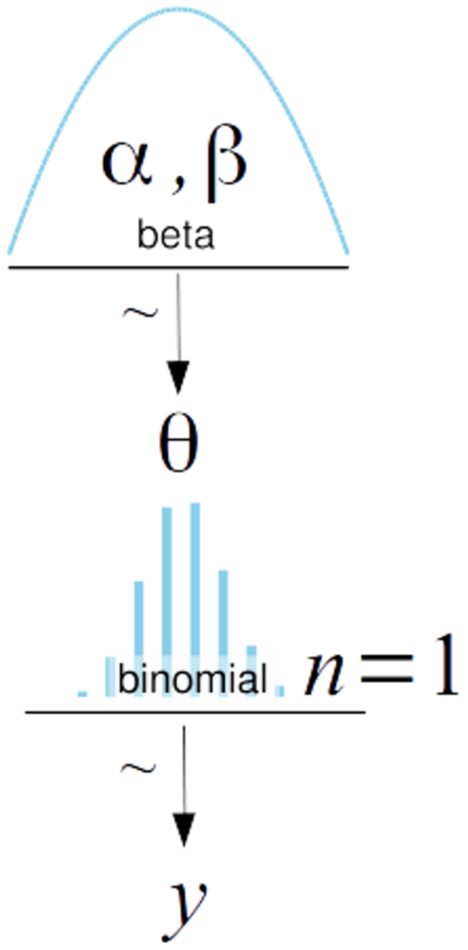

## MCMC Solvers:

- BUGS – Bayes Using Gibbs Sampling
- JAGS – Just Another Gibbs Sampler
- Stan – uses Hamiltonian Monte Carlo
- NUTS – No U-Turn Sampler

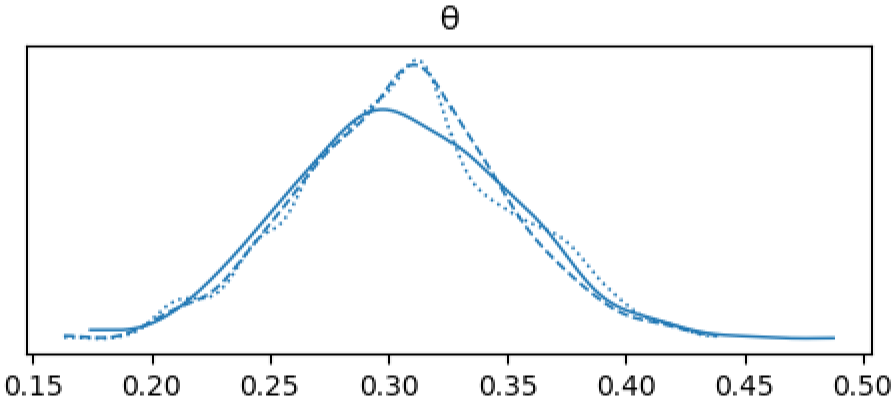
# Let’s toss a coin!

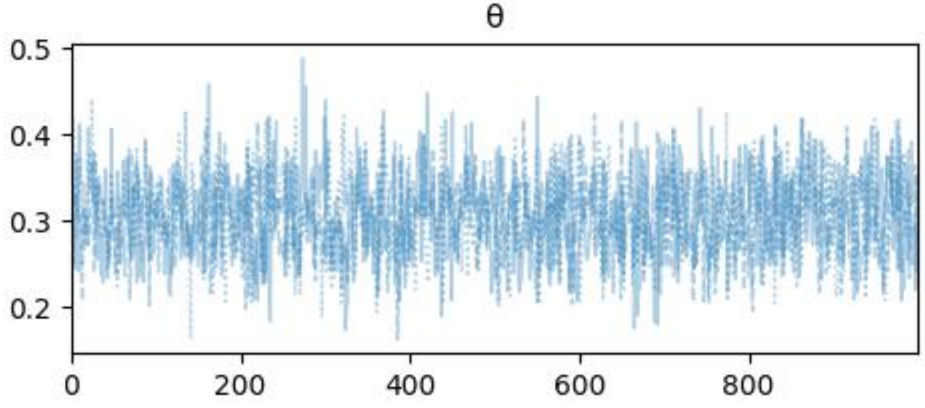
Colab: 19_PYMC_BayesianEZ.ipynb

# Let's toss a coin (with a PYMC)!




```
with pm.Model() as our_first_model:
    θ = pm.Beta('θ', alpha=1., beta=1.)
    y = pm.Bernoulli('y', p=θ, observed=data)
    trace = pm.sample(1000, random_seed=123, chains = 3)
```

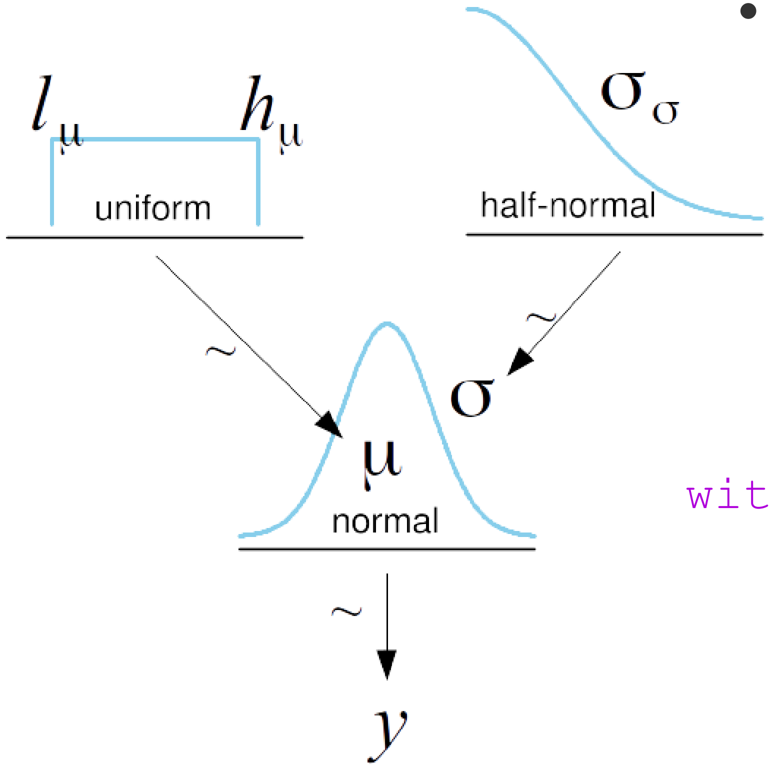



Colab: 19_PYMC_BayesianEZ.ipynb

# But how do we get physics done?

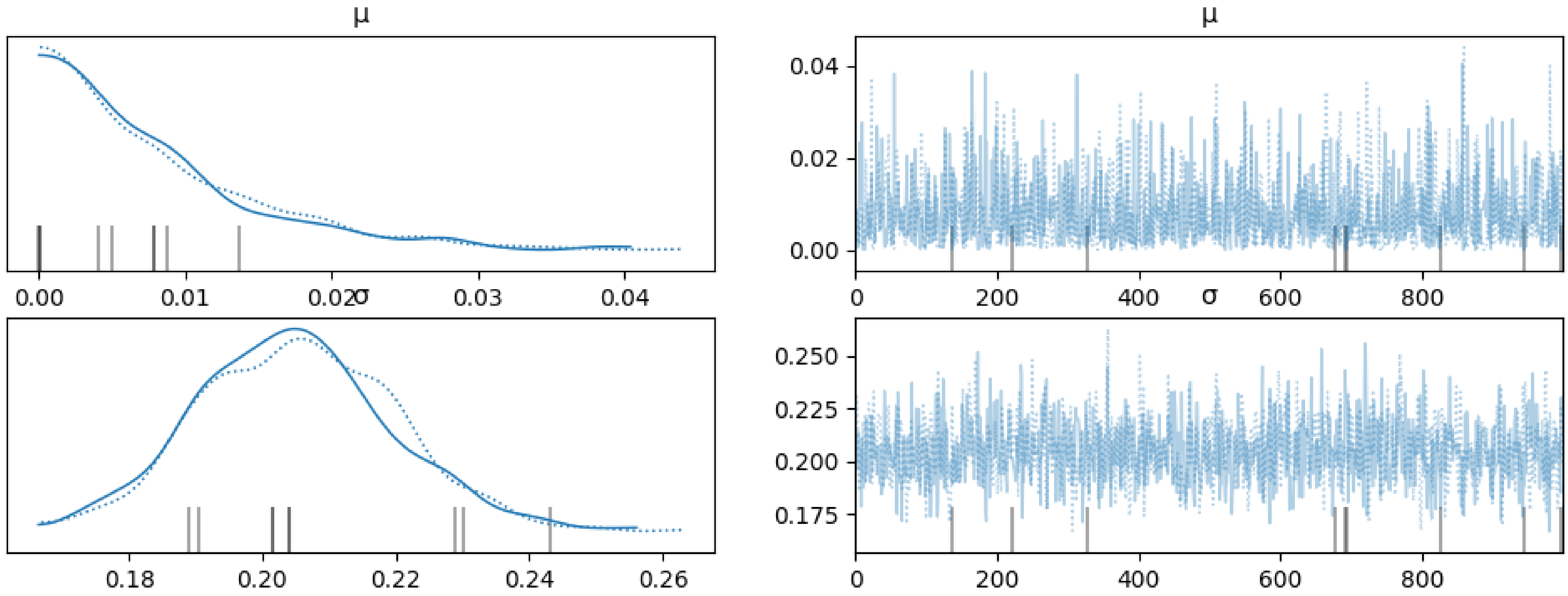
- Imagine that we have some observational data
- Can we say which distribution has it come from?




**Define Bayesian model!**

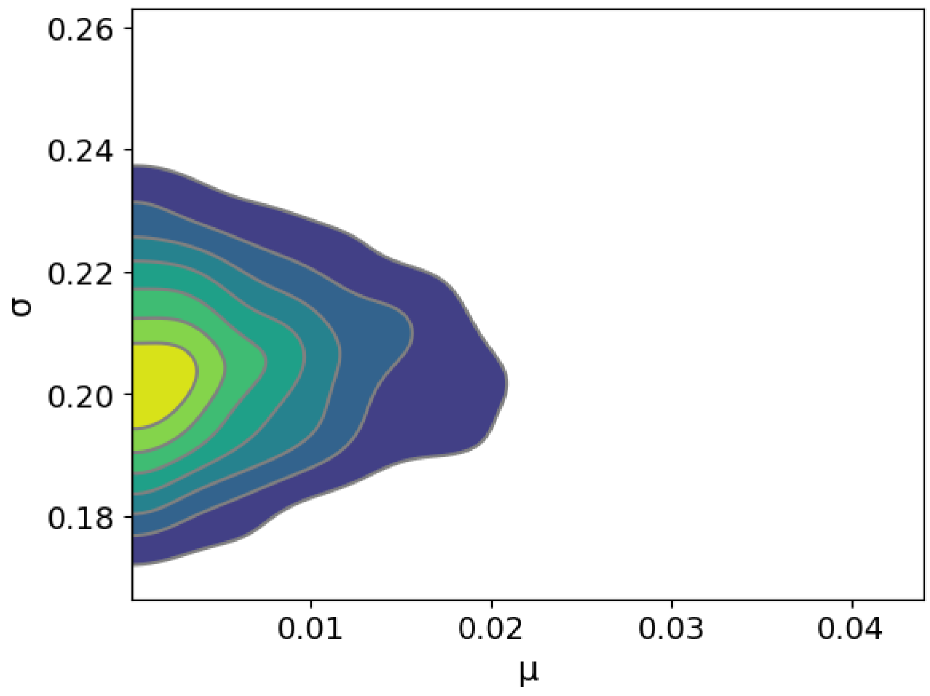
```
with pm.Model() as model_g:
    μ = pm.Uniform('μ', lower=0, upper=2)
    σ = pm.HalfNormal('σ', sigma=1)
    y = pm.Normal('y', mu=μ, sigma=σ, observed=arr)
    idata_g = pm.sample(1000)
```

# Presto!

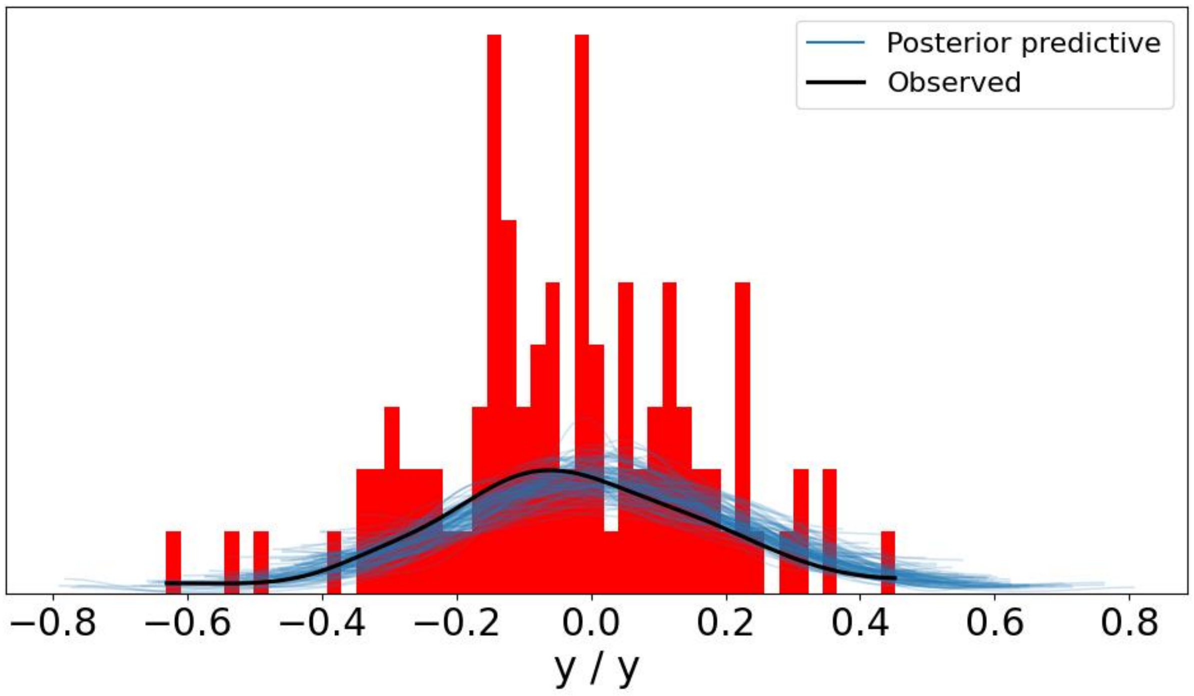


μ
0.00
0.01
0.02
0.03
0.04
σ
0.18
0.20
0.22
0.24
0.26
μ
0.04
0.02
0.00
0
200
400
600
800
σ
0.250
0.225
0.200
0.175
0
200
400
600
800

# Presto!



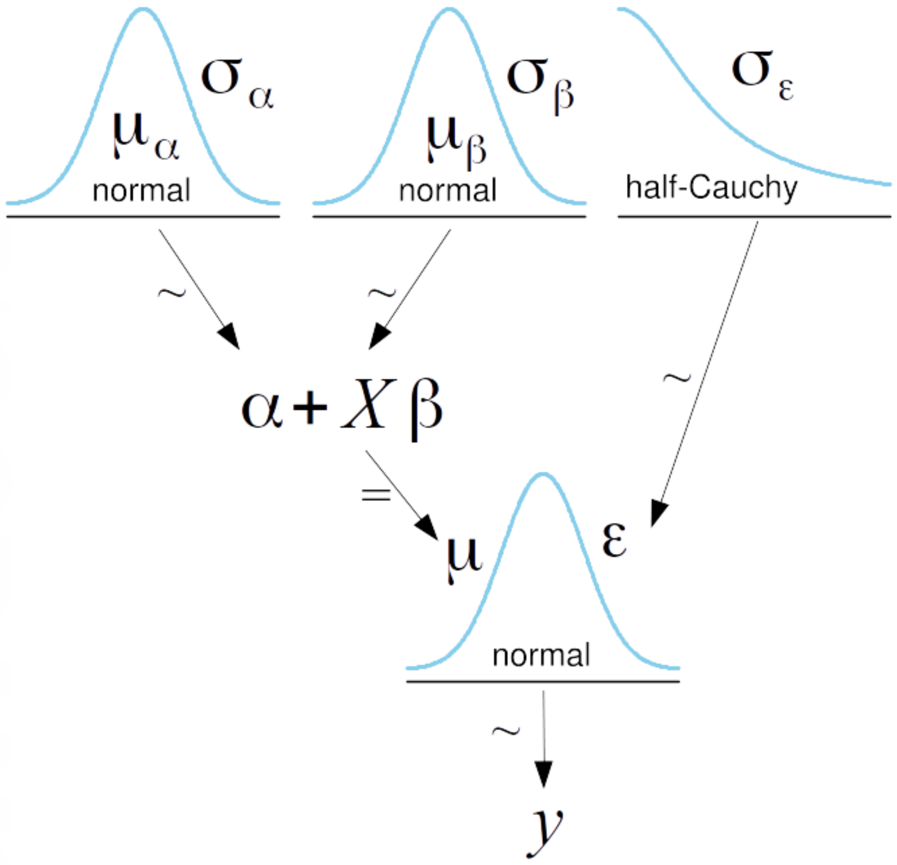
0.26
0.24
0.22
0.20
0.18
σ
0.01
0.02
0.03
0.04
μ

Posterior predictive
Observed
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
y / y

# Linear regression Bayesian style

- Imagine that we have some observational data
- Can we fit it by linear function?



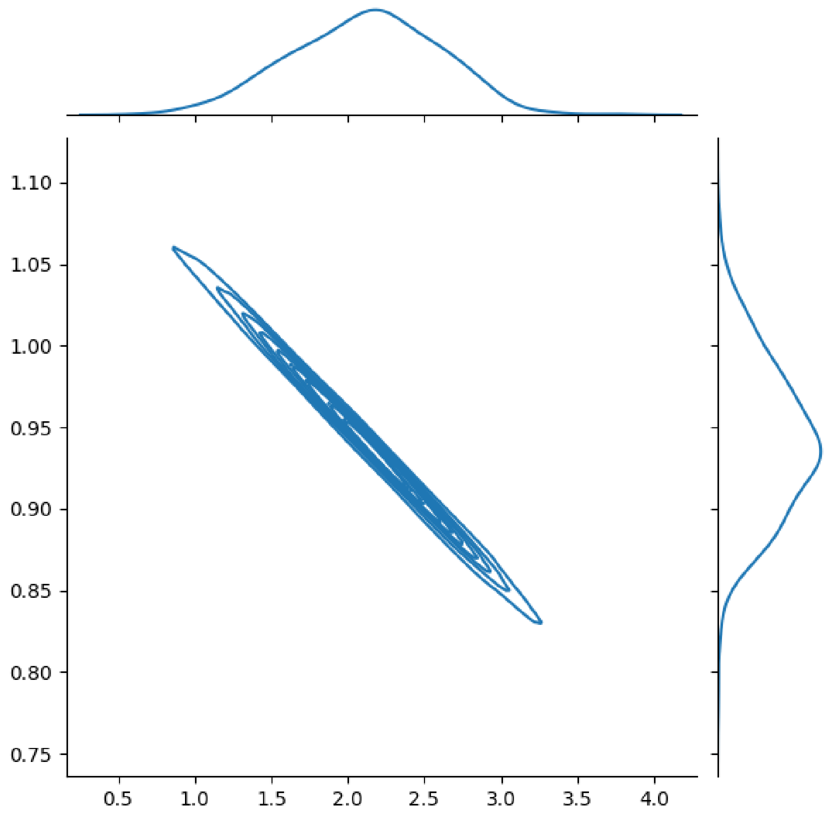

**Define Bayesian model!**

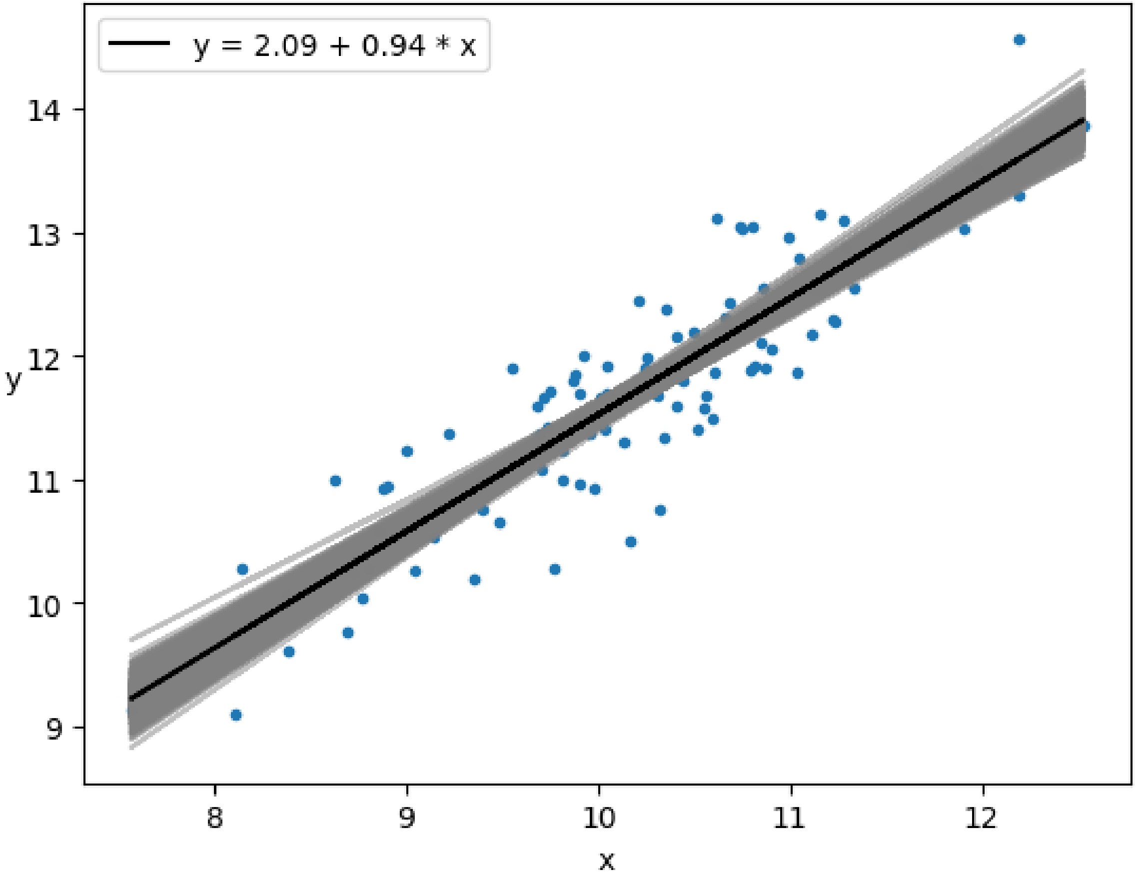
```
with pm.Model() as model_g:
    α = pm.Normal('α', mu=0, sigma=10)
    β = pm.Normal('β', mu=0, sigma=1)
    ϵ = pm.HalfCauchy('ϵ', 5)

    μ = pm.Deterministic('μ', α + β * x)
    y_pred = pm.Normal('y_pred', mu=μ,
                       sigma=ϵ, observed=y)

    idata_g = pm.sample(2000, tune=2000)
```

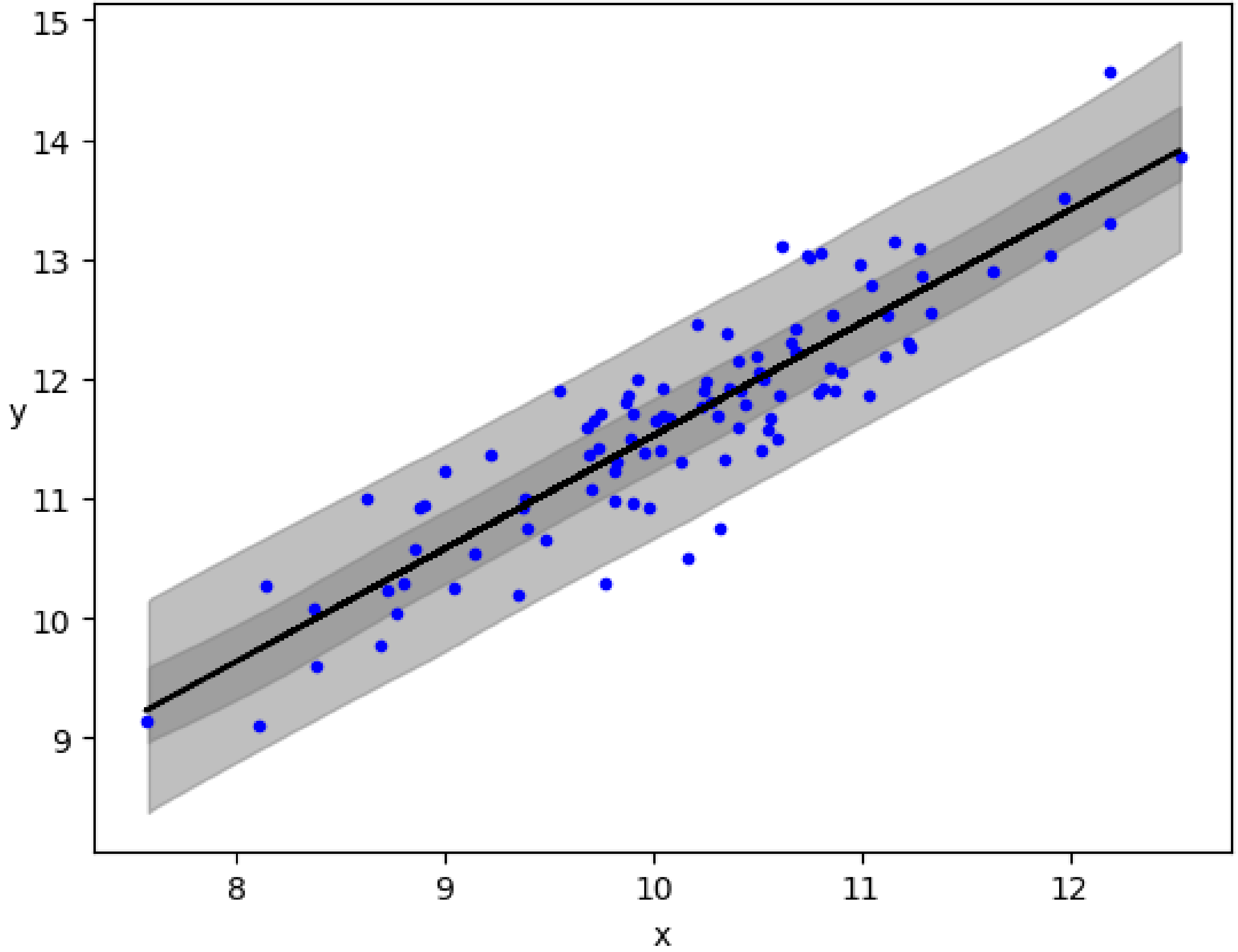
# The outputs?



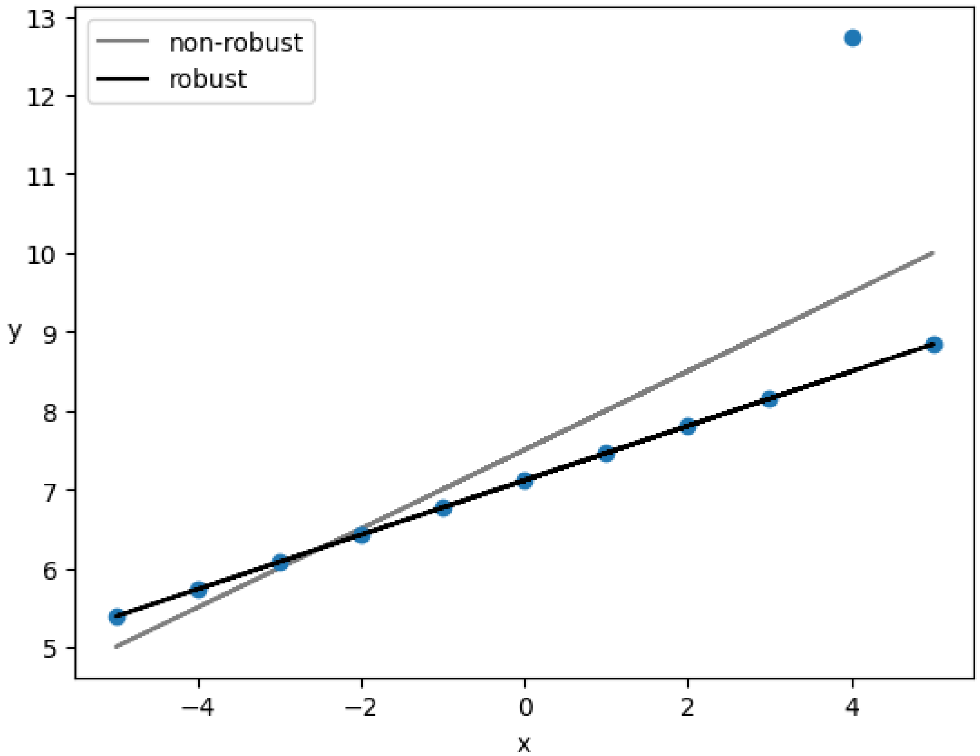

- The output of the Bayesian model will be draws for the slope and offset: families of lines consistent with data
- We can analyze their joint distribution
- And plot these lines

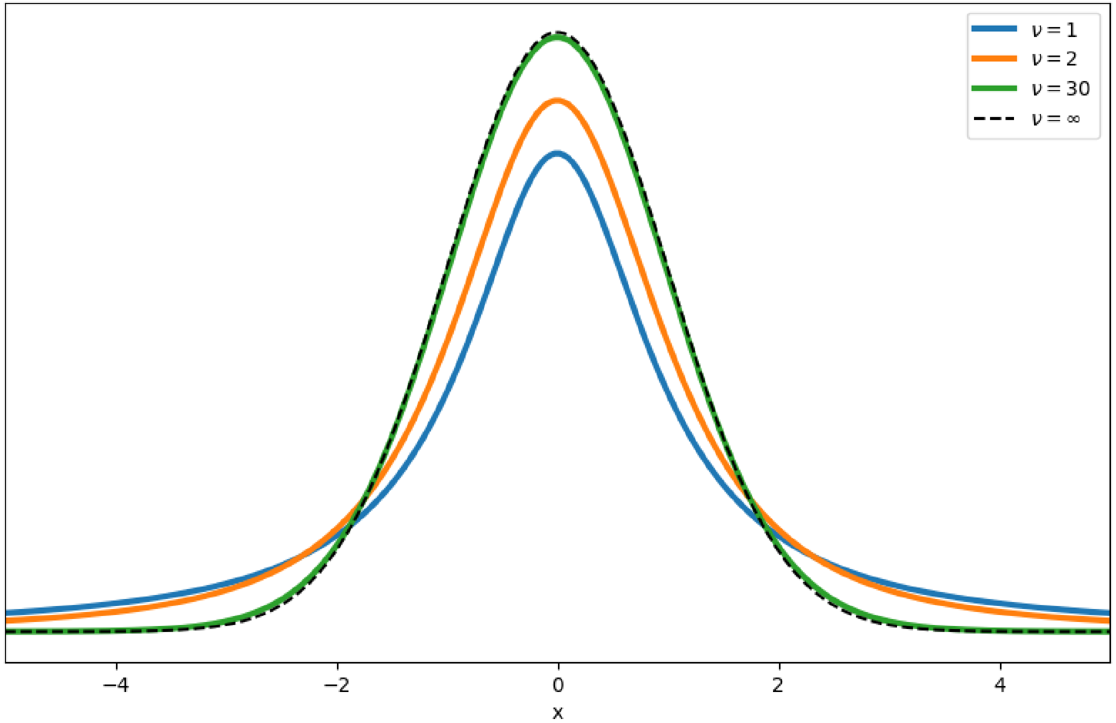
# How do we decide if the "data is reasonable"?




- That's what the posterior predictive checks are for!

# Bayesian methods for “impossible” problems




```
with pm.Model() as model_t:
    α = pm.Normal('α', mu=y_3.mean(), sigma=1)
    β = pm.Normal('β', mu=0, sigma=1)
    ϵ = pm.HalfNormal('ϵ', 5)
    ν_ = pm.Exponential('ν_', 1/29)
    ν = pm.Deterministic('ν', ν_ + 1)
    y_pred = pm.StudentT('y_pred', mu=α + β * x_3,sigma=ϵ, nu=ν, observed=y_3)
    idata_t = pm.sample(2000)
```

# Bayesian methods for “impossible” problems