Bayes Recap

Frequentist Stats

- parameters are fixed, data is stochastic
- true parameter θ^* characterizes population
- we estimate $\hat{\theta}$ on sample
- we can use MLE $\theta_{ML} = \underset{\theta}{\operatorname{argmax}} \mathcal{L}$
- we obtain sampling distributions (using bootstrap)

Bayesian Stats

- assume sample IS the data, no stochasticity
- parameters θ are stochastic random variables
- associate the parameter θ with a prior distribution $p(\theta)$
- The prior distribution generally represents our belief on the parameter values when we have not observed any data yet (to be qualified later)

Posterior distribution

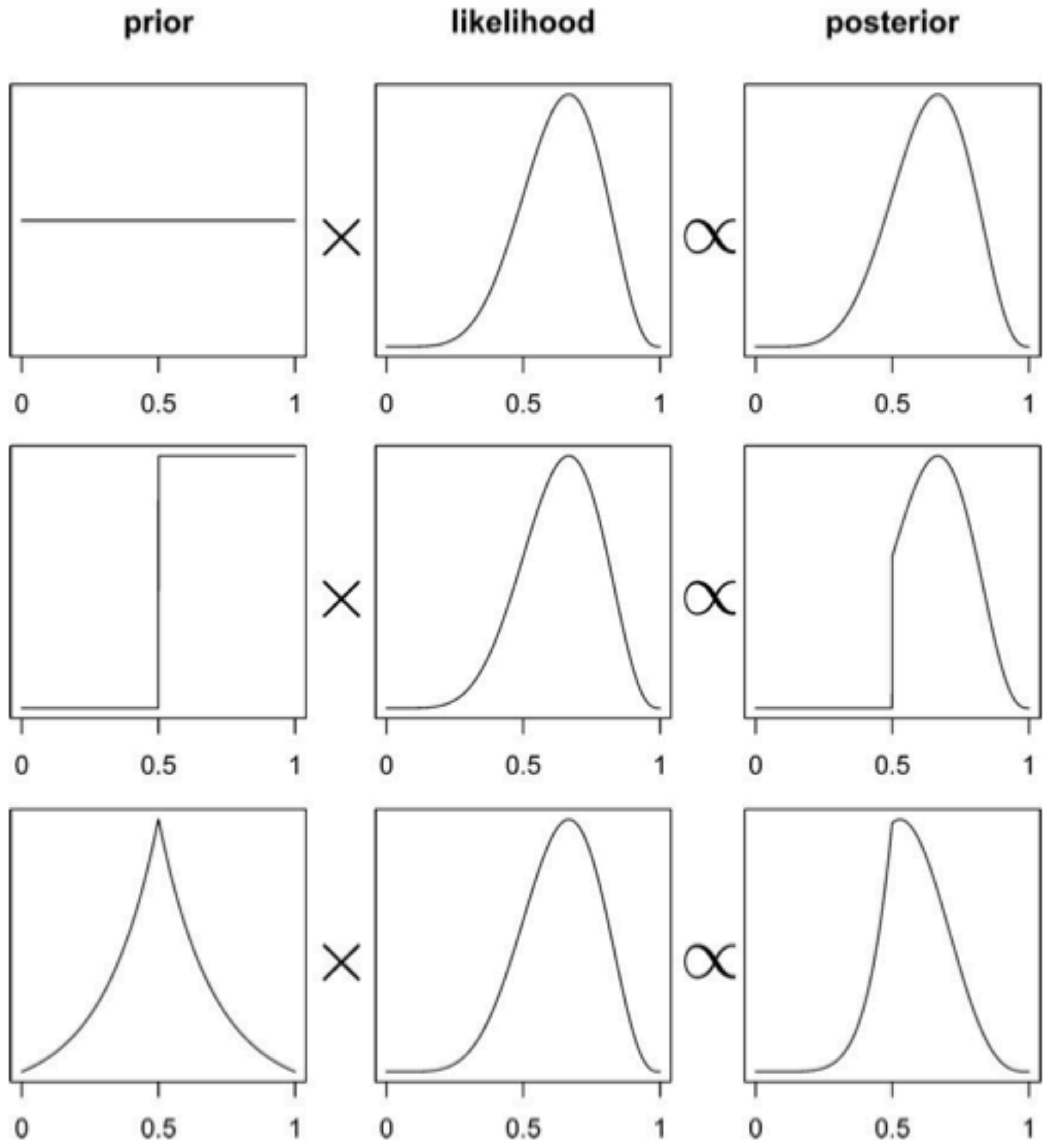
$$p(\theta|y) = \frac{p(y|\theta)\, p(\theta)}{p(y)}$$

with the **evidence** or **prior predicive distribution** $p(D)$ or $p(y)$ the expected likelihood (on existing data points) over the prior $E_{p(\theta)}[\mathcal{L}]$:

$$p(y) = \int d\theta p(y|\theta)p(\theta).$$

- $posterior = \frac{likelihood \times prior}{evidence}$
- evidence is just the normalization
- usually dont care about normalization (until model comparison), just samples
- What if θ is multidimensional? Marginal posterior:

$$p(\theta_1|D) = \int d\theta_{-1} p(\theta|D).$$



Posterior Predictive for predictions

The distribution of a future data point y^*:

$$p(y^*|D = \{y\}) = \int d\theta p(y^*|\theta)p(\theta|\{y\}).$$

Expectation of the likelihood at a new point(s) over the posterior $E_{p(\theta|D)}[p(y|\theta)]$.

(the expectation over the prior is the prior predictive or evidence)

Summary via MAP (a point estimate)

$$\begin{aligned}\theta_{\text{MAP}} &= \arg\max_{\theta} \, p(\theta|D) \\ &= \arg\max_{\theta} \frac{\mathcal{L}\, p(\theta)}{p(D)} \\ &= \arg\max_{\theta} \, \mathcal{L}\, p(\theta)\end{aligned}$$

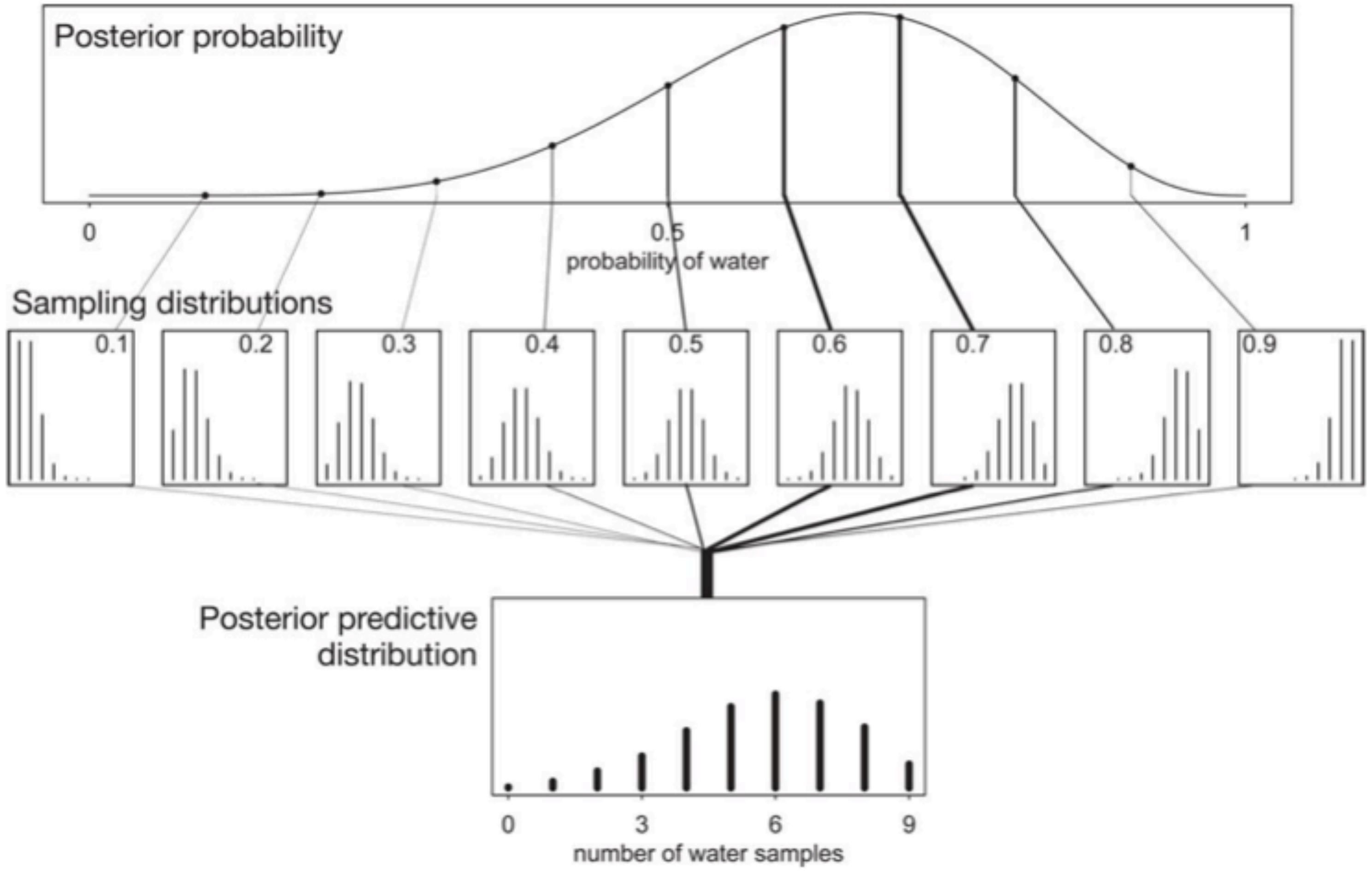
Plug-in Approximation: $p(\theta|y) = \delta(\theta - \theta_{MAP})$ and then draw

$p(y^*|y) = p(y^*|\theta_{MAP})$ a sampling distribution.

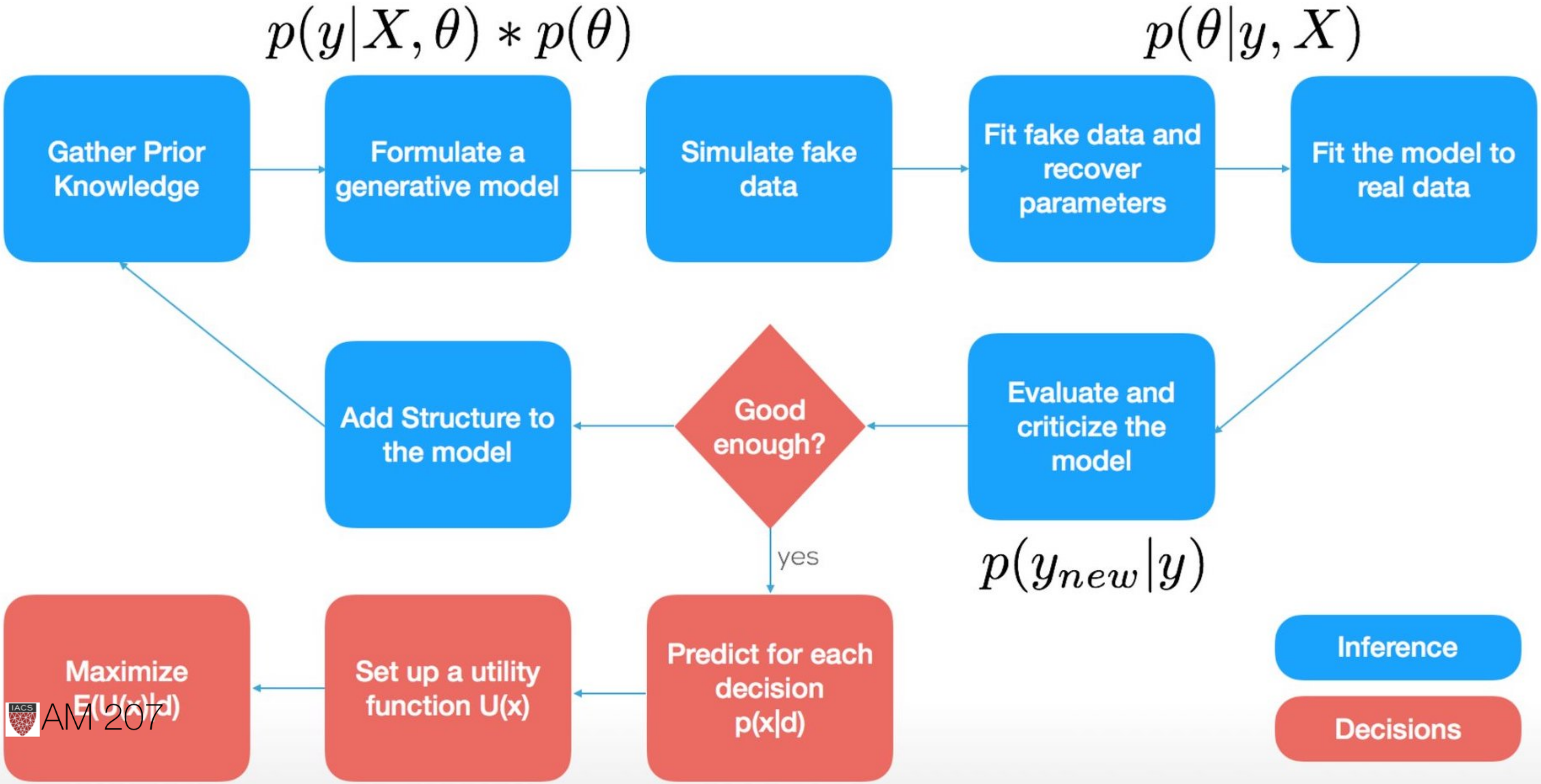
Posterior predictive from sampling

- first draw the thetas from the posterior
- then draw y's from the likelihood
- and histogram the likelihood
- these are draws from joint y, θ

Posterior predictive Idea



Bayesian Workflow
(from @ericnovik)



Conjugate Prior

- A **conjugate prior** is one which, when multiplied with an appropriate likelihood, gives a posterior with the same functional form as the prior.
- Likelihoods in the exponential family have conjugate priors in the same family
- analytical tractability AND interpretability

Coin Toss Model

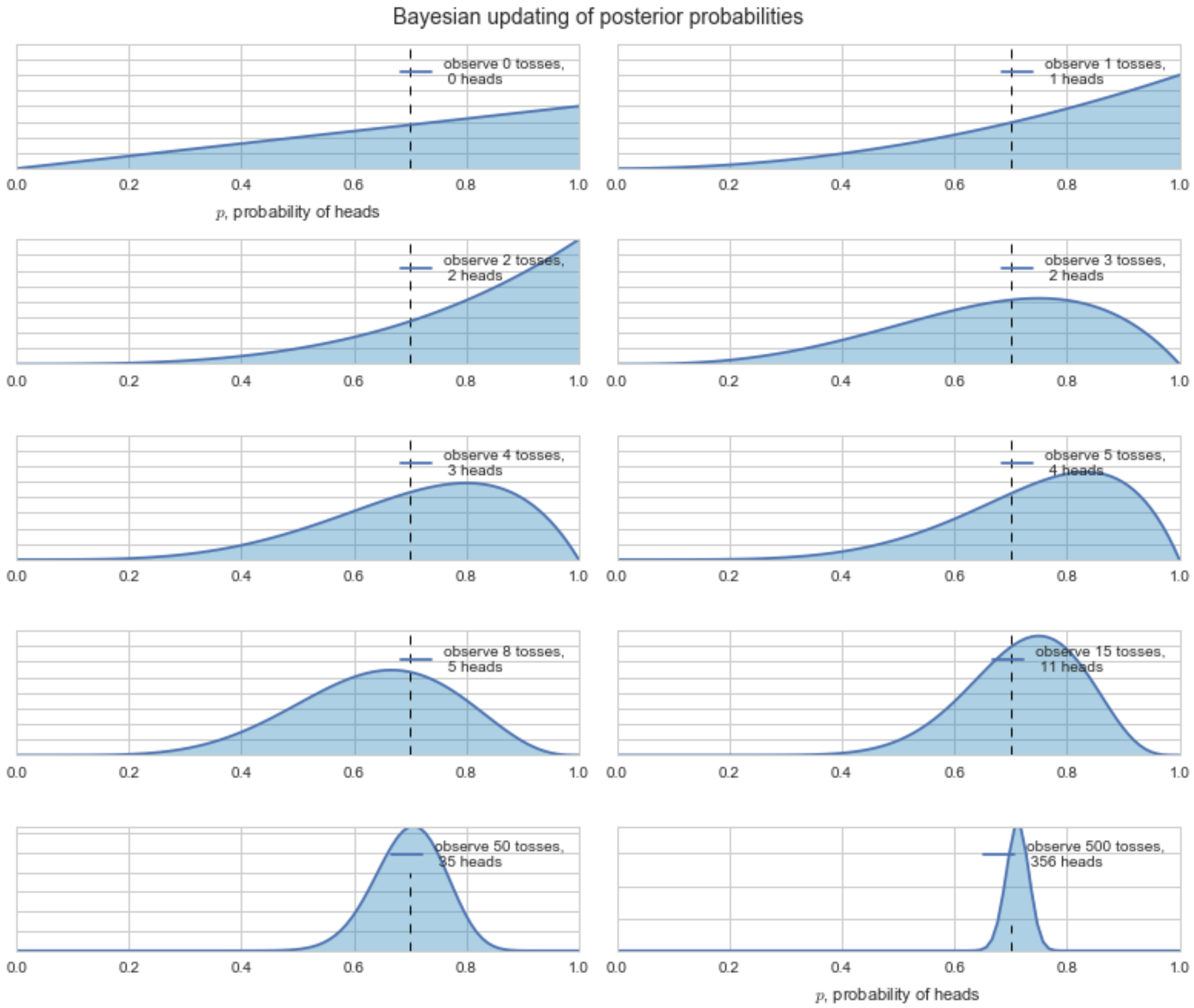
- Coin tosses are modeled using the Binomial Distribution, which is the distribution of a set of Bernoulli random variables.
- The Beta distribution is conjugate to the Binomial distribution

$$p(p|y) \propto p(y|p)P(p) = Binom(n, y, p) \times Beta(\alpha, \beta)$$

Because of the conjugacy, this turns out to be:

$$Beta(y + \alpha, n - y + \beta)$$

- think of a prior as a regularizer.
- a $Beta(1, 1)$ prior is equivalent to a uniform distribution.
- This is an **uninformative prior**. Here the prior adds one heads and one tails to the actual data, providing some "towards-center" regularization
- especially useful where in a few tosses you got all heads, clearly at odds with your beliefs.
- a $Beta(2, 1)$ prior would bias you to more heads (water in globe toss).


Bayesian updating of posterior probabilities
observe 0 tosses, 0 heads
observe 1 tosses, 1 heads
observe 2 tosses, 2 heads
observe 3 tosses, 2 heads
observe 4 tosses, 3 heads
observe 5 tosses, 4 heads
observe 8 tosses, 5 heads
observe 15 tosses, 11 heads
observe 50 tosses, 35 heads
observe 500 tosses, 356 heads
0.0
0.2
0.4
0.6
0.8
1.0
p, probability of heads

Bayesian Updating "on-line"

- as each piece of data comes in, you update the prior by multiplying by the one-point likelihood.

- the posterior you get becomes the prior for our next step

$$p(\theta \mid \{y_1, \ldots, y_{n+1}\}) \propto p(\{y_1, \ldots, y_n\} \mid \theta) \times p(\theta \mid \{y_1, \ldots, y_n\})$$

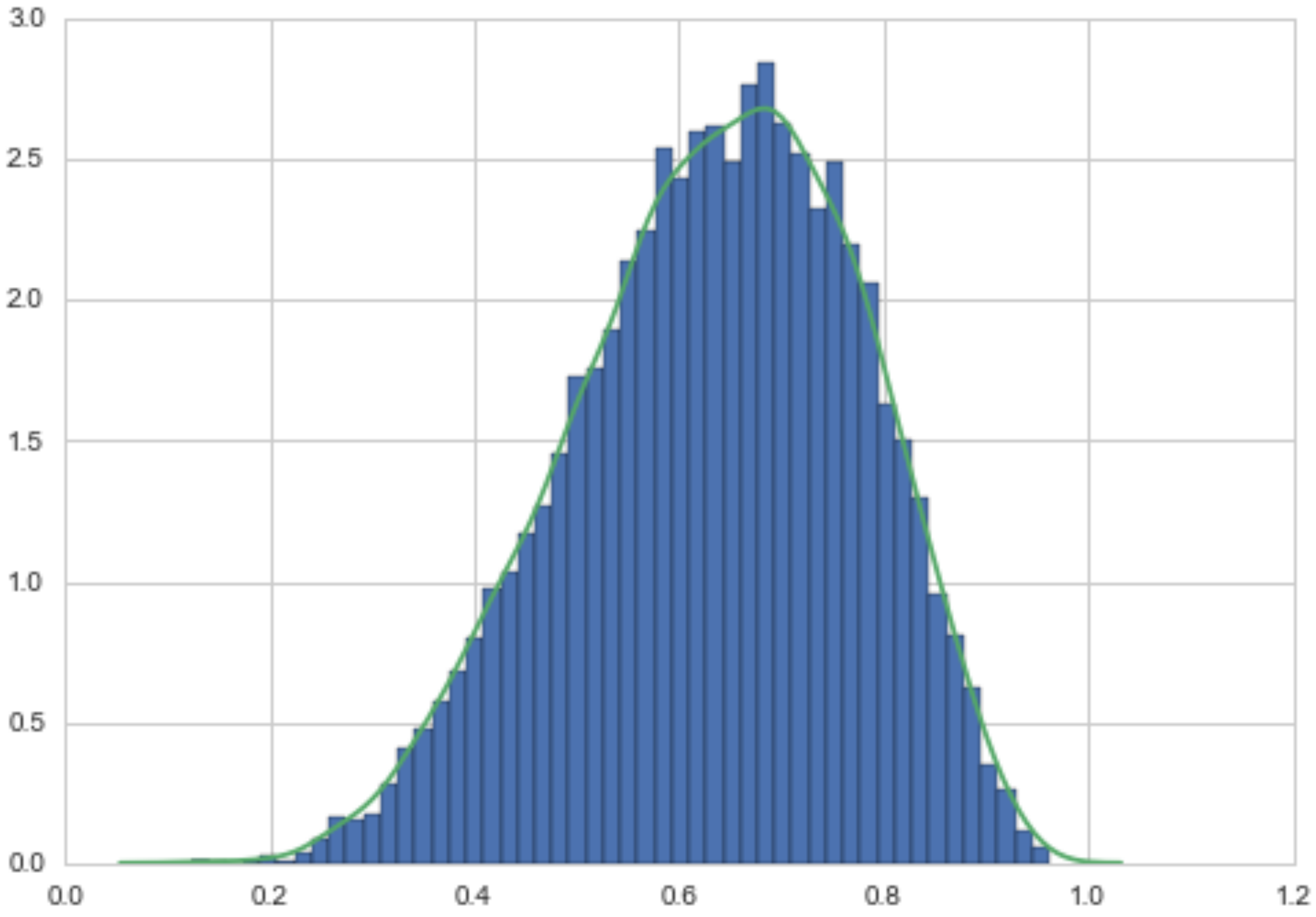
- the posterior predictive is the distribution of the next data point!

$$p(y_{n+1} | \{y_1, \ldots y_n\}) = E_{p(\theta|\{y_1, \ldots y_n\})}[p(y_{n+1}|\theta)] = \int d\theta p(y_{n+1}|\theta) p(\theta|\{y_1, \ldots y_n\})$$

Beta-Binomial all at once

- Seal tosses globe, θ is true water fraction
- The Beta distribution is conjugate to the Binomial distribution $p(\theta|y) \propto p(y|\theta)P(\theta) = Binom(n, y, \theta) \times Beta(\alpha, \beta)$
- Because of the conjugacy, this turns out to be: $Beta(y + \alpha, n - y + \beta)$
- a $Beta(1, 1)$ prior is equivalent to a uniform distribution.

Posterior



- The probability that the amount of water is less than 50%: `np.mean(samples < 0.5) = 0.173`

- Credible Interval: amount of probability mass. `np.percentile(samples, [10, 90]) = [ 0.44604094, 0.81516349]`

- `np.mean(samples), np.median(samples) = (0.63787343440335842, 0.6473143052303143)`

MAP, a point estimate

$$
\begin{aligned}
\theta_{\text{MAP}} &= \arg\max_{\theta} \, p(\theta|D) \\
&= \arg\max_{\theta} \frac{\mathcal{L} \, p(\theta)}{p(D)} \\
&= \arg\max_{\theta} \, \mathcal{L} \, p(\theta)
\end{aligned}
$$

```
sampleshisto = np.histogram(samples, bins=50)
maxcountindex = np.argmax(sampleshisto[0])
mapvalue = sampleshisto[1][maxcountindex]
print(maxcountindex, mapvalue)
```

```
31 0.662578641304
```

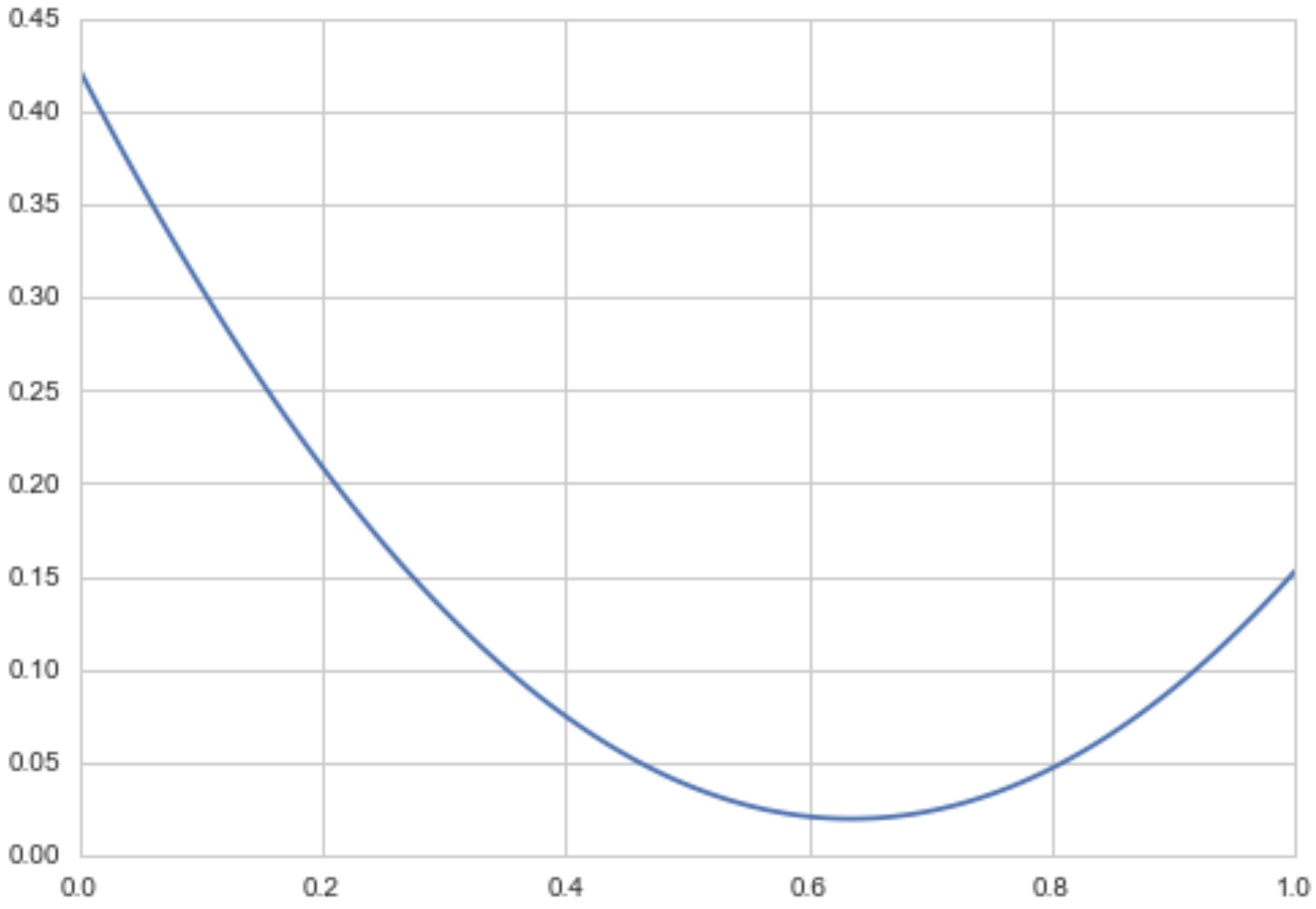
Posterior Mean minimizes squared loss

$$R(t) = E_{p(\theta|D)}[(\theta - t)^2] = \int d\theta (\theta - t)^2 p(\theta|D)$$

$$\frac{dR(t)}{dt} = 0 \implies t = \int d\theta \theta \, p(\theta|D)$$

```
mse = [np.mean((xi-samples)**2) for xi in x]
plt.plot(x, mse);
```

This is **Decision Theory**.



Posterior predictive

$$p(y^*|D) = \int d\theta p(y^*|\theta)p(\theta|D)$$

Risk Minimization holds here too: $y_{minmse} = \int dy\, y\, p(y|D)$

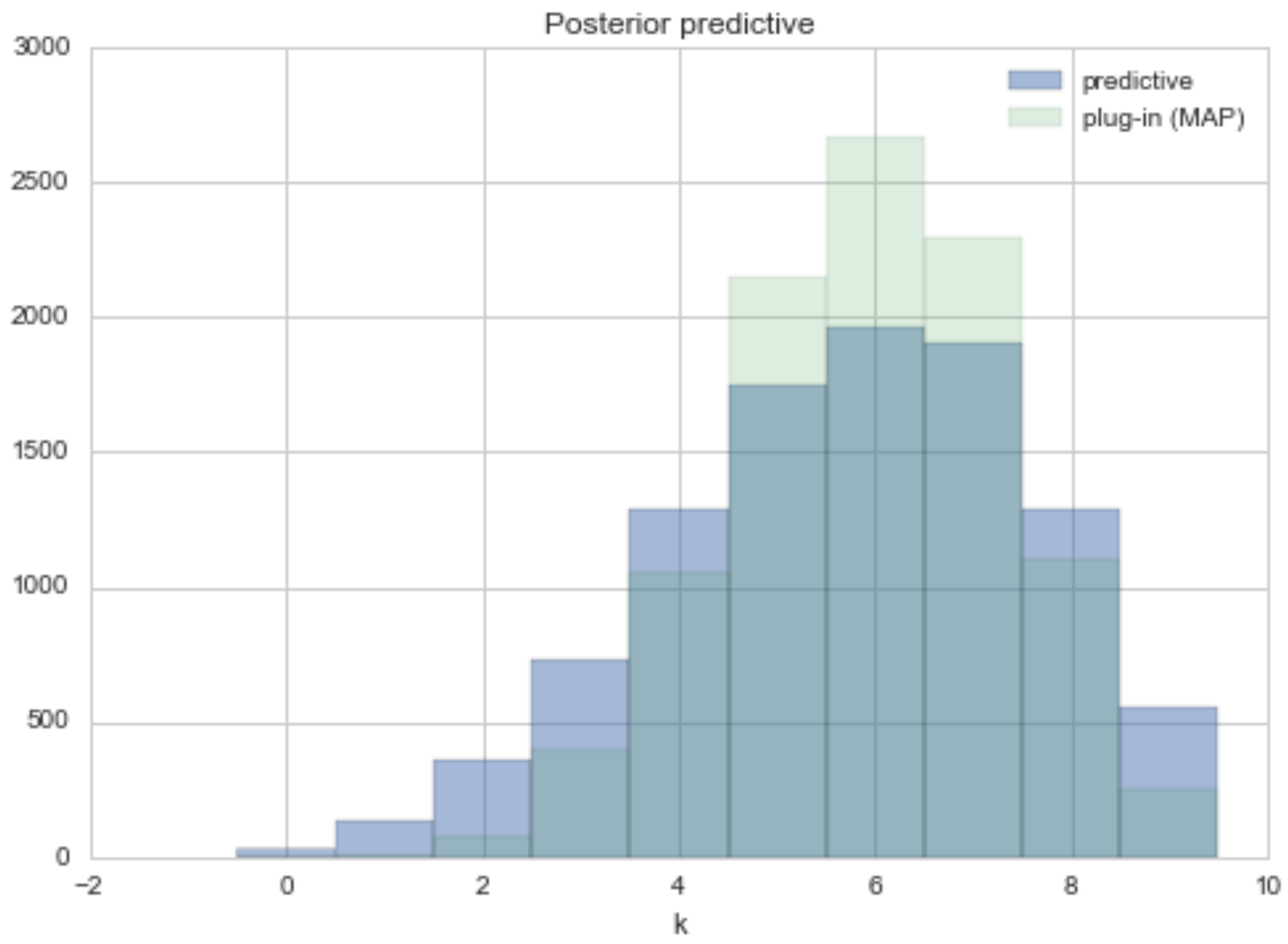
Plug-in Approximation: $p(\theta|D) = \delta(\theta - \theta_{MAP})$ and then draw

$p(y^*|D) = p(y^*|\theta_{MAP})$ a sampling distribution.

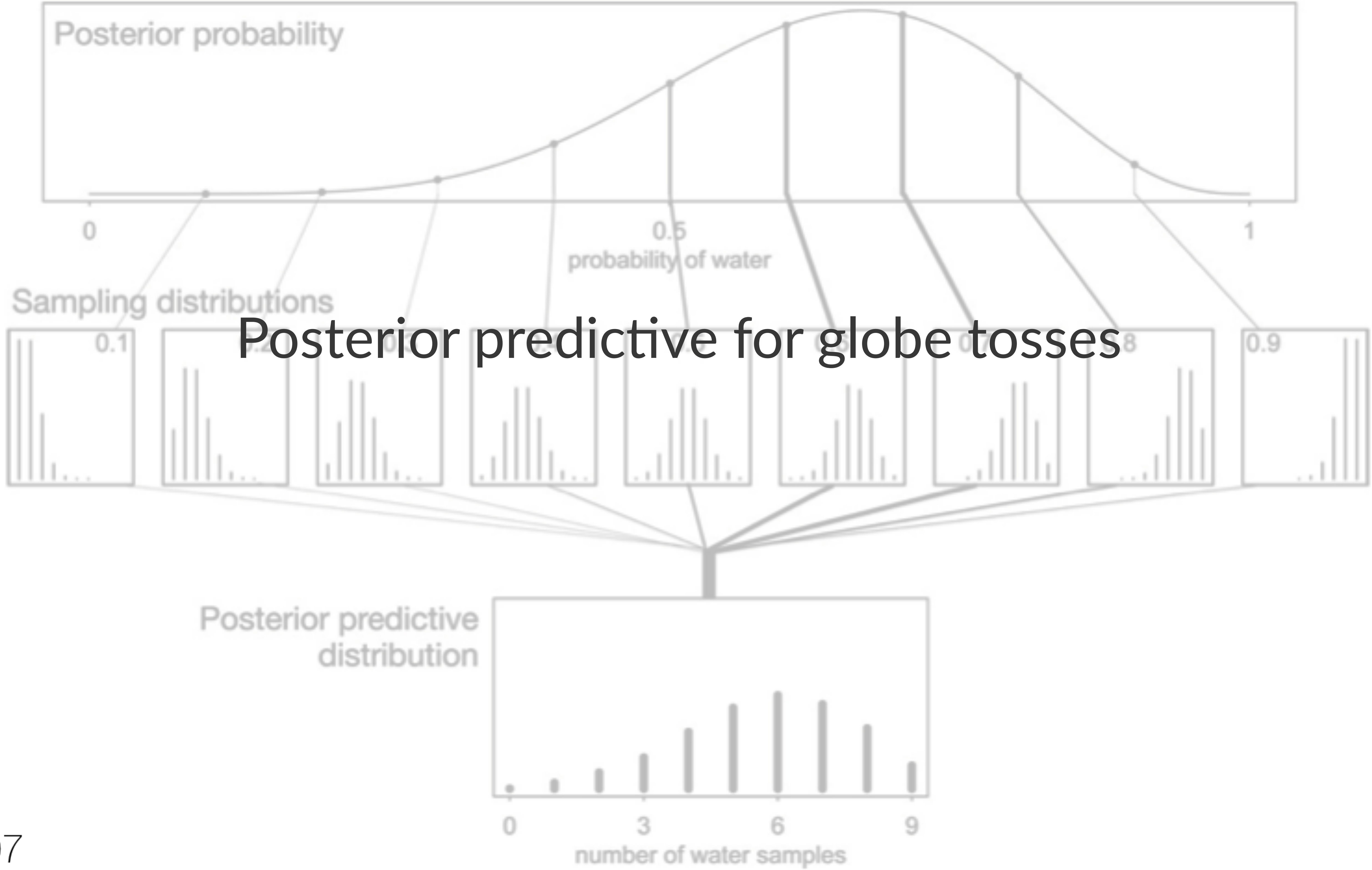
Posterior predictive from sampling

- first draw the thetas from the posterior
- then draw y's from the likelihood
- and histogram the likelihood
- these are draws from joint y, θ

```
postpred = np.random.binomial( len(data), samples);
```



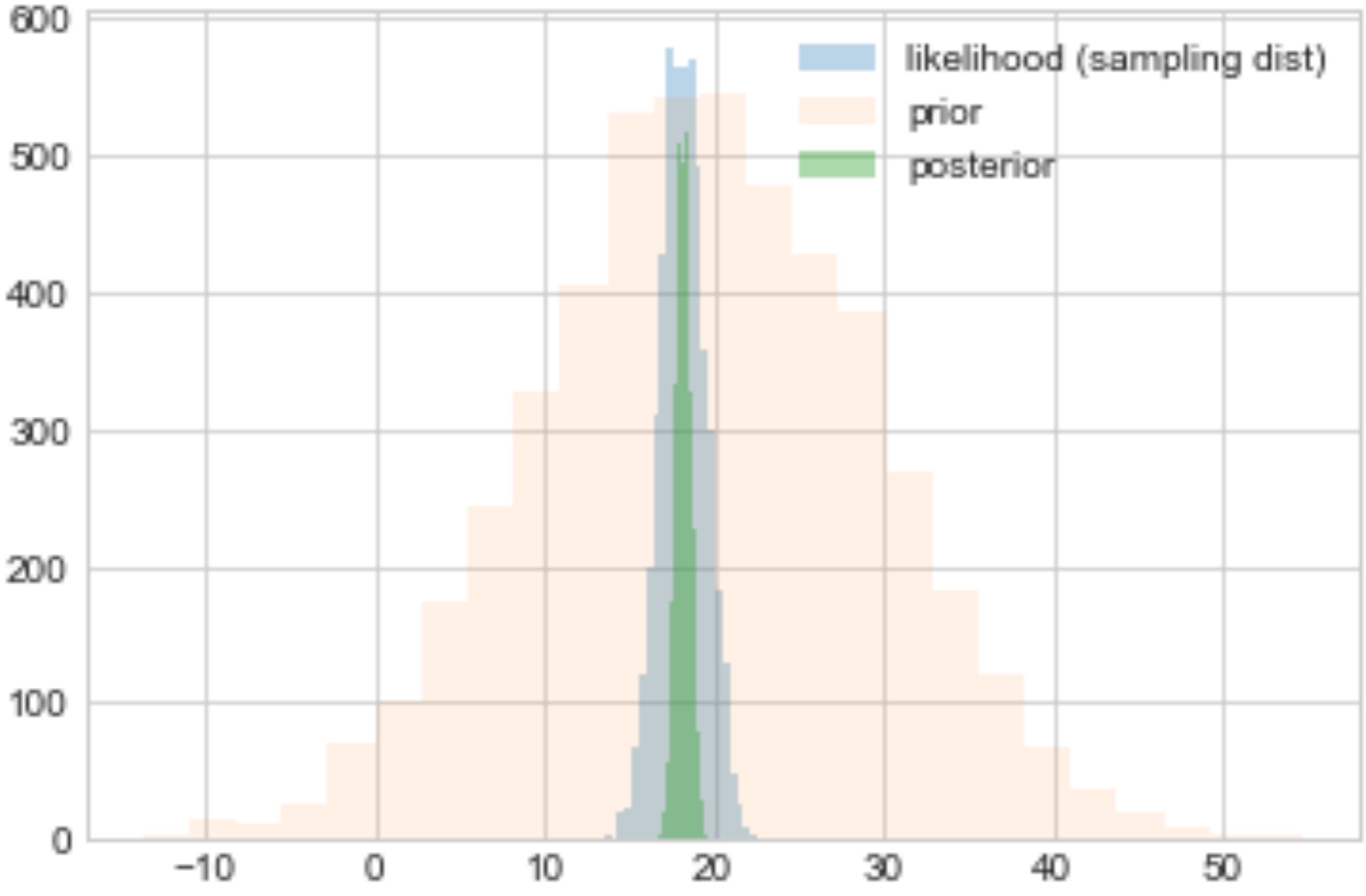
Posterior predictive for globe tosses



Normal-Normal Model

$$p(\mu, \sigma^2) = p(\mu|\sigma^2)p(\sigma^2)$$

- **fixed σ prior**: $p(\sigma^2) = \delta(\sigma^2 - \sigma_0^2)$
- **non-fixed σ prior**: Choose a functional form that is mildly informative, e.g., normal, half cauchy, half normal
- **μ prior**: Mildly informative normal with prior mean and wide standard deviation



- fixed σ

```
logprior = lambda mu:
    norm.logpdf(mu, loc=mu_prior, scale=std_prior)
loglike = lambda mu:
    np.sum(norm.logpdf(Y, loc=mu, scale=np.std(Y)))
logpost = lambda mu:
    loglike(mu) + logprior(mu)
```

- non-fixed σ:

```
logprior = lambda mu, sigma:
    norm.logpdf(mu, loc=mu_prior, scale=std_prior) +
    norm.logpdf(sigma, loc=sig_data, scale=2)
loglike = lambda mu, sigma:
    np.sum(norm.logpdf(Y, loc=mu, scale=sigma))
logpost = lambda mu, sigma:
    loglike(mu, sigma) + logprior(mu, sigma)
```

Marginalization

Marginal posterior:

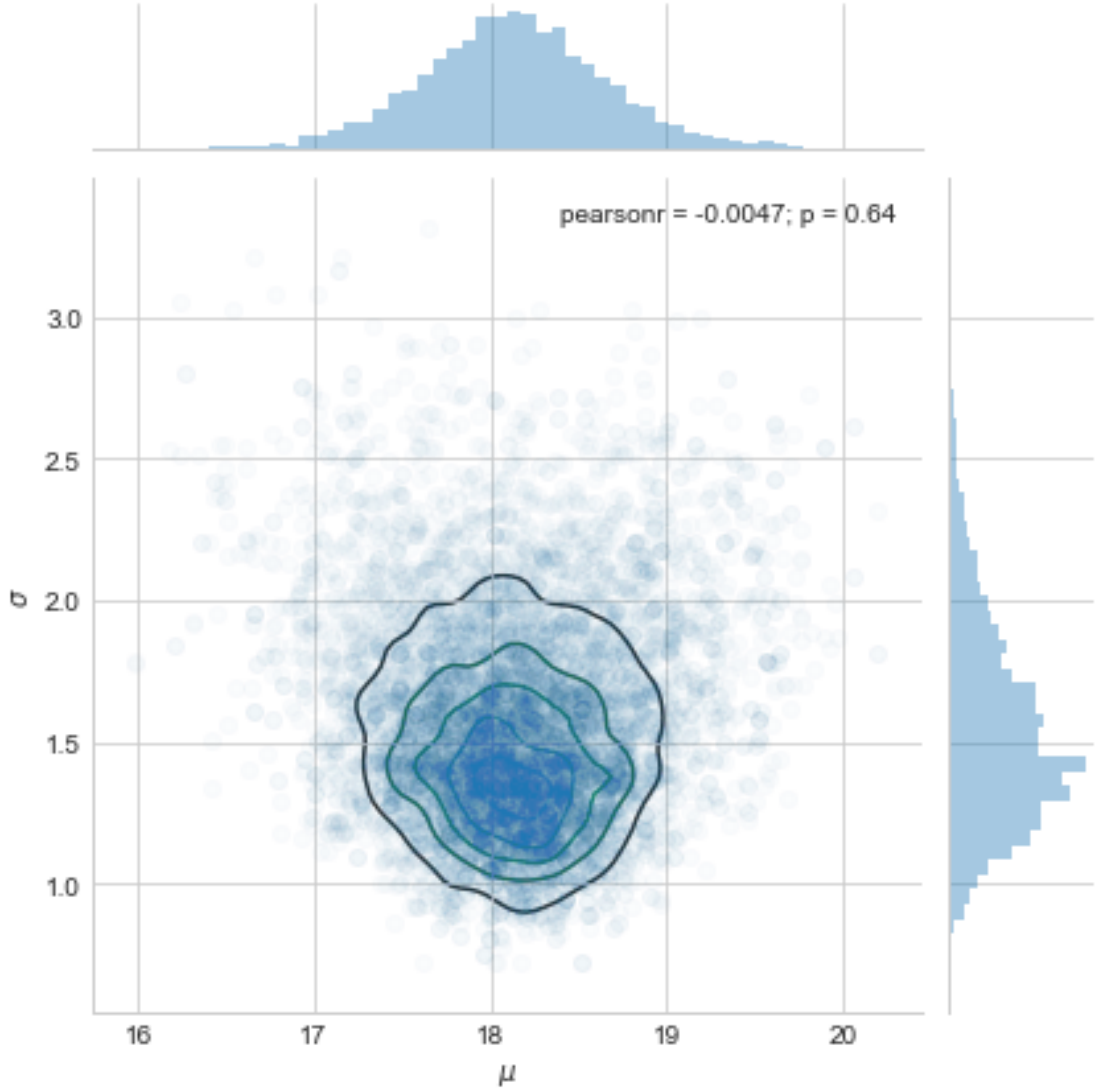
$$p(\theta_1|D) = \int d\theta_{-1} p(\theta|D).$$

```
samps[20000::,:].shape #(10001, 2)
```

```
sns.jointplot(
    pd.Series(samps[20000::,0], name="$\mu$"),
    pd.Series(samps[20000::,1], name="$\sigma$"),
    alpha=0.02)
    .plot_joint(
        sns.kdeplot,
    zorder=0, n_levels=6, alpha=1)
```

Marginals are just 1D histograms

```
plt.hist(samps[20000::,0])
```



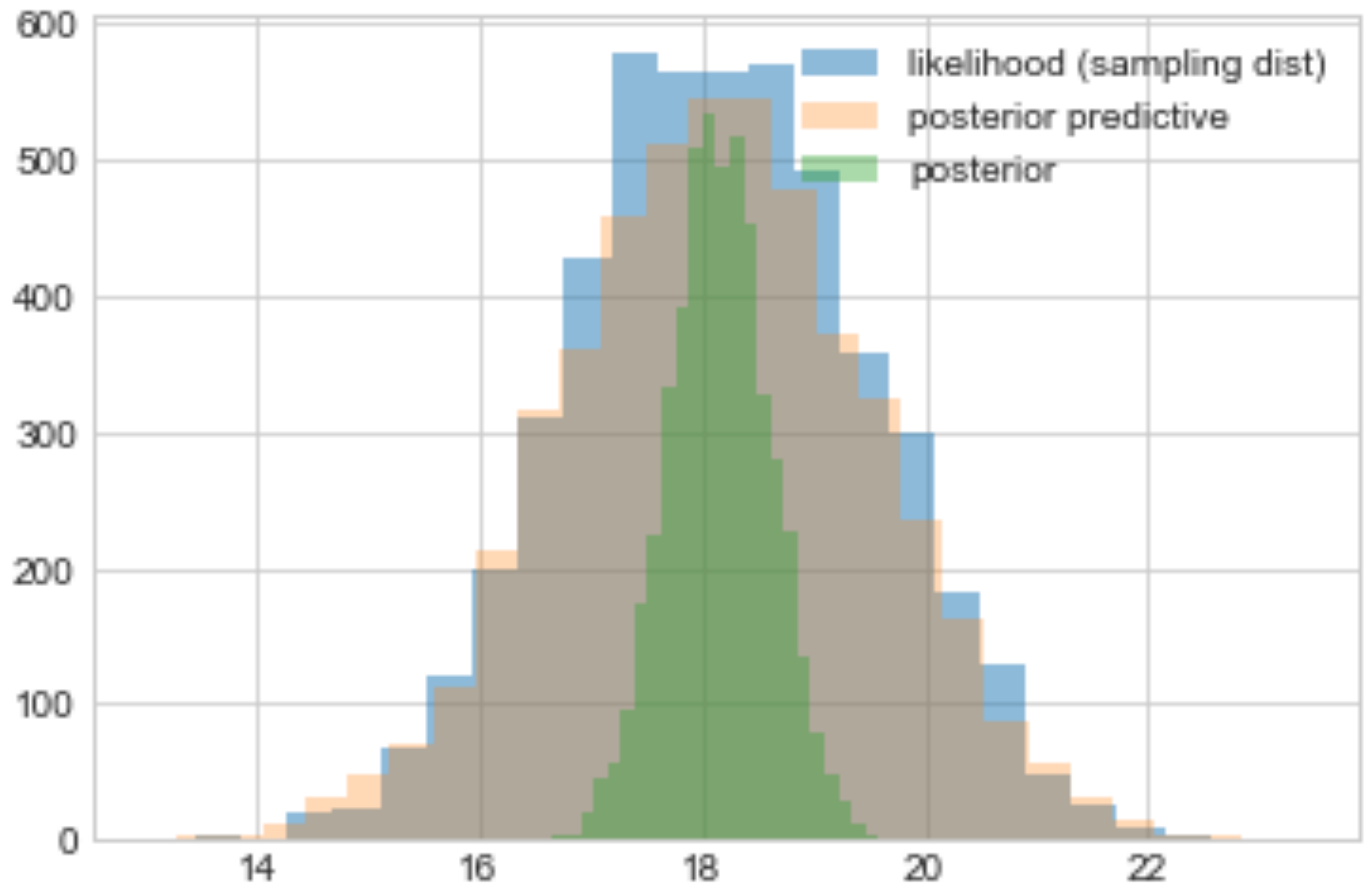
Posterior Predictive

The distribution of a future data point y^*:

$$p(y^*|D = \{y\}) = E_{p(\theta|D)}[p(y|\theta)]$$
$$= \int d\theta p(y^*|\theta)p(\theta|\{y\}).$$

First draw the thetas from the posterior, then draw y's from the likelihood (these are draws from joint y, θ)

```
post_pred_func = lambda post: norm.rvs(loc = post, scale = sig)
post_pred_samples = post_pred_func(post_samples)
```



Regularization in the Normal-Normal Model

Posterior for a gaussian likelihood:

$$p(\mu, \sigma^2 | y_1, \ldots, y_n, \sigma^2) \propto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum(y_i - \mu)^2} \, p(\mu, \sigma^2)$$

What is the posterior of μ assuming we know σ^2?

Prior for σ^2 is $p(\sigma^2) = \delta(\sigma^2 - \sigma_0^2)$

$$p(\mu|y_1,\ldots,y_n,\sigma^2=\sigma_0^2) \propto p(\mu|\sigma^2=\sigma_0^2)\, e^{-\frac{1}{2\sigma_0^2}\sum(y_i-\mu)^2}$$

The conjugate of the normal is the normal itself.

Say we have the prior

$$p(\mu|\sigma^2) = \exp\left\{-\frac{1}{2\tau^2}(\hat{\mu}-\mu)^2\right\}$$

posterior: $p(\mu|y_1,\ldots,y_n,\sigma^2) \propto \exp\left\{-\frac{a}{2}(\mu-b/a)^2\right\}$

Here

$$a = \frac{1}{\tau^2} + \frac{n}{\sigma_0^2}, \qquad b = \frac{\hat{\mu}}{\tau^2} + \frac{\sum y_i}{\sigma_0^2}$$

Define $\kappa = \sigma^2/\tau^2$

$$\mu_p = \frac{b}{a} = \frac{\kappa}{\kappa + n}\hat{\mu} + \frac{n}{\kappa + n}\bar{y}$$

which is a weighted average of prior mean and sampling mean.

The variance is

$$\tau_p^2 = \frac{1}{1/\tau^2 + n/\sigma^2}$$

or better

$$\frac{1}{\tau_p^2} = \frac{1}{\tau^2} + \frac{n}{\sigma^2}.$$

as n increases, the data dominates the prior and the posterior mean approaches the data mean, with the posterior distribution narrowing...

Posterior vs prior

```
Y = [16.4, 17.0, 17.2, 17.4, 18.2, 18.2, 18.2, 19.9, 20.8]
#Data Quantities
sig = np.std(Y) # assume that is the value of KNOWN sigma (in the likelihood)
mu_data = np.mean(Y)
n = len(Y)
# Prior mean
mu_prior = 19.5
# prior std
tau = 10
# plug in formulas
kappa = sig**2 / tau**2
sig_post =np.sqrt(1./( 1./tau**2 + n/sig**2));
# posterior mean
mu_post = kappa / (kappa + n) *mu_prior + n/(kappa+n)* mu_data
#samples
N = 15000
theta_prior = np.random.normal(loc=mu_prior, scale=tau, size=N);
theta_post = np.random.normal(loc=mu_post, scale=sig_post, size=N);
```