

# Summarizing the Posterior

Felipe José Bravo Márquez

October 5, 2021

# Summarizing the Posterior

- Once our Bayesian model produces a posterior distribution, it is necessary to summarize and interpret it.
- However, a posterior distribution is (usually) a high dimensional object that is hard to visualize and work with [Murphy, 2021].
- In this class we will learn how to draw estimates (e.g., point estimates, intervals, predictions) to summarize and interpret a posterior distribution.
- Exactly how it is summarized depends upon our purpose.
- Common questions include:
  - How much posterior probability lies below some parameter value?
  - How much posterior probability lies between two parameter values?
  - Which parameter value marks the lower 5% of the posterior probability?
  - Which range of parameter values contains 90% of the posterior probability?
  - Which parameter value has highest posterior probability?

# Sampling to summarize

- These questions can be usefully divided into questions about:
  - intervals of defined boundaries
  - intervals of defined probability area
  - point estimates
- In the theoretical world (when the posterior has a closed mathematical form), answering these questions implies calculating complicated integrals to cancel out (or average) different variables.
- In the practical world, however, the same results can be approximated using **samples** from the posterior.
- In this class we will approach the above questions using samples from the posterior.
- Another reason to learn to work with posterior samples is that methods like MCMC produce nothing but samples from the posterior.
- This class is based on Chapter 3 of [McElreath, 2020].

# Sampling from a grid-approximate posterior

- Before beginning to work with samples, we need to generate them.
- Here's a reminder for how to compute the posterior for the globe tossing model, using grid approximation:

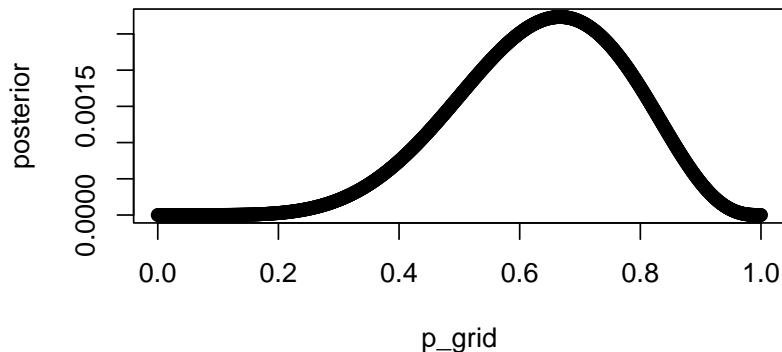
```
p_grid <- seq( from=0 , to=1 , length.out=1000 )  
prior <- rep( 1 , 1000 )  
likelihood <- dbinom( 6 , size=9 , prob=p_grid )  
posterior <- likelihood * prior  
posterior <- posterior / sum(posterior)
```

- Now we wish to draw 10,000 samples from this posterior.
- Imagine the posterior is a bucket full of parameter values, numbers such as 0.1, 0.7, 0.5, 1, etc.
- Within the bucket, each value exists in proportion to its posterior probability, such that values near the peak are much more common than those in the tails.

# Sampling from a grid-approximate posterior

- We can visualize this bucket by plotting the posterior probabilities against the grid of parameter values.

```
plot(posterior~p_grid)
```



# Sampling from a grid-approximate posterior

- We're going to scoop out 10,000 values from the bucket.
- Provided the bucket is well mixed, the resulting samples will have the same proportions as the exact posterior density.
- Therefore the individual values of  $p$  will appear in our samples in proportion to the posterior plausibility of each value.
- Here's how you can do this in R, with one line of code:

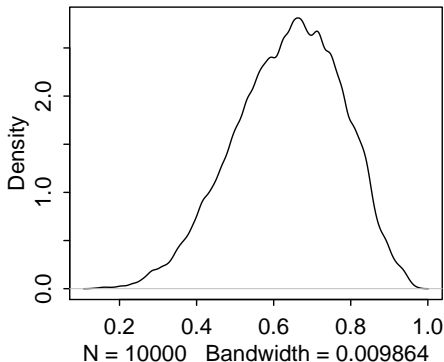
```
samples <- sample( p_grid , prob=posterior , size=1e4 ,  
replace=TRUE )
```

- We are randomly pulling values from the grid of parameter values where the probability of each value is given by the posterior.

# Sampling from a grid-approximate posterior

- We can visualize a density plot of our posterior sample as follows:

```
library(rethinking)  
dens(samples)
```

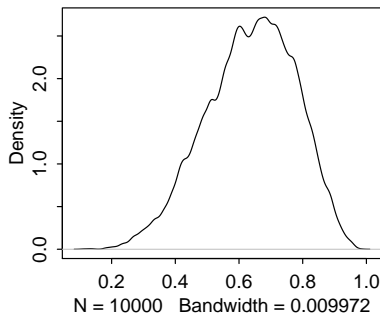


- We can see that the estimated density is very similar to the ideal posterior we computed via grid approximation in previous class.

# Sampling from the theoretical posterior

- We could get very similar results by sampling from the theoretical posterior using the beta distribution:

```
teo.samples<-rbeta(1e4,7,4)  
dens(teo.samples)
```



- We can see that the samples of the grid-approximated posterior and the theoretical posterior are indistinguishable.
- However, we should keep in mind that for complex models we will not have access to the posterior closed form.



# Intervals of defined boundaries

- Suppose we are asked for the posterior probability that the proportion of water is less than 0.5.
- We could calculate this from the theoretical posterior's CDF:

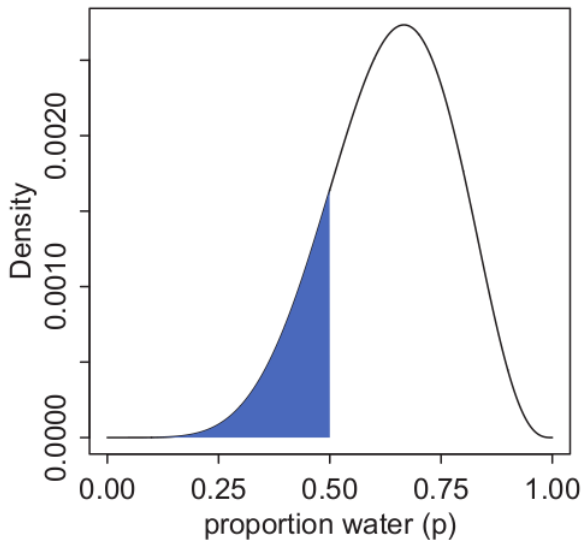
```
> pbeta(0.5, 7, 4)
[1] 0.171875
```

- Or alternatively we could calculate it from the grid-approximate posterior by adding up all of the probabilities where the corresponding parameter value is less than 0.5.

```
> sum( posterior[ p_grid < 0.5 ] )
[1] 0.1718746
```

- So about 17% of the posterior probability is below 0.5.

# Intervals of defined boundaries



# Intervals of defined boundaries

- Now, let's perform the same calculation, using samples from the posterior.
- Recall that in more complex models neither a grid-approximation nor a closed-form posterior will be available.
- All we have to do is add up all samples less than 0.5 and divide the resulting count by the total number of samples.

```
> sum( samples < 0.5 ) / 1e4  
[1] 0.1752
```

- In R, the condition `samples < 0.5` returns a logical vector, so since R treats TRUE values as 1, `sum` will count all the samples satisfying the condition.

# Intervals of defined boundaries

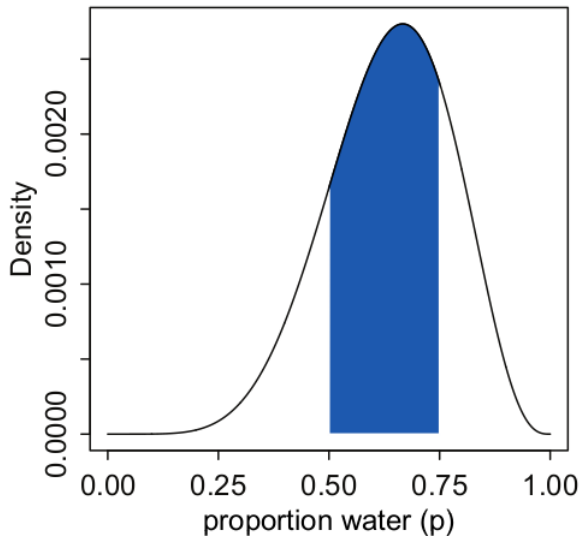
- Now, we can ask our sample how much posterior probability lies between 0.5 and 0.75.

```
> sum( samples > 0.5 & samples < 0.75 ) / 1e4  
[1] 0.6043
```

- So about 61% of the posterior probability lies between 0.5 and 0.75.
- Let's validate this result using the exact posterior:

```
> pbeta(0.75, 7, 4) - pbeta(0.5, 7, 4)  
[1] 0.6040001
```

# Intervals of defined boundaries



# Intervals of defined probability

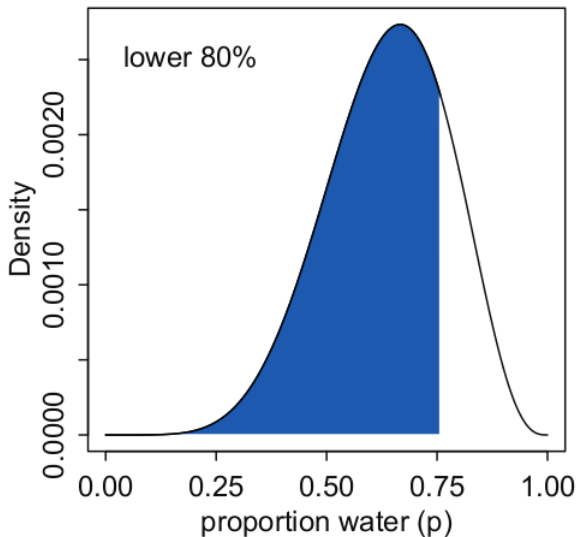
- Suppose we want to know the boundaries of the lower 80% posterior probability.
- We can answer this by obtaining the 80-th percentile of the posterior sample:

```
> quantile( samples , 0.8 )  
      80%  
0.7577578
```

- Or alternatively, using the quantile function of the beta distribution (the distribution of the exact posterior):

```
> qbeta(0.8, 7, 4 )  
[1] 0.7605588
```

# Intervals of defined probability



# Intervals of defined probability

- Similarly, we can calculate the middle 80% interval that lies between the 10th percentile and the 90th percentile:

```
> quantile( samples , c( 0.1 , 0.9 ) )  
      10%      90%  
0.4504505 0.8148148
```

- The “rethinking” package provides the function `PI` (from percentile interval) to calculate this type of interval:

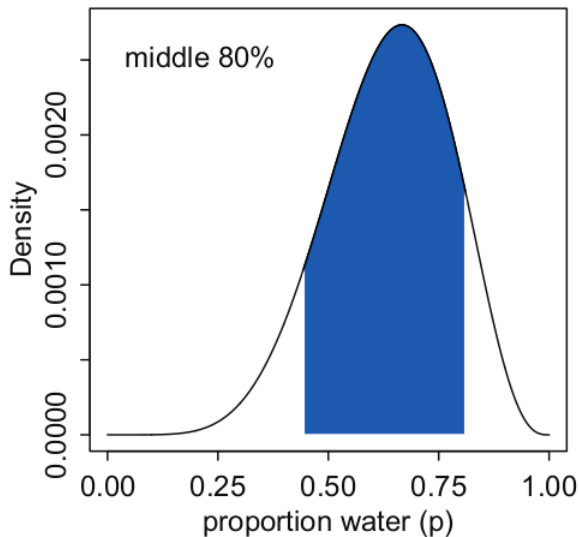
```
> PI( samples , prob=0.8 )  
      10%      90%  
0.4504505 0.8148148
```

- Notice that we are assigning  $(1 - 0.8)/2 = 0.1$  of probability above and below the interval.
- We can also obtain the exact interval from the exact posterior:

```
> c( "10%"=qbeta(0.1,7,4) , "90%"=qbeta(0.9,7,4) )  
      10%      90%  
0.4482692 0.8124377
```



# Intervals of defined probability



# Credible Intervals

- The intervals of posterior probability that assign equal probability to each tail are called **credible intervals**.
- These posterior intervals report two parameter values that contain between them a specified amount of posterior probability.
- What the interval indicates is a range of parameter values compatible with the model and data.
- Credible intervals resemble very much the confidence intervals seen in previous lectures on frequentist inference.
- The interpretations are very different though.
- A confidence interval is a region<sup>1</sup> that after infinitely repeating the data sampling experiment will contain the true parameter with a certain frequency.
- In contrast, a credible interval is a range of values that we believe our parameter can take with a certain probability according to both our prior beliefs and the evidence given by the data.

---

<sup>1</sup>Notice that the region will vary from one experiment to another.

# Credible Intervals

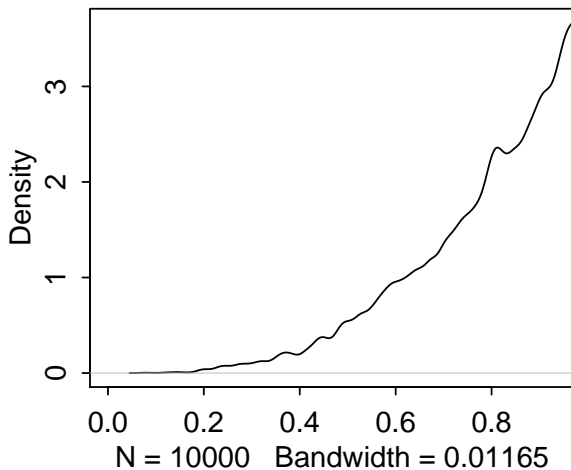
- Equal-tailed credible intervals do a good job of communicating the shape of a distribution, as long as the distribution isn't too asymmetrical.
- Suppose that in our globe tossing experiment we had observed 3 W and 0 L.
- If we again consider a flat prior, we will get a highly skewed posterior distribution with its maximum value at the boundary,  $p = 1$ .

```
p_grid.a <- seq( from=0 , to=1 , length.out=1000 )
prior.a <- rep(1,1000)
likelihood.a <- dbinom( 3 , size=3 , prob=p_grid.a )
posterior.a <- likelihood.a * prior.a
posterior.a <- posterior.a / sum(posterior.a)
samples.a <- sample( p_grid.a , size=1e4 ,
  replace=TRUE , prob=posterior.a )
dens(samples.a,xlim=c(0,0.935))
```

- Alternatively we could sample from the exact posterior  $Beta(\alpha + W, \beta + L) = Beta(1 + 3, 1 + 0) = Beta(4, 1)$ :

```
teo.samples.a<-rbeta(1e4,4,1)
dens(teo.samples.a,xlim=c(0,0.935))
```

# Credible Intervals



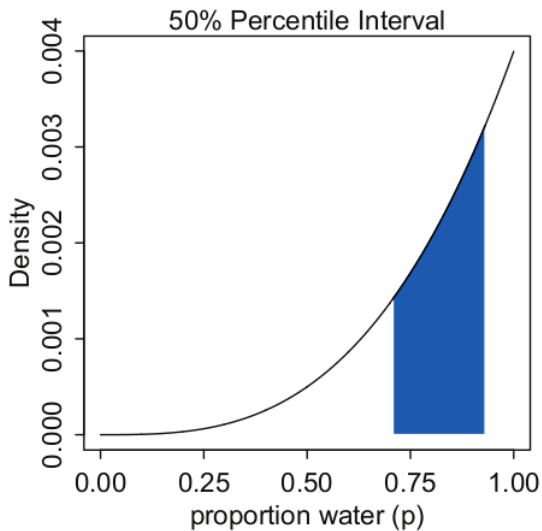
# Credible Intervals

- Let's compute a 50% equal-tailed credible interval for this posterior:

```
> PI( samples.a , prob=0.5 )  
      25%      75%  
0.7037037 0.9309309
```

- This interval assigns 25% of the probability area above and below the interval.
- So it provides the central 50% probability.
- But in this example, it ends up excluding the most probable parameter values, near  $p = 1$ .
- So, in terms of describing the shape of the posterior distribution it can be misleading.

# Credible Intervals



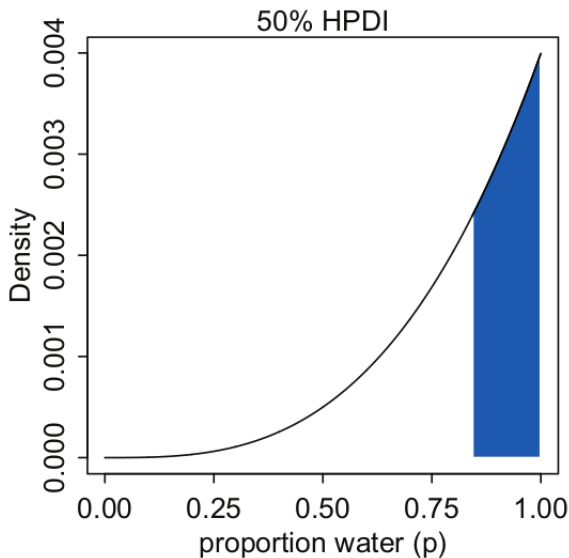
# Highest Posterior Density Intervals

- An alternative type of credible interval is the Highest Posterior Density Interval (HPDI).
- If we relax the restriction of assigning equal probability to each tail, we obtain an infinite number of intervals containing the specified probability area.
- The HPDI is the narrowest of those possible interval.
- It can be calculated from posterior samples using the HPDI function from the `rethinking` package.

```
> HPDI( samples.a , prob=0.5 )  
      |0.5      0.5|  
0.8368368 1.0000000
```

- This interval captures the parameters with highest posterior probability, as well as being noticeably narrower: 0.16 in width rather than 0.23 for the equal-tailed credible interval.

# Highest Posterior Density Intervals





# Highest Posterior Density Intervals

- A disadvantage of the HPDI, is that it is more computationally intensive than the equal-tailed credible interval.
- Apart from the cases when the posterior distribution is highly skewed, these two types of intervals are similar.
- For example, let's calculate an 80% HPDI for the the original posterior with 6 W and 3 L:

```
> HPDI( samples , prob=0.8 )  
      |0.8      0.8|  
0.4694695 0.8298298
```

- This interval is very similar to the equal-tailed credible interval calculated before.

# Point Estimates

- The idea of point estimation in a Bayesian setting is to summarize the posterior with a single value.
- The three most common options here:
  - The mode, which is the value with highest posterior probability, also known as the maximum a posteriori (MAP) estimate.
  - The mean
  - The median
- Let's calculate them for the globe tossing experiment in which we observe 3 waters out of 3 tosses.

- We can compute the MAP from the grid approximation of the posterior as follows:

```
> p_grid[ which.max(posterior.a) ]  
[1] 1
```

- Or we can approximate it using posterior samples:

```
> dd <- density(samples.a, adj=0.01)  
> dd$x[which.max(dd$y)]  
[1] 0.9971593
```

- The same procedure can be done more easily using the chainmode function from the rethinking package:

```
> chainmode( samples.a , adj=0.01 )  
[1] 0.9971593
```

# Point Estimates

- Now the posterior mean:

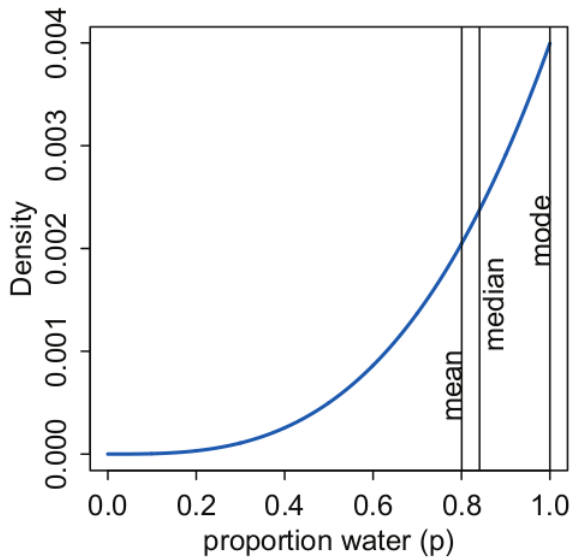
```
> mean(samples.a)
[1] 0.7988011
```

- and the median:

```
> median(samples.a)
[1] 0.8408408
```

- Which of these values should we report?
- Recall that our ultimate goal is to report the shape of the posterior.
- Hence, it is better to communicate as much as we can about it.
- This can include: density plots, HPDI, MAP estimates, mean, mode, etc..

# Point Estimates



# Sampling to Simulate Prediction

- Another useful thing we can do with the posterior, is to use it to simulate new data predictions.
- This can be particularly useful for evaluating our model in an empirical way.
- The idea is to contrast the simulated data with the expected behavior.
- These simulated predictions can also be used to forecast future observations.
- But, we must recall that the posterior is a distribution of the parameter given the data  $f(\theta|d)$ , so how can we use it to generate new unseen observations  $\tilde{d}$ ?
- To understand this, we need to learn about the **posterior predictive distribution**.
- But before introducing this complex new concept, we will learn how to generate simulated predictions using the likelihood function of the model.

# Sampling to Simulate Prediction

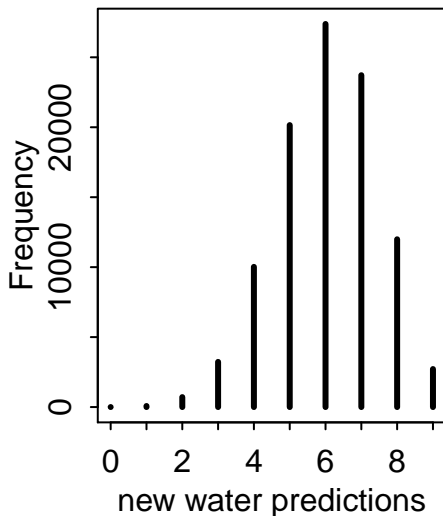
- In our original globe tossing experiment, the MAP estimate (the value of  $p$  that maximizes the posterior) was 0.67.
- We can use the likelihood function (a Binomial in this case) with  $p = 0.67$  to generate new observations of waters  $\tilde{d}$  with 9 new tosses.

```
> rbinom( 1, size=9 , prob=0.67)
[1] 6
```

- In this new data, we obtained 6 W out of 9 tosses, which is an expected behavior considering that the original data also had 6 W (out of 9).
- Now, we can repeat this process 100,000 times and observe a sampling distribution for the number of waters obtained:

```
> new_w <- rbinom( 1e5 , size=9 , prob=0.67 )
> simplehist( new_w , xlab="new water predictions")
```

# Sampling to Simulate Prediction





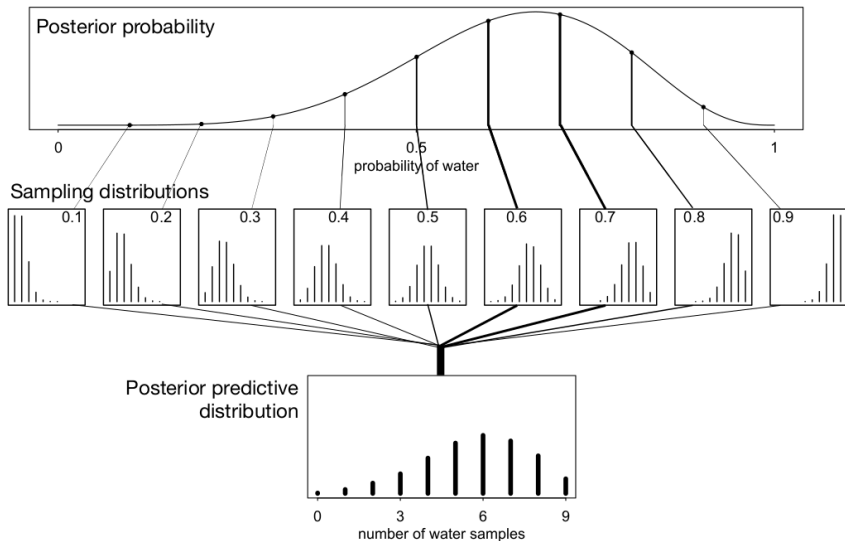
# Sampling to Simulate Prediction

- We can see that even 6 W is the most frequent case, 7 and 5 are also very likely to occur in our sampling distribution.
- These predictions embody the observation uncertainty: for a given value of  $p$  the number of W may vary according to our likelihood function (unless  $p = 0$  or  $p = 1$ ).
- But there is an additional source of uncertainty that our current predictions are not taking into account: the uncertainty about  $p$ .
- The posterior distribution over  $p$  embodies this uncertainty.
- And since there is uncertainty about  $p$ , there is uncertainty about everything that depends upon  $p$ .
- We lose this information when we pluck out a single parameter value (e.g., the MAP estimate) and then perform calculations with it.

# Posterior Predictive Distribution

- This loss of information leads to overconfidence.
- We'd like to propagate the parameter uncertainty as we evaluate the implied predictions.
- All that is required is averaging over the posterior density for  $p$ , while computing the predictions.
- For each possible value of the parameter  $p$ , there is an implied distribution of outcomes.
- So if you were to compute the sampling distribution of outcomes at each value of  $p$ , then you could average all of these prediction distributions together, using the posterior probabilities of each value of  $p$ , to get a **posterior predictive distribution**.

# Posterior Predictive Distribution



# Posterior Predictive Distribution

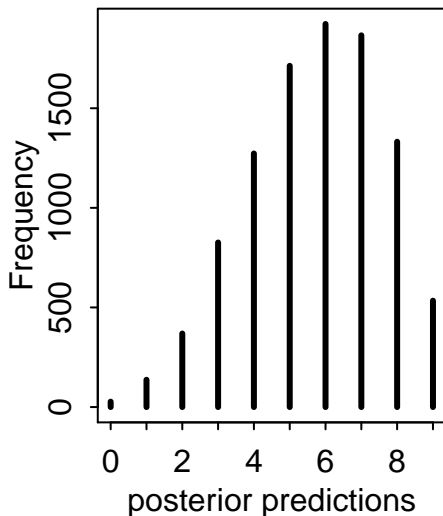
- The figure above illustrates this averaging.
- At the top, the posterior distribution is shown, with 10 unique parameter values highlighted by the vertical lines.
- The implied distribution of observations specific to each of these parameter values is shown in the middle row of plots.
- Observations are never certain for any value of  $p$ , but they do shift around in response to it.
- Finally, at the bottom, the sampling distributions for all values of  $p$  are combined, using the posterior probabilities to compute the weighted average frequency of each possible observation, zero to nine water samples.
- The resulting distribution is for predictions, but it incorporates all of the uncertainty embodied in the posterior distribution for the parameter  $p$ .
- As a result, it is more honest than the distribution of predictions computed with the MAP estimate.

# Posterior Predictive Distribution

- Generating predictions with the MAP estimate can lead us to believe that the model is more consistent with the data than it really is.
- This illusion arises from tossing away uncertainty about the parameters.
- Consequently, MAP predictions tend to cluster around the observations more tightly (i.e., produce narrower distributions).
- To generate predictions using the Posterior Predictive Distribution in R, we must replace the MAP parameter value with samples from the posterior:

```
> post_pred_w <- rbinom( 1e4 , size=9 , prob=samples )  
> simplehist( post_pred_w , xlab="posterior predictions")
```
- For each sampled value, a random binomial observation is generated.
- Since the sampled values appear in proportion to their posterior probabilities, the resulting simulated observations are averaged over the posterior.

# Posterior Predictive Distribution



# Posterior Predictive Distribution

- Let's discuss the posterior predictive distribution more formally as presented in [Gelman et al., 2013].
- After the data  $d$  have been observed, we can predict an unknown observable  $\tilde{d}$  from the same process.
- In the globe tossing experiment  $\tilde{d}$  may be the yet to be recorded number of waters  $W$  in a new tossing experiment.
- The distribution of  $\tilde{d}$  given  $d$  is called the **posterior predictive distribution**.
- Posterior because it is conditional on the observed  $d$  and predictive because it is a prediction for an observable  $\tilde{d}$ .

# Posterior Predictive Distribution

- Mathematical, the posterior predictive distribution is defined as follows:

$$\begin{aligned}f(\tilde{d}|d) &= \int_{\theta} f(\tilde{d}, \theta|d) d\theta \\&= \int_{\theta} f(\tilde{d}|\theta, d) f(\theta|d) d\theta \\&= \int_{\theta} f(\tilde{d}|\theta) f(\theta|d) d\theta\end{aligned}$$

- The second and third lines display the posterior predictive distribution as an average of conditional predictions over the posterior distribution of  $\theta$ .
- The last step follows from the assumed conditional independence of  $d$  and  $\tilde{d}$  given  $\theta$ .

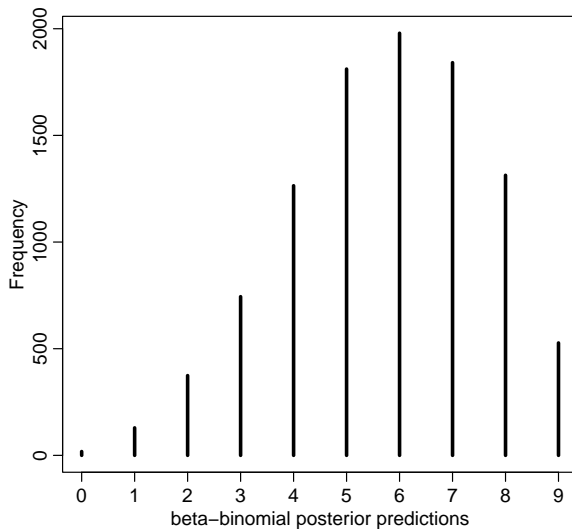


# Posterior Predictive Distribution

- When the posterior is a beta distribution and the likelihood is a binomial, such as in our globe tossing experiment, the posterior predictive distribution has a closed mathematical expression defined by the **beta-binomial** distribution.
- The beta-binomial distribution is the binomial distribution in which the probability of success at each of  $n$  trials is not fixed but randomly drawn from a beta distribution [Wikipedia, 2021].
- We can generate posterior predictive simulations using the “rbetabinom” function from the “rethinking” package:

```
post_pred_w_betabinom <-rbetabinom( n=1e4, size=9 ,  
                                     shape1=7 , shape2=4)  
simplehist( post_pred_w_betabinom ,  
            xlab="beta-binomial posterior predictions")
```

# Posterior Predictive Distribution



# Conclusions

- In this class we introduced the basic procedures for manipulating posterior distributions.
- Our fundamental tool is samples of parameter values drawn from the posterior distribution.
- Working with samples transforms a problem of integral calculus into a problem of data summary.
- These samples can be used to produce intervals, point estimates, posterior predictive checks, as well as other kinds of simulations.
- Posterior predictive checks combine uncertainty about parameters, as described by the posterior distribution, with uncertainty about outcomes, as described by the assumed likelihood function.
- These checks are useful for analyzing the behavior of the model.

# References I



Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013).

*Bayesian data analysis.*

CRC press.



McElreath, R. (2020).

*Statistical rethinking: A Bayesian course with examples in R and Stan.*

CRC press.



Murphy, K. P. (2021).

*Probabilistic Machine Learning: An introduction.*

MIT Press.



Wikipedia (2021).

Beta-binomial distribution — Wikipedia, the free encyclopedia.

<http://en.wikipedia.org/w/index.php?title=Beta-binomial%20distribution&oldid=1044534982>.

[Online; accessed 05-October-2021].