Statistical Rethinking

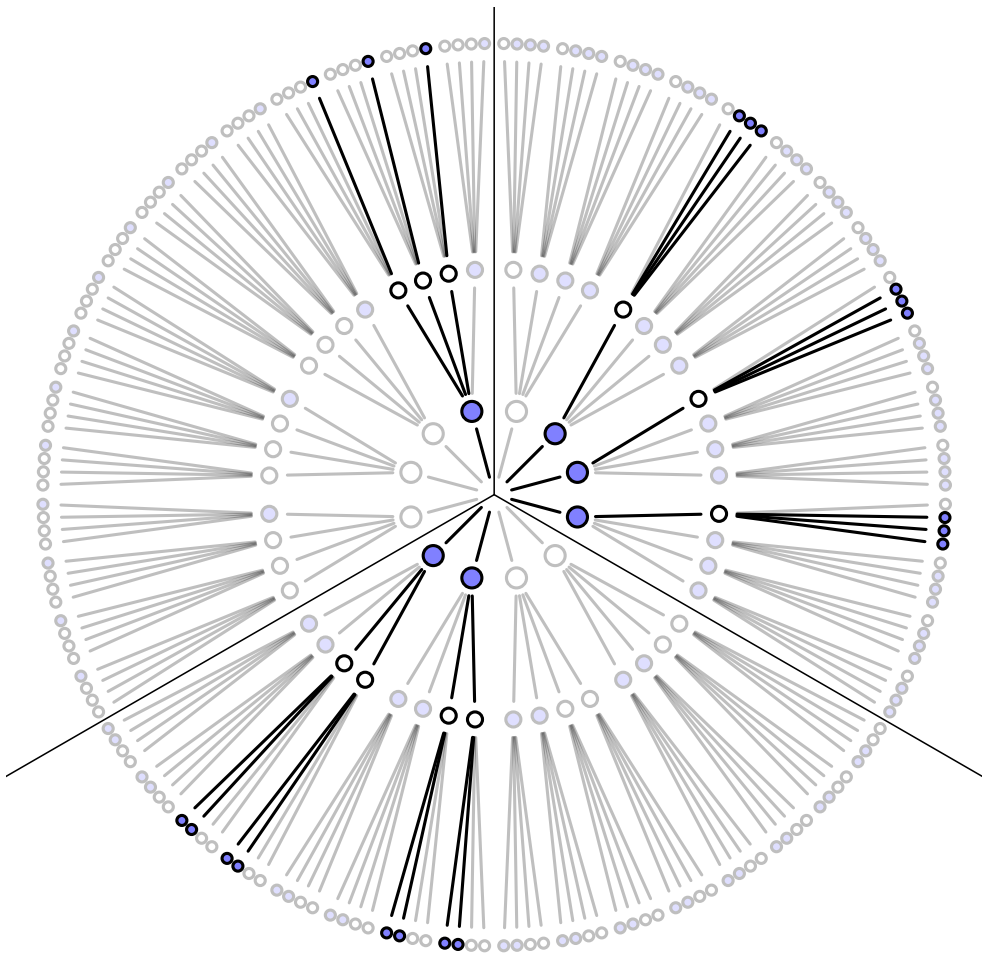
Week 1

Small Worlds and Large Worlds

(Chapter 2)

Garden of Forking Data

Conjecture	Ways to produce ●○●
[○○○○]	$0 \times 4 \times 0 = 0$
[●○○○]	$1 \times 3 \times 1 = 3$
[●●○○]	$2 \times 2 \times 2 = 8$
[●●●○]	$3 \times 1 \times 3 = 9$
[●●●●]	$4 \times 0 \times 4 = 0$



Updating

Another draw from the bag: 

Conjecture	Ways to produce ●	Previous counts	New count
[○○○○]	0	0	$0 \times 0 = 0$
[●○○○]	1	3	$3 \times 1 = 3$
[●●○○]	2	8	$8 \times 2 = 16$
[●●●○]	3	9	$9 \times 3 = 27$
[●●●●]	4	0	$0 \times 4 = 0$

Using other information

Factory says: ● marbles rare, but every bag contains at least one.

Conjecture	Factory count
[○○○○]	0
[●○○○]	3
[●●○○]	2
[●●●○]	1
[●●●●]	0

Using other information

Factory says: ● marbles rare.

Conjecture	Prior ways	Factory count	New count
[○○○○]	0	0	$0 \times 0 = 0$
[●○○○]	3	3	$3 \times 3 = 9$
[●●○○]	16	2	$16 \times 2 = 32$
[●●●○]	27	1	$27 \times 1 = 27$
[●●●●]	0	0	$0 \times 0 = 0$

Counts to plausibility

Unglamorous basis of applied probability:

Things that can happen more ways are more plausible.

Possible composition	p	ways to produce data	plausibility
[○○○○]	0	0	0
[●○○○]	0.25	3	0.15
[●●○○]	0.5	8	0.40
[●●●○]	0.75	9	0.45
[●●●●]	1	0	0

Counts to plausibility

Possible composition	p	ways to produce data	plausibility
[○○○○]	0	0	0
[●○○○]	0.25	3	0.15
[●●○○]	0.5	8	0.40
[●●●○]	0.75	9	0.45
[●●●●]	1	0	0

R code 2.1

```
ways <- c( 3 , 8 , 9 )
ways/sum(ways)
```

```
[1] 0.15 0.40 0.45
```

Counts to plausibility

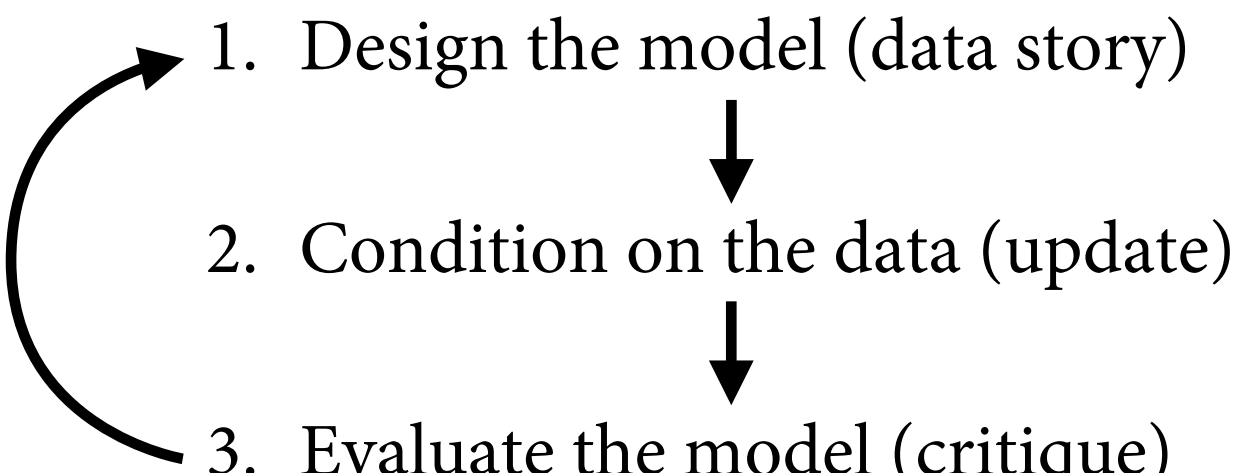
Possible composition	p	ways to produce data	plausibility
[○○○○]	0	0	0
[●○○○]	0.25	3	0.15
[●●○○]	0.5	8	0.40
[●●●○]	0.75	9	0.45
[●●●●]	1	0	0

Plausibility is *probability*: Set of non-negative real numbers that sum to one.

Probability theory is just a set of shortcuts for counting possibilities.

Building a model

- How to use probability to do typical statistical modeling?



Nine tosses of the globe:

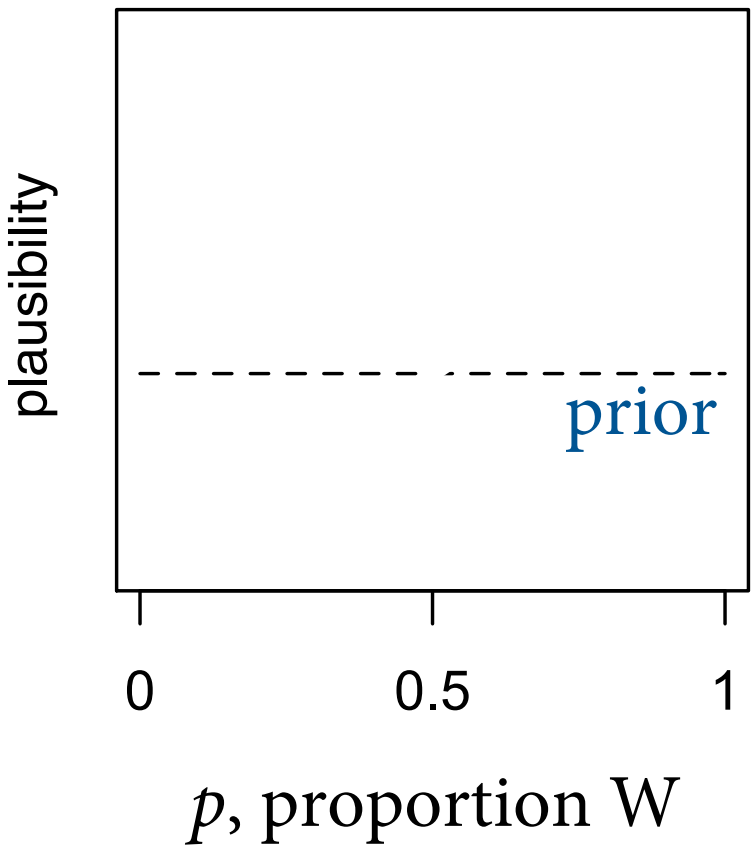
W L W W W L W L W

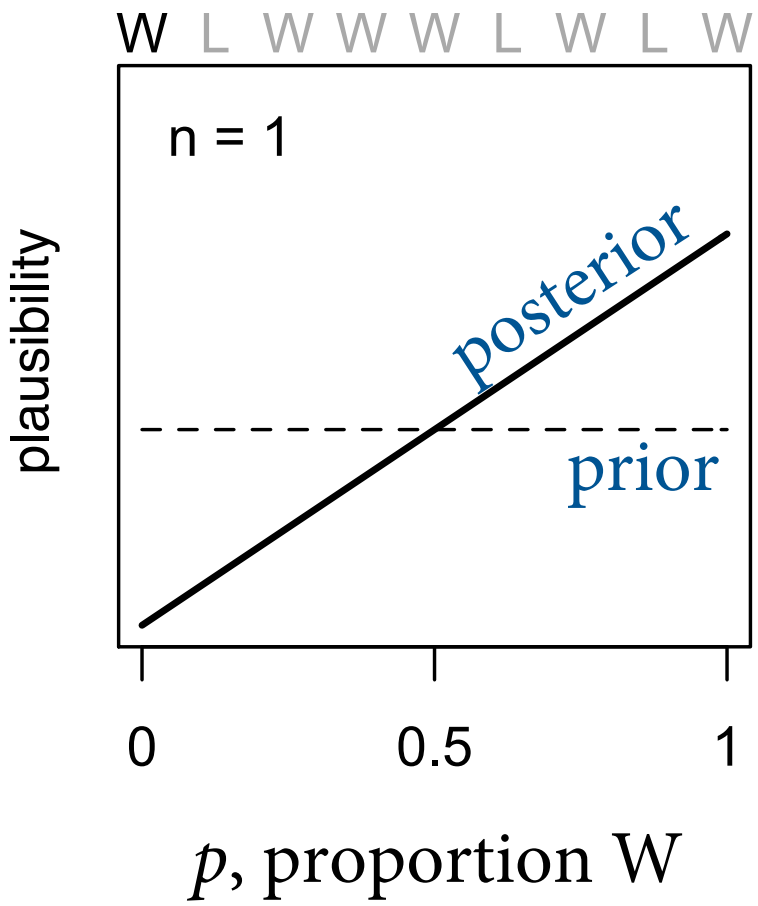
Design > Condition > Evaluate

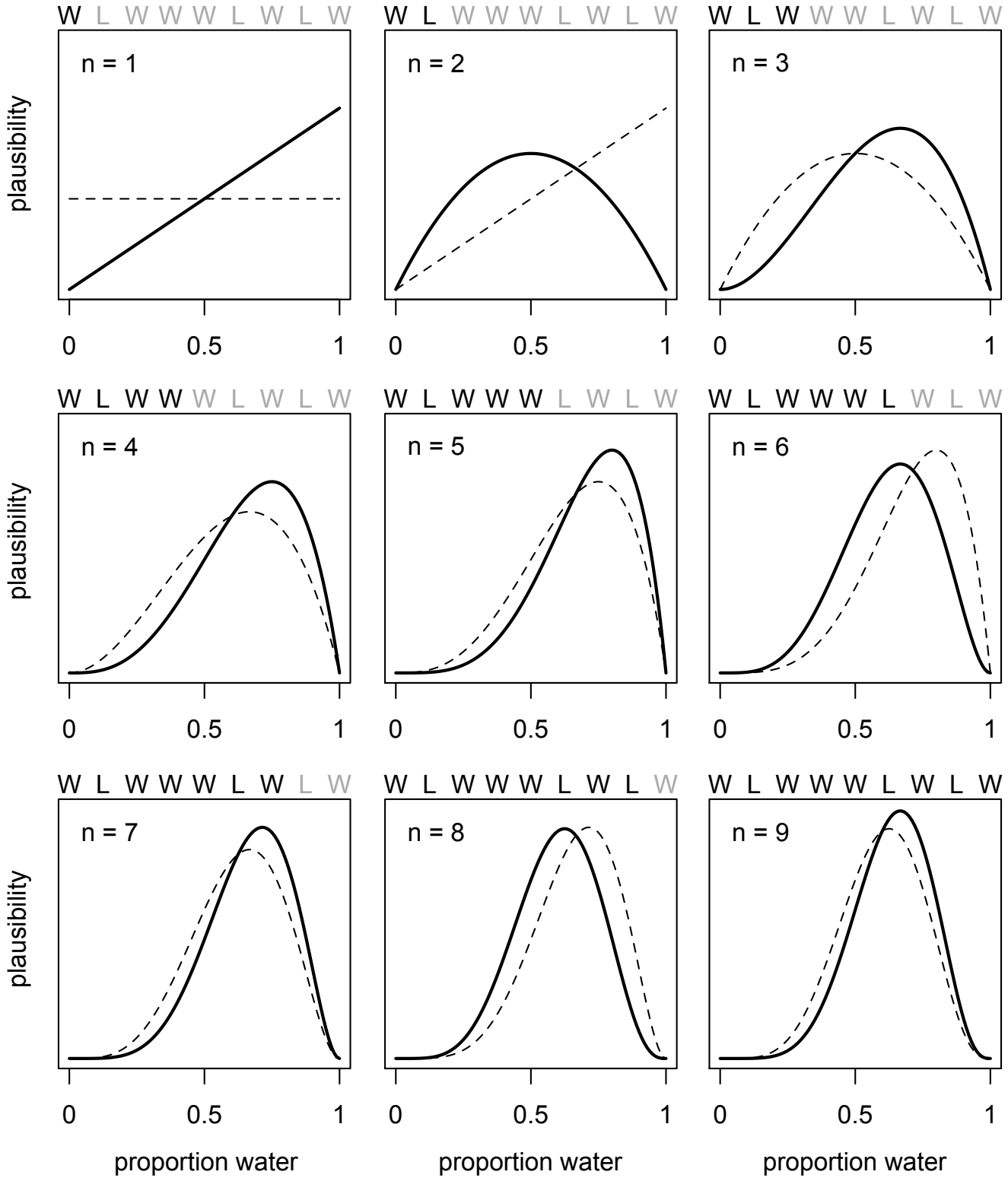
- Data story motivates the model
  - How do the data arise?
- For WLWWWLWLW:
  - Some true proportion of water, p
  - Toss globe, probability p of observing W, $1-p$ of L
  - Each toss therefore independent of other tosses
- Translate data story into probability statements

Design > **Condition** > Evaluate

- *Bayesian updating* defines optimal learning in small world, converts *prior* into *posterior*
  - Give your golem an information state, before the data: Here, an initial confidence in each possible value of p between zero and one
  - Condition on data to update information state: New confidence in each value of p, conditional on data

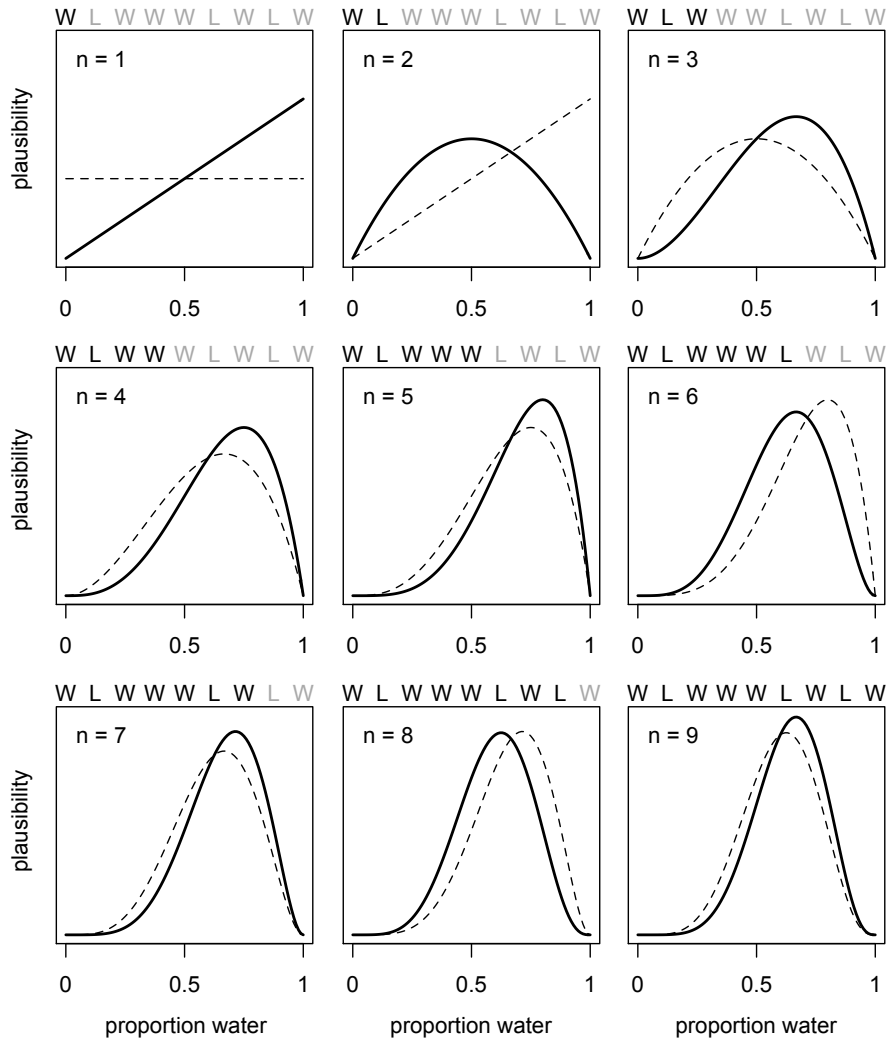

plausibility
prior
0
0.5
1
p, proportion W


W L W W W L W L W
n = 1
plausibility
posterior
prior
0
0.5
1
p, proportion W


W L W W W L W L W
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
n = 7
n = 8
n = 9
plausibility
0
0.5
1
proportion water

Design > **Condition** > Evaluate

- Data order irrelevant, because golem assumes order irrelevant
  - All-at-once, one-at-a-time, shuffled order all give same posterior
- Every posterior is a prior for next observation
- Every prior is posterior of some other inference

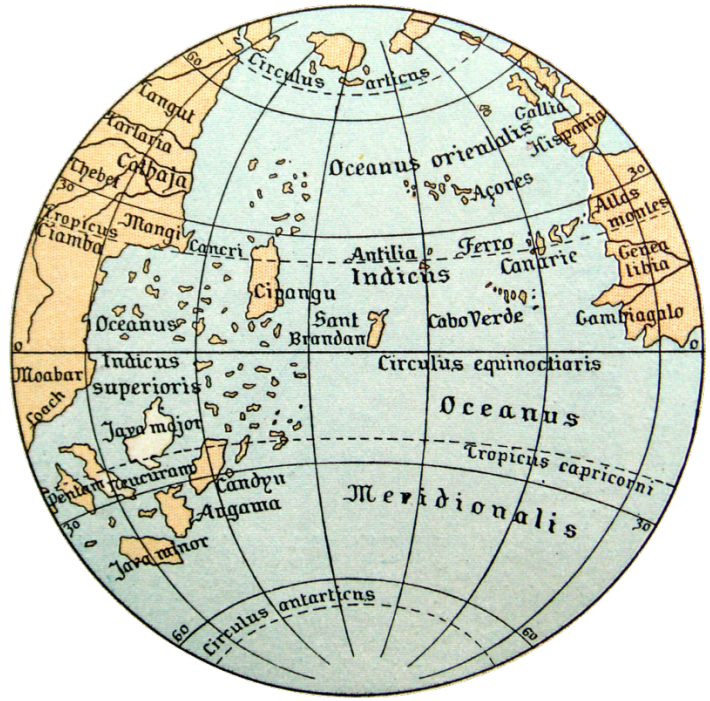


Design > Condition > **Evaluate**

- Bayesian inference: Logical answer to a question in the form of a model

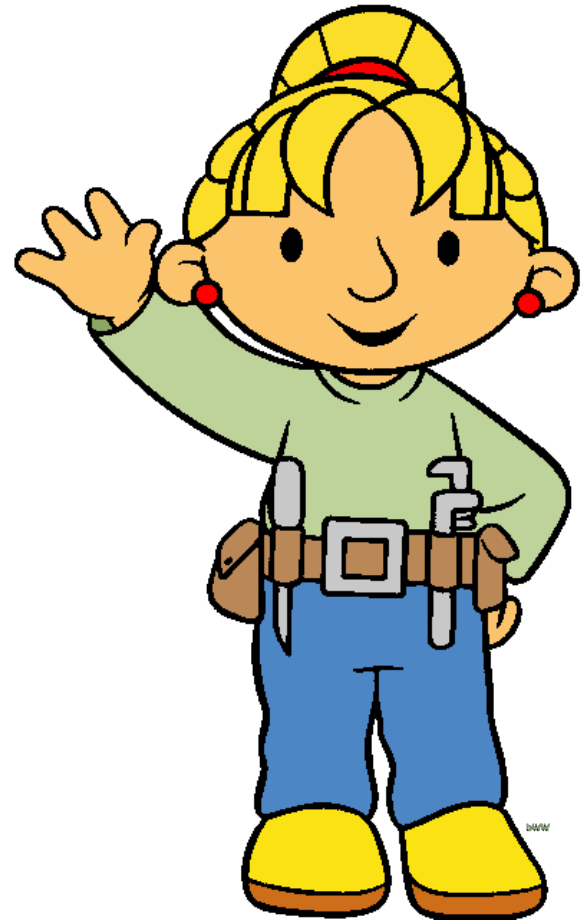
  "How plausible is each proportion of water, given these data?"

- Golem must be supervised
  - Did the golem malfunction?
  - Does the golem's answer make sense?
  - Does the question make sense?
  - Check sensitivity of answer to changes in assumptions



Construction perspective

- Build joint model:
  (1) List variables: data & parameters
  (2) Assign data distribution (likelihood)
  (3) Assign parameter distribution (prior)
- Input: Joint prior
- Deduce: Joint posterior



Variables: Data & Parameters

- Variables:
  - n: Number of globe tosses
  - n_W: Number of water landings
  - p: proportion of water on globe
- Some are *data* (n_W, n) – can be observed
- Others *parameters* (p) – cannot be observed
  - Define targets of inference, what is updated
  - These were the *conjectures* in the bag example
- Which are data and which parameters depend upon your context and question
  - e.g. mark-recapture: know n_W, must infer n, p

Data model: Likelihood

- Pr(data|assumptions)
  - Defines probability of each observation, conditional "|" on assumptions
  - i.e. relative count of number of ways of seeing data, given a particular conjecture
- In this case, binomial probability:

$$\Pr(n_{\mathrm{W}}|n,p) = \frac{n!}{n_{\mathrm{W}}!(n-n_{\mathrm{W}})!}p^{n_{\mathrm{W}}}(1-p)^{n-n_{\mathrm{W}}}$$

Data model: Likelihood

number tosses

$$\Pr(n_{\mathrm{W}}|n,p) = \frac{n!}{n_{\mathrm{W}}!(n-n_{\mathrm{W}})!}p^{n_{\mathrm{W}}}(1-p)^{n-n_{\mathrm{W}}}$$

count W *probability* W

The count of W's is distributed binomially, with probability p of a W on each toss and n tosses total.

Data model: Likelihood

$$\Pr(n_W|n,p) = \frac{n!}{n_W!(n-n_W)!}p^{n_W}(1-p)^{n-n_W}$$

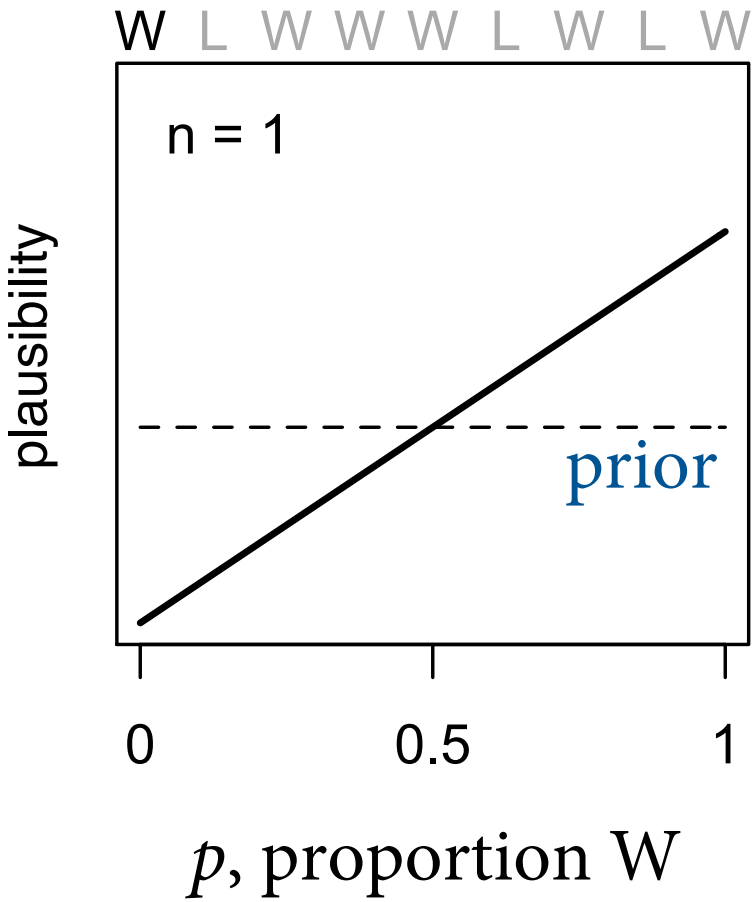
```
dbinom( 6 , size=9 , prob=0.5 )
```

R code 2.2

```
[1] 0.1640625
```

Parameter model: Prior

- What the golem believes before the data
- Likelihood & prior define *prior predictive distribution*
- More on this later – it helps us build priors that make sense



Parameter model: Prior

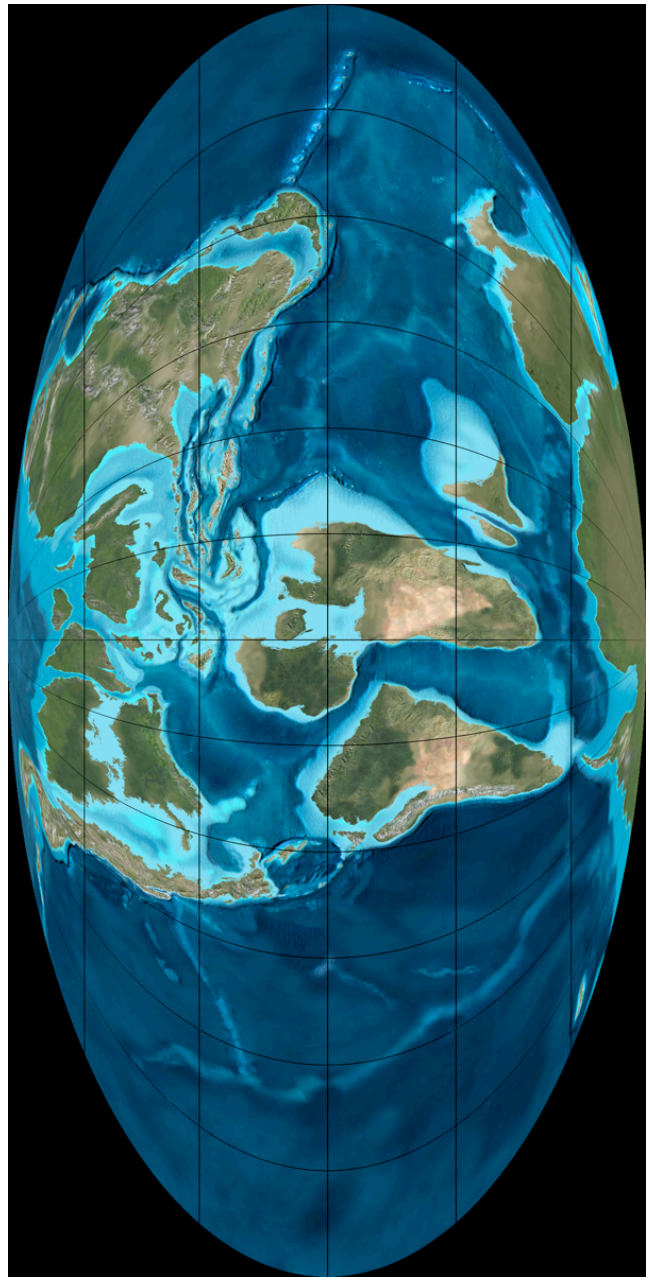
- Globe tossing model, a uniform (flat) prior:

$$\Pr(p) = \frac{1}{1-0} = 1$$

The prior distribution of p is assumed to be uniform in the interval from zero to one.

Prior literature

- Huge literature on choice of prior
- Flat prior conventional, but hardly ever best choice
  - Always know something (before data) that can improve inference
  - Are zero and one plausible values for p? Is $p < 0.5$ as plausible as $p > 0.5$?
  - There is no "true" prior
  - Just need to do better
- All above equally true of likelihood



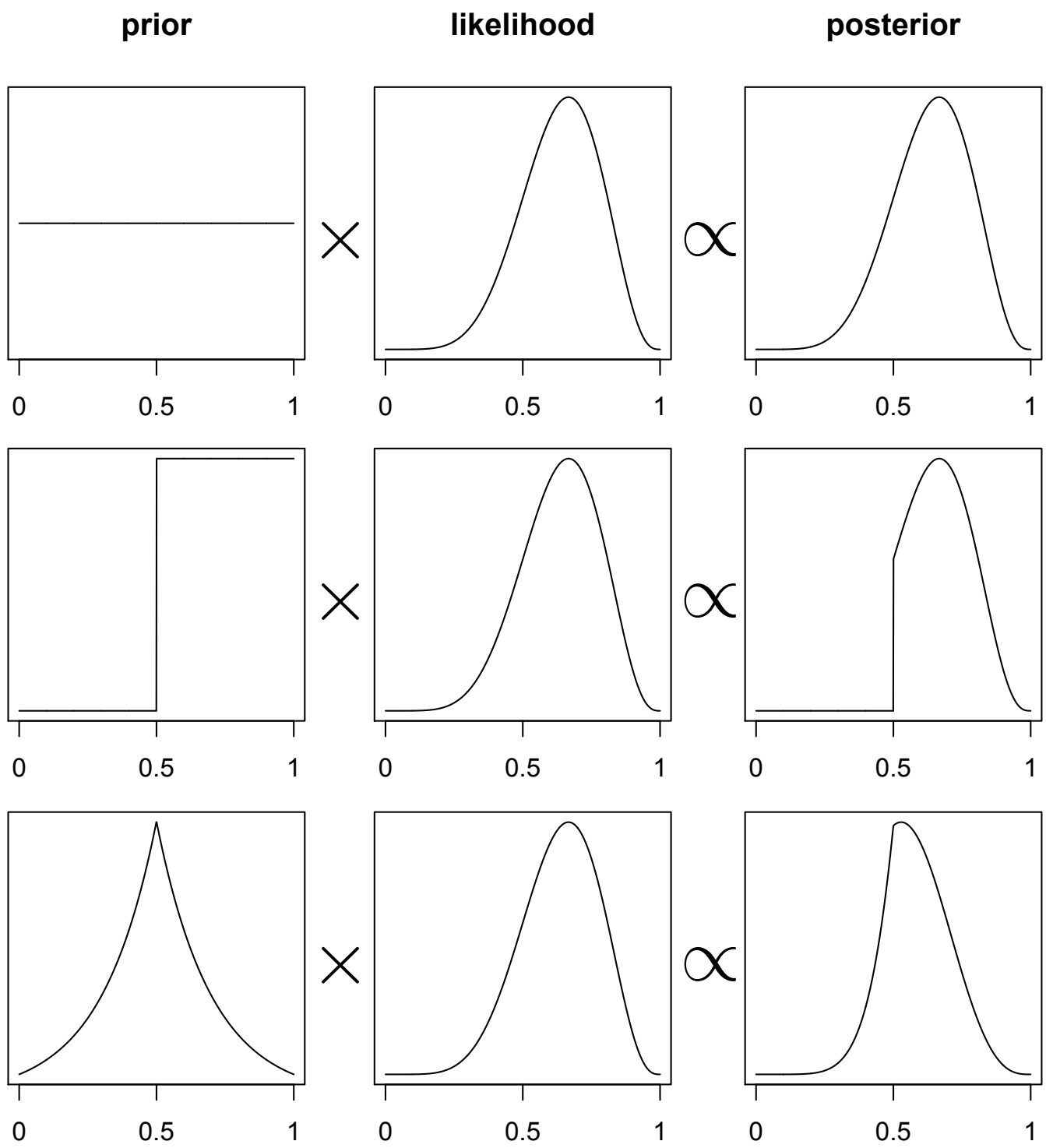
Late Cretaceous (90Mya)

Posterior

- Bayesian estimate is always *posterior distribution over parameters*, Pr(parameters|data)
- Here: $\Pr(p|n_{\mathrm{W}})$
- Compute using *Bayes' theorem*:

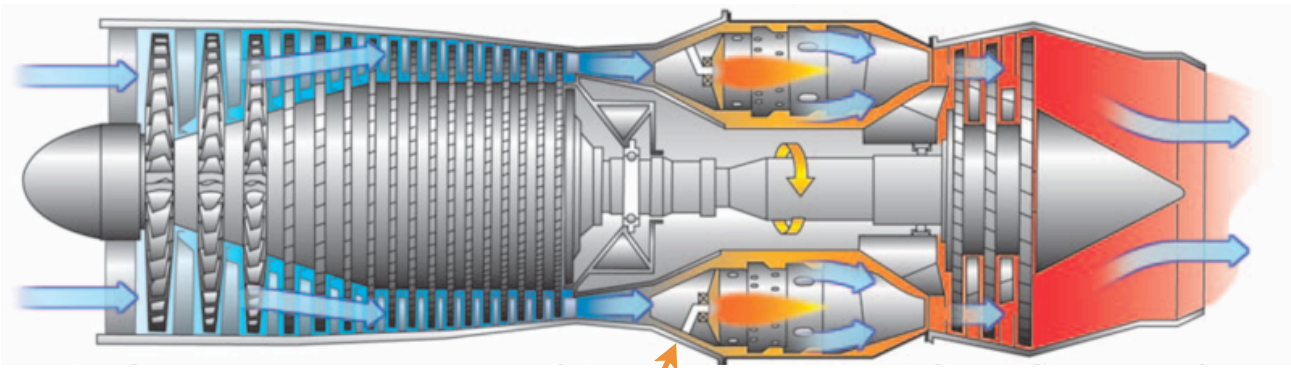
$$\Pr(p|n_{\mathrm{W}}) = \frac{\Pr(n_{\mathrm{W}}|p)\Pr(p)}{\Pr(n_{\mathrm{W}})}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Average Likelihood}}$$


prior
likelihood
posterior
0
0.5
1

The conditioning engine

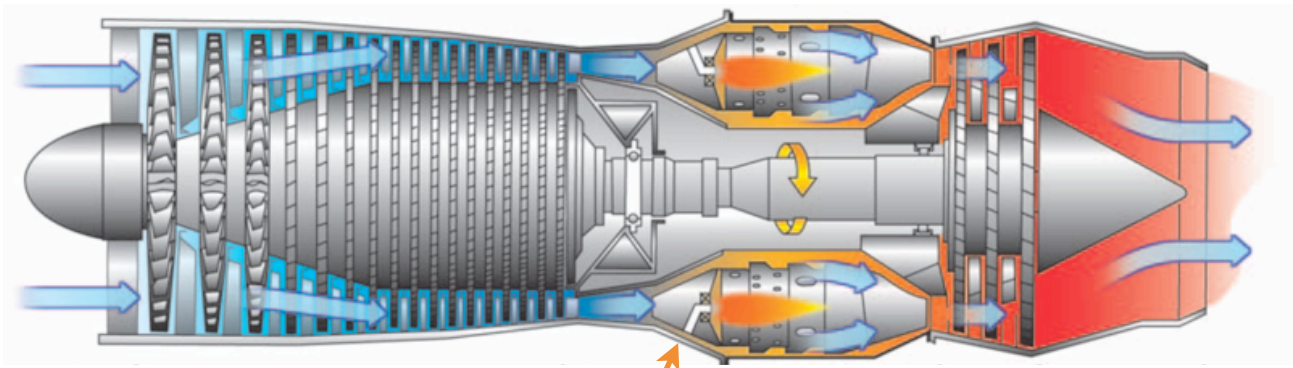
Prior



Likelihood

The conditioning engine

Prior



Posterior

Likelihood

Posterior is **prior** *conditioned* on **evidence**

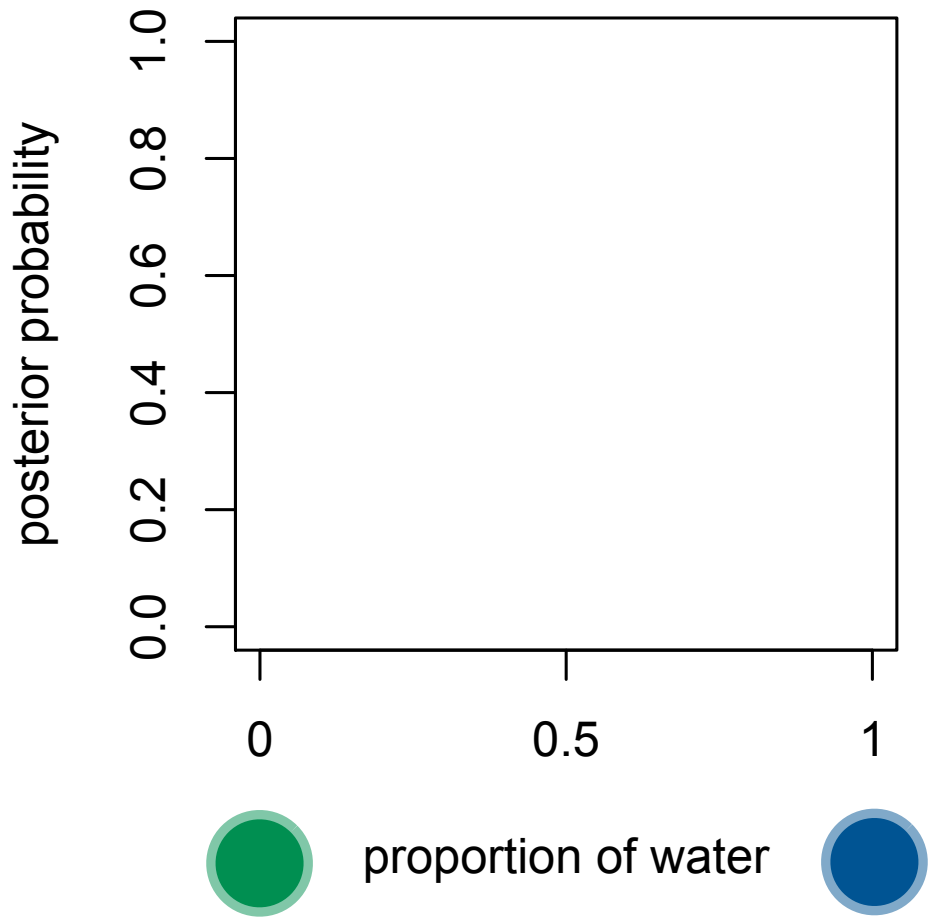
Computing the posterior

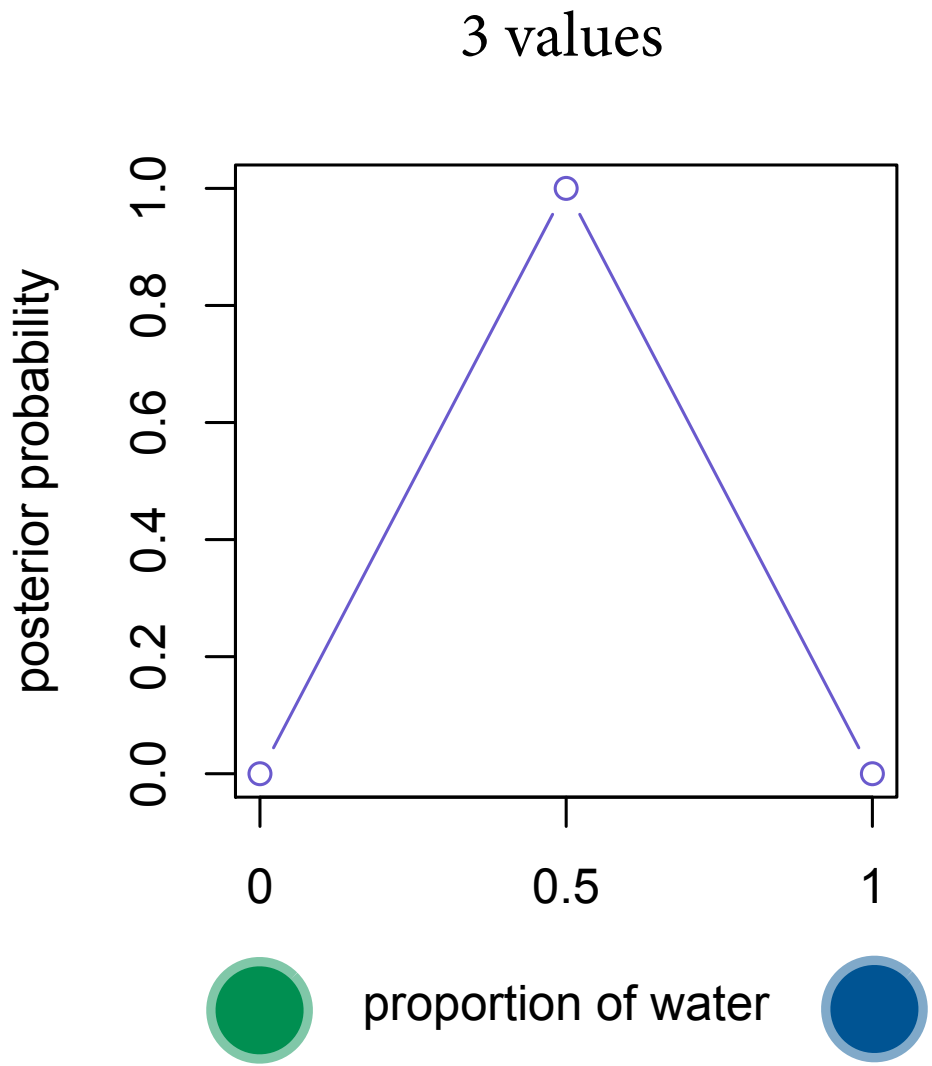
1. Analytical approach (often impossible)
2. Grid approximation (very intensive)
3. Quadratic approximation (approximate)
4. Markov chain Monte Carlo (intensive)

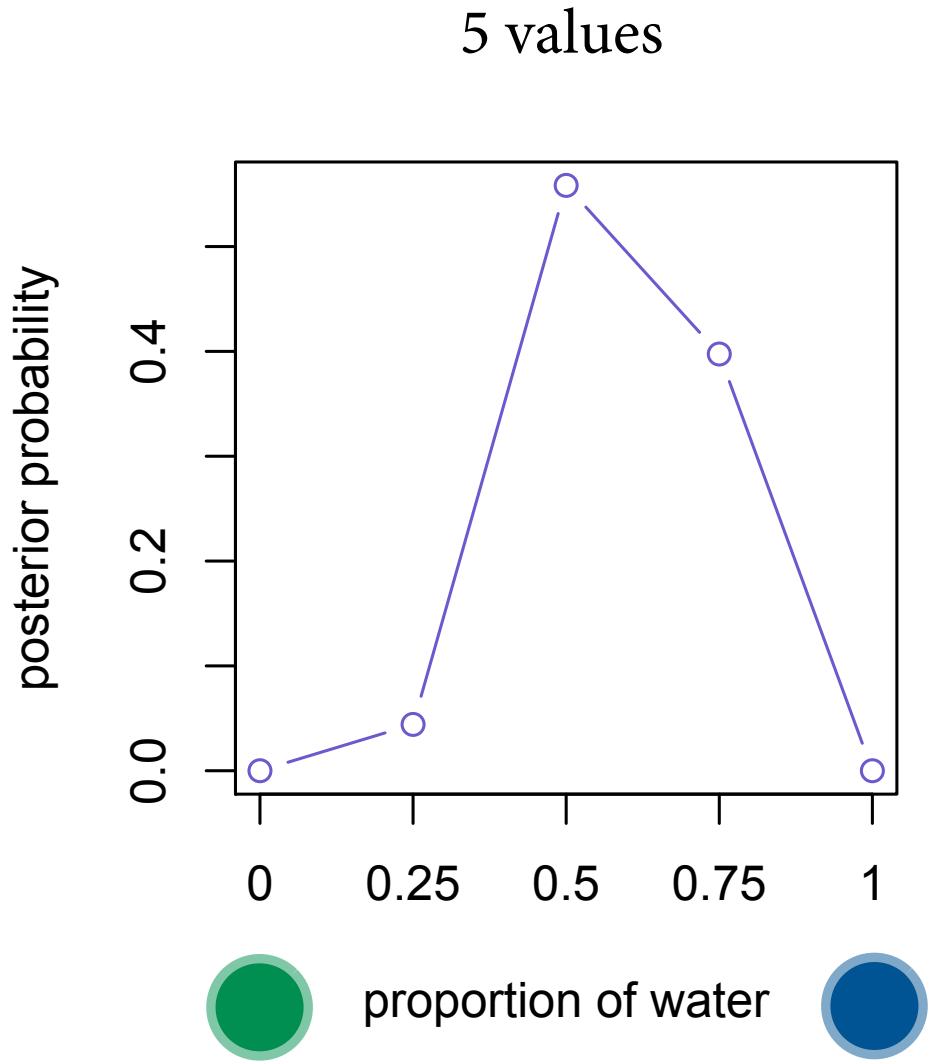
Grid approximation

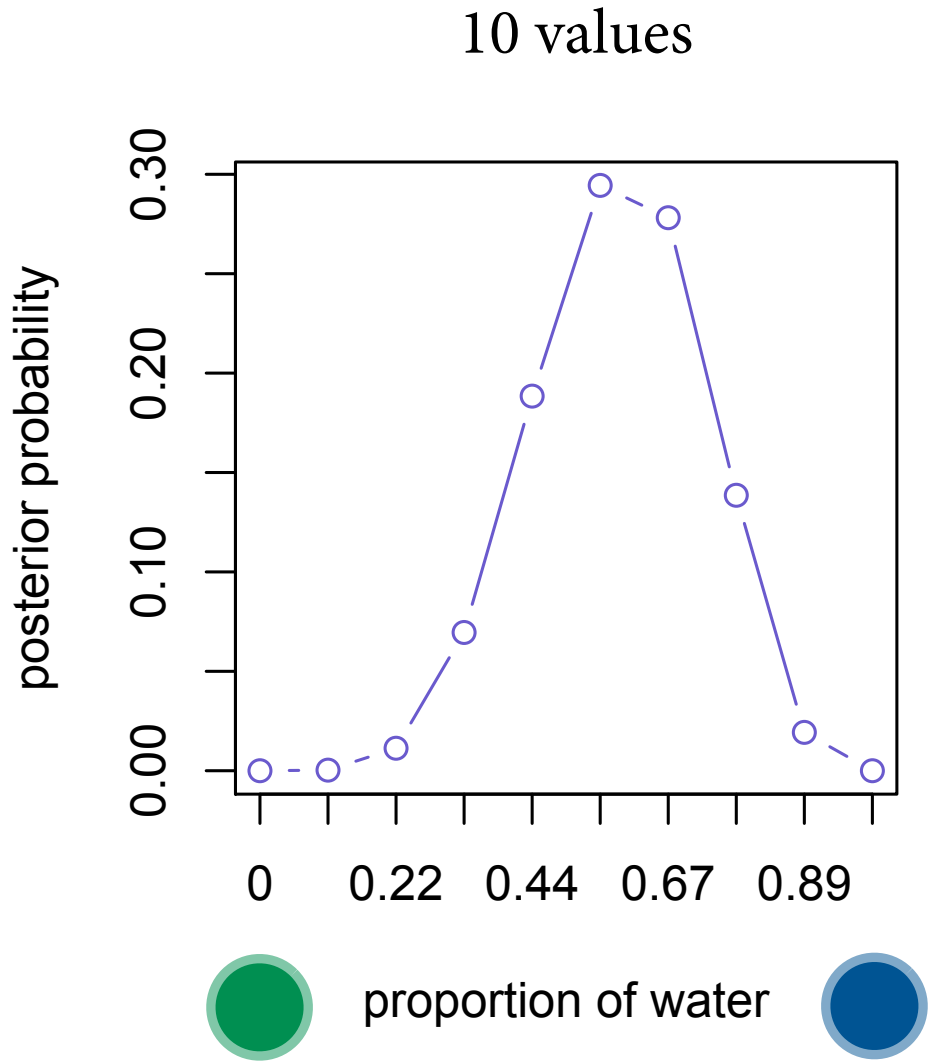
- The posterior is:
  standardized product of the likelihood and prior.

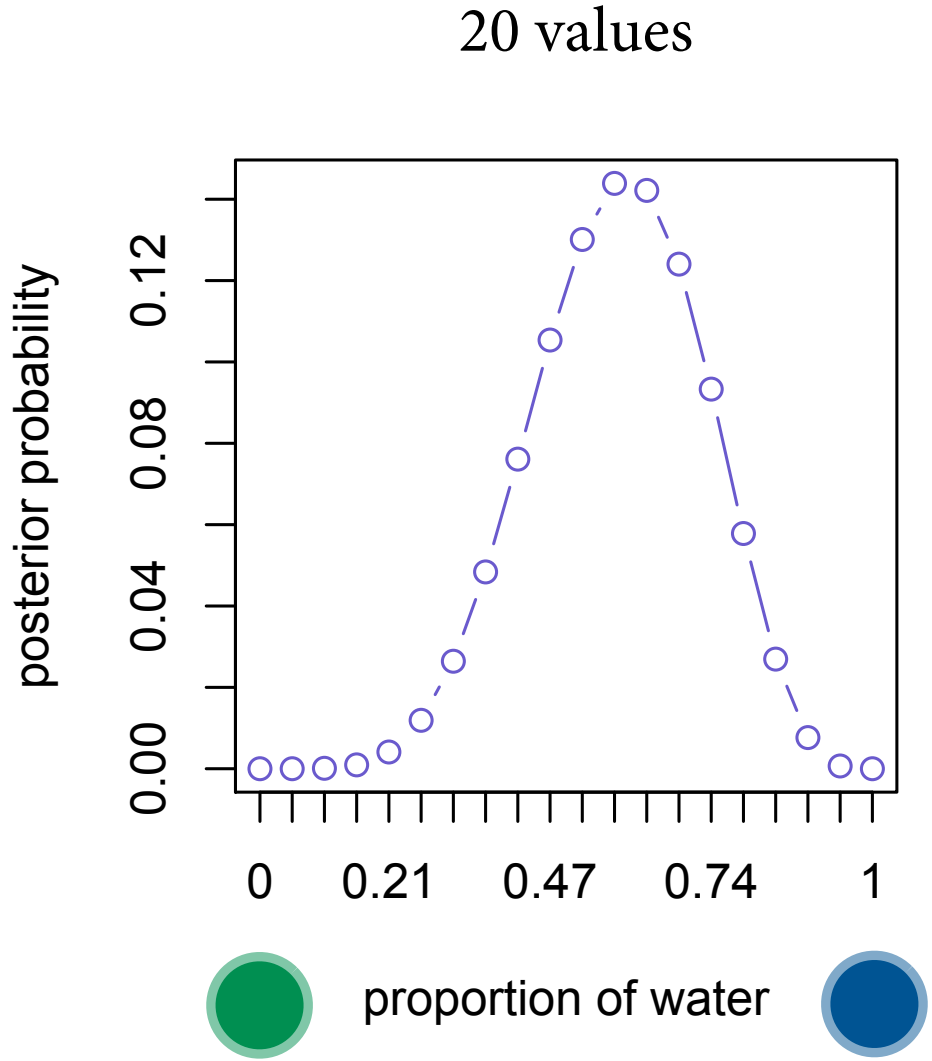
- Grid approximation uses *finite grid* of parameter values instead of continuous space

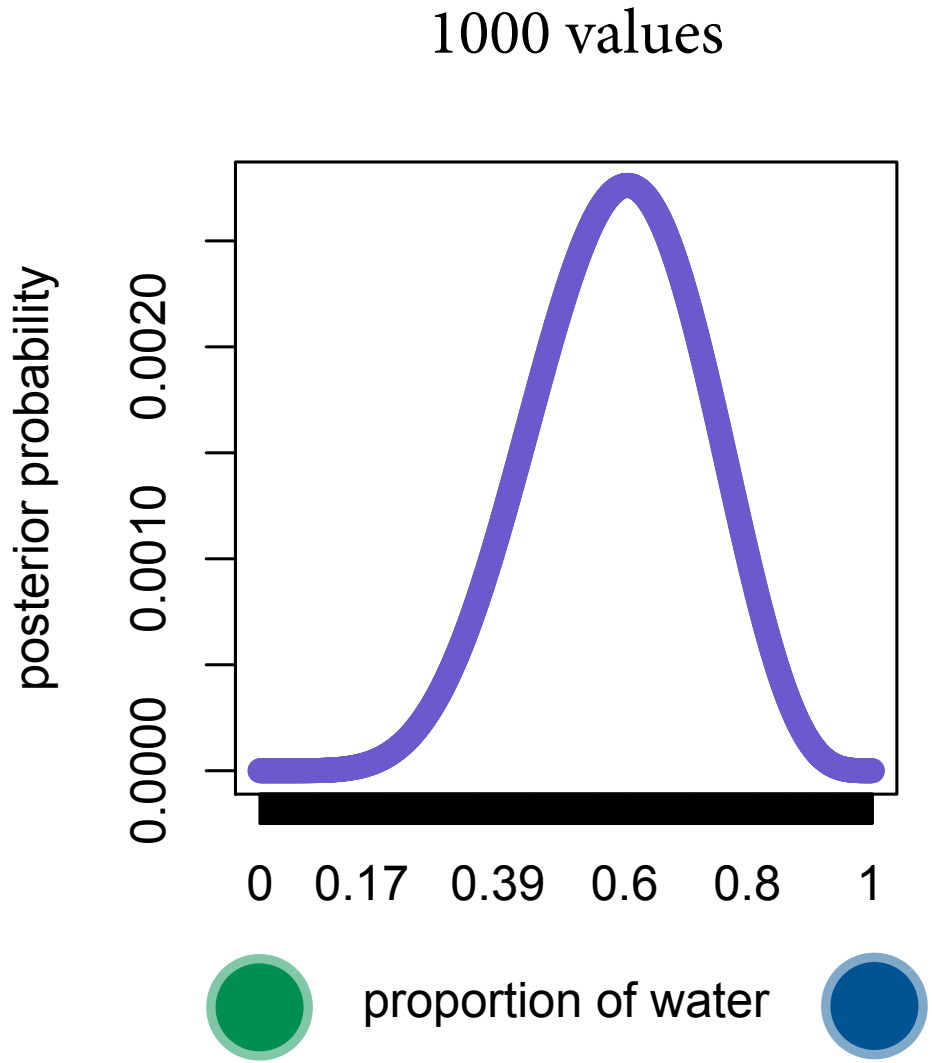

posterior probability
0.0
0.2
0.4
0.6
0.8
1.0
0
0.5
1
proportion of water


3 values
posterior probability
0.0
0.2
0.4
0.6
0.8
1.0
0
0.5
1
proportion of water


5 values
posterior probability
0.0
0.2
0.4
0
0.25
0.5
0.75
1
proportion of water


10 values
posterior probability
0.00
0.10
0.20
0.30
0
0.22
0.44
0.67
0.89
proportion of water


20 values
posterior probability
0.00
0.04
0.08
0.12
0
0.21
0.47
0.74
1
proportion of water


1000 values
posterior probability
0.0000
0.0010
0.0020
0
0.17
0.39
0.6
0.8
1
proportion of water

Quadratic approximation

- Assume posterior is normally distributed
- Can estimate with two numbers:
  - Peak of posterior, *maximum a posteriori* (MAP)
  - Standard deviation of posterior
- Lots of algorithms
- With flat priors, same as conventional *maximum likelihood estimation*

Statistical Rethinking

Week 1

Sampling the Imaginary

(Chapter 3)

Sampling from the posterior

- Incredibly useful to *sample randomly* from the posterior
  - Visualize uncertainty
  - Compute confidence intervals
  - Simulate observations
- MCMC produces only samples
- Above all, *easier to think with samples*

Sampling from the posterior

- Recipe:

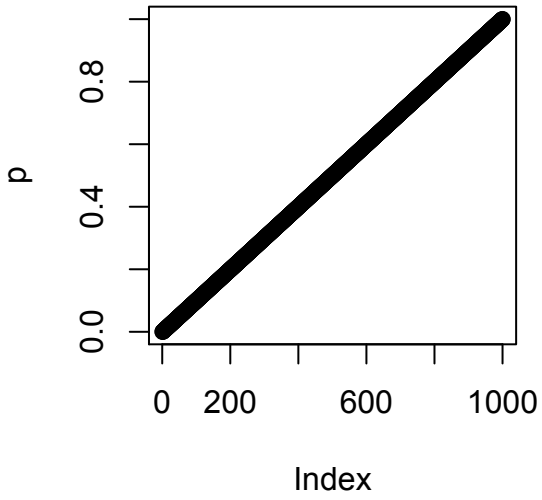
1. Compute or estimate posterior
2. Sample with replacement from posterior
3. Compute stuff from samples

Compute posterior

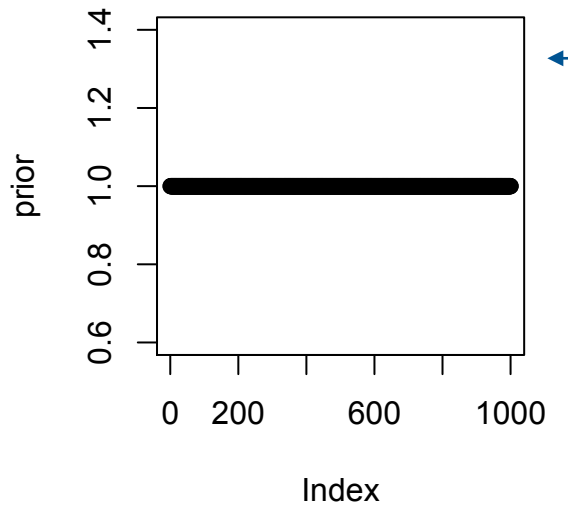
- Grid approximation

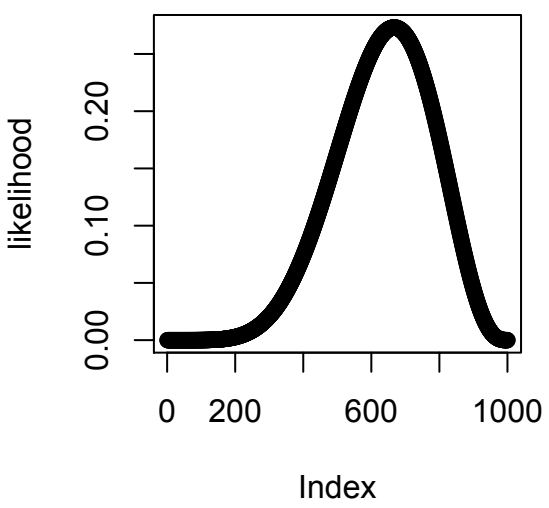
R code 3.2

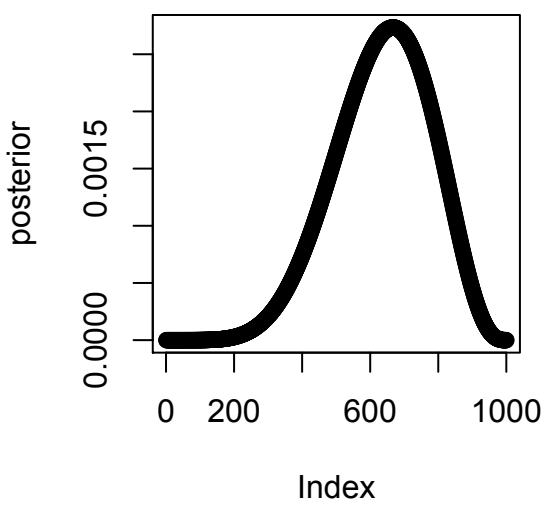
```
p <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 )
likelihood <- dbinom( 6 , size=9 , prob=p )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
```




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



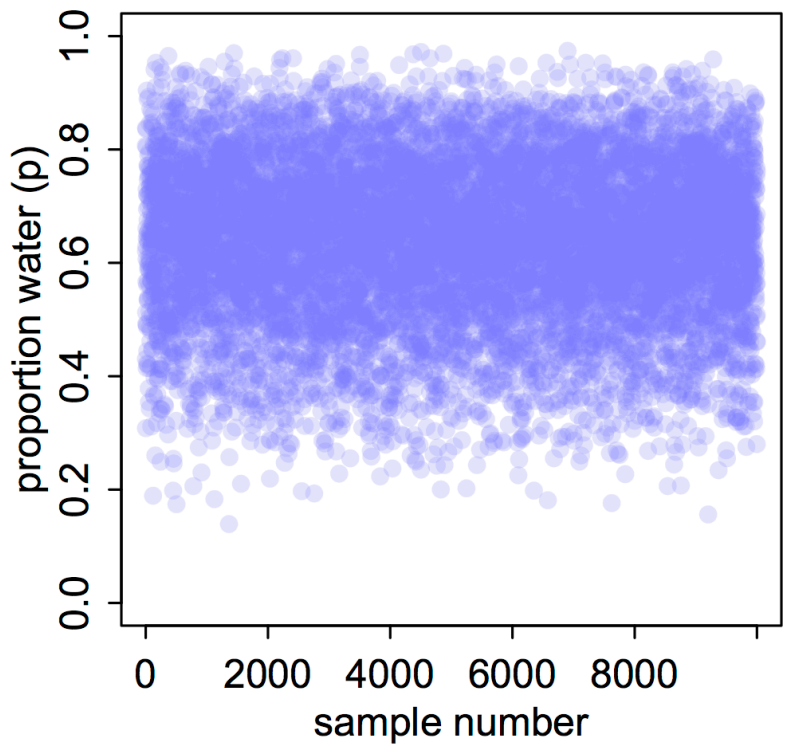




Sample from posterior

R code 3.3

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



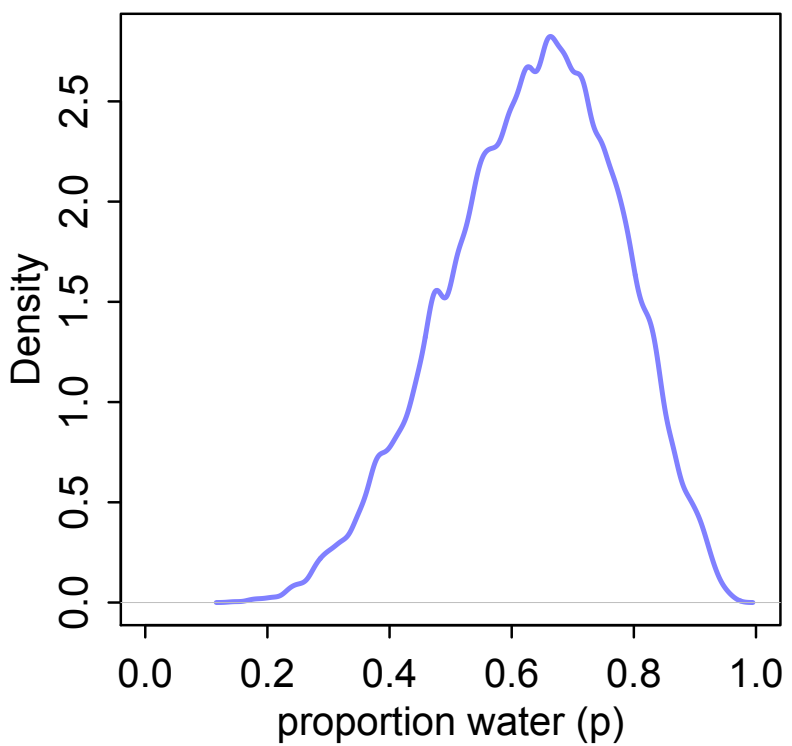


Figure 3.1

Compute stuff

- Summary tasks
  - How much posterior probability below/above/between specified parameter values?
  - Which parameter values contain 50%/80%/95% of posterior probability? "*Confidence" intervals*
  - Which parameter value maximizes posterior probability? Minimizes posterior loss? *Point estimates*
- You decide the question

Homework

- Practice problems at end of Chapter 3
- For certificate: Write up solutions to HARD problems: 3H1, 3H2, 3H3, 3H4, 3H5
  Turn them in to me next Friday (3 Nov)
- Next week: Rethinking regression (Chapter 4)