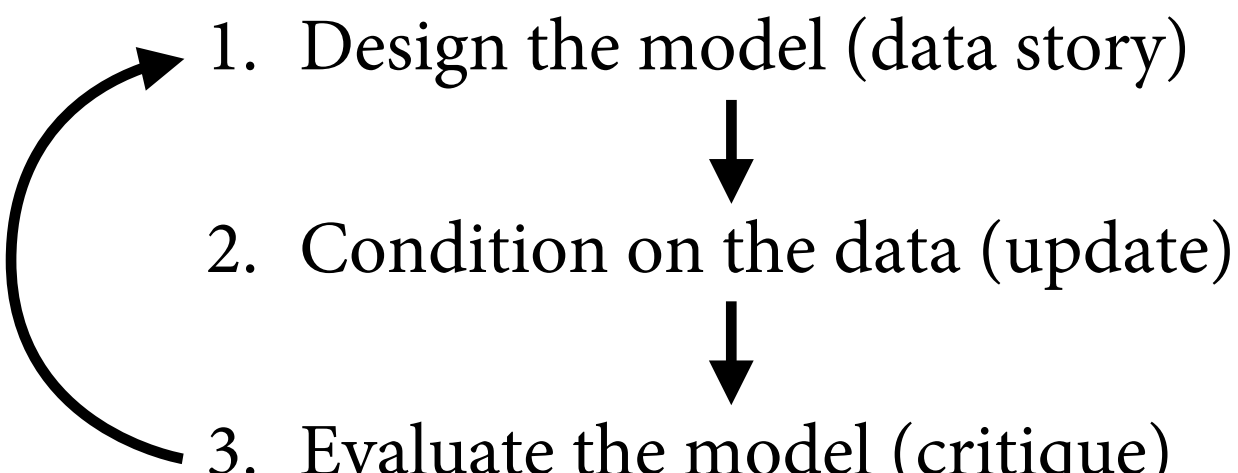
Statistical Rethinking
Winter 2019

Lecture 2 / Week 1

The Garden of Forking Data

Building a model

- How to use probability to do typical statistical modeling?



Nine tosses of the globe:

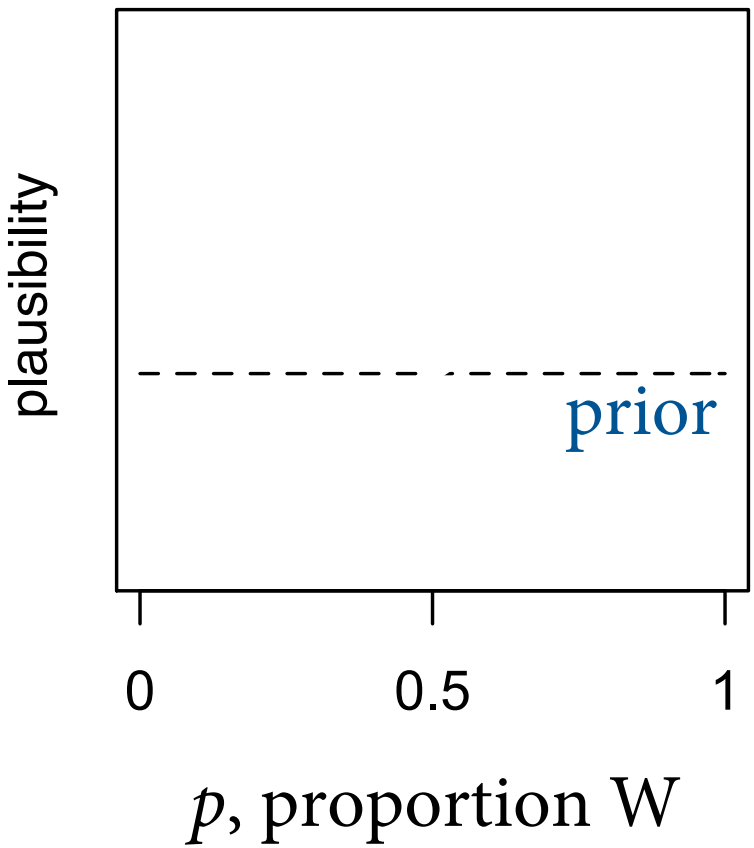
W L W W W L W L W

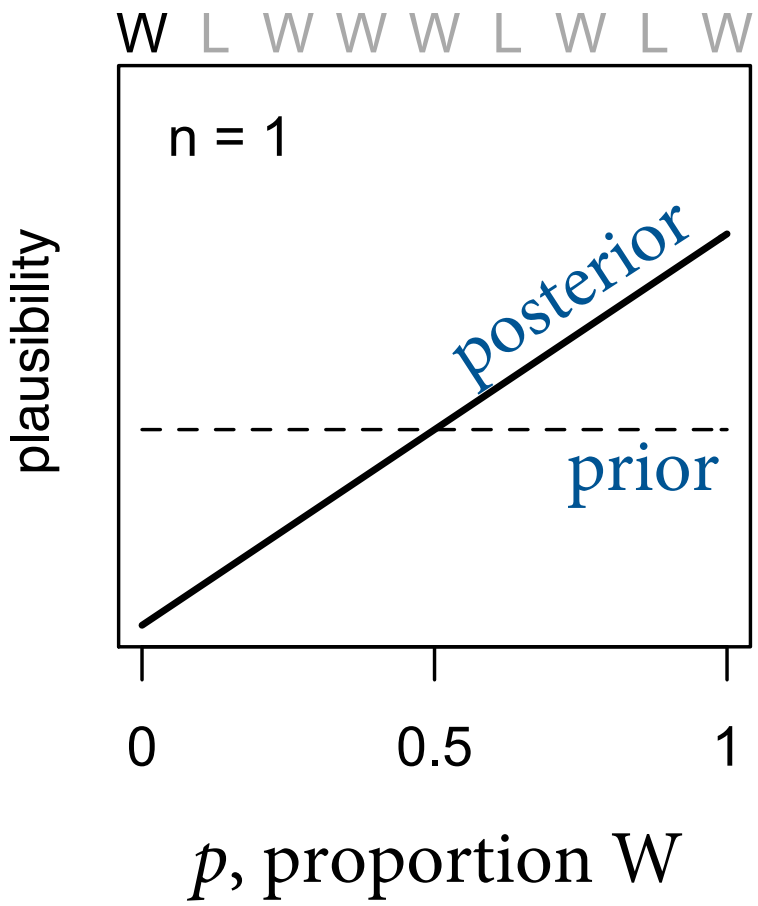
Design > Condition > Evaluate

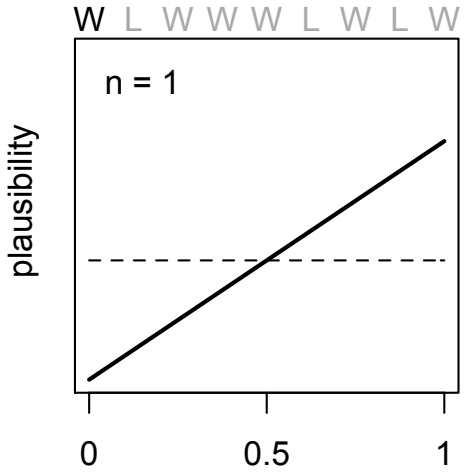
- Data story motivates the model
  - How do the data arise?
- For **W L W W W L W L W**:
  - Some true proportion of water, p
  - Toss globe, probability p of observing W, $1-p$ of L
  - Each toss therefore independent of other tosses
- Translate data story into probability statements

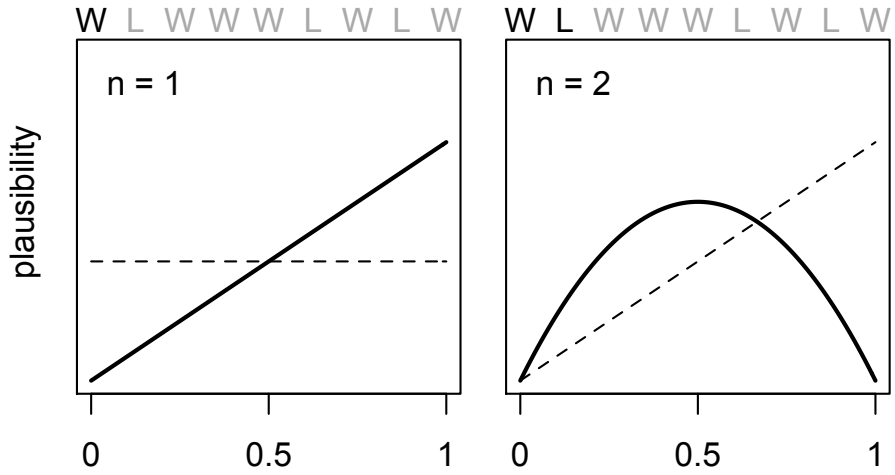
Design > **Condition** > Evaluate

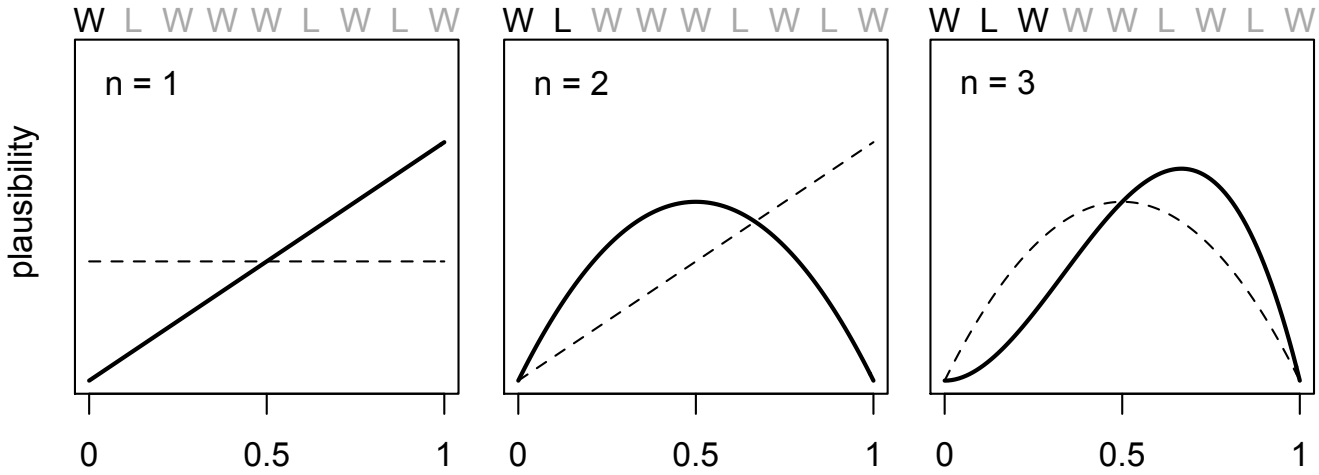
- *Bayesian updating* defines optimal learning in small world, converts *prior* into *posterior*
  - Give your golem an information state, before the data: Here, an initial confidence in each possible value of p between zero and one
  - Condition on data to update information state: New confidence in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

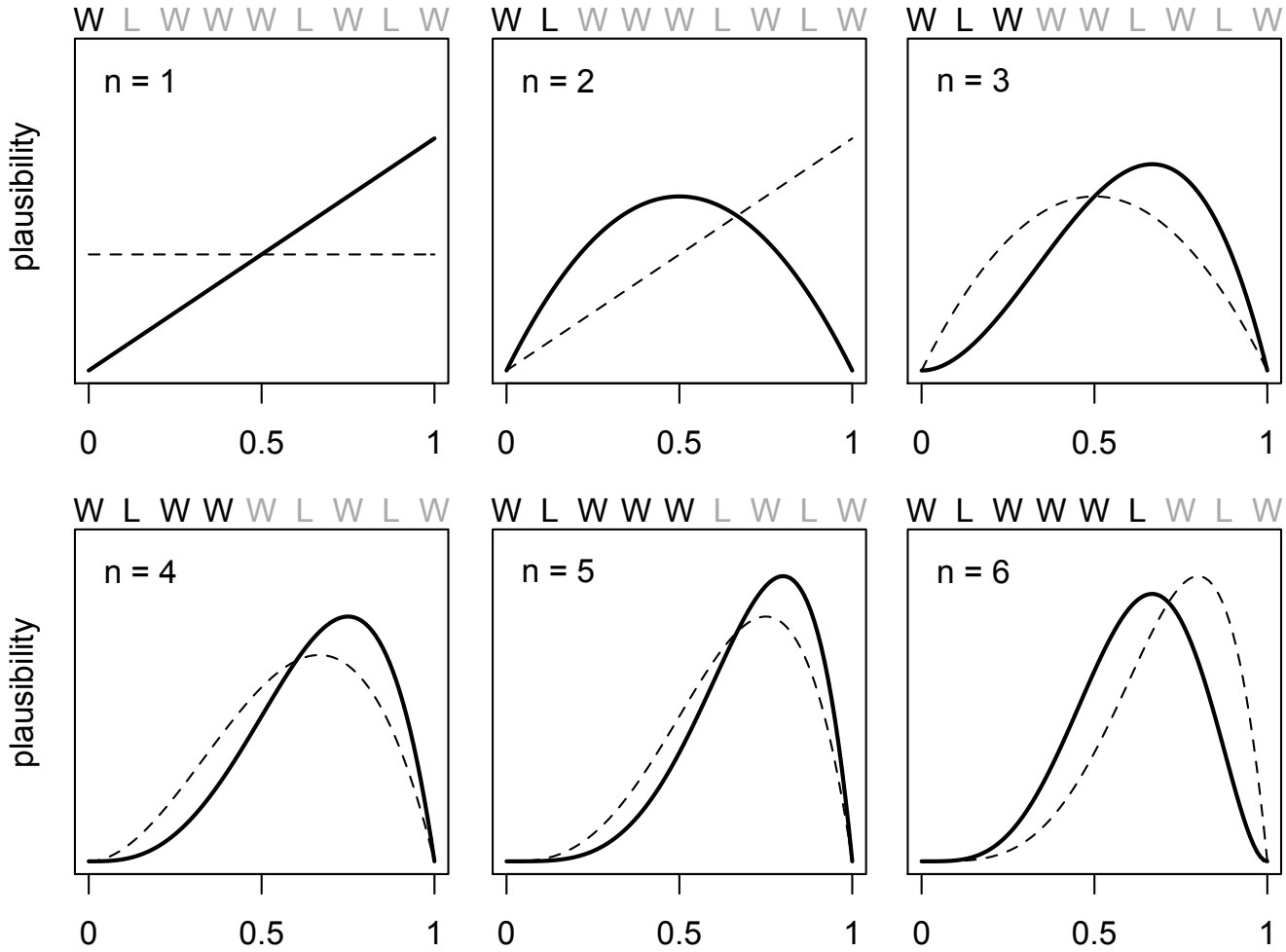

plausibility
prior
0
0.5
1
p, proportion W

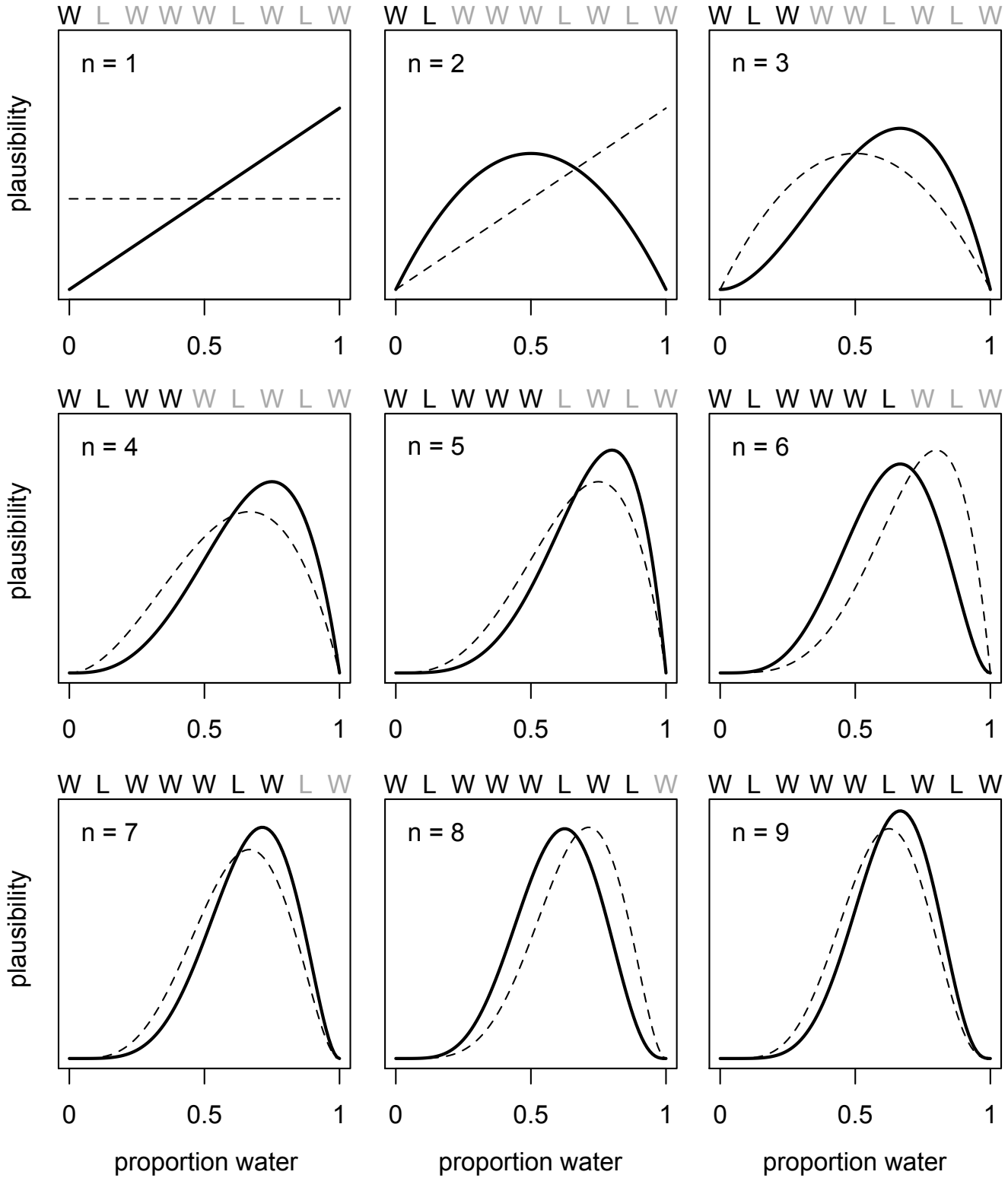

W L W W W L W L W
n = 1
plausibility
posterior
prior
0
0.5
1
p, proportion W


W L W W W L W L W
n = 1
plausibility
0
0.5
1


W L W W W L W L W
n = 1
plausibility
0
0.5
1
W L W W W L W L W
n = 2
0
0.5
1

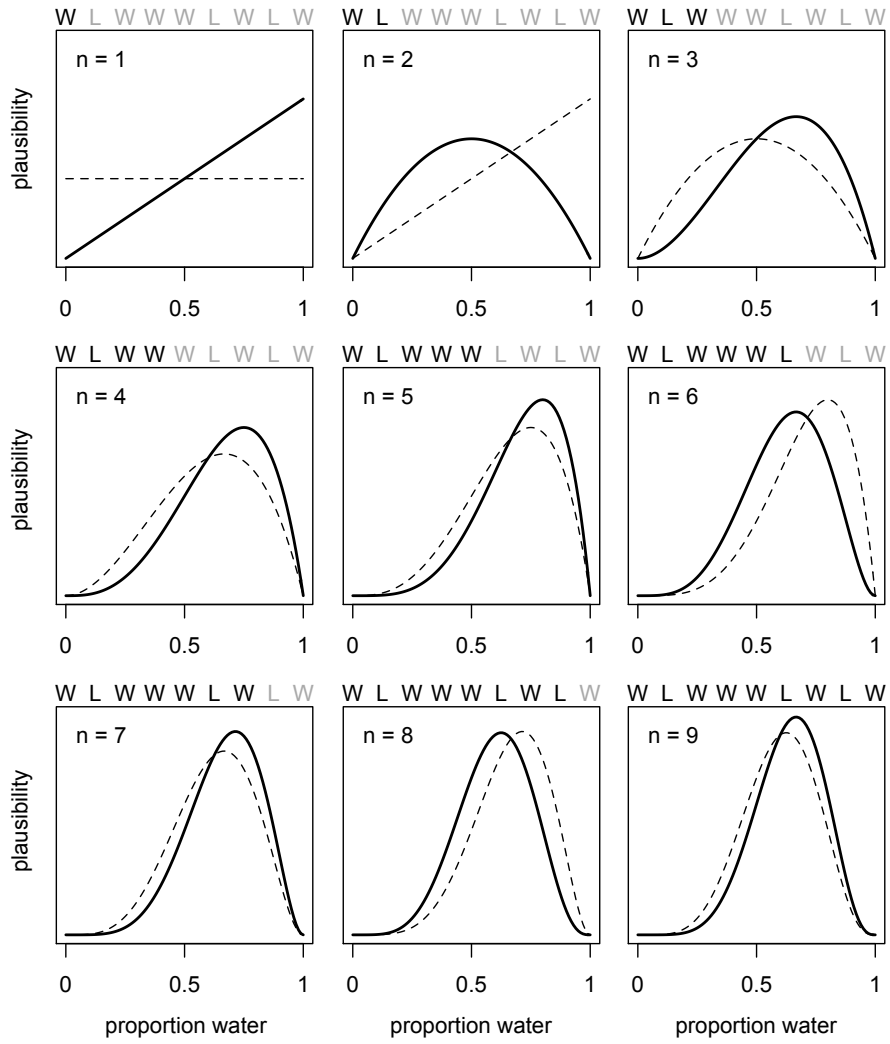

W L W W W L W L W
W L W W W L W L W
W L W W W L W L W
n = 1
n = 2
n = 3
plausibility
0
0.5
1
0
0.5
1
0
0.5
1


W L W W W L W L W
n = 1
n = 2
n = 3
n = 4
n = 5
n = 6
plausibility
plausibility
0
0.5
1


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
- Sample size automatically embodied in posterior



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?"

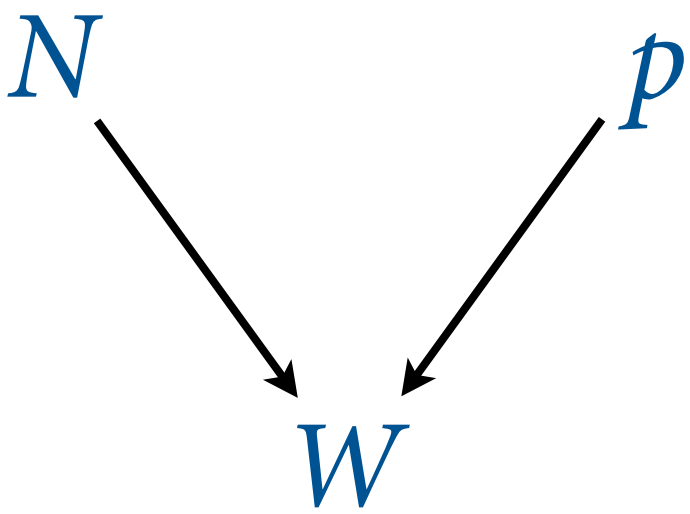
- Golem must be supervised
  - Did the golem malfunction?
  - Does the golem's answer make sense?
  - Does the question make sense?
  - Check sensitivity of answer to changes in assumptions

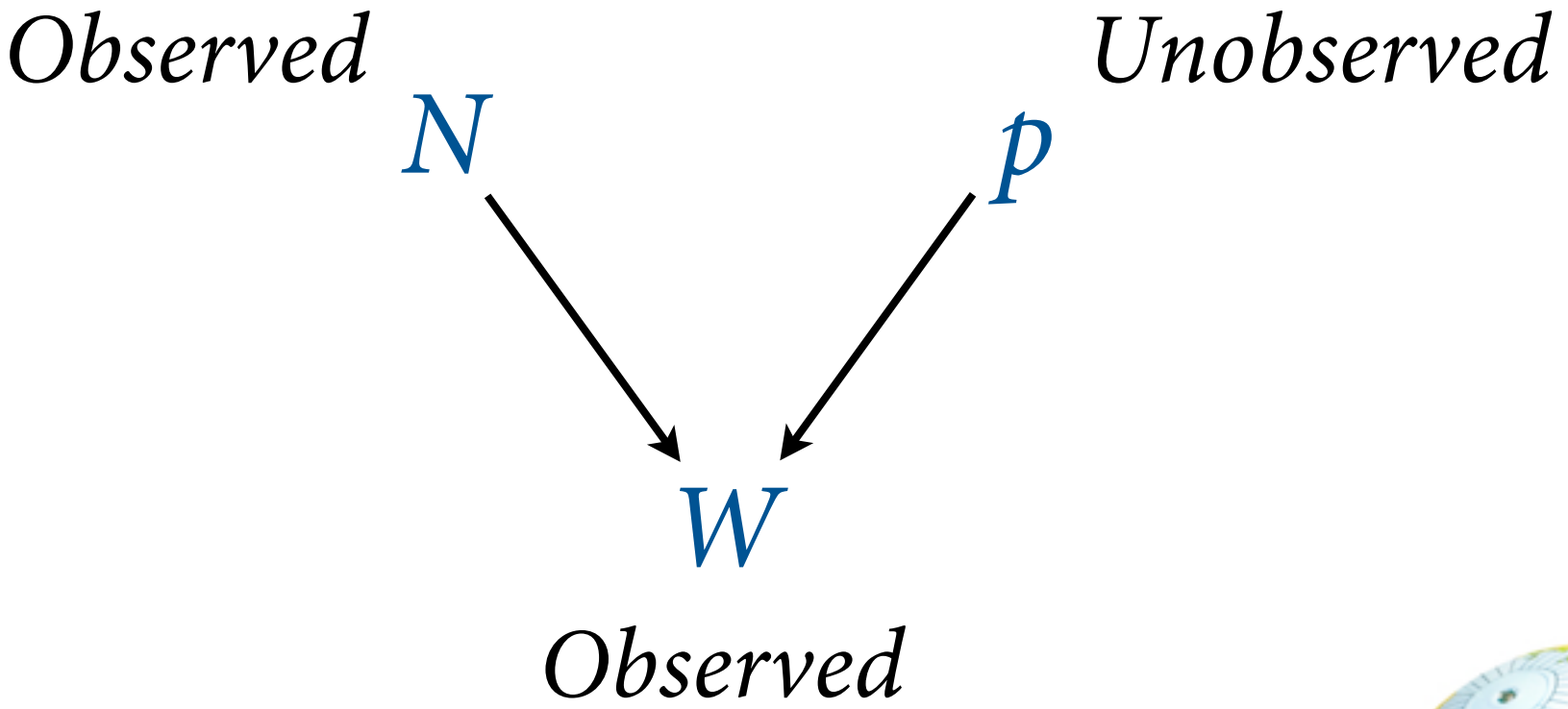
Construction perspective

- Build joint model:
  (1) List variables
  (2) Define generative relations
  (3) ???
  (4) Profit

- Input: Joint prior

- Deduce: Joint posterior


N
p
W

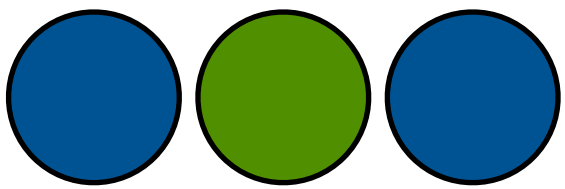

Observed
Unobserved
N
p
W
Observed

Definition of W

- Relative number of ways to see W, given N and p?
- Goal: Mathematical function to answer this question.
- The answer is a *probability distribution*.

Definition of *W*

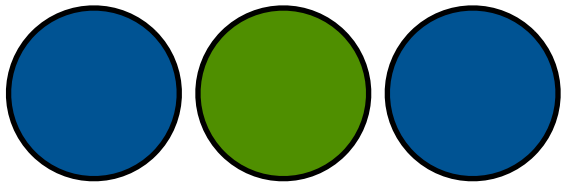
- Relative number of ways to see *W*, given *N* and *p*?



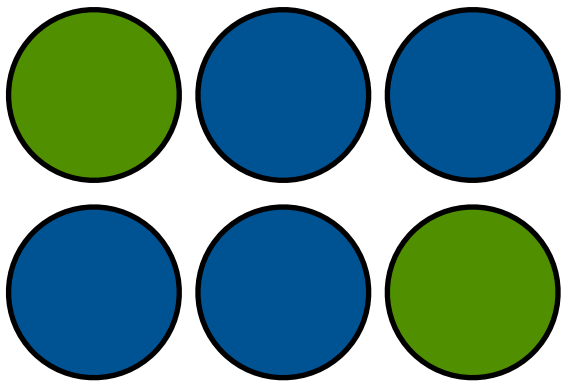
$$p \times (1-p) \times p \quad = p^2(1-p)^1$$

Definition of *W*

- Relative number of ways to see *W*, given *N* and *p*?

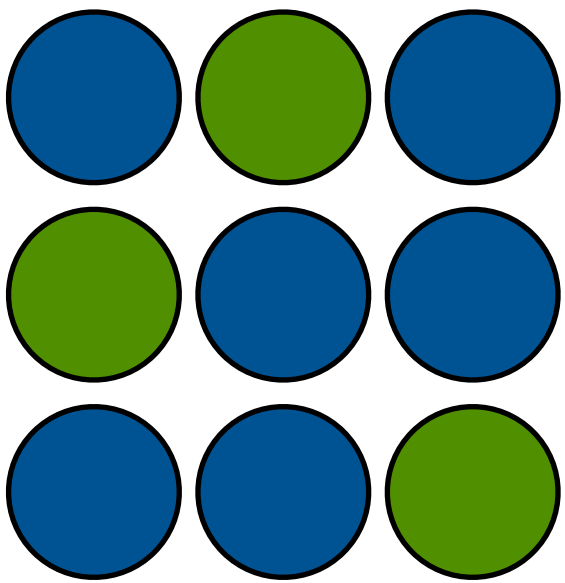


$$p \times (1-p) \times p \quad = p^2(1-p)^1$$



Definition of *W*

- Relative number of ways to see *W*, given *N* and *p*?



$$\Pr(2|3,p) = 3p^2(1-p)^1$$

W distribution (Likelihood)

number tosses

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

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.

W distribution (Likelihood)

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

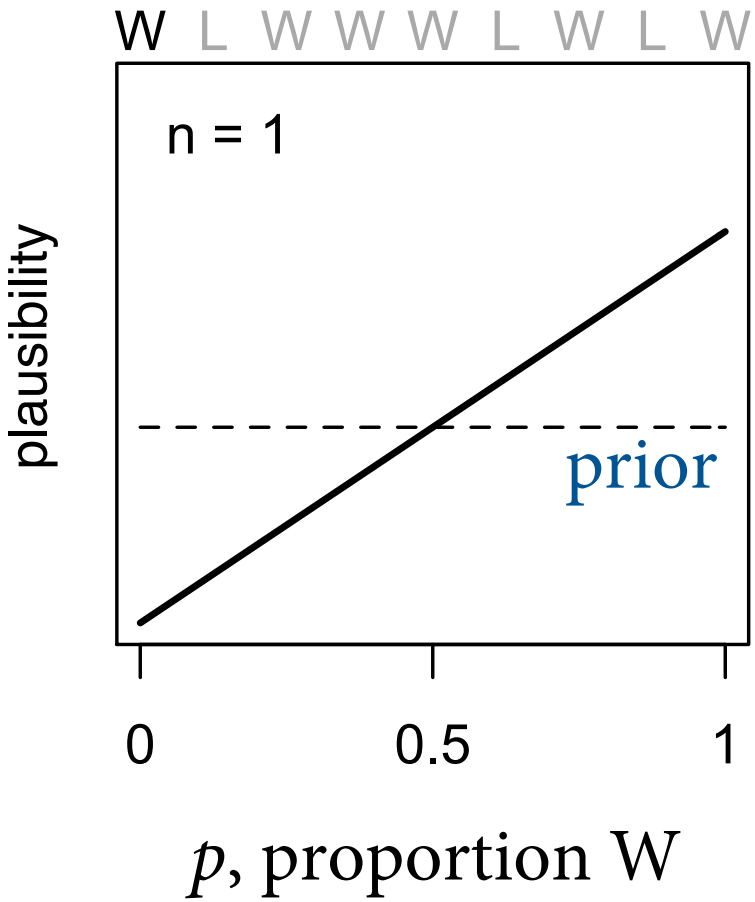
```
dbinom( 6 , size=9 , prob=0.5 )
```

R code 2.2

```
[1] 0.1640625
```

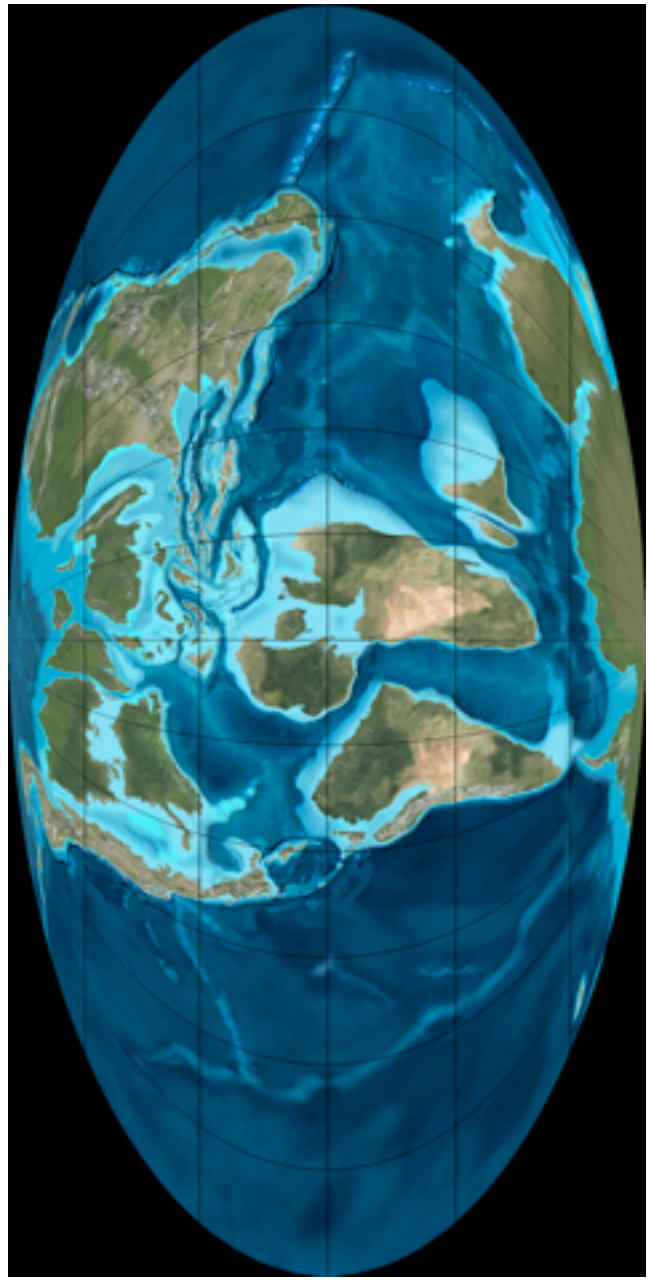
Prior probability *p*

- What the golem believes before the data arrive
- In this case, equal prior probability 0–1
- Pr(*W*) & Pr(*p*) define *prior predictive distribution*
- More on this later – it helps us build priors that make sense



Prior literature

- Huge literature on choice of prior
- Flat prior conventional & bad
  - 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 than flat
- All above equally true of likelihood



Late Cretaceous (90Mya)

The Joint Model

$$W \sim \text{Binomial}(N, p)$$

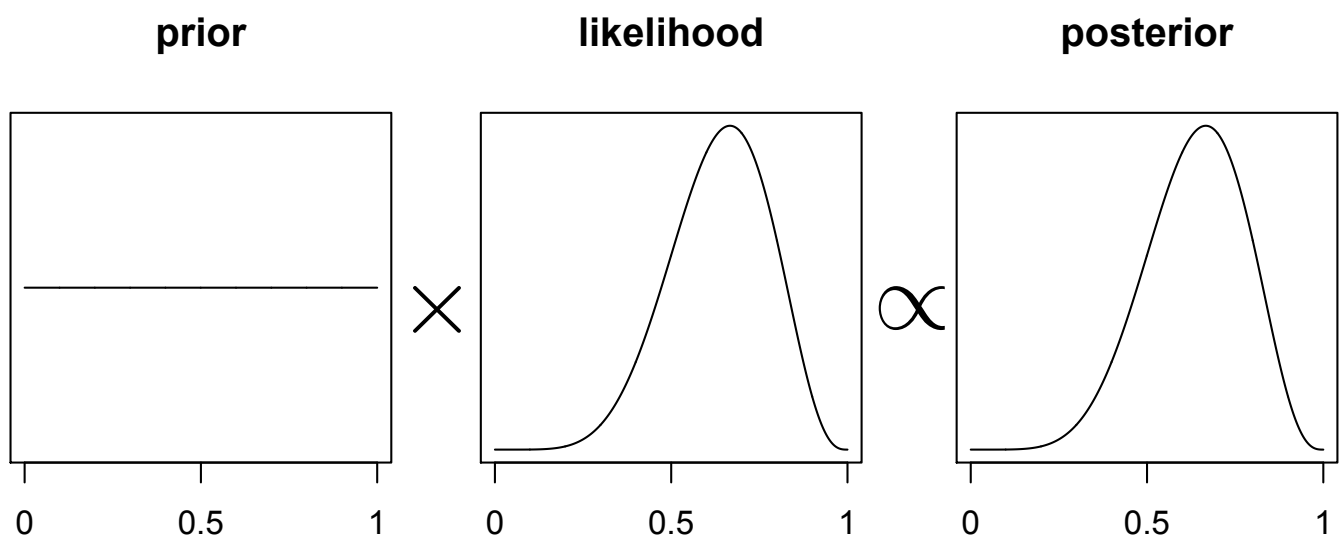
$$p \sim \text{Uniform}(0, 1)$$

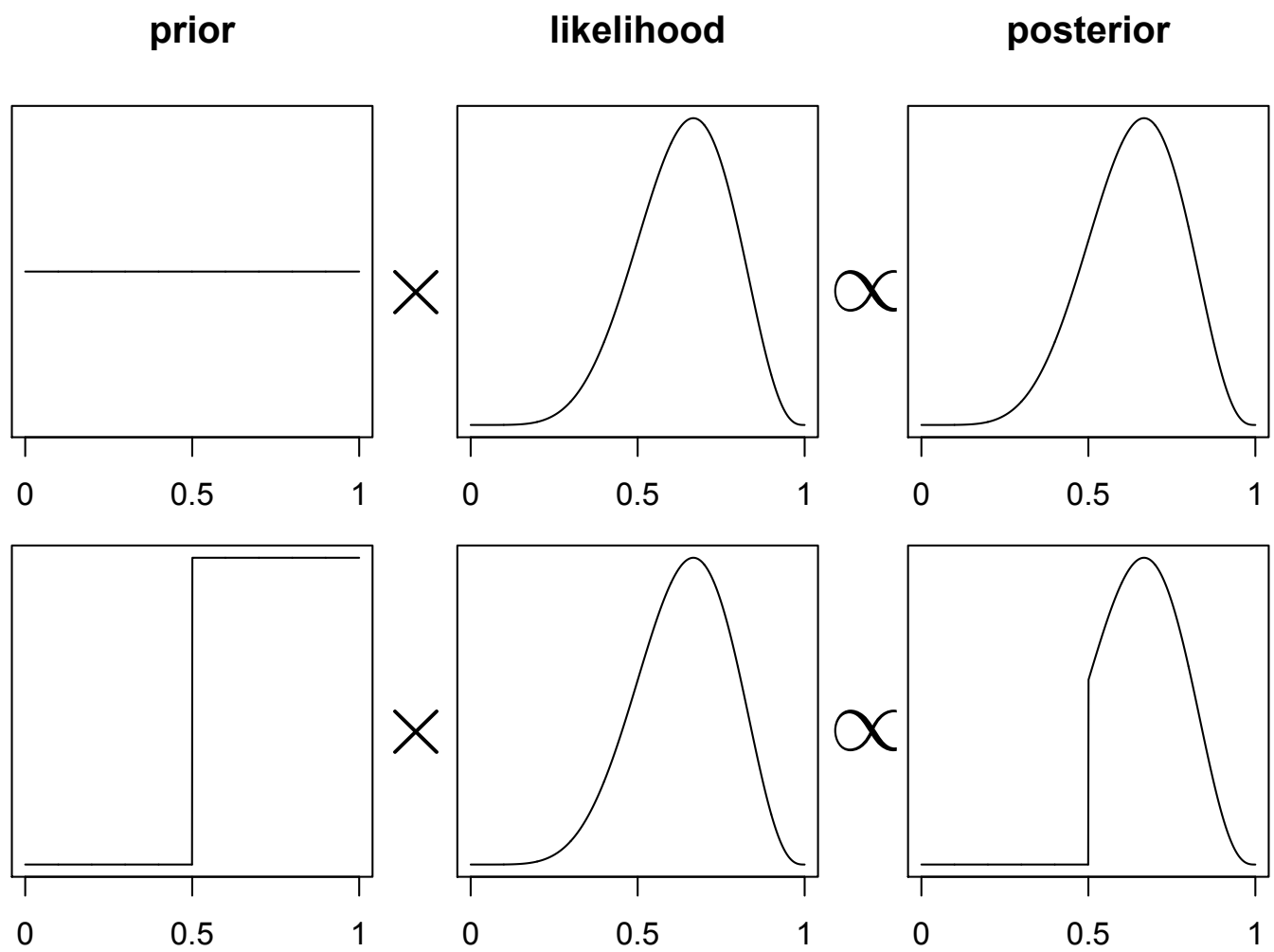
Posterior probability

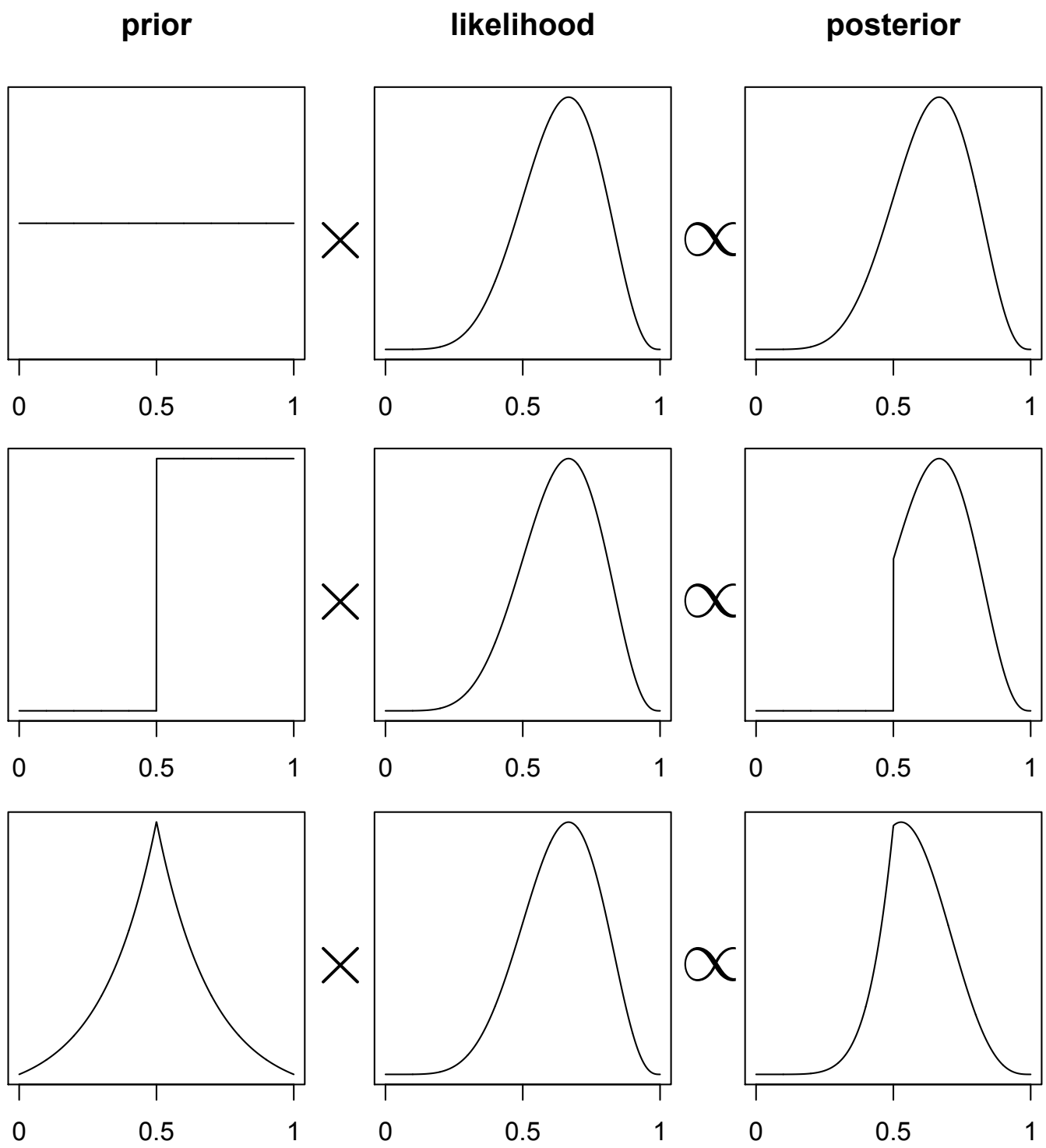
- Bayesian "estimate" is always *posterior distribution over parameters*, Pr(parameters|data)
- Here: Pr($p|W,N$)
- Compute using *Bayes' theorem*:

$$\Pr(p|W,N) = \frac{\Pr(W|N,p)\Pr(p)}{\sum \Pr(W|N,p)\Pr(p) \text{ for all } p}$$

$$\text{Posterior} = \frac{(\text{Prob observed variables}) \times (\text{Prior})}{\text{Normalizing constant}}$$

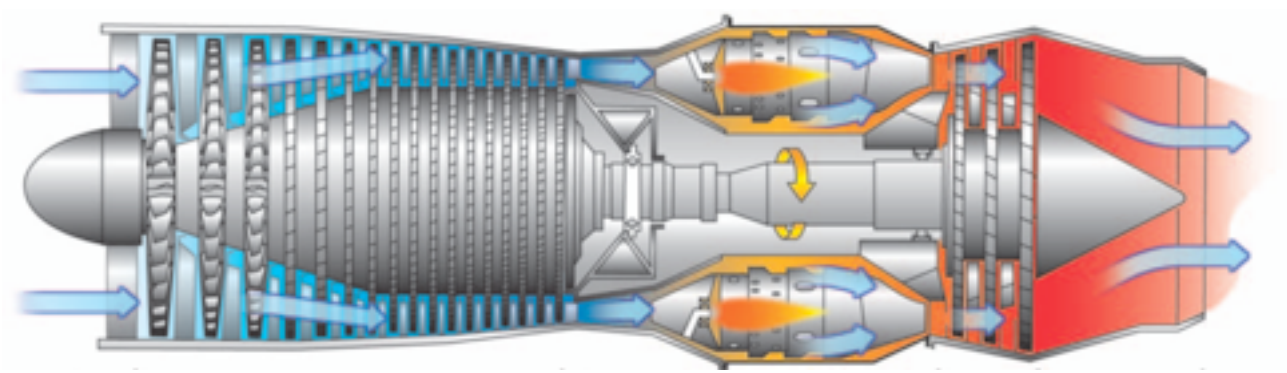

prior
likelihood
posterior
0
0.5
1
0
0.5
1
0
0.5
1


prior
likelihood
posterior
0
0.5
1
0
0.5
1
0
0.5
1
0
0.5
1
0
0.5
1
0
0.5
1


prior
likelihood
posterior
0
0.5
1

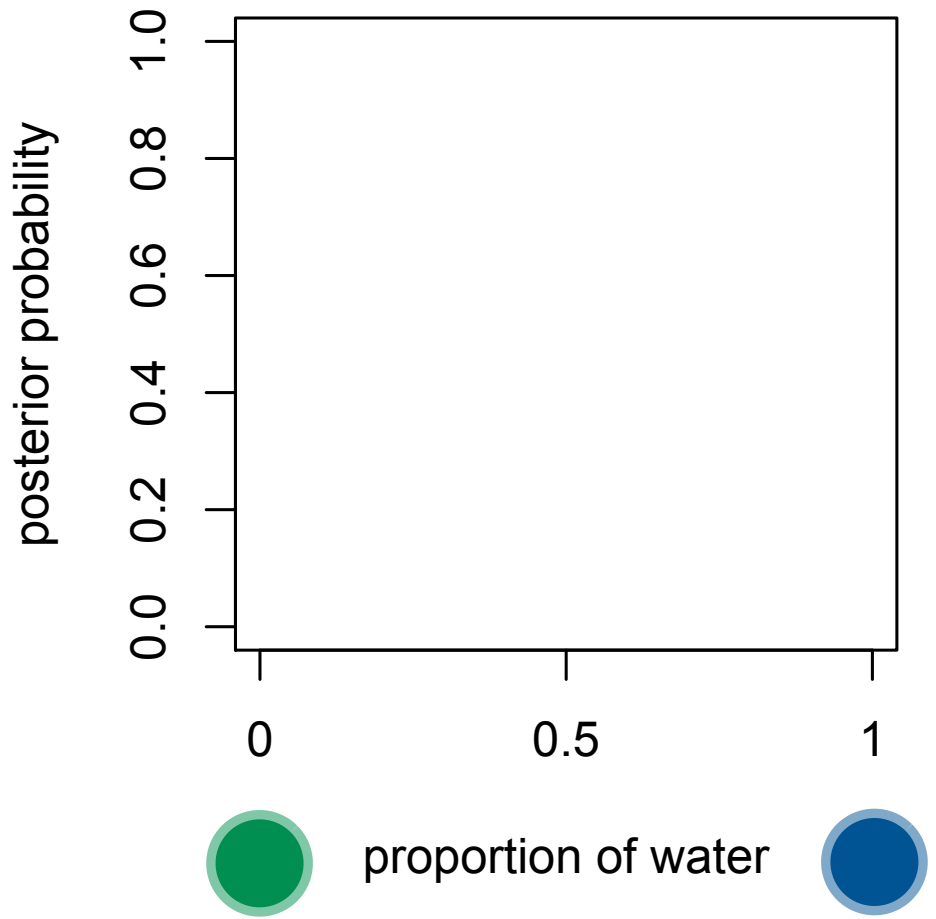
Computing the posterior

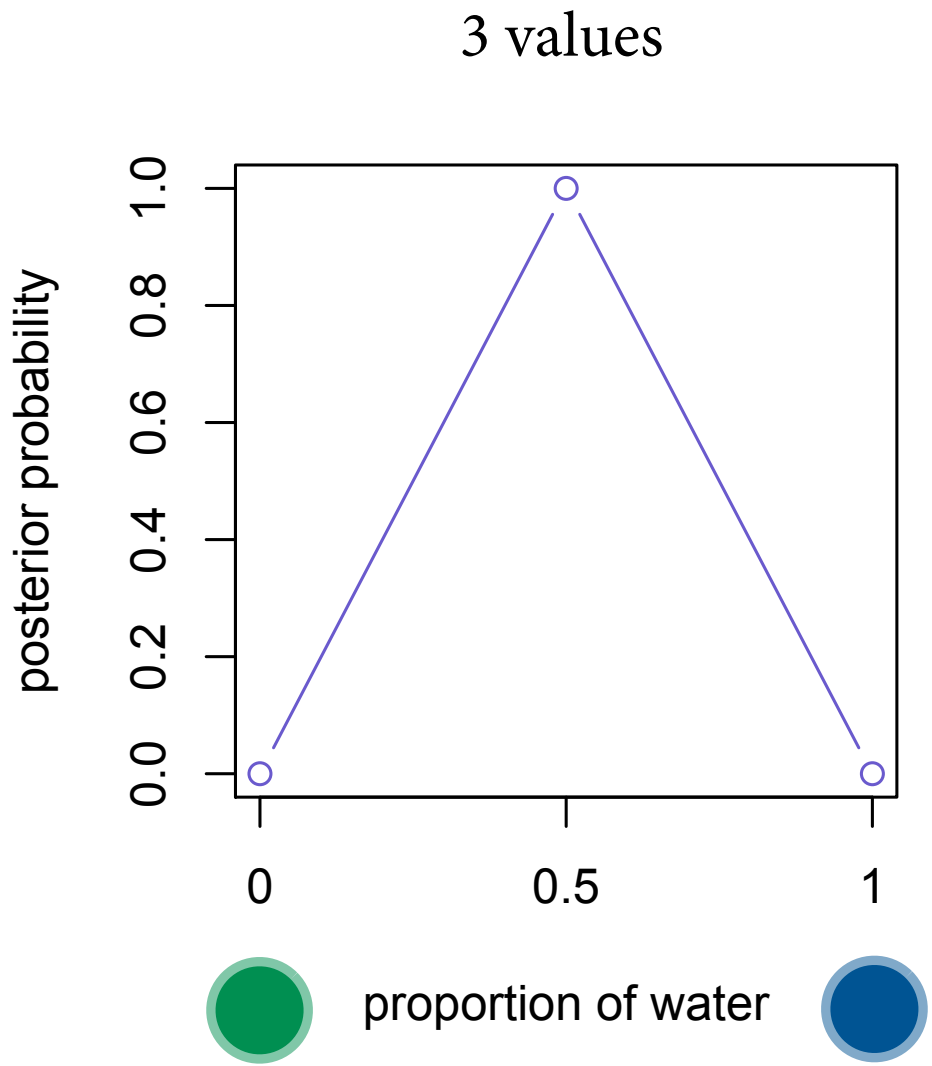
1. Analytical approach (often impossible)
2. Grid approximation (very intensive)
3. Quadratic approximation (limited)
4. Markov chain Monte Carlo (intensive)

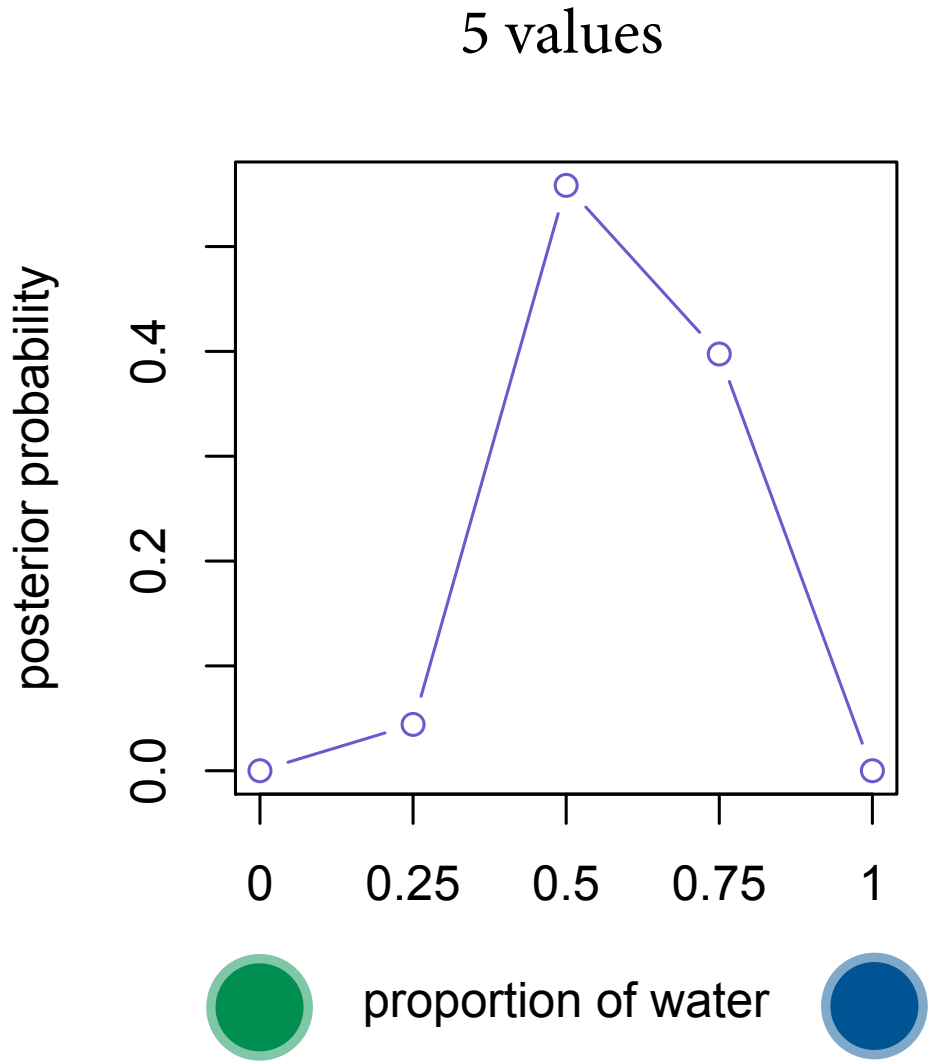


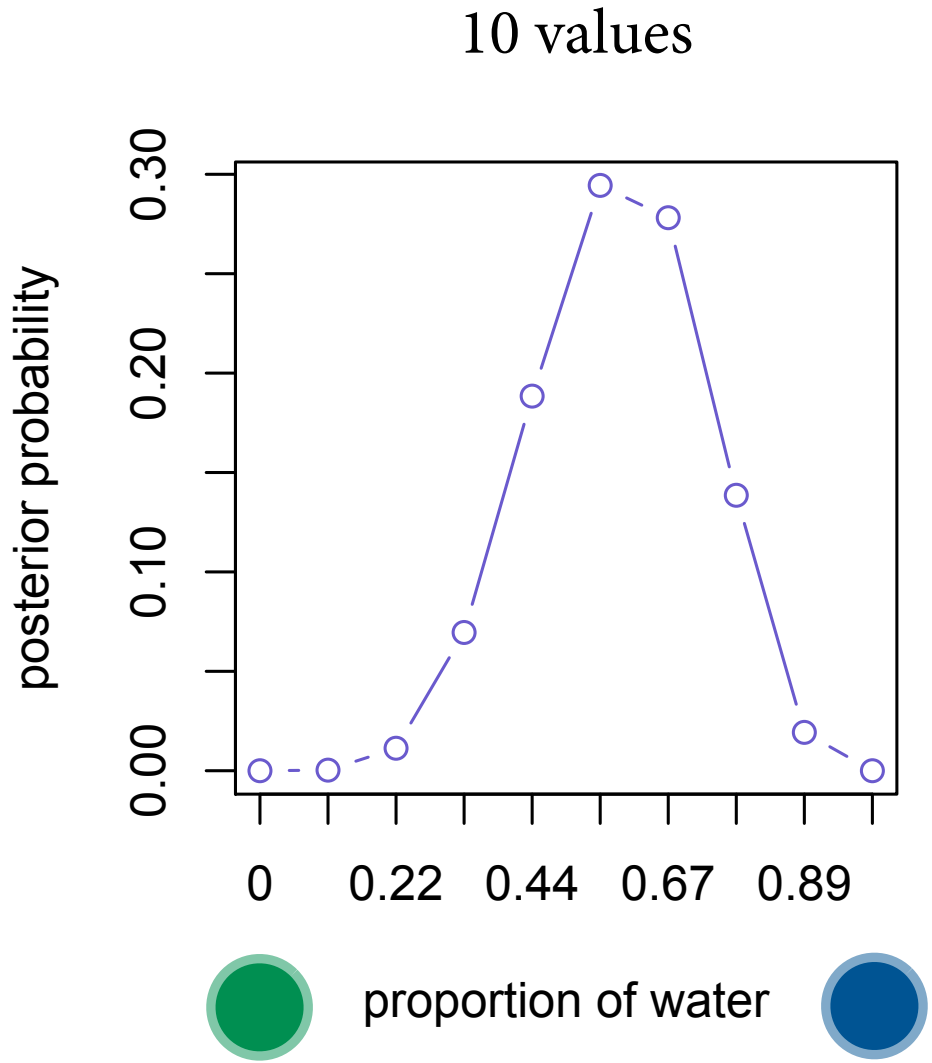
Grid approximation

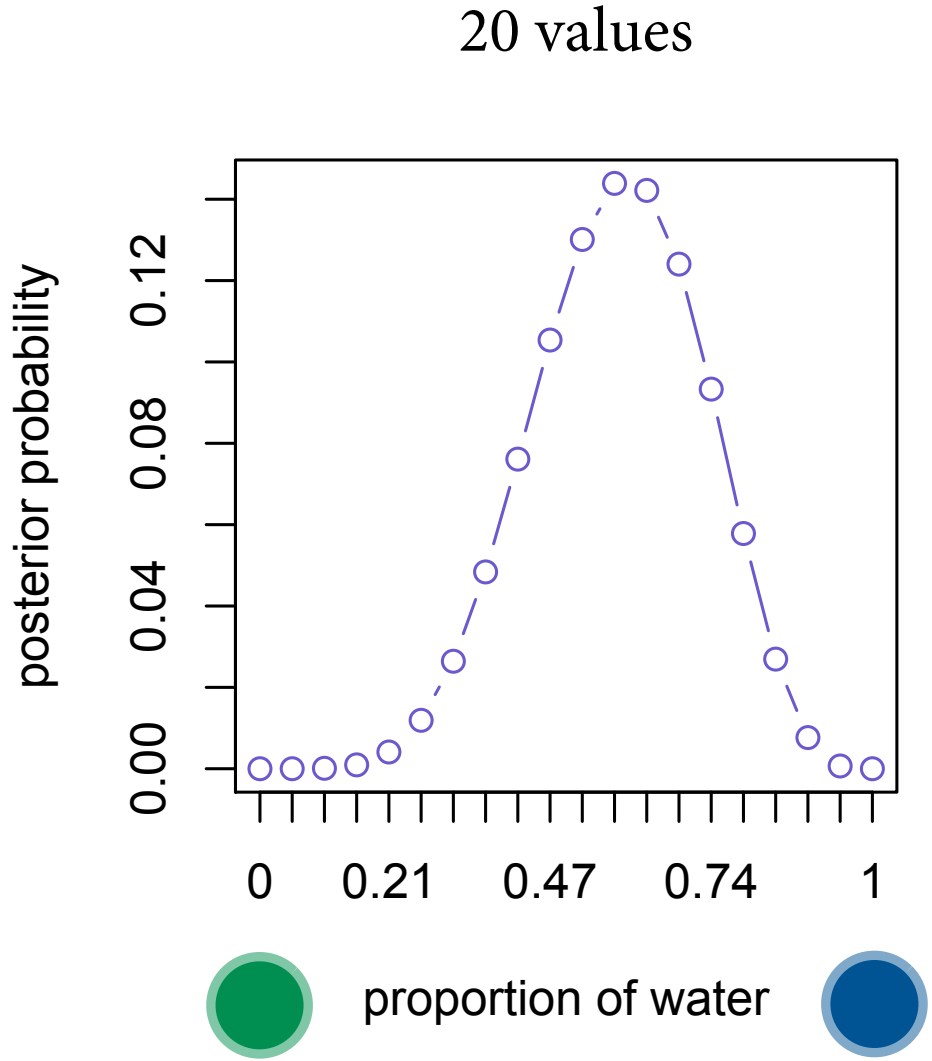
- The posterior probability is:
  standardized product of
  (1) probability of the data
  (2) prior probability
- "Standardized" means: Add up all the products and divide each by this sum
- Grid approximation uses *finite grid* of parameter values instead of continuous space
- Too expensive with more than a few parameters

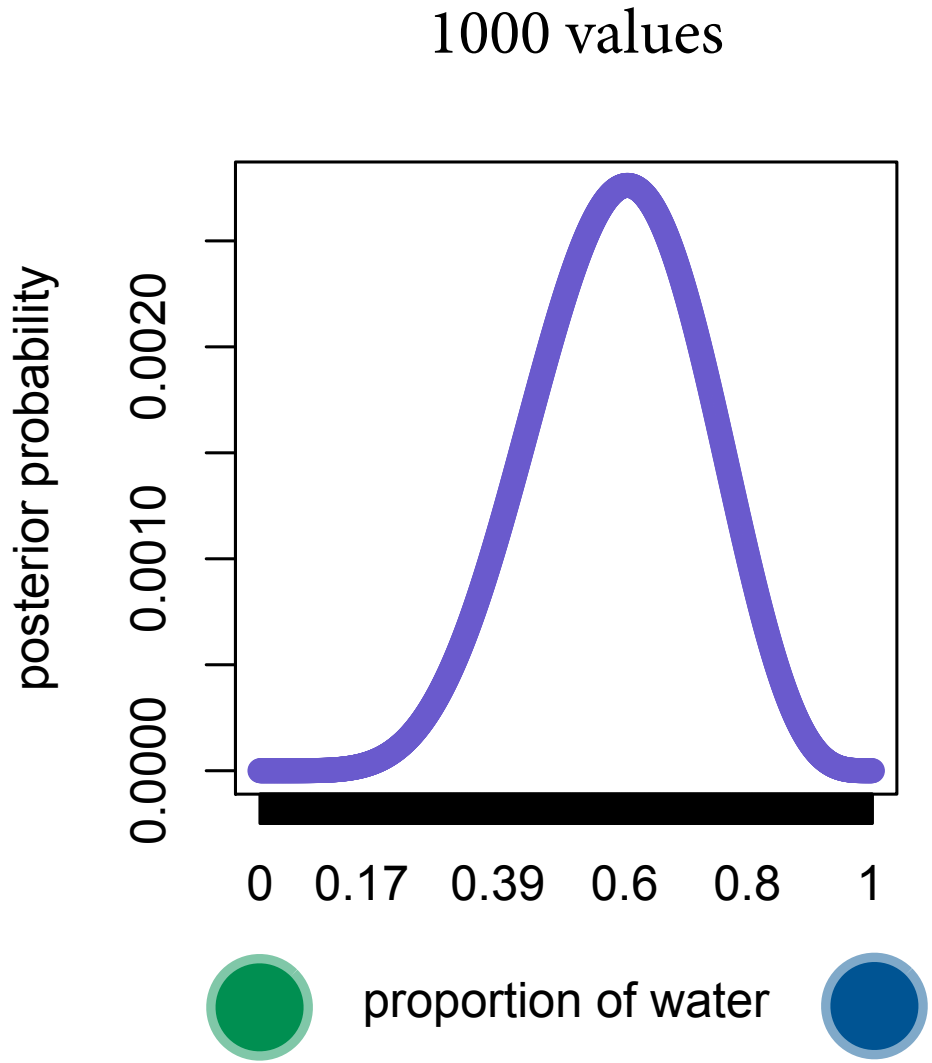

posterior probability
0.0
0.2
0.4
0.6
0.8
1.0
0
0.5
1
proportion of water


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

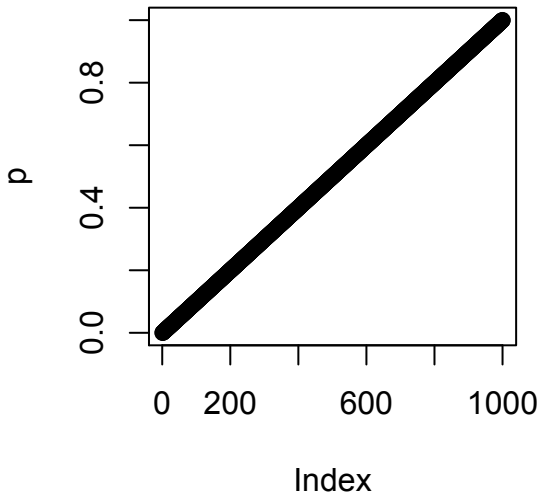
Compute posterior

- Grid approximation

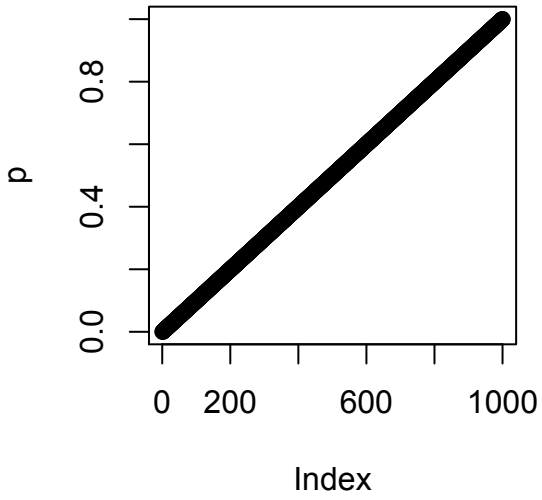
R code 3.2

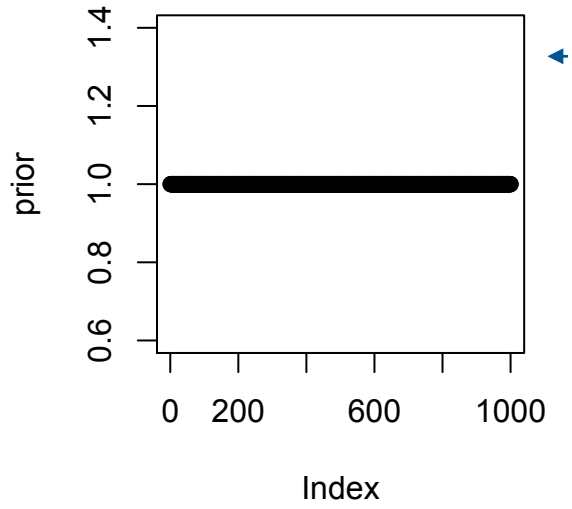
```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```

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

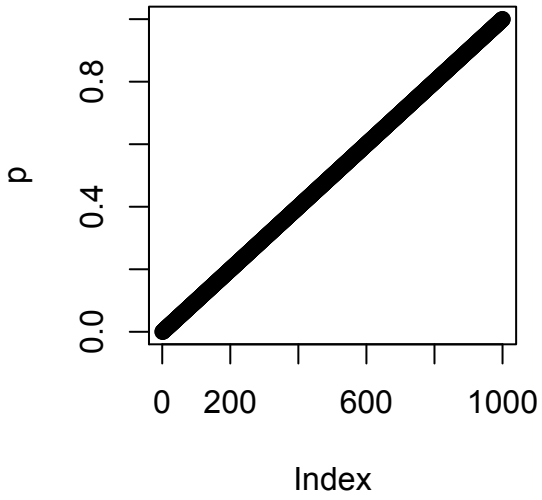



```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```

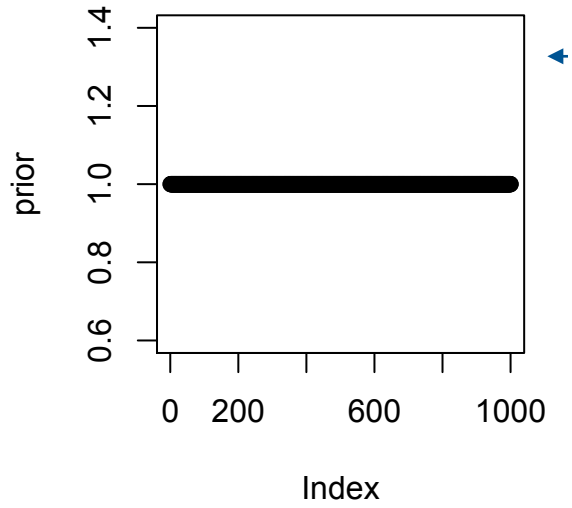


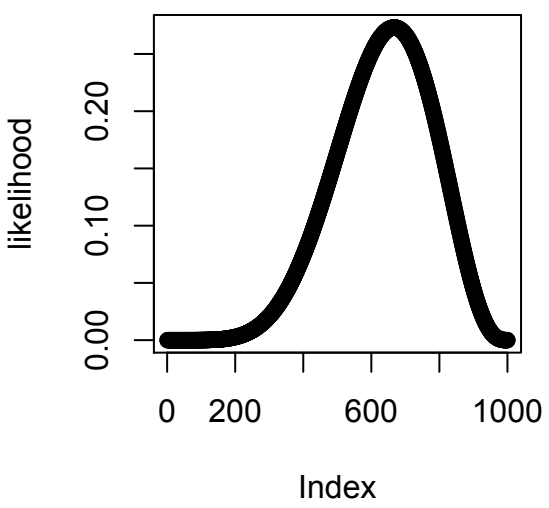


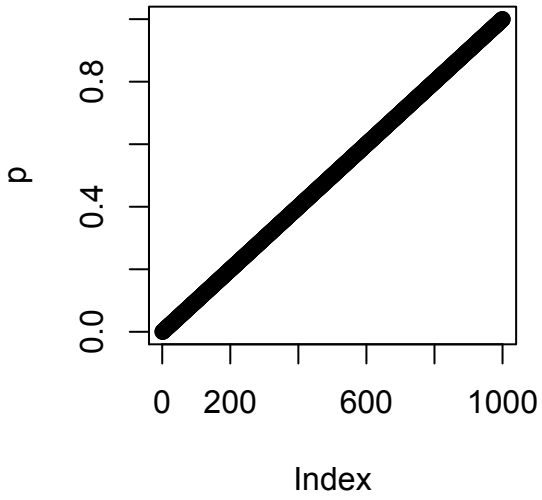

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```



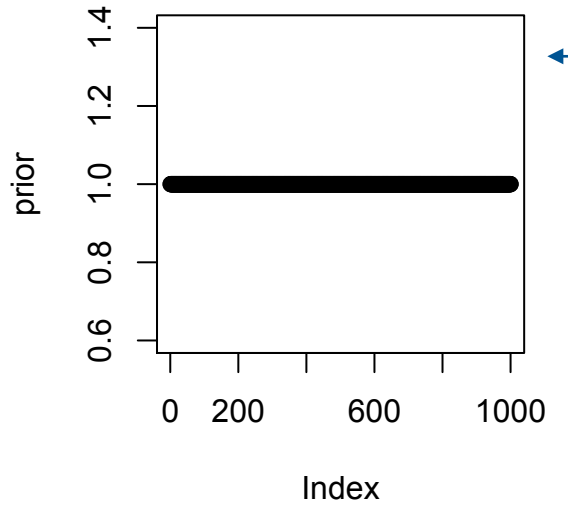

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```

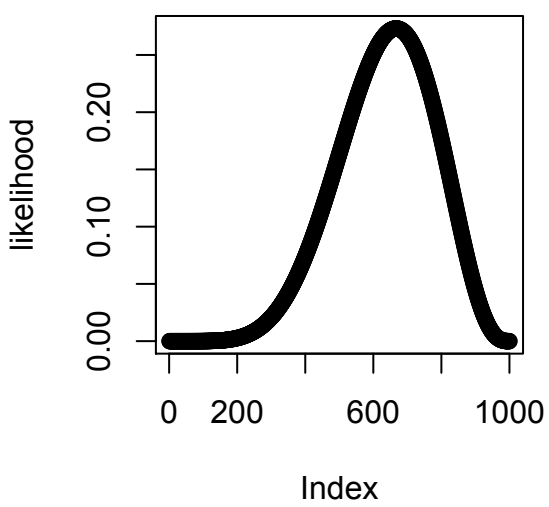







```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```





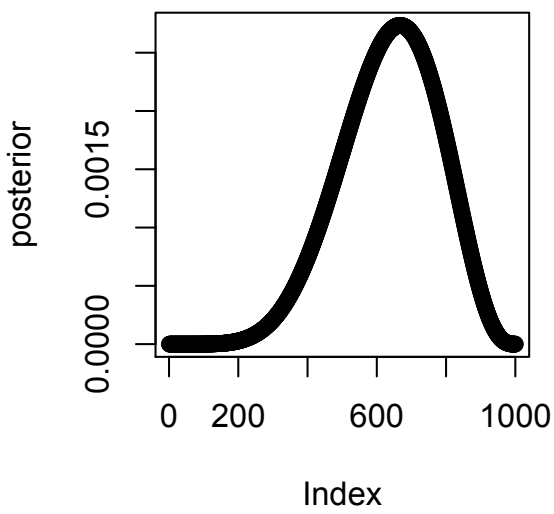


Table of random numbers 1 to 100 with no numbers repeated in each block of 25.

75 64 26 45 10	79 18 58 61 09	67 05 60 19 91	14 62 02 35 98	88 51 53 56 96
24 05 89 42 27	98 62 31 19 95	24 25 58 50 49	19 30 31 58 59	49 47 85 48 30
63 18 80 72 41	26 11 91 96 81	55 92 44 23 93	97 89 53 40 80	29 46 34 39 63
38 81 93 68 22	84 92 59 82 80	26 94 73 71 45	63 84 68 44 94	93 64 13 94 31
25 59 54 43 02	16 41 97 40 65	70 29 77 74 27	69 81 70 01 95	82 99 77 80 21
12 28 15 88 98	21 28 92 06 08	33 72 05 13 06	85 65 33 90 20	92 33 27 59 49
36 59 95 67 96	25 72 30 41 81	71 92 18 65 17	64 58 56 89 28	69 18 36 06 71
91 72 33 68 11	22 20 15 01 65	34 60 47 16 09	44 45 46 97 83	44 51 98 67 29
86 04 47 43 69	12 85 04 93 74	80 08 57 25 79	72 96 07 57 40	82 62 68 60 73
01 05 65 97 77	96 64 98 62 49	07 19 63 46 66	77 98 80 54 60	97 32 83 74 80
26 95 96 93 87	17 59 90 35 94	73 68 03 27 29	49 64 66 14 65	57 24 45 76 39
45 27 71 62 05	71 18 32 42 91	25 66 46 49 71	67 11 25 23 12	41 47 99 66 01
74 07 90 20 25	05 52 65 84 92	87 57 95 37 83	85 45 22 56 26	10 28 04 88 49
77 99 91 43 02	96 06 07 36 68	17 48 06 09 84	31 86 91 87 96	63 87 32 33 70
75 53 35 46 41	21 95 85 61 46	94 18 78 39 47	19 60 48 15 59	68 79 42 09 67
45 65 84 36 28	48 33 82 62 71	74 48 75 92 34	32 94 26 70 88	35 50 19 97 52
81 74 60 90 46	13 51 24 54 55	45 54 12 90 99	44 68 86 71 58	27 51 81 11 77
95 11 96 85 83	93 53 74 52 97	79 53 21 41 44	45 81 02 38 07	38 07 80 89 56
29 40 82 33 86	67 95 43 41 89	05 52 17 31 13	82 61 78 57 40	84 39 57 63 78
79 14 32 21 09	32 27 02 70 20	61 47 24 42 76	77 27 99 36 15	36 98 08 40 53
51 46 23 17 11	93 35 70 37 86	26 23 64 88 17	17 78 95 93 83	65 23 90 78 55
98 75 60 99 89	91 18 20 27 74	31 82 01 32 97	97 43 21 87 82	33 28 10 56 98
15 97 42 56 79	08 58 79 40 31	37 19 20 58 41	41 86 66 54 45	08 76 89 86 32
06 16 35 93 26	36 97 26 17 71	74 95 89 08 50	50 62 48 46 26	24 95 93 01 64
54 43 55 21 74	47 59 75 03 57	63 38 02 51 77	77 76 65 08 92	72 29 35 06 85

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*
- Transforms a hard calculus problem into an easy data summary problem

Sampling from the posterior

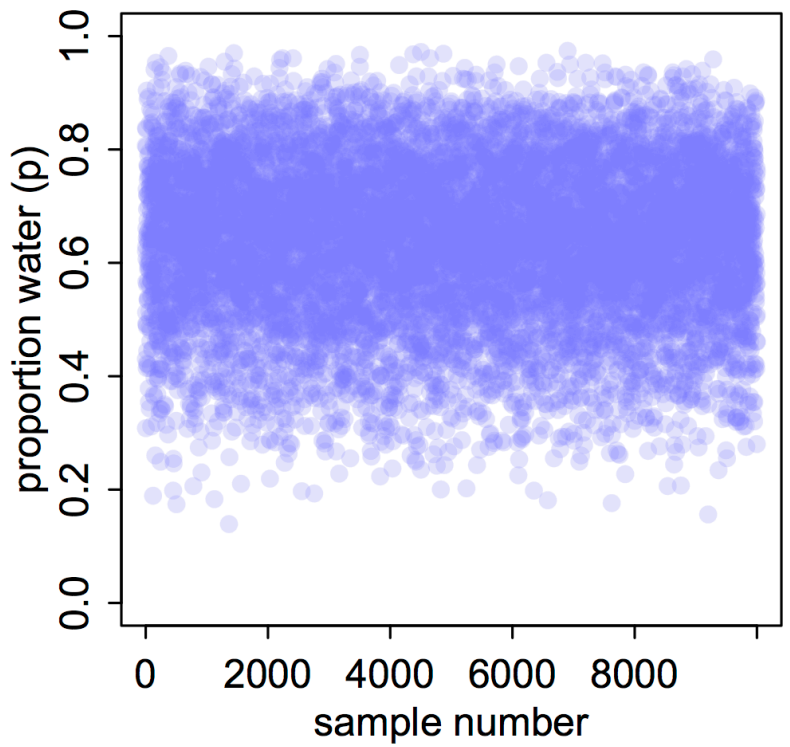
- Recipe:

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

Sample from posterior

R code 3.3

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



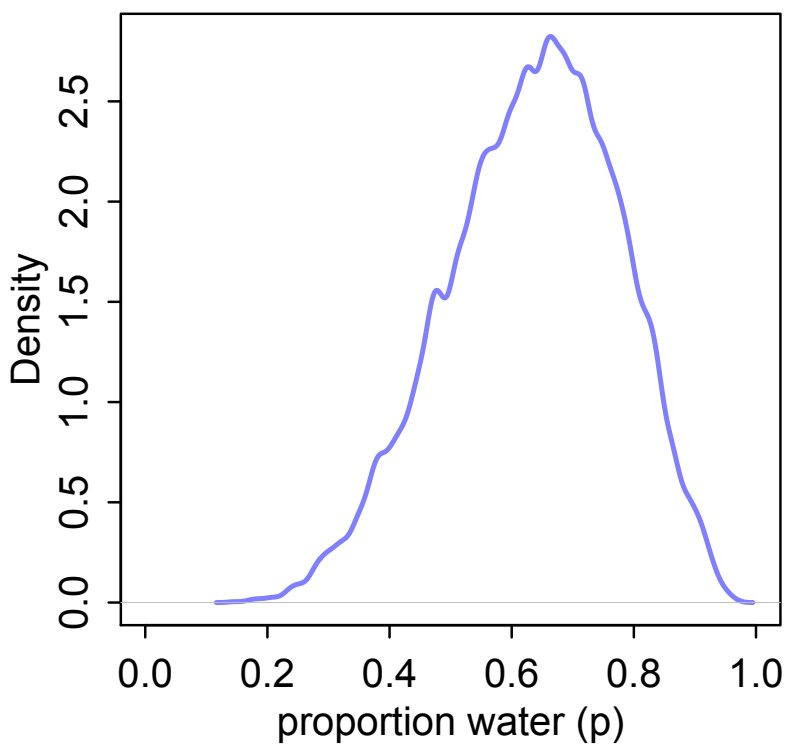
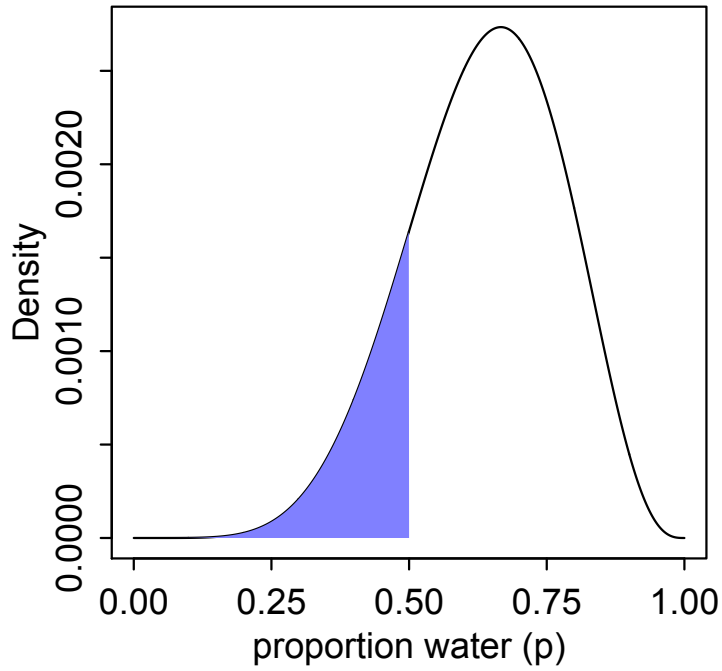


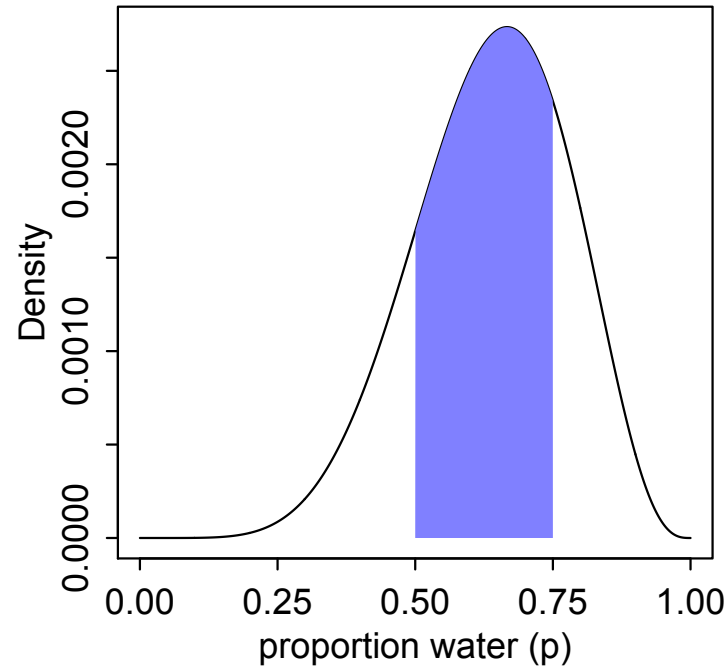
Figure 3.1

Compute stuff

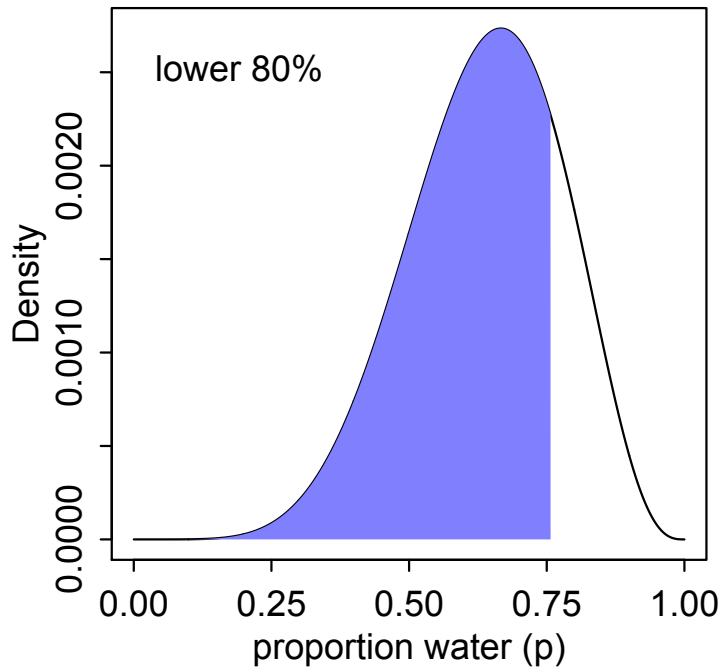
- Summary tasks
  - How much posterior probability below/above/between specified parameter values?
  - Which parameter values contain 50%/80%/95% of posterior probability? "*Confidence" intervals*
  - Which parameter value maximizes posterior probability? Minimizes posterior loss? *Point estimates*
-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

Intervals of defined boundary ask *how much mass?*





Intervals of defined mass ask *which values?*



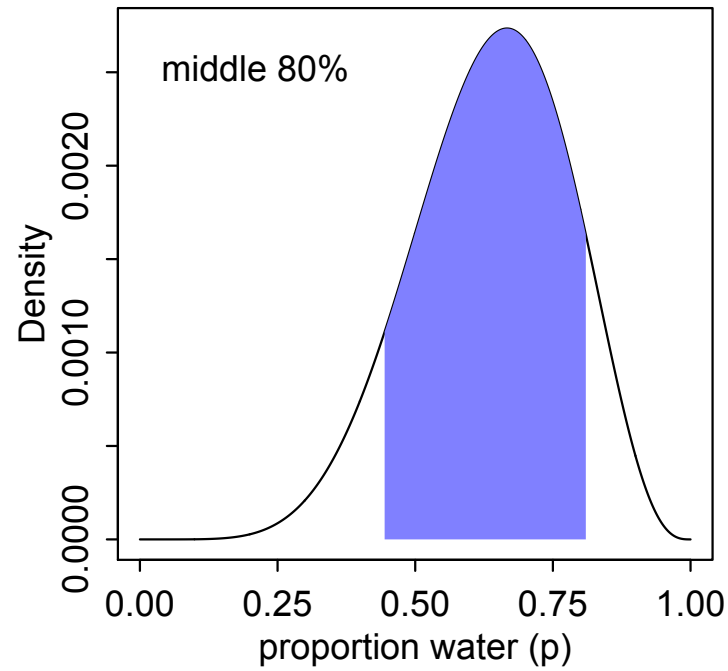
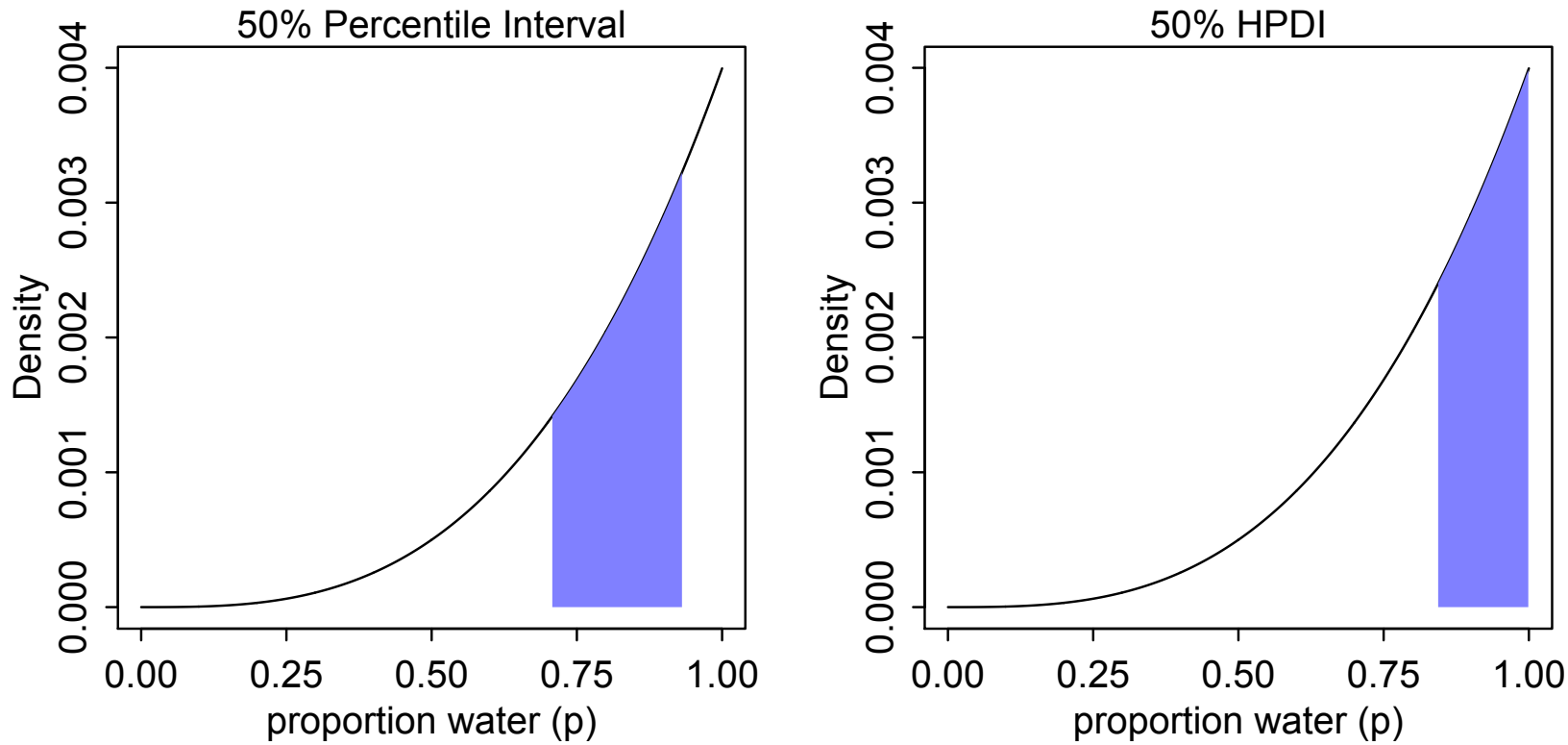
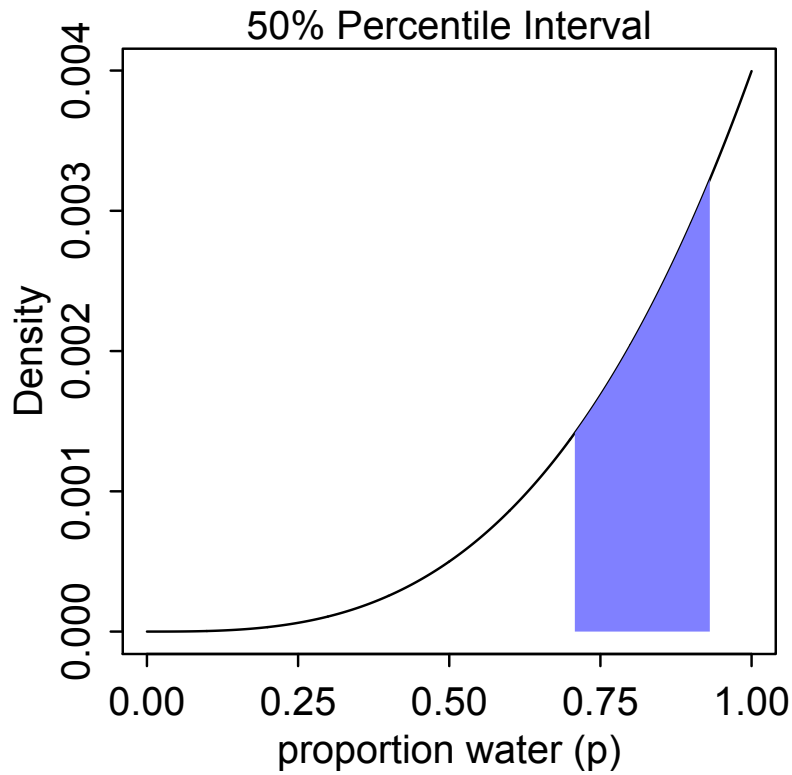


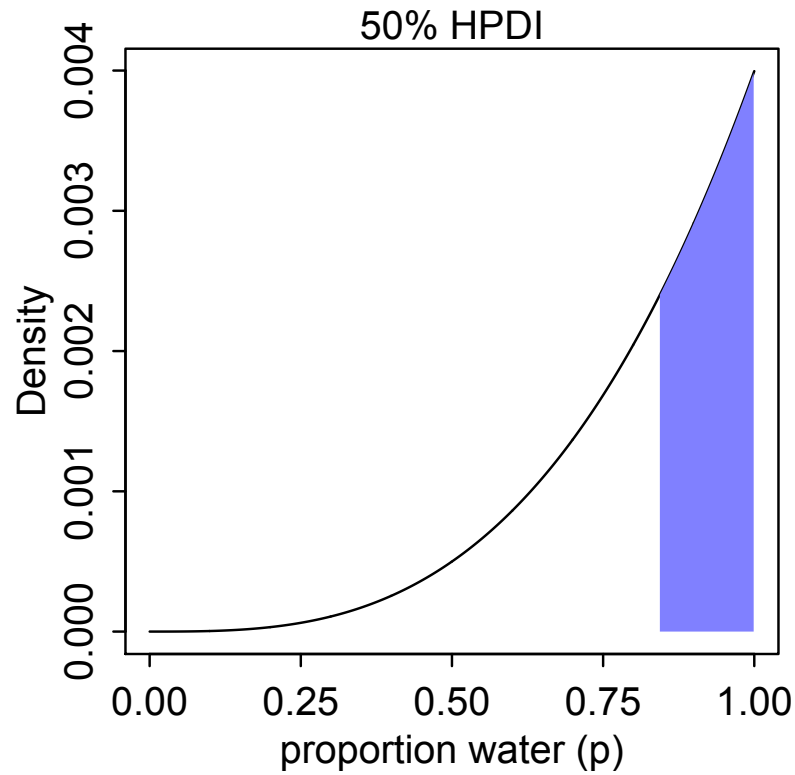
Figure 3.2



- *Percentile intervals* (PI): equal area in each tail
- *Highest posterior density intervals* (HPDI): narrowest interval containing mass

Figure 3.3





R code 3.12

```
PI( samples , prob=0.5 )
```

```
      25%       75%
0.7037037 0.9329329
```

R code 3.13

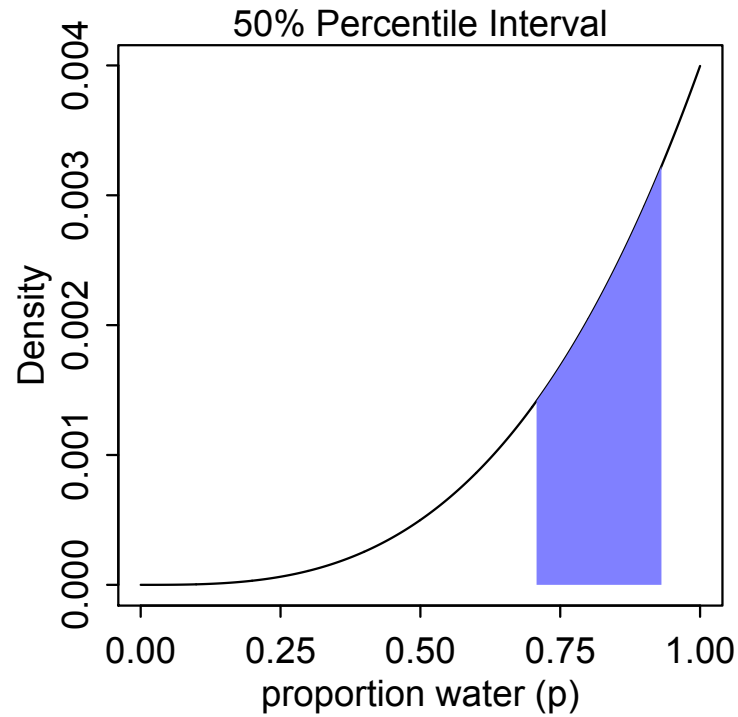
```
HPDI( samples , prob=0.5 )
```

```
lower 0.5 upper 0.5
0.8418418 1.0000000
```

Figure 3.3

Point estimates not the point

- Don't usually want point estimates
  - Entire posterior contains more information
  - "Best" point depends upon purpose
  - Mean nearly always more sensible than mode



Talking about intervals

- "Confidence interval"
  - A non-Bayesian term that doesn't even mean what it says
- "Credible interval"
  - The values are not "credible" unless you trust the model & data
- How about: *Compatibility interval*
  - Interval contains values compatible with model and data as provided
  - Small World interval

https://xkcd.com/2048/

Predictive checks

- Posterior probability never enough
- Even the best model might make terrible predictions
- Also want to check model assumptions
- Predictive checks: Can use samples from posterior to simulate observations
  - NB: Assumption about sampling is assumption

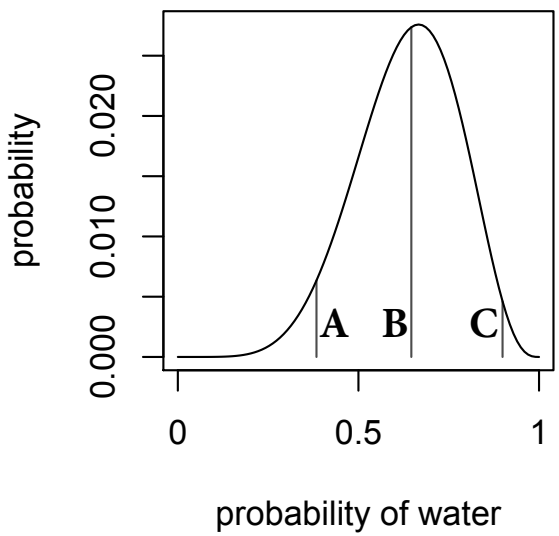


Figure 3.4

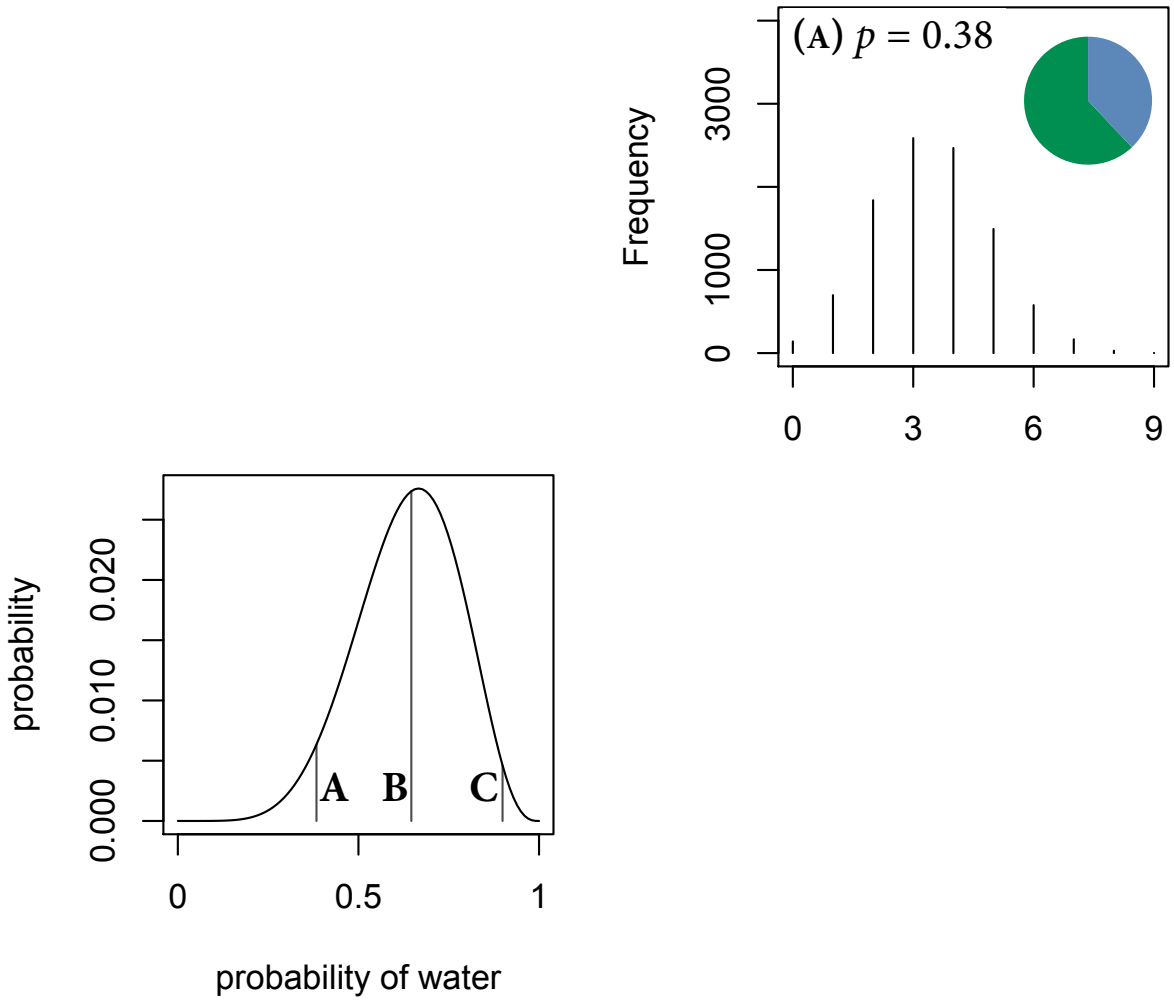

(A) $p = 0.38$
Frequency
0
1000
3000
0
3
6
9
probability
0.000
0.010
0.020
A
B
C
0
0.5
1
probability of water

Figure 3.4

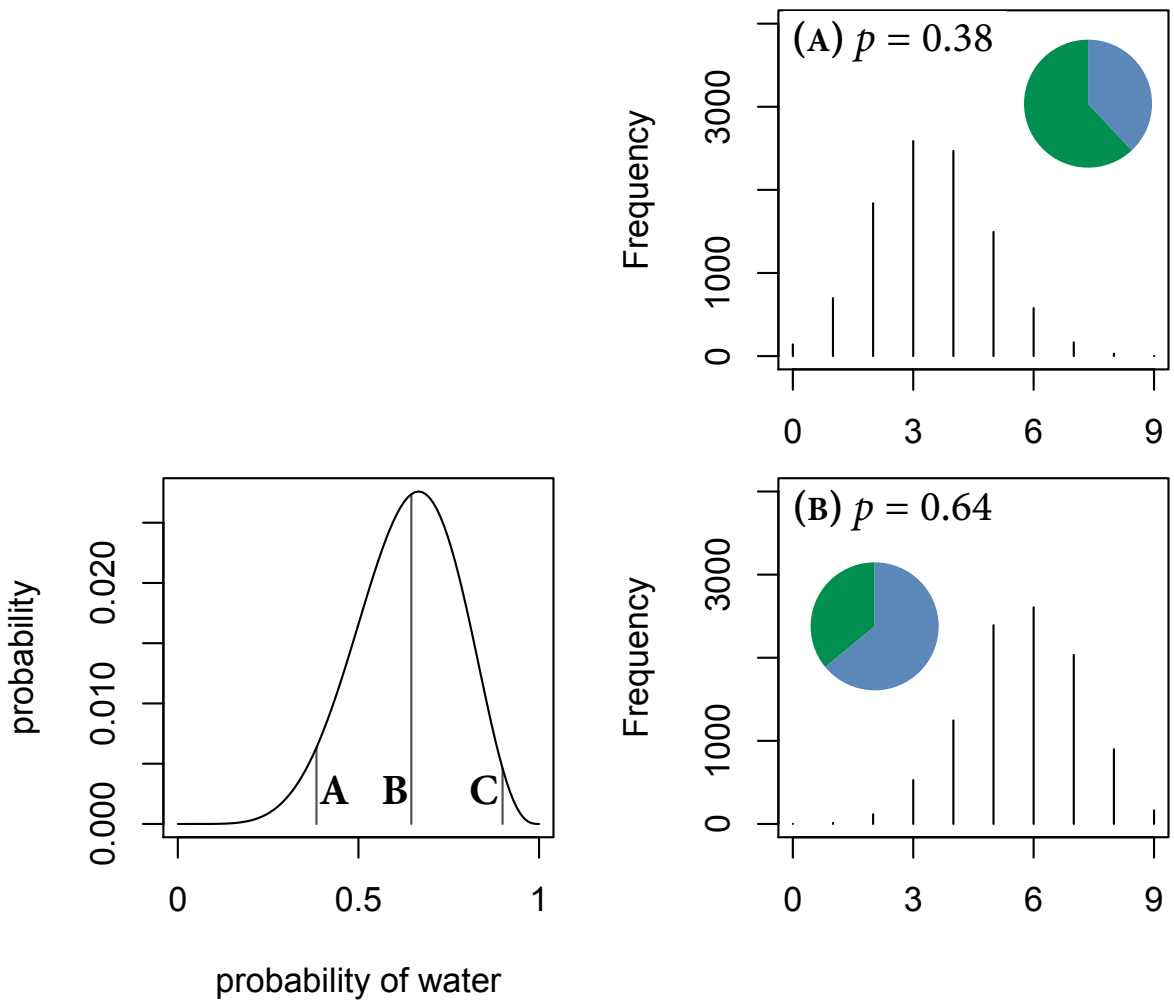

probability
0.000
0.010
0.020
A
B
C
0
0.5
1
probability of water
Frequency
0
1000
3000
(A) $p = 0.38$
0
3
6
9
Frequency
0
1000
3000
(B) $p = 0.64$
0
3
6
9

Figure 3.4

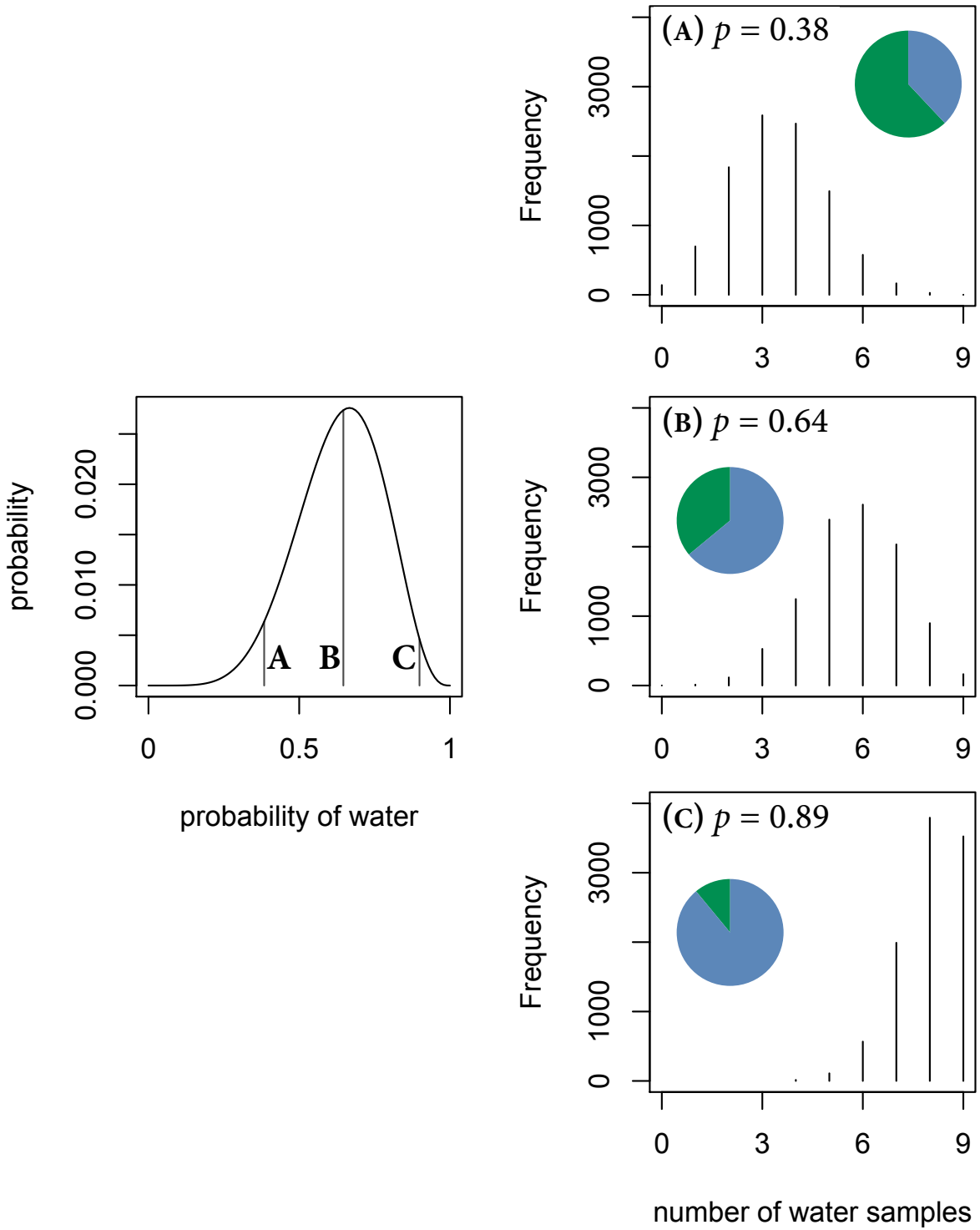

probability
0.000
0.010
0.020
A
B
C
0
0.5
1
probability of water
(A) $p = 0.38$
(B) $p = 0.64$
(C) $p = 0.89$
Frequency
0
1000
3000
0
3
6
9
number of water samples

Figure 3.4

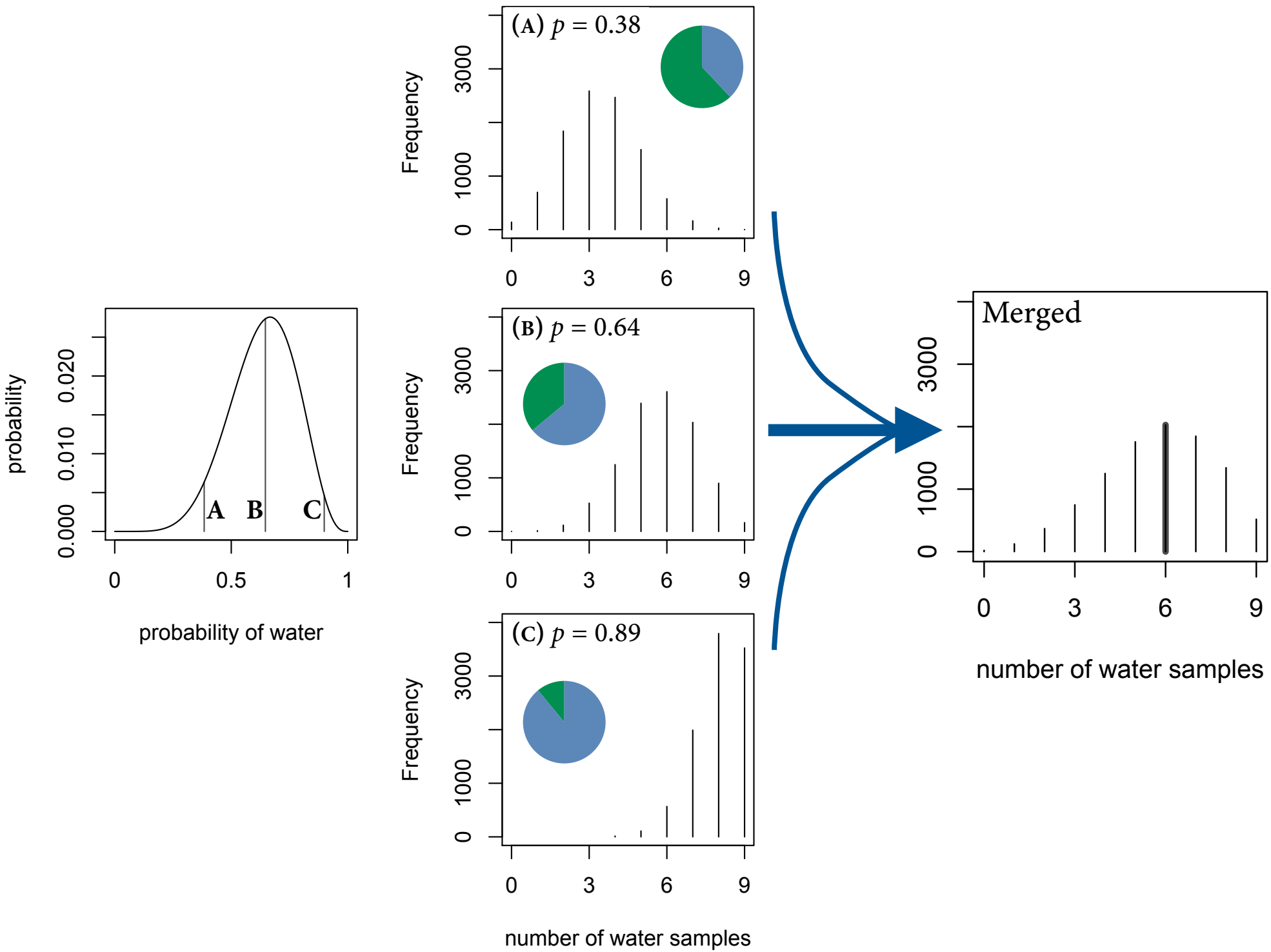
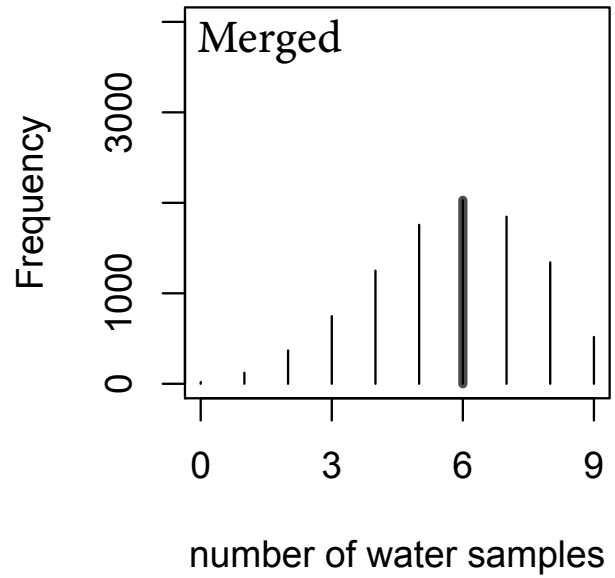

probability
0.000
0.010
0.020
A
B
C
0
0.5
1
probability of water
Frequency
(A) p = 0.38
(B) p = 0.64
(C) p = 0.89
0
1000
3000
0
3
6
9
number of water samples
Merged

Figure 3.4

Posterior predictions

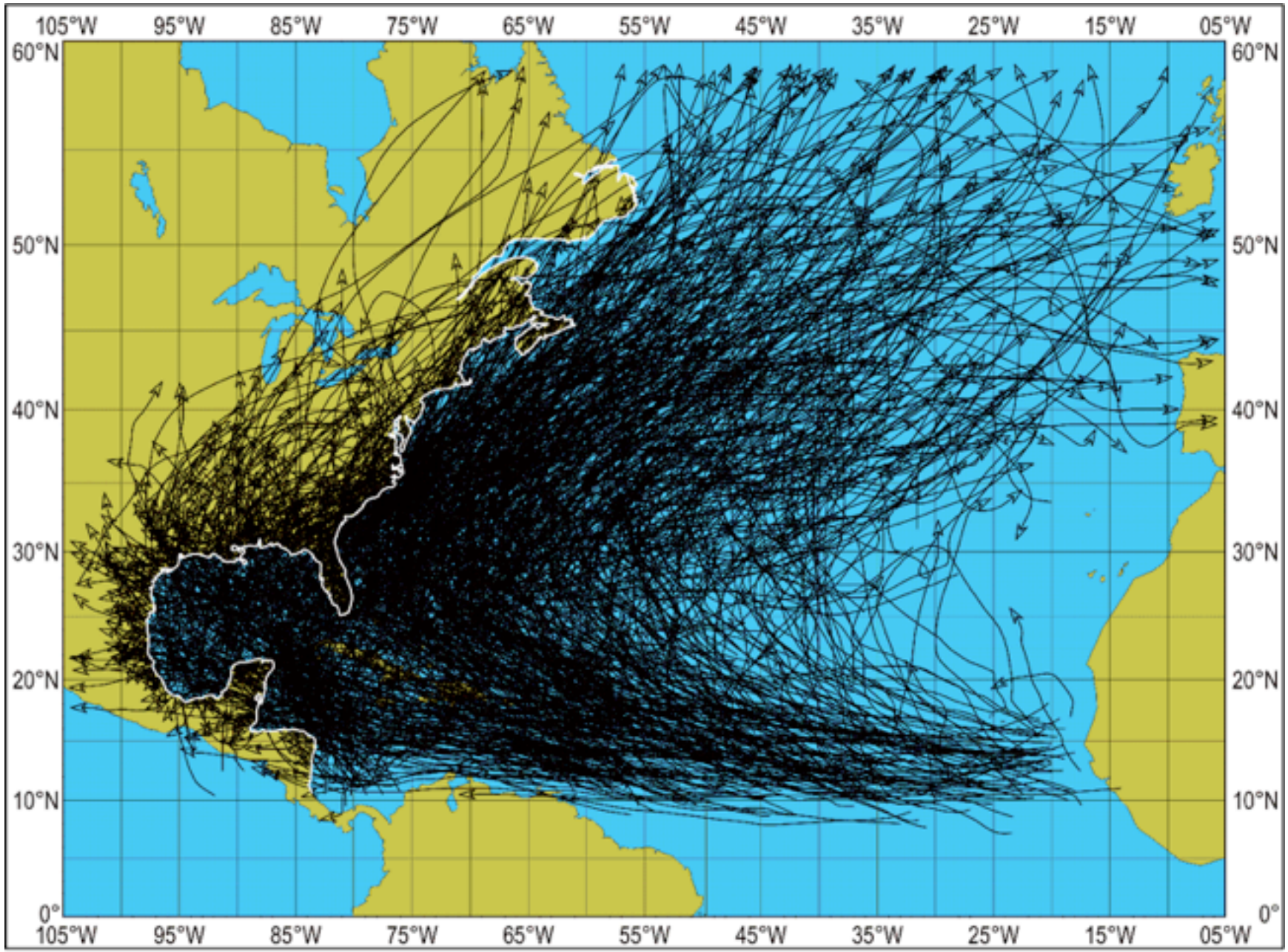


- One line of code

```
nw <- rbinom( 1e4 , size=9 , prob=samples )
```

R code 3.21

- Will get harder, later. But strategy remains the same.



NORTH ATLANTIC TROPICAL STORMS AND HURRICANES, 1851-2004 (1325 STORMS)

NOAA

Predictive checks

- Something like a *significance test*, but not
- No universally best way to evaluate adequacy of model-based predictions
- No way to justify always using a threshold like 5%
- Good predictive checks always depend upon purpose and imagination

"It would be very nice to have a formal apparatus that gives us some 'optimal' way of recognizing unusual phenomena and inventing new classes of hypotheses [...]; **but this remains an art for the creative human mind.**"
—E.T. Jaynes (1922–1998)

Homework

- Week 01 homework on the course website

`https://github.com/rmcelreath/statrethinking_winter2019homework/week01.pdf`

- Due Friday December 14

- Next week: Geocentric Models (Chapter 4)

- Be sure to update your book PDF! Typos have been fixed, more commits coming for later chapters. Password: tempest