Statistical Rethinking Notes

Tyler Burch

Chapter 1 - The Golem of Prague

1.1 - Statistical Golems

"Golem" analogy to clay robots that just do what they're told without thinking, leading to carelessness. Common in statistics too, consider flowcharts with many different tests at the end. Need a more generalized approach (think engineering - don't start with building bridges, start with physics), rethinking inference as a set of strategies, not a set of tools.

1.2 - Statistical Rethinking

Many have approach that the objective of inference is to test null hypotheses, but science isn't described by falsification standard.

Hypotheses are not models

- Models can correspond to multiple hypotheses, multiple hypotheses to a model
- All models are false, so what does it mean to falsify a model?

Progression: - Hypotheses - Statements - Process Models - Causal structure model that formalize cause/effect relationships - Statistical Model - Don't embody causal relationships; express associations among variables

How to get from statistical model back to process model? Derive a expected frequency distribution of some quantity - "statistic" from the causal model. Histogram, for example. Unfortunately, other models can imply the same statistical model, reinforcing the many-to-many relationships:

- Any statistical model can correspond to many process models
- Any hypothesis can correspond to multiple process models
- Any statistical model can correspond to multiple hypotheses

So what to do? If you have multiple process models that all make similar predictions, then you search for a description where the processes look different.

Measurement matters Logic of falsification is have hypothesis, look for observation. If not found, then hypothesis is false. If we formalize H_0 to "All swans are white", no number of observations can prove it, one observation disproves - powerful but prone to observation errors, and quantitative hypotheses have degrees of existence

Observation Error - Doubted observations, measurement/instrumentation error

Continuous Hypotheses - Not trying to disprove, but understand a distribution

Falsification is consensual Communities argue toward consensus about the meaning of evidence, can be messy

1.3 - Tools for Golem Engineering

If falsification isn't the way, we can model. Models then can be made to testing procedures, as well as designs, forecasts, and arguments. This text focuses on several tools: - Bayesian Data Analysis - Model comparison - Multilevel models - Graphical models

Bayesian data analysis Takes question in form of a model and uses logic to produce an answer in the form of probability distributions. Literally just counting how the data may look according to our assumptions. Compare to frequentest, which is defined by the frequencies of events in large samples, based on imaginary data resampling.

Bayesian approaches treat randomness as a property of information, not of the world.

Model comparison and prediction If multiple models, how to choose? Cross validation and information criteria.

Help in 3 ways: provide expectations of accuracy, give an estimate of overfitting, help spot influential observations

Multilevel Models Parameters all the way down - parameters support inference. Multiple levels of uncertainty feed into the next, a multilevel (hierarchical, random effects, varying effects, mixed effects) model. Help us with overfitting, by exploiting partial pooling, which can be used to adjust estimates for repeat sampling, imbalance, variation, and to avoid averaging.

Diverse models turn out to be multilevel: models for missing data (imputation), measurement, factor analysis, some time series models, and types of spatial and network regression are all special applications of multilevel.

Fitting and interpreting multilevel models can be harder than traditional

Graphical Causal Models Statistical models are association engines, detect (not infer!) association between cause and effect. Due to overfitting though, causally incorrect models can make better predictions than causally correct ones, can't focus on just prediction.

So instead, causal model that can be used to design one or more statistical models for causal identification. "Graphical Causal Model" represents a causal hypothesis, the most simple being a Directed Acyclic Graph (DAG).

1.4 - Summary

4 parts to book

1. Bayesian inference (ch 2,3)
2. Multiple linear regression (ch 4-9)
3. Generalized linear models, with MCMC and maximum entropy (ch 9-12)
4. Multilevel models, specialized models (ch 13-17)

Chapter 2 - Small Worlds and Large Worlds

Christopher Columbus thought the world was 10,000 km smaller than it is, his prediction made him think he could have enough supplies to circle it - contrast between model and reality.

Small world - self-contained logical world of model

Large World - larger context where model is deployed

2.1 - The Garden of Forking Data

Bayesian inference is really just counting and comparing possibilities. Cannot guarantee a correct answer, but can guarantee the best possible answer given information provided.

Use example of blue/white marbles out of bag, examine each path of pulls to exclude different "paths" through the data. Introduce a "prior," if all routes are equally likely, can base prior on the number of counts to an outcome. Update prior after a pull, or when new information is added (e.g. the company says blue is rare). The example outlines the following terms:

1. Conjectured proportion of marbles (p in example) is a "parameter" value
2. Number of ways a value can produce data is a "likelihood" - derived by enumerating all data sequences that could have happened and eliminate those inconsistent with data
3. Prior plausibility of p is the "prior probability"
4. Updated plausibility of p is the "posterior probability"

2.2 - Building a model

Design loop with 3 steps:

1. Data story: motivate model by narrating how the data might arise
2. Update: educate model by feeding it data
3. Evaluate: All models require supervision, possibly leading to model revision

Example in chapter - calculate the amount of water on earth by throwing globe up in the air, see if right thumb is on water or land.

Data Story May be *descriptive*, specifying associations to predict outcomes given observations; may be *causal*, a theory of how some events produce others. Generally causal stories are easily descriptive, descriptive stories may be hard to be causal. Can motivate by explaining how each piece of data is born. The value of the story is to more strongly define hypotheses and resolve ambiguities.

Example: true proportion of water is p, single toss has p chance of producing water W and $1 - p$ of land L, each toss is independent.

Bayesian Updating Model begins with one set of plausibilities assigned to each possibility. As data is collected, posteriors are produced.

Example: Give uniform prior. First case lands on water, $p = 0$ is excluded, since there is no longer a possibility of no water. Continue tossing and distribution shifts and closes more and more.

One benefit of Bayesian approach is that estimates are valid for any sample size. Of course better with more data.

Evaluate Because of differences between the model and real world, no guarantee of large world performance. Keep in mind two principles. Model's certainty is no guarantee the model is good - this is telling you that, given this model, plausible values are in some range. Second, it's important to supervise and critique the work - in the example, order of tosses shouldn't change final curve, but may indirectly affect it because data depends on order, so check on data it does not know about.

2.3 - Components of the Model

Variables Symbols that can take different values - for globe example: target p (proportion of water) cannot be observed. Unobserved are called "parameters." Other variables are count of water W and count of land L, these are observed.

Definitions In defining, we build a model relating variables to one another. For each value of unobserved, need to define the number of ways/probability that the values of each observed could arise. For each unobserved also need a prior.

Using example:

Observed Variables - For specific p, need to define how plausible combinations of W and L would be, using a mathematical function called a likelihood. In this case, if tosses are independent and probability are the same, we use binomial distribution.

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

In R:

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

```
## [1] 0.1640625
```

This gives the relative number of ways to get 6 water results for $p = 0.5$ after 9 total tosses ($N = W + L = 9$).

Unobserved Variables - Distributions for observed variables typically have own variables; p not observed, so a parameter. Many common data questions are answered directly by parameters, e.g. average difference between groups, association strength, covariate dependence, variation. For every parameter, you must define a prior.

Some schools of thought that emphasize choosing priors on personal belief, known as "subjective Bayesian." If you don't have a strong argument for any prior, try different ones.

Model is born Can summarize as the following:

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

Telling us W is binomial, and p is flat over the range 0 to 1.

2.4 - Making the Model Go

Once you have named all the variables, definitions, update prior to posterior - the relative plausibility of parameter values conditional on fdata and model, for our example $\Pr(p|W, L)$.

Bayes Theorem Mathematical definition of posterior arises from Bayes. Joint probability $\Pr(W, L, p) = \Pr(W, L|p)\Pr(p)$. The right side can also be reversed $\Pr(p|W, L)\Pr(W, L)$, which can be solved to

$$\Pr(p|W, L) = \frac{\Pr(W, L|p)\Pr(p)}{\Pr(W, L)}$$

This is Bayes theorem but can really just be said

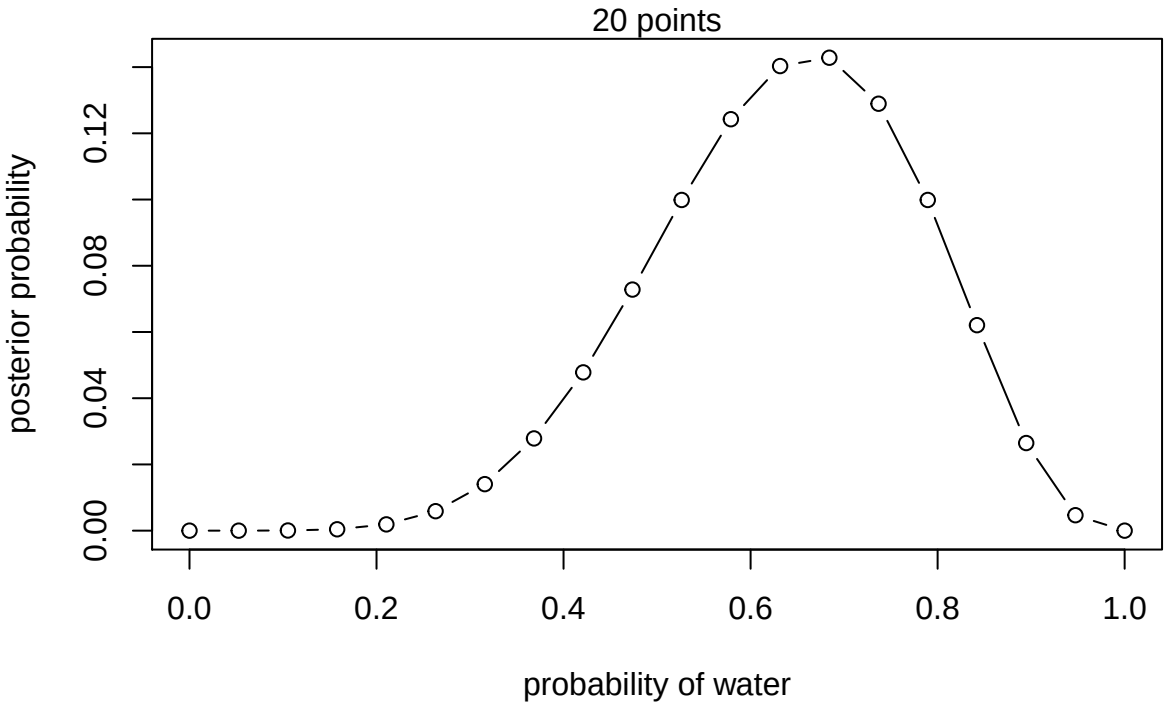
$$\text{Posterior} = \frac{\text{Probability of data} \times \text{Prior}}{\text{Average probability of the data}}$$

Denominator is averaged over the prior - meant to standardize the posterior to make the sum one.

Three different numerical techniques for computing posterior: grid approximation, quadratic approximation, MCMC.

Grid Approximation - Consider a finite number of values, compute posterior by multiplying prior by likelihood, repeat until getting an approximate picture of the posterior. Mostly a pedagogical tool, since not typically practical.

```
# define grid
p_grid <- seq( from=0 , to=1 , length.out=20 )
# define prior
prior <- rep( 1 , 20 )
# compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )
# compute product of likelihood and prior
unstd.posterior <- likelihood * prior
# standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
plot(p_grid , posterior , type="b" ,
     xlab="probability of water" , ylab="posterior probability")
mtext( "20 points" )
```



Quadratic Approximation - More parameters make grid approximations tough (N^p for p parameters and N data points). Use quadratic approximation when the region near the peak of the posterior will be Gaussian in shape, easy because it can be described by just mean and variance.

1. Find posterior mode
2. Estimate curvature, either analytically or computationally.

For this book use `quap` from rethinking programming package

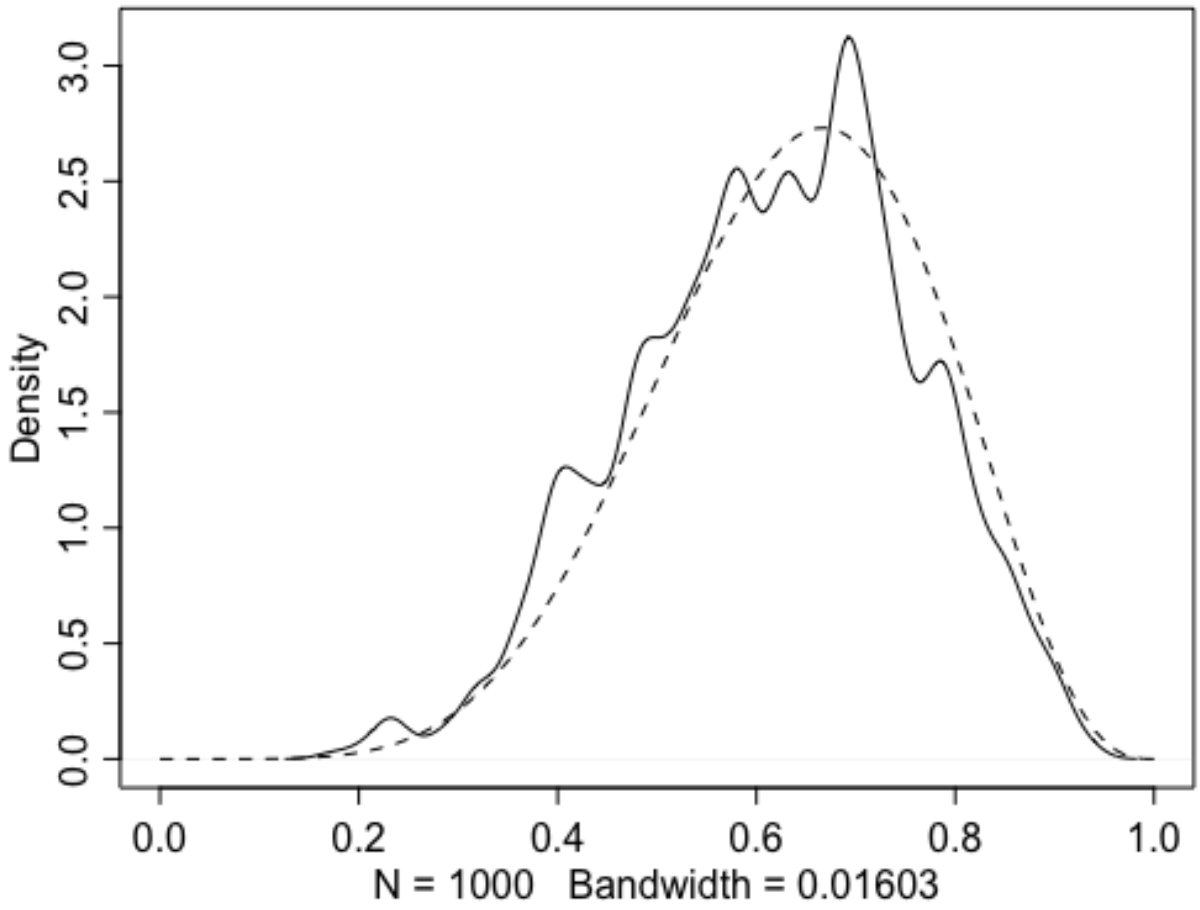
```
library(rethinking)
globe.qa <- quap(
  alist(
    W ~ dbinom (W+L, p), # Binomial
    p ~ dunif(0,1) # Uniform
  ), data = list(W=6, L=3)
)
precis(globe.qa)
```

```
##        mean        sd      5.5%     94.5%
## p 0.6666796 0.1571307 0.4155543 0.9178049
```

The quadratic approximation is often equivalent to a Maximum Likelihood Estimate and its standard error.

note - quadratic is solved by computing the Hessian, a square matrix of second derivatives of the log of posterior probability wrt parameters. Derivatives sufficient to describe a Gaussian. Std is typically computed from Hessian, which can occasionally cause problems in computation.

Markov chain Monte Carlo (MCMC) - Many models, like multilevel/mixed-effects don't work for grid approximation (many parameters) or quadratic (non-Gaussian posterior). Function to maximize isn't known, computed in pieces via MCMC. Rather than computing or approximating posterior, MCMC draws samples, a collection of parameter values.



2.5 - Summary

Looked at conceptual ideas in Bayesian data analysis. Models are composite of variables and distributional definitions, fit to data using numerical techniques

Chapter 3 - Sampling the Imaginary

Given example of test to check for vapmirism 0 highly accurate, but makes false positives at the rate of $\Pr(\text{positive test}|\text{mortal}) = 0.01$. Vampirism is rare, 0.1% of the population.

To solve given a positive test the likelihood that they are a vampire,

$$\Pr(\text{positive}) = \Pr(\text{positive}|\text{vampire})\Pr(\text{vampire}) + \Pr(\text{positive}|\text{mortal})(1 - \Pr(\text{vampire}))$$

and

$$\Pr(\text{vampire}|\text{positive}) = \frac{\Pr(\text{positive}|\text{vampire})\Pr(\text{vampire})}{\Pr(\text{positive})}$$

```
Pr_Positive_Vampire <- 0.95
Pr_Positive_Mortal <- 0.01
Pr_Vampire <- 0.001
Pr_Positive <- Pr_Positive_Vampire * Pr_Vampire +
  Pr_Positive_Mortal * ( 1 - Pr_Vampire )
( Pr_Vampire_Positive <- Pr_Positive_Vampire*Pr_Vampire / Pr_Positive )
```

```
## [1] 0.08683729
```

Gives an 8.6% chance they're actually a vampire, despite positive test. This is a canonical problem, broader in statistics - despite using Bayes' theorem, not uniquely Bayesian. Reframe using *natural frequencies*:

1. In a population of 100,000 people, 100 are vampires
2. Of 100, 95 test positive
3. Of 99,900 mortals, 999 test positive

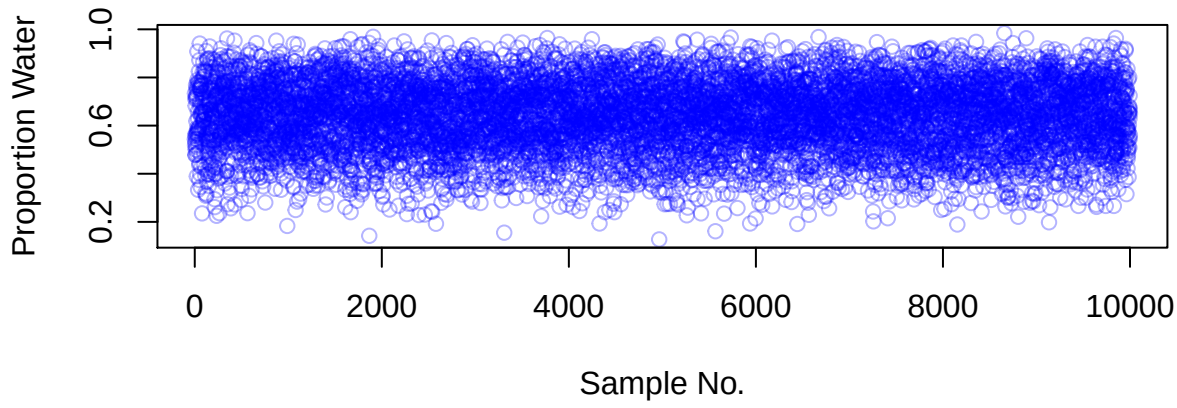
$$\begin{aligned}\Pr(\text{vampire}|\text{positive}) &= \frac{\text{true positives}}{\text{all positives}} \\ &= \frac{95}{1094} \approx 0.087\end{aligned}$$

Chapter is focused on working from samples from posterior, to make sense of model output.

3.1 - Sampling from a grid approximate posterior

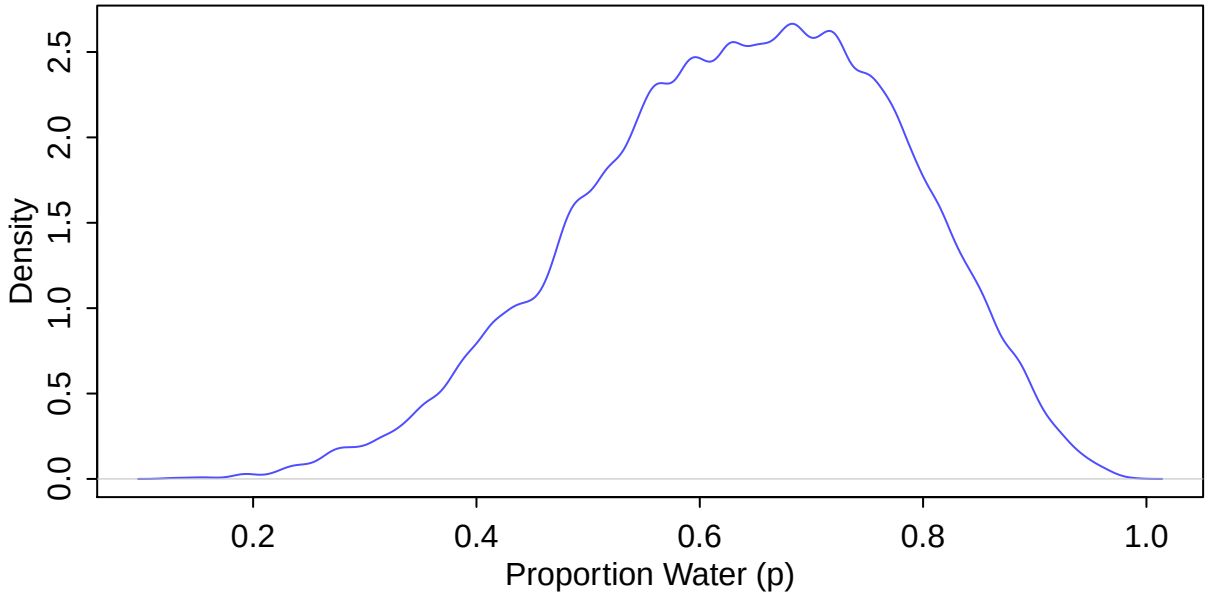
Rerun code for grid approximation posterior, then draw 10,000 samples.

```
  samples <- sample(p_grid, prob=posterior, size=1e4, replace=TRUE)
  plot(samples,col=alpha("blue",0.3), xlab = "Sample No.", ylab = "Proportion Water")
```



Can also draw a density plot:

```
dens(samples, col=alpha("blue",0.7),
     xlab = "Proportion Water (p)", ylab = "Density")
```



3.2 - Sampling to Summarize

Can ask many questions using your posterior: How much lies below a parameter value, or between two parameter values? Which parameter value marks the lower X%? What range contains most the posterior probability? Which parameter values are most likely?

Intervals of defined boundaries - Address probability that proportion of water is less than 0.5. Directly from grid:

```
sum( posterior[ p_grid < 0.5 ] )
```

```
## [1] 0.1718746
```

However, if not using grid, you can use samples and get a nearly identical result:

```
sum( samples < 0.5 ) / 1e4
```

```
## [1] 0.1724
```

Intervals of Defined Mass - "Confidence intervals" commonly used, but we work with "credible interval" or "compatibility interval." To get the 80% "percentile interval" (PI):

```
quantile( samples , c( 0.1 , 0.9 ) )
```

```
##       10%       90%
## 0.4473473 0.8158158
```

`PI` function in rethinking package does this as well. Also `HPDI` is the "Highest posterior density interval," the narrowest interval containing specified probability mass. Generally this interval best represents parameter values consistent with the data. Note HPDI is more computationally intensive and suffers from variance on number of samples drawn.

Point Estimates - Given the entire posterior, what number to report? Can do *maximum a posteriori* (MAP) by taking the mode. Other point estimates (mean, median) can also work, but often worse in terms of loss function.

3.3 - Sampling to Simulate Prediction

Useful for:

1. Model design - sampling from the prior can help understand implications
2. Model checking - see if the fit worked correctly
3. Software validation - does the model fitting software work alright? check by recovering parameter values
4. Research design - simulate observations from hypothesis, can evaluate weather research design is effective, *power analysis*.
5. Forecasting - simulate new predictions for the future

Dummy data Bayesian models are always generative, capable of simulating predictions. For the globe example

```
dbinom( 0:2 , size=2 , prob=0.7 )
```

```
## [1] 0.09 0.42 0.49
```

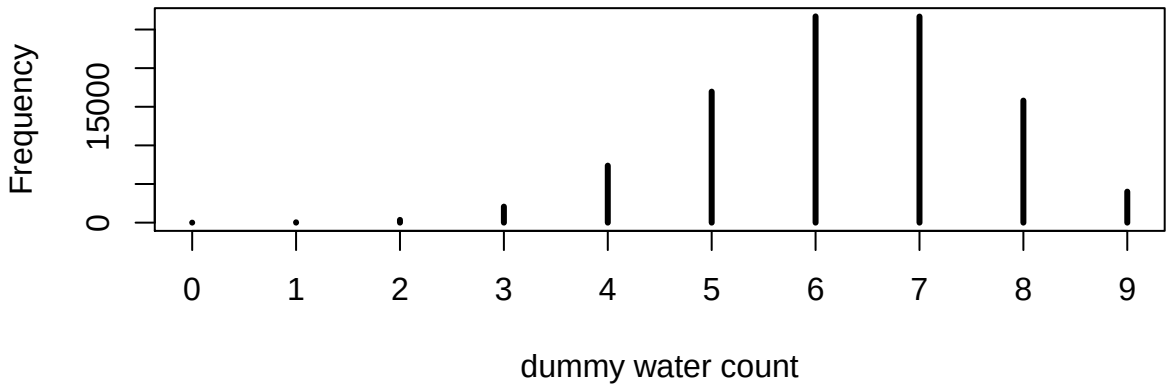
simulates 0, 1, 2 "water" results; 9% chance of not landing on water at all. Can simulate many dummy observations:

```
trials <-1e5
dummy_w <- rbinom( trials , size=2 , prob=0.7 ) # r for random, 10
table(dummy_w)/trials
```

```
## dummy_w
##       0       1       2
## 0.09119 0.41832 0.49049
```

which are close to analytical solution. Also can plot to make sure it looks binomial, now using 9 tosses

```
trials <-1e5
dummy_w <- rbinom( trials , size=9 , prob=0.7 )
simplehist( dummy_w , xlab="dummy water count" )
```



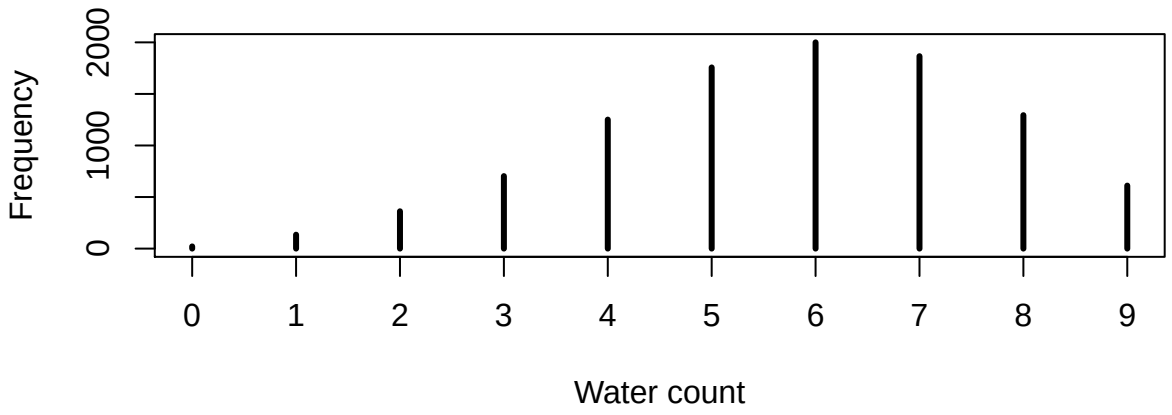
Model Checking

1. Ensure the fitting worked correctly
2. Evaluate the adequacy of the model for a purpose

Don't test whether assumptions are "true," assess exactly how it fails to describe the data. Basic model checks using samples from full posterior (not point estimates!).

- Observation uncertainty: sample variation - globe tossing, even if you know p exactly, you won't know the next globe toss results
- Parameter uncertainty: posterior distribution embodies this, will interact with sampling variation. Want to propagate this as evaluating predictions; computing sampling distribution at each value of p, averaging together, gets a "posterior predictive distribution

```
 w <- rbinom( 1e4 , size=9 , prob=samples )
simplehist( w , xlab="Water count" )
```



Here, for each posterior sample, a random binomial dataset is created. Wide spread, but arises from the binomial process itself. Can consider other metrics, like the longest consecutive Water results (mode=3, obs=3) or number of switches between water/land (mode=4, obs=6).

3.4 - Summary

Given basic procedures for manipulating posterior distributions, can be used for intervals, point estimates, posterior predictive checks, simulations. Encapsulate uncertainty about parameters with uncertainty about outcomes.

Problem sets

Chapter 2 problems

2E1.

Which of the expressions below correspond to the statement:the probability of rain on Monday?

1. Pr(rain)
2. **Pr(rain|Monday)**
3. Pr(Monday|rain)
4. Pr(rain,Monday)/Pr(Monday)

2E2.

Which of the following statements corresponds to the expression: Pr(Monday|rain)?

1. The probability of rain on Monday.
2. The probability of rain, given that it is Monday.
3. **The probability that it is Monday, given that it is raining.**
4. The probability that it is Monday and that it is raining.

2E3.

Which of the expressions below correspond to the statement: the probability that it is Monday, given that it is raining?

1. **Pr(Monday|rain)**
2. Pr(rain|Monday)
3. Pr(rain|Monday)Pr(Monday)
4. Pr(rain|Monday)Pr(Monday)/Pr(rain)
5. Pr(Monday|rain)Pr(rain)/Pr(Monday)

2E4.

The Bayesian statistician Bruno de Finetti (1906–1985) began his book on probability theory with the declaration: "PROBABILITY DOES NOT EXIST." The capitals appeared in the original, so I imagine de Finetti wanted us to shout this statement. What he meant is that probability is a device for describing uncertainty from the perspective of an observer with limited knowledge; it has no objective reality. Discuss the globe tossing example from the chapter, in light of this statement. What does it mean to say "the probability of water is 0.7"?

Based on the tosses that we've performed, on the globe we hold, we can expect 70% of future tosses to also land on water. The source of the uncertainty and limited knowledge is that the point we're landing on is random.

2M1.

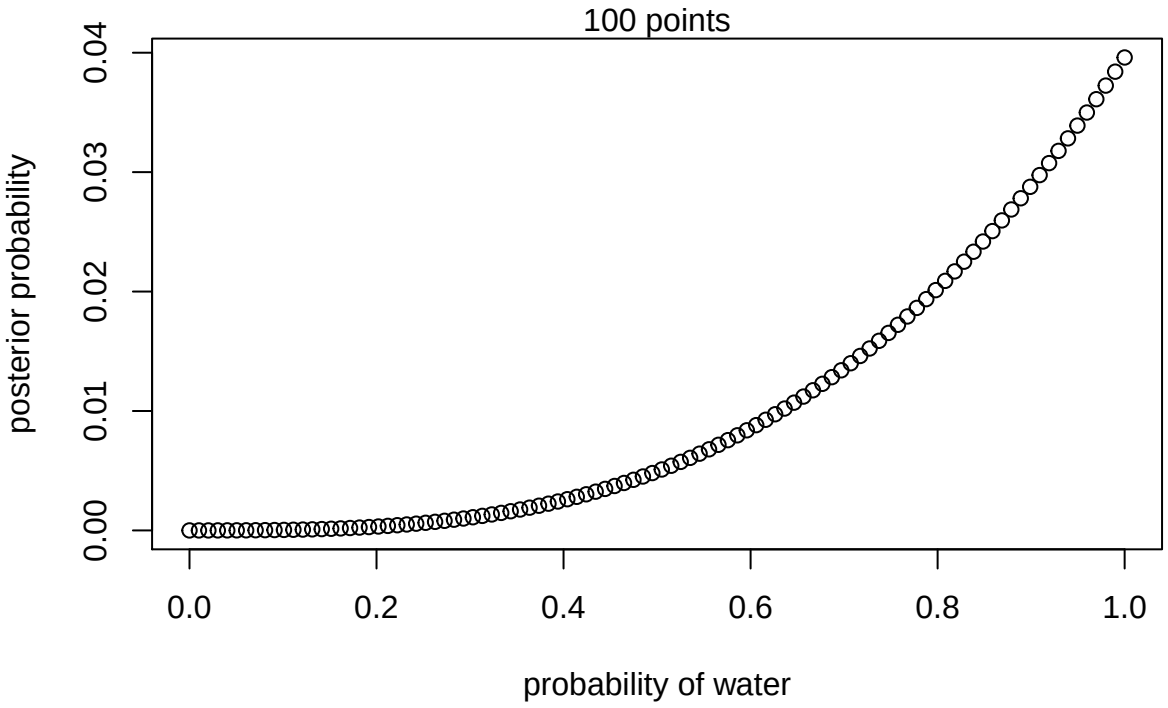
Recall the globe tossing model from the chapter. Compute and plot the grid approximate posterior distribution for each of the following sets of observations. In each case, assume a uniform prior for p.

```
globe_grid <- function(trial_list, grid_size){
  # define grid
  p_grid <- seq( from=0 , to=1 , length.out=grid_size )
  # define prior
  prior <- rep( 1 , grid_size )
  # compute likelihood at each value in grid
  likelihood <- dbinom( sum(trial_list) , size=length(trial_list) , prob=p_grid )
  # compute product of likelihood and prior
```

```
  unstd.posterior <- likelihood * prior
  # standardize the posterior, so it sums to 1
  posterior <- unstd.posterior / sum(unstd.posterior)
  plot(p_grid , posterior , type="b" ,
       xlab="probability of water" , ylab="posterior probability")
  mtext( sprintf("%i points", grid_size ))
}
```

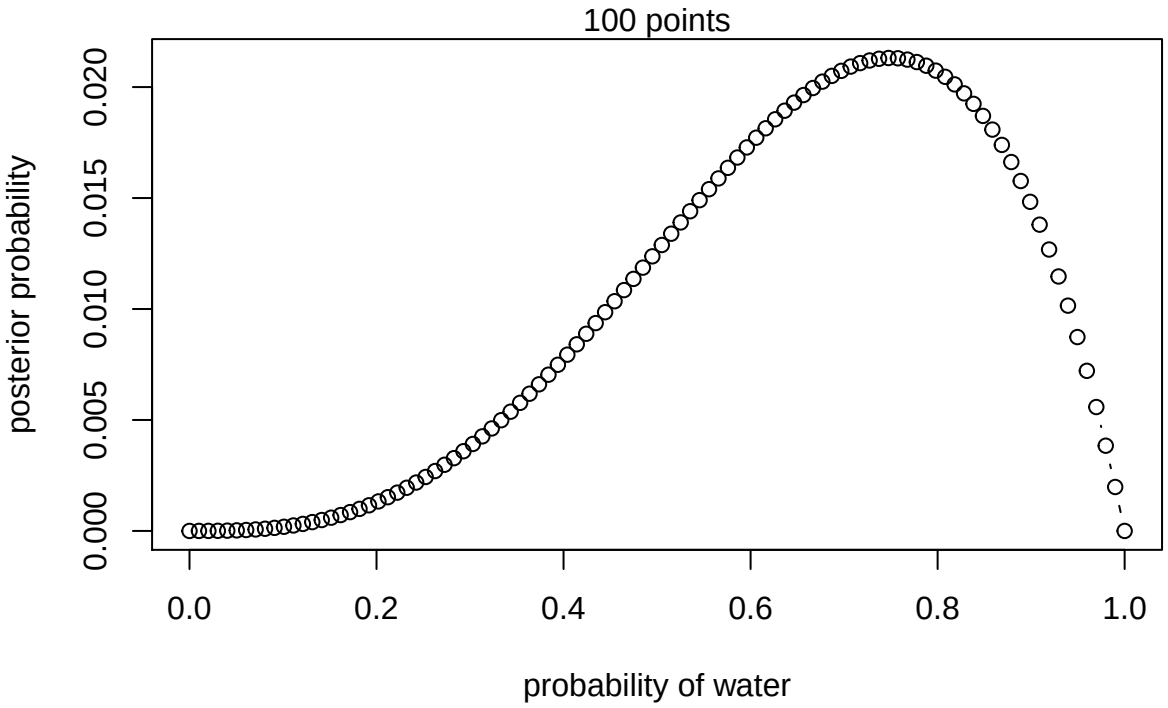
1. W, W, W

```
globe_grid(c(1,1,1),100)
```



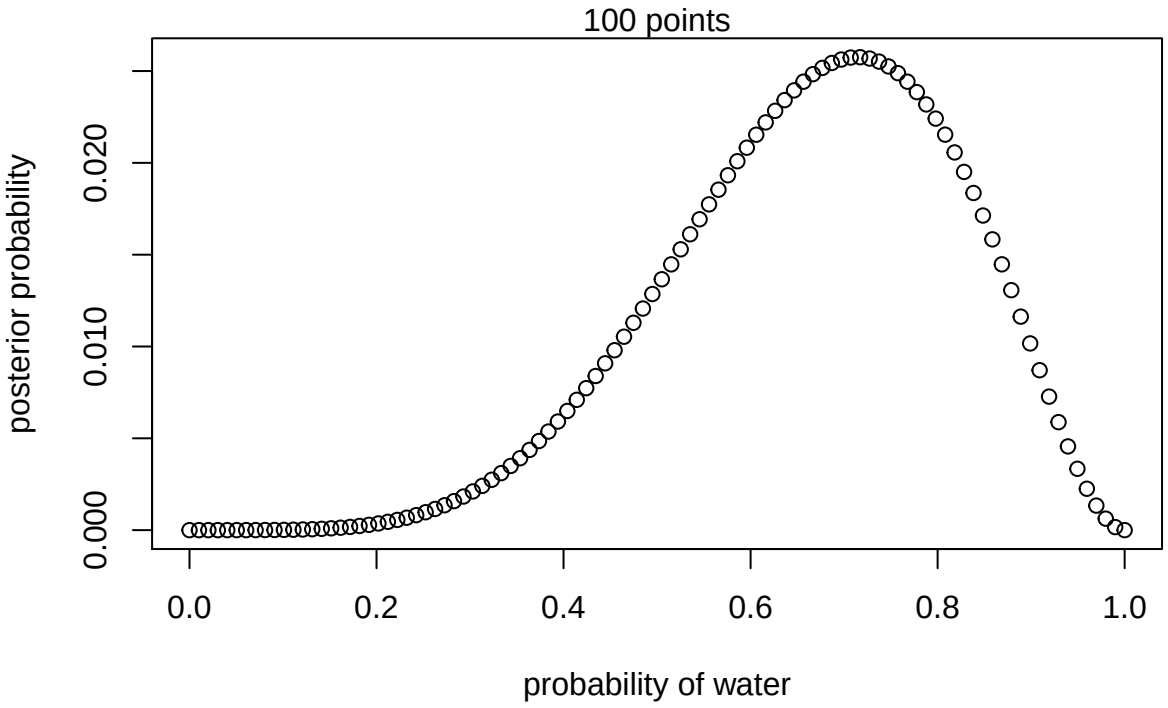
2. W, W, W, L

```
globe_grid(c(1,1,1,0),100)
```



3. L,W,W,L,W,W,W

```
globe_grid(c(0,1,1,0,1,1,1),100)
```



2M2.

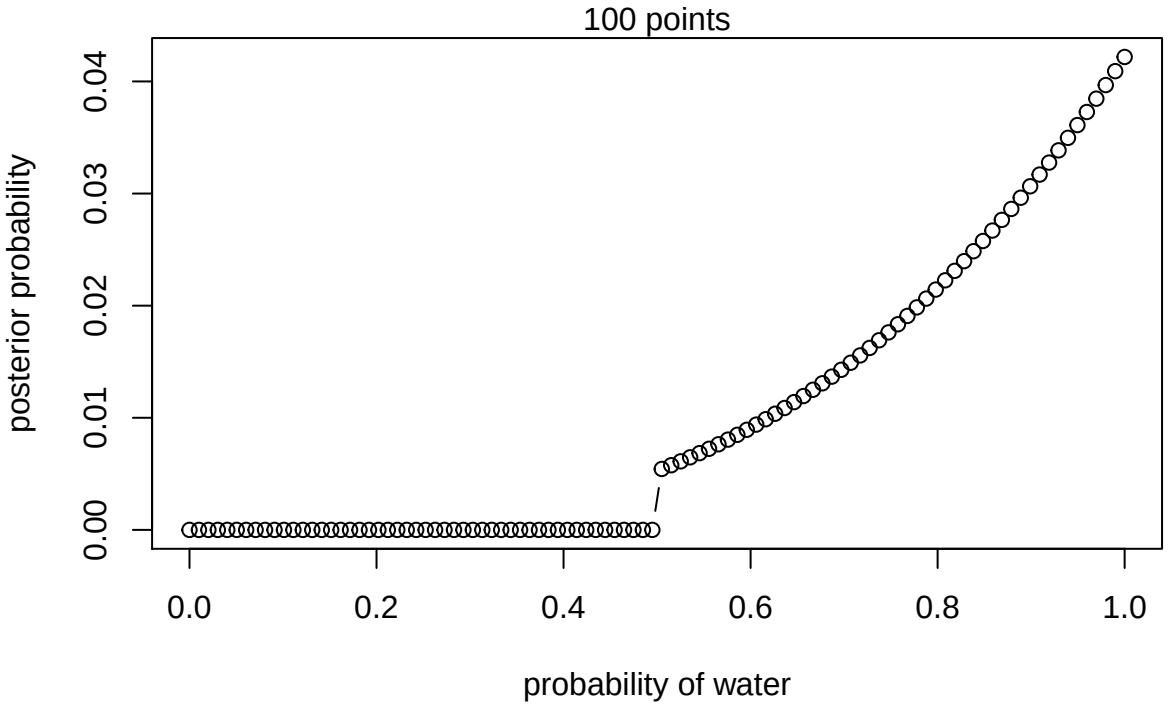
> Now assume a prior for p that is equal to zero when $p < 0.5$ and is a positive constant when $p \geq 0.5$. Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

Change function prior definition to:

```
# define prior
prior <- (p_grid >= 0.5) * prior_const
```

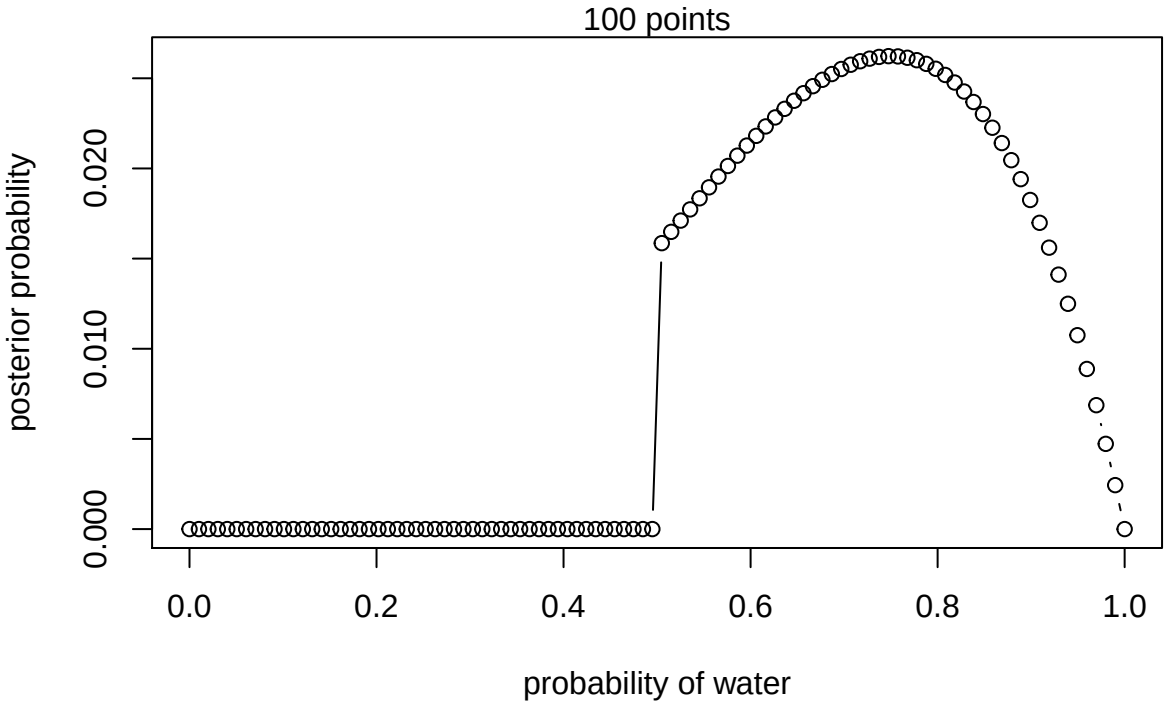
1. W, W, W

```
globe_grid_const(c(1,1,1),100, 1.0)
```



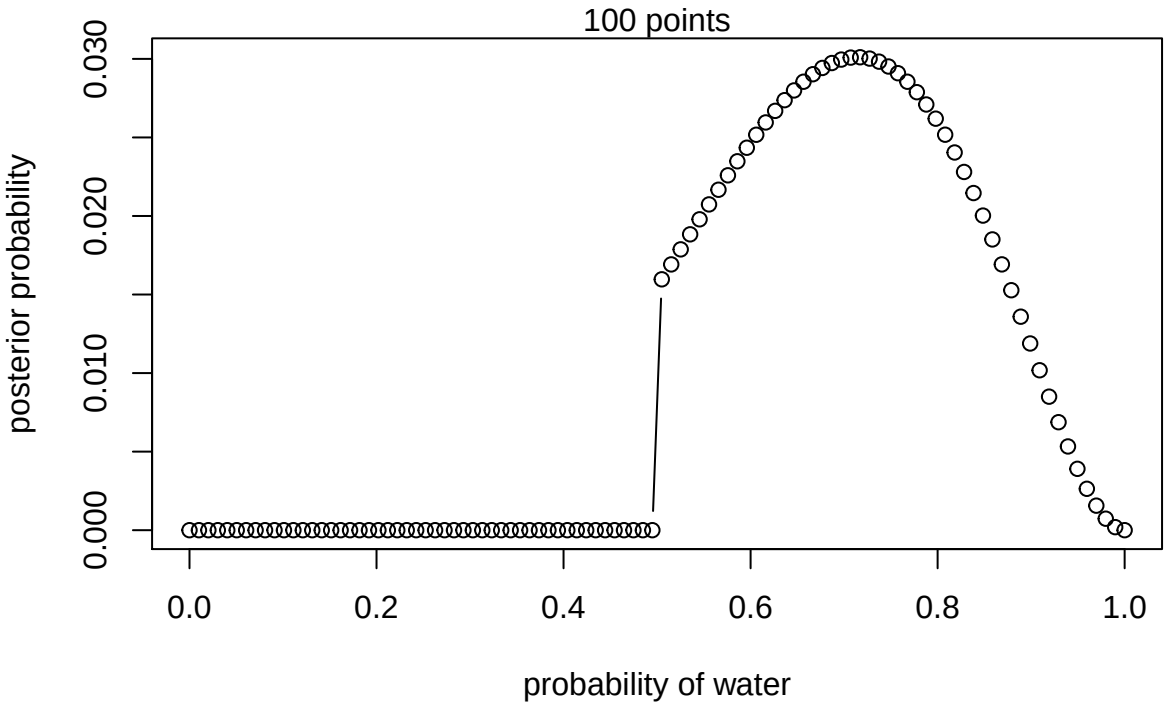
2. W, W, W, L

```
globe_grid_const(c(1,1,1,0),100, 1.0)
```



3. L,W,W,L,W,W,W

```
globe_grid_const(c(0,1,1,0,1,1,1),100,1)
```



2M3.

> Suppose there are two globes, one for Earth and one for Mars. The Earth globe is 70% covered in water. The Mars globe is 100% land. Further suppose that one of these globes—you don't know which—was tossed in the air and produced a "land" observation. Assume that each globe was equally likely to be tossed. Show that the posterior probability that the globe was the Earth, conditional on seeing "land" (Pr(Earth|land)), is 0.23.

First find total probability of land:

$$\Pr(\text{land}) = \Pr(\text{land}|\text{Earth})\Pr(\text{Earth}) + \Pr(\text{land}|\text{Mars})\Pr(\text{Mars}) = (0.3)(0.5) + (1.0)(0.5) = 0.65$$

Now solve for probability of Earth, given we have land:

$$\Pr(\text{Earth}|\text{land}) = \frac{\Pr(\text{land}|\text{Earth})\Pr(\text{Earth})}{\Pr(\text{land})} = \frac{(0.3)(0.5)}{0.65} \approx 0.23$$

2M4.

> Suppose you have a deck with only three cards. Each card has two sides, and each side is either black or white. One card has two black sides. The second card has one black and one white side. The third card has two white sides. Now suppose all three cards are placed in a bag and shuffled. Someone reaches into the bag and pulls out a card and places it flat on a table. A black side is shown facing up, but you don't know the color of the side facing down. Show that the probability that the other side is also black is 2/3. Use the counting method (Section 2 of the chapter) to approach this problem. This means counting up the ways that each card could produce the observed data (a black side facing up on the table).

Enumerating all possible scenarios (b=black, w=white):

1. w up; w down (w/w)

2. w up; w down (w/w)
3. w up; b down (w/b)
4. b up; w down (w/b)
5. b up; b down (b/b)
6. b up; b down (b/b)

Observation eliminates 1-3, so 3-6 remain. 2/3 of those are b/b card, so this is our solution.

2M5.

> Now suppose there are four cards: B/B, B/W, W/W, and another B/B. Again suppose a card is drawn from the bag and a black side appears face up. Again calculate the probability that the other side is black.

Add to the prior cases:

7. b up; b down (new b/b)
8. b up; b down (new b/b)

Again we can eliminate 1-3 from observation. This leaves 5 cases (3-8). Of those 4 are b/b, so 4/5.

2M6.

> Imagine that black ink is heavy, and so cards with black sides are heavier than cards with white sides. As a result, it's less likely that a card with black sides is pulled from the bag. So again assume there are three cards: B/B, B/W, and W/W. After experimenting a number of times, you conclude that for every way to pull the B/B card from the bag, there are 2 ways to pull the B/W card and 3 ways to pull the W/W card. Again suppose that a card is pulled and a black side appears face up. Show that the probability the other side is black is now 0.5. Use the counting method, as before.

Same cases as before, but now amend rates

up	down	card	rate
w	w	w/w	3
w	w	w/w	3
w	b	w/b	2
b	w	w/b	2
b	b	b/b	1
b	b	b/b	1

Now, we can cancel the first three cases, since we've pulled black. 2 b/b options at rate 1, 1 w/b option at rate 2. That means 2 positive chances out of 4 total rate chances, $2/4 = 0.5$.

2M7.

> Assume again the original card problem, with a single card showing a black side face up.Before looking at the other side, we draw another card from the bag and lay it face up on the table. The face that is shown on the new card is white. Show that the probability that the first card, the one showing a black side, has black on its other side is now 0.75. Use the counting method, if you can. Hint: Treat this like the sequence of globe tosses, counting all the ways to see each observation, for each possible first card.

Looking at scenarios that match data:

1. b/b, w/b
2. b/b (flipped), w/b
3. b/b, w/w
4. b/b, w/w (flipped)

5. b/b (flipped), w/w
6. b/b (flipped), w/w (flipped)
7. b/w, w/w
8. b/w, w/w (flipped)

1-6 are desired, 7-8 are not; therefore 6/8 or 75%.

2H1.

> Suppose there are two species of panda bear. Both are equally common in the wild and live in the same places. They look exactly alike and eat the same food, and there is yet no genetic assay capable of telling them apart. They differ however in their family sizes. Species A gives birth to twins 10% of the time, otherwise birthing a single infant. Species B births twins 20% of the time, otherwise birthing singleton infants. Assume these numbers are known with certainty, from many years of field research. Now suppose you are managing a captive panda breeding program. You have a new female panda of unknown species, and she has just given birth to twins. What is the probability that her next birth will also be twins?

```
# Givens
rate_a <- .5
rate_b <- .5
twin_rate_a <- .1
twin_rate_b <- .2
# Need to solve:
# P(twins) = P(twins|A)*P(A) + P(twins|B)P(B)
sum_probability_twins <- rate_a * twin_rate_a + rate_b * twin_rate_b # norm factor

pA_given_twins <- (twin_rate_a * rate_a) / sum_probability_twins
pB_given_twins <- (twin_rate_b * rate_b) / sum_probability_twins
p_twins <- twin_rate_a * pA_given_twins + twin_rate_b * pB_given_twins
p_twins
```

```
## [1] 0.1666667
```

16.7% chance

2H2.

> Recall all the facts from the problem above. Now compute the probability that the panda we have is from species A, assuming we have observed only the first birth and that it was twins.

```
pA_given_twins
```

```
## [1] 0.3333333
```

33% chance

2H3.

> Continuing on from the previous problem, suppose the same panda mother has a second birth and that it is not twins, but a singleton infant. Compute the posterior probability that this panda is species A.

```
# Givens
updated_rate_a <- pA_given_twins
updated_rate_b <- pB_given_twins
single_rate_a <- 1-twin_rate_a
single_rate_b <- 1-twin_rate_b

# Repeat calculations for new single birth
# norm factor
```

```r
sum_probability_single <- single_rate_a * twin_rate_a +
  single_rate_b * twin_rate_b
# Calculate probabilities
pA_given_single <- (single_rate_a * twin_rate_a) / sum_probability_single
pB_given_single <- (single_rate_b * twin_rate_b) / sum_probability_single
pA_given_single
```

```
## [1] 0.36
```

36% chance

2H4.

> A common boast of Bayesian statisticians is that Bayesian inference makes it easy to use all of the data, even if the data are of different types. So suppose now that a veterinarian comes along who has a new genetic test that she claims can identify the species of our mother panda. But the test, like all tests, is imperfect. This is the information you have about the test: - The probability it correctly identifies a species A panda is 0.8. - The probability it correctly identifies a species B panda is 0.65. The vet administers the test to your panda and tells you that the test is positive for species A. First ignore your previous information from the births and compute the posterior probability that your panda is species A. Then redo your calculation, now using the birth data as well.

Starting with data-free solution:

```r
# Givens
a_given_pos <- 0.8
b_given_pos <- 1 - a_given_pos
b_given_neg <- 0.65
b_given_pos <- 1 - b_given_neg

pA_given_test <- a_given_pos * rate_a /
                 (a_given_pos * rate_a + b_given_pos * rate_b)
pA_given_test
```

```
## [1] 0.6956522
```

Probability of A given test returns an "A" reading is 69.6%

Now adding the data:

```r
# P(A| positive test, twins, single) =
# P(positive test|A) * P(twins|A) * P(single|A) * P(A) /
# P(postive test, twins, single)
numerator <- a_given_pos * twin_rate_a * single_rate_a * rate_a

# P(positive test, twins, single) =
# P(positive test|A) * P(twins|A) *  P(single|A) * P(A) +
# (positive test|B) * P(twins|B) *  P(single|B) * P(B)
denom <- numerator + b_given_pos * twin_rate_b * single_rate_b * rate_b

numerator/denom
```

```
## [1] 0.5625
```

56.25% chance

Chapter 3 problems

Given:

```
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)
set.seed(100)
trial_size <- 1e4 # Tyler added
samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
```

3E1.

How much posterior probability lies below p = 0.2?

```
sum(samples < 0.2) / trial_size
```

```
## [1] 4e-04
```

3E2.

How much posterior probability lies above p = 0.8?

```
sum(samples > 0.8) / trial_size
```

```
## [1] 0.1116
```

3E3.

How much posterior probability lies between p = 0.2 and p = 0.8?

```
sum(samples < 0.8 & samples > 0.2) / trial_size
```

```
## [1] 0.888
```

3E4.

20% of the posterior probability lies below which value of p?

```
quantile(samples, 0.2)
```

```
##       20%
## 0.5185185
```

3E5.

20% of the posterior probability lies above which value of p?

```
quantile(samples, 1-0.2)
```

```
##       80%
## 0.7557558
```

3E6.

Which values of p contain the narrowest interval equal to 66% of the posterior probability?

```
HPDI(samples,prob=.66)
```

```
##     |0.66     0.66|
## 0.5085085 0.7737738
```

3E7.

Which values of p contain 66% of the posterior probability, assuming equal posterior probability both below and above the interval?

```
PI(samples,prob=.66)
```

```
##       17%       83%
## 0.5025025 0.7697698
```

3M1.

Suppose the globe tossing data had turned out to be 8 water in 15 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as before.

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom( 8 , size=15 , prob=p_grid )
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```

3M2.

Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for p.

```
samples <- sample(p_grid, prob=posterior, size=1e5, replace=TRUE)
HPDI(samples, prob=0.9)
```

```
##      |0.9      0.9|
## 0.3413413 0.7267267
```

3M3. >Construct a posterior predictive check for this model and data. This means simulate the distribution of samples, averaging over the posterior uncertainty in p. What is the probability of observing 8 water in 15 tosses?

```
simulations <- 1e4
w <- rbinom( simulations, size=15 , prob=samples )
sum(w==8)/simulations
```

```
## [1] 0.1473
```

3M4.

Using the posterior distribution constructed from the new (8/15) data, now calculate the probability of observing 6 water in 9 tosses.

```
simulations <- 1e4
w_2 <- rbinom( simulations, size=9 , prob=samples )
sum(w==6)/simulations
```

```
## [1] 0.1108
```

3M5.

Start over at 3M1, but now use a prior that is zero below p=0.5 and a constant above p=0.5. This corresponds to prior information that a majority of the Earth's surface is water. Repeat each problem above and compare the inferences. What difference does the better prior make? If it helps, compare inferences (using both priors) to the true value p = 0.7.

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior_const <- 1
prior <- (p_grid >= 0.5) * prior_const
prob_data_new <- dbinom( 8 , size=15 , prob=p_grid )
```

```
posterior_new <- prob_data_new * prior
posterior_new <- posterior_new / sum(posterior)
samples_new <- sample(p_grid, prob=posterior_new, size=1e5, replace=TRUE)
```

Tackling all the old problems:

```
print("Problem 2")
```

```
## [1] "Problem 2"
```

```
HPDI(samples_new, prob=0.9)
```

```
##      |0.9      0.9|
## 0.5005005 0.7117117
```

```
print("Problem 3")
```

```
## [1] "Problem 3"
```

```
simulations <- 1e4
w <- rbinom( simulations, size=15 , prob=samples_new )
sum(w==8)/simulations
```

```
## [1] 0.1634
```

```
print("Problem 4")
```

```
## [1] "Problem 4"
```

```
w_2 <- rbinom( simulations, size=9 , prob=samples_new )
sum(w==6)/simulations
```

```
## [1] 0.0659
```

HPDI is far narrower. Likelihood of 8/15 is slightly increased, likelihood of 6/9 increases considerably - effectively we've removed the opportunity for fewer than 50% water cases to be considered, which will subsequently increase the likelihood of all >50% cases.

3M6.

> Suppose you want to estimate the Earth's proportion of water very precisely. Specifically, you want the 99% percentile interval of the posterior distribution of p to be only 0.05 wide. This means the distance between the upper and lower bound of the interval should be 0.05. How many times will you have to toss the globe to do this?

```
interval_width <- 1
nSimulations <- 0
p <- 0.7
while (interval_width > 0.05)
{
  nSimulations <- nSimulations + 10
  p_grid <- seq( from=0 , to=1 , length.out=1000 )
  prob_p <- rep( 1 , 1000 )

  # Simulate data
  simulations <- nSimulations
  likelihood <- dbinom( round(simulations*p), size=simulations, prob=p_grid )
  posterior <- likelihood * prob_p
  posterior <- posterior / sum(posterior)
  #print(posterior)
```

```
  trial_size <- 1e4
  #print(trial_size)
  samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
  interval_width <- quantile(samples,0.995) - quantile(samples, 0.005)
}
nSimulations
```

```
## [1] 2200
```

About 2200 trials.

3H1.

> Using grid approximation, compute the posterior distribution for the probability of a birth being a boy. Assume a uniform prior probability. Which parameter value maximizes the posterior probability?

```
all_births <- c(birth1,birth2)
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 ) #Uniform
likelihood <- dbinom( sum(all_births) , size=length(all_births) , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
p_grid[which.max(posterior)]
```

```
## [1] 0.5545546
```

3H2.

> Using the sample function, draw 10,000 random parameter values from the posterior distri- bution you calculated above. Use these samples to estimate the 50%, 89%, and 97% highest posterior density intervals.

```
trial_size <- 10000
samples <- sample( p_grid , prob=posterior , size=trial_size , replace=TRUE )
HPDI(samples, prob=.5)
```

```
##      |0.5      0.5|
## 0.5305305 0.5775776
```

```
HPDI(samples, prob=.89)
```

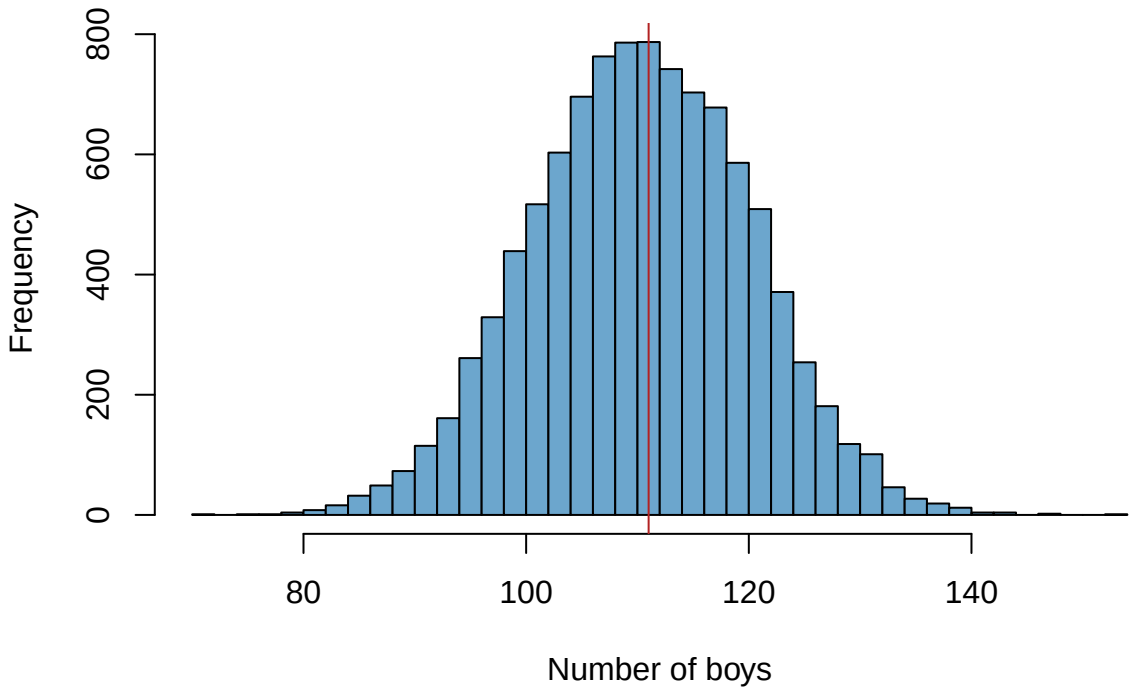
```
##     |0.89     0.89|
## 0.5005005 0.6116116
```

```
HPDI(samples, prob=.97)
```

```
##     |0.97     0.97|
## 0.4794795 0.6296296
```

3H3. Use rbinom to simulate 10,000 replicates of 200 births. You should end up with 10,000 numbers, each one a count of boys out of 200 births. Compare the distribution of predicted numbers of boys to the actual count in the data (111 boys out of 200 births). There are many good ways to visualize the simulations, but the dens command (part of the rethinking package) is probably the easiest way in this case. Does it look like the model fits the data well? That is, does the distribution of predictions include the actual observation as a central, likely outcome?

```
sim <- rbinom(10000, size=200, prob=samples)
hist(sim, c="skyblue3", breaks=50, xlab="Number of boys", main="")
abline(v=sum(all_births), col="firebrick")
```

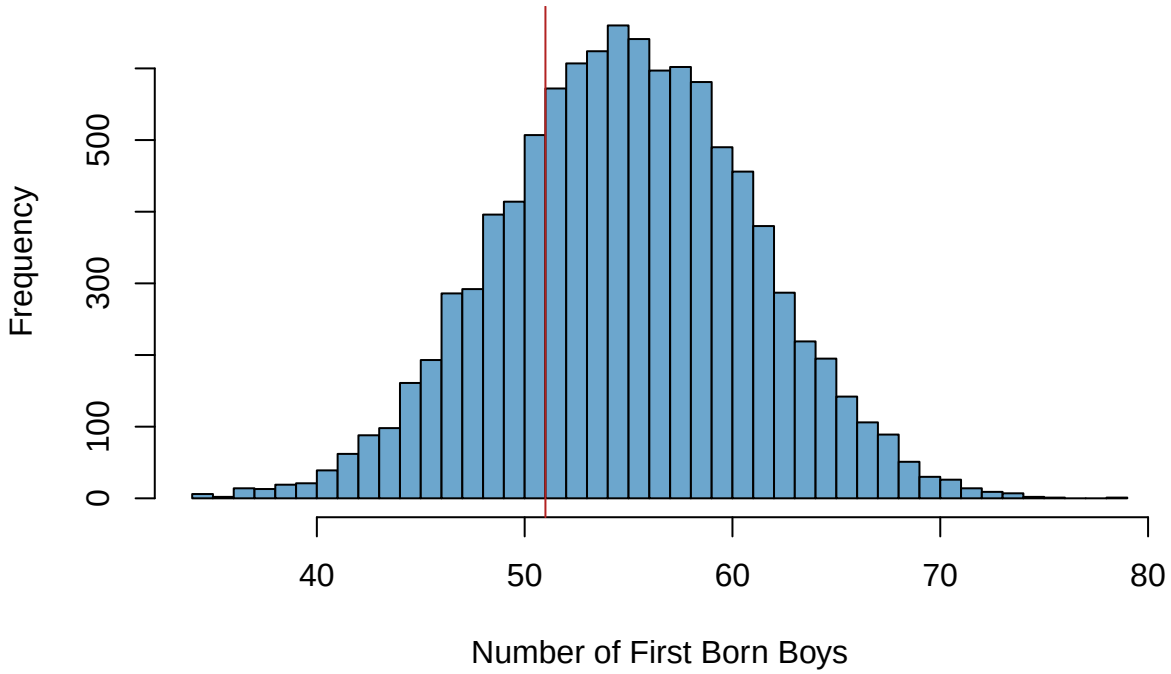


Fits right on mean, this seems like a likely outcome.

3H4.

Now compare 10,000 counts of boys from 100 simulated first borns only to the number of boys in the first births, birth1. How does the model look in this light?

```
sim <- rbinom(10000, size=100, prob=samples)
hist(sim, c="skyblue3", breaks=50, xlab="Number of First Born Boys", main="")
abline(v=sum(birth1), col="firebrick")
```



It's not on the maximum likelihood location, but it's still a reasonable value.

```
sprintf("Value: %i",sum(birth1))
```

```
## [1] "Value: 51"
```

```
PI(sim,prob=0.60)
```

```
## 20% 80%
##  50  61
```

The value is within the inner 60% of posterior density

3H5.

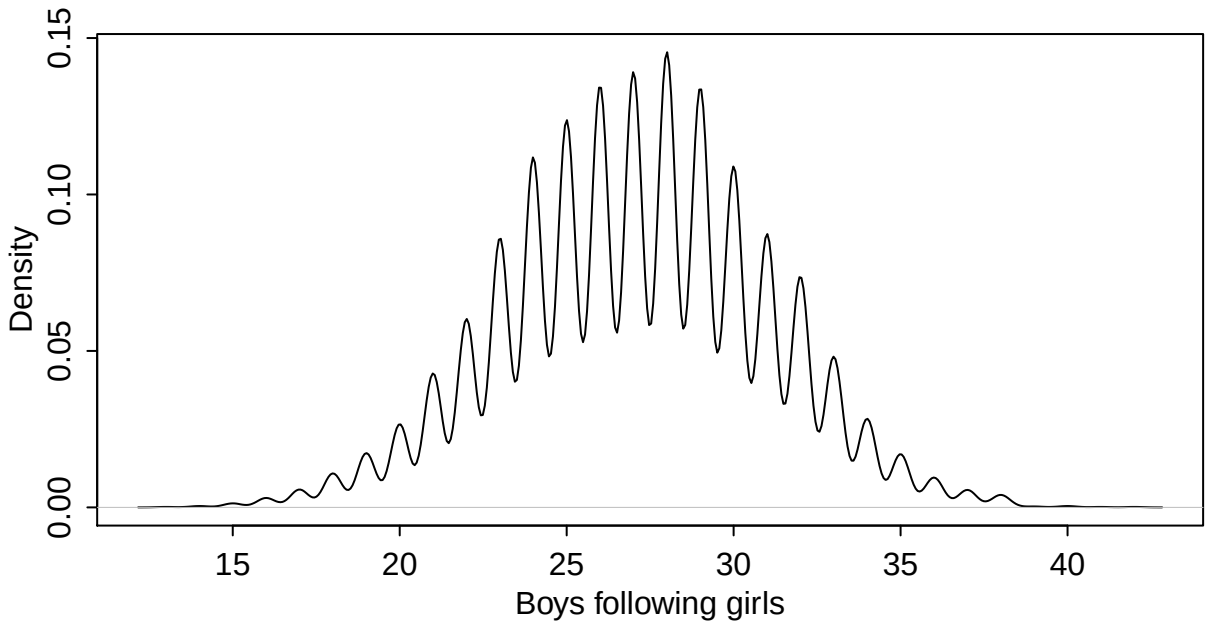
The model assumes that sex of first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only those second births that followed girls. To do this correctly, you need to count the number of first borns who were girls and simulate that many births, 10,000 times. Compare the counts of boys in your simulations to the actual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?

```
boys_after_girls <- birth2[birth1==0 & birth2 ==1]
sum(boys_after_girls)
```

```
## [1] 39
```

39 cases of boys born after a girl

```
count_first_girls <- sum(birth1==0)
sim_girl <- rbinom(10000, size=count_first_girls, prob=samples)
dens(sim_girl, xlab="Boys following girls", main="")
```



This doesn't look like anything normal. The biggest thing is that binomial assumes that trials are independent, and it's very possible these are not.