Chapter 2: Bayes' theorem

Rafal Urbaniak and Nikodem Lewandowski

1 Bayesian inference

1.1 Odds and binomial probability distribution

Read/watch first

- *Bayesian statistics the fun way* (Will Kurt), Chapters 1-4.

If $\mathsf{P}(H)$ is the probability of H, then $\frac{\mathsf{P}(H)}{\mathsf{P}(\neg H)}$ is the *odds* of H. These are interchangeable, and can be translated back and forth as follows. First let's do this for one case.

```
pH <- 0.3  #example of a pr of H
pnH <- 1 - pH #calculate the probability of not-H
odds <- pH/pnH
odds
```

```
## [1] 0.4285714
```

We can generalize and define a function that takes an arbitrary probability and converts it to odds.

```
prToOdds <- function (pH){pH/ (1- pH)} #define the function
                                       #of one variable, denoted pH
prToOdds(.3)  #check if the result is the same
```

```
## [1] 0.4285714
```

The formula for the odds-to-probabilities direction is:

$$\mathsf{P}(H) = \frac{odds(H)}{odds(H)+1}. \tag{1}$$

Exercise 1

Write a function that converts odds to probabilities, call it oddsToPr. Apply it to odds = .42 and check if you get the result approximately close to .3.

Binomial coefficient: the number of ways you can choose k objects from among n objects, where the order doesn't matter. Mathematically, count all permutations of n objects, divide by the number of permutations of your subset, because the order within it doesn't matter, and divide by the number of permutations of the objects not in your subset, because their ordering doesn't matter either. (Note for future reference that that $0! = 1$).

$$\binom{n}{k} = \frac{n!}{k! \times (n-k)!} \tag{2}$$

In **R**, this is a one-liner.

```
choose(3,2) #ways to choose 2 objects from among 3 objects
```

```
## [1] 3
```

What's the probability of getting exactly 3 heads in 4 tosses of a fair coin?

- First, count ways you could get 3 heads in 5 tosses, $\binom{5}{3}$.
- Then, calculate the probability of each such an outcome: $\mathrm{P}(heads)^3 \times \mathrm{P}(1-\mathrm{P}(heads))^{5-3}$.
- Finally multiply each such probability by the number of the ways.

The general formula for binomial probability calculation requires: the number of trials n, the number of successes k, the probability of success in each trial, p. Denote the probability of exactly k sucesses in n trials with fixed success probability p as $B(k,n,p)$.

$$B(k,n,p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{3}$$

This can be easily calculated in **R**. Say we're tossing a fair coin 24 times, and we want to know what's the probability of obtaining exactly 12 heads:

```
dbinom(x = 12, size = 24, prob = .5)
```

```
## [1] 0.1611803
```

```
dbinom(12,24,.5) #note argument names are optional
```

```
## [1] 0.1611803
```

If you specify the parameters in the first line (successes, trials, probability at each trial), you can give them in any order. If you just give the numerical values (as in the second line above), this is exactly the order in which they will be interpreted.

Exercise 2

What is the probability of obtaining exactly 3 heads in 6 tosses, if the coin is unfair and the probability of heads is .6? What is the probability of 3 heads in 6 tosses, if the probability of success is .55?

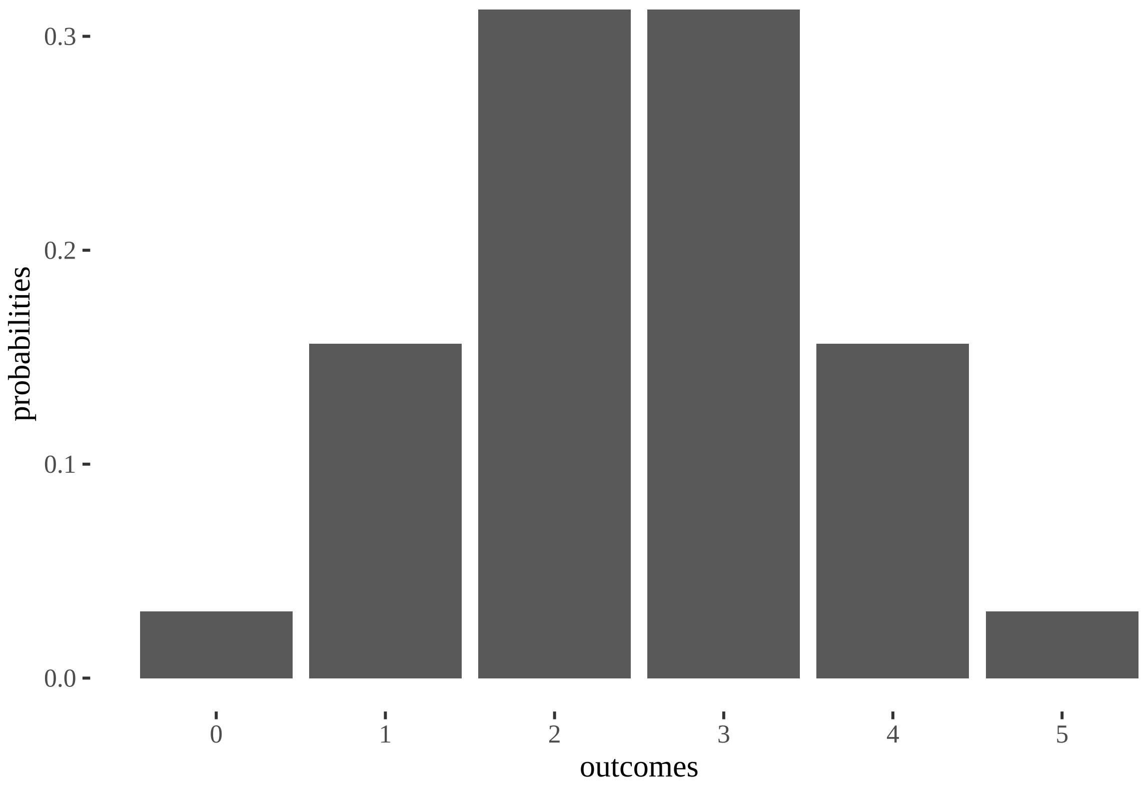
Now, let's take five tosses of fair coins and obtain the probability mass function for the possible outcomes. For five tosses there are six possible outcomes: 0, 1, 2, 3, 4, 5. Note the use of vectorized calculations.

```
outcomes <- 0:5 #create sequence 0, ..., 5, call it outcomes
probabilities <- dbinom(outcomes, 5, .5)
probabilities
```

```
## [1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
```

We can visualise this PMF using a barplot in ggplot2.

```
#put together with format: column name = data
table <- data.frame(outcomes = outcomes,
                    probabilities =  probabilities)

#now plot
ggplot(table, aes(x = outcomes, y = probabilities))+  #set up
geom_bar(stat="identity")+  #barplot layer
  scale_x_continuous(breaks = outcomes)+ th #fix ticks on x axis
```



Exercise 3

Calculate and visualise the PMF for the outcomes of 15 coin tosses with the probability of success = .7.

The $e-$ notation might look fun. This is the so-called standard index form for numbers that are too small or too large to write normally. A nonzero number can be written as $m \times 10^n$, which is written also as men. For instance, 0.02 is 2×10^2, that is, $2e-2$, and 2000 is 2×10^3, that is $2e3$. Basically, think of the number after e to as telling you how many slots you need to move the decimal point, and the sign (plus or minus) as telling you which direction to move it.

Sometimes we will not be interested in the probability of exactly k successes, but rather in the probability of at most k successes (let's call this the *cumulative (probability) mass function*), or that the number of successes is greater than k. Mathematically, these can be obtained by summing all the relevant probabilities. Programmatically, we can use pbinom.

```
pbinom(3,6,.5) #at most 3 successes in 6 tosses of a fair coin
```

```
## [1] 0.65625
```

```
pbinom(3,6,.5, lower.tail = FALSE) # at least 4 successes
```

```
## [1] 0.34375
```

```
pbinom(3,6,.5) + pbinom(3,6,.5, lower.tail = FALSE) #note these should add to 1
```

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

Exercise 4

Calculate and visualize the cumulative probabilities for the outcomes of 10 coin tosses with the probability of success = .3. The plot should be increasing and approaching 1.

1.2 Beta distributions

Read/watch first

- *Bayesian statistics the fun way* (Will Kurt), Chapter 5.

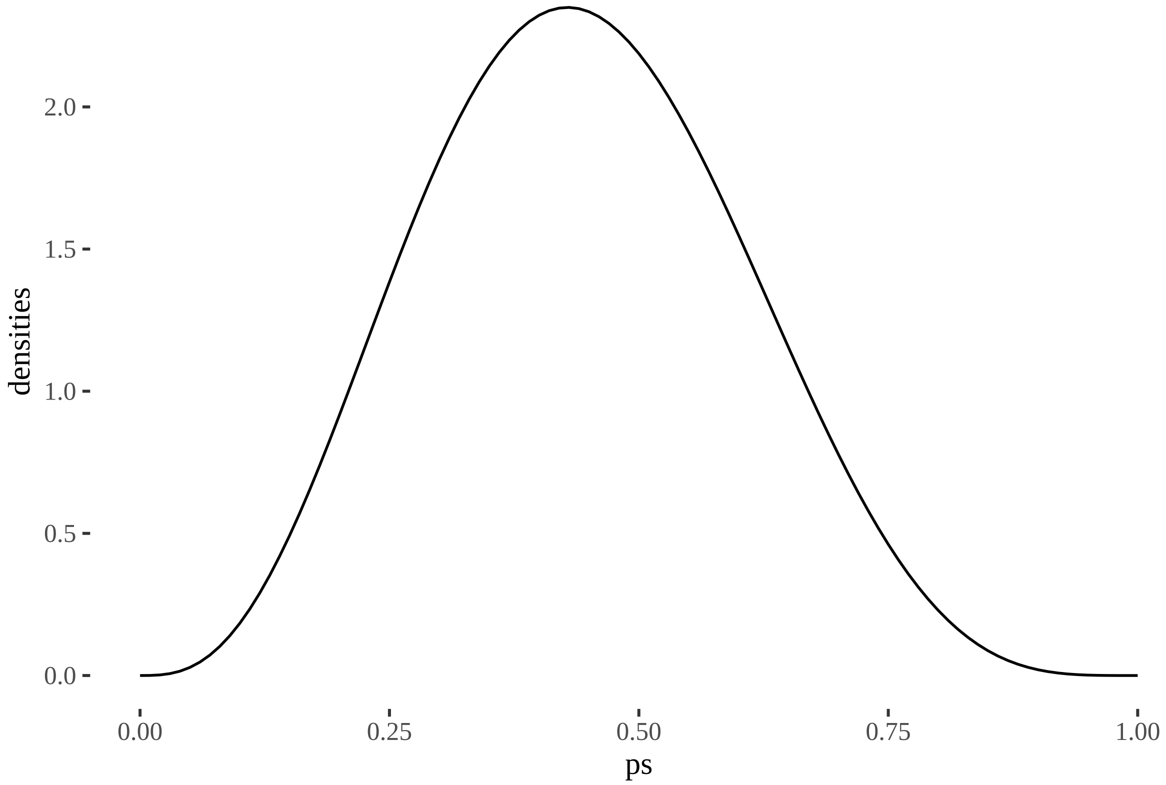
Recall, beta distribution can be thought as a continous generalization of the binomial distribution.

$$\beta(p,a,b) = \frac{p^{a-1}(1-p)^{b-1}}{Beta(a,b)} \tag{4}$$

where $Beta(a,b)$ is a normalizing constant (the integral from 0 to 1 of $p^{a-1} \times (1-p)^{b-1}$.), p is the probability of an event (success), a expresses how many times the event has been (or is imagined to have been) observed, and b how many a failure has been (or is imagined to have been) observed, and the total number of trials is $a+b$.

Just like we calculated probabilities for the binomial distribution, we can calculate densities for the beta distribution as well. Let's do so with a visualisation, say for $a = 4, b = 5$. This time we'll use a line, not a barplot. This is, intuitively, what you belief distribution should be in the actual bias of a coin if all that you have observed is four heads and five tails in nine tosses. Moreover, note how we run calculations only for a finite selection of points – this is called *grid appoximation*.

```
ps <- seq(0,1,by = 0.01) #note we take a selection of p
                         #to use in grid approximation
densities <- dbeta(ps,4,5)
table <- data.frame(p = ps, density = densities)

ggplot(table, aes(x = ps, y = densities))+ th+
geom_line()
```



Observe that the values go above 1. This clearly suggests that they aren't probabilities. And indeed, probabilities correspond rather to areas under the curve not to the curve itself (the curve is just the derivative of cumulative probability distribution).

Observe that you can find the "center" of the distribution and its variance using dBETA from the fitODBOD package. The first output is just the same as that of dbeta, so we look only at the remaining ones.

```
dBETA(ps,4,5)[-1]
```

```
## $mean
## [1] 0.4444444
##
## $var
## [1] 0.02469136
```

Note that the distribution is simply centered around the frequency of heads.

```
4/9
```

```
## [1] 0.4444444
```

To calculate probabilities for various ranges of p values, we need to calculate areas under the curve within certain limits. For instance, to calculate the probability that $p \leq .5$ we calculate the area under the curve between 0 and .5, we do so by integrating.

```
integrate ( function (ps) dbeta(ps,4,5), lower = 0, upper = .5)
```

```
## 0.6367187 with absolute error < 7.1e-15
```

Exercise 5

With the same beta distribution, what's the probability that p is between .4 and .7?

Exercise 6

Now suppose the distribution captures 25 observations, 17 of which were heads. Visualise the beta distribution. What is its mean and variance? What's the probability that p is above .7?

1.3 Conditional probability and Bayes' theorem

Read/watch first

- *Bayesian statistics the fun way* (Will Kurt), Chapters 6, 7.
- *Statistical rethinking* (Richard McElreath), Section 2.1.

The *product rule* has it that:

$$\mathrm{P}(B \wedge A) = \mathrm{P}(B)\mathrm{P}(A|B) = \mathrm{P}(A)\mathrm{P}(B|A) \tag{5}$$
$$P(B \wedge A) = P(B)P(A|B) = P(A)P(B|A) \tag{6}$$

For instance, if the probability that you will study half an hour a day for this course is .1, and the probability that you will pass if you do study that much is .8, the probability that you will study that much and pass can be calculated as follows:

```
.1 * .8
```

```
## [1] 0.08
```

Exercise 7

Say the probability that you will fail if you don't study that much is .75. What's the probability that you will study and fail? What's the probablity that you will not study and pass?

Bayes' theorem, in its simplest formulation, is:

$$\mathsf{P}(H|E) = \frac{\mathsf{P}(E|H)\mathsf{P}(H)}{\mathsf{P}(E)} \tag{7}$$

For instance suppose that our evidence E is that you passed the exam, and I'm interested in the conditional probability of the hypothesis H that you in fact studied hard based on this evidence. Suppose further that 40% students pass, and the probabilities of studying hard and passing if you do are as before (.1 and .8). The required conditional probability can be calculated like this (for clarity, we play with variable assignents in the process):

```
pass <- .4
passIfStudy <- .8
study <- .1
studiedIfPassed <-    (passIfStudy * study)  /   (pass)
studiedIfPassed
```

```
## [1] 0.2
```

Now, consider two categorical variables and what a two-way probability distribution for them specifies. Let's take the example from Kruschke:

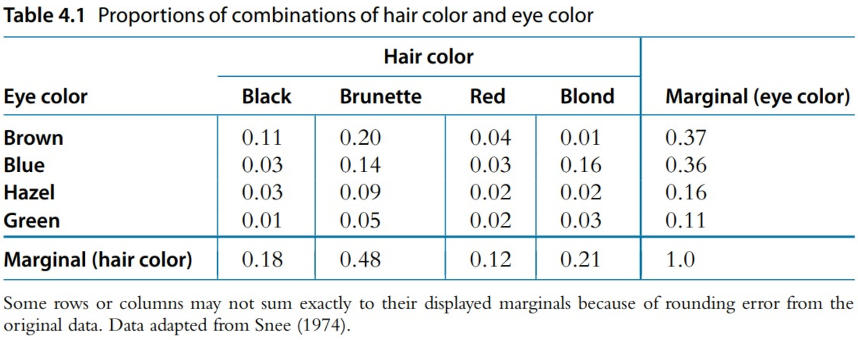


Table 4.1 Proportions of combinations of hair color and eye color

	Hair color				
Eye color	**Black**	**Brunette**	**Red**	**Blond**	**Marginal (eye color)**
Brown	0.11	0.20	0.04	0.01	0.37
Blue	0.03	0.14	0.03	0.16	0.36
Hazel	0.03	0.09	0.02	0.02	0.16
Green	0.01	0.05	0.02	0.03	0.11
Marginal (hair color)	0.18	0.48	0.12	0.21	1.0

Some rows or columns may not sum exactly to their displayed marginals because of rounding error from the original data. Data adapted from Snee (1974).

- The main cells contain joint probabilities.
- The marginal probabilities are of the form $p(e) = \sum_h p(e,h)$ or $p(h) = \sum_e p(e,h)$ and are used in normalization for calculating conditional probabilities.
- In continuous cases: $p(r,c)$ is a pdf (probability density function, the derivative of cumulative probability), and the summation is an integral, $p(r) = \int \mathrm{d}c\, p(r,c)$, where $p(r)$ is also a pdf.

Conditional probability can be though of in terms of rows and columns. In a discrete case, we have:

$$\mathsf{P}(c|r) = \frac{\mathsf{P}(r,c)}{\sum_{c^\star} \mathsf{P}(r,c^\star)} = \frac{\mathsf{P}(r,c)}{\mathsf{P}(r)} \tag{8}$$

In the continuous case, this becomes:

$$\mathsf{P}(c|r) = \frac{\mathsf{P}(r,c)}{\int \mathrm{d}c\, \mathsf{P}(r,c)} = \frac{\mathsf{P}(r,c)}{\mathsf{P}(r)} \tag{9}$$

In this setup, the formulaic version of *Bayes' rule* is:

$$\mathsf{P}(c|r) = \frac{\mathsf{P}(r|c)p(c)}{\sum_{c^\star} \mathsf{P}(r|c^\star)\mathsf{P}(c^\star)} \tag{10}$$

A tabular intuition behind this can be illustrated as follows:

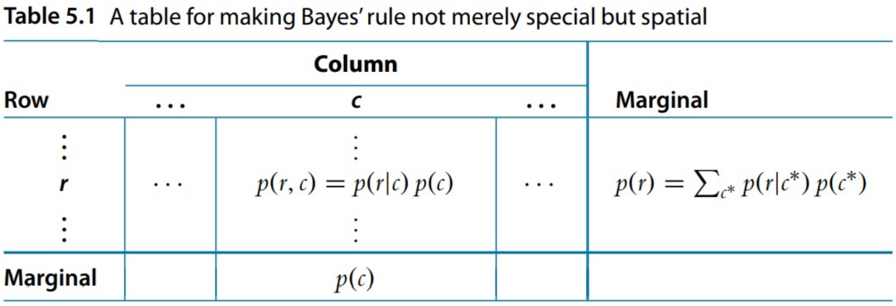


Table 5.1 A table for making Bayes' rule not merely special but spatial

	Column			
Row	...	*c*	...	**Marginal**
⋮		⋮		
r	⋯	$p(r,c) = p(r\|c)\,p(c)$	⋯	$p(r) = \sum_{c^*} p(r\|c^*)\,p(c^*)$
⋮		⋮		
Marginal		$p(c)$		

Exercise 8

Suppose the prior probability of a student liking statistical programming is 0.3. The probability that someone who passed this course likes statistical programming is .6. Moreover, the prior probability of passing the course is .4. What's the probability that someone who likes statistical programming passes the course?

1.4 Prior, likelihood and posterior

Read/watch first

- *Bayesian statistics the fun way* (Will Kurt), Chapters 8.
- *Statistical rethinking* (Richard McElreath), Section 2.2., introudction to Chapter 3 (before Section 3.1)

Recall the following terminology related to Bayes' Theorem:

$$\underbrace{\mathsf{P}(H|E)}_{\text{posterior}} = \frac{\overbrace{\mathsf{P}(E|H)}^{\text{likelihood}}\ \overbrace{\mathsf{P}(H)}^{\text{prior}}}{\underbrace{\mathsf{P}(E)}_{\text{normalization factor/base rate}}} \tag{11}$$

Let's follow the example used in Ch. 8 of the book:

> You come home from work one day and find your window broken, your front door open, and your laptop missing. Your first thought is probably "I've been robbed!".

We need to evaluate the probability of your hypothesis given the evidence. First, we need the likelihood: the probability of the evidence assuming the hypothesis. The example goes: say it's 1/3, and say the prior probability of being robbed is 1/1000. Unnormalized posterior is the prior times likelihood.

```
likelihood <- 3/10
prior <- 1/1000
posteriorUnnormalized <- likelihood * prior
posteriorUnnormalized == 3/10000
```

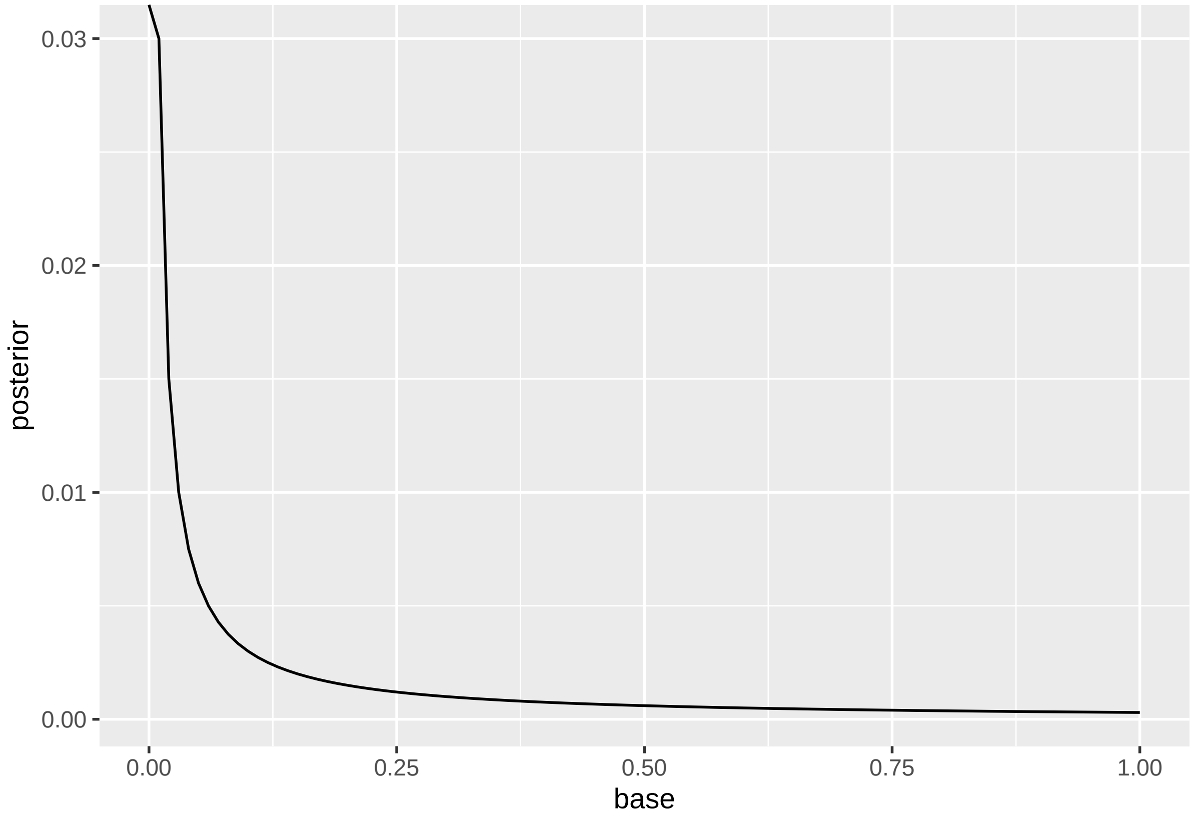
```
## [1] TRUE
```

The key fact about unnormalized posterior is that it is proportional to the normalized posterior:

$$\mathsf{posterior} \propto \mathsf{likelihood} \times \mathsf{prior}$$

We can now easily inspect the impact of the choice of normalization factor on the posterior. The book takes 4 values. We'll take more and visualise.

```
base <- seq(0,1, by = 0.01)
posterior <- posteriorUnnormalized/base

table <- data.frame(base = base,
                    posterior =  posterior)
#put together with format: column name = data
ggplot(table, aes(x = base, y = posterior))+  #set up
geom_line()
```



One key lesson is that the posterior probability is highly sensitive to the base rate. So we'd better think hard about how we choose it. One way to go is to consider alternative hypotheses. Say here we have one, H_A: you left the door unlocked, you left your laptop at work, and kids broke the window with a ball. The probability of your evidence given this hypothesis is pretty much 1. If H and H_A are exclusive and jointly exhaustive, the *law of total probability* say:

$$\mathsf{P}(E) = \mathsf{P}(E|H)\mathsf{P}(H) + \mathsf{P}(E|H_A)\mathsf{P}(H_A) \tag{12}$$

We need one more thing: $\mathsf{P}(H_A)$ We're invited to think of this three-fold alternative explanation as follows. The probability of a ball breaking the window is 1/2000, the probability of leaving door unlocked is 1/30 and the probability of leaving the laptop at work is 1/365. Supposing these events are independent, the probability of the whole alternative hypothesis is the result of multiplying these values.

```
HA <- 1/2000 * 1/30 * 1/365
HA == 1/21900000
```

```
## [1] TRUE
```

```
HA
```

```
## [1] 4.56621e-08
```

The unnormalized posterior probability for H_A is:

```
EifHA <- 1
posteriorEifHAunnormalized <- EifHA * HA
posteriorEifHAunnormalized
```

```
## [1] 4.56621e-08
```

It is the same as the prior of H_A, because the corresponding likelihood is unexcitingly 1.

One thing we can look at is the ratio of two unnormalized posteriors:

```
ratio <- posteriorUnnormalized / posteriorEifHAunnormalized
ratio ==   (3/10000) / (1/21900000)
```

```
## [1] TRUE
```

```
ratio
```

```
## [1] 6570
```

It indicates how many times better H explains the evidence than H_A.

Now let's plug the values to LOTP to calculate the base and use it with Bayes theorem to calculate the posterior probability of H (and the posterior probability of H_A).

```
base <- posteriorUnnormalized + posteriorEifHAunnormalized
posterior <- posteriorUnnormalized/base
posteriorHA <- posteriorEifHAunnormalized/base
posterior
```

```
## [1] 0.9998478
```

```
posteriorHA
```

```
## [1] 0.0001521838
```

Exercise 9

Extend the previous situation with a second alternative hypothesis H_B which says that your annoying brother decided to play a stupid joke on you. Say your prior of H_B is .05, and the probability that he'd hide your laptop, break the window and leave the door unlocked if he wanted to play a stupid joke on you is .001. What are the posterior probabilities of all the hypotheses involved now, rounded to three digits?

Your posteriors for H, H_A and H_B should be, respectively:

```
## [1] 0.8570 0.0001 0.1428
```

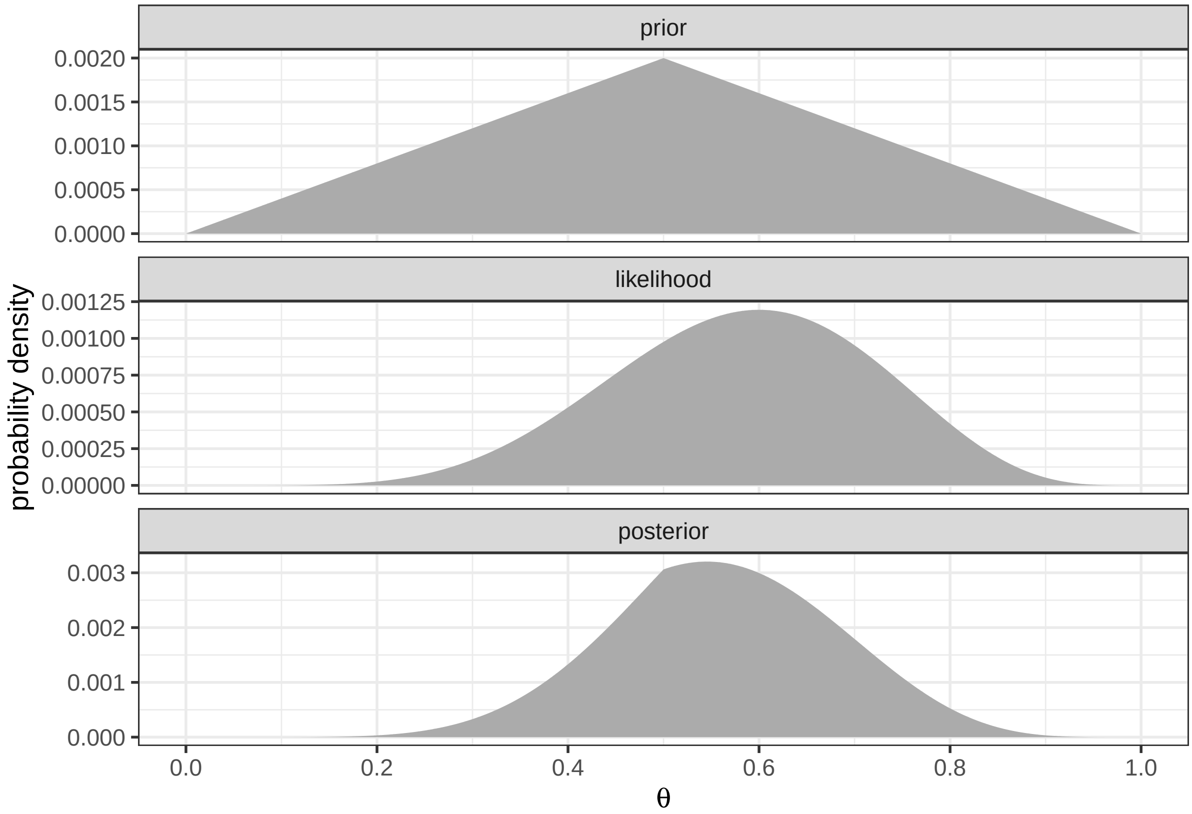
One advantage of working with grid approximations instead of beta functions is flexibility when it comes to the scale of the distribution. For instance, what if for some reason your prior looks more like a triangle?

```
# define bernoulli likelihood function
bernoulliLikelihood <- function(theta, data) {
    # `theta` = success probability parameter ranging from 0 to 1
    # `data` = the vector of data (i.e., a series of 0s and 1s)
    n   <- length(data)
    return(theta^sum(data) * (1 - theta)^(n - sum(data)))
}

smallData <- rep(0:1, times = c(4, 6)) #6 heads in 10 tosses.

#possible parameters
theta =   seq(from = 0,     to = 1, by = .001)

#triangular sequence of values
prior = c(seq(from = 0,     to = 1, length.out = 501),
          seq(from = 0.998, to = 0, length.out = 500))
prior <- prior / sum(prior)  # we need to rescale the shape, so
                             #  that the values sum up to 1

sum(prior)  #check if this worked
```

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

```
#now the likelihood function for our data and the possible parameters
likelihood <- bernoulliLikelihood(theta = theta, data = smallData)

#find out the numerator for Bayes theorem
```

```r
normalizationFactor <- sum(prior * likelihood)

#now use Bayes theorem
posterior <- (prior * likelihood) / normalizationFactor

#put them together
table <- data.frame(theta, prior, likelihood, posterior)

#now we convert this to long format for plotting
tableLong <- table %>% gather(key, value, -theta)

#the key column has to be made a factor with
#three levels in appropriate order now
tableLong$key <- factor(tableLong$key,
                        levels  = c("prior", "likelihood","posterior"))

ggplot(tableLong, aes(x = theta, ymin = 0, ymax = value)) +
  geom_ribbon(fill = "grey67") +
  scale_x_continuous(expression(theta), breaks = seq(from = 0, to = 1, by = .2)) +
  ylab("probability density") +
  facet_wrap(~key, scales = "free_y", ncol = 1)+#note this separates plots by key
  theme_bw()
```



Exercise 10

Now work with the same grid for theta, but suppose your unnormalized prior is flat at 1 up to 0.3, then flat at 3 up to 0.7, and then flat at 2 up to 1. Run the same analysis but suppose you observed 24 heads in 44 tosses.

In general, bayesian *model* specifies:

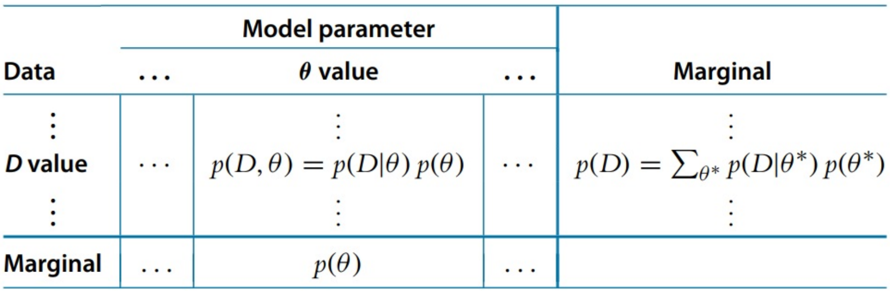
- P(data values|parameter values) (likelihood)
- P(parameter values) (prior)

and then the Bayesian calculation results in a posterior.

P(parameter values|data values)

Let's look at Kruschke's tabular presentation of this:

Table 5.5 Applying Bayes' rule to data and parameters



	Model parameter			
Data	. . .	θ value	. . .	Marginal
$\vdots$		$\vdots$		$\vdots$
D value	$\cdots$	$p(D,\theta) = p(D\|\theta)\,p(\theta)$	$\cdots$	$p(D) = \sum_{\theta^*} p(D\|\theta^*)\,p(\theta^*)$
$\vdots$		$\vdots$		$\vdots$
Marginal	. . .	$p(\theta)$	. . .	

Let's think again about the *Bernoulli setup*. While this is not really about coins (rather it's about any chances that you want to estimate if individual outcomes are binary and cases are independent), it makes sense to abstract from particularities and imagine that you're tossing a coin with an unknown parameter θ, and want to estimate θ. What you observe is the results of particular tosses. Say you toss a coin once, and get the data $\gamma = 1$ (heads) or $\gamma = 0$ (tails). Clearly:

- $\mathsf{P}(\gamma = 1|\theta) = \theta$.
- $\mathsf{P}(\gamma = 0|\theta) = 1 - \theta$.

These two can be combined into a single formula:

$$\mathsf{P}(\gamma|\theta) = \theta^{\gamma}(1-\theta)^{(1-\gamma)}. \tag{13}$$

Now say you toss the coin multiple times and let the outcome of the i-th flip be γ_i, think data= γ_i.

$$\begin{aligned}
\mathsf{P}(\gamma_i) &= \prod_i \mathsf{P}(\gamma_i|\theta) && (14)\\
&= \prod_i \theta^{\gamma_i}(1-\theta)^{(1-\gamma_i)} && (15)\\
&= \theta^{\sum_i \gamma_i}(1-\theta)^{\sum_i (1-\gamma_i)} && (16)\\
&= \theta^{\#\text{heads}}(1-\theta)^{\#\text{tails}} && (17)
\end{aligned}$$

Let's take an unrealistically simple case of a mystery coin in which after a fair draw of one of two biased coins, with biases 0.4 and 0.6, the coin is tossed and you're supposed to have some distribution over the possible biases after you observe the result.

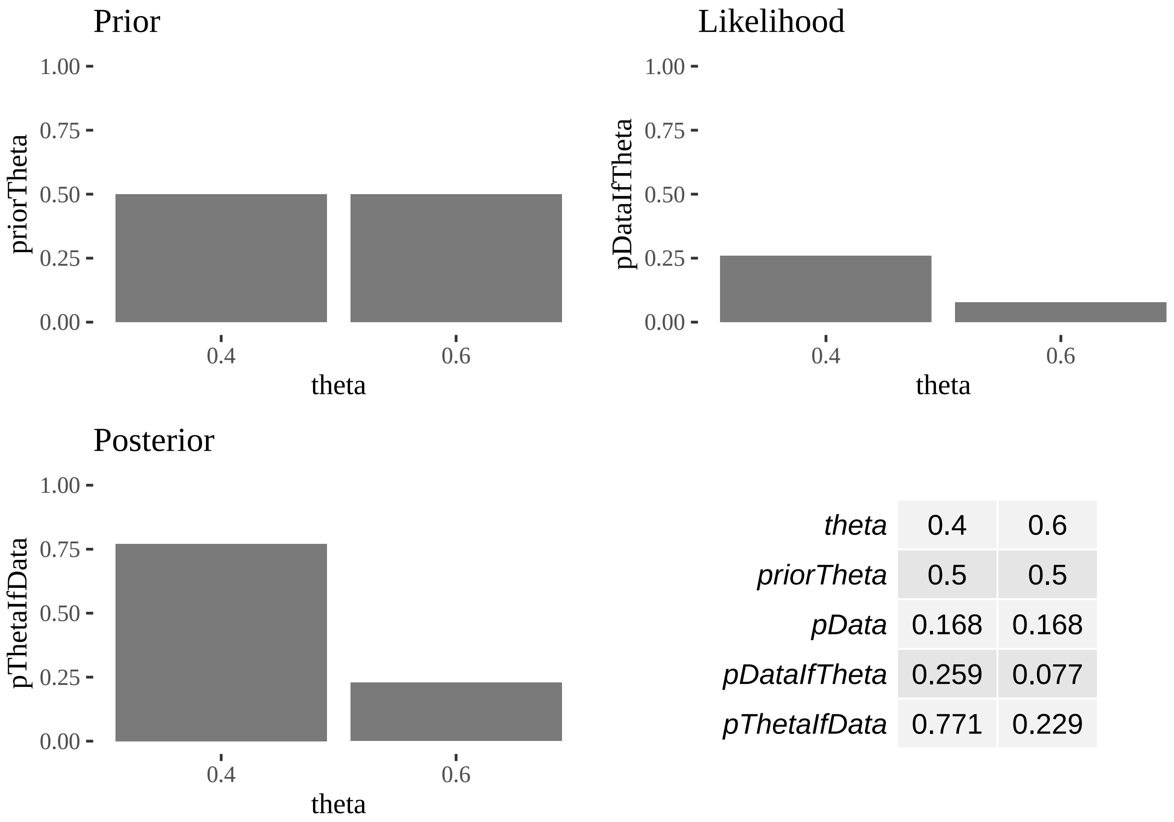
- The only candidate parameters are 0.4 and 0.6.
- Your prior gives probability of .05 to each of them, as the draw of the coin is fair.
- Say you toss five times and observe $\gamma = 1$, one head heads.

you need to describe your prior, calculate the likelihood and use these to calculate the posterior. First, let's do this in slow motion.

```
theta <- c(0.4, 0.6) #possible parameter values
priorTheta <- c(0.5,0.5) #priors assigned to them
data <- c(1,0,0,0,0) #1 heads with five tosses
z = sum(data) # number of 1's in Data
N = length(data) #number of tosses

# Compute the Bernoulli likelihood at each value of Theta:
pDataIfTheta <- dbinom(z,N,theta)
# Compute the probability of the evidence and
# the posterior via Bayes' rule:
pData <- sum(pDataIfTheta * priorTheta)
pThetaIfData <- pDataIfTheta * priorTheta / pData

#check the posteriors add to 1
sum(pThetaIfData)
```

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

```
#put together
table <- data.frame(theta,  priorTheta,
                     pData,  pDataIfTheta, pThetaIfData)


#plot of the prior
prior <- ggplot(table, aes(x = theta, y = priorTheta))+
  geom_bar(stat="identity", alpha = 0.8)+
  scale_x_continuous(breaks = theta)+theme_tufte()+ylim(c(0,1))+ggtitle("Prior")

#plot of the likelihood
likelihood <-  ggplot(table, aes(x = theta, y = pDataIfTheta))+
  geom_bar(stat="identity", alpha = 0.8)+
  scale_x_continuous(breaks = theta)+
  theme_tufte()+ylim(c(0,1))+ggtitle("Likelihood")


#plot of the posterior
posterior <- ggplot(table, aes(x = theta, y = pThetaIfData))+  #set up
  geom_bar(stat="identity", alpha = 0.8)+  #barplot layer
  scale_x_continuous(breaks = theta)+
  theme_tufte()+ylim(c(0,1))+ggtitle("Posterior")

#convert the table to graphics, transpose for readability
tableGG <- ggtexttable(t(round(table,3)))

#plot together
ggarrange(prior, likelihood, posterior, tableGG)
```



theta	0.4	0.6
priorTheta	0.5	0.5
pData	0.168	0.168
pDataIfTheta	0.259	0.077
pThetaIfData	0.771	0.229

Exercise 11

Run a similar analysis for the mystery coin, but this time suppose the possible biases are $0.4, 0.5$, and 0.6 and .8, your priors for these biases are $.1, .3, .4$ and .2, and you toss 15 times and observe $y = 6$, six heads.

Exercise 12

This is a variation on the globe scenario from *Statistical rethinking*, Section 2.2. The original story goes:

> Suppose you have a globe representing our planet, the Earth. This version of the world is small enough to hold in your hands. You are curious how much of the surface is covered in water. You adopt the following strategy: You will toss the globe up in the air. When you catch it, you will record whether or not the surface under your right index finger is water or land. Then you toss the globe up in the air again and repeat the procedure.

Now, suppose there are two globes, one for Earth and one for Mars. The Earth globe is 65% covered in water. The Mars globe is 95% land. 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.

- Calculate the posterior probability that the globe was the Earth.
- What is this probability if Mars is completely covered by land?
- What is this probability if Mars is completely covered by land, the Earth is 70% covered in water, and its chance of being chosen is .4?

Baby panda born in Malaysia in 2018

Exercise 13

Let's get serious. Pandas! 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 easy way to tell them apart. They differ in their family sizes. Assume that whether a subsequent birth is that of twins does not depend on whether the previous one was that of twins.

- Species A gives birth to twins 10% of the time, otherwise birthing a single infant.
- Species B births twins 20% of the time, otherwise a birthing single infant.

You manage a captive panda breeding program.

1. You have a new female panda of unknown species, and she has just given birth to twins. What is the probability that she belongs to species A?
2. What is now the probability that her next birth will also be twins?
3. The same panda mother has a second birth and it is not twins, but a singleton infant. Compute the posterior probability that this panda is species A.

Exercise 14

Now a veterinarian comes along who has a new genetic test that she claims can identify the species of our mother panda. But the test is imperfect:

- 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.

1. Ignore your previous information from the births and compute the posterior probability that your panda is species A.
2. Redo your calculation, now using the birth data from the two births as well.

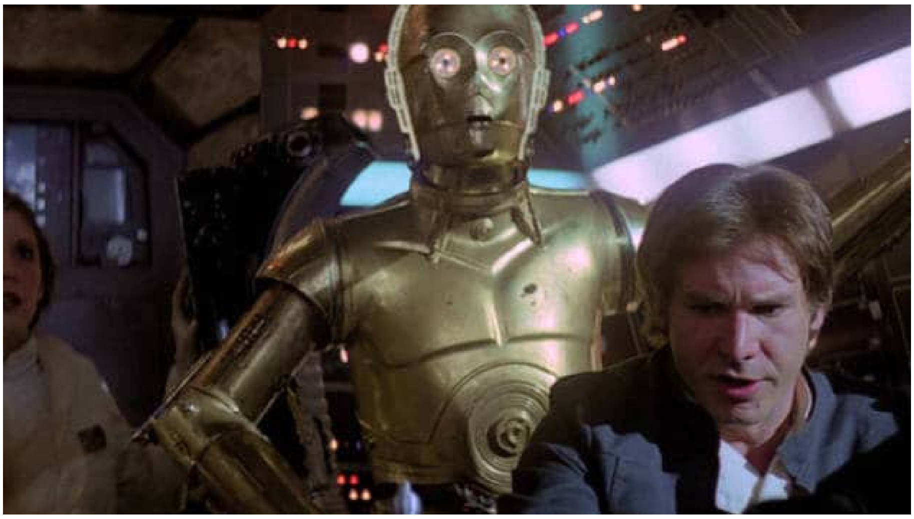
1.5 Beta priors and Bayesian inference

Read/watch first

- *Bayesian statistics the fun way* (Will Kurt), Chapter 9.
- *Statistical rethinking* (Richard McElreath), Sections 2.2, 2.3.

Let's go through the C-3PO example.

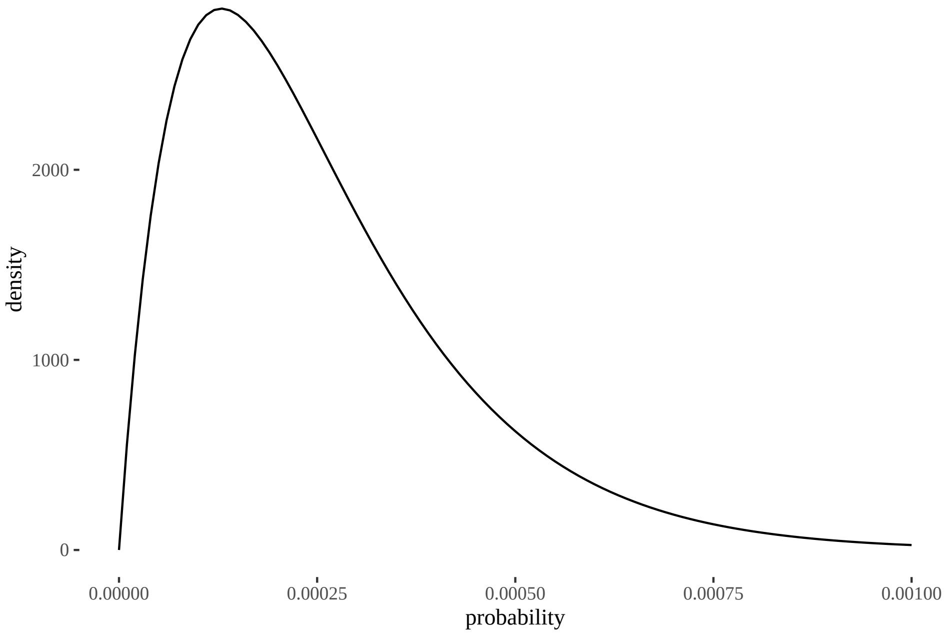
> When Han Solo, attempting to evade enemy fighters, flies the Millennium Falcon into an asteroid field, the ever-knowledgeable C-3PO informs Han that probability isn't on his side. C-3PO says, "Sir, the possibility of successfully navigating an asteroid field is approximately 3,720 to 1!" "Never tell me the odds!" replies Han.



Never tell me the odds!

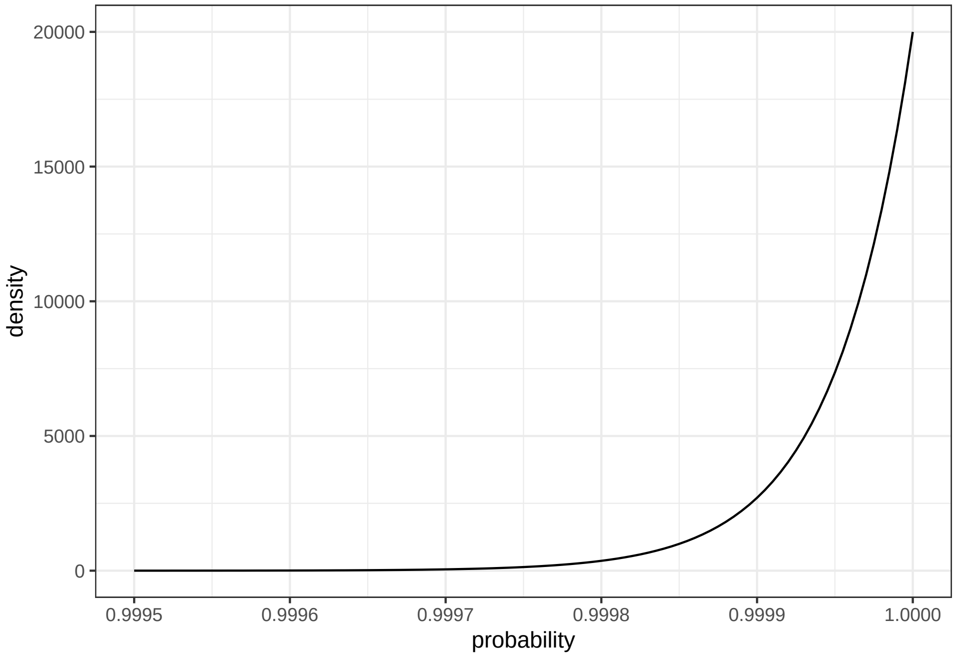
So how is C-3PO to update his pessimistic priors with the information that Han is a badass? In our terminology, C-3PO prior takes the odds of succeeding to be 1/3720. This isn't enough to uniquely determine the beta function, because it's unclear how much data C-3PO had to analyze. Was it information about 3721 attempts, only one of which succeeded, or was it, say, information of 7442 attempts, two of which succeeded, or…? Let's follow the author supposing it was the latter. So here's the distribution. Note this time we do not use grid approximation, but instead, set up a dummy plot p and then add to it the plot of a pre-defined function directly.

```
p <- ggplot(data = data.frame(probability = seq(0:0.003)),
            mapping = aes(x = probability))

c3po <- function(p) dbeta(p,2,7740) #define density as a function of
#probability only.

p + #basic layer
  stat_function(fun=c3po)+ #add the function
  xlim(c(0,0.001))+ #limit the x axis
  ylab("density")+theme_tufte()
#  theme_tufte(): I changed the theme to show you some possibilities
```



Now, let’s visualise our prior in Han’s badassery, the odds of his survival being 20000/1.

```
p <- ggplot(data = data.frame(probability = seq(0:1)),
            mapping = aes(x = probability))

han <- function(p) dbeta(p,20000,1)

p +
    xlim(c(0.9995,1))+
  stat_function(fun=han)+
  ylab("density")+theme_bw()
```



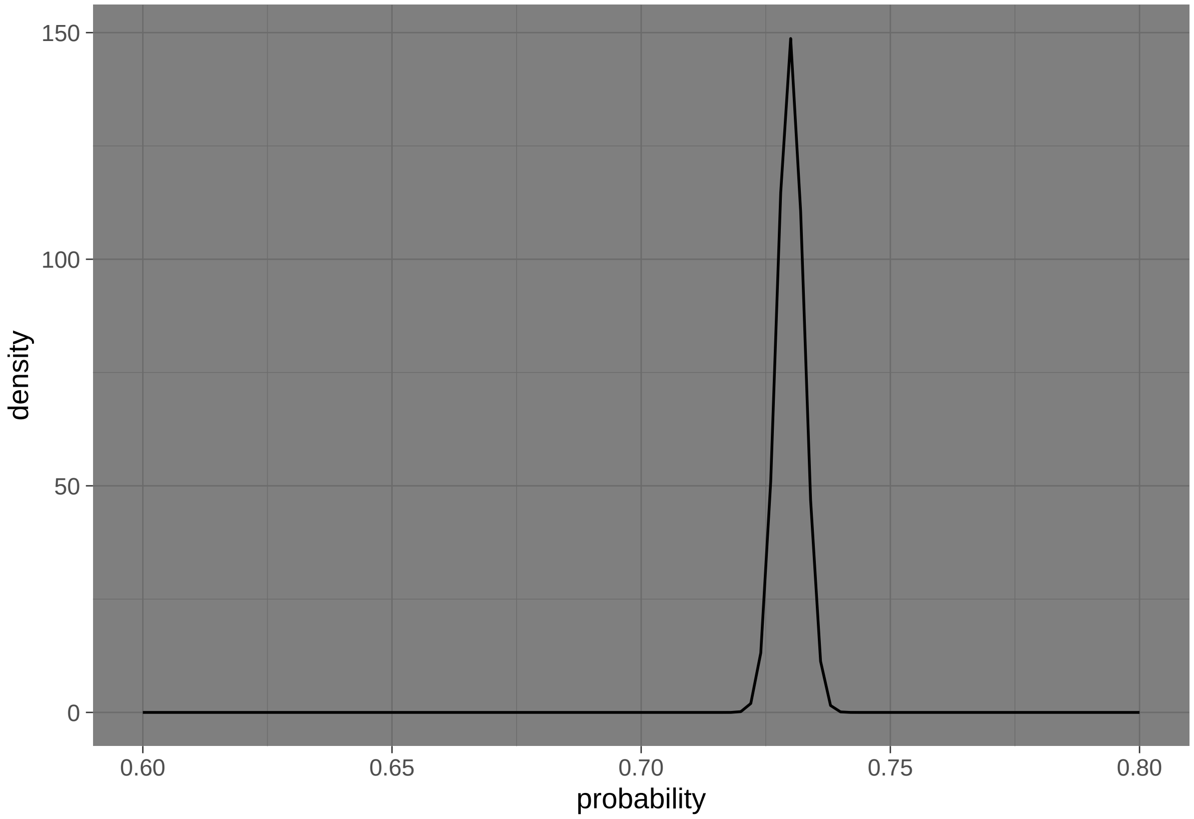
Now here’s the key move in building the posterior:

$$Beta(a_{\text{posterior}}, b_{\text{posterior}}) = Beta(a_{\text{likelihood}} + a_{\text{prior}}, b_{\text{likelihood}} + b_{\text{prior}}) \tag{18}$$

In our case, $a_{\text{posterior}} = 20002$ and $b_{\text{posterior}} = 7741$. Let’s visualise this posterior beta and find the center of this distribution.

```
p <- ggplot(data = data.frame(probability = seq(0:1)),
            mapping = aes(x = probability))

posterior <- function(p) dbeta(p,20002,7401)

p +
    xlim(c(.6,.8))+
  stat_function(fun=posterior)+
  ylab("density")+theme_dark()

ps <- seq(0,1,by = 0.0001)
dBETA(ps,20002,7401)[-1]
```

```
## $mean
## [1] 0.7299201
##
## $var
## [1] 7.193722e-06
```



Exercise 15

Modify this scenario a bit. What would your posterior look like and where would it center be if (i) C-3PO only knew of 450 attempts with 2 successes? (ii) Additionally, your belief in Han's badassery had odds 2000/1? You might need to change the x limits to make the plots sensible. Add plot titles with ggtitle().

For a slightly different perspective, let's walk through the water example from *Statistical rethinking*, Section 2.2. Let's recall the story:

> Suppose you have a globe representing our planet, the Earth. This version of the world is small enough to hold in your hands. You are curious how much of the surface is covered in water. You adopt the following strategy: You will toss the globe up in the air. When you catch it, you will record whether or not the surface under your right index finger is water or land. Then you toss the globe up in the air again and repeat the procedure.

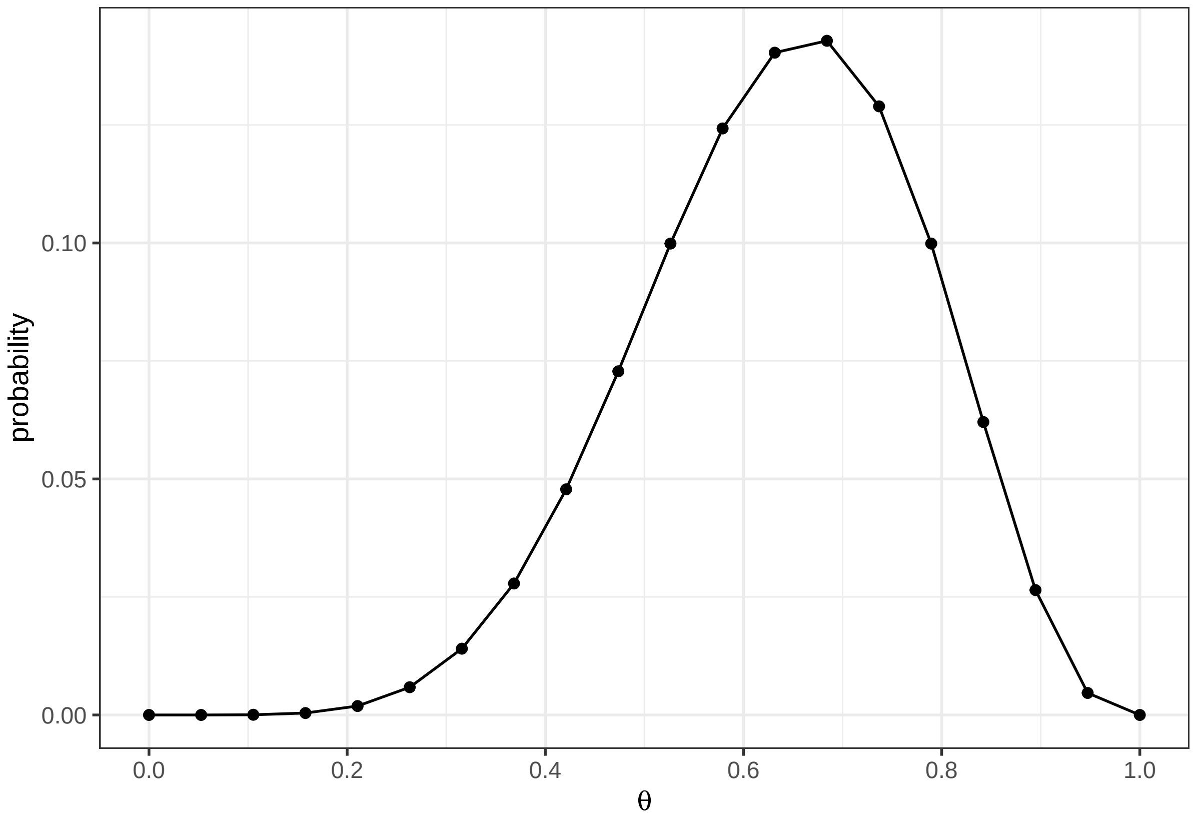
If the actual coverage is 0.5, you toss the globe 9 times, what's the probability of seeing six W's can be calculated using the binomial distribution:

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

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

Here's how we can use grid approximation to compute the posterior distribution over possible parameters, this should be already familiar. We then plot the approximation as a line, but emphasizing that we calculated this only in 20 points by putting points in the visualization as well.

```
# define grid
ps <- seq(from=0 , to=1 , length.out=20)
# define prior
#(note it's not really a probability, we'll standardize later)
prior <- rep(1, 20)
# compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob = ps )
# compute product of likelihood and prior (unstandardized prior)
unPosterior <- likelihood * prior
# standardize the posterior, so it sums to 1
posterior <- unPosterior / sum(unPosterior)

#put together
table <- data.frame(theta = ps,posterior)

#plot using ggplot
ggplot(table, aes(x = theta, y = posterior)) +
  geom_line() + geom_point()+
  scale_x_continuous(expression(theta), breaks = seq(from = 0, to = 1, by = .2)) +
  ylab("probability") +  theme_bw()
```



Exercise 16

Work with the globe example. Calculate the posterior distribution for a uniform prior, for three sets of observations:

- (A) W, W, W
- (B) W, W, W, L
- (C) W, L, W, L, W, W

Please:

- Approximate for 1000 points.
- Don't use points, just lines.
- Use ggarange (google) to put three plots in one row.
- Use ggtitle to add appropriate titles.
- Use tufte theme.

Exercise 17

Calculate the posterior distribution using grid approximation for the same nine observations for a potential prior that assigns 0.3 to all values below 0.6 and 1 to all the other values. Use "ifelse" (you can google its use). Approximate for 50 points.

Now, let's switch back to thinking in terms of functions rather than grid approximations. What if your prior is approximated by, say a uniform distribution and we want to run our estimation of the posterior using a fairly transparent formula notation in a way that easily generalizable to functions that are hard to work with analytically? Here's a solution that uses the rethinking package. The cost is, however, it approximates the posterior using the normal distribution.

```
globe <- map(
alist(
w ~ dbinom(9,p) , # binomial likelihood
p ~ dunif(0,1) # uniform prior
) ,
data=list(w=6) )
# display summary of quadratic approximation

# columns: mean / sd / 5.5% / 94.5%
precis(globe) %>%
  kable("latex", booktabs = T, linesep = "") %>%
kable_styling(latex_options = c("HOLD_position", "striped"),font_size = 9)
```

x	x	x	x
0.6666666	0.1571338	0.4155365	0.9177968

The interpretation: assuming the posterior is normal, it is maximized at 0.67, and its standard deviation is 0.16. Normal approximations for such cases get better with sample size (see Section 2.4.2 of *Statistical Rethinking* for a bit more details).

1.6 Sampling from the posterior and intervals

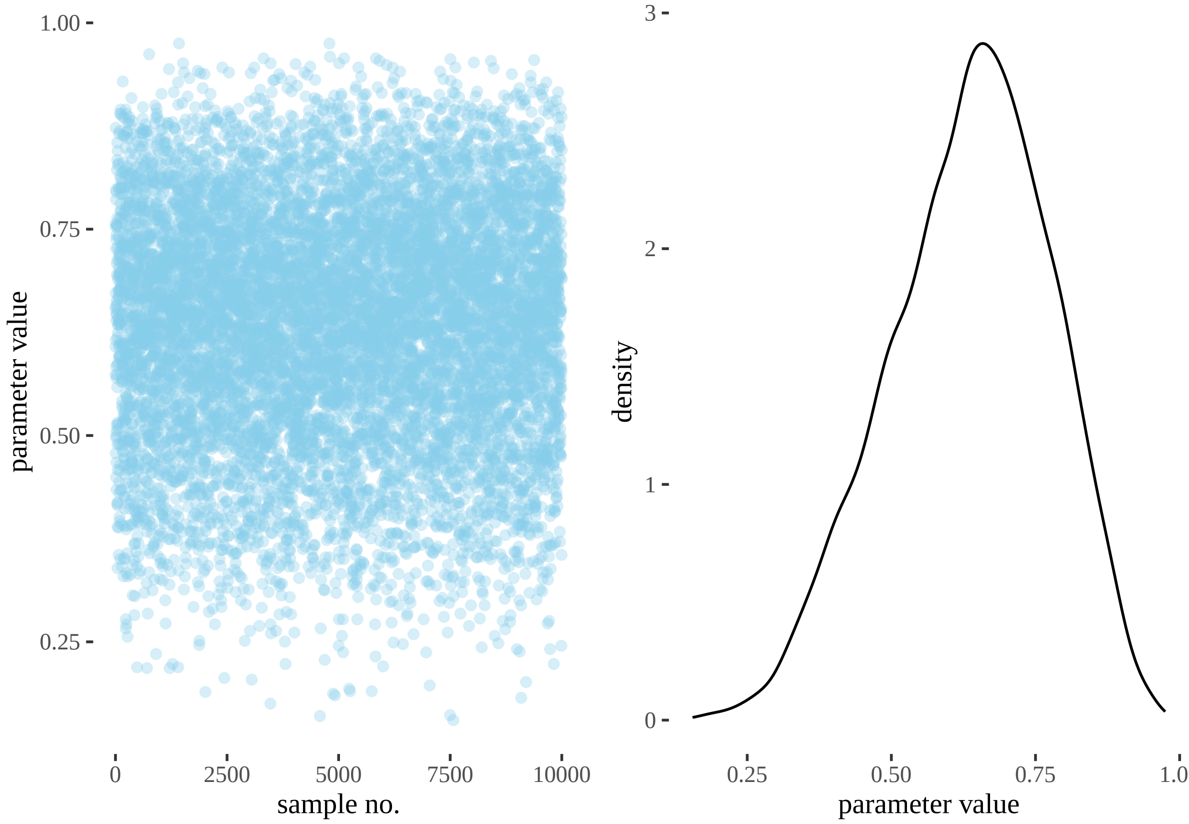
Read/watch first

- *Statistical rethinking* (Richard McElreath), Sections 3.1, 3.2.
- *Doing Bayesian Data Analysis*, John Kruschke, chapters 2, 3, 4.

It's a bit of overkill do to posterior sampling for Bernoulli trials, but our point we'll be conceptual. Keep in mind that simulations are *very* handy when it comes to more complicated distributions, combining models, and more complicated predictions.

First, let's play around with a grid approximation, create and plot a sample of posterior probabilities. We'll use the example involving nine tosses of a globe. Note each parameter has the probability of being drawn obtained by the bayesian calculation of the posteriors.

```
set.seed(666)
ps <- seq(from=0 , to=1 , length.out=1000)
prior <- rep(1 , 1000)
likelihood <- dbinom( 6 , size=9 , prob=ps)
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)

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

points <- ggplot()+geom_point(aes(x = 1:1e4, y=samples),
                              col = "skyblue", alpha = 0.35)+
          theme_tufte()+xlab("sample no.")+ylab("parameter value")

density <- ggplot()+geom_density(aes(x=samples))+theme_tufte()+
            xlab("parameter value")

ggarrange(points, density, ncol = 2)
```



Such samples are useful for various approximations. For instance, we might estimate the probability that the proportion of water is less than 0.6 given our nine observations, six of which were that of water. Make sure you understand what the brackets do. First, let's do this analytically: basically we sum the posterior probabilities for parameter values less than 0.6.

```
sum(posterior[ ps < 0.6 ])
```

```
## [1] 0.3825314
```

Now, we approximate the answer by checking what proportion of parameters in our sample in fact is less than 0.6.

```
sum(samples < 0.6) / 1e4
```

```
## [1] 0.3773
```

Pretty close, right?

Exercise 18

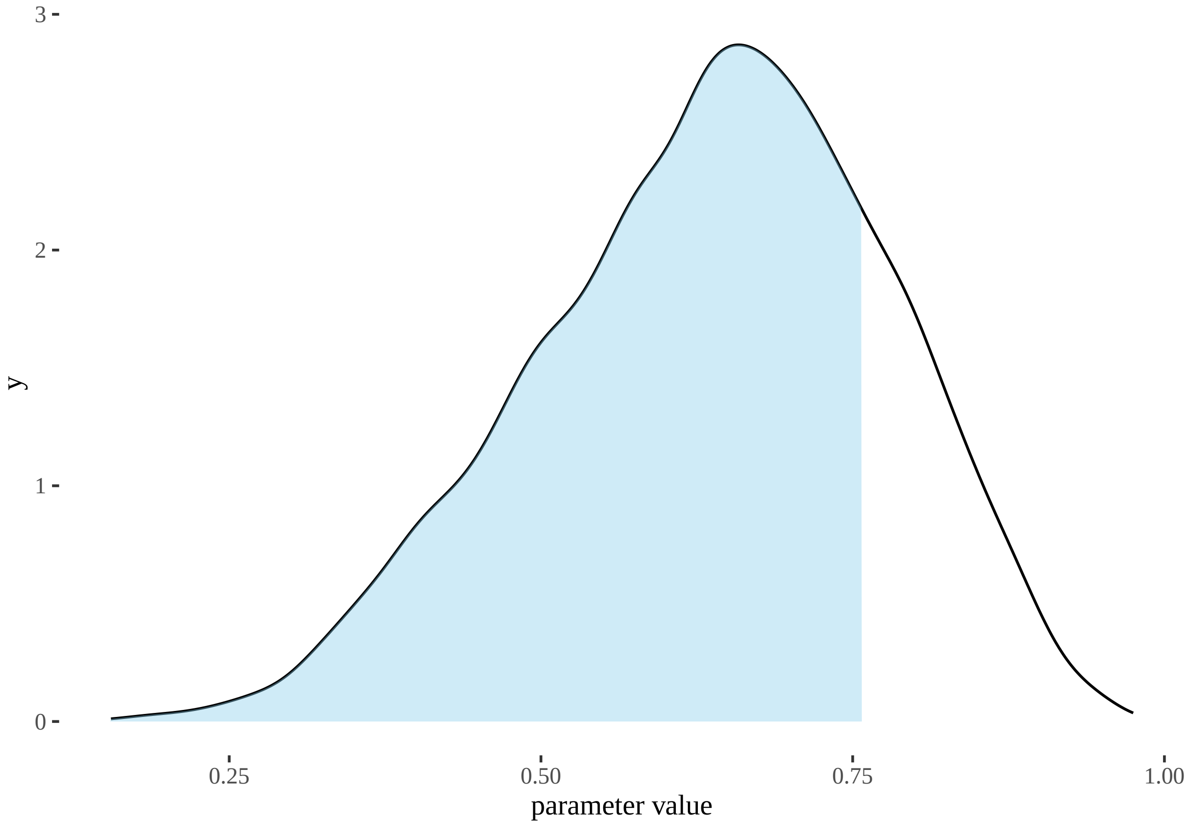
In the same setup, first calculate analytically and then approximate the probability that the proportion of water is strictly between .4 and .8.

Now we can think about intervals. An interval of posterior probability is called *posterior interval*. For instance, you might want to know the parameter value such that 70% of the posterior is below it. Note that 70% is the proportion of the area under the density curve that is below .8.

```
quantile(samples, 0.8)
```

```
##       80%
## 0.7587588
```

```
#extract numeric account from the plot and use to shade the interval
d <- ggplot_build(density)$data[[1]]
density+geom_area(data = subset(d, x < 0.758),
        aes(x=x, y = y), fill="skyblue", alpha = 0.4)
```



Exercise 19

Calculate and plot the middle 50% interval for the same density.

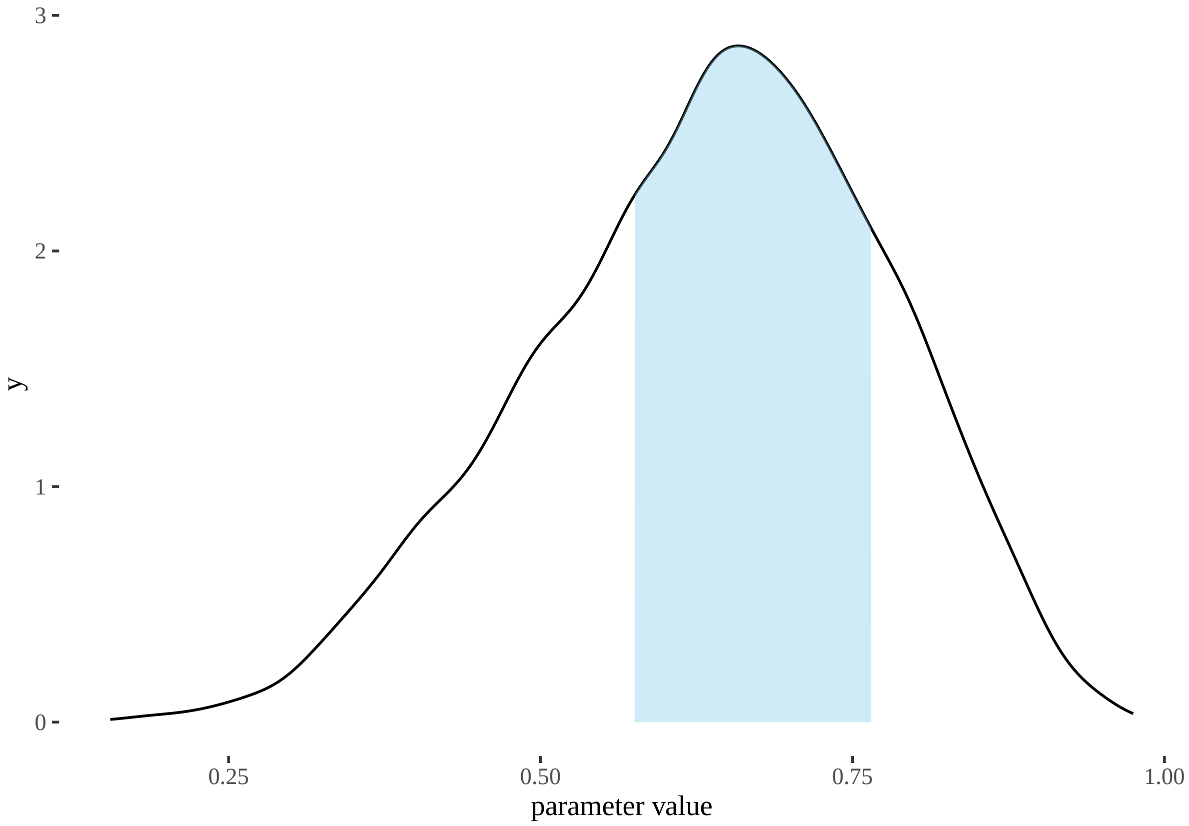
These are *percentile intervals* they sort of make sense when you're summarizing data for fairly symmetric distribution.

Percentile intervals are contrasted with the *highest posterior density intervals (HPDI)*. The HPDI is the narrowest interval containing the specified probability mass of the posterior distribution. If you think about it, there must be an infinite number of posterior intervals with the same mass. But if you want an interval that best represents the parameter values most consistent with the data, then you want the densest of these intervals. It's only a bit different in this particular case:

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

```
##      |0.5      0.5|
## 0.5745746 0.7657658
```

```
density+geom_area(data = subset(d, x > 0.574 & x < 0.765),
        aes(x=x, y = y), fill="skyblue", alpha = 0.4)
```



However, they might be quite different if the distribution is more assymetric.

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


density <- ggplot()+geom_density(aes(x=samples))+theme_tufte()+
            xlab("parameter value")

quantile(samples, c(0.25,0.75))
```

```
##       25%       75%
## 0.7047047 0.9309309
```

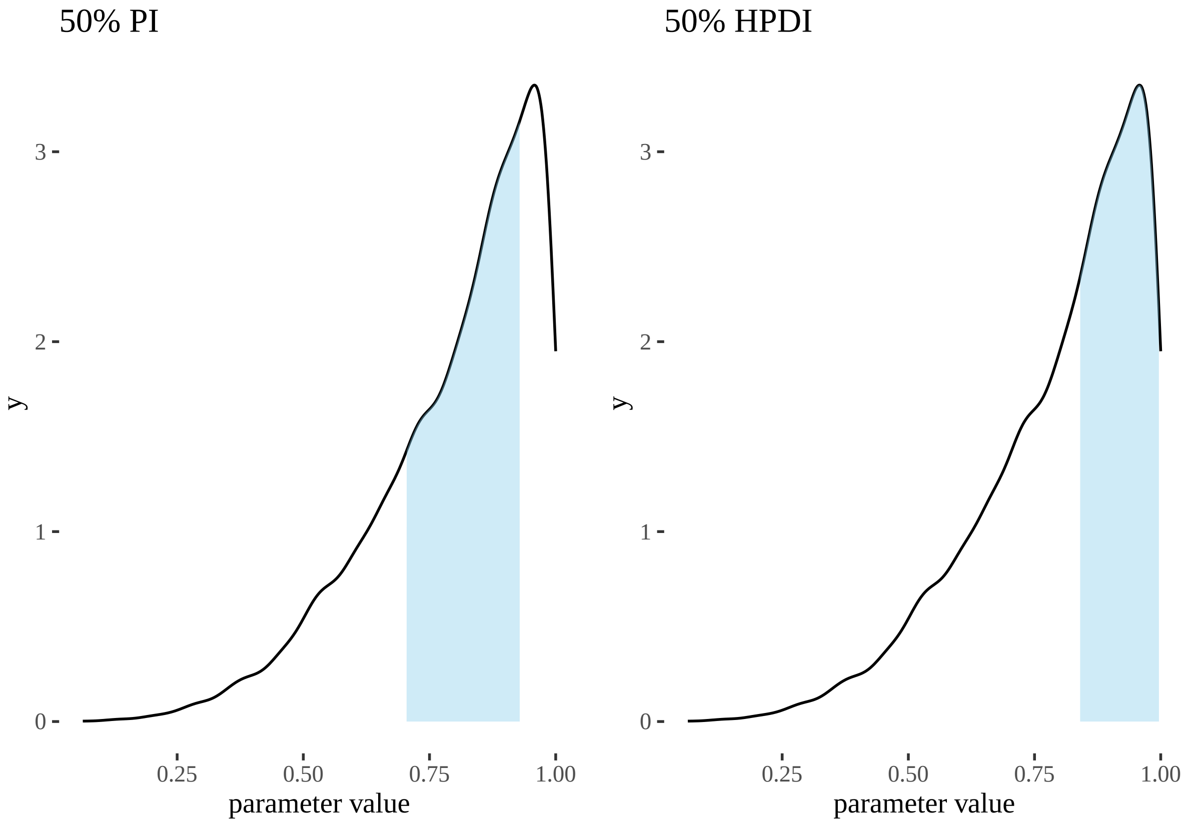
```
#In fact, you can use PI and HDPI from rethinking

PI(samples, prob = 0.5)
```

```
##       25%       75%
## 0.7047047 0.9309309
```

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

```
##      |0.5      0.5|
## 0.8408408 0.9989990
```

```
d <- ggplot_build(density)$data[[1]]
density50PI <- density + geom_area(data = subset(d, x > 0.704 & x < 0.93),
        aes(x=x, y = y), fill="skyblue", alpha = 0.4)+ggtitle("50% PI")

density50HPDI <- density + geom_area(data = subset(d, x > 0.840 & x < 0.998),
        aes(x=x, y = y), fill="skyblue", alpha = 0.4)+ggtitle("50% HPDI")

ggarrange(density50PI, density50HPDI, ncol=2)
```

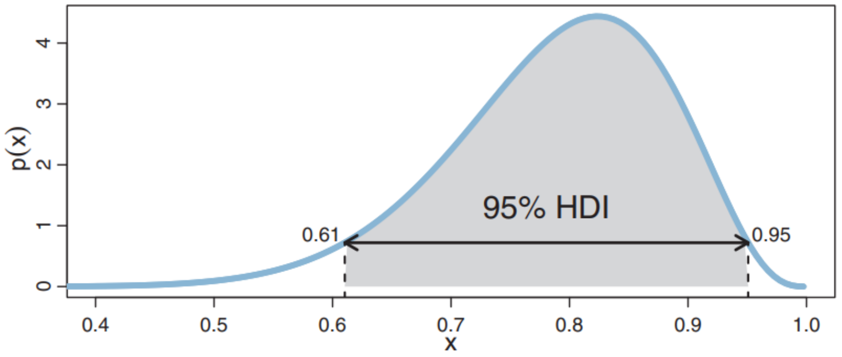


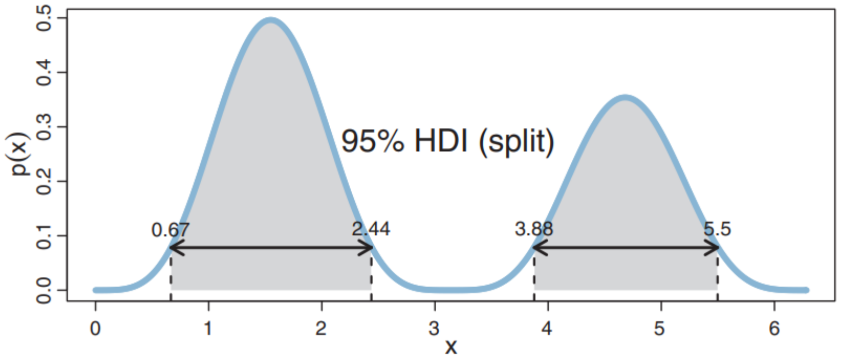
Exercise 20

Start with the same prior, but now imagine you observed water 10 out of 11 times. Sample from the posterior, calculate 80

The key point: HDPI seems better at capturing the center of the mass of a distribution. Although note that if the type of the interval makes a huge difference, you should report both and indicate the difference.

In fact, it makes sense to talk about HDI of other distributions as well. Here is another way to conceptualize this. *Highest density interval* is an interval (or intervals) that spans most (say, 95%) of the distribution such that every point inside the interval(s) have higher credibility than any point outside of it. Here's a rather uneven example from Kruschke's book.





Let's learn another way to find and plot the HDI of a given vector. We'll work with a longitudinal set connecting heights to earnings coming from the **modelr** package.

```
?heights   # find the information about the dataset
```

```
## uruchamianie serwera httpd dla pomocy ... wykonano
```

```
heights <- as.data.frame(heights)    # convert to a standarda data frame format
          # inspect the first few lines to get a feel for the dataset,
          # note the pipe operator (google it and learn to use it)
head(heights) %>% kable("latex", booktabs = T, linesep = "") %>%
          kable_styling(latex_options = c("HOLD_position", "striped"),font_size = 9)
```

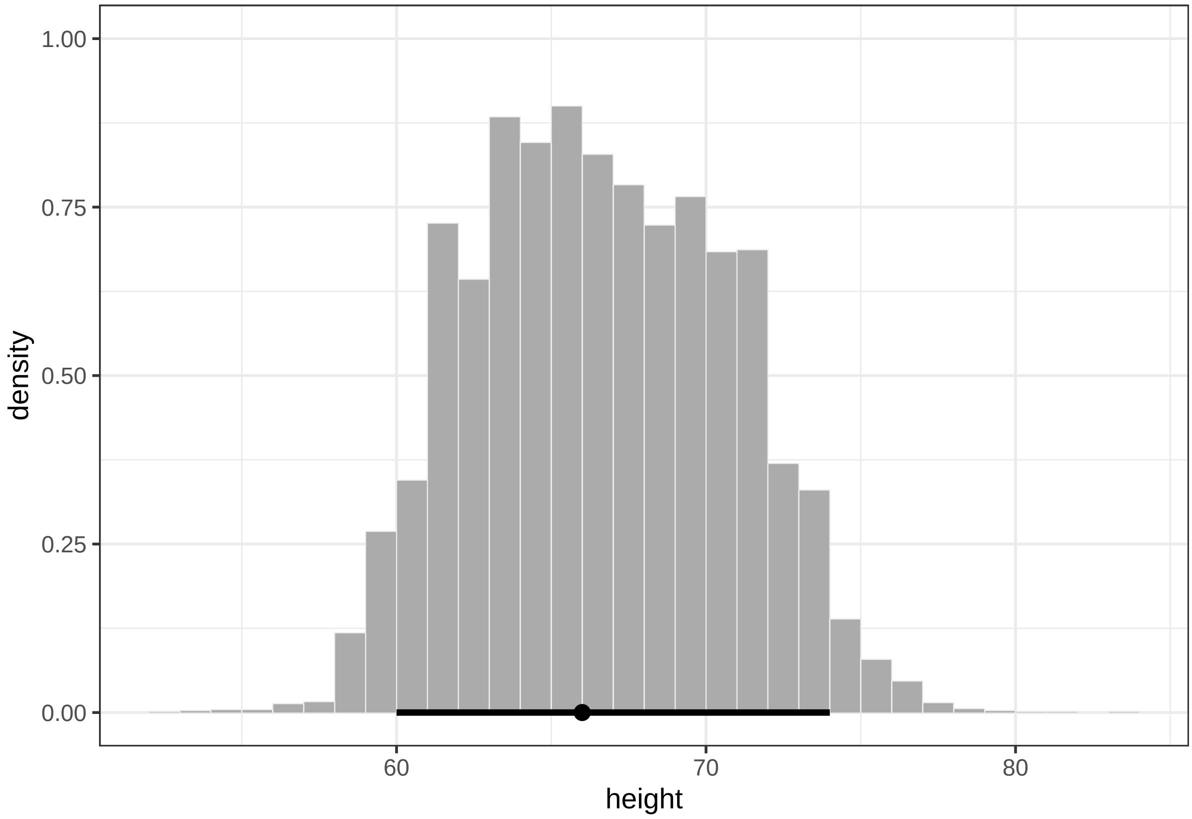
income	height	weight	age	marital	sex	education	afqt
19000	60	155	53	married	female	13	6.841
35000	70	156	51	married	female	10	49.444
105000	65	195	52	married	male	16	99.393
40000	63	197	54	married	female	14	44.022
75000	66	190	49	married	male	14	59.683
102000	68	200	49	divorced	female	18	98.798

```
         # now, let's find the 95% HDI of the heights
         # note I used kable to plot a nice table, you don't have to do this
mode_hdi(heights$height, .width = 0.95)  %>%
          kable("latex", booktabs = T, linesep = "") %>%
          kable_styling(latex_options = c("striped", "HOLD_position"),font_size = 9)
```

y	ymin	ymax	.width	.point	.interval
66	60	74	0.95	mode	hdi

```
         #now, plot the histogram with HDI
heights %>%
  ggplot(aes(x = height, y = 0))+
  stat_histinterval(point_interval = mode_hdi, .width = .95,
                    fill = "grey67", slab_color = "grey92",
                    breaks = 40, slab_size = .2, outline_bars = T)+
                   ylab("density")+theme_bw()
```

```
## Warning: Using the `size` aesthietic with geom_segment was deprecated in ggplot2 3.4.0.
## i Please use the `linewidth` aesthetic instead.
```



Exercise 21

Calculate and visualize the 90

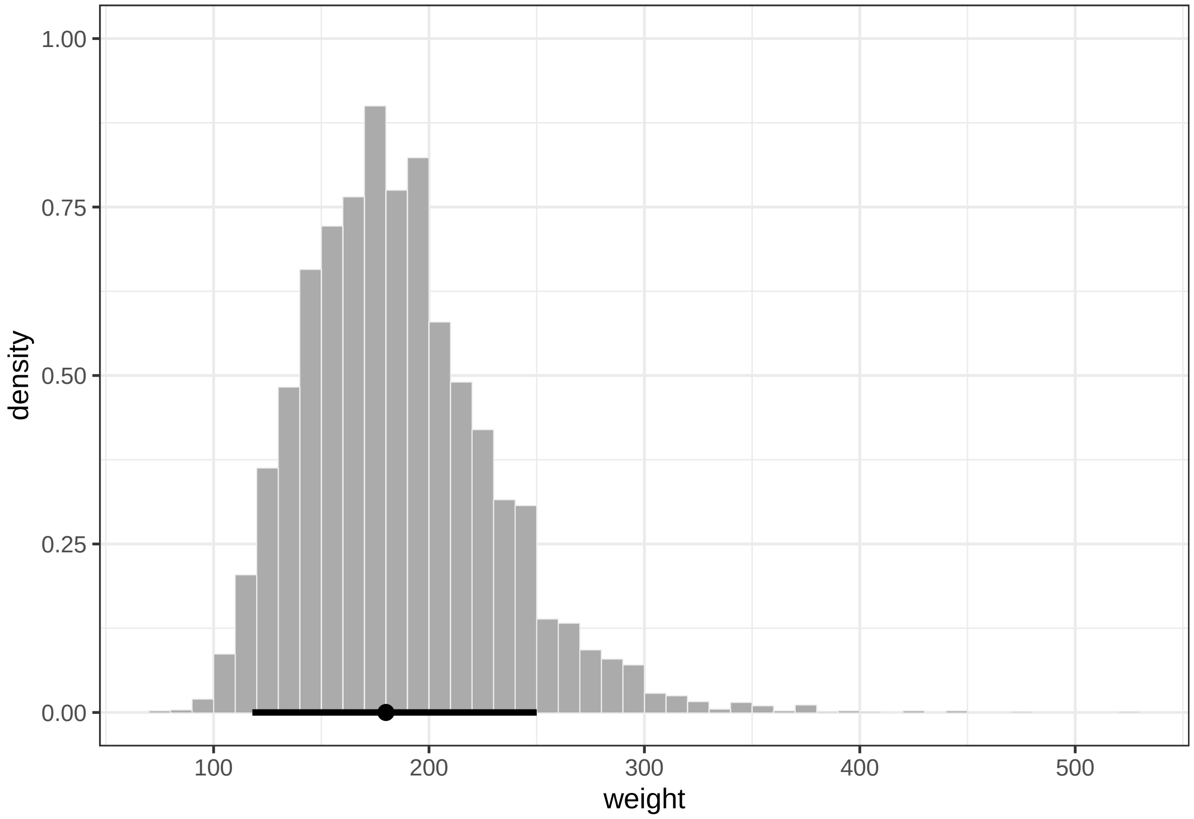
```
sum(is.na(heights$weight)) # check how many you're removing
```

```
## [1] 95
```

```
weights <- heights[!is.na(heights$weight),]
```

And your result should look like this:

y	ymin	ymax	.width	.point	.interval
180	118	250	0.9	mode	hdi



It will be useful to have a function that calculates HDI for a given distribution. For technical reasons, it's easier to do so for the quantile function (cumulative inverse probability). The function goes like this. You don't need to understand it, but if you're curious feel free to go over it:

```
hdi_of_icdf <- function(name, width = .95, tol = 1e-8, ... ) {
  # Arguments:
  #   `name` is R's name for the inverse cumulative density function
  #   of the distribution.
  #   `width` is the desired mass of the HDI region.
  #   `tol` is passed to R's optimize function.
  # Return value:
  #   Highest density iterval (HDI) limits in a vector.
  # Example of use: For determining HDI of a beta(30, 12) distribution, type
  #   `hdi_of_icdf(qbeta, shape1 = 30, shape2 = 12)`
  #   Notice that the parameters of the `name` must be explicitly stated;
  #   e.g., `hdi_of_icdf(qbeta, 30, 12)` does not work.
  # Adapted and corrected from Greg Snow's TeachingDemos package.
  incredible_mass <-  1.0 - width
  interval_width  <- function(low_tail_prob, name, width, ...) {
    name(width + low_tail_prob, ...) - name(low_tail_prob, ...)
  }

  opt_info            <- optimize(interval_width, c(0, incredible_mass),
                                  name = name, width = width,
                                  tol = tol, ...)

  hdi_lower_tail_prob <- opt_info$minimum

  return(c(name(hdi_lower_tail_prob, ...),
           name(width + hdi_lower_tail_prob, ...)))

}
```

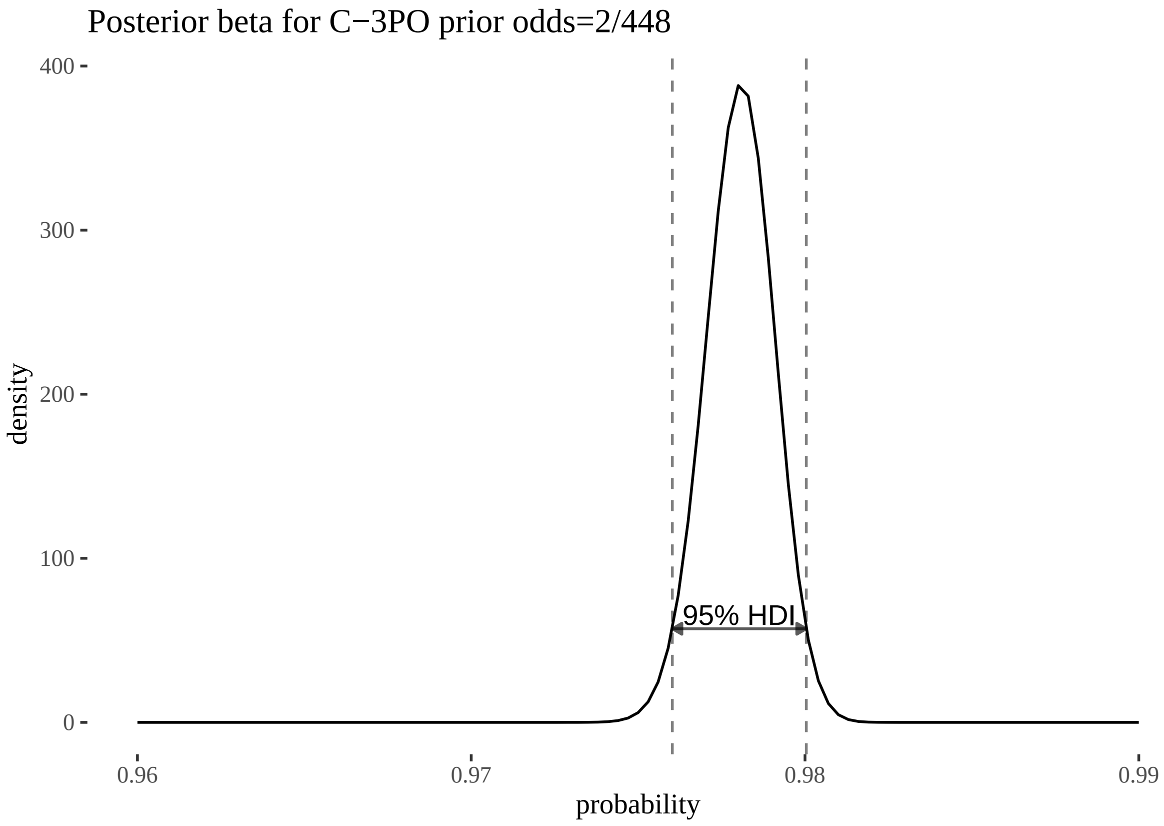
With this function ready, we can go back to our Star Wars example and calculate and plot the HDIs. Let's do this for one of the posteriors.

```
#this was our posterior
posteriorI <- function(p) dbeta(p,20002,449)
```

```
#note shape parameters are just a and b, and
#that we used qbeta for the quantile function
hdiI <- hdi_of_icdf(qbeta, shape1 = 20002, shape2 = 449, width = .95)

#print the HDI limits:
hdiI
```

```
## [1] 0.9760250 0.9800389
```

```
#now the plot
#dummy background
p <- ggplot(data = data.frame(probability = seq(0:0.003)),
            mapping = aes(x = probability))

#layers
p + xlim(c(0.96,.99)) +
  stat_function(fun=posteriorI)+
    ylab("density")+theme_bw()+
  ggtitle("Posterior beta for C-3PO prior odds=2/448")+
  #so far nothing new
  geom_vline(aes(xintercept = hdiI),
             lty = 2, alpha = 0.5) + #add dashed lines
  geom_segment(aes(x = hdiI[1], xend = hdiI[2], y = posteriorI(hdiI[1]),
                   yend = posteriorI(hdiI[1])),
            arrow = arrow(length = unit(.15, "cm"),
                          ends = "both",
                          type = "closed"), alpha = 0.4)+ #arrow
   annotate("text", x = mean(hdiI), y = posteriorI(hdiI[1])*1.15,
            label = "95% HDI") +theme_tufte()  #text
```



Exercise 22

Calculate and plot 90% HDI for posteriorII as defined in the Star Wars example. Remember to change the annotation to "90%", move the annotation if need be.

```
posteriorII <- function(p) dbeta(p,2002,449) #recall
```

1.7 Estimation and prediction

Let's work with our example of nine tosses of a globe.

```
ps <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep(1,1000)
likelihood <- dbinom( 6 , size=9 , prob=ps )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
samples <- sample(ps, size=1e4 , replace=TRUE , prob=posterior)
```

Suppose we are asked for a point estimate of the water coverage. One thing we could do is to report the parameter with the highest posterior probability, *maximum a posteriori estimate*, MAP. If we have the posterior probabilities calculated analytically, we can do this analytically.

```
ps[which.max(posterior)]
```

```
## [1] 0.6666667
```

If, however, we want to rely only on a sample from the posterior distribution, we can use chainmode to approximate this value.

```
chainmode(samples)
```

```
## [1] 0.6714762
```

You might be tempted to report mean or median of the samples:

```
mean(samples)
```

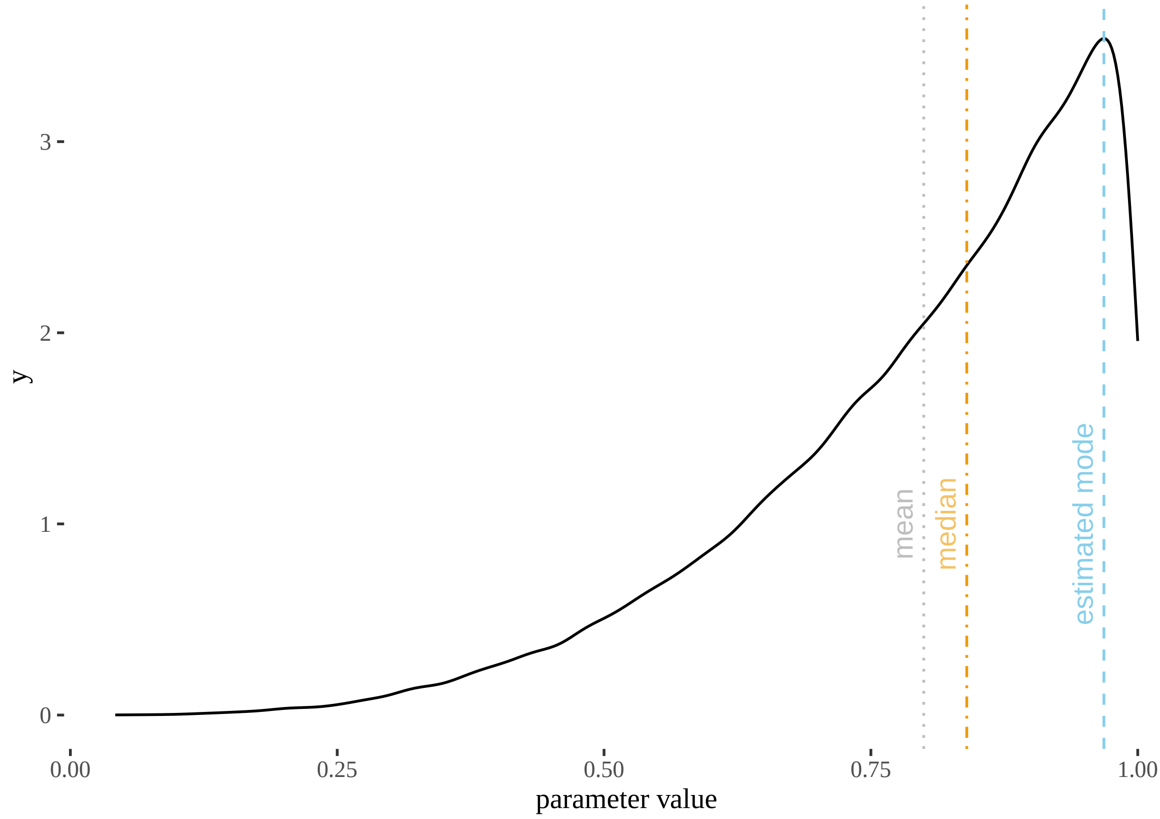
```
## [1] 0.6366261
```

```
median(samples)
```

```
## [1] 0.6466466
```

In this case there isn't too much of a difference. But let's look at an assymetric example.

```
ps <- seq(from=0 , to=1 , length.out=1000)
prior <- rep(1,1000)
likelihood <- dbinom(3 , size=3 , prob=ps) #3 out of 3 times water!
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
samples <- sample(ps, size=1e5 , replace=TRUE , prob=posterior)

density <- ggplot()+geom_density(aes(x=samples))+theme_tufte()+
            xlab("parameter value")


density <- ggplot()+geom_density(aes(x=samples))+theme_tufte()+
  xlab("parameter value")


density+
  geom_vline(xintercept = chainmode(samples), lty = 2, color = "skyblue")+
  geom_text(aes(x=chainmode(samples)-0.02, y=1), label="estimated mode",
            color="skyblue", angle=90)+
  geom_vline(xintercept = mean(samples), lty = 3,
             color = "grey")+
  geom_text(aes(x=mean(samples)-0.02, y=1), label="mean", color="grey", angle=90)+
  geom_vline(xintercept = median(samples), lty = 4, color = "orange2")+
  geom_text(aes(x=median(samples)-0.02, y=1), label="median", color="orange2",
            angle=90, alpha = 0.6)
```



Also, the choice of the estimation method depends on how you're penalized for being wrong! Give me your estimate e of the proportion of water. Say the real proportion is p. One *loss* function we could use is that the penalty is proportional to how far e and p are, that is to the absolute value of $e - p$.

Now here's the trick. When you estimate using this loss function, you use your posterior to figure out the parameter value that you expect to minimize this distance. Let's calculate this *expected loss*. The idea is, for each particular possible parameter value, we measure its absolute distance from our estimate. Then we take a weighted average of these distances, where for each parameter, its corresponding weight is our posterior probability of that parameter. For instance, if our estimate is .6, our expected loss is:

```
e <- .6  #estimate
sum(posterior*abs(e-ps))
```

```
## [1] 0.231442
```

But we want to minimize the loss, so we have to calculate the loss for all possible estimates that we could give, and then choose the estimate which results in a minimial loss. So, basically, we need to repeat the calculations for each potential estimate. One way to do this is to use "for". There is also a trick that uses "sapply", which is less intuitive, but results in a one-liner.

```
loss <- NULL #initiate empty list
for (i in 1:length(ps)){
  loss[i] <- sum(posterior*abs(ps[i]-ps))
}

#alternatively
loss <- sapply( ps, function(e) sum( posterior*abs( e - ps ) ) )

ps[which.min(loss)]
```

```
## [1] 0.8408408
```

```
median(samples) #this turns out to be pretty much the posterior median
```

```
## [1] 0.8398398
```

Importantly, different choices of the loss function lead to different estimates. If, for instance, we take the quadratic loss $(e-p)^2$, then it is the mean that minimizes the expected loss. Practical considerations might suggest using a different loss function (for instance, if you incorrectly predict a high probability of a hurricane might result in other losses than if you incorrectly predict a low probability thereof).

Exercise 23

Suppose the globe tossing data had turned out to be 8 water in 14 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as before. Use set.seed(666) first for replicability.

1. How much posterior probability lies below $p = 0.3$?
2. How much posterior probability lies above $p = 0.9$?
3. How much posterior probability lies between $p = 0.2$ and $p = 0.8$?
4. 20% of the posterior probability lies below which value of p?
5. 20% of the posterior probability lies above which value of p?
6. Which values of p contain the narrowest interval equal to 66% of the posterior probability?
7. Which values of p contain 66% of the posterior probability, assuming equal posterior probability both below and above the interval?

Bayesian models are generative: given a parameter the likelihood defines a distribution of possible observations that we can sample from, to simulate observation. For instance, if we toss the globe two times, there are three possible observations (0 water, 1 water, 2 water). Given a particular parameter assumed, say 0.6, we can calculate the probability of each of these possible outcomes:

```
dbinom(0:2,2,0.66)
```

```
## [1] 0.1156 0.4488 0.4356
```

Now, we can also simulate random data that is obtained by employing these probabilities. Say we want to simulate one test. That is: we “toss” the globe twice and “write down” the number of times we observed water.

```
rbinom(1, size = 2, prob = 0.66)
```

```
## [1] 2
```

We can also simulate repeating a series of experiment, each time tossing the globe twice and writing down the number of times we observed water. Say we do this ten times.

```
rbinom(10 , size=2 , prob=0.66)
```

```
##  [1] 2 1 1 1 0 1 2 2 2 1
```

Let’s generate *dummy data* by simulating the experiment 10000 times and look at the frequencies of the possible outcomes.

```
dummy <- rbinom( 1e5 , size=2 , prob=0.66)
table(dummy)/1e5
```

```
## dummy
##       0       1       2
## 0.11456 0.45211 0.43333
```

The key observation is that with a sufficiently large dummy set, the frequencies approximate the probabilities we obtained analytically.

Exercise 24

Simulate 1e5 experiments of nine tosses of the globe and plot them in a histogram.

Now comes the trick. Instead of using a single assumed parameter, as we did for .6, we can use combine our dummy dataset generation with our posterior distribution. This captures the idea that instead of knowing that the parameter is .6 we only have some probability distribution for the potential parameters. That is, we combine two sorts of uncertainty, *propagating* the parameter uncertainty as we come up with predictions. Instead of keeping the probability fixed, we take our posterior sample (recall, it was obtained for the observation of 6 water out of 9, with a flat prior), and for each of its elements, we run one experiment of nine observations and write down the number of times we observed water (note the number of experiments is the same size as our posterior sample). We also plot the histogram

```
w <- rbinom( 1e5 , size=9 , prob=samples)
```

Now, notice, there are three different distributions that we might be interested in. The distribution of the posterior samples, whose density we have already plotted, which captures our posterior uncertainty. Then, we might be interested in the distribution of potential outcomes assuming a particular point estimate (we just plotted a histogram using .6). Finally, we might be interested in the distribution of potential outcomes not assuming any particular point estimate, but using the complete information about our uncertainty about the parameter as captured by the posterior sample. Let's see a histogram:

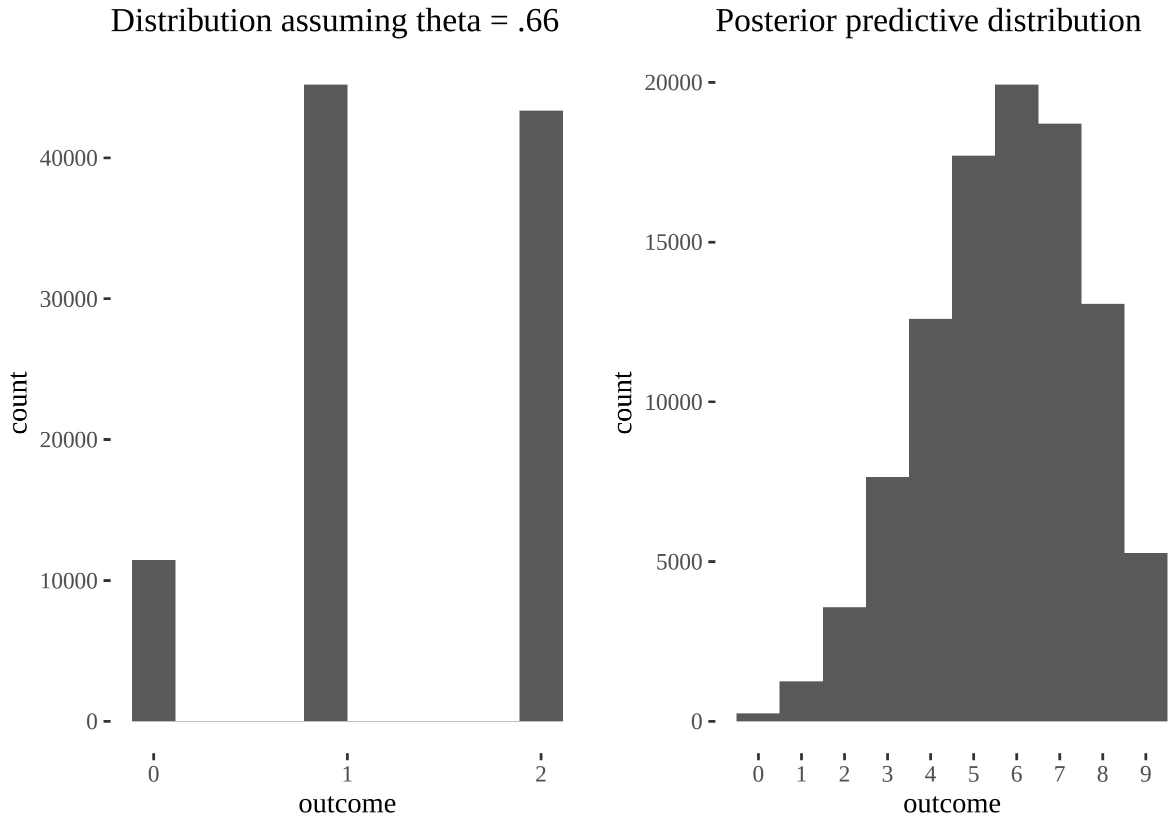
```
strict <- ggplot()+geom_histogram(aes(x=dummy), bins = 10)+
  scale_x_continuous(breaks = 0:9)+
  theme_tufte()+xlab("outcome")+
  ggtitle("Distribution assuming theta = .66")


ppd <- ggplot()+geom_histogram(aes(x=w), bins = 10)+
  scale_x_continuous(breaks = 0:9)+
  theme_tufte()+xlab("outcome")+
  ggtitle("Posterior predictive distribution")

mean(dummy)
```

```
## [1] 1.31877
```

```
mean(w)
```

```
## [1] 5.72879
```

```
ggarrange(strict, ppd, ncol = 2)
```



Note that the distributions are centered around pretty much the same mean, but they look quite different. The one on the left is more centered and doesn't incorporate our uncertainty about the parameter, and so might give us a false sense of certainty, whereas the one that does incorporate the higher-order uncertainty is less self-confident, but also more honest.

Exercise 25

Use our posterior predictive simulation to estimate the probability that the number of times you observe water in a new experiment is strictly between 2 and 7. Then use the dummy data obtained assuming the parameter value 0.66. Then calculate this probability analytically using pbinom.

Exercise 26

Let's start over. Suppose you know that a majority of the Earth's surface is water, but not more. That is, use a prior that is zero below p = 0.5 and a constant above p = 0.5. Suppose you observed water 6 out of 9 times. Use set.seed(666) for replicability. We'll investigate what difference the better prior makes, so run this along the original case of flat prior.

1. Start with creating posterior samples for both priors (call them different names).
2. Calculate 90% HPDI for both posterior samples.
3. Plot densities of both posterior samples next to each other. Use xlim to make sure the x axis ranges from 0 to 1 in both cases and ylim to make both y axes range between 0 and 4. Use the trick we already used to shade the HPDI areas that you already calculated.
4. Now construct the posterior predictive samples for both options. Plot them using histograms next to each other.
5. Use the PPD to calculate the probability of observing 6 in nine tosses, do this first for the flat prior, then for the informed one.

Exercise 27

Let's do the homework from *Statistical rethinking*. Load the data and take a look. These data indicate the gender (male=1, female=0) of officially reported first and second born children in 100 two-child families.

How to read this? The first family in the data reported a boy (1) and then a girl (0). The second family reported a girl (0) and then a boy (1). The third family reported two girls.

1. Calculate the posterior for a flat prior. Use lines to plot both the posterior and the likelihood on the same plot (use color and different line type for the likelihood). Set seed to 666. Which p maximizes the likelihood? Which p maximizes the posterior?
2. Use the sample function to draw 10,0000 random parameter values from the posterior distribution you calculated above. Use these samples to approximate the 50%, 89%, and 97% highest posterior density intervals. Plot the density posterior sample with shaded 89% HPDI.

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

HPDI(samples, .5)
HPDI(samples, .89)
HPDI(samples, .97)

density <- ggplot()+geom_density(aes(x = samples))+theme_tufte()



d <- ggplot_build(density)$data[[1]]
density+geom_area(data = subset(d, x < 0.6 & x > .505),
        aes(x=x, y = y), fill="skyblue", alpha = 0.4)
```

3. Use rbinom to draw a random sample of 10000 observations of births out of 200, using the optimizing parameter that you have found. Plot the density of this set adding a vline at 111. Does the model fit the data nicely? That is, is the actual observation close to the middle of what the model predicts?
4. Now use the same method to compare 10,000 counts of boys from 100 simulated first borns only to the number of boys in the first births. Use the same optimized parameter. How does the model look in this light?
5. 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. First you need to select these births using standard R methods. For instance, if I want to randomly sample 100 natural numbers from the interval 1:10 twice and then select those from the second sample for which the corresponding ones in the first sample were less than 3 I do this:

```
s1 <- sample(1:10,size = 100, replace = TRUE)
s2 <- sample(1:10,size = 100, replace = TRUE)
selected <- s2[s1<3]
selected
```

```
##  [1]  7  8  5 10 10  7  7  7  5  2  5
```

Now, use the density plot with a vline to compare 10,000 simulated counts of boys to the number of boys born in only those second births that followed girls. How does the model look in this light? Any guesses what is going on?
