Small Worlds and Large Worlds

Chapter 2

Daniela Palleschi

```
# install if needed
# install.packages("pacman")
# devtools::install_github("rmcelreath/rethinking")
```

Terms and Concepts

term	definition
conjecture	possible outcomes (like the 5 possible proportions of blue/white marbles in the ba
plausibility	things that can happen more than one way are plausible; we want to find out whic
likelihood (general)	relative number of ways that a value *p* can produce the data; derived by enumerat
likelihood (Bayesian)	distributions of variables; the observed data
prior probability	prior plausibility of any specific *p*
prior	distribution of prior plausibility
posterior probability	new, updated plausibility of any specific *p* given priors + data
Bayesian updating	updating prior plausibilties in light of the data to produce posterior plausibilties
variables	symbols that can tkae on a different value; e.g., the coutns of water and land in th
parameters	unobserved variables; e.g., the proportion of water on the globe (in the land/water
maximal entropy	the distrubtion contains no additional information other then: tehre are 2 events,
Bayes' theorem	$$Pr(p\|W,L) = \frac{Prob\ of\ data\ x\ Prior}{Average\ Prob\ of\ data}$$
Grid approximation	compute the posterior probability for any and every particular value of a paramte
quadratic approximation	a Guassian approximation is quadratic approximation because the log of the Guas

Code from chapter

Plausibilties

How probable is each possible combination of white/blue marbles? (divide each # of ways each combination could produce the observed data by sum of all ways. Ways: all white = 0 ways, 1 blue = 3, 2 blue = 8, etc.)

```
# R code 2.1: compute plausibilities
ways <- c(0,3,8,9,0)
ways/sum(ways)
```

```
[1] 0.00 0.15 0.40 0.45 0.00
```

So, given that we have taken 2 blue and 1 white out of the bag is 0, the plausibility of:

- all marbles being white (0 ways) = 0
- 1 blue, 3 white (3 ways) = 0.5
- 2 blue, 2 white (8 ways) = 0.4
- 3 blue, 1 white (9 ways) = .5
- 4 blue, 0 white = 0

Binomial distribution (`dbinom`)

In the water/land world-tossing example, what is the lieklihood of the data (where we observed 6 water and 3 land) if we assume the probability of observing 'water' is 0.5?

```
# R code 2.2: binomial distribution
dbinom(6, size = 9, prob = .5)
```

```
[1] 0.1640625
```

Grid approximation

Build a grid approximation for the model we've built so far using the following steps:

1) Define the grid (decide how many points to use in estimateing the posterior, and make a list of the parameter value on the grid)
2) Compute the value of the prior at each parameter value on the grid.
3) Compute the likelihood at each parameter value.
4) Compute the unstandardized posterior at each parameter value, by multiplying the prior by the likelihood.
5) Finally, standardize the posterior, by dividing each value by the sum of all values.

Here we will make a grid of just 20 points:

```
# R code 2.4: grid approximation for 20 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=20 )

# 2) define prior
prior <- rep( 1 , 20 )

# 3) compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

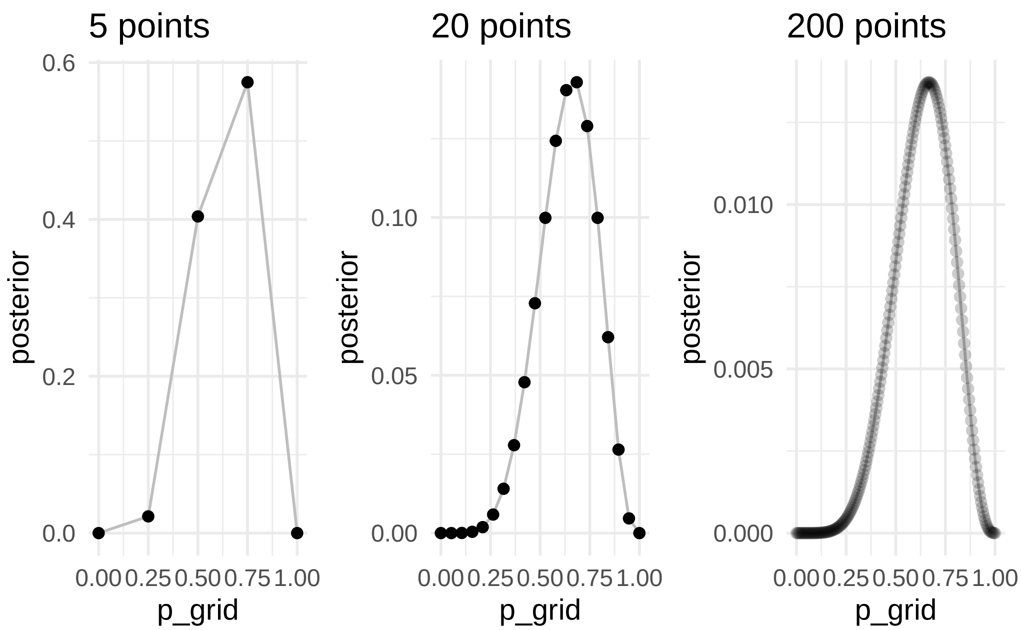
```
# R code 2.4 (my tidyverse version)

fig_grid_20 <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "20 points")
```

```
## Repeat for 200 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=200 )

# 2) define prior
prior <- rep( 1 , 200 )

# 3) compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )
```

```r
# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```r
## Repeat for 5 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=5 )

# 2) define prior
prior <- rep( 1 , 5 )

# 3) compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```r
fig_grid_5 + fig_grid_20 + fig_grid_200
```

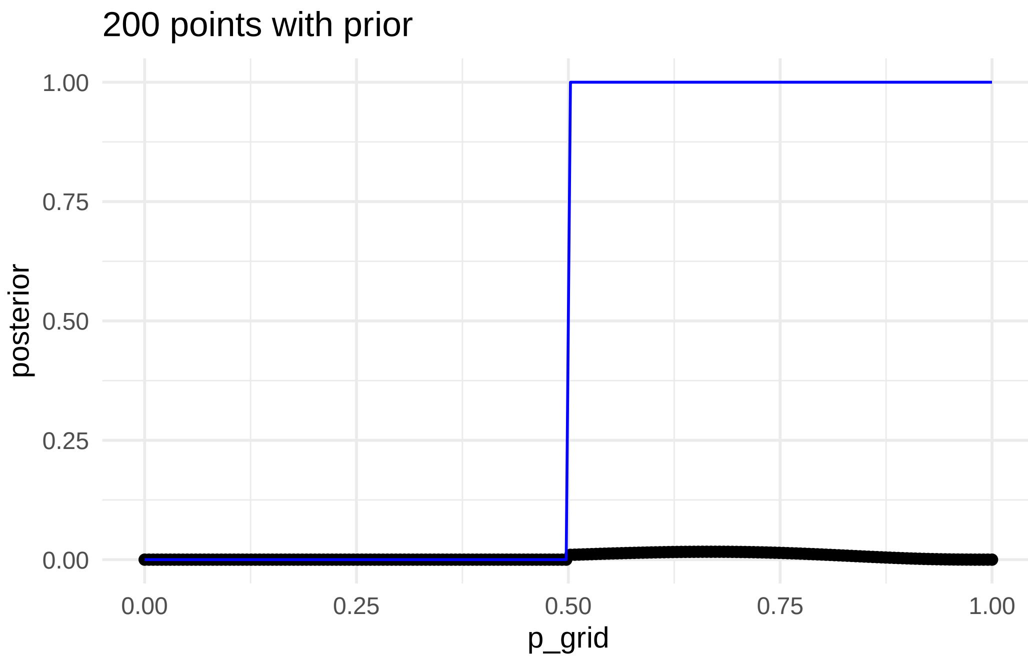


Adjust priors

```
## Repeat for 5 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=200 )

# 2) define prior
prior <- ifelse( p_grid < 0.5 , 0 , 1 )
# prior <- exp( -5*abs( p_grid - 0.5 ) )

# 3) compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```
# fig_grid_5_prior <-
  cbind(p_grid, posterior, prior) |>
```

```
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  geom_line(aes(y = prior), colour = "blue") +
  theme_minimal() +
  labs(title = "200 points with prior")
```



Okay a bit wonky (different y-scales), but we get the point.

Quadratic approximation

We use the `quap` function from the `rethinking` package. This function takes a formula which defines the probability of the data in the prior.

```
# R code 2.6
globe.qa <- quap( alist( W ~ dbinom( W+L ,p) , # binomial likelihood
                         p ~ dunif(0,1) # uniform prior
                         ), data=list(W=6,L=3) )
# display summary of quadratic approximation
```

```r
precis( globe.qa )
```

```
       mean        sd      5.5%     94.5%
p 0.6666664 0.1571338 0.4155361 0.9177966
```

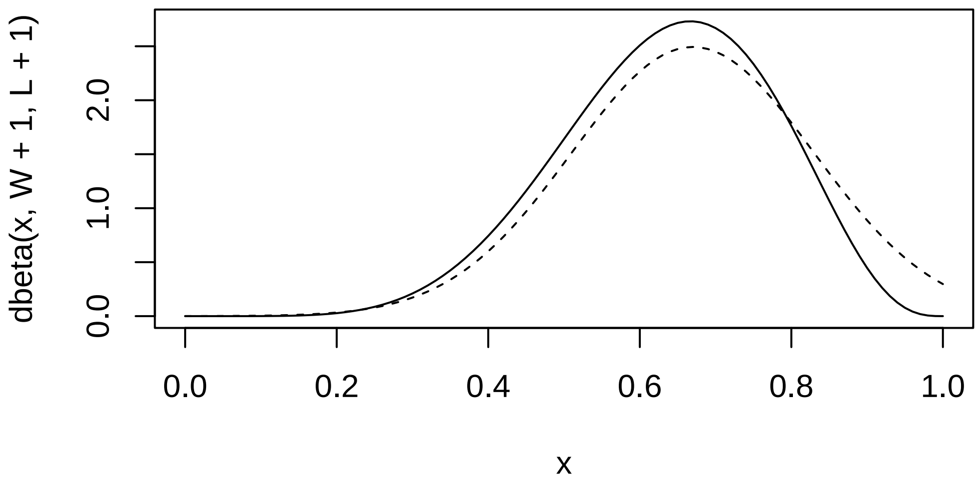
Print output with `precis()`:

- `mean`: posterior mean value of p (peak of the curvature)
- `sd`: the curvature; standard deviation of the posterior distribution
- `5.5%`-94.5%: 89% percentile

This output can be read *Assuming the posterior is Gaussian, it is maximized at 0.67, and its standard deviation is 0.16.*

Check curvature

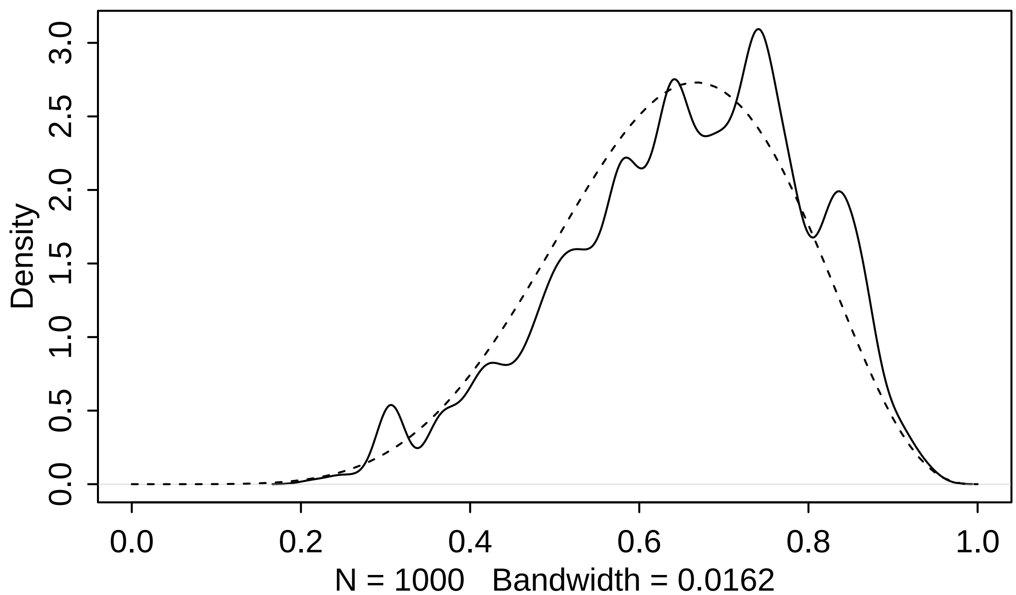
```r
# analytical calculation
W <- 6
L <- 3
curve( dbeta( x , W+1 , L+1 ) , from=0 , to=1 )
# quadratic approximation
curve( dnorm( x , 0.67 , 0.16 ) , lty=2 , add=TRUE )
```



Markov chain Monte Carlo

```
# R code 2.8

n_samples <- 1000
p <- rep(NA , n_samples)
p[1] <- 0.5
W <- 6
L <- 3

for (i in 2:n_samples) {
    p_new <- rnorm(1 , p[i - 1] , 0.1)
    if (p_new < 0)  p_new <- abs(p_new)
    if (p_new > 1) p_new <- 2 - p_new
    q0 <- dbinom(W , W + L , p[i - 1])
    q1 <- dbinom(W , W + L , p_new)
    p[i] <- ifelse(runif(1) < q1 / q0 , p_new , p[i - 1])
  }
```

```
# R code 2.9
dens( p , xlim=c(0,1) )
curve( dbeta( x , W+1 , L+1 ) , lty=2 , add=TRUE )
```



Practice

Easy

2E1: $\frac{Pr(rain,Monday)}{Pr(Monday)}$ is read *the probability of rain on Monday*

2E2: $Pr(Monday|rain)$ is read *the probability that it is Monday, given that it is raining*

2E3: $Pr(Monday|rain)$ is read *the probability that it is Monday, given that it is raining*

2E4: *The probability of water in 0.7* means that…

Medium

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.

1. W W W
2. W W W L
3. L W W L W W W

W W W

```
# R code 2.4: grid approximation for 20 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=20 )

# 2) define prior
prior <- rep( 1 , 20 )

# 3) compute likelihood at each value in grid
likelihood <- dbinom(3 , size=3 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```r
# R code 2.4 (my tidyverse version)

fig_grid_www <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "W W W")
```

W W W L

```r
# R code 2.4: grid approximation for 20 points

# 3) compute likelihood at each value in grid
likelihood <- dbinom(3 , size=4 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

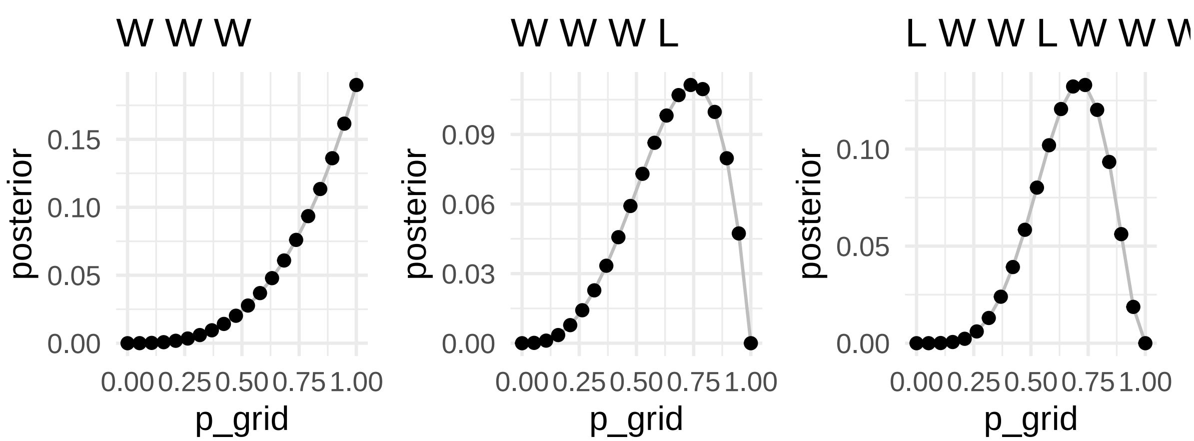
```r
# R code 2.4 (my tidyverse version)

fig_grid_wwwl <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "W W W L")
```

L W W L W W W

```r
# R code 2.4: grid approximation for 20 points

# 3) compute likelihood at each value in grid
likelihood <- dbinom(5 , size=7 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```r
# R code 2.4 (my tidyverse version)

fig_grid_lwwlwww <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "L W W L W W W")
```

```r
library(patchwork)
fig_grid_www + fig_grid_wwwl + fig_grid_lwwlwww
```



2M2:

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

```
# R code 2.4: grid approximation for 20 points

# 1) define grid
p_grid <- seq( from=0 , to=1 , length.out=20 )

# 2) define prior
prior <- ifelse(p_grid < .5, 0, 1)

# 3) compute likelihood at each value in grid
likelihood <- dbinom(3 , size=3 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```
# R code 2.4 (my tidyverse version)

fig_grid_www <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "W W W")
```

W W W L

```
# R code 2.4: grid approximation for 20 points

prior <- ifelse(p_grid < .5, 0, 1)
```

```
# 3) compute likelihood at each value in grid
likelihood <- dbinom(3 , size=4 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```
# R code 2.4 (my tidyverse version)

fig_grid_wwwl <-
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "W W W L")
```

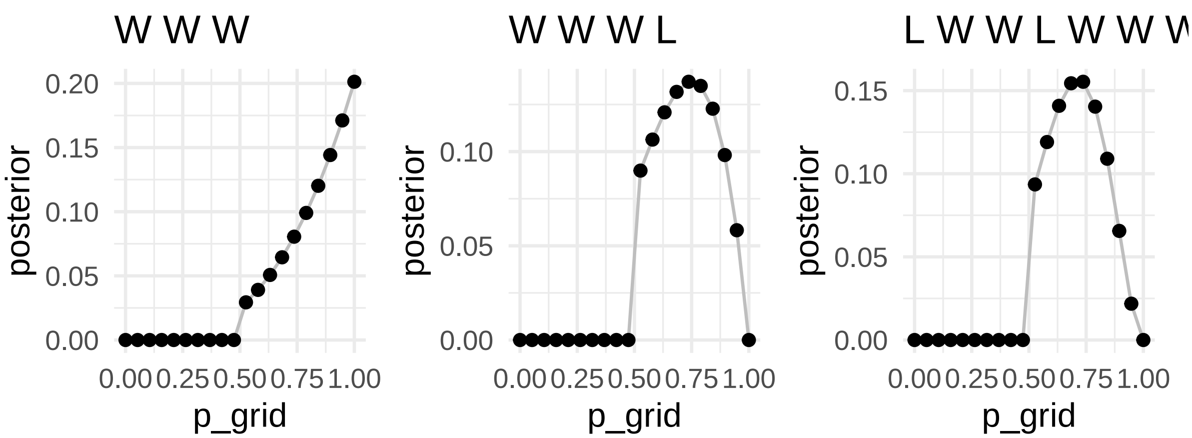
L W W L W W W

```
# R code 2.4: grid approximation for 20 points

prior <- ifelse(p_grid < .5, 0, 1)

# 3) compute likelihood at each value in grid
likelihood <- dbinom(5 , size=7 , prob=p_grid )
# 3 waters (successes) with 3 tosses (size)

# 4) compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# 5) standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

```
# R code 2.4 (my tidyverse version)

fig_grid_lwwlwww <-
```

```r
  cbind(p_grid, posterior) |>
  as_tibble() |>
  ggplot() +
  aes(x = p_grid, y = posterior) +
  geom_line(colour = "grey") +
  geom_point() +
  theme_minimal() +
  labs(title = "L W W L W W W")
```

```r
library(patchwork)
fig_grid_www + fig_grid_wwwl + fig_grid_lwwlwww
```



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.

$$
\begin{align}
P(A|B) &= \frac{P(B|A)P(A)}{P(B)} \tag{1}\\
P(Earth|land) &= \frac{P(land|Earth)P(Earth)}{P(land)} \tag{2}\\
0.23 &= \frac{.3 * .5}{P(land)} \tag{3}\\
0.23 &= \frac{.3 * .5}{P(land|Earth) * P(land|Mars)} \tag{4}\\
0.23 &= \frac{.3 * .5}{(0.3 * 0.5) + (1 * 0.5)} \tag{5}\\
0.23 &= \frac{.3 * .5}{((0.3 + 1)/2)} \tag{6}
\end{align}
$$

```
0.3*0.5/((0.3+1)/2) #0.2307692
```

```
[1] 0.2307692
```

```
#prob(earth|land) = (prob(land/earth)*prob(earth))/prob(land)
```

2M4:

2M5:

2M6:

2M7:

Hard