Untitled

Angel V. Jimenez

22 March 2019

# CHAPTER 2

# 2.4.3. GRID APPROXIMATION

Problem: distribution of water and land on Earth. 9 tossess of the globe, 6 waters.

# define grid  
p\_grid <- seq( from=0 , to=1 , length.out=20 ) # Creation of grid of 20 points between 0 and 1 (continuous)  
p\_grid

## [1] 0.00000000 0.05263158 0.10526316 0.15789474 0.21052632 0.26315789  
## [7] 0.31578947 0.36842105 0.42105263 0.47368421 0.52631579 0.57894737  
## [13] 0.63157895 0.68421053 0.73684211 0.78947368 0.84210526 0.89473684  
## [19] 0.94736842 1.00000000

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

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

# compute likelihood at each value in grid  
likelihood <- dbinom( 6 , size=9 , prob=p\_grid ) # likelihood of 6 waters in 9 tossess given the different likelihoods given by the grid  
likelihood

## [1] 0.000000e+00 1.518149e-06 8.185093e-05 7.772923e-04 3.598575e-03  
## [6] 1.116095e-02 2.668299e-02 5.292110e-02 9.082698e-02 1.383413e-01  
## [11] 1.897686e-01 2.361147e-01 2.666113e-01 2.714006e-01 2.450051e-01  
## [16] 1.897686e-01 1.179181e-01 5.026670e-02 8.853845e-03 0.000000e+00

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

## [1] 0.000000e+00 1.518149e-06 8.185093e-05 7.772923e-04 3.598575e-03  
## [6] 1.116095e-02 2.668299e-02 5.292110e-02 9.082698e-02 1.383413e-01  
## [11] 1.897686e-01 2.361147e-01 2.666113e-01 2.714006e-01 2.450051e-01  
## [16] 1.897686e-01 1.179181e-01 5.026670e-02 8.853845e-03 0.000000e+00

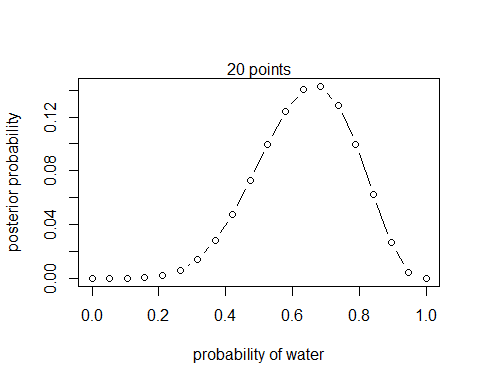
# standardize the posterior, so it sums to 1  
posterior <- unstd.posterior / sum(unstd.posterior)  
posterior

## [1] 0.000000e+00 7.989837e-07 4.307717e-05 4.090797e-04 1.893887e-03  
## [6] 5.873873e-03 1.404294e-02 2.785174e-02 4.780115e-02 7.280739e-02  
## [11] 9.987296e-02 1.242643e-01 1.403143e-01 1.428349e-01 1.289433e-01  
## [16] 9.987296e-02 6.205890e-02 2.645477e-02 4.659673e-03 0.000000e+00

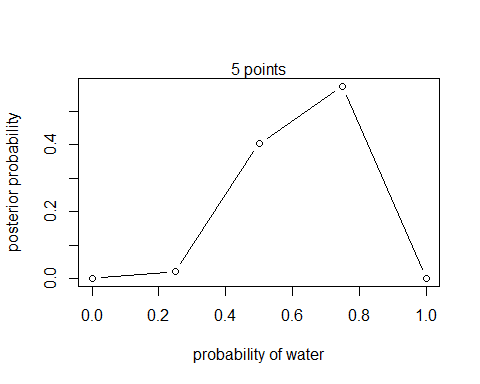
sum(posterior)# check that it sums to 1

## [1] 1

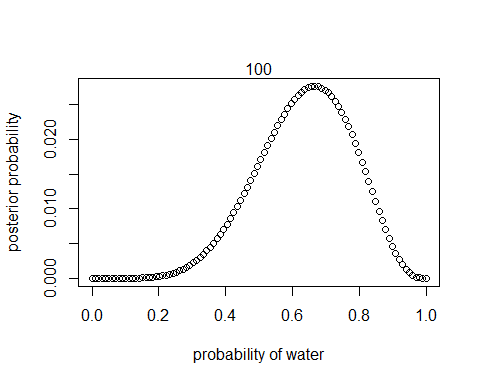
# display of posterior distribution  
plot( p\_grid , posterior , type="b" ,  
 xlab="probability of water" , ylab="posterior probability" )  
mtext( "20 points" )



# grid of only 5 points  
# define grid  
p\_grid <- seq( from=0 , to=1 , length.out=5 )  
# define prior  
prior <- rep( 1 , 5 )  
# 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( "5 points" )



# grid of 100 points  
# define grid  
p\_grid <- seq( from=0 , to=1 , length.out=100 )  
# define prior  
prior <- rep( 1 , 100 )  
# 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( "100" )

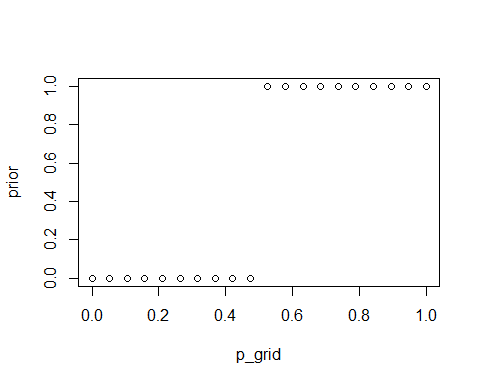


The results are quite similar with 5, 20 and 100 points but the uncertainty reduces with more points (the posterior probability is much smaller with 100 points)

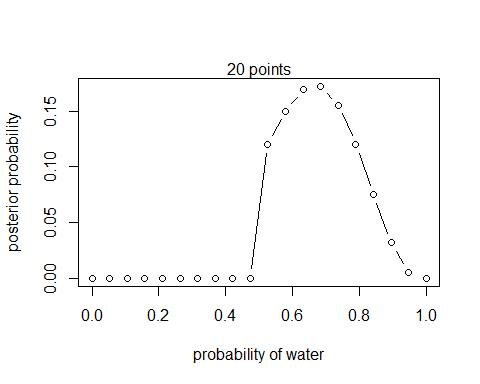
# DIFFERENT PRIOR: We think the probability of water is between 0.5 and 1  
# define grid  
p\_grid <- seq( from=0 , to=1 , length.out=20 )  
# Different prior  
prior <- ifelse( p\_grid < 0.5 , 0 , 1 )  
prior

## [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1

plot(p\_grid, prior)



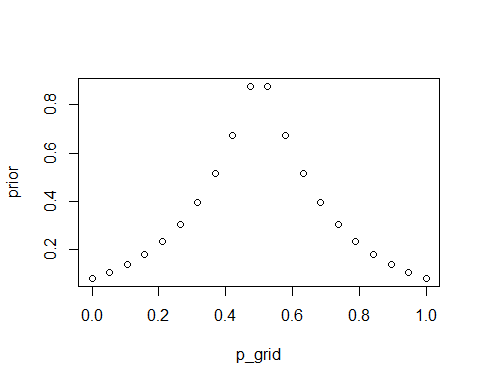
# compute likelihood at each value in grid  
likelihood <- dbinom( 6 , size=9 , prob=p\_grid )  
# compute product of likelihood and prior  
unstd.posterior <- likelihood \* prior  
# standardize the posterior, so it sums to 1  
posterior <- unstd.posterior / sum(unstd.posterior)  
plot( p\_grid , posterior , type="b" ,  
xlab="probability of water" , ylab="posterior probability" )  
mtext( "20 points" )



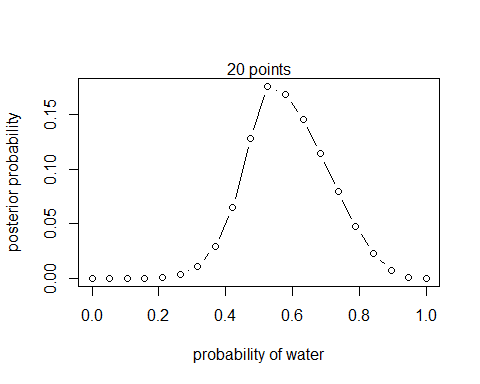
# DIFFERENT PRIOR:   
# define grid  
p\_grid <- seq( from=0 , to=1 , length.out=20 )  
prior <- exp( -5\*abs( p\_grid - 0.5 ) ) # What the hell does this mean??  
prior

## [1] 0.0820850 0.1067952 0.1389440 0.1807706 0.2351884 0.3059877 0.3980998  
## [8] 0.5179406 0.6738573 0.8767101 0.8767101 0.6738573 0.5179406 0.3980998  
## [15] 0.3059877 0.2351884 0.1807706 0.1389440 0.1067952 0.0820850

plot(p\_grid, prior)



# 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" )



# 2.4.4. QUADRATIC APPROXIMATION

library(rethinking)

## Loading required package: rstan

## Loading required package: ggplot2

## Loading required package: StanHeaders

## rstan (Version 2.18.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)

## For improved execution time, we recommend calling  
## Sys.setenv(LOCAL\_CPPFLAGS = '-march=native')  
## although this causes Stan to throw an error on a few processors.

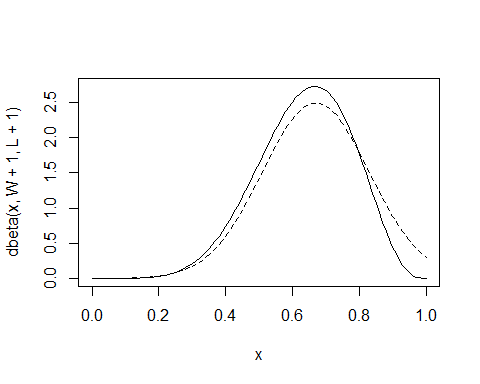
## Loading required package: parallel

## rethinking (Version 1.88)

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  
precis( globe.qa )

## mean sd 5.5% 94.5%  
## p 0.6666605 0.1571352 0.4155281 0.917793

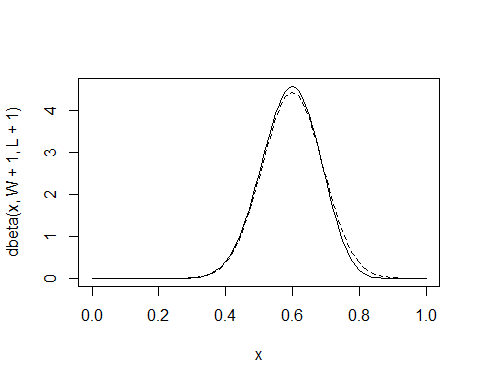
# analytical calculation  
W <- 6  
L <- 3  
curve( dbeta( x , W+1 , L+1 ) , from=0 , to=1 )  
# quadratic approximation  
curve( dnorm( x , 0.67 , 0.16 ) , lty=2 , add=TRUE )



# ANALYTICAL CALCULATION WITH DOUBLE OF DATA BUT SAME PROPORTION  
globe.qa <- quap(  
alist(  
W ~ dbinom( W+L ,p) , # binomial likelihood  
p ~ dunif(0,1) # uniform prior  
) ,  
data=list(W=18,L=12) )  
# display summary of quadratic approximation  
precis( globe.qa )

## mean sd 5.5% 94.5%  
## p 0.6000002 0.08944227 0.4570542 0.7429462

# Analytical calculation  
W <- 18  
L <- 12  
curve( dbeta( x , W+1 , L+1 ) , from=0 , to=1 )  
# quadratic approximation  
curve( dnorm( x , 0.6 , 0.09 ) , lty=2 , add=TRUE )



# 2.4.5. MCMC

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] )  
}

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

