Statistical Rethinking Notes - Chapter 3

Zachary Levonian

2022

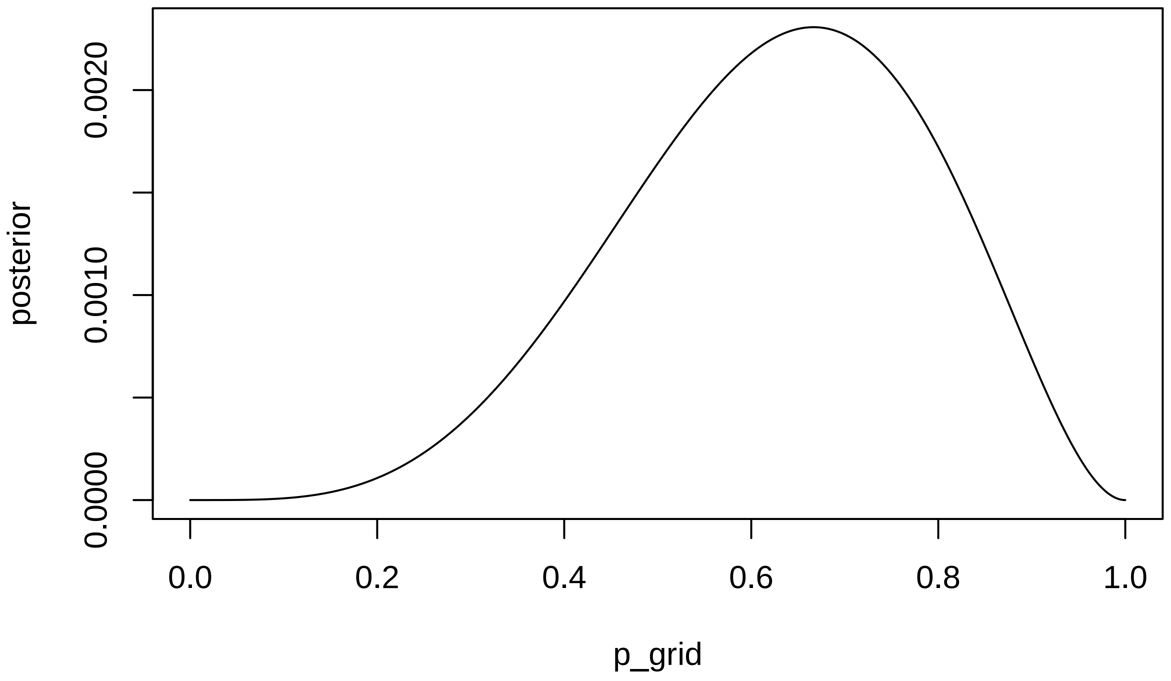
Chapter 3

Book code and notes

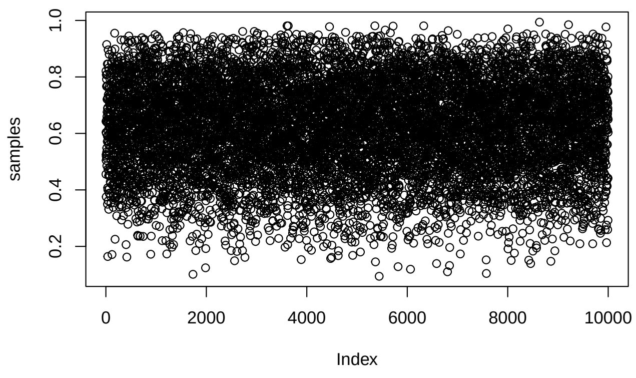
```
library(rethinking)
library(scales)  # for alpha in the plots
```

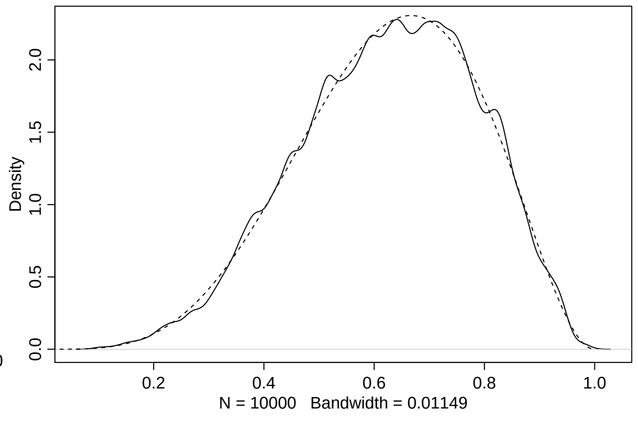
```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prob_p <- rep( 1 , 1000 )
prob_data <- dbinom(4, size=6, prob=p_grid)
posterior <- prob_data * prob_p
posterior <- posterior / sum(posterior)
```

```
plot(p_grid, posterior, type="l")
```



```
samples <- sample(p_grid, prob=posterior, size=1e4, replace=TRUE)
plot(samples)
dens(samples)
lines(p_grid, posterior * 1000, lty=2)  # why do I need to multiply by 1000?
```





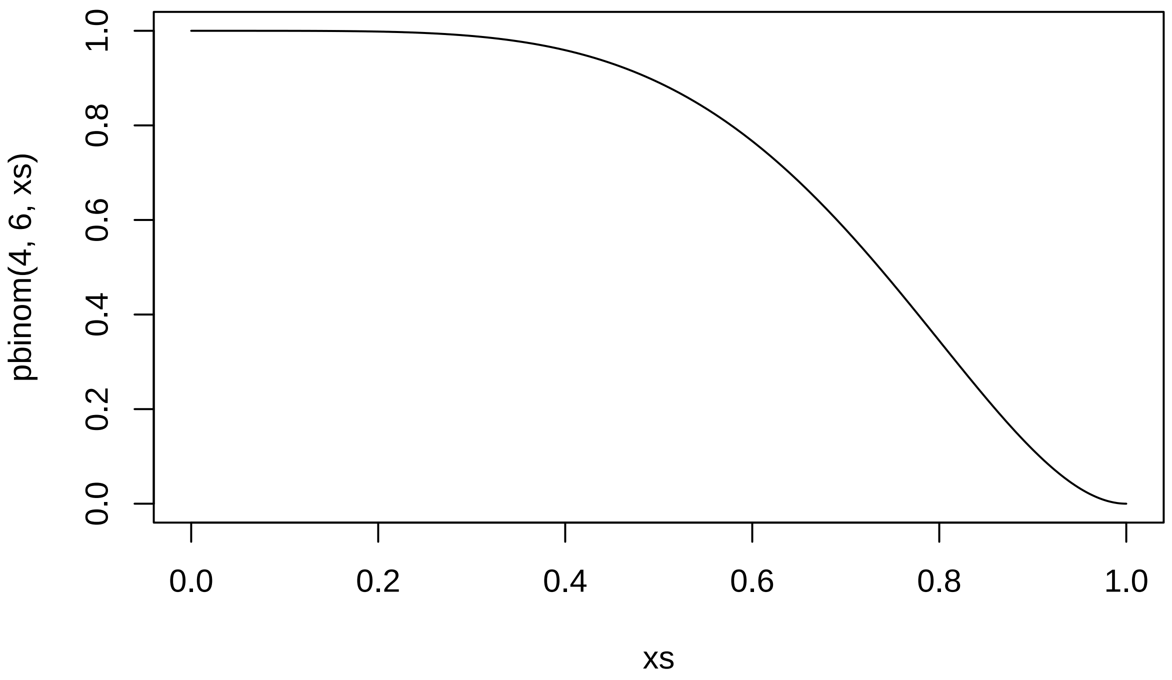
```
#dbinom(4, size=6, prob=0.5)  # binomial density at 0.5; probability of getting 4 or less lands from 6
sum(posterior[p_grid < 0.5])  # grid approximation
```

```
## [1] 0.2265622
```

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

```
## [1] 0.226
```

```
xs <- seq( from=0 , to=1 , length.out=1000 )
plot(xs, pbinom(4, 6, xs), type="l")
```



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

```
##       10%       90%
## 0.4053053 0.8308308
```

```
# PI = "percentile interval"
PI(samples , prob=0.8)  # convenience form of the above
```

```
##       10%       90%
## 0.4053053 0.8308308
```

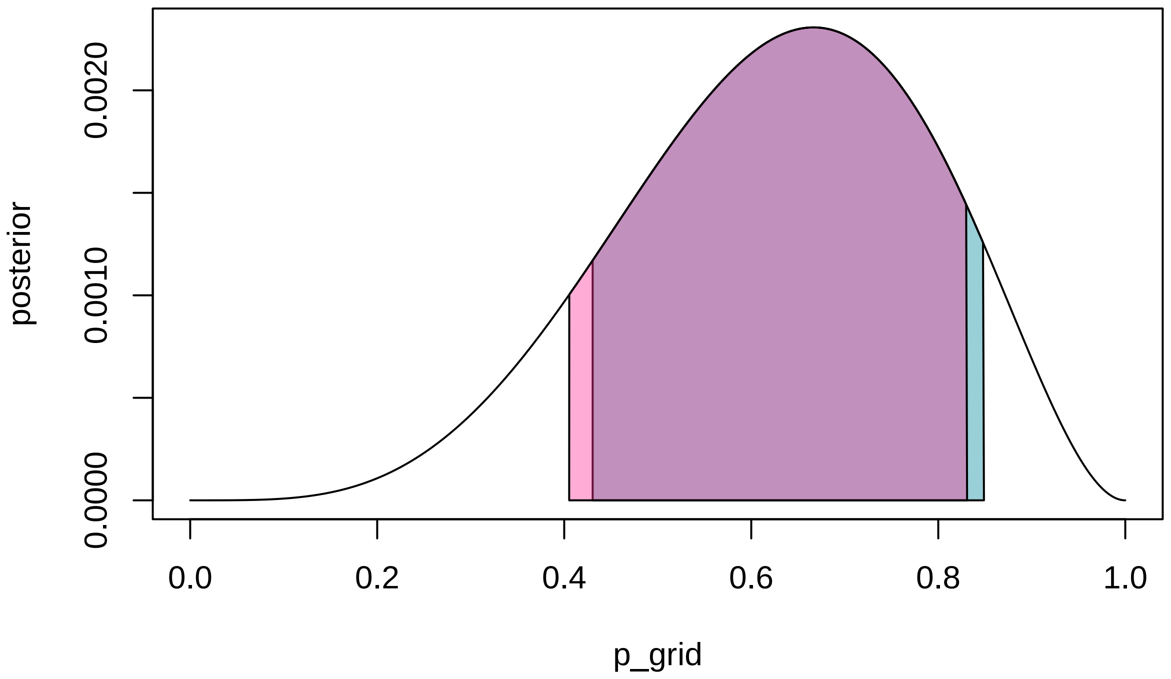
```
# HPDI = "highest posterior density interval"
HPDI(samples , prob=0.8)
```

```
##      |0.8      0.8|
## 0.4304304 0.8488488
```

```
plot(p_grid, posterior, type="l")
x <- p_grid

# in green, the highest posterior density interval
bounds <- HPDI(samples, prob=0.8)
lb <- bounds[1]
ub <- bounds[2]
polygon(c(x[x>=lb & x<ub], ub, lb), c(posterior[x>=lb & x<ub], 0, 0), col=alpha("#008999", 0.4))

# in pink, the percentile interval
bounds <- PI(samples, prob=0.8)
lb <- bounds[1]
ub <- bounds[2]
polygon(c(x[x>=lb & x<ub], ub, lb), c(posterior[x>=lb & x<ub], 0, 0), col=alpha("#FF3399", 0.4))
```



```
# MAP point estimation (Maximum a posteriori)
p_grid[which.max(posterior)]
```

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

```
chainmode(samples , adj=0.01)  # samples based approximation
```

```
## [1] 0.6313499
```

Absolute loss:

```
loss <- sapply( p_grid , function(d) sum( posterior*abs( d - p_grid ) ) )
p_grid[which.min(loss)]
```

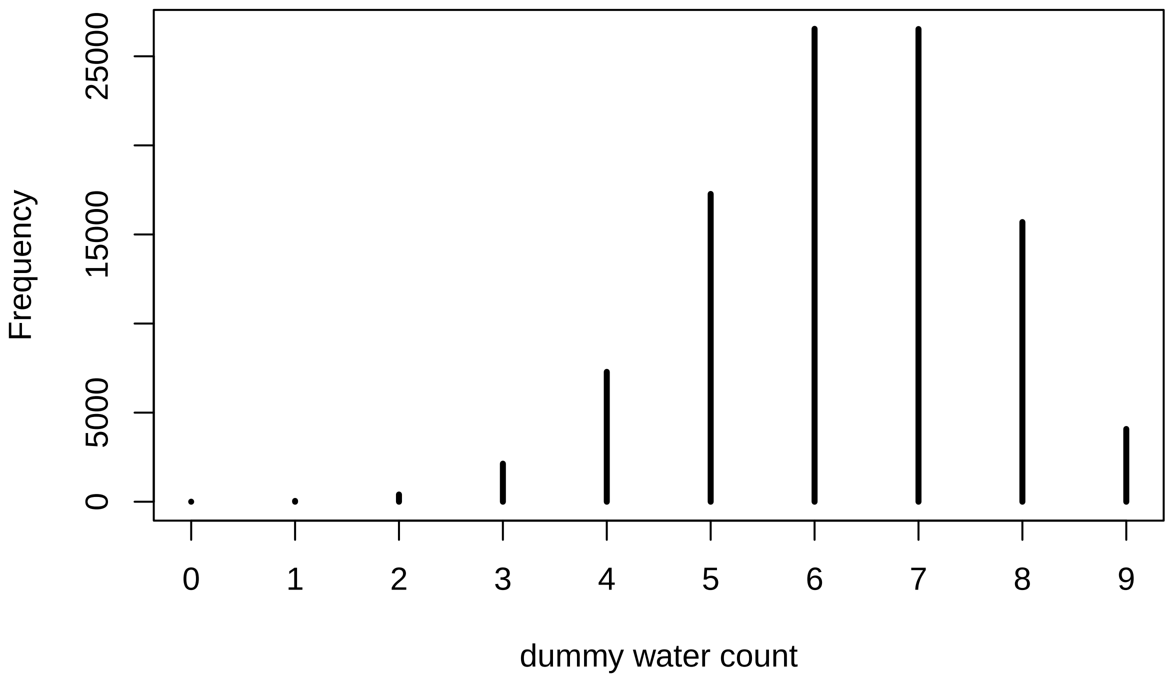
```
## [1] 0.6356356
```

```
median(samples)
```

```
## [1] 0.6356356
```

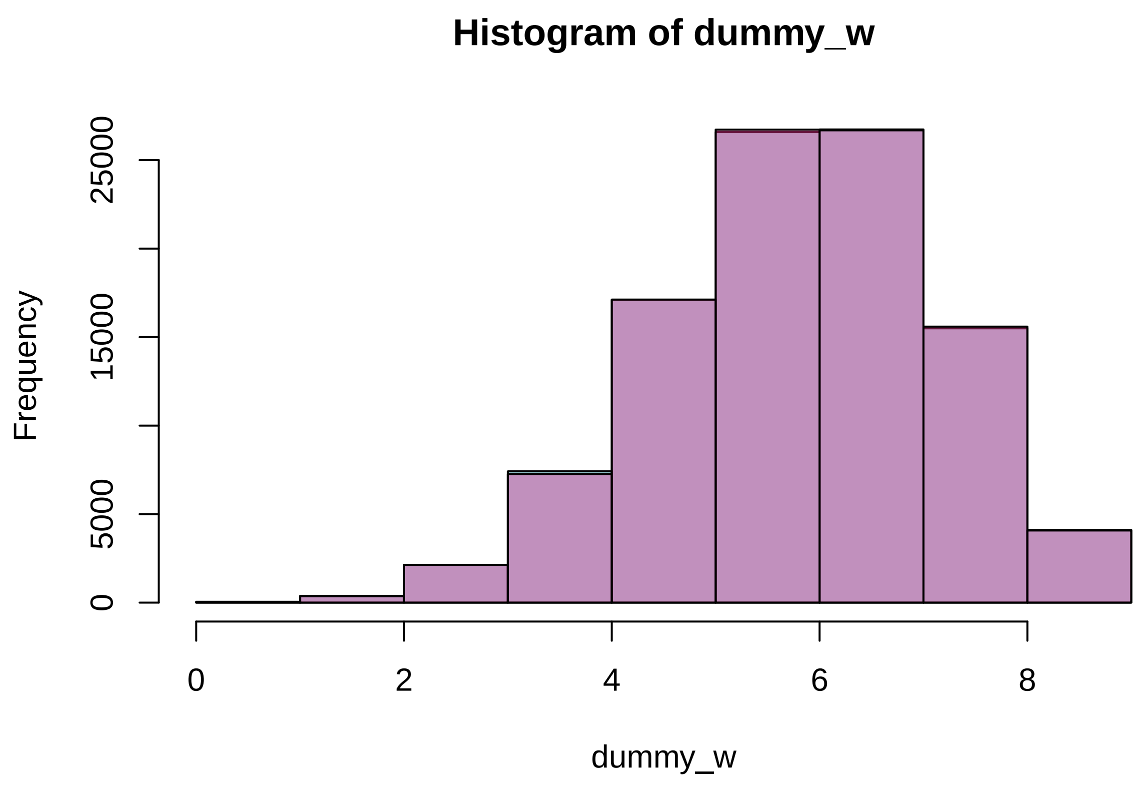
Key point: "Different loss functions nominate different point estimates."

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

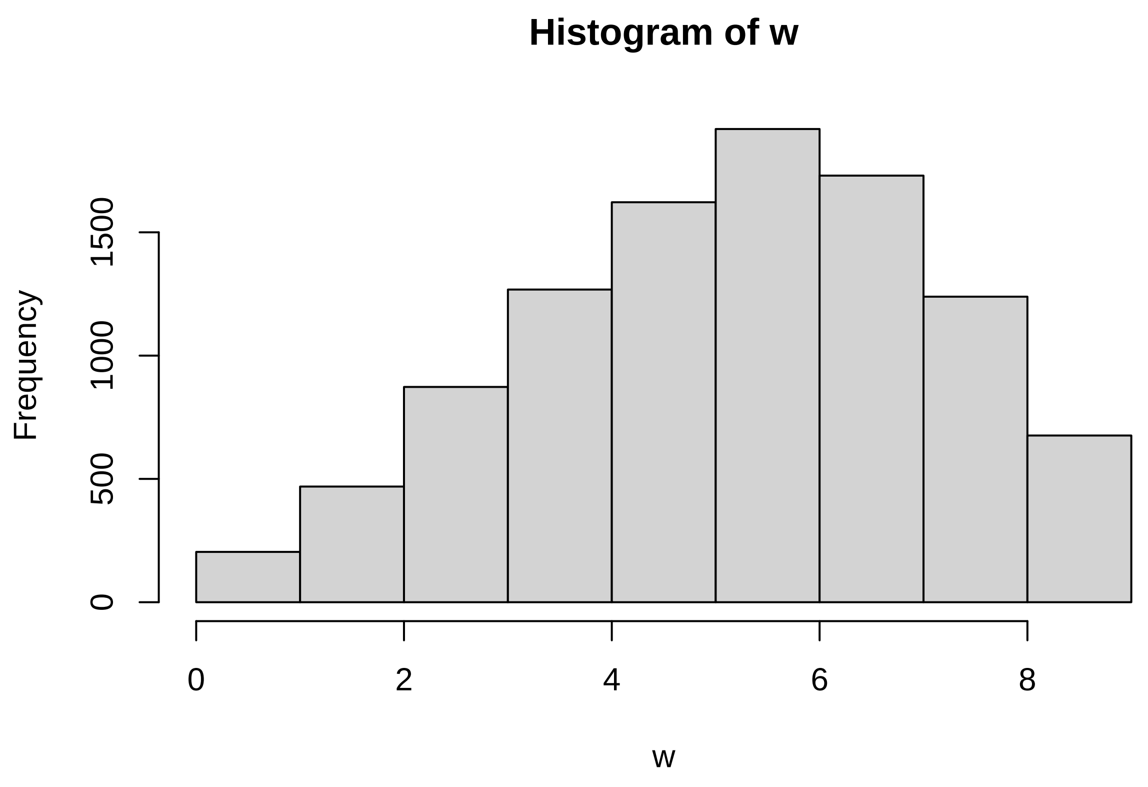


```
n <- 1e5
size <- 9
prob <- 0.7
breaks <- 0:size

dummy_w <- rbinom(n, size=size, prob=prob)
hg1 <- hist(dummy_w, plot=FALSE, breaks=breaks)
dummy_w <- rbinom(n, size=size, prob=prob)
#dummy_w <- sample(p_grid, prob=posterior, size=n, replace=TRUE)
hg2 <- hist(dummy_w, plot=FALSE, breaks=breaks)
plot(hg1, col=alpha("#008999", 0.4)) # Plot 1st histogram using a transparent color
plot(hg2, col=alpha("#FF3399", 0.4), add = TRUE) # Add 2nd histogram using different color
```



```
# "simulated observations averaged over the posterior"
# in other words, this is a realization of the posterior in a sample
w <- rbinom(1e4 , size=9 , prob=samples)
hist(w, breaks=0:9)
```



Book problems

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

```r
sum(samples < 0.2) / length(samples)
```

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

```r
sum(samples > 0.8) / length(samples)
```

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

```r
sum(samples > 0.2 & samples < 0.8) / length(samples)
```

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

```r
quantile(samples, probs=c(0.2, 0.8))
```

```
##       20%       80% 
## 0.5185185 0.7557558
```

```r
HPDI(samples, 0.66)
```

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

```r
PI(samples, 0.66)
```

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