Statistical Rethinking Winter 2020 – Homework Week 1

Alessandro Gentilini .@gmail (just auditing)

November 25, 2020

1. Suppose the globe tossing data (Chapter 2) had turned out to be 4 water in 15 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as in the book.

The peel is the land	All the peel	All the peel minus the "water" peel
		

I pretend to ignore the fact that in the globe tossing experiment the likelihood is a binomial and so I create my own likelihood function on the basis of my comprehension of the data generation mechanism (for me code efficiency is not a goal here).

I assume a simplified world with a fully connected "slice" ocean extending from the South (geographic) Pole to the North Pole and from the zero degree longitude to the $p \cdot 2\pi$ longitude where p is the water proportion (`p` in the following code), i.e. $p = 1$ means a globe with no land at all and $p = 0$ means a globe with no water at all (in the above picture the *Citrus deliciosa* is the globe, the peel is the land, and $p = \frac{22-20}{22} = 0.0909091$); the maximum longitude is expressed in radians as 2π. The tossing is then simulated drawing `N_tosses` times a longitude angle from a uniform distribution spanning the whole globe and counting how many times that longitude is in the water; the above is repeated many times (`sz` times) in order to get an estimation of the sought probability.

```
my_likelihood <- function(N_tosses,n_water,water_proportion)
{
    stopifnot(water_proportion>=0)
    stopifnot(water_proportion<=1)
    Greenwich_longitude <- 0
    # assuming a fully connected ocean (and a fully connected continent)
    max_water_longitude <- water_proportion*2*pi
    sz <- 10e3
    cnt <- 0
    for (i in 1:sz){
        # throw N_tosses darts to the sphere
```

```
        longitudes <- runif(N_tosses,min=Greenwich_longitude,max=2*pi)
        # count the times the darts sink in the water
        darts_sank_in_water <- sum(longitudes <= max_water_longitude)
        # count the time we get exactly the given data
        if(n_water==darts_sank_in_water){
            cnt <- cnt+1
        }
    }
    # return a frequency representing the likelihood probability
    return(cnt/sz)
}
```

Then I define a grid for p:

```
pars <- expand.grid(p=seq(from=0,to=1,length.out=20))
```

and I assume a uniform prior for p:

```
pars$prior <- dunif(pars$p,0,1)
```

The following are the given data:

```
N_tosses <- 15
n_water <- 4
```

For each node in the grid I compute the likelihood:

```
for (i in 1:nrow(pars)) {
  likelihoods <- my_likelihood(N_tosses,n_water,pars$p[i])
  pars$likelihood[i] <- prod(likelihoods)
}
```

And I compute the (normalized) posterior:

```
pars$unnormalized_posterior <- pars$likelihood * pars$prior
pars$posterior <- pars$unnormalized_posterior/sum(pars$unnormalized_posterior)
```

I have a look at the numbers:

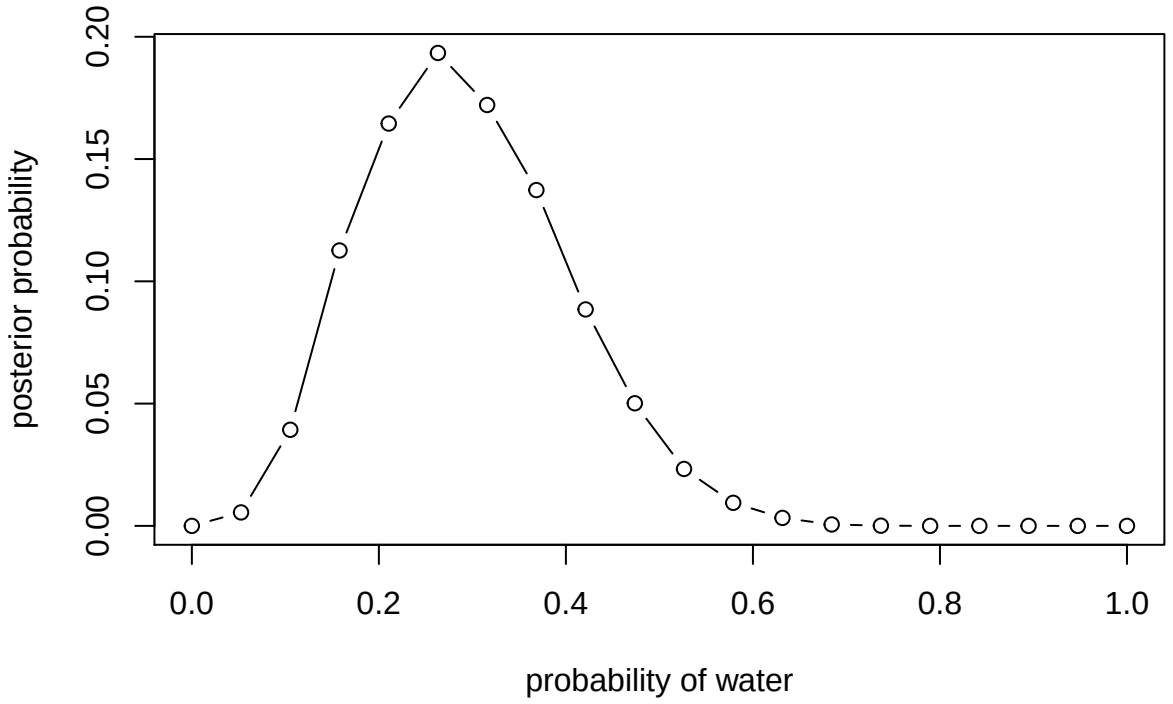
```
print(pars)
```

```
##             p prior likelihood unnormalized_posterior    posterior
## 1  0.00000000     1     0.0000                 0.0000 0.000000e+00
## 2  0.05263158     1     0.0066                 0.0066 5.505965e-03
## 3  0.10526316     1     0.0471                 0.0471 3.929257e-02
## 4  0.15789474     1     0.1350                 0.1350 1.126220e-01
## 5  0.21052632     1     0.1972                 0.1972 1.645116e-01
## 6  0.26315789     1     0.2318                 0.2318 1.933762e-01
## 7  0.31578947     1     0.2063                 0.2063 1.721031e-01
## 8  0.36842105     1     0.1646                 0.1646 1.373154e-01
## 9  0.42105263     1     0.1061                 0.1061 8.851256e-02
## 10 0.47368421     1     0.0601                 0.0601 5.013765e-02
## 11 0.52631579     1     0.0279                 0.0279 2.327521e-02
## 12 0.57894737     1     0.0113                 0.0113 9.426879e-03
## 13 0.63157895     1     0.0039                 0.0039 3.253525e-03
## 14 0.68421053     1     0.0007                 0.0007 5.839660e-04
## 15 0.73684211     1     0.0001                 0.0001 8.342371e-05
## 16 0.78947368     1     0.0000                 0.0000 0.000000e+00
```

```
## 17 0.84210526     1     0.0000                0.0000 0.000000e+00
## 18 0.89473684     1     0.0000                0.0000 0.000000e+00
## 19 0.94736842     1     0.0000                0.0000 0.000000e+00
## 20 1.00000000     1     0.0000                0.0000 0.000000e+00
```

And finally I plot the posterior:

```
plot(pars$p,pars$posterior,type='b',
xlab='probability of water',
ylab='posterior probability')
```



---

2. Start over in **1**, 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. What difference does the better prior make?

I define this prior as `step_prior` and the `constant` is computed in order to give $\int_0^1 \texttt{step\_prior}(p)\, dp = 1$:

```
step_prior <- function(water_proportion)
{
    stopifnot(water_proportion>=0)
    stopifnot(water_proportion<=1)
    threshold <- .5
    area <- 1
    constant <- area/threshold
    return(ifelse(water_proportion<threshold,0,constant))
}
```

I the compute the new prior:

```
pars$prior2 <- step_prior(pars$p)
```

There is no need to recompute the likelihood because they do not depend on the prior and so I only compute the new (normalized) posterior:

```
pars$unnormalized_posterior2 <- pars$likelihood * pars$prior2
pars$posterior2 <- pars$unnormalized_posterior2/sum(pars$unnormalized_posterior2)
```

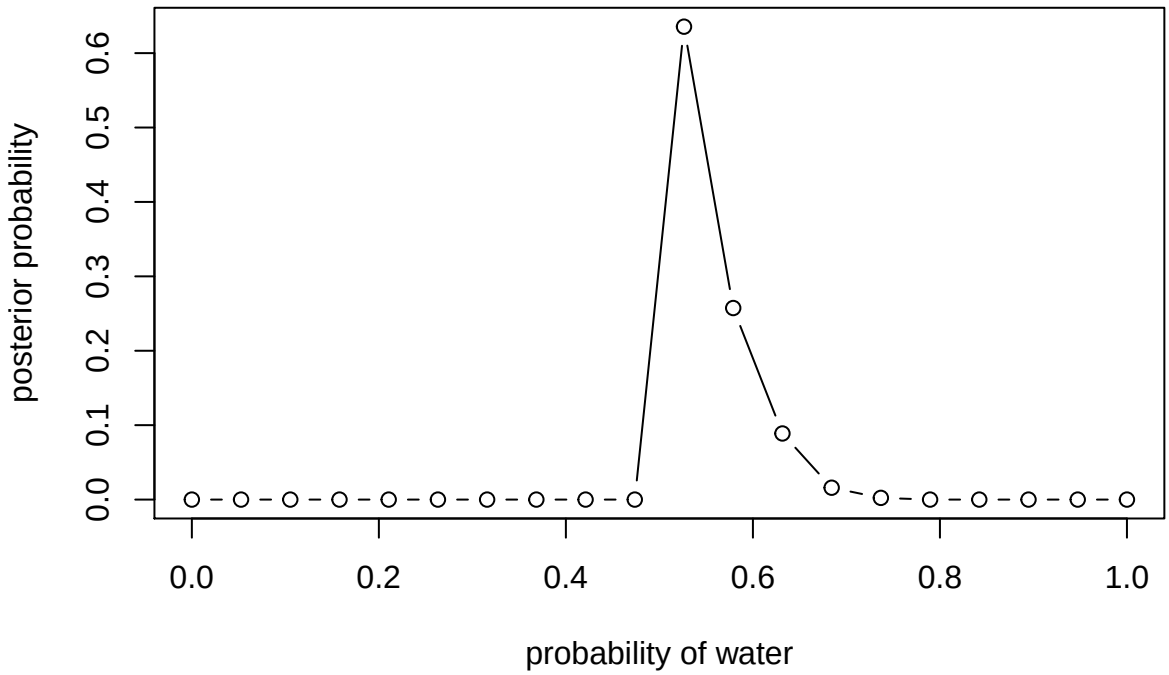
I have a look at the numbers:

```
print(pars)
```

```
##             p prior likelihood unnormalized_posterior    posterior prior2
## 1  0.00000000     1     0.0000                  0.0000 0.000000e+00      0
## 2  0.05263158     1     0.0066                  0.0066 5.505965e-03      0
## 3  0.10526316     1     0.0471                  0.0471 3.929257e-02      0
## 4  0.15789474     1     0.1350                  0.1350 1.126220e-01      0
## 5  0.21052632     1     0.1972                  0.1972 1.645116e-01      0
## 6  0.26315789     1     0.2318                  0.2318 1.933762e-01      0
## 7  0.31578947     1     0.2063                  0.2063 1.721031e-01      0
## 8  0.36842105     1     0.1646                  0.1646 1.373154e-01      0
## 9  0.42105263     1     0.1061                  0.1061 8.851256e-02      0
## 10 0.47368421     1     0.0601                  0.0601 5.013765e-02      0
## 11 0.52631579     1     0.0279                  0.0279 2.327521e-02      2
## 12 0.57894737     1     0.0113                  0.0113 9.426879e-03      2
## 13 0.63157895     1     0.0039                  0.0039 3.253525e-03      2
## 14 0.68421053     1     0.0007                  0.0007 5.839660e-04      2
## 15 0.73684211     1     0.0001                  0.0001 8.342371e-05      2
## 16 0.78947368     1     0.0000                  0.0000 0.000000e+00      2
## 17 0.84210526     1     0.0000                  0.0000 0.000000e+00      2
## 18 0.89473684     1     0.0000                  0.0000 0.000000e+00      2
## 19 0.94736842     1     0.0000                  0.0000 0.000000e+00      2
## 20 1.00000000     1     0.0000                  0.0000 0.000000e+00      2
##    unnormalized_posterior2  posterior2
## 1                   0.0000 0.000000000
## 2                   0.0000 0.000000000
## 3                   0.0000 0.000000000
## 4                   0.0000 0.000000000
## 5                   0.0000 0.000000000
## 6                   0.0000 0.000000000
## 7                   0.0000 0.000000000
## 8                   0.0000 0.000000000
## 9                   0.0000 0.000000000
## 10                  0.0000 0.000000000
## 11                  0.0558 0.635535308
## 12                  0.0226 0.257403189
## 13                  0.0078 0.088838269
## 14                  0.0014 0.015945330
## 15                  0.0002 0.002277904
## 16                  0.0000 0.000000000
## 17                  0.0000 0.000000000
## 18                  0.0000 0.000000000
## 19                  0.0000 0.000000000
## 20                  0.0000 0.000000000
```

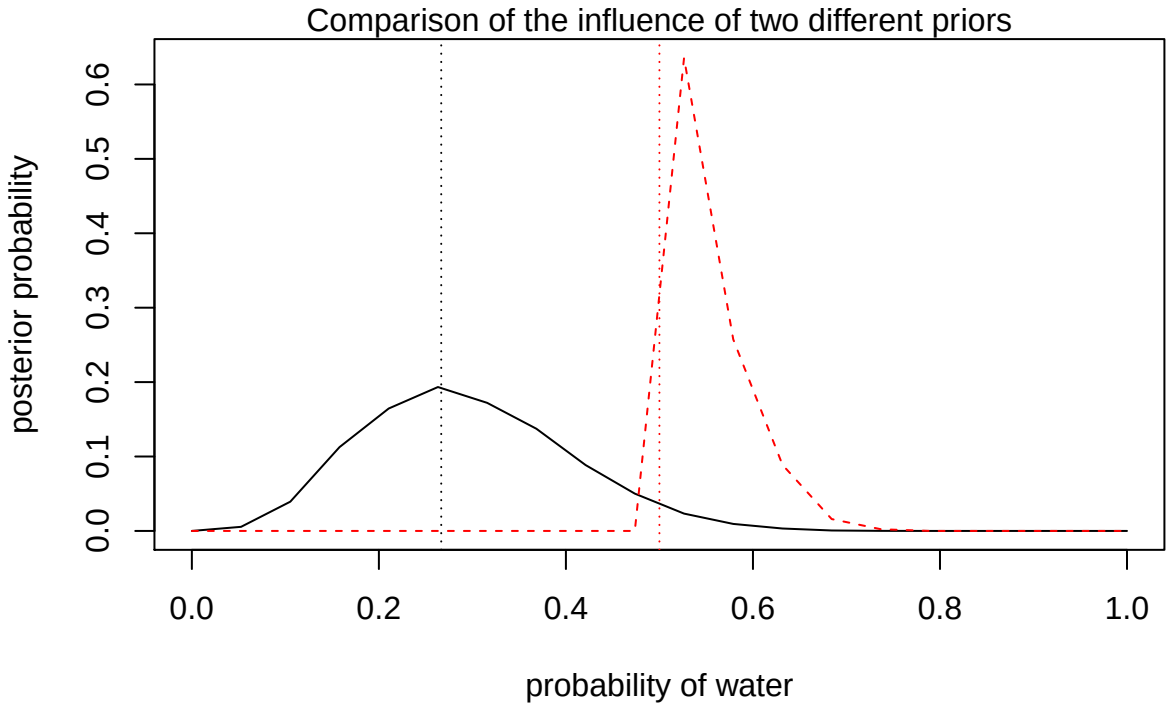
And finally I plot the new posterior:

```
plot(pars$p,pars$posterior2,type='b',
xlab='probability of water',
ylab='posterior probability')
```



The difference between the previous posterior and the new one is that the first posterior, assuming a flat probability for p, has its peak near the simplest estimation (given the data) of $\frac{4}{15} = 0.2666667$ (dotted black vertical line in the following plot) and so it is fully dominated by the data while the new posterior has its peak slightly greater than $p = 0.5$ because the given data is not supporting a probability of water greater than 50% (dotted red vertical line in the following plot).

```
matplot(pars$p,cbind(pars$posterior,pars$posterior2),
type=c('l'),
#pch=c(1,3),
xlab='probability of water',
ylab='posterior probability')
abline(v=n_water/N_tosses,col='black',lty='dotted')
abline(v=.5,col='red',lty='dotted')
mtext('Comparison of the influence of two different priors')
```



---

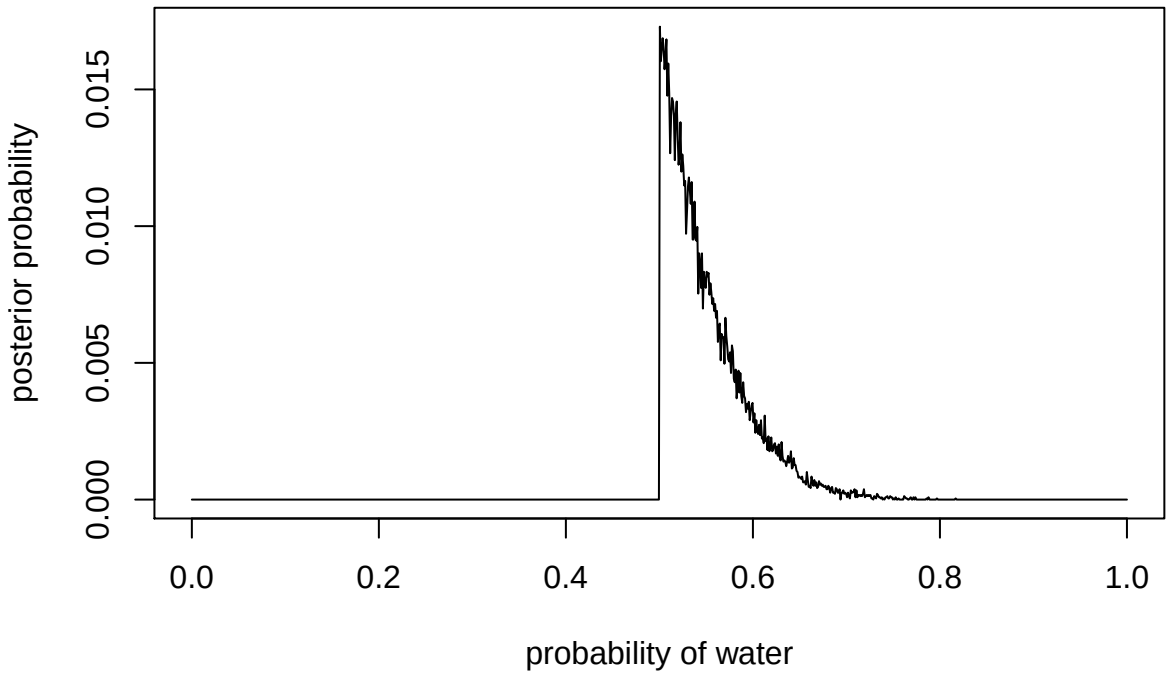
3. For the posterior distribution from **2**, compute 89% percentile and HPDI intervals. Compare the widths of these intervals. Which is wider? Why? If you had only the information in the interval, what might you misunderstand about the shape of the posterior distribution?

Following the R code snippet 3.11 from the book, I choose a larger number of nodes in the grid:

```
pars <- expand.grid(p=seq(from=0,to=1,length.out=1000))
pars$prior2 <- step_prior(pars$p)
for (i in 1:nrow(pars)) {
  likelihoods <- my_likelihood(N_tosses,n_water,pars$p[i])
  pars$likelihood[i] <- prod(likelihoods)
}
pars$unnormalized_posterior2 <- pars$likelihood * pars$prior2
pars$posterior2 <- pars$unnormalized_posterior2/sum(pars$unnormalized_posterior2)
```

I redo the plot

```
plot(pars$p,pars$posterior2,type='l',
xlab='probability of water',
ylab='posterior probability')
```



and now I get a jagged plot...I do not know why, maybe `my_likelihood` has some numeric problem (more on this later)? Let's pretend I can ignore the jaggedness and let's compute the 89% percentile and the HPDI intervals:

```
library(rethinking)
sample_indices <- sample(1:nrow(pars), size=1e4, replace = TRUE, prob=pars$posterior2)
samples <- pars[sample_indices,'p']
percentile_interval <- PI(samples,prob=.89)
percentile_interval
```

```
##        5%       94%
## 0.5035035 0.6306306
```

```
hpdi_interval <- HPDI(samples,prob=.89)
hpdi_interval
```

```
##     |0.89     0.89|
## 0.5005005 0.6036036
```

The width of the percentile interval is 0.1271271 and the width of the HPD interval is 0.1031031 and as I would have expected the percentile is wider than the HPD; I guess it is that because when the ditribution is not symmetrical (around the mode) but skewed then the HPD interval is always narrower than the percentile interval (but that should be proved as a theorem given the definitions for the percentile and the HPD).

If I had only the information in the interval I might be misleaded into thinking that the distribution was symmetrical.

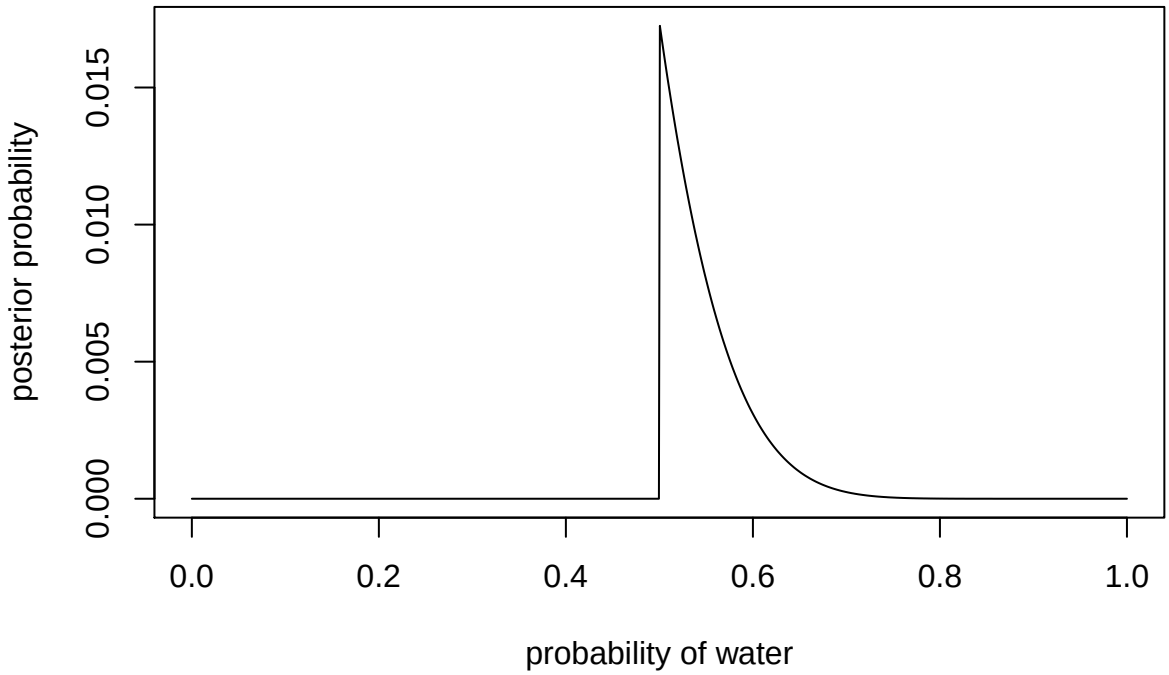
And now a little more about the jaggedness, now I will use the exact likelihood:

```
# exact likelihood
my_likelihood <- function(N_tosses,n_water,water_proportion)
{
    stopifnot(water_proportion>=0)
    stopifnot(water_proportion<=1)
    return(dbinom(n_water,N_tosses,prob=water_proportion))
}
```

```r
# grid
pars <- expand.grid(p=seq(from=0,to=1,length.out=1000))

# step prior
pars$prior3 <- step_prior(pars$p)
for (i in 1:nrow(pars)) {
  likelihoods <- my_likelihood(N_tosses,n_water,pars$p[i])
  pars$likelihood[i] <- prod(likelihoods)
}
pars$unnormalized_posterior3 <- pars$likelihood * pars$prior3
pars$posterior3 <- pars$unnormalized_posterior3/sum(pars$unnormalized_posterior3)

# plot posterior
plot(pars$p,pars$posterior3,type='l',
xlab='probability of water',
ylab='posterior probability')
```




```r
# intervals
sample_indices <- sample(1:nrow(pars), size=1e4, replace = TRUE, prob=pars$posterior3)
samples <- pars[sample_indices,'p']
percentile_interval <- PI(samples,prob=.89)
percentile_interval
```

```
##        5%       94%
## 0.5035035 0.6316316
```

```r
hpdi_interval <- HPDI(samples,prob=.89)
hpdi_interval
```

```
##     |0.89     0.89|
## 0.5005005 0.6056056
```

No jaggedness! And the intervals are almost equal to the ones of before. That's quite consolatory but it

is also a warning about the need of math knowledge. Is the jaggedness an issue related to the *simulation variance*? Maybe a larger `sz` lessen the jaggedness?