Statistical Inference: Uncertainty About One Parameter

Recap of Last Week

We tossed a beach ball around

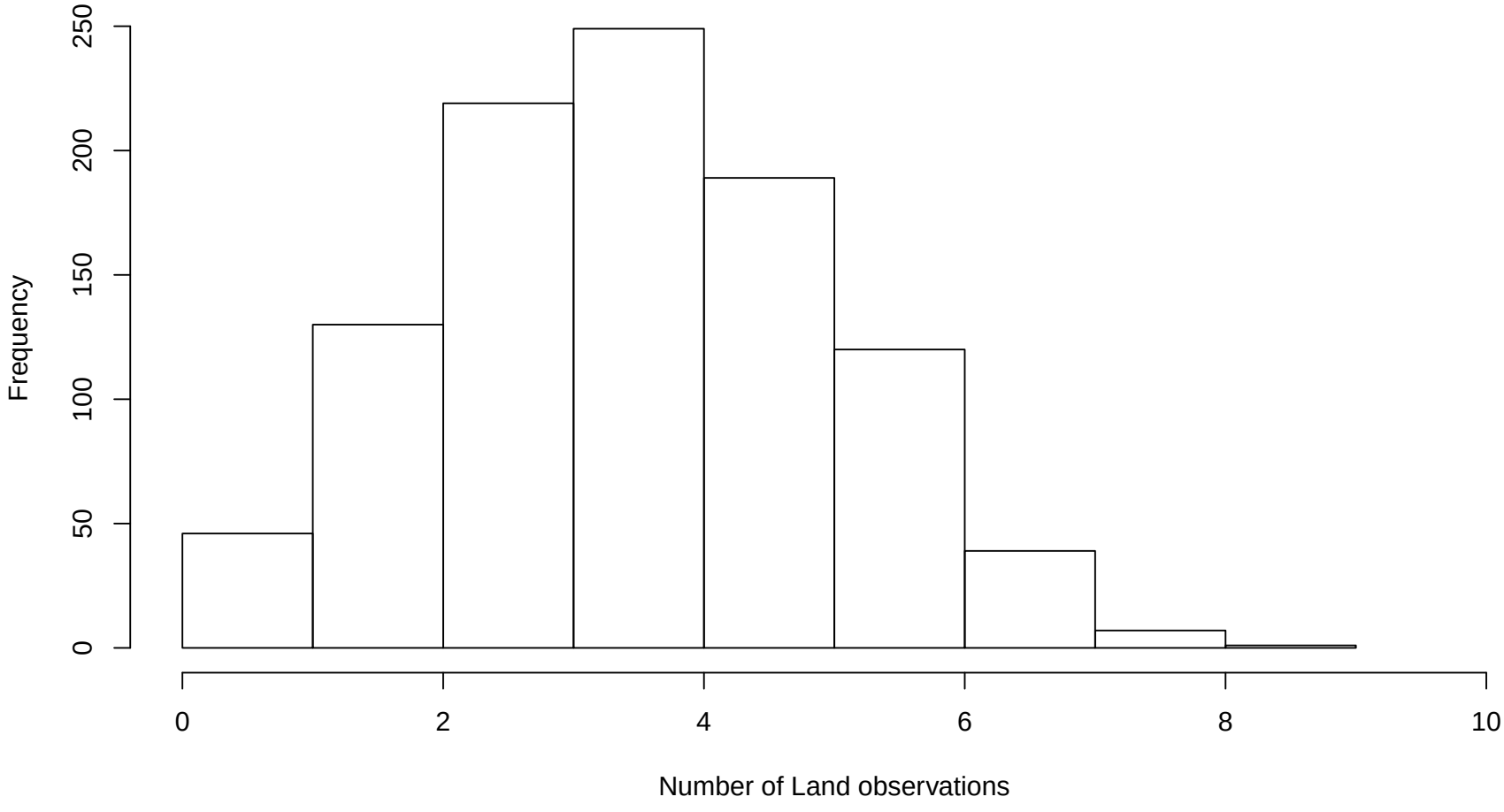
We saw land 4 times and sea 6

The second time we saw land 6 times and sea 4

We want to estimate the proportion of land

Recap of Last Week

We saw that there would be variation when we replicate the experiment



Recap of Last Week

We can estimate the proportion by

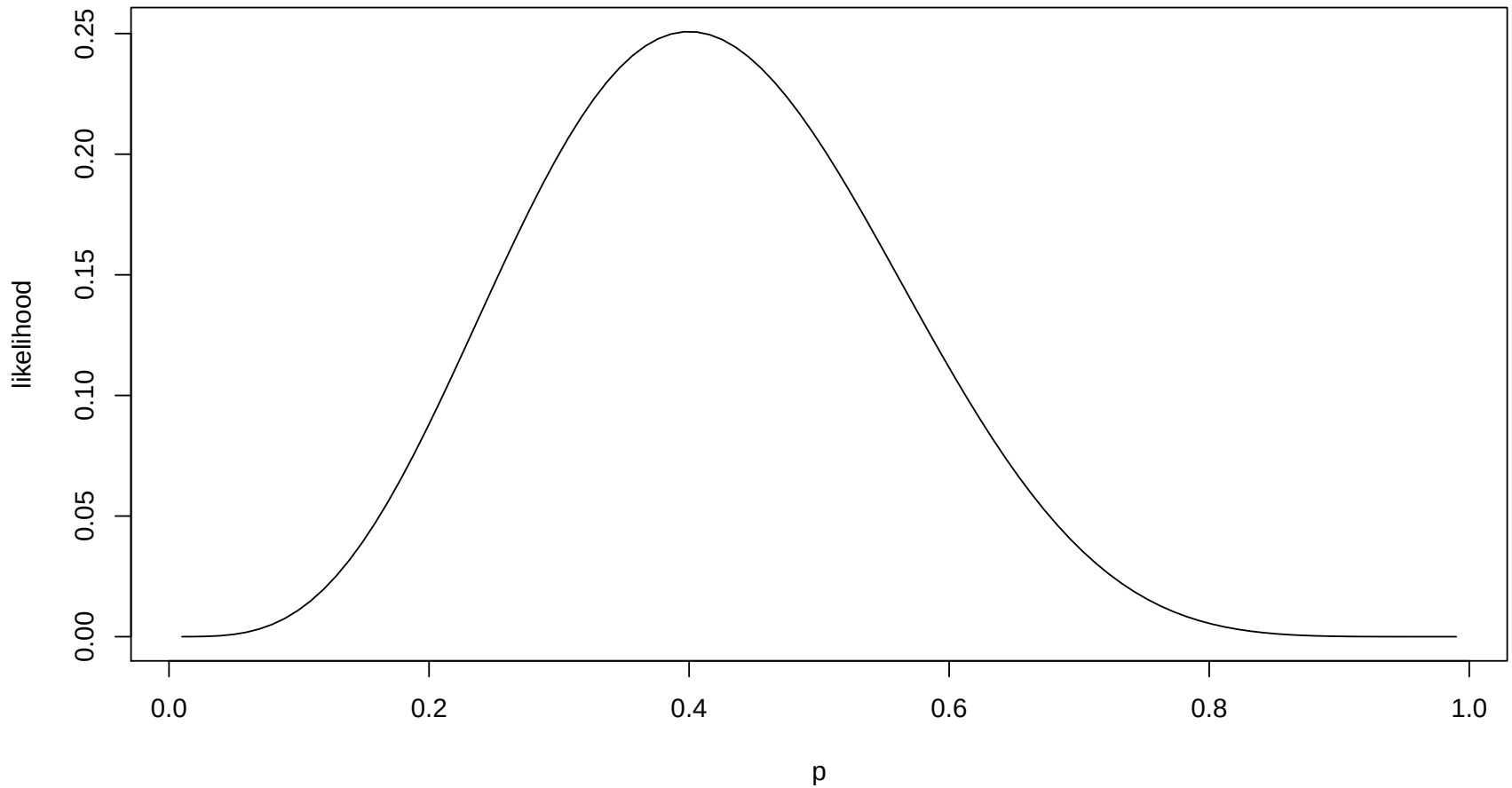
- building a model
- finding the parameters that are model likely to give the data

This is the *maximum likelihood estimate*

- maximise Pr(Data|parameters) w.r.t parameters

Recap of Last Week

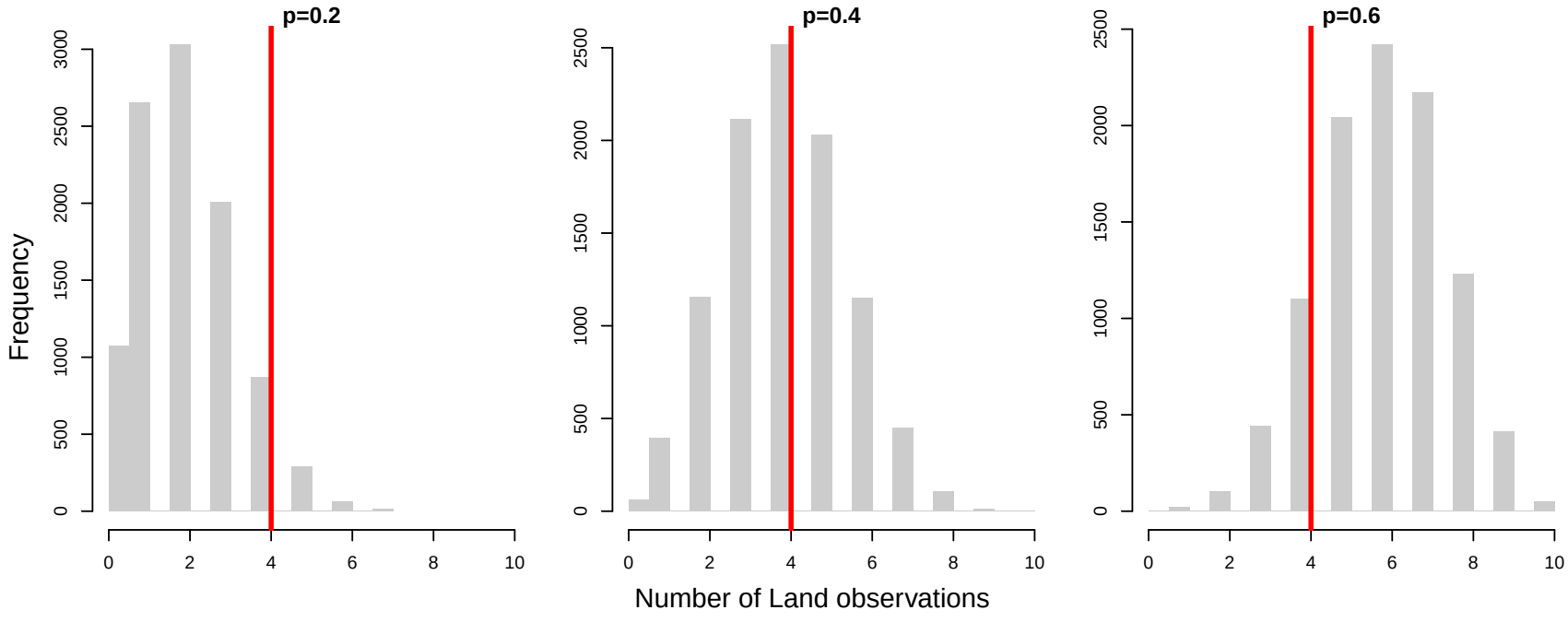
For this problem we can maximise the likelihood analytically



This week

How good is our estimate?

We saw last week that different values of p can give the same data



The Question

Because different samples give different estimates, we want to quantify this - suggest plausible values

What summaries could we use?

(What summaries do we use for simple statistics?)

Simulating the Sampling Distribution

From our data, we have our estimate of p (which we call $\hat{p}$)

If this is the true value, what values are we likely to estimate?

What to do

Simulate the data. For each simulation calculate $\hat{p}$, the m.l.e. of p.

Look at the histogram of the distribution

- code on next 2 slides

Simulations of the sampling distribution

This is the function we used last week

```
simGlobe <- function(probability=0.5, NTrials=10) {
  sim <- rbinom(NTrials, 1, probability)
  Res <- factor(c("Sea", "Land")[1+sim],
                levels = c("Sea", "Land"))
  return(table(Res))
}
```

We know that the MLE for p is r/N, e.g. Land/(Land + Sea), so we can calculate it from the simulations:

```
sim <- replicate(3, simGlobe(probability=0.4, NTrials=10))
# apply(, 2, ) loops over the columns
apply(sim, 2, function(x) x[2]/sum(x))
```

```
## [1] 0.5 0.3 0.3
```

All in one

But is might be easier to use a single function

```
simGlobeandEst <- function(prob=0.5, NTrials=10) {
  sim <- rbinom(NTrials, 1, prob)
  Res <- factor(c("Sea", "Land")[1+sim],
                levels = c("Sea", "Land"))
  p.hat <- mean(Res=="Land")
# = sum(Res=="Land")/length (Res)
  return(p.hat)
}
replicate(3, simGlobeandEst(prob=0.4, NTrials=10))
```

```
## [1] 0.5 0.3 0.3
```

(you will need to increase the number of replicates, of course)

How can we summarise the distribution?

Can we give a range of probable values?

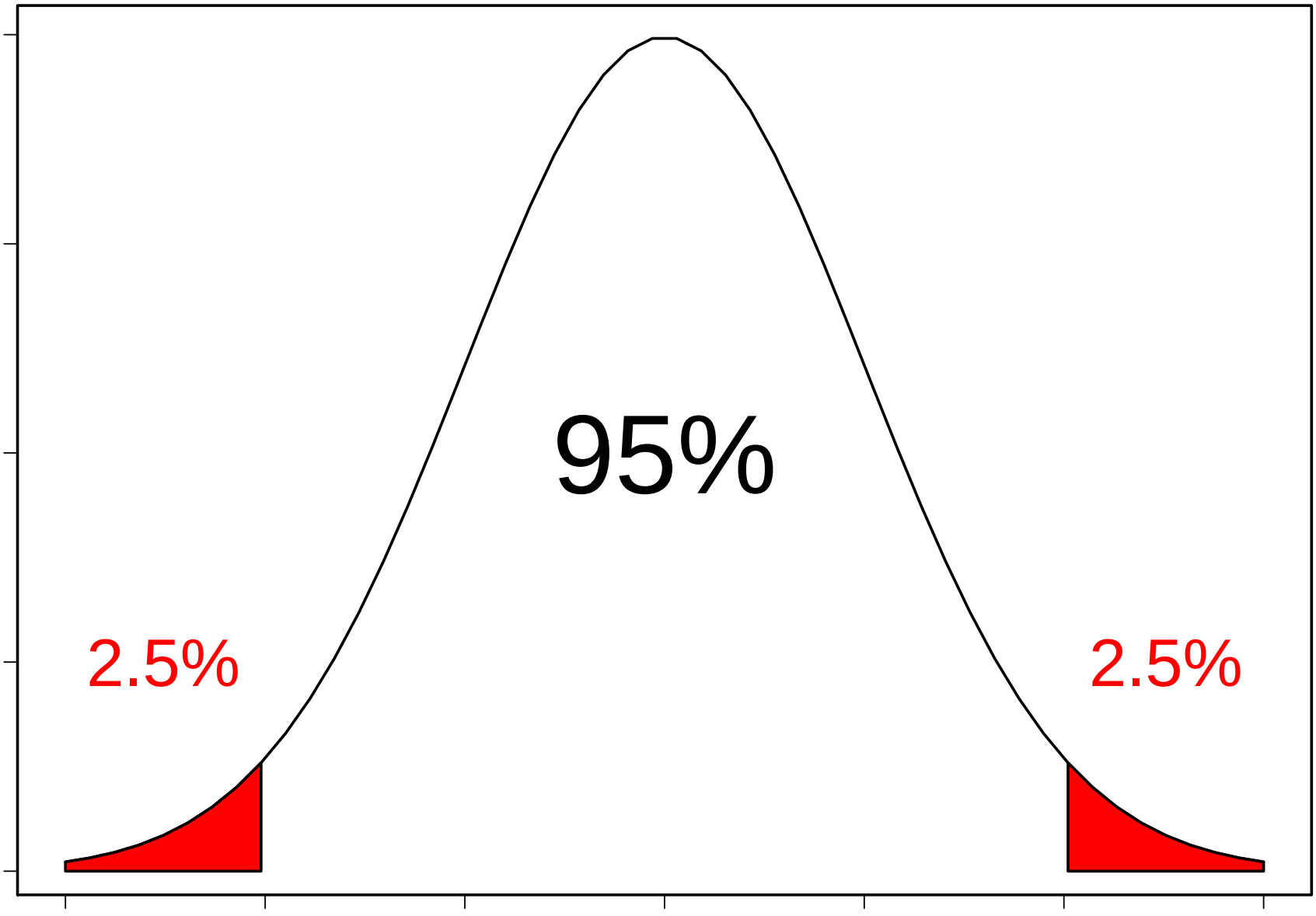
Confidence Intervals

We can give an interval within which we think we would see the sample statistic

- the confidence interval
- usually use 95%

Confidence Intervals

For continuous data the 95% confidence interval is constructed like this



Confidence Intervals

For discrete data it is a bit more difficult to get an exact interval.

Your task: try to calculate an approximate 95% confidence interval for your data

Confidence Intervals with more data

Now imagine that rather than 10 trials, you have 1000. As before, you see 40% of the observations are land (i.e. 400 out of 1000)

```
# 1e3 = 1x10^3 = 1000
replicate(3, simGlobeandEst(prob=0.4, NTrials=1e3))
```

```
## [1] 0.376 0.427 0.392
```

Try to find a 95% confidence interval for this

Basically, we want to remove the outer 2.5% of values, and see what is left.

Confidence Intervals and Tables

One way to create the 95% confidence intervals is to use a table (calculated as a %):

```
SimDist <- replicate(1e5, simGlobeandEst(prob=0.4,
                                         NTrials=10))
round(100*(Tab <- table(SimDist)/length(SimDist)), 1)
```

```
## SimDist
##    0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9    1
##  0.6  4.1 12.2 21.4 25.2 20.0 11.1  4.3  1.0  0.2  0.0
```

If we remove the outer values, we have 0.6% of the distribution outside, so 99.3 inside, which is too much.

If we remove the next two, values (i.e. 0.1 and 0.8) we have 95.2% inside, so this is approximately a 95% confidence interval.

Confidence Intervals and Quantiles I

Using a table is OK when there are only a few values. But if we have, say, 1000 trials we could have a table with 1000 values

A better way to do this is to sort the numbers from lowest to highest

```
SimDist1k <- replicate(1e3, simGlobeandEst(prob=0.4,
                                           NTrials=1000))
sort(SimDist1k)[1:10]
```

```
##  [1] 0.348 0.353 0.356 0.358 0.361 0.362 0.362 0.363 0.3
```

and then take the values that are 2.5% of the way from the bottom, and 2.5% of the way from the top:

```
sort(SimDist1k)[c(0.025*length(SimDist1k),
                  0.975*length(SimDist1k))]
```

```
## [1] 0.372 0.431
```

Confidence Intervals and Quantiles II

We can do this even more efficiently with quantiles.

A x% quantile is a values of a distribution with x% of the distribuition less than it

- a median is the 50% quantile
- the 25% and 75% quantiles are called quartiles (they plus the median split the data into 4 quarters)

So, we can just need the 2.5% and 97.5% quantiles. There is a function in R to do this:

```
quantile(SimDist1k, c(0.025, 0.975))
```

```
##  2.5% 97.5%
## 0.372 0.431
```

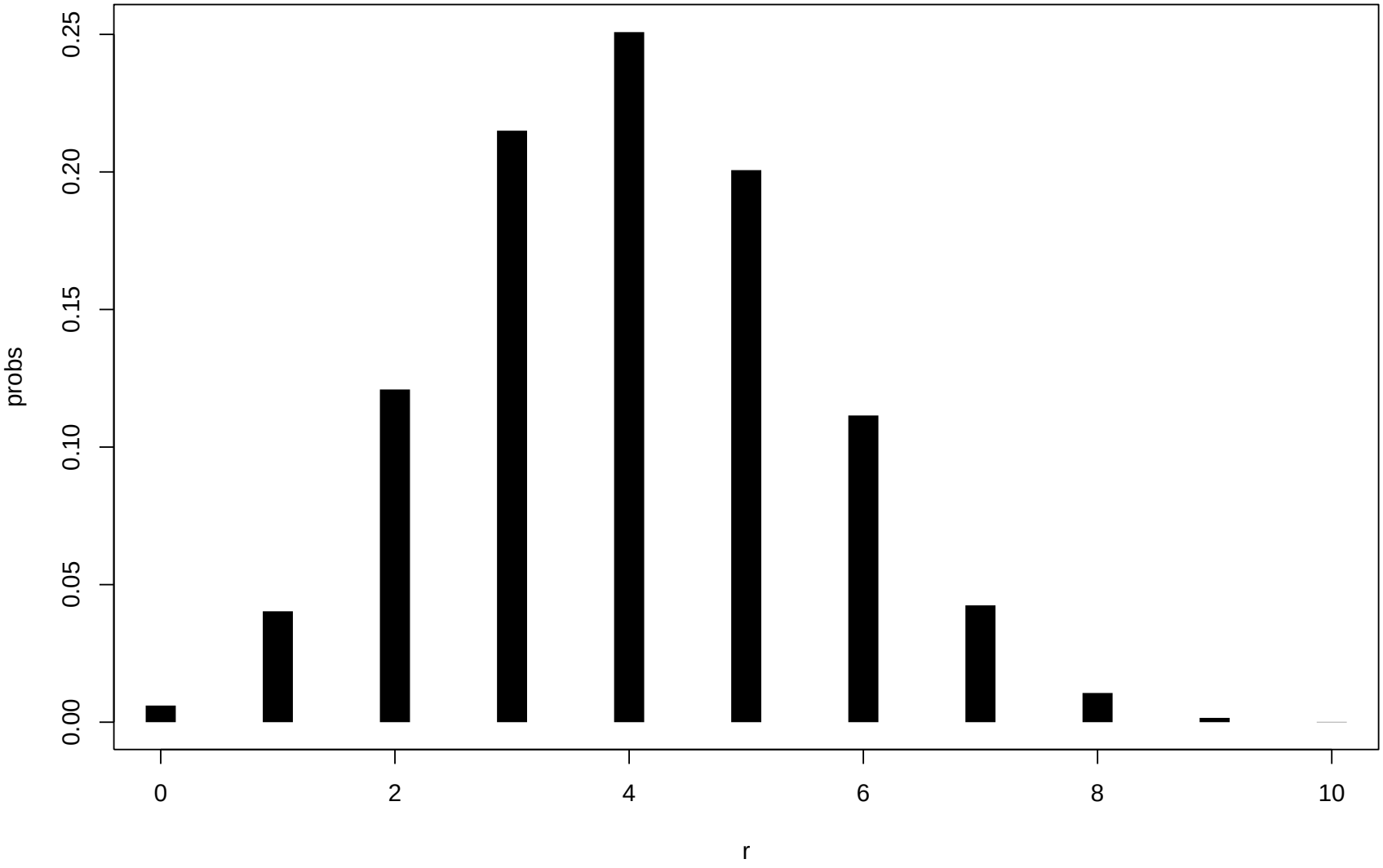
Confidence Intervals with more data

What are the differences in the confidence intervals?

- in their size
- in how well they cover 95% of the sampling distribution

More Exact Confidence Intervals

We are simulating the distributions of the estimates of $\hat{p}$, but we can calculate the exact probabilities



Your next task: different calculations of intervals

Calculate the more exact confidence intervals using `dbinom()`. This calculates $Pr(n = r)$, so you will have to sum over the correct values

Then look at `pbinom()`, which calculates $Pr(n \leq r)$ (e.g. `pbinom(0:10, 10, 0.4)`).

Then look at `qbinom()`, which calculates the value of `r` for which $Pr(n \leq r) = p$ (e.g. `qbinom(c(0.025, 0.975), 10, 0.4)`). This will make calculating confidence intervals easier

What, exactly, is a confidence interval?

Remember, our parameters are fixed, and our data are random.

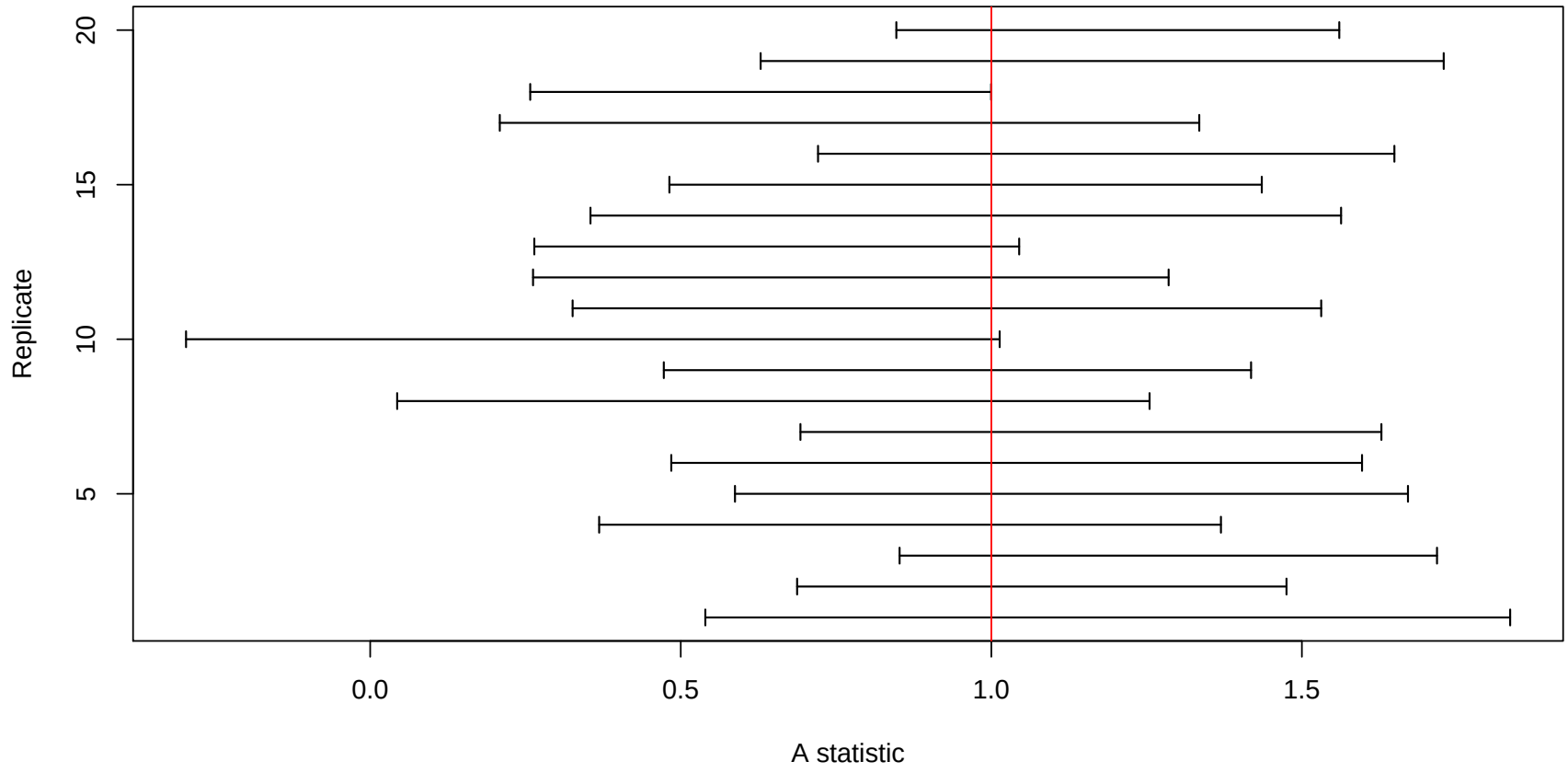
We calculate statistics from the data, so our statistics are random

Thus our confidence interval has to say something about the data (and statistics), not the parameter

- the estimator, not the estimand

OK, so what, exactly, is a confidence interval?

A confidence interval is an interval that will contain a population parameter a specified proportion of the time.



i.e. if we repeatedly sample the same population, 95% of confidence intervals will include the "true" parameter

Asymptotic Confidence Intervals

In statistics, large numbers usually make things much nicer: there are a lot of asymptotic results (i.e. approximations that work well when there is a lot of data).

One of these is the that most sampling distributions of statistics look like normal distributions, with enough data.

So, if we can construct a normal distribution's CI, we can make an approximation.

Normal Confidence Intervals

We can calculate a normal confidence interval like this:

```
c(qnorm(0.025, mu, sigma), qnorm(0.975, mu, sigma))
```

The parameters are the mean and standard deviation, e.g.

```
## [1] -5.839856  9.839856
```

Normal Approximations

If we know the mean and standard deviation of the sampling distribution, then we can use a normal approximation.

- the standard deviation of the sampling distribution is called the standard error

Normal Approximation for the Binomial

We can use the MLE, $\hat{p}$ as the mean of the normal

The standard error for the binomial distribution is

$$\frac{p(1-p)}{\sqrt{n}}$$

Your turn

Calculate the mean and standard error for binomial distributions where - we see 4 lands out of 10 - we see 400 lands out of 1000

Then calculate the (approximate) confidence intervals for each.

Compare the confidence intervals to the ones you calculated earlier. How similar are they? How does changing the sample size affect this?