Inference and simulation

Frank Edwards

School of Criminal Justice, Rutgers - Newark

What is simulation?

Statistical simluation is a flexible tool that uses computers to take random draws from probability distributions to learn about features of random variables and their relationships.

Rather than relying on exact mathematical solutions, we can use computers to *brute force* an approximate solution by repeating an experiment a large number of times.

Simulation applications

1) Approximation of the sampling distribution of real-world data through generative models
2) Evaluating impacts of assumptions
3) Prediction and inference

The sampling model and population inference

Under the **sampling model** we use a subset (or sample) to **infer** characteristics about a population.

All data (that aren't a full population) represent a sample. Our data represent one possible outcome of many.

The full set of possibilities and their probabilities is called the *sampling distribution*. We use our data and properties of its sampling distribution to learn about (unknown) population parameters.

Simulation basics

Let's assume that undergrads taking a criminology course score an average of 70 points on an exam. What kinds of outcomes could we expect to see for a classes scores?

We can *simulate* trials of giving the exam by assuming that a student's grade y is randomly distributed, here we'll use the Normal distribution.

$y_i \sim \mathrm{N}(\mu = 70, \sigma = 10)$

Simulating in R

```r
### define a function to draw samples
give_exam <- function(n) {
    exam_grades <- rnorm(n, mean = 70, sd = 10)
    # use return() to produce output from a function
    return(exam_grades)
}

### Simulate a class of 20 students
class1 <- give_exam(n = 20)
class1
```

```
##  [1] 64.40774 62.15134 78.44804 63.64354 66.73441 72.62718 65.27635 75.90916
##  [9] 72.79042 72.70882 76.01089 64.24594 83.10778 55.87107 63.97269 80.88813
## [17] 80.22564 76.17495 44.85225 70.37321
```

```r
round(mean(class1), 1)
```

```
## [1] 69.5
```

Statistical inference for a mean

We've observed $\bar{y}$. In this case, we know the 'true' mean μ, but that's almost never the case in the real world.

```
mean(class1)
```

```
## [1] 69.52098
```

We generally have to use an observed $\bar{y}$ to try to learn something about μ, which is not observed.

This single simulation draw represents only one possible realization of $\bar{y}$ of many (infinite).

Describing uncertainty in our inference

We could have observed many possible samples of distributions of grades

```
## map is a tidyverse version of replicate() this generates 30 classes with 20
## student exam grades
class30 <- map(rep(20, 30), give_exam)
## here's two of our 9
class30[[2]]
```

```
##  [1] 69.69380 73.48258 63.71843 75.52911 72.99329 63.22241 64.95386 74.56343
##  [9] 71.79015 67.20654 67.79923 83.88718 75.78619 60.51848 56.34144 64.27987
## [17] 67.72973 62.78014 74.74162 68.09735
```

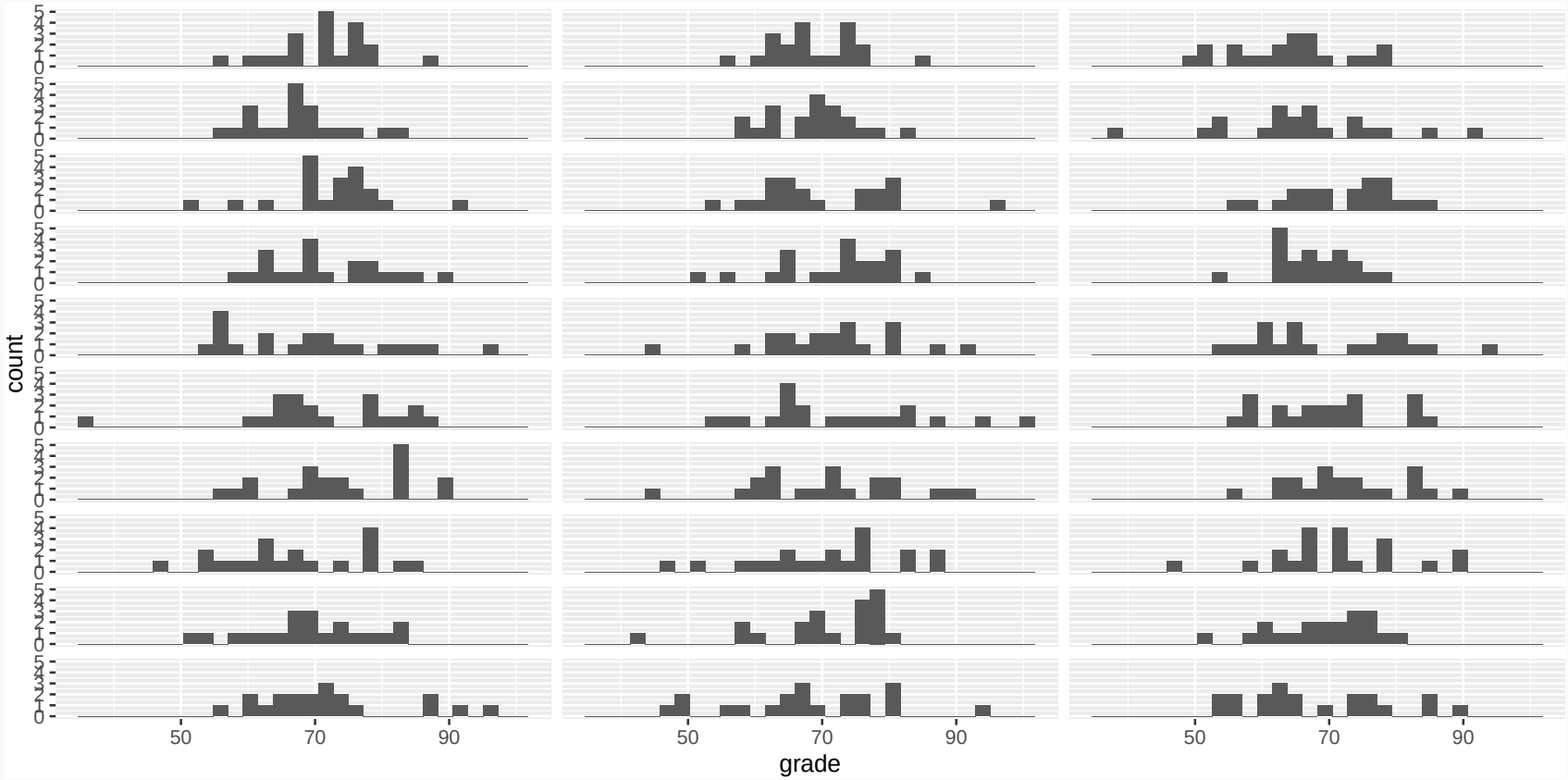
```
class30[[7]]
```

```
##  [1] 51.79099 57.65728 81.62022 75.94543 75.30291 75.48186 63.37693 90.61633
##  [9] 73.65359 68.93125 73.30743 68.55772 71.23561 68.59226 69.84009 76.19648
## [17] 78.89270 74.87046 69.59540 77.73545
```

Implications of sampling variability

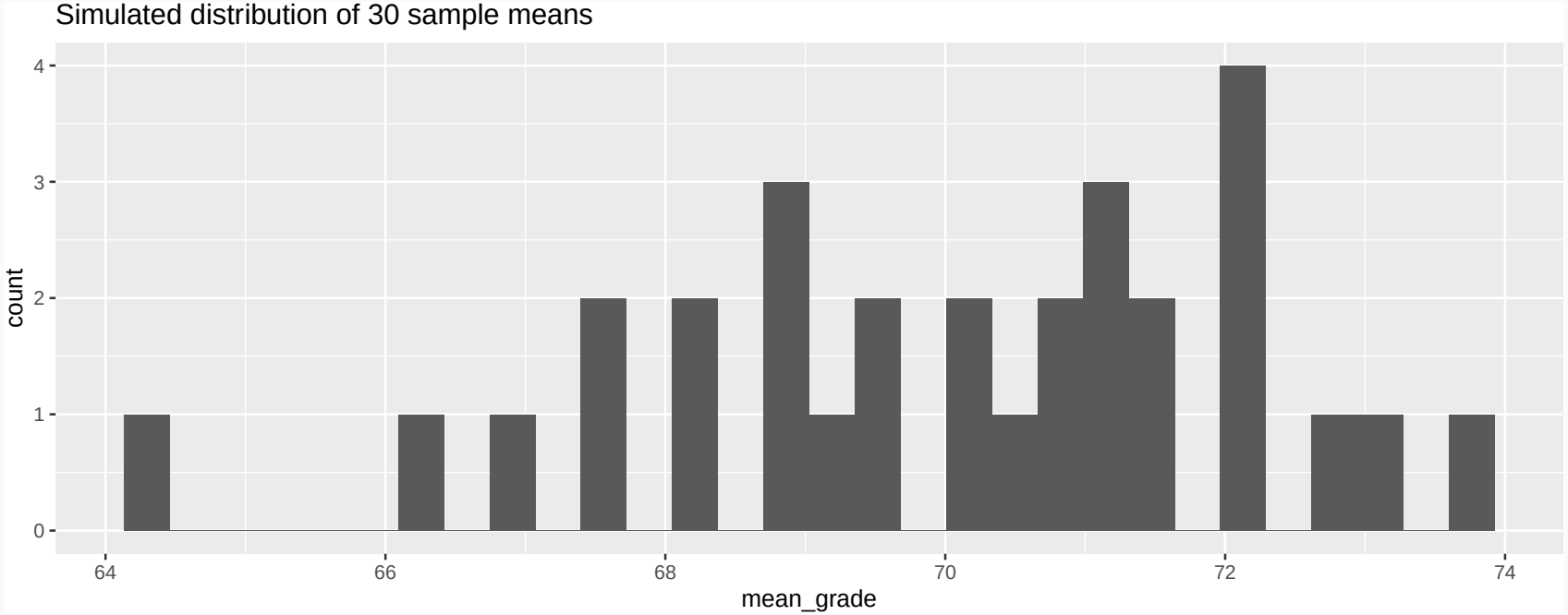
Each classroom could have one of a potentially infinite set of distributions. Here are 30

What do you notice?



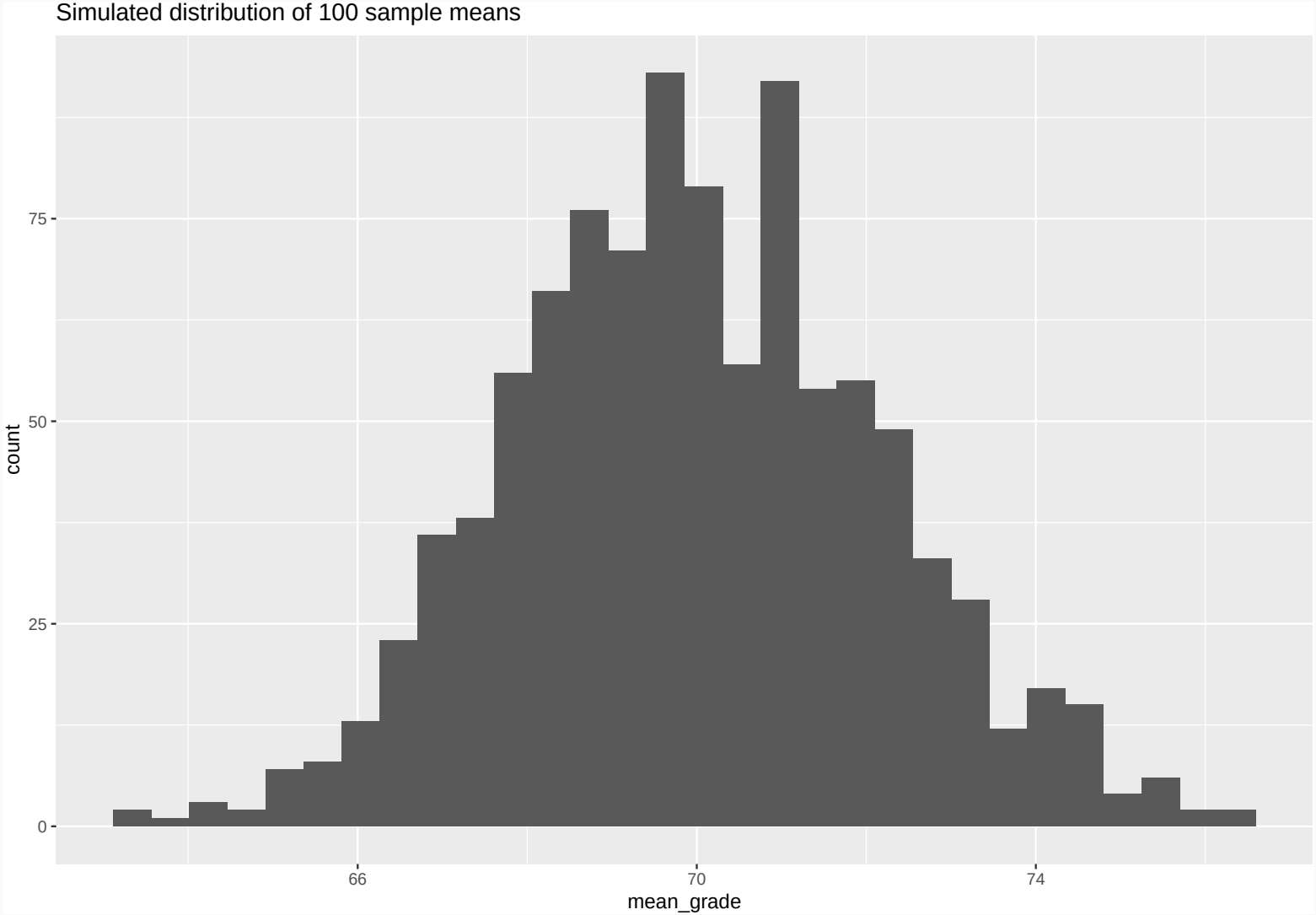
The sampling distribution of a parameter

Just as our sample has a theoretical sampling distribution, our estimate of the sample mean $\bar{y}$ has a sampling distribution.



For a large number of trials

Let's see what the distribution of $\bar{y}$ looks like if we sample 1000 classrooms



Constructing a parameter estimate from a sampling distribution estimate

The *central limit theorem* tells us that

$$\lim_{n \to \infty} \bar{y} \sim \text{Normal}(\mu, \sigma)$$

The logic of frequentist inference

Given that we know that the sampling distribution of $\bar{y}$ is Normal with mean μ for large N, we can use our data to *approximate* this sampling distribution.

We compute the sample mean ($\bar{y}$) and the *standard error* of the sample mean ($sd_y/\sqrt{n}$) to describe this distribution.

```
class1  # the sample (x)
```

```
##  [1] 64.40774 62.15134 78.44804 63.64354 66.73441 72.62718 65.27635 75.90916
##  [9] 72.79042 72.70882 76.01089 64.24594 83.10778 55.87107 63.97269 80.88813
## [17] 80.22564 76.17495 44.85225 70.37321
```

```
mean(class1)  # xbar
```

```
## [1] 69.52098
```

```
sd(class1)/sqrt(length(class1))  # s_x
```

```
## [1] 2.09057
```

Visualizing the sampling distribution of sample means

We can describe our uncertainty in the location of the mean with the approximated sampling distribution estimated from the data.

Visualizing the sampling distribution of sample means

We can describe our uncertainty in the location of the mean with the approximated sampling distribution estimated from the data.

We use these estimates to describe the approximate range of our uncertainty in the value of the *test statistic* $\bar{y}$.

Visualizing the sampling distribution of sample means

We can describe our uncertainty in the location of the mean with the approximated sampling distribution estimated from the data.

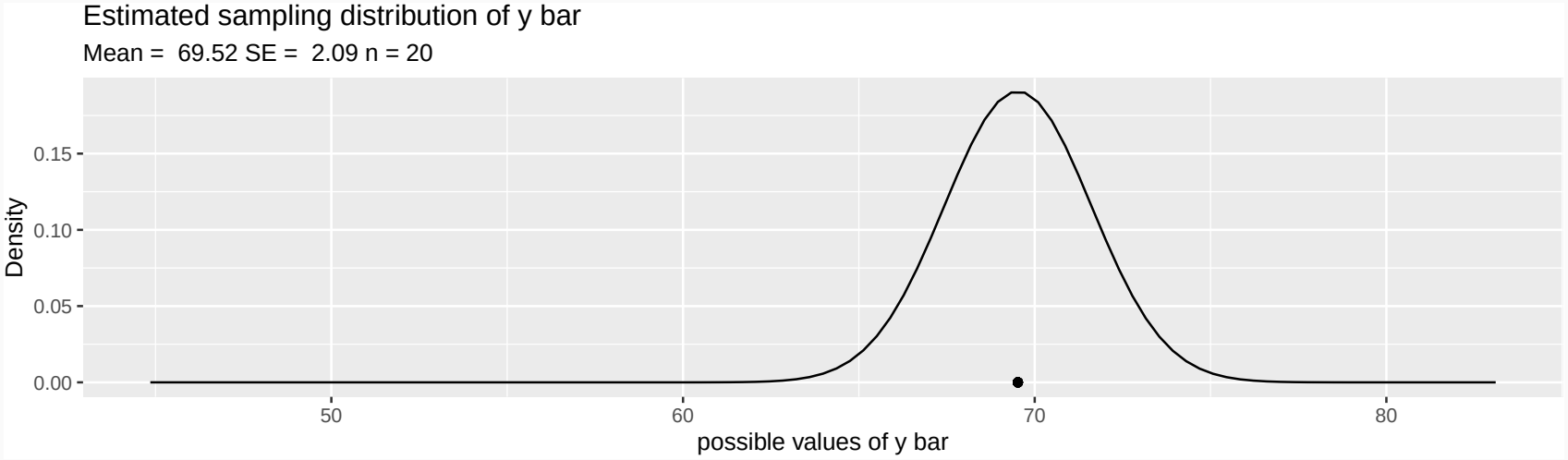
We use these estimates to describe the approximate range of our uncertainty in the value of the *test statistic* $\bar{y}$.

Under what conditions do we directly learn about μ?

Question

Using this estimated sampling distribution, compute a 95 percent confidence interval for $\bar{y}$.

Hint: you can use `pnorm(0.025, 0, 1)` and `pnorm(0.975, 0, 1)` to obtain critical values for z.



Review

1. What is a parameter?

Review

1. What is a parameter?
2. What is the difference between $\bar{x}$ and μ?

Review

1. What is a parameter?
2. What is the difference between $\bar{x}$ and μ?
3. What is the difference between a sample and a sampling distribution?

Review

1. What is a parameter?
2. What is the difference between $\bar{x}$ and μ?
3. What is the difference between a sample and a sampling distribution?
4. Briefly explain the logic of a confidence interval through the logic of a sampling distribution

Confidence intervals and sampling distributions

1. Let's draw 50 classrooms with 20 students each

```
# set variables
classrooms <- 50
students <- 20
# create empty list for storage, they can grow
score_out <- list()
# generate simulated classes with a for loop this could be done with map() or
# replicate() as well
for (i in 1:50) {
    scores <- data.frame(sample_n = i, score = rnorm(n = students, mean = 70, sd = 10))
    score_out[[i]] <- scores
}
# force list with elements of identical structure into data.frame
samp_dat <- bind_rows(score_out)
```

Confidence intervals and sampling distributions

1. Let's draw 50 classrooms with 20 students each
2. Let's compute 95 percent confidence intervals for $\bar{y}$ for each sample

```
samp_ci <- samp_dat %>%
    group_by(sample_n) %>%
    summarise(ybarhat = mean(score), se = sd(score)/sqrt(students)) %>%
    mutate(ci_lwr = ybarhat - 1.96 * se, ci_upr = ybarhat + 1.96 * se)
```

Confidence intervals and sampling distributions

1. Let's draw 50 samples with 100 households sampled
2. Let's compute 95 percent confidence intervals for $\bar{y}$ for each sample
3. Let's add a binary variable indicating whether the interval includes μ (70)

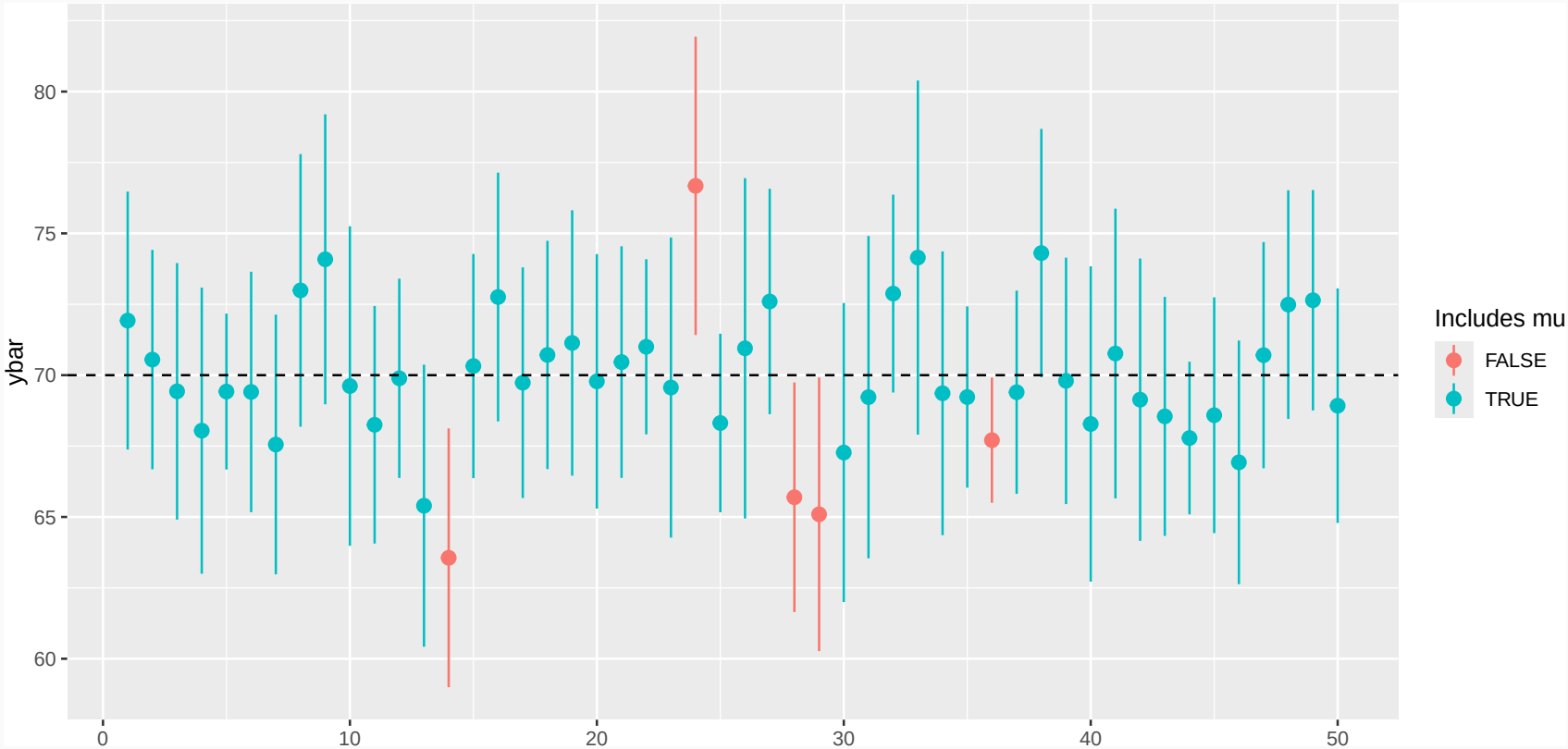
```
samp_ci <- samp_ci %>%
    mutate(sig_test.95 = ci_lwr < 70 & ci_upr > 70)
```

Confidence intervals and sampling distributions

1. Let's draw 50 samples with 100 households sampled
2. Let's compute 95 percent confidence intervals for $\bar{x}$ for each sample
3. Let's add a binary variable indicating whether the interval includes μ (70)
4. Plot it!

Visualizing CI coverage

```
ggplot(samp_ci, aes(ymin = ci_lwr, ymax = ci_upr, y = ybarhat, x = sample_n, color = sig_test.95)) +
    geom_pointrange() + geom_hline(yintercept = 70, lty = 2) + labs(x = "", y = "ybar",
    color = "Includes mu")
```



Summary

Confidence intervals give you a crude sense of the magnitude of variability in the sampling distribution of a parameter. For a critical value of 0.05 (a 95 percent interval), 95 percent of estimated intervals will cover μ. We have no guarantee that our estimated interval covers μ!

Simulation helps us see how our inferences work and evaluate different features of the data generating process.