Session 7: Sampling Bias

Mark Buntaine

Outline & Goals

1. Simulation for statistical reasoning
2. Samples vs. populations
3. Sampling distributions
   - realization vs. expectation
   - standard deviation vs. standard error
4. Sources of sampling bias
   - sample / population mismatch
   - response bias
5. Declaring populations and sampling in code (R)

Simulation for statistical reasoning

- One of the best ways to gain an intuition about populations, samples, bias, etc. is to simulate data and examine its properties
  - By simulating data, you are also forced to be explicit about the assumptions in your measurement approach
  - By working with simulated data, you also have the chance to try out different approach to analysis
  - This is especially important with *prospective evaluations*, where designs are put forward to collect data before it is available
- We're going to simulate data and sampling designs in *R*.

R Preliminaries

Make sure your R and RStudio are up-to-date. Then install the required packages.

```
#install.packages("DeclareDesign", "ggplot2")
```

Load the required packages.

```
library(DeclareDesign)
library(knitr)
library(ggplot2)
library(grid)
library(gridExtra)
library(dplyr)
library(kableExtra)
```

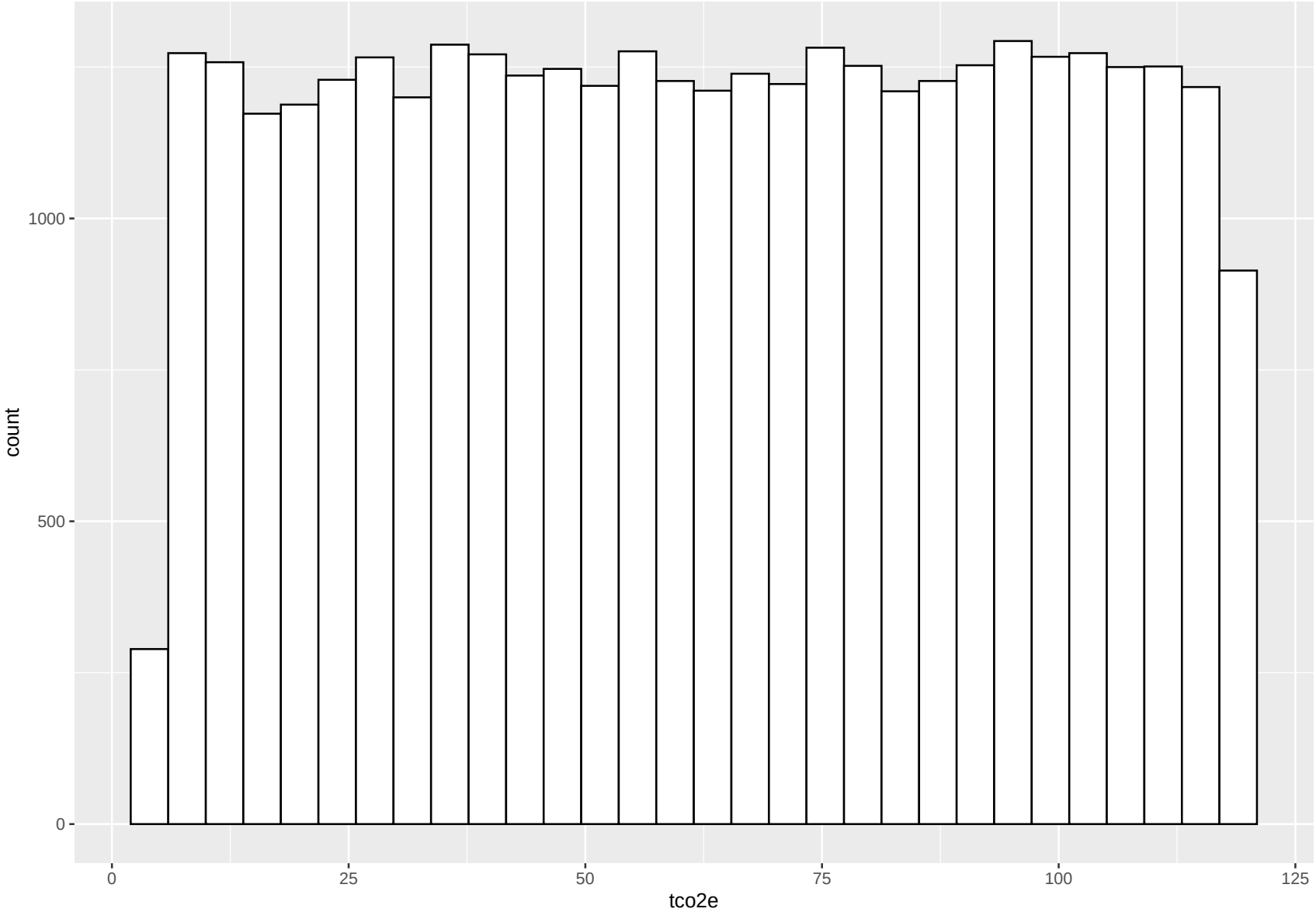
Toy example w/ single item

Carbon footprint of Santa Barbara county households:

- declare_population() allows you to declare the assumed characteristics of the population that you want to study.

```
set.seed(228)
population <- declare_population(
  households = add_level(N=36000,
     tco2e=runif(n=N, min=5, max=120))
)
pop <- population()
plot <- ggplot(pop, aes(x=tco2e)) +
  geom_histogram(color="black", fill="white")
```

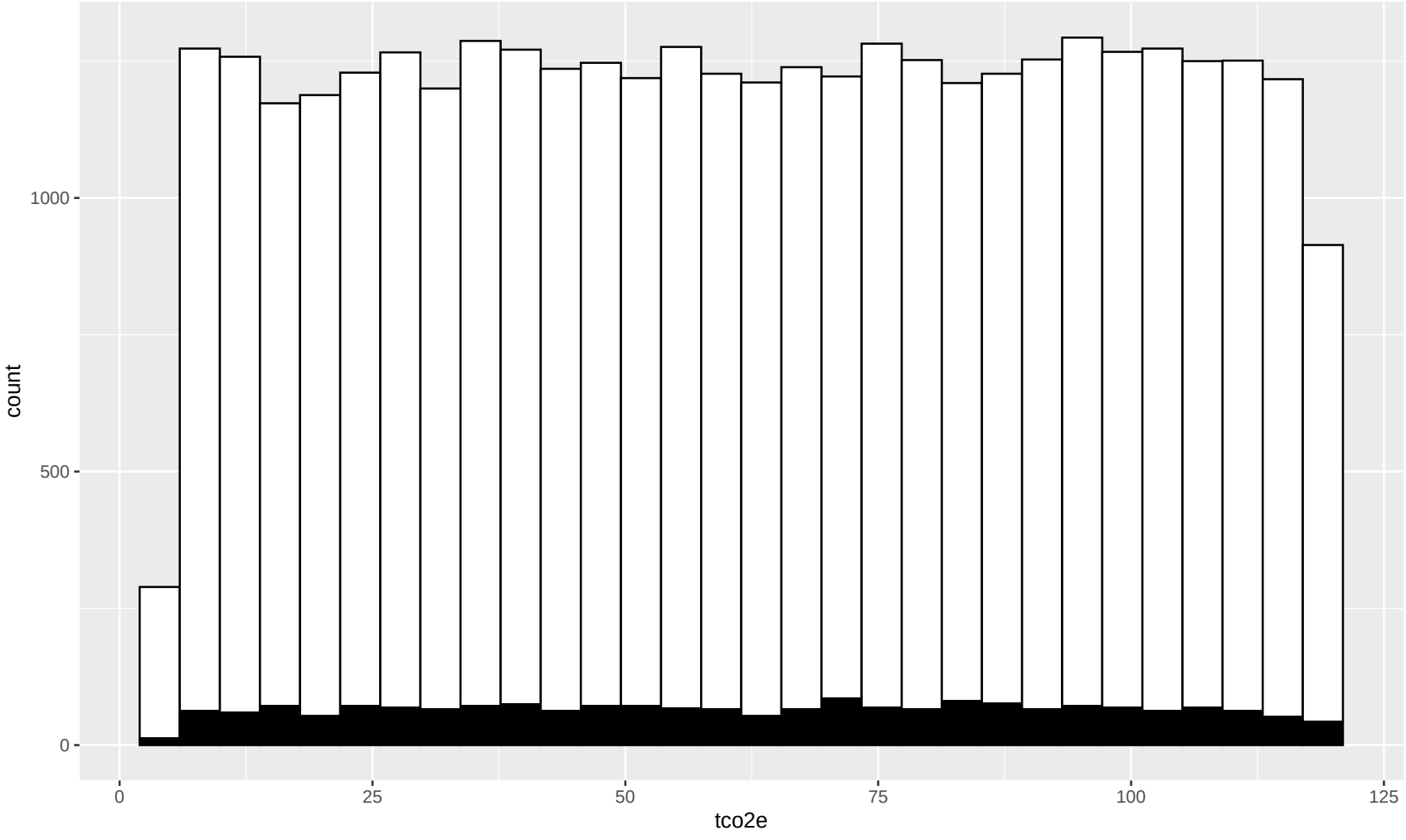
Toy example w/ single item



Drawing a sample

```
sam <- sample(1:36000,2000)
plot2 <- ggplot(pop, aes(x=tco2e)) +
  geom_histogram(color="black", fill="white") +
  geom_histogram(data=pop[sam,], fill = "black")
```

Drawing a sample



- We're hoping that the sample (black) can say something meaningful about the population (white)

Samples vs. populations

- *Population*: the complete set of units about which we intend to draw inferences
- *Sample*: the set of units that we are able to collect data about

We work with samples because it is almost never feasible to collect data about all units of interest.

We always evaluate our sampling design with reference to a population.

Realizations vs. expectations

Let's say we're interested in the mean household carbon footprint in Santa Barbara:

```
mean(pop$tco2e)
```

```
## [1] 62.61263
```

```
mean(pop[sam,"tco2e"])
```

```
## [1] 62.38949
```

Why do these quantities differ?

Realizations vs. expectations

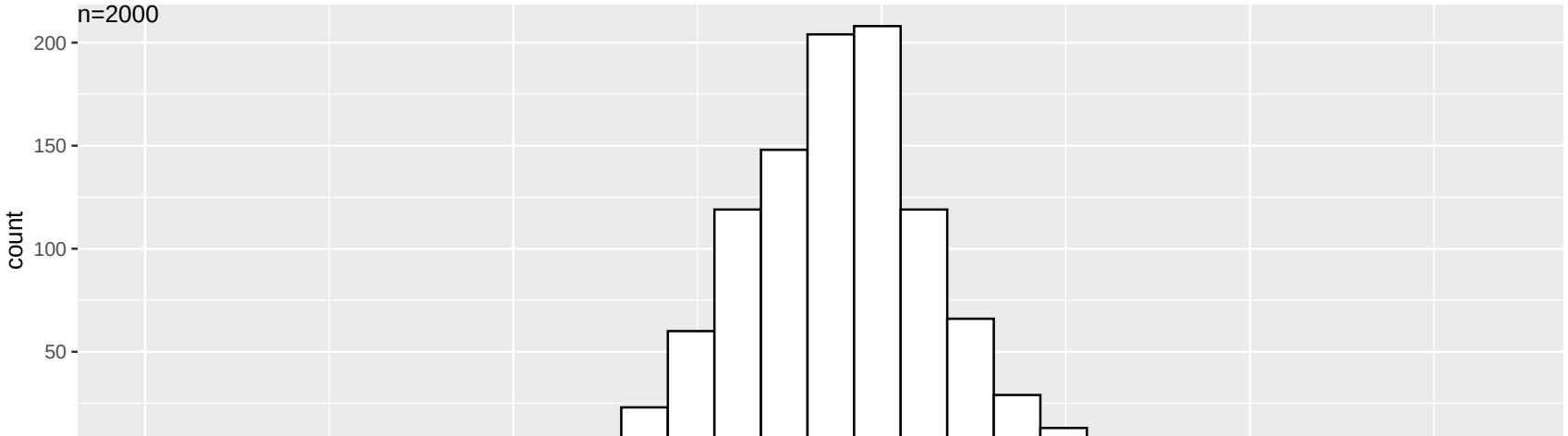
- Even when we draw a true representative sample, we can expect variation in the sample value across repeated draws.
  - The uncertainty created by *sampling variation* influences the degree to which we can be certain about our conclusions.
  - We typically want to choose sample sizes to keep sampling variation managable, given inferential goals.
- When you hear a polling result with a stated *margin of error*, that error is many comprised of expected sampling variation.
- Let's see:

Sampling distribution

- **Sampling distribution**: the distribution of sample values with a repeated draw of a given sampling frame.
- **Sampling frame**: this procedure describing the sample to be drawn.

```
sims <- 1000
store <- rep(NA, sims)
for (i in 1:sims){
  store[i] <- mean(pop[sample(1:36000,2000),"tco2e"])
}
```

##Sampling distribution



Standard deviation vs. standard error

```
sd(pop[sam,"tco2e"]) #standard deviation of sample
```

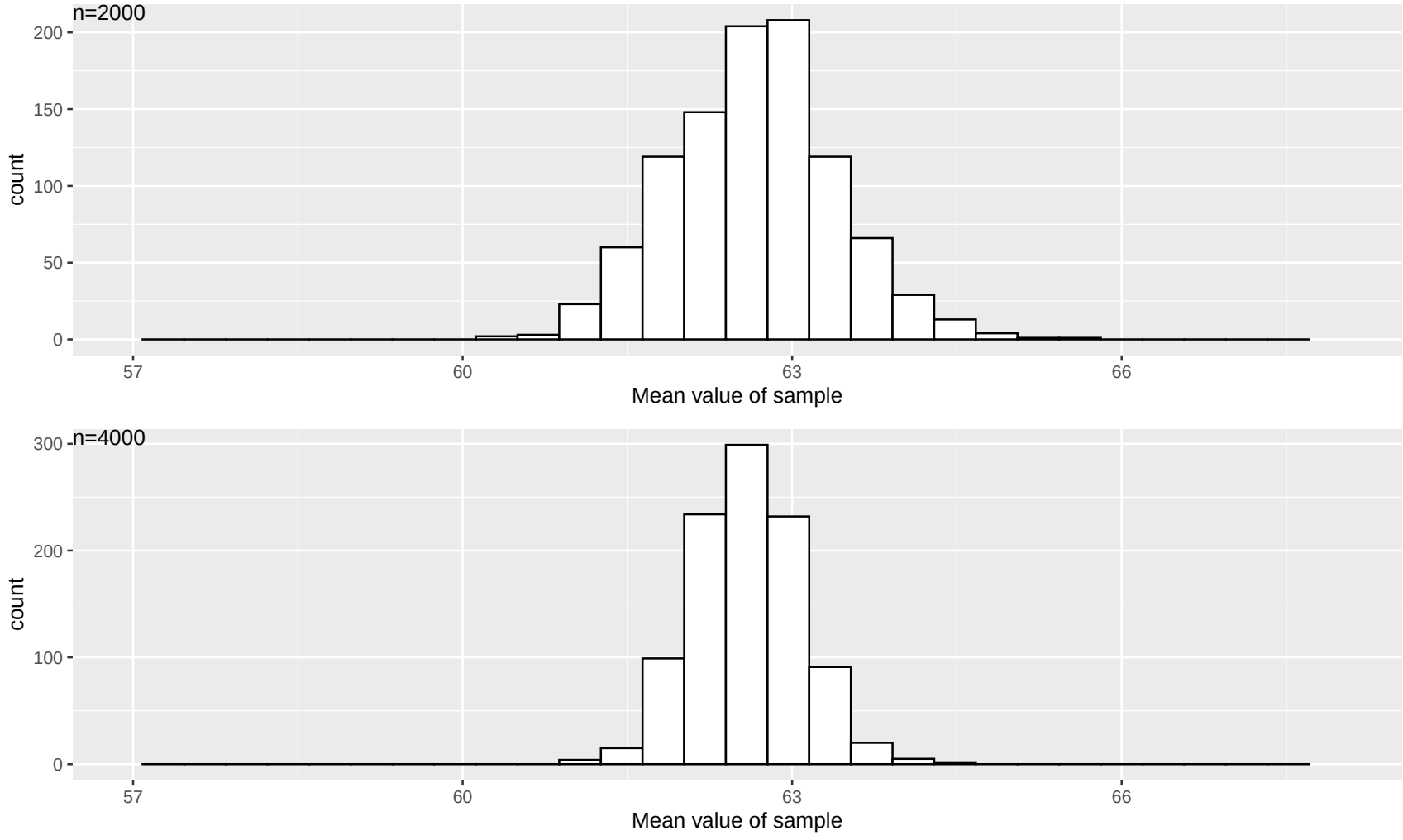
```
## [1] 32.43769
```

```
sd(store) #standard error of sample
```

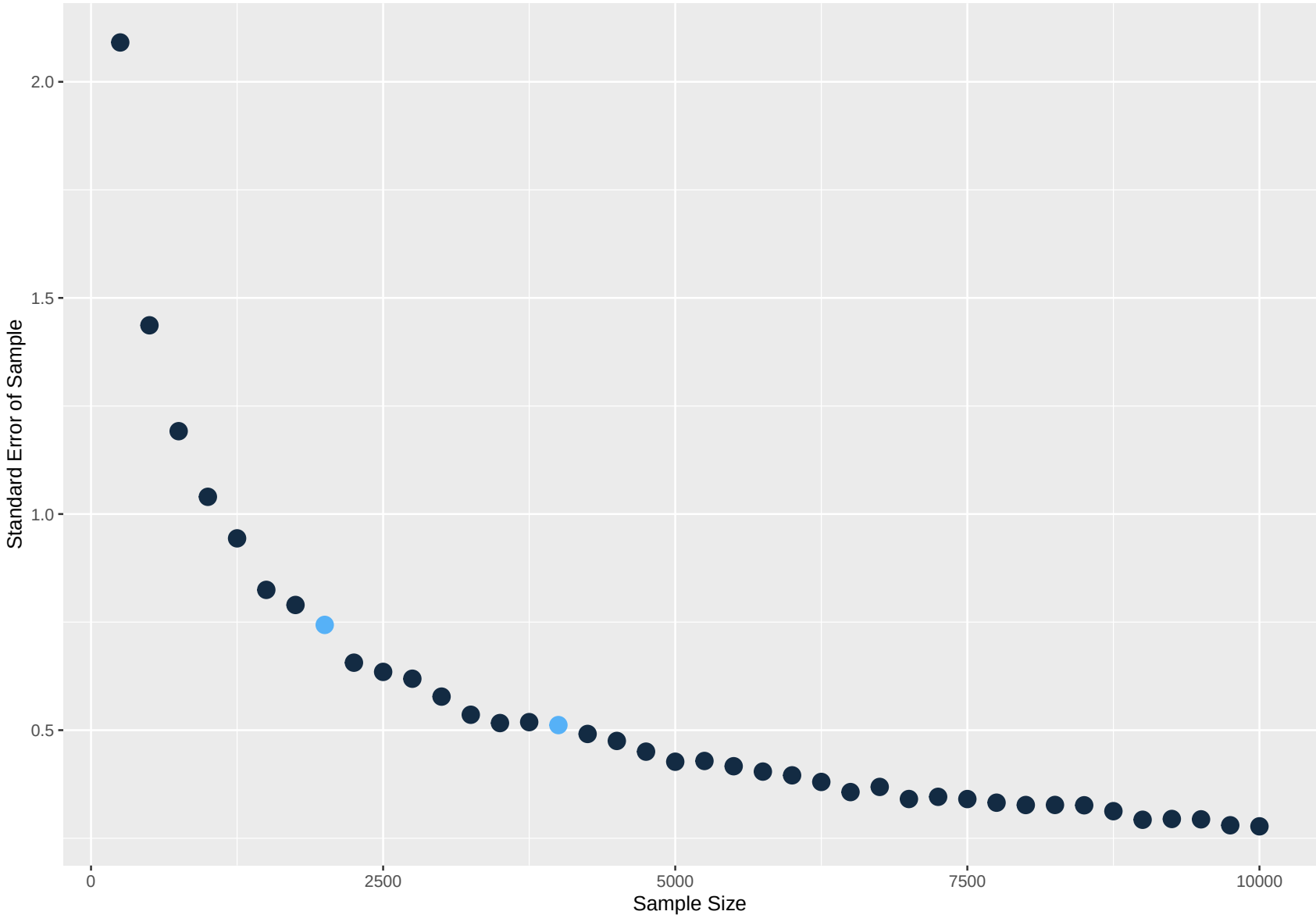
```
## [1] 0.7432648
```

- *Standard deviation* of a sample describes the variance in the data ($\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$)
- *Standard error* of a sample describes the expected sampling variance based on the sampling frame over repeated draws

Standard error and sample size



Standard error and sample size



Sampling bias

So far we've assumed that we can take a random ("representative") sample from the population and then examined the properties of those samples. In practice, it is often difficult to take a random sample from our target population, which leads to sampling bias.

- **Sampling bias** is the difference between the true value of the population parameter we are trying to discover and the *expected value* of that parameter based on the sampling procedure.
  - Sampling bias is **not** the difference between the true value of the population parameter and the realized value in a sample.
  - Sampling procedures that deviate from a random sample cause sampling bias.
- There are two main sources of sampling bias we will discuss:
  - Population / sample mismatches
  - Reporting bias

Population / Sample Mismatches

- This occurs when your sampling frame does not match your target population. Some examples:

Population	Sample
Likely voters	Voters with landline telephones
Water users	Single family households
Households	Households on main road
Fishers	Commercial fishers who use certain port

- This matters when the outcome covaries with sample frame exclusion criteria

Mismatches: An Example

Declaring a population: an example

```
set.seed(228)
population <- declare_population(
  households = add_level(N=500,
     main=draw_binary(N=N, prob = 0.5),
     satisfied=correlate(given = main, rho = 0.5,
                         draw_binary, prob = 0.5)
))
pop <- population()

kable(table(pop$main,pop$satisfied)) %>%
  add_header_above(c("main"=1,"satisfied"=2))
```

main	satisfied	
	0	1
0	170	81
1	89	160

Consequences of sampling procedures

```
mean(pop$satisfied) #target population parameter
```

```
## [1] 0.482
```

```
mean(pop %>% filter(main==1) %>% pull(satisfied))
```

```
## [1] 0.6425703
```

The difference between these two values is **bias**, not sampling variability.

- Look for any part of the population systematically excluded from the sample.
- Change interpretation to match sample actually drawn.

Population / Sample Mismatches

- Some examples:

Population	Sample
Likely voters	Voters with landline telephones
Water users	Single family households
Households	Households on main road
Fishers	Commercial fishers who use certain port

Response bias

Response bias is the difference between the true parameter of interest and the expected sample value of the parameter based on unequal probabilities of reporting.

- Often times harder to address than sample-population mismatches
- Can create large errors in measurement if not managed carefully

Let's continue with the previous example and assume:

1. We now take a random sample of all households by knocking on doors
2. If you live on the main street the chance that you are home is 50%
3. If you live on the side street the chance that you are home is 20%

Declaring response bias

```
reporting <- declare_assignment(blocks=main,
                    assignment_variable = "R",
                    block_prob=c(0.2,0.5))
pop <- reporting(pop)
kable(pop[1:6,])
```

households	main	satisfied	R	R_cond_prob
001	0	1	0	0.8
002	1	0	1	0.5
003	1	1	0	0.5
004	0	0	1	0.2
005	0	1	0	0.8
006	0	0	0	0.8

Declaring response bias

```
table(pop$main,pop$R)
```

```
##
##         0   1
##    0 201  50
##    1 125 124
```

Examining sample characteristics

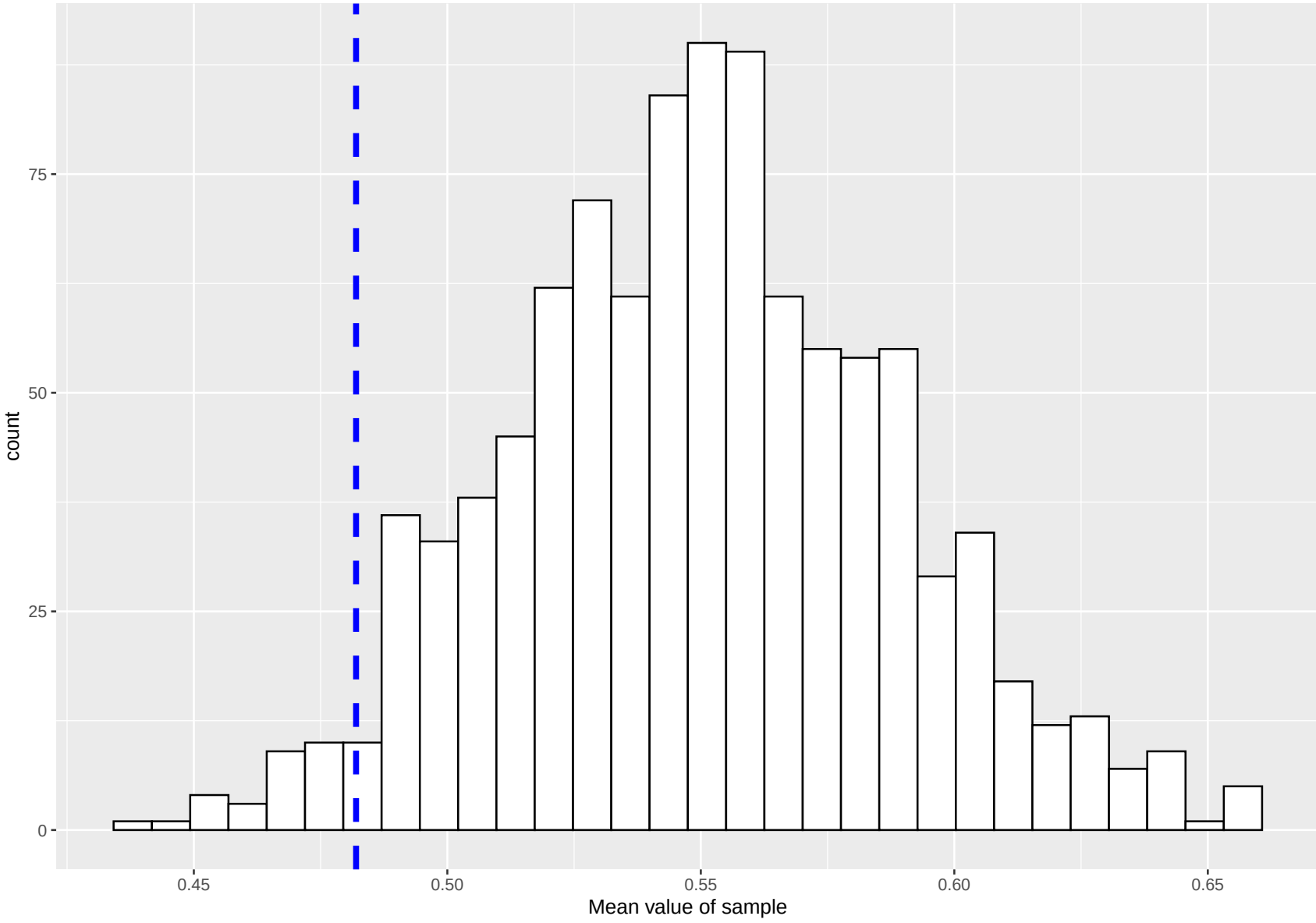
```
sims <- 1000 #simulations
sam.n <- 250 #attempted sample size

store <- rep(NA, sims)
for (i in 1:sims){
  store[i] <- mean(pop[sample(1:500,sam.n),] %>%
                     filter(R==1) %>%
                     pull(satisfied))
}

summary(store)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4396  0.5244  0.5500  0.5503  0.5748  0.6585
```

Response bias visualization



Common sources of response bias

- Difficulty of reaching certain groups given a sampling procedure
- Convenience samples
- Differential interest in participating
- Different times of availability

Remember both population-sample mismatches and sampling bias can be relevant at the same time

DeclareDesign()

A flexible framework for making declarations about our population, samples, and diagnosing bias. Let's do what we just did entirely within the DeclareDesign() framework:

```
population <- declare_population(
  households = add_level(N=500,
     main=draw_binary(N=N, prob = 0.5),
     satisfied=correlate(given = main, rho = 0.5,
                         draw_binary, prob = 0.5)
))
```

DeclareDesign()

```
reporting <- declare_assignment(blocks=main,
                    assignment_variable = "R",
                    block_prob=c(0.2,0.5))

sampling <- declare_sampling(n=250)

my_estimand <- declare_estimands(mean(satisfied),
                                 label = "Ybar")

answer <- declare_estimator(satisfied ~ 1,
                            subset = (R==1),
                            model = lm_robust,
                            label = "est.")
```

DeclareDesign()

```
design <- population + reporting + sampling +
  my_estimand + answer
diagnosis <- diagnose_design(design)

diagnosis$diagnosands_df[,c(5,11,13,15)] %>%
  kable()
```

bias	coverage	mean_estimate	sd_estimate
0.0644282	0.84	0.5633482	0.0510381