

Fitting the outcome model

Malcolm Barrett
Stanford University

Outcome Model

```
1 library(broom)
2
3 lm(outcome ~ exposure, data = df, weights = wts) |>
4   tidy()
```



This will get us the point estimate



This will get NOT us the correct confidence intervals



Let's bootstrap them with rsample

1. Create a function to run your analysis once on a sample of your data

```
1 fit_ipw <- function(.split, ...) {  
2   .df <- as.data.frame(.split)  
3  
4   # fit propensity score model  
5   propensity_model <- glm(  
6     qsmk ~ sex +  
7       race + age + I(age^2) + education +  
8       smokeintensity + I(smokeintensity^2) +  
9       smokeyrs + I(smokeyrs^2) + exercise + active +  
10      wt71 + I(wt71^2),  
11     family = binomial(),  
12     data = .df  
13   )  
14  
15   # calculate inverse probability weights  
16   .df <- propensity_model |>  
17     augment(type.predict = "response", data = .df) |>  
18     mutate(wts = wt_ate(.fitted, qsmk, exposure_type = "binary"))  
19  
20   # fit correctly bootstrapped ipw model  
21   lm(wt82_71 ~ qsmk, data = .df, weights = wts) |>  
22     tidy()  
23 }
```

2. Use {rsample} to bootstrap our causal effect

```
1 library(rsample)
2
3 # fit ipw model to bootstrapped samples
4 bootstrapped_nhefs <- bootstraps(
5   nhefs_complete_uc,
6   times = 1000,
7   apparent = TRUE
8 )
9
10 bootstrapped_nhefs
```

2. Use {rsample} to bootstrap our causal effect

```
# Bootstrap sampling with apparent sample
```

```
# A tibble: 1,001 × 2
```

	splits	id
	<list>	<chr>
1	<split [1566/575]>	Bootstrap0001
2	<split [1566/575]>	Bootstrap0002
3	<split [1566/583]>	Bootstrap0003
4	<split [1566/573]>	Bootstrap0004
5	<split [1566/549]>	Bootstrap0005
6	<split [1566/570]>	Bootstrap0006
7	<split [1566/589]>	Bootstrap0007
8	<split [1566/567]>	Bootstrap0008
9	<split [1566/563]>	Bootstrap0009
10	<split [1566/586]>	Bootstrap0010

2. Use {rsample} to bootstrap our causal effect

```
1 fit_ipw(bootstrapped_nhefs$splits[[1]])
```

```
# A tibble: 2 × 5
```

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	2.05	0.296	6.92	6.47e-12
2	qsmk	3.28	0.418	7.84	7.99e-15

2. Use {rsample} to bootstrap our causal effect

```
1 ipw_results <- bootstrapped_nhefs |>
2   mutate(boot_fits = map(splits, fit_ipw))
3
4 ipw_results
```

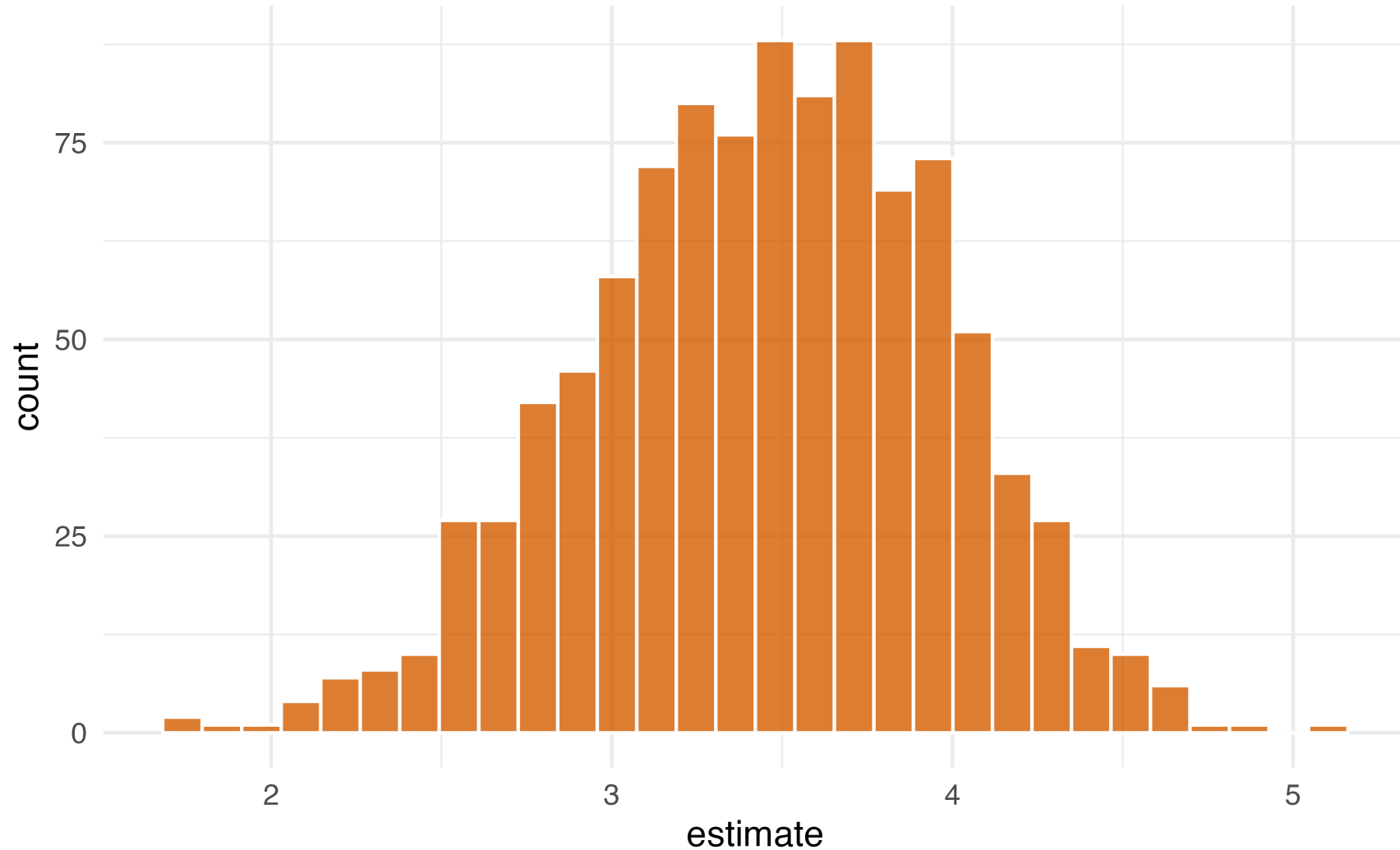
2. Use {rsample} to bootstrap our causal effect

```
# Bootstrap sampling with apparent sample
```

```
# A tibble: 1,001 × 3
```

	splits	id	boot_fits
	<list>	<chr>	<list>
1	<split [1566/564]>	Bootstrap0001	<tibble [2 × 5]>
2	<split [1566/573]>	Bootstrap0002	<tibble [2 × 5]>
3	<split [1566/575]>	Bootstrap0003	<tibble [2 × 5]>
4	<split [1566/586]>	Bootstrap0004	<tibble [2 × 5]>
5	<split [1566/592]>	Bootstrap0005	<tibble [2 × 5]>
6	<split [1566/585]>	Bootstrap0006	<tibble [2 × 5]>
7	<split [1566/551]>	Bootstrap0007	<tibble [2 × 5]>
8	<split [1566/584]>	Bootstrap0008	<tibble [2 × 5]>
9	<split [1566/588]>	Bootstrap0009	<tibble [2 × 5]>
10	<split [1566/567]>	Bootstrap0010	<tibble [2 × 5]>

2. Use `{rsample}` to bootstrap our causal effect



3. Pull out the causal effect

```
1 # get t-statistic-based CIs
2 boot_estimate <- int_t(ipw_results, boot_fits) |>
3   filter(term == "qsmk")
4
5 boot_estimate
```

```
# A tibble: 1 × 6
  term    .lower .estimate .upper .alpha .method
<chr>   <dbl>     <dbl>   <dbl> <dbl>   <chr>
1 qsmk    2.50      3.45    4.47  0.05   student-t
```

Your Turn

Create a function called `ipw_fit` that fits the propensity score model and the weighted outcome model for the effect between `park_extra_magic_morning` and `wait_minutes_posted_avg`

Using the `bootstraps()` and `int_t()` functions to estimate the final effect.