

# Fitting the outcome model

Malcolm Barrett

2021-09-01

# 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



{rsample}

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

```
1 fit_ipw <- function(split, ...) {  
2   .df <- analysis(split)  
3  
4   # fit propensity score model  
5   propensity_model <- glm(  
6     exposure ~ confounder_1 + confounder_2 + ...  
7     family = binomial(),  
8     data = .df  
9   )  
10  
11  # calculate inverse probability weights  
12  .df <- propensity_model |>  
13    augment(type.predict = "response", data = .df) |>  
14    mutate(wts = 1 / ifelse(exposure == 0, 1 - .fitted, .fitted))  
15  
16  # fit correctly bootstrapped ipw model  
17  lm(outcome ~ exposure, data = .df, weights = wts) |>  
18    tidy()  
19 }
```

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

```
1 library(rsample)
2
3 # fit ipw model to bootstrapped samples
4 ipw_results <- bootstraps(df, 1000, apparent = TRUE) |>
5   mutate(results = map(splits, fit_ipw))
```

### 3. Pull out the causal effect

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

## *Your Turn*

Create a function called **ipw\_fit** that fits the propensity score model and the weighted outcome model for the effect between **extra\_magic\_morning** and **avg\_spostmin**

Using the **bootstraps()** and **int\_t()** functions to estimate the final effect.

