WP6 Solution

Read in data

```
segsub <- read_csv("Segregation_HS.csv")
```

```
## New names:
## * ‘‘ -> ...1
```

```
## Rows: 4751 Columns: 32
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): unit
## dbl (31): ...1, obs, date_id, period, total_fatal, white_fatal, black_fatal,...
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
int <- read_csv("Integration_HS.csv")
```

```
## New names:
## * ‘‘ -> ...1
```

```
## Rows: 6162 Columns: 22
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): unit
## dbl (21): ...1, obs, date_id, total_fatal, white_fatal, black_fatal, auth, e...
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Figure 2

We need to do a bit of data wrangling to get this answer

First, blackbn is a number. We should turn it into a character.

```
segsub <- segsub %>%
    mutate(blackbn = as.character(blackbn))
```

Next, we need to group by the count and blackbn variables. Within each of these groupings we want to get the total number (`n()`) and the average fatality rate. The variable for this from the replication codebook is `spec_fatal_rate`

```
fig2Data <- segsub %>%
    group_by(count, blackbn) %>%
    summarise(number = n(), avg = mean(spec_fatal_rate, na.rm = T))
```

```
## ‘summarise()‘ has grouped output by ’count’. You can override using the ‘.groups‘ argument.
```

Now that we have the data, we can make the figure.

```
ggplot(fig2Data, aes(x = count, y = avg, group = blackbn, linetype = blackbn)) +
    geom_line(size = 1) + geom_point(data = segsub, aes(x = count,
    y = spec_fatal_rate, color = blackbn), alpha = 0.5) + theme_bw()
```

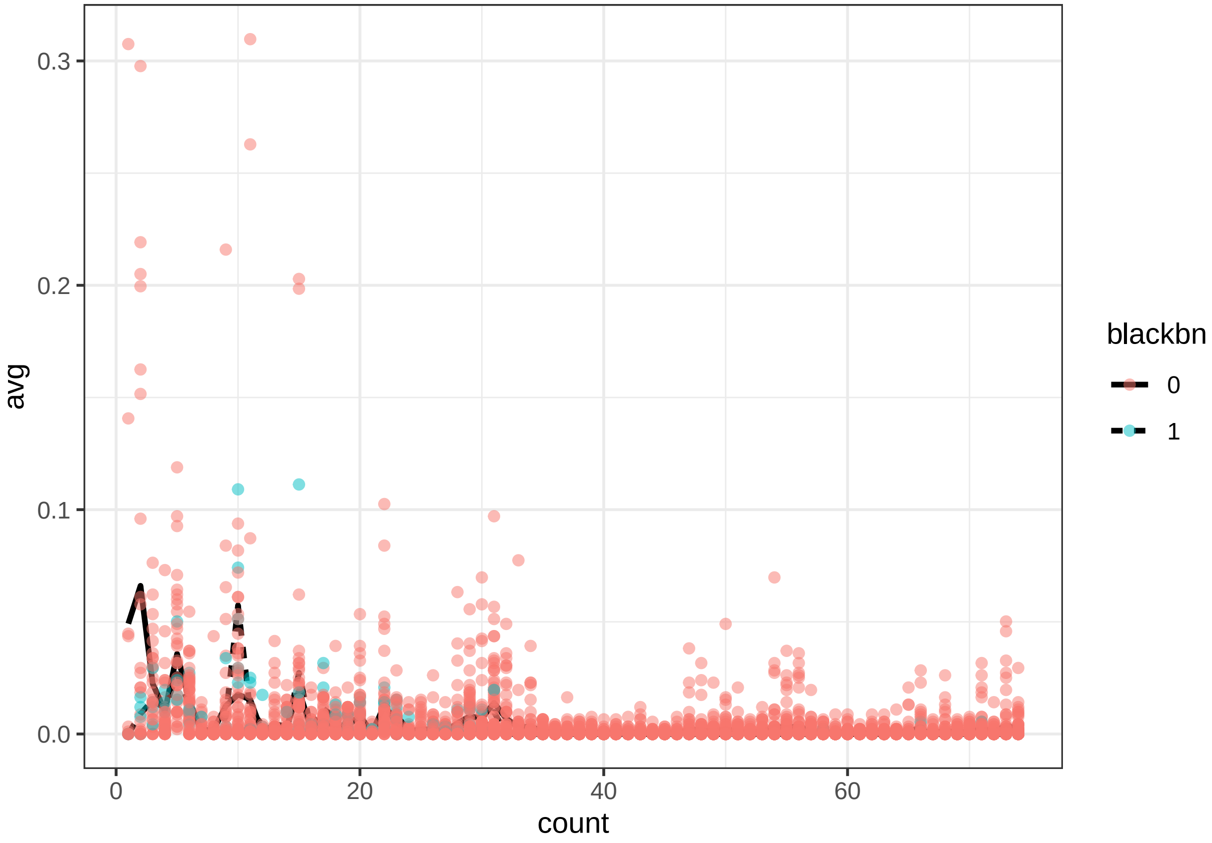


Figure 3

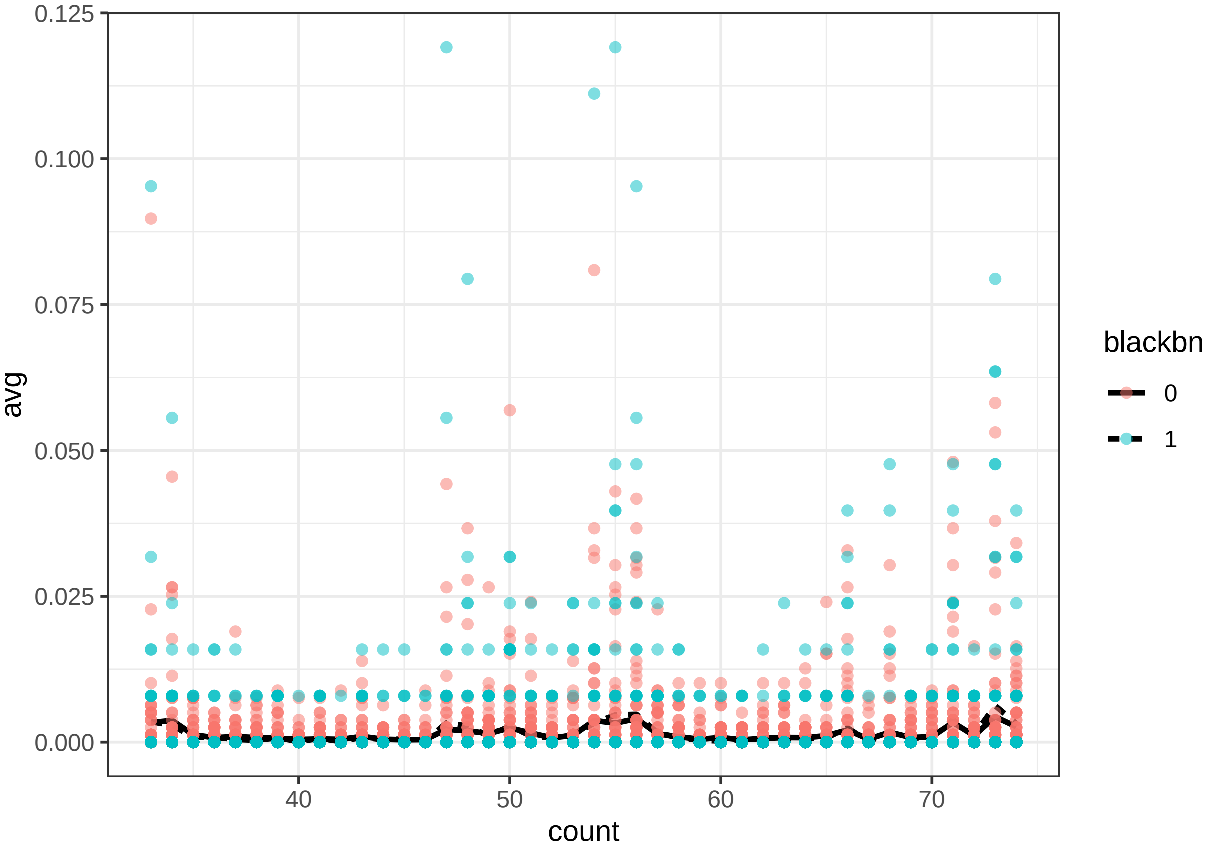
We follow the same steps as before but with the int data set.

```
int <- int %>%
    mutate(blackbn = as.character(blackbn))


fig3Data <- int %>%
    group_by(count, blackbn) %>%
    summarise(number = n(), avg = mean(specpct_imputed, na.rm = T))
```

```
## ‘summarise()‘ has grouped output by ’count’. You can override using the ‘.groups‘ argument.
```

```
## Figure 3
ggplot(fig3Data, aes(x = count, y = avg, group = blackbn, linetype = blackbn)) +
    geom_line(size = 1) + geom_point(data = int, aes(x = count,
    y = specpct_imputed, color = blackbn), alpha = 0.5) + theme_bw()
```



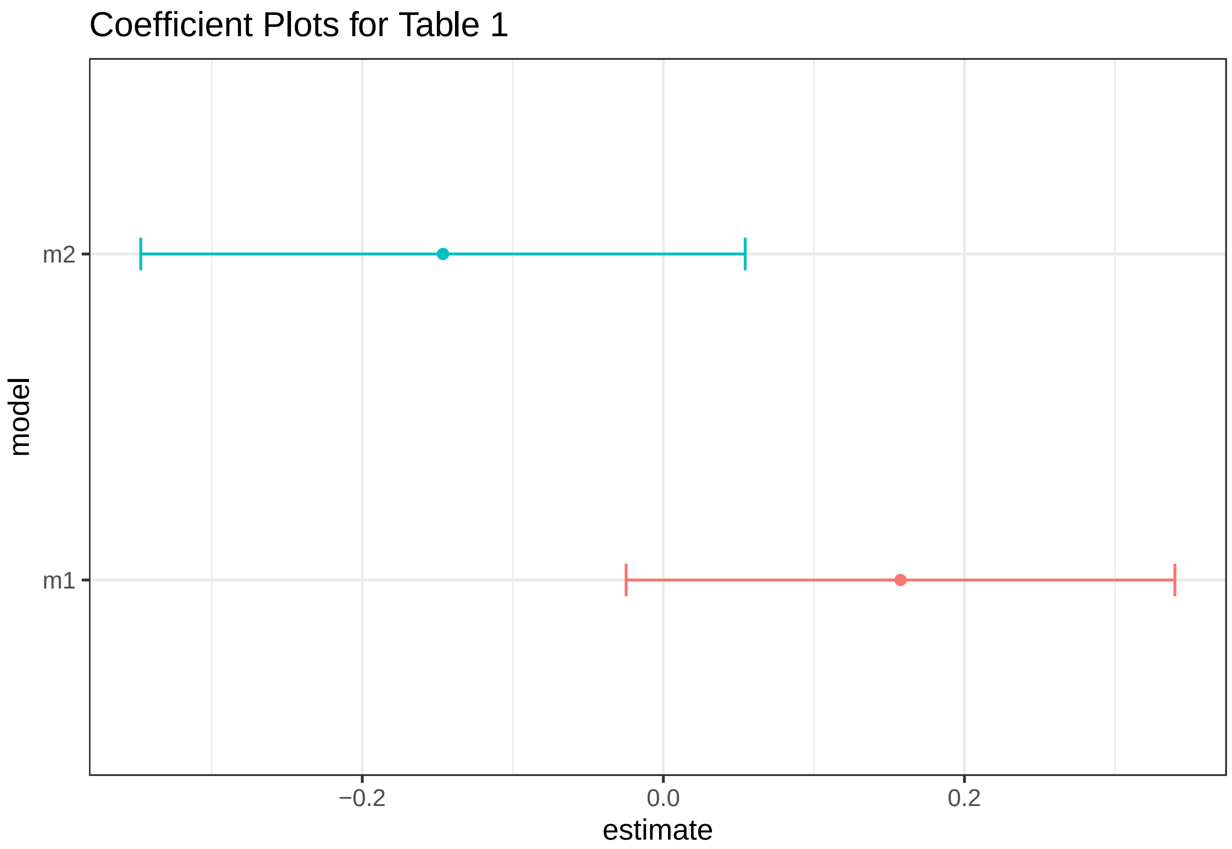
Replicate OLS models for Table 1

To replicate the models, we look to see the appropriate variables, and then make the coefficient plot as usual.

The one "new" thing here is fixed effects, which we will cover next week.

```r
m1 <- lm_robust(spec_fatal_rate100 ~ blackbn, data = segsub,
    se_type = "HC3") %>%
    tidy() %>%
    mutate(model = "m1")
m2 <- lm_robust(spec_fatal_rate100 ~ blackbn, data = segsub,
    fixed_effects = ~count, se_type = "HC3") %>%
    tidy() %>%
    mutate(model = "m2")

## Coefficient plot of models for Table 1
cplot <- bind_rows(m1, m2) %>%
    filter(term == "blackbn1") %>%
    ggplot(aes(x = estimate, y = model, color = model)) + geom_point() +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high, height = 0.1)) +
    theme_bw() + guides(color = "none") + ggtitle("Coefficient Plots for Table 1")
cplot
```

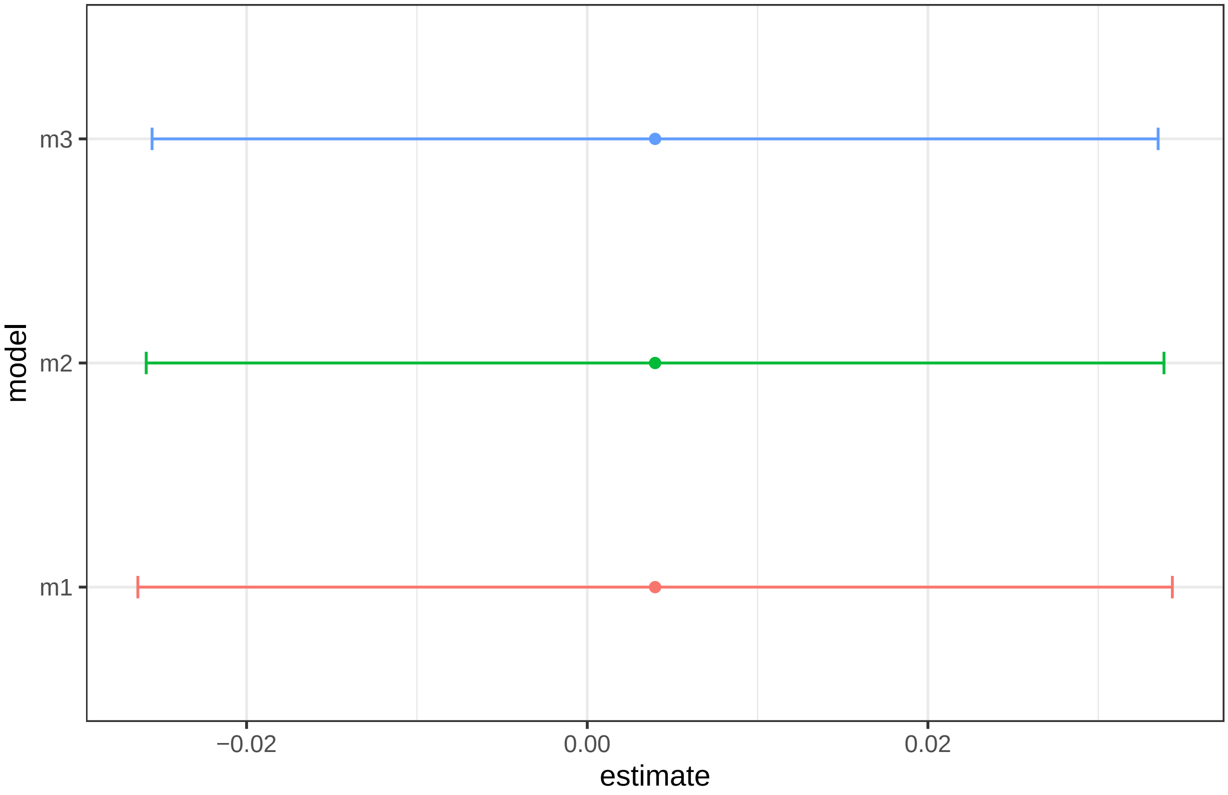


Replicate Table 3

Conceptually, nothing is different than what was required for the replication of Table 1.

```r
m31 <- lm_robust(specpct_imputed100 ~ blackbn, data = int, se_type = "HC3") %>%
    tidy() %>%
    mutate(model = "m1")
m32 <- lm_robust(specpct_imputed100 ~ blackbn, data = int, fixed_effects = ~count,
    se_type = "HC3") %>%
    tidy() %>%
    mutate(model = "m2")
m33 <- lm_robust(specpct_imputed100 ~ blackbn, data = int, fixed_effects = ~count +
    obs, se_type = "HC3") %>%
    tidy() %>%
    mutate(model = "m3")

cplot2 <- bind_rows(m31, m32, m33) %>%
    filter(term == "blackbn1") %>%
    ggplot(aes(x = estimate, y = model, color = model)) + geom_point() +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high, height = 0.1)) +
    theme_bw() + guides(color = "none") + ggtitle("Coefficient Plots for Table 3")
cplot2
```



Coefficient Plots for Table 3