Maryland Voter Turnout

Josh Stim

2025-02-15

Introduction

Data Cleaning

The data used for this project come from two sources: L2 (via Redistricting Hub), and the American Community Survey (ACS via `tidycensus`). These data sources are summarized below:

L2 (Redistricting Hub):

We obtained demographic and turnout data on the census-block-level for *registered voters* in Maryland in 2022. These included: 1) the total number of registered voters for the 2020 and 2022 general and primary elections for a given census block, 2) the turnout of these registered voters for each election, and 3) race/ethnicity, binned age, and gender counts of the registered voters who reported living in that block as of 2022.

To convert the L2 variables into percentages, I divided each election turnout count by the total number of registered voters in the census block for a given election, and divided the race/ethnicity, binned age, and gender counts by the total number of registered voters who reported living in the block in 2022.

Note that because the registered voter data (L2) is at the census block level, we will aggregate up to the census tract level in order to match the data from ACS. This can easily be done by extracting the census-tract code from the block code, and then summing the count variables across blocks within a given census tract.

American Community Survey (ACS):

We obtained census-tract-level data on the *total population* in Maryland in 2022. These census-tract-level data included **median household income** as well as count variables for **work transportation methods** (driving vs. public transportation), **educational attainment** (high school vs bachelors vs graduate), and **unemployment**.

Load datasets

```r
# 2022 L2 Registered Voter Turnout and Turnout History Data (block level)
md2022_voter_turnout <- read.csv(here('../Data/MD_l2_2022stats_2020block/MD_l2_2022stats_2020block.csv')

# Census tract demographic data
combined_tract_data <- read.csv(here('combined_tract_data.csv'))

# Get Maryland county names
county_names <- get_acs(
  geography = "county",
  state = "MD",
  year = 2022,
  variables = c(
```

```
    variable = "B19013_001"
  )) %>%
  mutate(NAME = substr(NAME,1, nchar(NAME) - 10),
         fip = GEOID) %>%
  mutate(county_name = NAME) %>%
  select(-c(GEOID, NAME, variable, estimate, moe))
```

```
## Getting data from the 2018-2022 5-year ACS
```

```
# Get median income data (units of $10000) for each county tract
md_income_data <- get_acs(
  geography = "tract",
  variables = "B19013_001",
  state = "MD",
  year = 2022,  # Update to the latest available year, if necessary
  survey = "acs5",  # Use the 5-year estimates
  geometry = TRUE
) %>%
  mutate(tract_id = GEOID,
         median_income = estimate/1000) %>%
  select(-c(NAME, GEOID, variable, estimate, moe))
```

```
## Getting data from the 2018-2022 5-year ACS

## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session

##   |                                                                              |
```

Join datasets and extract/compute relevant variables

```
# make sure join keys are aligned across datasets
combined_tract_data <- combined_tract_data %>%
  mutate(tract_id = GEOID) %>%
  mutate(fip = substr(tract_id, 1,5))

## md2022 data are at block level. We will aggregate up to tract level
md2022_voter_turnout.tract_level <- md2022_voter_turnout %>%
  mutate(tract_id = substr(geoid20,1,11)) %>%
  group_by(tract_id) %>%
  summarise(across(.cols = where(is.numeric), .fns = sum)) %>%
  ungroup()

# join all loaded datasets into one dataset
data_joined <- md2022_voter_turnout.tract_level %>%
  full_join(combined_tract_data, by = "tract_id") %>%
  full_join(md_income_data, by = "tract_id") %>%
  full_join(county_names, by = "fip")

# extract relevant variables from joined dataset
variables <- c("tract_id", "fip", "county_name",
               "age_18_19", "age_20_24", "age_25_29", "age_30_34", "age_35_44", "age_45_54", "age_55_64
               "age_65_74", "age_75_84", "age_85over",
               "voters_gender_m", "voters_gender_f", "voters_gender_unknown",
               "median_income",
               "Total_Employed", "Total_Unemployed",
```

```r
               "High_School_Graduate", "Total_Education_Population",
               "Public_Transportation_Users", "Total_Commuters",
               "eth1_eur", "eth1_hisp", "eth1_aa", "eth1_esa", "eth1_oth", "eth1_unk",
               "g20221108_voted_all", "g20221108_reg_all","p20220719_voted_all", "p20220719_reg_all",
               "g20201103_voted_all", "g20201103_reg_all", "p20200602_voted_all", "p20200602_reg_all",
               "geometry")

data_joined.truncated <- data_joined %>%
  select(all_of(variables))

# rename some variables for easier handling
data_joined.truncated <- data_joined.truncated %>%
  mutate(gen2022_voted = g20221108_voted_all,
         gen2022_reg = g20221108_reg_all,
         gen2020_voted = g20201103_voted_all,
         gen2020_reg = g20201103_reg_all,
         prim2022_voted = p20220719_voted_all,
         prim2022_reg = p20220719_reg_all,
         prim2020_voted = p20200602_voted_all,
         prim2020_reg = p20200602_reg_all) %>%
  select(-ends_with("all"))

# identify tracts with fewer than 50 registered voters over any election
data_joined.truncated <- data_joined.truncated %>%
  mutate(more_than_50 = gen2022_reg > 50 | gen2020_reg > 50 | prim2022_reg > 50 | prim2020_reg > 50)

# calculate percentages
data_joined.truncated <- data_joined.truncated %>%
  mutate(pct_over_65 = 100 * (age_65_74 + age_75_84 + age_85over) / (age_18_19 + age_20_24 + age_25_29 +
         pct_male = 100 * voters_gender_m / (voters_gender_f + voters_gender_m + voters_gender_unknown)
         pct_unemployed = 100 * Total_Unemployed / (Total_Employed + Total_Unemployed),
         pct_hs = 100 * High_School_Graduate / Total_Education_Population,
         pct_transit = 100 * Public_Transportation_Users / Total_Commuters,
         pct_aa = 100 * eth1_aa / (eth1_eur + eth1_hisp + eth1_aa + eth1_esa + eth1_oth + eth1_unk),
         pct_hisp = 100 * eth1_hisp / (eth1_eur + eth1_hisp + eth1_aa + eth1_esa + eth1_oth + eth1_unk)
         turnout_gen_2022 = 100 * gen2022_voted / gen2022_reg,
         turnout_gen_2020 = 100 * gen2020_voted / gen2020_reg,
         turnout_prim_2022 = 100 * prim2022_voted / prim2022_reg,
         turnout_prim_2020 = 100 * prim2020_voted / prim2020_reg) %>%
  select(tract_id, fip, county_name, starts_with("turnout"), median_income, starts_with("pct"), geometry

# create indicator for census tracts that are in Baltimore City
data_joined.truncated <- data_joined.truncated %>%
  mutate(isBmore = ifelse(county_name == "Baltimore city", TRUE, FALSE))

# determine which census tracts are have data for all relevant variables
data_joined.truncated$complete_case <- data_joined.truncated %>%
  select(-c(tract_id, fip, county_name, geometry)) %>%
  complete.cases(.)

# summarized joined dataset
summary(data_joined.truncated)
```

```
##     tract_id              fip              county_name         turnout_gen_2022
```

```
##  Length:1499        Length:1499        Length:1499        Min.   : 0.00
##  Class :character   Class :character   Class :character   1st Qu.:38.45
##  Mode  :character   Mode  :character   Mode  :character   Median :48.22
##                                                           Mean   :46.64
##                                                           3rd Qu.:56.52
##                                                           Max.   :72.14
##                                                           NA's   :14
##  turnout_gen_2020 turnout_prim_2022 turnout_prim_2020 median_income
##  Min.   :20.57    Min.   : 0.00     Min.   : 0.00     Min.   : 10.00
##  1st Qu.:66.93    1st Qu.:18.71     1st Qu.:31.47     1st Qu.: 70.53
##  Median :74.57    Median :24.26     Median :35.78     Median : 98.16
##  Mean   :72.00    Mean   :24.12     Mean   :36.59     Mean   :104.59
##  3rd Qu.:79.68    3rd Qu.:28.71     3rd Qu.:41.31     3rd Qu.:132.05
##  Max.   :89.00    Max.   :53.30     Max.   :63.56     Max.   :250.00
##  NA's   :14       NA's   :14        NA's   :14        NA's   :45
##   pct_over_65       pct_male     pct_unemployed       pct_hs
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 0.00
##  1st Qu.:22.55   1st Qu.:44.92   1st Qu.: 2.726   1st Qu.:13.64
##  Median :27.11   Median :47.00   Median : 4.439   Median :21.37
##  Mean   :27.21   Mean   :46.59   Mean   : 5.384   Mean   :21.44
##  3rd Qu.:31.36   3rd Qu.:48.57   3rd Qu.: 7.141   3rd Qu.:29.12
##  Max.   :89.94   Max.   :97.78   Max.   :45.455   Max.   :62.37
##  NA's   :14      NA's   :14      NA's   :43       NA's   :39
##   pct_transit          pct_aa          pct_hisp               geometry
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   MULTIPOLYGON :1499
##  1st Qu.: 0.7283   1st Qu.: 2.914   1st Qu.: 2.802   epsg:4269    :   0
##  Median : 3.1189   Median :18.179   Median : 4.405   +proj=long...:   0
##  Mean   : 6.1970   Mean   :30.718   Mean   : 6.552
##  3rd Qu.: 8.7794   3rd Qu.:55.527   3rd Qu.: 7.197
##  Max.   :52.6767   Max.   :93.768   Max.   :61.749
##  NA's   :43        NA's   :14       NA's   :14
##  more_than_50     isBmore        complete_case
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:23        FALSE:1276      FALSE:45
##  TRUE :1476      TRUE :199       TRUE :1454
##                  NA's :24
##
##
##
```

Evaluate data missingness

Below, I show all rows in the joined dataset that were missing values for one or more variables. From this summary, we can see that the first 23 rows are missing full block codes, which makes it impossible to determine which census tract they are located in. The remaining 20 rows are missing turnout data or are missing some subset of the demographic variables. Notice that many of the valid census tracts with missing data also have fewer than 50 registered voters in them.

```r
data_joined.truncated %>%
  filter(complete_case == FALSE)
```

```
## # A tibble: 45 x 19
##    tract_id    fip   county_name turnout_gen_2022 turnout_gen_2020
##    <chr>       <chr> <chr>                  <dbl>            <dbl>
##  1 001 - NO BL <NA>  <NA>                    44.8             59.9
```

```
##   2 003 - NO BL <NA>  <NA>                       42.9             73.4
##   3 005 - NO BL <NA>  <NA>                       42.5             66.8
##   4 009 - NO BL <NA>  <NA>                       65.6             68.8
##   5 011 - NO BL <NA>  <NA>                       36.2             52.3
##   6 013 - NO BL <NA>  <NA>                       52.3             77.4
##   7 015 - NO BL <NA>  <NA>                       36.4             65.3
##   8 017 - NO BL <NA>  <NA>                       29.9             72.4
##   9 019 - NO BL <NA>  <NA>                       52.7             75.4
##  10 021 - NO BL <NA>  <NA>                       53.2             81.3
## # i 35 more rows
## # i 14 more variables: turnout_prim_2022 <dbl>, turnout_prim_2020 <dbl>,
## #   median_income <dbl>, pct_over_65 <dbl>, pct_male <dbl>,
## #   pct_unemployed <dbl>, pct_hs <dbl>, pct_transit <dbl>, pct_aa <dbl>,
## #   pct_hisp <dbl>, geometry <MULTIPOLYGON [°]>, more_than_50 <lgl>,
## #   isBmore <lgl>, complete_case <lgl>
```

Below, I've calculated the percentage of tracts within each county that are missing some data.

```
data_joined.truncated %>%
  group_by(county_name) %>%
  summarize(n_missing = sum(!complete_case), n_total = n(), pct_w_some_missing_data = mean(!complete_ca
  arrange(desc(pct_w_some_missing_data))
```

```
## # A tibble: 25 x 4
##    county_name         n_missing n_total pct_w_some_missing_data
##    <chr>                   <int>   <int>                   <dbl>
##  1 <NA>                       24      24                  100
##  2 Queen Anne's County         3      14                   21.4
##  3 Kent County                 1       6                   16.7
##  4 Somerset County             1       7                   14.3
##  5 Worcester County            2      18                   11.1
##  6 Dorchester County           1      10                   10
##  7 Talbot County               1      11                    9.09
##  8 Calvert County              1      19                    5.26
##  9 St. Mary's County           1      23                    4.35
## 10 Charles County              1      35                    2.86
## # i 15 more rows
```

Excluding the first row from the table above, we can see that tract-level data missingness within counties ranges from 21% to 0%, with the majority of counties missing some data for less than 3% of their census tracts.

Create and summarize complete dataset of census tracts with more than 50 registered voters

The summary of 1454 complete cases (i.e., census tracts with data for all relevant demographic variables) is shown below.

```
data_joined.complete <- data_joined.truncated %>%
  filter(complete_case == TRUE & more_than_50 == TRUE)

data_joined.complete.long <- data_joined.complete %>%
  pivot_longer(cols = turnout_gen_2022:turnout_prim_2020,
               names_to = "election",
               values_to = "turnout") %>%
  mutate(election = case_when(election == "turnout_gen_2022" ~ "gen_2022",
```

```r
                              election == "turnout_gen_2020" ~ "gen_2020",
                              election == "turnout_prim_2022" ~ "prim_2022",
                              election == "turnout_prim_2020" ~ "prim_2020")) %>%
  mutate(election_type = ifelse(election %in% c("gen_2022", "gen_2020"), "General", "Primary"),
         election_year = factor(ifelse(election %in% c("gen_2022", "prim_2022"), 2022, 2020)))

colMeans(data_joined.complete[,4:15])
```

```
## turnout_gen_2022  turnout_gen_2020 turnout_prim_2022 turnout_prim_2020
##        46.771563         72.163219         24.231235         36.811850
##    median_income       pct_over_65          pct_male    pct_unemployed
##       104.585027         27.222383         46.549480          5.349029
##           pct_hs       pct_transit            pct_aa          pct_hisp
##        21.440223          6.182527         30.837459          6.617117
```

```r
apply(data_joined.complete[,4:15], 2, sd)
```

```
## turnout_gen_2022  turnout_gen_2020 turnout_prim_2022 turnout_prim_2020
##        11.904182          9.876745          7.112290          7.678808
##    median_income       pct_over_65          pct_male    pct_unemployed
##        46.506269          7.810658          2.796965          3.855217
##           pct_hs       pct_transit            pct_aa          pct_hisp
##        10.543522          7.867824         31.928689          6.842342
```

I've also summarized the dataframe looking only at census tracts in Baltimore City.

```r
summary(data_joined.complete %>% filter(isBmore == TRUE))
```

```
##    tract_id             fip            county_name        turnout_gen_2022
##  Length:196         Length:196         Length:196         Min.   :14.69
##  Class :character   Class :character   Class :character   1st Qu.:24.41
##  Mode  :character   Mode  :character   Mode  :character   Median :33.80
##                                                           Mean   :35.59
##                                                           3rd Qu.:43.43
##                                                           Max.   :71.43
##  turnout_gen_2020 turnout_prim_2022 turnout_prim_2020 median_income
##  Min.   :34.13    Min.   : 8.459    Min.   :19.16     Min.   : 13.04
##  1st Qu.:48.74    1st Qu.:16.586    1st Qu.:32.42     1st Qu.: 40.02
##  Median :57.70    Median :22.081    Median :39.34     Median : 54.41
##  Mean   :58.45    Mean   :22.805    Mean   :39.42     Mean   : 62.72
##  3rd Qu.:66.93    3rd Qu.:27.742    3rd Qu.:45.89     3rd Qu.: 71.56
##  Max.   :86.90    Max.   :53.298    Max.   :63.30     Max.   :229.74
##   pct_over_65        pct_male     pct_unemployed       pct_hs
##  Min.   : 6.697   Min.   :33.72   Min.   : 0.000   Min.   : 1.777
##  1st Qu.:19.754   1st Qu.:43.13   1st Qu.: 3.107   1st Qu.:15.254
##  Median :24.185   Median :45.05   Median : 6.935   Median :24.723
##  Mean   :23.925   Mean   :45.30   Mean   : 7.998   Mean   :24.507
##  3rd Qu.:28.435   3rd Qu.:47.31   3rd Qu.:11.159   3rd Qu.:33.142
##  Max.   :64.000   Max.   :55.41   Max.   :41.256   Max.   :62.370
##   pct_transit         pct_aa          pct_hisp               geometry
##  Min.   : 0.000   Min.   : 1.506   Min.   : 1.005   MULTIPOLYGON :196
##  1st Qu.: 6.232   1st Qu.:38.025   1st Qu.: 1.868   epsg:4269    :  0
##  Median :12.028   Median :79.935   Median : 2.527   +proj=long...:  0
##  Mean   :15.160   Mean   :63.642   Mean   : 3.396
##  3rd Qu.:21.187   3rd Qu.:90.839   3rd Qu.: 3.934
##  Max.   :52.677   Max.   :93.768   Max.   :27.453
```

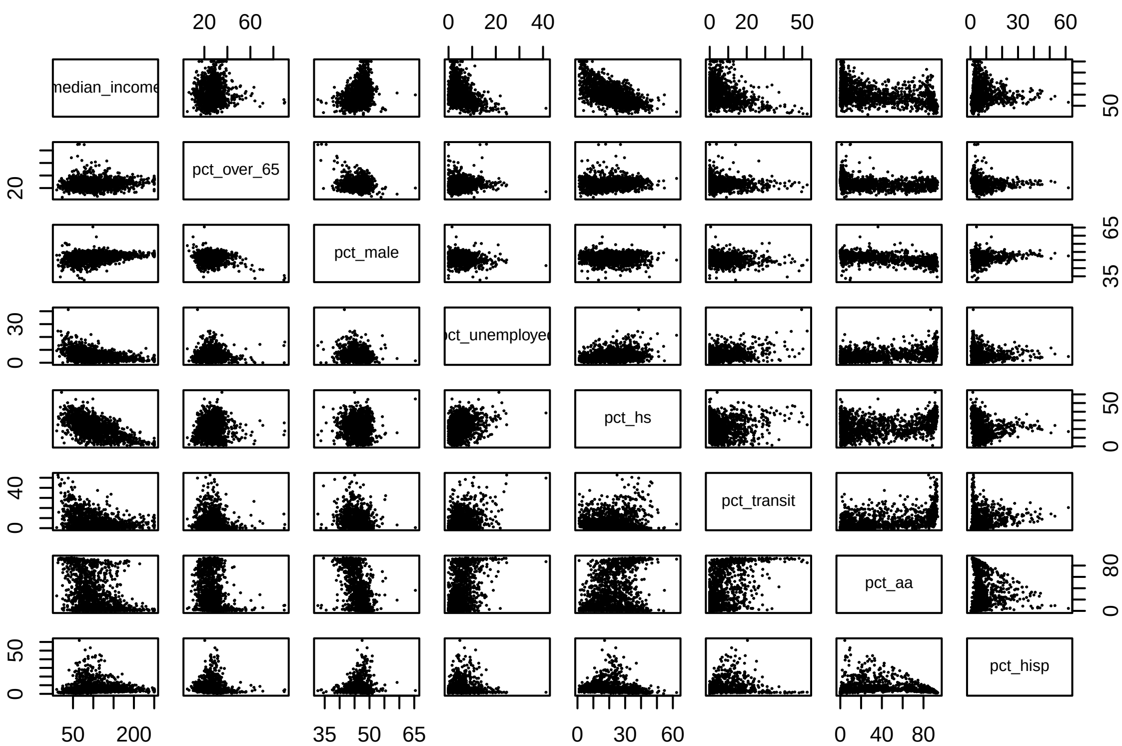
```
## more_than_50    isBmore         complete_case
## Mode:logical    Mode:logical    Mode:logical
## TRUE:196        TRUE:196        TRUE:196
##
##
##
##
```

Data Description

Below, I've created a scatterplot matrix to see relationships between each predictor variable in the complete dataframe for all census tracts in Maryland. I've also provided a correlation matrix to show correlations between predictors.

The strongest positive correlations (r = 0.54) is between percentage relying on public transportation (`pct_transit`) and percentage African American (`pct_aa`), and the strongest negative correlations (r = -0.62 and r = -0.58) are between median income (`median_income`) and percent with high school as the highest educational attainment (`pct_hs`), and between percent male (`pct_male`) and `pct_aa`, respectively.

```
pairs(data_joined.complete %>% select(-c(tract_id, fip, county_name, starts_with("turnout"), geometry, c
```

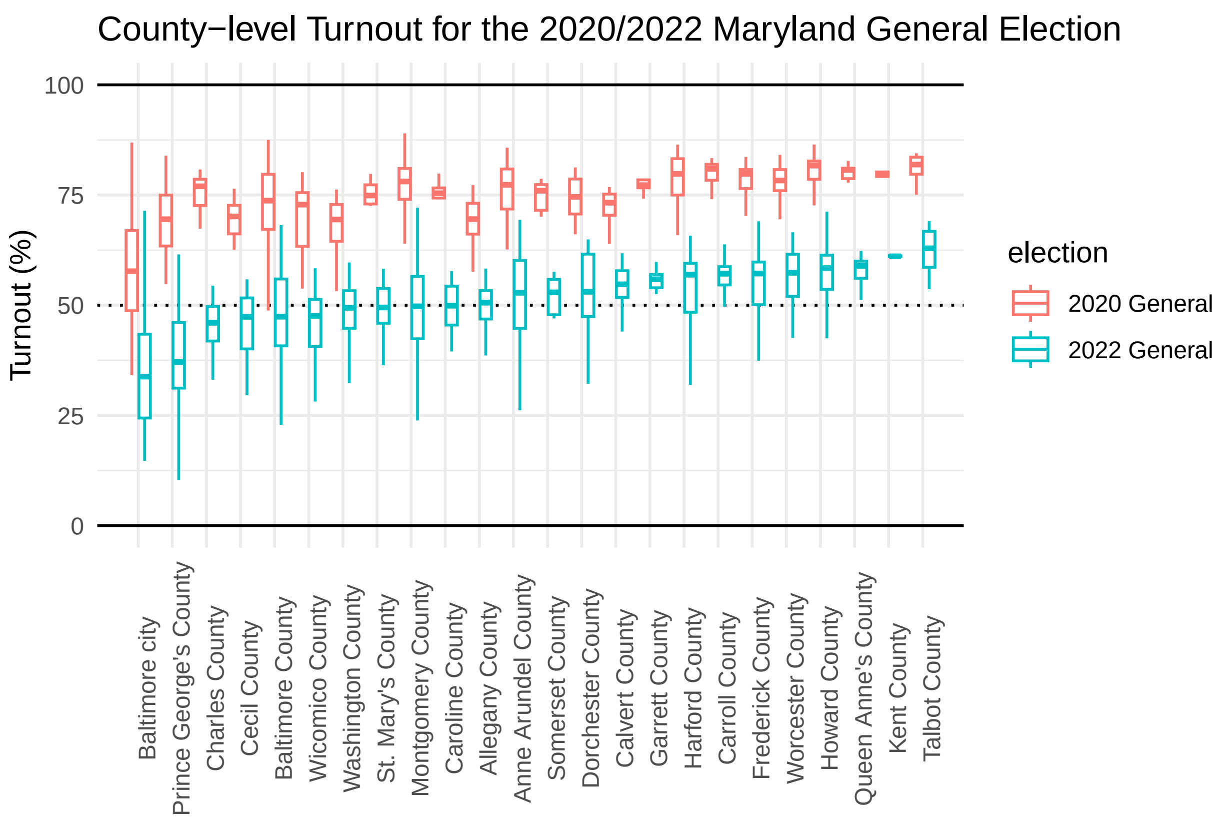



```
ggcorrplot(cor(data_joined.complete %>% select(-c(tract_id, fip, county_name, starts_with("turnout"), ge
```

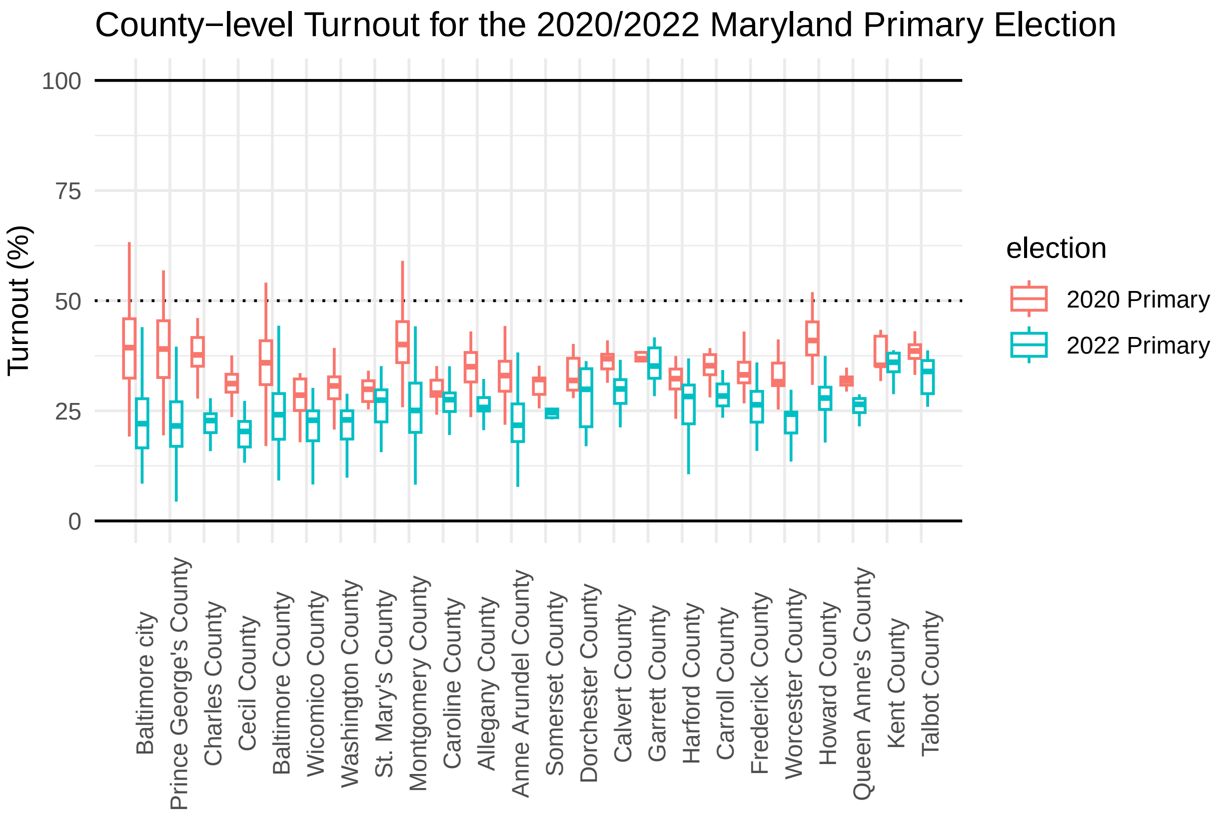
Below displays the distributions of voter turnouts by County in Maryland.

```
turnout_2022_general <- data_joined.complete.long %>%
  filter(election == "gen_2022") %>%
  select(county_name, election, turnout) %>%
  group_by(county_name) %>%
  summarize(turnout_median = median(turnout)) %>%
  arrange(turnout_median) %>%
  select(-turnout_median) %>%
  as.matrix()

md_county_general_turnout <- data_joined.complete.long %>%
  filter(election_type == "General") %>%
  mutate(county_name = factor(county_name, levels = turnout_2022_general)) %>%
  mutate(election = factor(election, labels = c("2020 General", "2022 General"))) %>%
  ggplot(aes(x = county_name, y = turnout, color = election)) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 50), linetype = "dotted") +
  geom_hline(aes(yintercept = 100)) +
  geom_boxplot(outliers = FALSE) +
  #stat_summary(fun = mean, color = "black", size = 0.3) +
  labs(title = "County-level Turnout for the 2020/2022 Maryland General Election", x = "", y = "Turnout
  #ylim(0,0.9) +
  theme_minimal() +
  scale_x_discrete(expand = c(0.05, 0.05)) +
  theme(axis.text.x = element_text(angle = 90))
```

```
md_county_primary_turnout <- data_joined.complete.long %>%
  mutate(county_name = factor(county_name, levels = turnout_2022_general)) %>%
  filter(election_type == "Primary") %>%
  mutate(election = factor(election, labels = c("2020 Primary", "2022 Primary"))) %>%
  ggplot(aes(x = county_name, y = turnout, color = election)) +
  geom_hline(aes(yintercept = 50), linetype = "dotted") +
  geom_boxplot(outliers = FALSE) +
  #stat_summary(fun = mean, color = "black", size = 0.3) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 100)) +
  labs(title = "County-level Turnout for the 2020/2022 Maryland Primary Election", x = "", y = "Turnout
  #ylim(0,0.9) +
  theme_minimal() +
  scale_x_discrete(expand = c(0.05, 0.05)) +
  theme(axis.text.x = element_text(angle = 90))

md_county_general_turnout
```



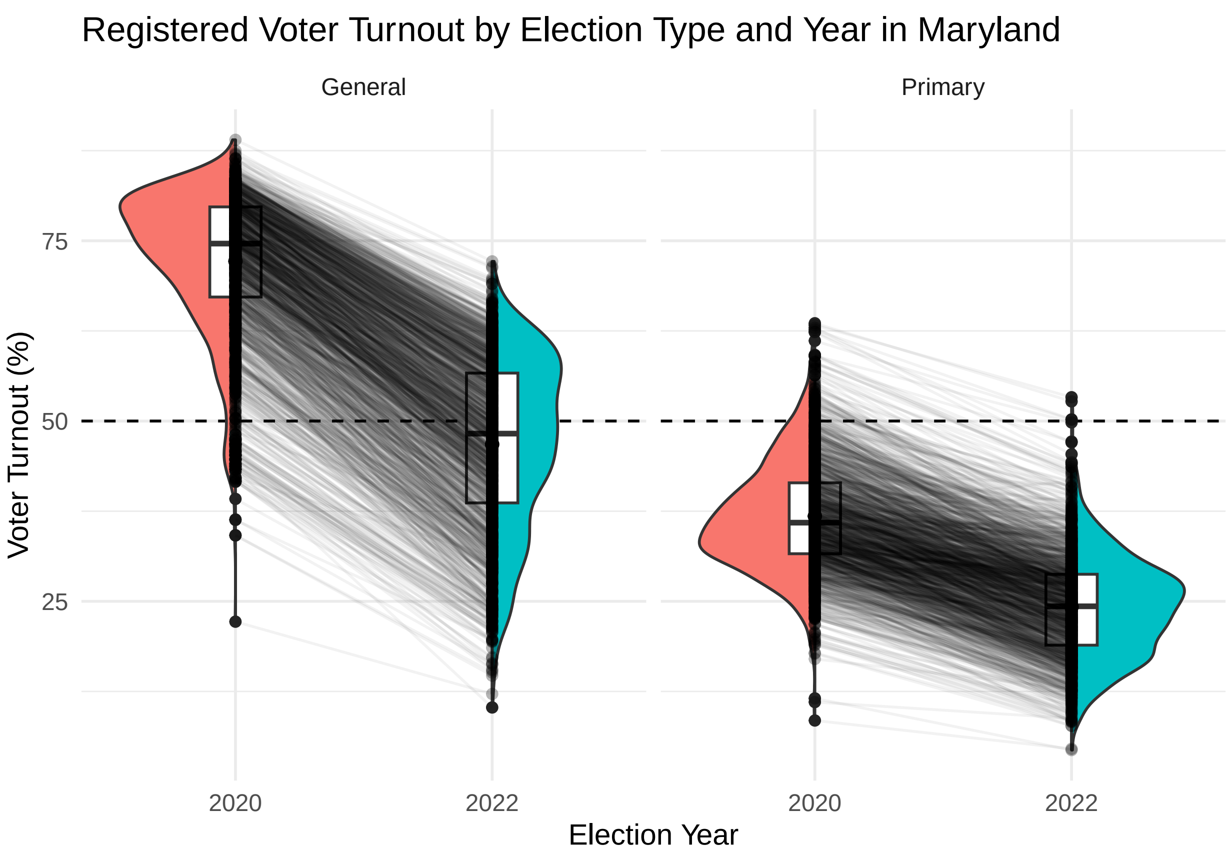

```
md_county_primary_turnout
```



Below is a visual summary of the distribution of tract-level voter turnouts for each election among complete cases in Maryland and Baltimore City. It also shows the change in voter turnout from 2022 to 2020 within each census tract. Before making this plot, I reformatted the dataframe into long format.

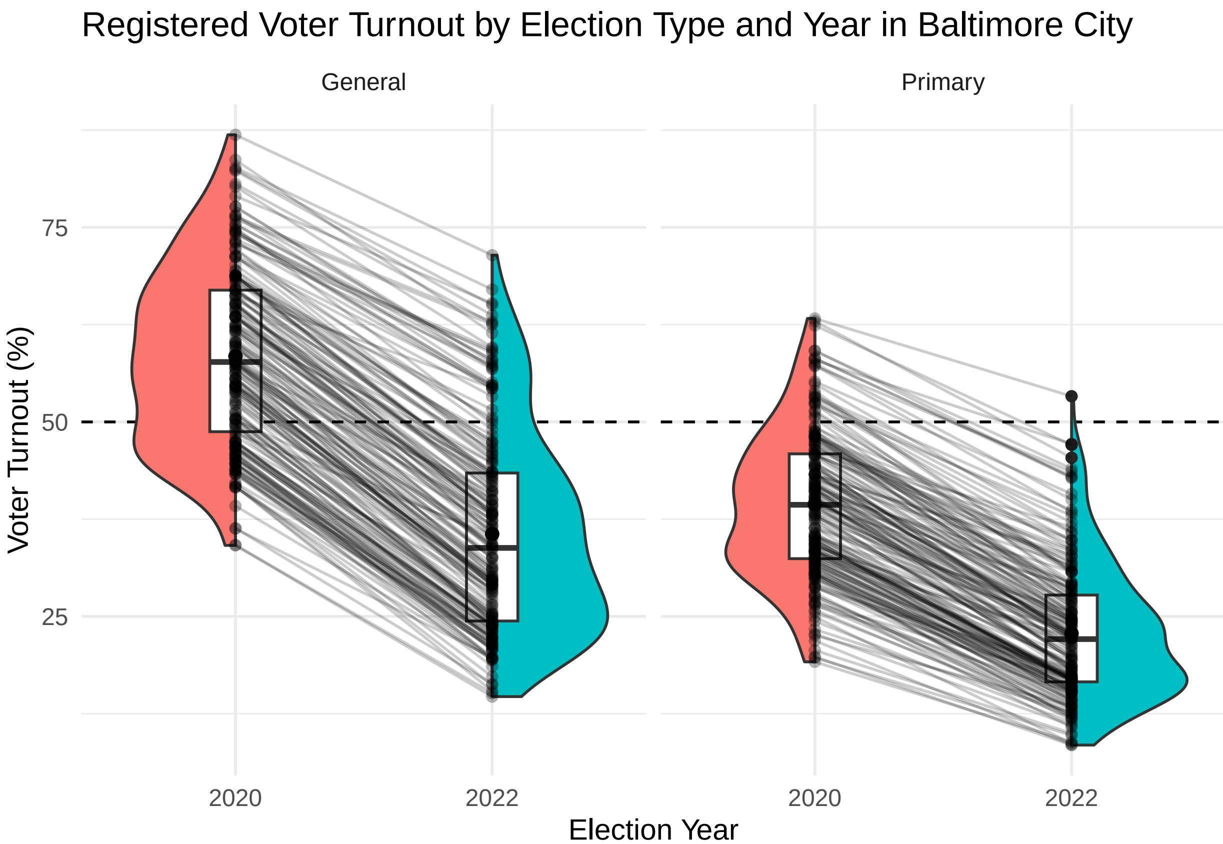
```
data_joined.complete.long %>%
  ggplot(aes(x = election_year, y = turnout, fill = election_year)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_violinhalf(flip = c(1,3)) +
  geom_boxplot(aes(fill = NULL), width = 0.2) +
  geom_point(alpha = 0.3) +
  geom_line(aes(fill = NULL, group = tract_id), alpha = 0.05) +
  stat_summary(fun = mean, size = 0.3) +
  facet_wrap(~election_type) +
  labs(title = "Registered Voter Turnout by Election Type and Year in Maryland", x = "Election Year", y
  theme_minimal() +
  theme(legend.position = "none")
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_segment()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```




```
data_joined.complete.long %>%
  filter(isBmore == TRUE) %>%
  ggplot(aes(x = election_year, y = turnout, fill = election_year)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_violinhalf(flip = c(1,3)) +
  geom_boxplot(aes(fill = NULL), width = 0.2) +
  geom_point(alpha = 0.3) +
  geom_line(aes(fill = NULL, group = tract_id), alpha = 0.2) +
  stat_summary(fun = mean, size = 0.3) +
  facet_wrap(~election_type) +
  labs(title = "Registered Voter Turnout by Election Type and Year in Baltimore City", x = "Election Ye
  theme_minimal() +
  theme(legend.position = "none")
```

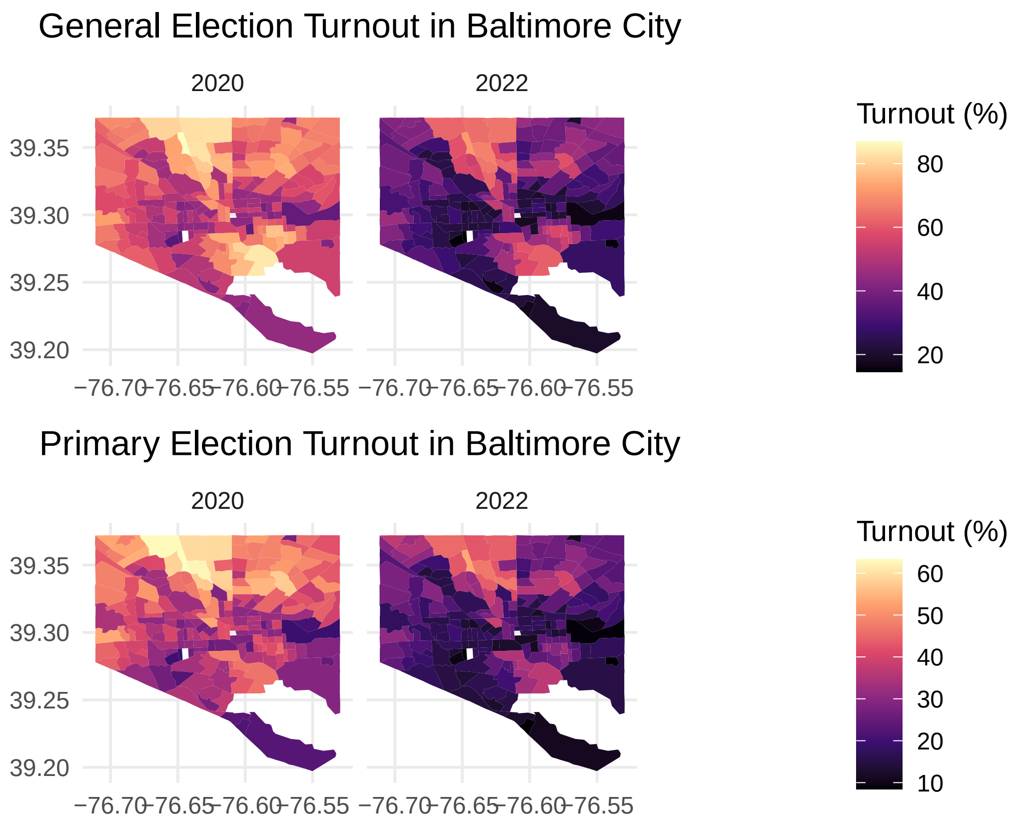
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_segment()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```



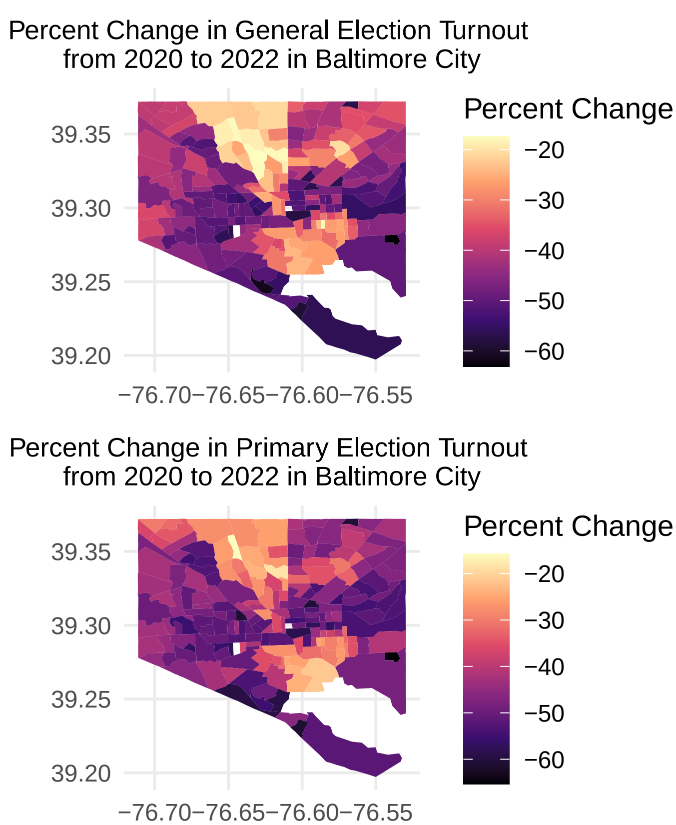
I also looked at the spatial distribution of voter turnout and change in voter turnout across election years in Baltimore City.

```
# Voter turnout general elections
bcity_general_turnout_map <- data_joined.complete.long %>%
  filter(isBmore == TRUE,
         election_type == "General") %>%
  ggplot(aes(fill = turnout)) +
  geom_sf(aes(geometry = geometry), color = NA) +
  scale_fill_viridis_c(option = "magma") +
  facet_wrap(~election_year) +
  labs(title = "General Election Turnout in Baltimore City",
       fill = "Turnout (%)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Percent change in general election voter turnout from 2020 to 2022
bcity_general_diff_map <- data_joined.complete.long %>%
  filter(isBmore == TRUE,
         election_type == "General") %>%
  pivot_wider(names_from = c(election, election_year, election_type),
              values_from = turnout) %>%
  mutate(pct_change = 100 * (gen_2022_2022_General - gen_2020_2020_General) / gen_2020_2020_General) %>%
  ggplot(aes(fill = pct_change)) +
  geom_sf(aes(geometry = geometry), color = NA) +
  scale_fill_viridis_c(option = "magma") +
  labs(title = "Percent Change in General Election Turnout \nfrom 2020 to 2022 in Baltimore City",
       fill = "Percent Change") +
```

```r
  theme_minimal() +
  theme(plot.title = element_text(size = 10, hjust = 0.5))

# Voter turnout primary elections
bcity_primary_turnout_map <- data_joined.complete.long %>%
  filter(isBmore == TRUE,
         election_type == "Primary") %>%
  ggplot(aes(fill = turnout)) +
  geom_sf(aes(geometry = geometry), color = NA) +
  scale_fill_viridis_c(option = "magma") +
  facet_wrap(~election_year) +
  labs(title = "Primary Election Turnout in Baltimore City",
       fill = "Turnout (%)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Percent change in primary election voter turnout from 2020 to 2022
bcity_primary_diff_map <- data_joined.complete.long %>%
  filter(isBmore == TRUE,
         election_type == "Primary") %>%
  pivot_wider(names_from = c(election, election_year, election_type),
              values_from = turnout) %>%
  mutate(pct_change = 100 * (prim_2022_2022_Primary - prim_2020_2020_Primary) / prim_2020_2020_Primary)
  ggplot(aes(fill = pct_change)) +
  geom_sf(aes(geometry = geometry), color = NA) +
  scale_fill_viridis_c(option = "magma") +
  labs(title = "Percent Change in Primary Election Turnout \nfrom 2020 to 2022 in Baltimore City",
       fill = "Percent Change") +
  theme_minimal() +
  theme(plot.title = element_text(size = 10, hjust = 0.5))

bcity_general_turnout_map / bcity_primary_turnout_map
```




```
bcity_general_diff_map / bcity_primary_diff_map
```

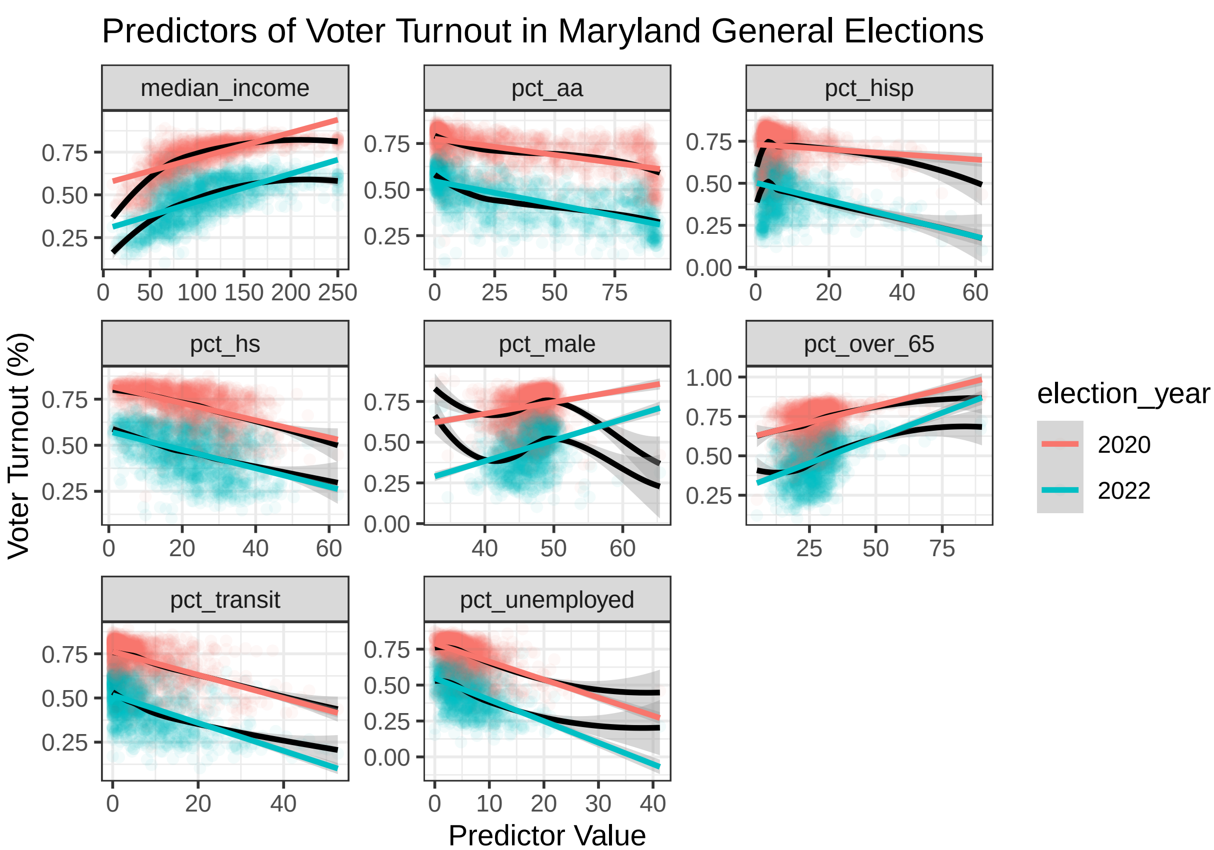


These plots suggest that Baltimore City voter turnouts (and changes in turnouts over time) very much follow the "Black butterfly" pattern. That is, majority black neighborhoods tended to show lower election turnouts in 2020 compared to other neighborhoods in Baltimore City. On top of this, majority black neighborhoods had steeper decreases in voter turnouts from 2020 to 2022 compared to other neighborhoods in Baltimore City.

Finally, I produced some plots to visualize unadjusted marginal relationships between voter turnout and each of the predictor variables.

```
data_joined.complete.extra_long <- data_joined.complete.long %>%
  pivot_longer(cols = median_income:pct_hisp, names_to = "Variable", values_to = "Value")

data_joined.complete.extra_long %>%
  filter(election_type == "General") %>%
  ggplot(aes(x = Value, y = turnout/100, color = election_year)) +
  geom_point(alpha = 0.05) +
  geom_smooth(aes(group = election_year), color = "black", method = "loess") +
  geom_smooth(aes(group = election_year), method = "lm") +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Predictors of Voter Turnout in Maryland General Elections",
       y = "Voter Turnout (%)",
       x = "Predictor Value") +
  theme_bw()
```

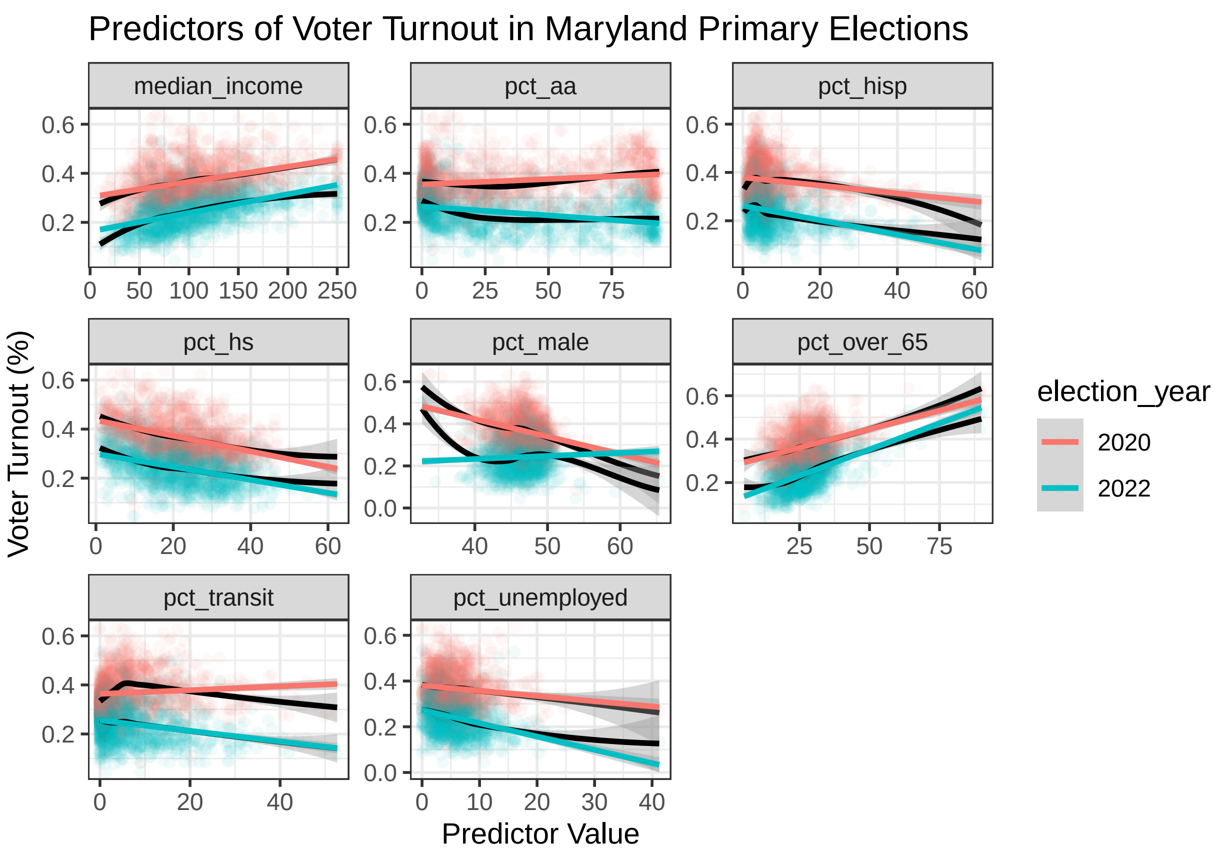
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
data_joined.complete.extra_long %>%
  filter(election_type == "Primary") %>%
```

```
ggplot(aes(x = Value, y = turnout/100, color = election_year)) +
geom_point(alpha = 0.05) +
geom_smooth(aes(group = election_year), color = "black", method = "loess") +
geom_smooth(aes(group = election_year), method = "lm") +
facet_wrap(~Variable, scales = "free") +
labs(title = "Predictors of Voter Turnout in Maryland Primary Elections",
     y = "Voter Turnout (%)",
     x = "Predictor Value") +
theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



For general elections, median income appears to have a particularly non-linear association with voter turnout percentage; the slope is steeper from 0 - 75 compared to from >75 - 250. Therefore, I will use a linear spline with a knot at 75 to model this relationship when fitting future models. While the other variables show varying levels of deviation from linearity when plotted against voter turnout, I will refrain from using splines for these variables in efforts to avoid overfitting. Future analyses should consider further exploring these non-linearities, as it would be interesting to see if they persist over multiple (more than 2) election cycles.

For primary elections, most variables should relatively linear relationships with voter turnout percentages. In efforts to be consistent with the analysis on general elections, I will also use a linear spline with a knot at 75 to model the relationship between median income and voter turnout.

Model Fitting

This portion of the analysis is dedicated to estimating adjusted effects of the demographic variables explored above on voter turnouts from 2020 to 2022 for primary and general elections.

As stated in the data description section, I want to model the relationship between median income and voter turnout using a linear spline with one knot at 75. I create this linear spline below.

```
# consolidate variables for general election models
dat.gen <- data_joined.complete.long %>%
  filter(election_type == "General") %>%
  select(tract_id, county_name, turnout, election, isBmore, median_income, pct_over_65, pct_male, pct_un
  mutate(median_income_sp1 = ifelse(median_income > 75, median_income - 75, 0))

# make a standardized dataset
mean.inc.gen <- mean(dat.gen$median_income)
sd.inc.gen <- sd(dat.gen$median_income)
knot.inc.gen.st <- (75 - mean.inc.gen)/sd.inc.gen

dat.gen.st <- dat.gen %>%
  select(tract_id, county_name, turnout, election, isBmore, median_income, pct_over_65, pct_male, pct_un
  mutate(turnout = scale(turnout),
         median_income = scale(median_income),
         pct_over_65 = scale(pct_over_65),
         pct_male = scale(pct_male),
         pct_unemployed = scale(pct_unemployed),
         pct_hs = scale(pct_hs),
         pct_transit = scale(pct_transit),
         pct_aa = scale(pct_aa),
         pct_hisp = scale(pct_hisp)) %>%
  mutate(median_income_sp1 = ifelse(median_income > knot.inc.gen.st, median_income - knot.inc.gen.st, 0)

# consolidate variables for primary election models
dat.prim <- data_joined.complete.long %>%
  filter(election_type == "Primary") %>%
  select(tract_id, county_name, turnout, election, isBmore, median_income, pct_over_65, pct_male, pct_un
  mutate(median_income_sp1 = ifelse(median_income > 75, median_income - 75, 0))

# make a standardized dataset
mean.inc.prim <- mean(dat.prim$median_income)
sd.inc.prim <- sd(dat.prim$median_income)
knot.inc.prim.st <- (75 - mean.inc.prim)/sd.inc.prim

dat.prim.st <- dat.prim %>%
  select(tract_id, county_name, turnout, election, isBmore, median_income, pct_over_65, pct_male, pct_un
  mutate(turnout = scale(turnout),
         median_income = scale(median_income),
         pct_over_65 = scale(pct_over_65),
         pct_male = scale(pct_male),
         pct_unemployed = scale(pct_unemployed),
         pct_hs = scale(pct_hs),
         pct_transit = scale(pct_transit),
         pct_aa = scale(pct_aa),
         pct_hisp = scale(pct_hisp)) %>%
  mutate(median_income_sp1 = ifelse(median_income > knot.inc.prim.st, median_income - knot.inc.prim.st,
```

I also wrote a function that gets 95% confidence intervals for linear combinations of fixed effects estimates in lme models.

```
confint.lme <- function(mod, ind, alpha) {

  coefs <- summary(mod)$coefficients %>%
    as.data.frame() %>%
    select(Estimate)

  z_crit <- qnorm(1-(alpha/2)) # assumes sample size is large enough to use z instead of t distribution
  cov_mat <- vcov(mod)

  outp <- matrix(rep(NA,ncol(ind)*3), nrow = ncol(ind))

  for (j in 1:ncol(ind)) {

    tmp.ind <- ind[,j]
    tmp.est <- t(coefs) %*% tmp.ind
    tmp.se <- sqrt(diag(t(tmp.ind) %*% cov_mat %*% tmp.ind))
    tmp.ci <- c(tmp.est - z_crit*tmp.se, tmp.est + z_crit*tmp.se)

    outp[j,1] <- tmp.est[1]
    outp[j,2:3] <- tmp.ci

  }

  return(outp)

}
```

General elections

Single predictor models (LME)

Below I fit univariate LME models for each predictor on voter turnout by election year and `isBmore` status. I included random intercepts for census tracts nested within counties. A summary table of the unadjusted coefficient estimates are shown after this code block.

```
################ median income ################
# fit model
gen_lme.unadj.medinc <- lmer(turnout ~ election*isBmore*(median_income + median_income_sp1) + (1|county_

# get model summary
gen_lme.unadj.medinc.summary <- summary(gen_lme.unadj.medinc)
ncoef <- nrow(gen_lme.unadj.medinc.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*8), nrow = ncoef, ncol = 8)
ind[4,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(4,7),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(4,7,9),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(4,7,9,11),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(4,5),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(4,5,7,8),6] <- 1 # median_income >= 75 & !bmore & 2022
ind[c(4,5,9:10),7] <- 1 # median_income >= 75 & bmore & 2020
```

```
ind[c(4,5,7:12),8] <- 1 # median_income >= 75 & bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.medinc <- confint.lme(gen_lme.unadj.medinc, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.medinc) <- c("Est", "Lower", "Upper")
gen_ci.unadj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
gen_ci.unadj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
gen_ci.unadj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################

# fit model
gen_lme.unadj.over65 <- lmer(turnout ~ election*isBmore*(pct_over_65) + (1|county_name/tract_id), dat =

# get model summary
gen_lme.unadj.over65.summary <- summary(gen_lme.unadj.over65)
ncoef <- nrow(gen_lme.unadj.over65.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.over65 <- confint.lme(gen_lme.unadj.over65, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.over65) <- c("Est", "Lower", "Upper")
gen_ci.unadj.over65$variable <- c(rep("pct_over_65", 4))
gen_ci.unadj.over65$bmore <- c(0,0,1,1)
gen_ci.unadj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################

# fit model
gen_lme.unadj.pctmale <- lmer(turnout ~ election*isBmore*(pct_male) + (1|county_name/tract_id), dat = da

# get model summary
gen_lme.unadj.pctmale.summary <- summary(gen_lme.unadj.pctmale)
ncoef <- nrow(gen_lme.unadj.pctmale.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
```

```
gen_ci.unadj.pctmale <- confint.lme(gen_lme.unadj.pctmale, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.pctmale) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pctmale$variable <- c(rep("pct_male", 4))
gen_ci.unadj.pctmale$bmore <- c(0,0,1,1)
gen_ci.unadj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################

# fit model
gen_lme.unadj.pctunemp <- lmer(turnout ~ election*isBmore*(pct_unemployed) + (1|county_name/tract_id), d

# get model summary
gen_lme.unadj.pctunemp.summary <- summary(gen_lme.unadj.pctunemp)
ncoef <- nrow(gen_lme.unadj.pctunemp.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.pctunemp <- confint.lme(gen_lme.unadj.pctunemp, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.pctunemp) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pctunemp$variable <- c(rep("pct_unemployed", 4))
gen_ci.unadj.pctunemp$bmore <- c(0,0,1,1)
gen_ci.unadj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################

# fit model
gen_lme.unadj.pcths <- lmer(turnout ~ election*isBmore*(pct_hs) + (1|county_name/tract_id), dat = dat.ge

# get model summary
gen_lme.unadj.pcths.summary <- summary(gen_lme.unadj.pcths)
ncoef <- nrow(gen_lme.unadj.pcths.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.pcths <- confint.lme(gen_lme.unadj.pcths, ind, 0.05) %>%
  as.data.frame()
```

```r
colnames(gen_ci.unadj.pcths) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pcths$variable <- c(rep("pct_hs", 4))
gen_ci.unadj.pcths$bmore <- c(0,0,1,1)
gen_ci.unadj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################

# fit model
gen_lme.unadj.pcttransit <- lmer(turnout ~ election*isBmore*(pct_transit) + (1|county_name/tract_id), da

# get model summary
gen_lme.unadj.pcttransit.summary <- summary(gen_lme.unadj.pcttransit)
ncoef <- nrow(gen_lme.unadj.pcttransit.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.pcttransit <- confint.lme(gen_lme.unadj.pcttransit, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.pcttransit) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pcttransit$variable <- c(rep("pct_transit", 4))
gen_ci.unadj.pcttransit$bmore <- c(0,0,1,1)
gen_ci.unadj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################

# fit model
gen_lme.unadj.pctaa <- lmer(turnout ~ election*isBmore*(pct_aa) + (1|county_name/tract_id), dat = dat.g

# get model summary
gen_lme.unadj.pctaa.summary <- summary(gen_lme.unadj.pctaa)
ncoef <- nrow(gen_lme.unadj.pctaa.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.pctaa<- confint.lme(gen_lme.unadj.pctaa, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.pctaa) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pctaa$variable <- c(rep("pct_aa", 4))
gen_ci.unadj.pctaa$bmore <- c(0,0,1,1)
```

```
gen_ci.unadj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################

# fit model
gen_lme.unadj.pcthisp <- lmer(turnout ~ election*isBmore*(pct_hisp) + (1|county_name/tract_id), dat = da

# get model summary
gen_lme.unadj.pcthisp.summary <- summary(gen_lme.unadj.pcthisp)
ncoef <- nrow(gen_lme.unadj.pcthisp.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
gen_ci.unadj.pcthisp <- confint.lme(gen_lme.unadj.pcthisp, ind, 0.05) %>%
  as.data.frame()

colnames(gen_ci.unadj.pcthisp) <- c("Est", "Lower", "Upper")
gen_ci.unadj.pcthisp$variable <- c(rep("pct_hisp", 4))
gen_ci.unadj.pcthisp$bmore <- c(0,0,1,1)
gen_ci.unadj.pcthisp$year <- rep(c(2020,2022), 2)
```

I've created a table displaying the unadjusted parameter estimates below:

```
gen_ci.unadj <- rbind(gen_ci.unadj.medinc, gen_ci.unadj.over65, gen_ci.unadj.pctaa, gen_ci.unadj.pcthisp
                  gen_ci.unadj.pcths, gen_ci.unadj.pctmale, gen_ci.unadj.pcttransit, gen_ci.unadj.pctunc

gen_ci.unadj[,1:3] <- round(gen_ci.unadj[,1:3],3)
gen_ci.unadj
```

```
##       Est  Lower  Upper           variable bmore year
## 1   0.432  0.384  0.480  median_income < 75     0 2020
## 2   0.365  0.316  0.413  median_income < 75     0 2022
## 3   0.391  0.324  0.457  median_income < 75     1 2020
## 4   0.491  0.435  0.546  median_income < 75     1 2022
## 5   0.094  0.084  0.103 median_income >= 75     0 2020
## 6   0.126  0.116  0.135 median_income >= 75     0 2022
## 7   0.115  0.073  0.157 median_income >= 75     1 2020
## 8   0.162  0.120  0.204 median_income >= 75     1 2022
## 9   0.366  0.305  0.427         pct_over_65     0 2020
## 10  0.655  0.594  0.716         pct_over_65     0 2022
## 11  0.171  0.016  0.325         pct_over_65     1 2020
## 12  0.130 -0.025  0.284         pct_over_65     1 2022
## 13 -0.096 -0.117 -0.075              pct_aa     0 2020
## 14 -0.211 -0.232 -0.190              pct_aa     0 2022
## 15 -0.210 -0.242 -0.177              pct_aa     1 2020
## 16 -0.270 -0.302 -0.237              pct_aa     1 2022
## 17 -0.383 -0.458 -0.308            pct_hisp     0 2020
## 18 -0.764 -0.839 -0.689            pct_hisp     0 2022
```

```
## 19  0.078 -0.333  0.490           pct_hisp     1 2020
## 20  0.308 -0.103  0.720           pct_hisp     1 2022
## 21 -0.438 -0.488 -0.389             pct_hs     0 2020
## 22 -0.462 -0.511 -0.412             pct_hs     0 2022
## 23 -0.624 -0.710 -0.538             pct_hs     1 2020
## 24 -0.774 -0.860 -0.688             pct_hs     1 2022
## 25  0.050 -0.147  0.247           pct_male     0 2020
## 26  0.749  0.552  0.945           pct_male     0 2022
## 27  0.182 -0.176  0.539           pct_male     1 2020
## 28  0.633  0.275  0.990           pct_male     1 2022
## 29 -0.304 -0.402 -0.206        pct_transit     0 2020
## 30 -0.645 -0.742 -0.547        pct_transit     0 2022
## 31 -0.561 -0.657 -0.464        pct_transit     1 2020
## 32 -0.626 -0.723 -0.530        pct_transit     1 2022
## 33 -0.603 -0.753 -0.452     pct_unemployed     0 2020
## 34 -1.073 -1.224 -0.923     pct_unemployed     0 2022
## 35 -0.963 -1.152 -0.774     pct_unemployed     1 2020
## 36 -1.060 -1.249 -0.872     pct_unemployed     1 2022
```

Adjusted model (LME)

```r
# fit adjusted model for general elections
gen_lme.adj <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + median_i

gen_lme.adj.summary <- summary(gen_lme.adj)
gen_lme.adj.summary
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: turnout ~ isBmore * election * (pct_aa + pct_hisp + pct_male +
##     median_income + median_income_sp1 + pct_over_65 + pct_unemployed +
##     pct_hs + pct_transit) + (1 | county_name/tract_id)
##    Data: dat.gen
##
##      AIC      BIC   logLik deviance df.resid
##  15593.4  15850.4  -7753.7  15507.4     2865
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0824 -0.4570  0.0005  0.4623  2.7939
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  tract_id:county_name (Intercept) 14.611   3.822
##  county_name          (Intercept)  5.469   2.339
##  Residual                          4.218   2.054
## Number of obs: 2908, groups:  tract_id:county_name, 1454; county_name, 24
##
## Fixed effects:
##                                                  Estimate Std. Error         df
## (Intercept)                                     5.673e+01  3.616e+00  1.735e+03
## isBmoreTRUE                                     2.838e+01  8.003e+00  1.054e+03
## electiongen_2022                               -3.855e+01  2.252e+00  1.454e+03
## pct_aa                                         -7.514e-03  8.508e-03  1.592e+03
```

```
## pct_hisp                                   -1.684e-01  2.373e-02  1.666e+03
## pct_male                                   -1.990e-01  6.913e-02  1.773e+03
## median_income                               3.427e-01  1.822e-02  1.739e+03
## median_income_sp1                          -2.885e-01  1.916e-02  1.775e+03
## pct_over_65                                 2.629e-01  1.901e-02  1.737e+03
## pct_unemployed                             -1.238e-01  4.430e-02  1.786e+03
## pct_hs                                     -1.527e-01  1.841e-02  1.772e+03
## pct_transit                                -6.567e-02  2.935e-02  1.734e+03
## isBmoreTRUE:electiongen_2022                8.286e+00  5.046e+00  1.454e+03
## isBmoreTRUE:pct_aa                         -9.311e-02  2.202e-02  1.776e+03
## isBmoreTRUE:pct_hisp                       -7.246e-01  1.518e-01  1.783e+03
## isBmoreTRUE:pct_male                       -5.344e-01  1.365e-01  1.781e+03
## isBmoreTRUE:median_income                  -4.701e-02  3.374e-02  1.794e+03
## isBmoreTRUE:median_income_sp1               6.048e-02  3.985e-02  1.790e+03
## isBmoreTRUE:pct_over_65                     1.156e-02  5.184e-02  1.787e+03
## isBmoreTRUE:pct_unemployed                 -3.271e-02  7.795e-02  1.783e+03
## isBmoreTRUE:pct_hs                         -4.510e-02  4.655e-02  1.784e+03
## isBmoreTRUE:pct_transit                     4.795e-04  4.577e-02  1.775e+03
## electiongen_2022:pct_aa                    -7.904e-02  4.278e-03  1.454e+03
## electiongen_2022:pct_hisp                  -3.476e-01  1.358e-02  1.454e+03
## electiongen_2022:pct_male                   3.642e-01  4.423e-02  1.454e+03
## electiongen_2022:median_income             -5.053e-02  1.140e-02  1.454e+03
## electiongen_2022:median_income_sp1          4.376e-02  1.220e-02  1.454e+03
## electiongen_2022:pct_over_65                1.824e-01  1.167e-02  1.454e+03
## electiongen_2022:pct_unemployed            -7.664e-02  2.926e-02  1.454e+03
## electiongen_2022:pct_hs                    -3.944e-02  1.163e-02  1.454e+03
## electiongen_2022:pct_transit                6.008e-02  1.753e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_aa         2.595e-02  1.425e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_hisp       1.417e-01  1.013e-01  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_male      -1.102e-01  9.035e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:median_income  3.757e-02  2.217e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:median_income_sp1 -3.292e-02  2.638e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_over_65   -1.120e-01  3.433e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_unemployed  1.175e-01  5.196e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_hs        -2.771e-02  3.089e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_transit   -3.806e-02  2.932e-02  1.454e+03
##                                            t value Pr(>|t|)
## (Intercept)                                 15.689  < 2e-16 ***
## isBmoreTRUE                                  3.547 0.000408 ***
## electiongen_2022                           -17.119  < 2e-16 ***
## pct_aa                                      -0.883 0.377285
## pct_hisp                                    -7.098 1.87e-12 ***
## pct_male                                    -2.879 0.004039 **
## median_income                               18.811  < 2e-16 ***
## median_income_sp1                          -15.056  < 2e-16 ***
## pct_over_65                                 13.831  < 2e-16 ***
## pct_unemployed                              -2.794 0.005259 **
## pct_hs                                      -8.293  < 2e-16 ***
## pct_transit                                 -2.237 0.025391 *
## isBmoreTRUE:electiongen_2022                 1.642 0.100761
## isBmoreTRUE:pct_aa                          -4.229 2.47e-05 ***
## isBmoreTRUE:pct_hisp                        -4.773 1.97e-06 ***
## isBmoreTRUE:pct_male                        -3.915 9.39e-05 ***
## isBmoreTRUE:median_income                   -1.393 0.163739
```

```
## isBmoreTRUE:median_income_sp1                      1.518 0.129242
## isBmoreTRUE:pct_over_65                            0.223 0.823506
## isBmoreTRUE:pct_unemployed                        -0.420 0.674817
## isBmoreTRUE:pct_hs                                -0.969 0.332709
## isBmoreTRUE:pct_transit                            0.010 0.991641
## electiongen_2022:pct_aa                          -18.475  < 2e-16 ***
## electiongen_2022:pct_hisp                        -25.597  < 2e-16 ***
## electiongen_2022:pct_male                          8.235 3.95e-16 ***
## electiongen_2022:median_income                    -4.431 1.01e-05 ***
## electiongen_2022:median_income_sp1                 3.587 0.000345 ***
## electiongen_2022:pct_over_65                      15.623  < 2e-16 ***
## electiongen_2022:pct_unemployed                   -2.619 0.008912 **
## electiongen_2022:pct_hs                           -3.392 0.000712 ***
## electiongen_2022:pct_transit                       3.427 0.000627 ***
## isBmoreTRUE:electiongen_2022:pct_aa                1.821 0.068757 .
## isBmoreTRUE:electiongen_2022:pct_hisp              1.399 0.162153
## isBmoreTRUE:electiongen_2022:pct_male             -1.220 0.222633
## isBmoreTRUE:electiongen_2022:median_income         1.695 0.090315 .
## isBmoreTRUE:electiongen_2022:median_income_sp1   -1.248 0.212203
## isBmoreTRUE:electiongen_2022:pct_over_65          -3.263 0.001127 **
## isBmoreTRUE:electiongen_2022:pct_unemployed        2.262 0.023824 *
## isBmoreTRUE:electiongen_2022:pct_hs               -0.897 0.369853
## isBmoreTRUE:electiongen_2022:pct_transit          -1.298 0.194488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 40 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```r
# fit standardized adjusted model for general elections
gen_lme.adj.st <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + median

gen_lme.adj.st.summary <- summary(gen_lme.adj.st)
gen_lme.adj.st.summary
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: turnout ~ isBmore * election * (pct_aa + pct_hisp + pct_male +
##     median_income + median_income_sp1 + pct_over_65 + pct_unemployed +
##     pct_hs + pct_transit) + (1 | county_name/tract_id)
##    Data: dat.gen.st
##
##      AIC      BIC   logLik deviance df.resid
##   -801.1   -544.2    443.6   -887.1     2865
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.0824 -0.4570  0.0005  0.4623  2.7939
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  tract_id:county_name (Intercept) 0.05203  0.2281
##  county_name          (Intercept) 0.01948  0.1396
```

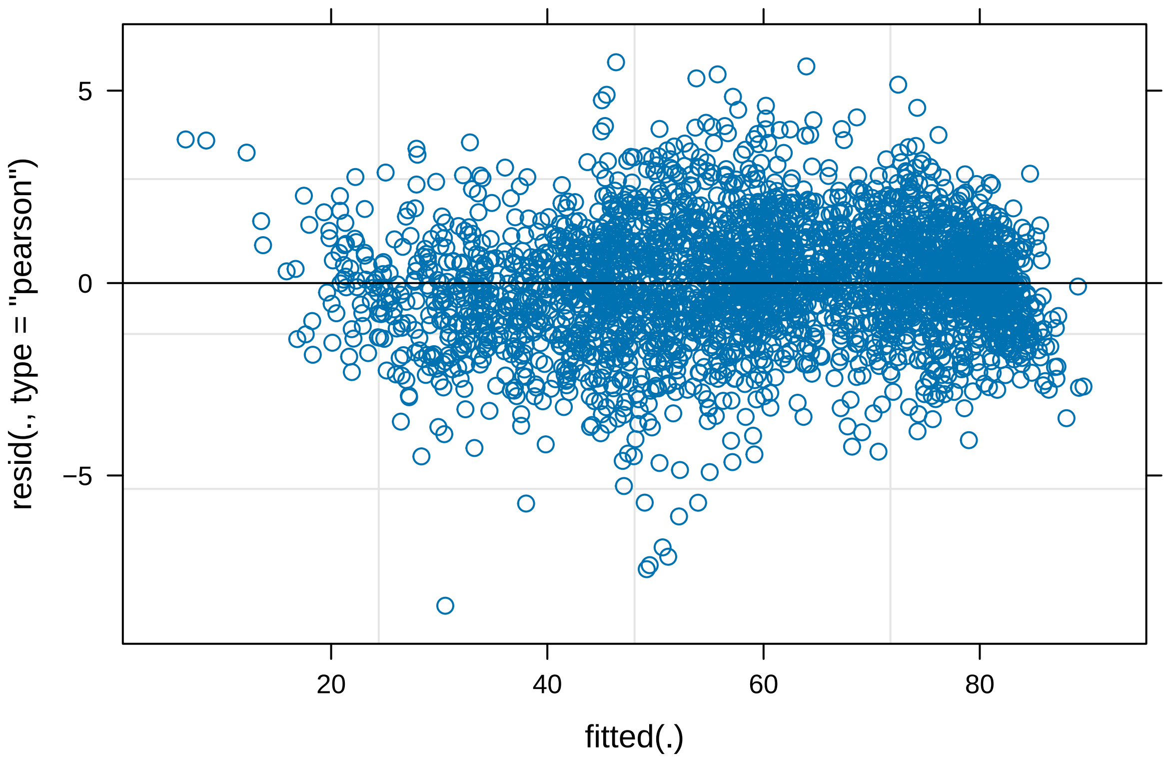
```
##  Residual                         0.01502  0.1226
## Number of obs: 2908, groups:  tract_id:county_name, 1454; county_name, 24
##
## Fixed effects:
##                                                Estimate Std. Error         df
## (Intercept)                                   1.511e+00  5.155e-02  1.269e+02
## isBmoreTRUE                                  -5.906e-01  1.705e-01  3.244e+01
## electiongen_2022                             -1.643e+00  2.505e-02  1.454e+03
## pct_aa                                       -1.431e-02  1.621e-02  1.592e+03
## pct_hisp                                     -6.875e-02  9.686e-03  1.666e+03
## pct_male                                     -3.321e-02  1.154e-02  1.773e+03
## median_income                                 9.510e-01  5.055e-02  1.739e+03
## median_income_sp1                            -8.004e-01  5.316e-02  1.775e+03
## pct_over_65                                   1.225e-01  8.858e-03  1.737e+03
## pct_unemployed                               -2.847e-02  1.019e-02  1.786e+03
## pct_hs                                       -9.604e-02  1.158e-02  1.772e+03
## pct_transit                                  -3.083e-02  1.378e-02  1.734e+03
## isBmoreTRUE:electiongen_2022                  3.325e-01  6.115e-02  1.454e+03
## isBmoreTRUE:pct_aa                           -1.774e-01  4.194e-02  1.776e+03
## isBmoreTRUE:pct_hisp                         -2.958e-01  6.198e-02  1.783e+03
## isBmoreTRUE:pct_male                         -8.917e-02  2.278e-02  1.781e+03
## isBmoreTRUE:median_income                    -1.304e-01  9.362e-02  1.794e+03
## isBmoreTRUE:median_income_sp1                 1.678e-01  1.106e-01  1.790e+03
## isBmoreTRUE:pct_over_65                       5.389e-03  2.416e-02  1.787e+03
## isBmoreTRUE:pct_unemployed                   -7.524e-03  1.793e-02  1.783e+03
## isBmoreTRUE:pct_hs                           -2.837e-02  2.928e-02  1.784e+03
## isBmoreTRUE:pct_transit                       2.251e-04  2.148e-02  1.775e+03
## electiongen_2022:pct_aa                      -1.506e-01  8.150e-03  1.454e+03
## electiongen_2022:pct_hisp                    -1.419e-01  5.543e-03  1.454e+03
## electiongen_2022:pct_male                     6.078e-02  7.381e-03  1.454e+03
## electiongen_2022:median_income               -1.402e-01  3.164e-02  1.454e+03
## electiongen_2022:median_income_sp1            1.214e-01  3.385e-02  1.454e+03
## electiongen_2022:pct_over_65                  8.499e-02  5.440e-03  1.454e+03
## electiongen_2022:pct_unemployed              -1.763e-02  6.731e-03  1.454e+03
## electiongen_2022:pct_hs                      -2.481e-02  7.314e-03  1.454e+03
## electiongen_2022:pct_transit                  2.820e-02  8.228e-03  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_aa           4.944e-02  2.715e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_hisp         5.783e-02  4.135e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_male        -1.840e-02  1.508e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:median_income    1.042e-01  6.150e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:median_income_sp1 -9.134e-02  7.319e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_over_65     -5.220e-02  1.600e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_unemployed   2.704e-02  1.195e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_hs          -1.743e-02  1.943e-02  1.454e+03
## isBmoreTRUE:electiongen_2022:pct_transit     -1.787e-02  1.376e-02  1.454e+03
##                                              t value Pr(>|t|)
## (Intercept)                                   29.299  < 2e-16 ***
## isBmoreTRUE                                   -3.463 0.001520 **
## electiongen_2022                             -65.611  < 2e-16 ***
## pct_aa                                        -0.883 0.377285
## pct_hisp                                      -7.098 1.87e-12 ***
## pct_male                                      -2.879 0.004039 **
## median_income                                 18.811  < 2e-16 ***
## median_income_sp1                            -15.056  < 2e-16 ***
```

```
## pct_over_65                                      13.831  < 2e-16 ***
## pct_unemployed                                   -2.794 0.005259 **
## pct_hs                                           -8.293  < 2e-16 ***
## pct_transit                                      -2.237 0.025391 *
## isBmoreTRUE:electiongen_2022                      5.437 6.35e-08 ***
## isBmoreTRUE:pct_aa                               -4.229 2.47e-05 ***
## isBmoreTRUE:pct_hisp                             -4.773 1.97e-06 ***
## isBmoreTRUE:pct_male                             -3.915 9.39e-05 ***
## isBmoreTRUE:median_income                        -1.393 0.163739
## isBmoreTRUE:median_income_sp1                     1.518 0.129242
## isBmoreTRUE:pct_over_65                           0.223 0.823506
## isBmoreTRUE:pct_unemployed                       -0.420 0.674817
## isBmoreTRUE:pct_hs                               -0.969 0.332709
## isBmoreTRUE:pct_transit                           0.010 0.991641
## electiongen_2022:pct_aa                         -18.475  < 2e-16 ***
## electiongen_2022:pct_hisp                       -25.597  < 2e-16 ***
## electiongen_2022:pct_male                         8.235 3.95e-16 ***
## electiongen_2022:median_income                   -4.431 1.01e-05 ***
## electiongen_2022:median_income_sp1                3.587 0.000345 ***
## electiongen_2022:pct_over_65                     15.623  < 2e-16 ***
## electiongen_2022:pct_unemployed                  -2.619 0.008912 **
## electiongen_2022:pct_hs                          -3.392 0.000712 ***
## electiongen_2022:pct_transit                      3.427 0.000627 ***
## isBmoreTRUE:electiongen_2022:pct_aa               1.821 0.068757 .
## isBmoreTRUE:electiongen_2022:pct_hisp             1.399 0.162153
## isBmoreTRUE:electiongen_2022:pct_male            -1.220 0.222633
## isBmoreTRUE:electiongen_2022:median_income        1.695 0.090315 .
## isBmoreTRUE:electiongen_2022:median_income_sp1  -1.248 0.212203
## isBmoreTRUE:electiongen_2022:pct_over_65         -3.263 0.001127 **
## isBmoreTRUE:electiongen_2022:pct_unemployed       2.262 0.023824 *
## isBmoreTRUE:electiongen_2022:pct_hs              -0.897 0.369853
## isBmoreTRUE:electiongen_2022:pct_transit         -1.298 0.194488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 40 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

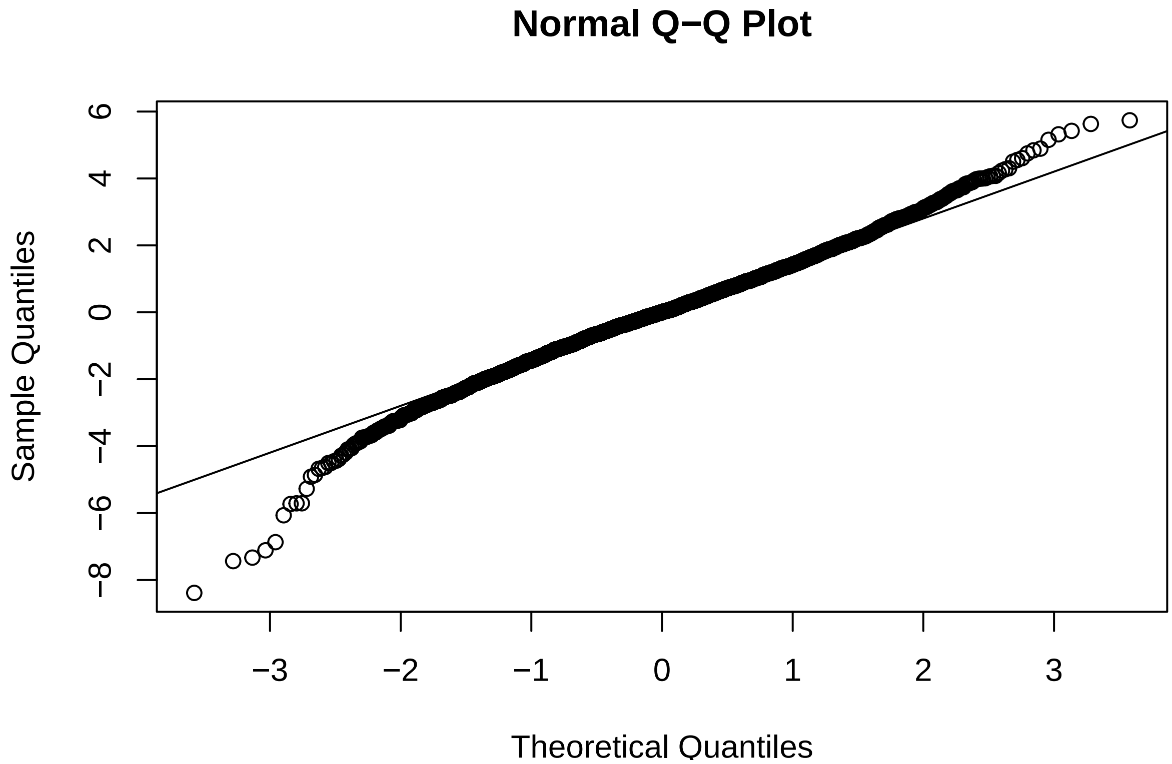
Diagnostics I want to evaluate the fit of the (standardized) LME model and assess its sensitivity to outliers and influential points.

Step 1) Plot residuals against fitted values from standardized adjusted model.

```r
plot(gen_lme.adj)
```




```r
qqnorm(resid(gen_lme.adj))
qqline(resid(gen_lme.adj))
```



Step 2) Identify census tracts with the largest residuals

```r
dat.gen.st$scaled_res <- gen_lme.adj.st.summary$residuals
dat.gen.st$fitted <- predict(gen_lme.adj.st, dat.gen.st, re.form = NULL)

dat.gen.st %>%
  select(tract_id, county_name, turnout, fitted, scaled_res) %>%
  arrange(desc(abs(scaled_res))) %>%
  head(15)
```

```
## # A tibble: 15 x 5
##    tract_id    county_name            turnout[,1] fitted scaled_res
##    <chr>       <chr>                        <dbl>  <dbl>      <dbl>
##  1 24033980000 Prince George's County     -2.23   -1.73       -4.08
##  2 24031701221 Montgomery County          -1.06   -0.613      -3.62
##  3 24005492500 Baltimore County           -1.03   -0.597      -3.57
##  4 24033805602 Prince George's County     -0.919  -0.495      -3.46
##  5 24031700608 Montgomery County          -0.935  -0.525      -3.34
##  6 24031700905 Montgomery County          -0.797  -0.435      -2.95
##  7 24033804700 Prince George's County     -0.440  -0.783       2.79
##  8 24031701425 Montgomery County          -1.62   -1.28       -2.79
##  9 24031701220 Montgomery County          -0.670  -0.330      -2.78
## 10 24033805601 Prince George's County     -0.965  -0.624      -2.78
## 11 24031702401 Montgomery County           0.604   0.268       2.74
## 12 24021750100 Frederick County            0.102  -0.222       2.64
## 13 24510130805 Baltimore city             -0.0215 -0.339       2.59
## 14 24510272005 Baltimore city             -1.05   -0.739      -2.57
## 15 24031701221 Montgomery County           1.08    0.775       2.51
```

Step 2) Use Cook's distance to identify influential data points.

```r
dat.gen.st$cooks_distance <- cooks.distance(gen_lme.adj.st)

# identify most influential points
dat.gen.st %>%
  arrange(desc(cooks_distance)) %>%
  select(tract_id, county_name, election, cooks_distance, fitted, scaled_res) %>%
  head()
```

```
## # A tibble: 6 x 6
##   tract_id    county_name            election cooks_distance fitted scaled_res
##   <chr>       <chr>                  <chr>             <dbl>  <dbl>      <dbl>
## 1 24033980000 Prince George's County gen_2020          0.715 -1.73       -4.08
## 2 24005492500 Baltimore County       gen_2020          0.563 -0.597      -3.57
## 3 24033805602 Prince George's County gen_2020          0.469 -0.495      -3.46
## 4 24031701221 Montgomery County      gen_2022          0.458 -0.613      -3.62
## 5 24031700608 Montgomery County      gen_2022          0.401 -0.525      -3.34
## 6 24033805601 Prince George's County gen_2020          0.320 -0.624      -2.78
```

Step 3) Compare model excluding the most influential census tracts to the original model

I will now refit the standardized lme model excluding the 2 census tracts with cooks distance > 0.5, and compare it to the original model fit.

```r
tracts_to_exclude <- c(24033980000, 24005492500)
dat.gen.st.ex <- dat.gen.st %>%
  filter(!tract_id %in% tracts_to_exclude)

# refit model
```

```r
gen_lme.adj.st.ex <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + med

gen_lme.adj.st.ex.summary <- summary(gen_lme.adj.st.ex)

# compare new coefficient estimates with original ones
gen_lme.adj.st.t <- round(cbind(gen_lme.adj.st.summary$coefficients[,c(1,5)],
                           gen_lme.adj.st.ex.summary$coefficients[,c(1,5)]),3)

gen_lme.adj.st.t
```

```
##                                               Estimate Pr(>|t|) Estimate
## (Intercept)                                      1.511    0.000    1.468
## isBmoreTRUE                                     -0.591    0.002   -0.548
## electiongen_2022                                -1.643    0.000   -1.631
## pct_aa                                          -0.014    0.377   -0.011
## pct_hisp                                        -0.069    0.000   -0.077
## pct_male                                        -0.033    0.004   -0.010
## median_income                                    0.951    0.000    0.891
## median_income_sp1                               -0.800    0.000   -0.752
## pct_over_65                                      0.123    0.000    0.121
## pct_unemployed                                  -0.028    0.005   -0.028
## pct_hs                                          -0.096    0.000   -0.102
## pct_transit                                     -0.031    0.025   -0.031
## isBmoreTRUE:electiongen_2022                     0.332    0.000    0.321
## isBmoreTRUE:pct_aa                              -0.177    0.000   -0.181
## isBmoreTRUE:pct_hisp                            -0.296    0.000   -0.287
## isBmoreTRUE:pct_male                            -0.089    0.000   -0.112
## isBmoreTRUE:median_income                       -0.130    0.164   -0.070
## isBmoreTRUE:median_income_sp1                    0.168    0.129    0.119
## isBmoreTRUE:pct_over_65                          0.005    0.824    0.007
## isBmoreTRUE:pct_unemployed                      -0.008    0.675   -0.008
## isBmoreTRUE:pct_hs                              -0.028    0.333   -0.023
## isBmoreTRUE:pct_transit                          0.000    0.992    0.000
## electiongen_2022:pct_aa                         -0.151    0.000   -0.155
## electiongen_2022:pct_hisp                       -0.142    0.000   -0.140
## electiongen_2022:pct_male                        0.061    0.000    0.050
## electiongen_2022:median_income                  -0.140    0.000   -0.117
## electiongen_2022:median_income_sp1               0.121    0.000    0.102
## electiongen_2022:pct_over_65                     0.085    0.000    0.085
## electiongen_2022:pct_unemployed                 -0.018    0.009   -0.018
## electiongen_2022:pct_hs                         -0.025    0.001   -0.021
## electiongen_2022:pct_transit                     0.028    0.001    0.026
## isBmoreTRUE:electiongen_2022:pct_aa              0.049    0.069    0.054
## isBmoreTRUE:electiongen_2022:pct_hisp            0.058    0.162    0.056
## isBmoreTRUE:electiongen_2022:pct_male           -0.018    0.223   -0.008
## isBmoreTRUE:electiongen_2022:median_income       0.104    0.090    0.081
## isBmoreTRUE:electiongen_2022:median_income_sp1  -0.091    0.212   -0.072
## isBmoreTRUE:electiongen_2022:pct_over_65        -0.052    0.001   -0.053
## isBmoreTRUE:electiongen_2022:pct_unemployed      0.027    0.024    0.027
## isBmoreTRUE:electiongen_2022:pct_hs             -0.017    0.370   -0.021
## isBmoreTRUE:electiongen_2022:pct_transit        -0.018    0.194   -0.016
##                                               Pr(>|t|)
## (Intercept)                                      0.000
## isBmoreTRUE                                      0.002
```

```
## electiongen_2022                                    0.000
## pct_aa                                              0.506
## pct_hisp                                            0.000
## pct_male                                            0.397
## median_income                                       0.000
## median_income_sp1                                   0.000
## pct_over_65                                         0.000
## pct_unemployed                                      0.005
## pct_hs                                              0.000
## pct_transit                                         0.023
## isBmoreTRUE:electiongen_2022                        0.000
## isBmoreTRUE:pct_aa                                  0.000
## isBmoreTRUE:pct_hisp                                0.000
## isBmoreTRUE:pct_male                                0.000
## isBmoreTRUE:median_income                           0.452
## isBmoreTRUE:median_income_sp1                       0.278
## isBmoreTRUE:pct_over_65                             0.768
## isBmoreTRUE:pct_unemployed                          0.658
## isBmoreTRUE:pct_hs                                  0.431
## isBmoreTRUE:pct_transit                             0.985
## electiongen_2022:pct_aa                             0.000
## electiongen_2022:pct_hisp                           0.000
## electiongen_2022:pct_male                           0.000
## electiongen_2022:median_income                      0.000
## electiongen_2022:median_income_sp1                  0.003
## electiongen_2022:pct_over_65                        0.000
## electiongen_2022:pct_unemployed                     0.007
## electiongen_2022:pct_hs                             0.004
## electiongen_2022:pct_transit                        0.001
## isBmoreTRUE:electiongen_2022:pct_aa                 0.045
## isBmoreTRUE:electiongen_2022:pct_hisp               0.174
## isBmoreTRUE:electiongen_2022:pct_male               0.612
## isBmoreTRUE:electiongen_2022:median_income          0.186
## isBmoreTRUE:electiongen_2022:median_income_sp1      0.324
## isBmoreTRUE:electiongen_2022:pct_over_65            0.001
## isBmoreTRUE:electiongen_2022:pct_unemployed         0.021
## isBmoreTRUE:electiongen_2022:pct_hs                 0.276
## isBmoreTRUE:electiongen_2022:pct_transit            0.245
```

We can see that, while the estimates do change a tiny bit after removing the 2 influential census tracts, our key inferences from the model remain almost entirely the same, suggesting robustness to the inclusion of these census tracts.

Model Summary

```r
gen_lme.adj.coefs <- summary(gen_lme.adj)$coefficients %>%
  as.data.frame() %>%
  select(Estimate)

gen_lme.adj.coefs$idx <- 1:nrow(gen_lme.adj.coefs)
gen_lme.adj.coefs
```

Unstandardized Model

```
##                                                        Estimate idx
```

```
## (Intercept)                                   5.673138e+01   1
## isBmoreTRUE                                   2.838299e+01   2
## electiongen_2022                             -3.855109e+01   3
## pct_aa                                       -7.513597e-03   4
## pct_hisp                                     -1.684112e-01   5
## pct_male                                     -1.990024e-01   6
## median_income                                 3.427223e-01   7
## median_income_sp1                            -2.884532e-01   8
## pct_over_65                                   2.628935e-01   9
## pct_unemployed                               -1.237881e-01  10
## pct_hs                                       -1.526785e-01  11
## pct_transit                                  -6.566908e-02  12
## isBmoreTRUE:electiongen_2022                  8.286096e+00  13
## isBmoreTRUE:pct_aa                           -9.310761e-02  14
## isBmoreTRUE:pct_hisp                         -7.245829e-01  15
## isBmoreTRUE:pct_male                         -5.343565e-01  16
## isBmoreTRUE:median_income                    -4.700564e-02  17
## isBmoreTRUE:median_income_sp1                 6.047703e-02  18
## isBmoreTRUE:pct_over_65                       1.156323e-02  19
## isBmoreTRUE:pct_unemployed                   -3.271100e-02  20
## isBmoreTRUE:pct_hs                           -4.510094e-02  21
## isBmoreTRUE:pct_transit                       4.795479e-04  22
## electiongen_2022:pct_aa                      -7.904043e-02  23
## electiongen_2022:pct_hisp                    -3.475742e-01  24
## electiongen_2022:pct_male                     3.642319e-01  25
## electiongen_2022:median_income               -5.052793e-02  26
## electiongen_2022:median_income_sp1            4.376128e-02  27
## electiongen_2022:pct_over_65                  1.823895e-01  28
## electiongen_2022:pct_unemployed              -7.663923e-02  29
## electiongen_2022:pct_hs                      -3.944167e-02  30
## electiongen_2022:pct_transit                  6.007521e-02  31
## isBmoreTRUE:electiongen_2022:pct_aa           2.595440e-02  32
## isBmoreTRUE:electiongen_2022:pct_hisp         1.416609e-01  33
## isBmoreTRUE:electiongen_2022:pct_male        -1.102325e-01  34
## isBmoreTRUE:electiongen_2022:median_income    3.756786e-02  35
## isBmoreTRUE:electiongen_2022:median_income_sp1 -3.291962e-02  36
## isBmoreTRUE:electiongen_2022:pct_over_65     -1.120192e-01  37
## isBmoreTRUE:electiongen_2022:pct_unemployed   1.175472e-01  38
## isBmoreTRUE:electiongen_2022:pct_hs          -2.770734e-02  39
## isBmoreTRUE:electiongen_2022:pct_transit     -3.805806e-02  40
```

```
################ median income ###############

# isbmore and gen_2020
ind <- matrix(0,nrow=nrow(gen_lme.adj.coefs),ncol=8)
ind[7,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(7,26),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(7,17),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(7,17,26,35),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(7,8),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(7,8,26,27),6] <- 1 # median_income >= 75 & !bmore & 2022
ind[c(7,8,17,18),7] <- 1 # median_income >= 75 & bmore & 2020
ind[c(7,8,17,18,26,27,35,36),8] <- 1 # median_income >= 75 & bmore & 2022
```

```r
# get confidence intervals for desired linear combination of estimates
ci.adj.medinc <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.medinc) <- c("Est", "Lower", "Upper")
ci.adj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
ci.adj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
ci.adj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[9,1] <- 1 # !bmore & 2020
ind[c(9,28),2] <- 1 # !bmore & 2022
ind[c(9,19),3] <- 1 # bmore & 2020
ind[c(9,19,28,37),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.over65 <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.over65) <- c("Est", "Lower", "Upper")
ci.adj.over65$variable <- c(rep("pct_over_65", 4))
ci.adj.over65$bmore <- c(0,0,1,1)
ci.adj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[6,1] <- 1 # !bmore & 2020
ind[c(6,25),2] <- 1 # !bmore & 2022
ind[c(6,16),3] <- 1 # bmore & 2020
ind[c(6,16,25,34),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctmale <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctmale) <- c("Est", "Lower", "Upper")
ci.adj.pctmale$variable <- c(rep("pct_male", 4))
ci.adj.pctmale$bmore <- c(0,0,1,1)
ci.adj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[10,1] <- 1 # !bmore & 2020
ind[c(10,29),2] <- 1 # !bmore & 2022
ind[c(10,20),3] <- 1 # bmore & 2020
ind[c(10,20,29,38),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctunemp <- confint.lme(gen_lme.adj, ind, 0.05) %>%
```

```r
  as.data.frame()

colnames(ci.adj.pctunemp) <- c("Est", "Lower", "Upper")
ci.adj.pctunemp$variable <- c(rep("pct_unemployed", 4))
ci.adj.pctunemp$bmore <- c(0,0,1,1)
ci.adj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[11,1] <- 1 # !bmore & 2020
ind[c(11,30),2] <- 1 # !bmore & 2022
ind[c(11,21),3] <- 1 # bmore & 2020
ind[c(11,21,30,39),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcths <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcths) <- c("Est", "Lower", "Upper")
ci.adj.pcths$variable <- c(rep("pct_hs", 4))
ci.adj.pcths$bmore <- c(0,0,1,1)
ci.adj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[12,1] <- 1 # !bmore & 2020
ind[c(12,31),2] <- 1 # !bmore & 2022
ind[c(12,22),3] <- 1 # bmore & 2020
ind[c(12,22,31,40),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcttransit <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcttransit) <- c("Est", "Lower", "Upper")
ci.adj.pcttransit$variable <- c(rep("pct_transit", 4))
ci.adj.pcttransit$bmore <- c(0,0,1,1)
ci.adj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,23),2] <- 1 # !bmore & 2022
ind[c(4,14),3] <- 1 # bmore & 2020
ind[c(4,14,23,32),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctaa <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()
```

```r
colnames(ci.adj.pctaa) <- c("Est", "Lower", "Upper")
ci.adj.pctaa$variable <- c(rep("pct_aa", 4))
ci.adj.pctaa$bmore <- c(0,0,1,1)
ci.adj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.coefs), ncol = 4)
ind[5,1] <- 1 # !bmore & 2020
ind[c(5,24),2] <- 1 # !bmore & 2022
ind[c(5,15),3] <- 1 # bmore & 2020
ind[c(5,15,24,33),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcthisp <- confint.lme(gen_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcthisp) <- c("Est", "Lower", "Upper")
ci.adj.pcthisp$variable <- c(rep("pct_hisp", 4))
ci.adj.pcthisp$bmore <- c(0,0,1,1)
ci.adj.pcthisp$year <- rep(c(2020,2022), 2)

################ combine tables ################
gen_ci.adj <- rbind(ci.adj.medinc, ci.adj.over65, ci.adj.pctaa, ci.adj.pcthisp,
                  ci.adj.pcths, ci.adj.pctmale, ci.adj.pcttransit, ci.adj.pctunemp)

gen_ci.adj[,1:3] <- round(gen_ci.adj[,1:3],3)
gen_ci.adj
```

```
##       Est  Lower  Upper           variable bmore year
## 1   0.343  0.307  0.378  median_income < 75     0 2020
## 2   0.292  0.256  0.328  median_income < 75     0 2022
## 3   0.296  0.240  0.351  median_income < 75     1 2020
## 4   0.283  0.227  0.338  median_income < 75     1 2022
## 5   0.054  0.045  0.064 median_income >= 75     0 2020
## 6   0.048  0.038  0.057 median_income >= 75     0 2022
## 7   0.068  0.031  0.104 median_income >= 75     1 2020
## 8   0.066  0.029  0.102 median_income >= 75     1 2022
## 9   0.263  0.226  0.300         pct_over_65     0 2020
## 10  0.445  0.408  0.483         pct_over_65     0 2022
## 11  0.274  0.180  0.369         pct_over_65     1 2020
## 12  0.345  0.250  0.439         pct_over_65     1 2022
## 13 -0.008 -0.024  0.009              pct_aa     0 2020
## 14 -0.087 -0.103 -0.070              pct_aa     0 2022
## 15 -0.101 -0.140 -0.061              pct_aa     1 2020
## 16 -0.154 -0.194 -0.114              pct_aa     1 2022
## 17 -0.168 -0.215 -0.122            pct_hisp     0 2020
## 18 -0.516 -0.562 -0.469            pct_hisp     0 2022
## 19 -0.893 -1.187 -0.599            pct_hisp     1 2020
## 20 -1.099 -1.393 -0.805            pct_hisp     1 2022
## 21 -0.153 -0.189 -0.117              pct_hs     0 2020
## 22 -0.192 -0.228 -0.156              pct_hs     0 2022
## 23 -0.198 -0.282 -0.114              pct_hs     1 2020
## 24 -0.265 -0.349 -0.181              pct_hs     1 2022
```

```
## 25 -0.199 -0.334 -0.064           pct_male     0 2020
## 26  0.165  0.030  0.301           pct_male     0 2022
## 27 -0.733 -0.964 -0.503           pct_male     1 2020
## 28 -0.479 -0.710 -0.249           pct_male     1 2022
## 29 -0.066 -0.123 -0.008        pct_transit     0 2020
## 30 -0.006 -0.063  0.052        pct_transit     0 2022
## 31 -0.065 -0.134  0.004        pct_transit     1 2020
## 32 -0.043 -0.112  0.026        pct_transit     1 2022
## 33 -0.124 -0.211 -0.037     pct_unemployed     0 2020
## 34 -0.200 -0.287 -0.114     pct_unemployed     0 2022
## 35 -0.156 -0.282 -0.031     pct_unemployed     1 2020
## 36 -0.116 -0.241  0.010     pct_unemployed     1 2022
```

```
# define controls for plotting model fits
bmore_dat <- dat.gen %>%
  filter(isBmore == 1, election == "gen_2020")

controls <- bmore_dat %>%
  summarize(med_median_income = median(median_income),
            max_median_income = max(median_income),
            min_median_income = min(median_income),
            med_pct_aa = median(pct_aa),
            max_pct_aa = max(pct_aa),
            min_pct_aa = min(pct_aa),
            med_pct_hisp = median(pct_hisp),
            max_pct_hisp = max(pct_hisp),
            min_pct_hisp = min(pct_hisp),
            med_pct_over_65 = median(pct_over_65),
            max_pct_over_65 = max(pct_over_65),
            min_pct_over_65 = min(pct_over_65),
            med_pct_male = median(pct_male),
            max_pct_male = max(pct_male),
            min_pct_male = min(pct_male),
            med_pct_unemployed = median(pct_unemployed),
            max_pct_unemployed = max(pct_unemployed),
            min_pct_unemployed = min(pct_unemployed),
            med_pct_hs = median(pct_hs),
            max_pct_hs = max(pct_hs),
            min_pct_hs = min(pct_hs),
            med_pct_transit = median(pct_transit),
            max_pct_transit = max(pct_transit),
            min_pct_transit = min(pct_transit))

controls
```

```
## # A tibble: 1 x 24
##   med_median_income max_median_income min_median_income med_pct_aa max_pct_aa
##               <dbl>             <dbl>             <dbl>      <dbl>      <dbl>
## 1              54.4              230.              13.0       79.9       93.8
## # i 19 more variables: min_pct_aa <dbl>, med_pct_hisp <dbl>,
## #   max_pct_hisp <dbl>, min_pct_hisp <dbl>, med_pct_over_65 <dbl>,
## #   max_pct_over_65 <dbl>, min_pct_over_65 <dbl>, med_pct_male <dbl>,
## #   max_pct_male <dbl>, min_pct_male <dbl>, med_pct_unemployed <dbl>,
## #   max_pct_unemployed <dbl>, min_pct_unemployed <dbl>, med_pct_hs <dbl>,
## #   max_pct_hs <dbl>, min_pct_hs <dbl>, med_pct_transit <dbl>,
```

```
## #   max_pct_transit <dbl>, min_pct_transit <dbl>
```

I wrote a function below for getting synthetic data for plotting model fits.

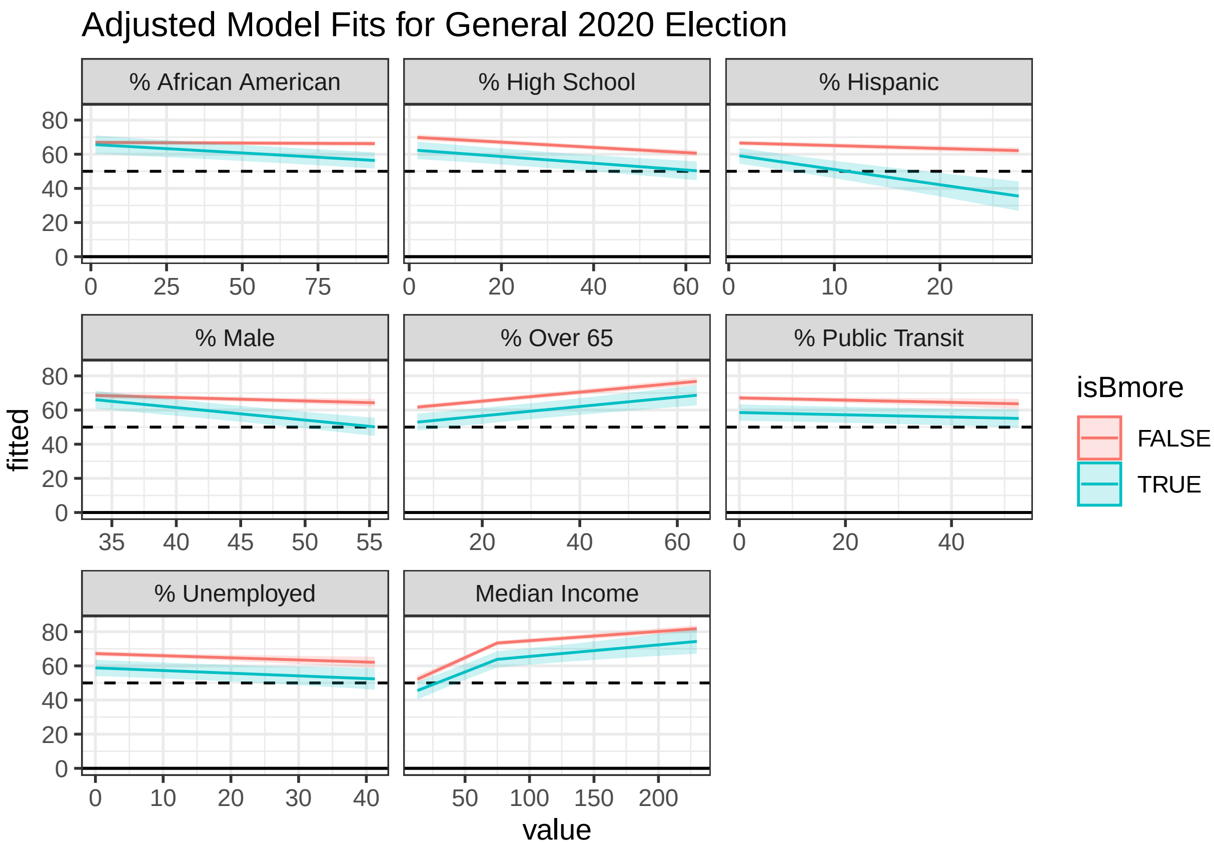
```
get_new_data <- function(controls, variable, elections, knot) {

  if (missing(knot)) {
    knot = 75
  }

  medians <- controls %>%
    select(starts_with("med") & !contains(variable)) %>%
    as.matrix()

  variable_controls <- controls %>%
    select(contains(variable)) %>%
    as.matrix()

  variable_names <- controls %>%
    select(starts_with("med") & !contains(variable)) %>%
    colnames()

  variable_names <- substring(variable_names,5)

  new_data <- expand.grid(
    election = elections,
    isBmore = c(TRUE,FALSE),
    var1 = seq(variable_controls[1,3], variable_controls[1,2], length.out = 50),
    var2 = medians[1],
    var3 = medians[2],
    var4 = medians[3],
    var5 = medians[4],
    var6 = medians[5],
    var7 = medians[6],
    var8 = medians[7]
  ) %>%
    mutate(election = factor(election),
           isBmore = factor(isBmore))

  colnames(new_data)[3:10] <- c(variable, variable_names)

  if (variable == "median_income") {
    new_data$median_income_sp1 <- ifelse(new_data$median_income > knot,
                                         new_data$median_income - knot, 0)
  } else if (medians[1,"med_median_income"] > knot) {
    new_data$median_income_sp1 <- medians[1,"med_median_income"] - knot
  } else {
    new_data$median_income_sp1 <- 0
  }

  return(new_data)

}
```

```r
# general elections

################ median income ################
new_data.medinc <- get_new_data(controls, "median_income", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.medinc, re.form = NA, se.fit=TRUE)

new_data.medinc$fitted <- pred$fit
new_data.medinc$se <- pred$se.fit
new_data.medinc$variable <- "Median Income"
new_data.medinc <- new_data.medinc %>%
  mutate(value = median_income) %>%
  select(variable, isBmore, election, value, fitted, se)

################ over 65 ################
new_data.over65 <- get_new_data(controls, "pct_over_65", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.over65, re.form = NA, se.fit=TRUE)

new_data.over65$fitted <- pred$fit
new_data.over65$se <- pred$se.fit
new_data.over65$variable <- "% Over 65"
new_data.over65 <- new_data.over65 %>%
  mutate(value = pct_over_65) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent male ################
new_data.pctmale <- get_new_data(controls, "pct_male", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.pctmale, re.form = NA, se.fit=TRUE)

new_data.pctmale$fitted <- pred$fit
new_data.pctmale$se <- pred$se.fit
new_data.pctmale$variable <- "% Male"
new_data.pctmale <- new_data.pctmale %>%
  mutate(value = pct_male) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent unemployed ################
new_data.pctunemp <- get_new_data(controls, "pct_unemployed", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.pctunemp, re.form = NA, se.fit=TRUE)

new_data.pctunemp$fitted <- pred$fit
new_data.pctunemp$se <- pred$se.fit
new_data.pctunemp$variable <- "% Unemployed"
new_data.pctunemp <- new_data.pctunemp%>%
  mutate(value = pct_unemployed) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent high school attainment ################
new_data.pcths <- get_new_data(controls, "pct_hs", c("gen_2020", "gen_2022"))
```

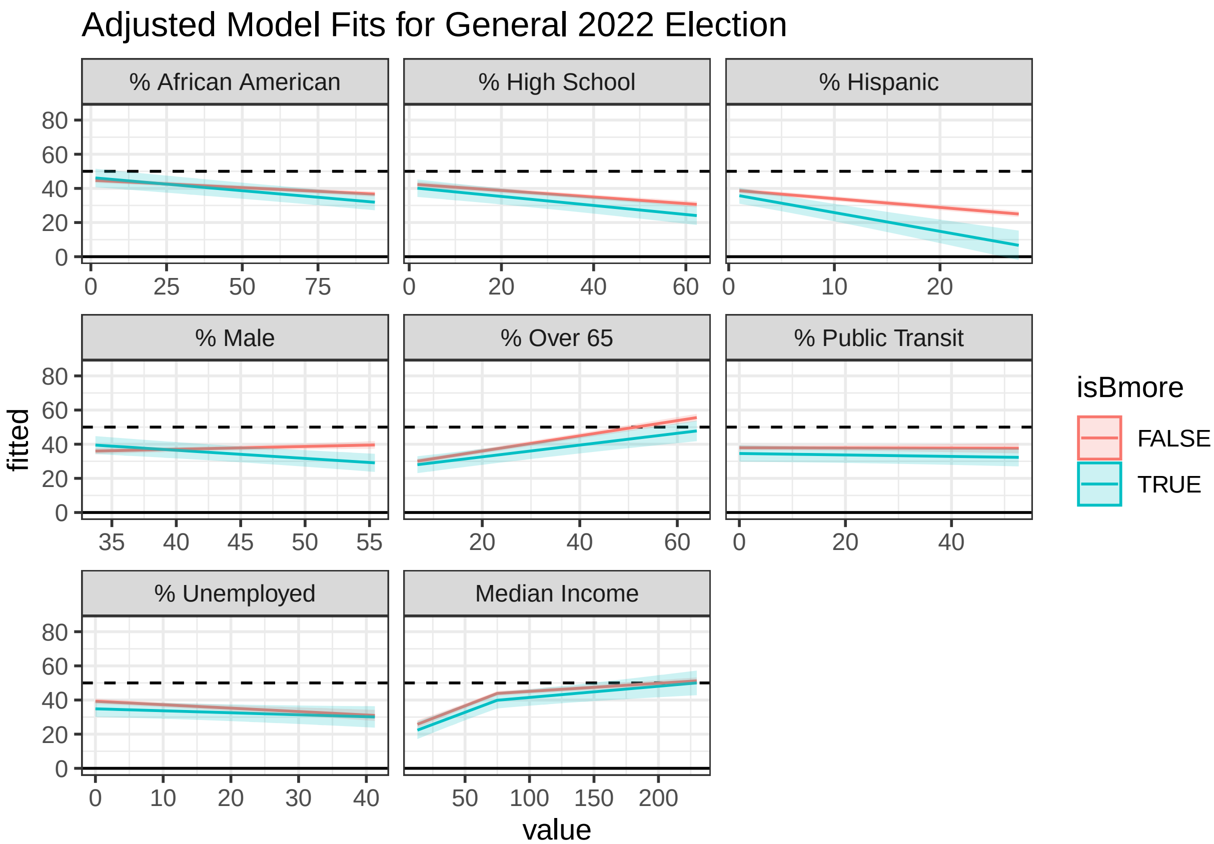
```r
pred <- predict(gen_lme.adj, new_data.pcths, re.form = NA, se.fit=TRUE)

new_data.pcths$fitted <- pred$fit
new_data.pcths$se <- pred$se.fit
new_data.pcths$variable <- "% High School"
new_data.pcths <- new_data.pcths %>%
  mutate(value = pct_hs) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent relying on public transit ################
new_data.pcttransit <- get_new_data(controls, "pct_transit", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.pcttransit, re.form = NA, se.fit=TRUE)

new_data.pcttransit$fitted <- pred$fit
new_data.pcttransit$se <- pred$se.fit
new_data.pcttransit$variable <- "% Public Transit"
new_data.pcttransit <- new_data.pcttransit %>%
  mutate(value = pct_transit) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent African American ################
new_data.pctaa <- get_new_data(controls, "pct_aa", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.pctaa, re.form = NA, se.fit=TRUE)

new_data.pctaa$fitted <- pred$fit
new_data.pctaa$se <- pred$se.fit
new_data.pctaa$variable <- "% African American"
new_data.pctaa <- new_data.pctaa %>%
  mutate(value = pct_aa) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent hispanic ################
new_data.pcthisp <- get_new_data(controls, "pct_hisp", c("gen_2020", "gen_2022"))

pred <- predict(gen_lme.adj, new_data.pcthisp, re.form = NA, se.fit=TRUE)

new_data.pcthisp$fitted <- pred$fit
new_data.pcthisp$se <- pred$se.fit
new_data.pcthisp$variable <- "% Hispanic"
new_data.pcthisp <- new_data.pcthisp %>%
  mutate(value = pct_hisp) %>%
  select(variable, isBmore, election, value, fitted, se)

################ combine data frames ################
new_data <- rbind(new_data.pctaa, new_data.pcthisp, new_data.pcths,
                  new_data.pctmale, new_data.pcttransit, new_data.pctunemp,
                  new_data.medinc, new_data.over65)

# calculate confidence intervals
new_data <- new_data %>%
  mutate(lower = fitted - 1.96*se, upper = fitted + 1.96*se)
```

Plot model fits

```
gen_lme.plot.2020 <- new_data %>%
  filter(election == "gen_2020") %>%
  ggplot(aes(x = value, y = fitted, color = isBmore)) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_line() +
  geom_hline(aes(yintercept = 0)) +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = isBmore, color = NULL), alpha = 0.2) +
  coord_cartesian(ylim = c(0, 85)) +
  facet_wrap(~variable, scale = "free_x") +
  labs(title = "Adjusted Model Fits for General 2020 Election") +
  theme_bw()

gen_lme.plot.2022 <- new_data %>%
  filter(election == "gen_2022") %>%
  ggplot(aes(x = value, y = fitted, color = isBmore)) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = isBmore, color = NULL), alpha = 0.2) +
  coord_cartesian(ylim = c(0, 85)) +
  facet_wrap(~variable, scale = "free_x") +
  labs(title = "Adjusted Model Fits for General 2022 Election") +
  theme_bw()

gen_lme.plot.2020
```




```
gen_lme.plot.2022
```




```r
gen_lme.adj.st.coefs <- summary(gen_lme.adj.st)$coefficients %>%
  as.data.frame() %>%
  select(Estimate)

gen_lme.adj.st.coefs$idx <- 1:nrow(gen_lme.adj.st.coefs)
gen_lme.adj.st.coefs
```

Standardized Model

```
##                                                  Estimate idx
## (Intercept)                                  1.5105111410   1
## isBmoreTRUE                                 -0.5905962451   2
## electiongen_2022                            -1.6432331908   3
## pct_aa                                      -0.0143131088   4
## pct_hisp                                    -0.0687512848   5
## pct_male                                    -0.0332085827   6
## median_income                                0.9509525659   7
## median_income_sp1                           -0.8003717484   8
## pct_over_65                                  0.1225102549   9
## pct_unemployed                              -0.0284729646  10
## pct_hs                                      -0.0960436242  11
## pct_transit                                 -0.0308262355  12
## isBmoreTRUE:electiongen_2022                 0.3324758924  13
## isBmoreTRUE:pct_aa                          -0.1773663493  14
## isBmoreTRUE:pct_hisp                        -0.2957997439  15
## isBmoreTRUE:pct_male                        -0.0891709003  16
```

```
## isBmoreTRUE:median_income                    -0.1304266654  17
## isBmoreTRUE:median_income_sp1                 0.1678057660  18
## isBmoreTRUE:pct_over_65                       0.0053885486  19
## isBmoreTRUE:pct_unemployed                   -0.0075239802  20
## isBmoreTRUE:pct_hs                           -0.0283711066  21
## isBmoreTRUE:pct_transit                       0.0002251084  22
## electiongen_2022:pct_aa                      -0.1505689283  23
## electiongen_2022:pct_hisp                    -0.1418918001  24
## electiongen_2022:pct_male                     0.0607812997  25
## electiongen_2022:median_income               -0.1401999888  26
## electiongen_2022:median_income_sp1            0.1214245337  27
## electiongen_2022:pct_over_65                  0.0849948207  28
## electiongen_2022:pct_unemployed              -0.0176280765  29
## electiongen_2022:pct_hs                      -0.0248110980  30
## electiongen_2022:pct_transit                  0.0282003755  31
## isBmoreTRUE:electiongen_2022:pct_aa           0.0494421079  32
## isBmoreTRUE:electiongen_2022:pct_hisp         0.0578308772  33
## isBmoreTRUE:electiongen_2022:pct_male        -0.0183950879  34
## isBmoreTRUE:electiongen_2022:median_income    0.1042396481  35
## isBmoreTRUE:electiongen_2022:median_income_sp1 -0.0913421513  36
## isBmoreTRUE:electiongen_2022:pct_over_65     -0.0522017362  37
## isBmoreTRUE:electiongen_2022:pct_unemployed   0.0270374624  38
## isBmoreTRUE:electiongen_2022:pct_hs          -0.0174295207  39
## isBmoreTRUE:electiongen_2022:pct_transit     -0.0178651320  40
```

```
################ median income ################

# isbmore and gen_2020
ind <- matrix(0,nrow=nrow(gen_lme.adj.st.coefs),ncol=8)
ind[7,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(7,26),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(7,17),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(7,17,26,35),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(7,8),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(7,8,26,27),6] <- 1 # median_income >= 75 & !bmore & 2022
ind[c(7,8,17,18),7] <- 1 # median_income >= 75 & bmore & 2020
ind[c(7,8,17,18,26,27,35,36),8] <- 1 # median_income >= 75 & bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.medinc <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.medinc) <- c("Est", "Lower", "Upper")
ci.adj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
ci.adj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
ci.adj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[9,1] <- 1 # !bmore & 2020
ind[c(9,28),2] <- 1 # !bmore & 2022
ind[c(9,19),3] <- 1 # bmore & 2020
ind[c(9,19,28,37),4] <- 1 # bmore & 2022
```

```r
# get confidence intervals for desired linear combination of estimates
ci.adj.over65 <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.over65) <- c("Est", "Lower", "Upper")
ci.adj.over65$variable <- c(rep("pct_over_65", 4))
ci.adj.over65$bmore <- c(0,0,1,1)
ci.adj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[6,1] <- 1 # !bmore & 2020
ind[c(6,25),2] <- 1 # !bmore & 2022
ind[c(6,16),3] <- 1 # bmore & 2020
ind[c(6,16,25,34),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctmale <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctmale) <- c("Est", "Lower", "Upper")
ci.adj.pctmale$variable <- c(rep("pct_male", 4))
ci.adj.pctmale$bmore <- c(0,0,1,1)
ci.adj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[10,1] <- 1 # !bmore & 2020
ind[c(10,29),2] <- 1 # !bmore & 2022
ind[c(10,20),3] <- 1 # bmore & 2020
ind[c(10,20,29,38),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctunemp <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctunemp) <- c("Est", "Lower", "Upper")
ci.adj.pctunemp$variable <- c(rep("pct_unemployed", 4))
ci.adj.pctunemp$bmore <- c(0,0,1,1)
ci.adj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[11,1] <- 1 # !bmore & 2020
ind[c(11,30),2] <- 1 # !bmore & 2022
ind[c(11,21),3] <- 1 # bmore & 2020
ind[c(11,21,30,39),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcths <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
```

```r
  as.data.frame()

colnames(ci.adj.pcths) <- c("Est", "Lower", "Upper")
ci.adj.pcths$variable <- c(rep("pct_hs", 4))
ci.adj.pcths$bmore <- c(0,0,1,1)
ci.adj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[12,1] <- 1 # !bmore & 2020
ind[c(12,31),2] <- 1 # !bmore & 2022
ind[c(12,22),3] <- 1 # bmore & 2020
ind[c(12,22,31,40),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcttransit <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcttransit) <- c("Est", "Lower", "Upper")
ci.adj.pcttransit$variable <- c(rep("pct_transit", 4))
ci.adj.pcttransit$bmore <- c(0,0,1,1)
ci.adj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,23),2] <- 1 # !bmore & 2022
ind[c(4,14),3] <- 1 # bmore & 2020
ind[c(4,14,23,32),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctaa <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctaa) <- c("Est", "Lower", "Upper")
ci.adj.pctaa$variable <- c(rep("pct_aa", 4))
ci.adj.pctaa$bmore <- c(0,0,1,1)
ci.adj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(gen_lme.adj.st.coefs), ncol = 4)
ind[5,1] <- 1 # !bmore & 2020
ind[c(5,24),2] <- 1 # !bmore & 2022
ind[c(5,15),3] <- 1 # bmore & 2020
ind[c(5,15,24,33),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcthisp <- confint.lme(gen_lme.adj.st, ind, 0.05) %>%
  as.data.frame()
```

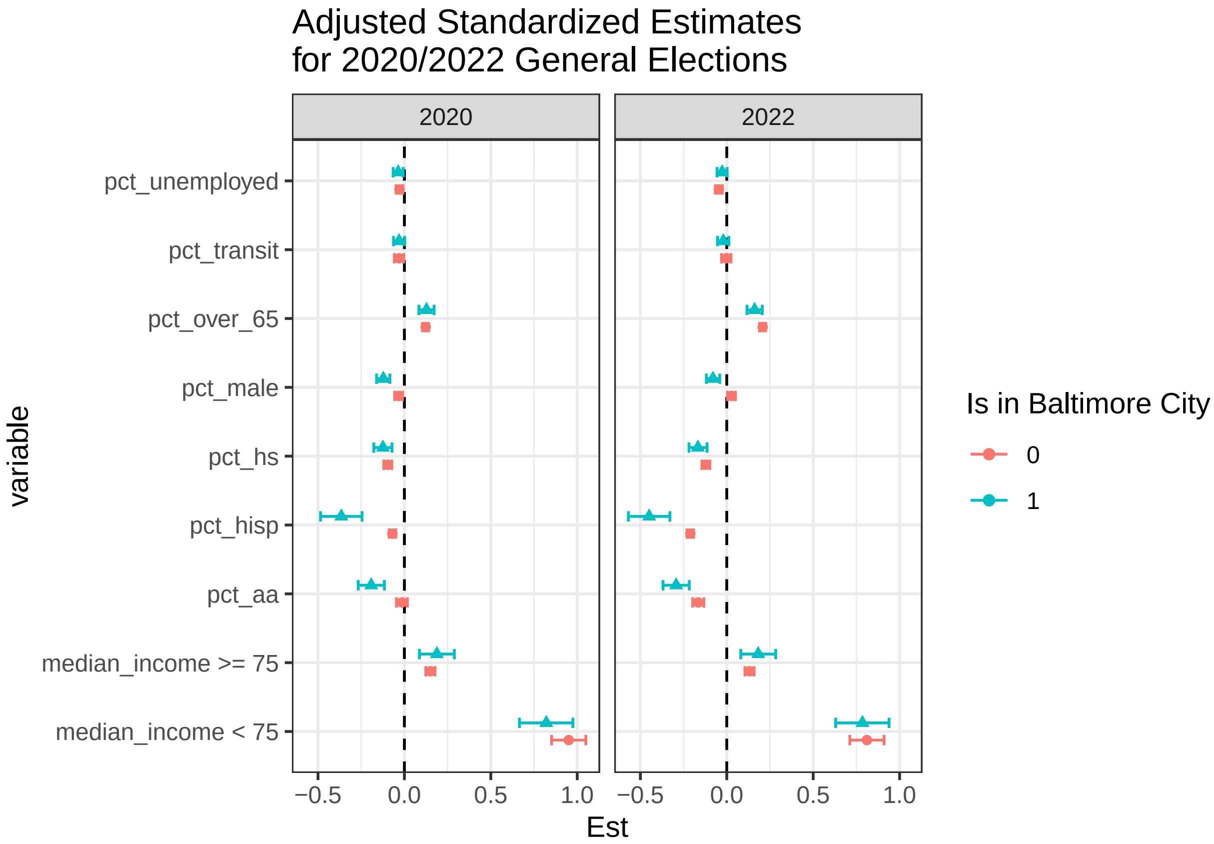
```r
colnames(ci.adj.pcthisp) <- c("Est", "Lower", "Upper")
ci.adj.pcthisp$variable <- c(rep("pct_hisp", 4))
ci.adj.pcthisp$bmore <- c(0,0,1,1)
ci.adj.pcthisp$year <- rep(c(2020,2022), 2)

################ combine tables ################
gen_ci.adj.st <- rbind(ci.adj.medinc, ci.adj.over65, ci.adj.pctaa, ci.adj.pcthisp,
                  ci.adj.pcths, ci.adj.pctmale, ci.adj.pcttransit, ci.adj.pctunemp)

gen_ci.adj.st[,1:3] <- round(gen_ci.adj.st[,1:3],3)
gen_ci.adj.st
```

```
##       Est  Lower  Upper           variable bmore year
## 1   0.951  0.852  1.050  median_income < 75     0 2020
## 2   0.811  0.712  0.910  median_income < 75     0 2022
## 3   0.821  0.666  0.975  median_income < 75     1 2020
## 4   0.785  0.630  0.939  median_income < 75     1 2022
## 5   0.151  0.124  0.177 median_income >= 75     0 2020
## 6   0.132  0.105  0.158 median_income >= 75     0 2022
## 7   0.188  0.087  0.289 median_income >= 75     1 2020
## 8   0.182  0.081  0.283 median_income >= 75     1 2022
## 9   0.123  0.105  0.140         pct_over_65     0 2020
## 10  0.208  0.190  0.225         pct_over_65     0 2022
## 11  0.128  0.084  0.172         pct_over_65     1 2020
## 12  0.161  0.117  0.205         pct_over_65     1 2022
## 13 -0.014 -0.046  0.017              pct_aa     0 2020
## 14 -0.165 -0.197 -0.133              pct_aa     0 2022
## 15 -0.192 -0.267 -0.116              pct_aa     1 2020
## 16 -0.293 -0.369 -0.217              pct_aa     1 2022
## 17 -0.069 -0.088 -0.050            pct_hisp     0 2020
## 18 -0.211 -0.230 -0.192            pct_hisp     0 2022
## 19 -0.365 -0.485 -0.245            pct_hisp     1 2020
## 20 -0.449 -0.569 -0.329            pct_hisp     1 2022
## 21 -0.096 -0.119 -0.073              pct_hs     0 2020
## 22 -0.121 -0.144 -0.098              pct_hs     0 2022
## 23 -0.124 -0.177 -0.072              pct_hs     1 2020
## 24 -0.167 -0.219 -0.114              pct_hs     1 2022
## 25 -0.033 -0.056 -0.011            pct_male     0 2020
## 26  0.028  0.005  0.050            pct_male     0 2022
## 27 -0.122 -0.161 -0.084            pct_male     1 2020
## 28 -0.080 -0.118 -0.041            pct_male     1 2022
## 29 -0.031 -0.058 -0.004         pct_transit     0 2020
## 30 -0.003 -0.030  0.024         pct_transit     0 2022
## 31 -0.031 -0.063  0.002         pct_transit     1 2020
## 32 -0.020 -0.053  0.012         pct_transit     1 2022
## 33 -0.028 -0.048 -0.009      pct_unemployed     0 2020
## 34 -0.046 -0.066 -0.026      pct_unemployed     0 2022
## 35 -0.036 -0.065 -0.007      pct_unemployed     1 2020
## 36 -0.027 -0.056  0.002      pct_unemployed     1 2022
```

Plot standardized estimates

```r
gen_ci.adj.st %>%
  ggplot(aes(x = Est, y = variable)) +
  geom_vline(aes(xintercept = 0), linetype = "dashed") +
```

```
  geom_point(aes(shape = factor(bmore), color = factor(bmore)),position = position_dodge(width = 0.5)) +
  geom_errorbarh(aes(xmin = Lower, xmax = Upper, color = factor(bmore)), height = 0.3, position = positi
  labs(title = "Adjusted Standardized Estimates \nfor 2020/2022 General Elections", color = "Is in Balti
  guides(shape = "none") +
  facet_wrap(~year) +
  theme_bw()
```



Primary elections

Single predictor models (LME)

```
################ median income ################
# fit model
prim_lme.unadj.medinc <- lmer(turnout ~ election*isBmore*(median_income + median_income_sp1) + (1|county

# get model summary
prim_lme.unadj.medinc.summary <- summary(prim_lme.unadj.medinc)
ncoef <- nrow(prim_lme.unadj.medinc.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*8), nrow = ncoef, ncol = 8)
ind[4,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(4,7),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(4,7,9),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(4,7,9,11),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(4,5),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(4,5,7,8),6] <- 1 # median_income >= 75 & !bmore & 2022
```

```
ind[c(4,5,9:10),7] <- 1 # median_income >= 75 & bmore & 2020
ind[c(4,5,7:12),8] <- 1 # median_income >= 75 & bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.medinc <- confint.lme(prim_lme.unadj.medinc, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.medinc) <- c("Est", "Lower", "Upper")
prim_ci.unadj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
prim_ci.unadj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
prim_ci.unadj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################
# fit model
prim_lme.unadj.over65 <- lmer(turnout ~ election*isBmore*(pct_over_65) + (1|county_name/tract_id), dat =

# get model summary
prim_lme.unadj.over65.summary <- summary(prim_lme.unadj.over65)
ncoef <- nrow(prim_lme.unadj.over65.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.over65 <- confint.lme(prim_lme.unadj.over65, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.over65) <- c("Est", "Lower", "Upper")
prim_ci.unadj.over65$variable <- c(rep("pct_over_65", 4))
prim_ci.unadj.over65$bmore <- c(0,0,1,1)
prim_ci.unadj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################
# fit model
prim_lme.unadj.pctmale <- lmer(turnout ~ election*isBmore*(pct_male) + (1|county_name/tract_id), dat =

# get model summary
prim_lme.unadj.pctmale.summary <- summary(prim_lme.unadj.pctmale)
ncoef <- nrow(prim_lme.unadj.pctmale.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pctmale <- confint.lme(prim_lme.unadj.pctmale, ind, 0.05) %>%
```

```
  as.data.frame()

colnames(prim_ci.unadj.pctmale) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pctmale$variable <- c(rep("pct_male", 4))
prim_ci.unadj.pctmale$bmore <- c(0,0,1,1)
prim_ci.unadj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################
# fit model
prim_lme.unadj.pctunemp <- lmer(turnout ~ election*isBmore*(pct_unemployed) + (1|county_name/tract_id),

# get model summary
prim_lme.unadj.pctunemp.summary <- summary(prim_lme.unadj.pctunemp)
ncoef <- nrow(prim_lme.unadj.pctunemp.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pctunemp <- confint.lme(prim_lme.unadj.pctunemp, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.pctunemp) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pctunemp$variable <- c(rep("pct_unemployed", 4))
prim_ci.unadj.pctunemp$bmore <- c(0,0,1,1)
prim_ci.unadj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################
# fit model
prim_lme.unadj.pcths <- lmer(turnout ~ election*isBmore*(pct_hs) + (1|county_name/tract_id), dat = dat.

# get model summary
prim_lme.unadj.pcths.summary <- summary(prim_lme.unadj.pcths)
ncoef <- nrow(prim_lme.unadj.pcths.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pcths <- confint.lme(prim_lme.unadj.pcths, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.pcths) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pcths$variable <- c(rep("pct_hs", 4))
prim_ci.unadj.pcths$bmore <- c(0,0,1,1)
```

```
prim_ci.unadj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################
# fit model
prim_lme.unadj.pcttransit <- lmer(turnout ~ election*isBmore*(pct_transit) + (1|county_name/tract_id), d

# get model summary
prim_lme.unadj.pcttransit.summary <- summary(prim_lme.unadj.pcttransit)
ncoef <- nrow(prim_lme.unadj.pcttransit.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pcttransit <- confint.lme(prim_lme.unadj.pcttransit, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.pcttransit) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pcttransit$variable <- c(rep("pct_transit", 4))
prim_ci.unadj.pcttransit$bmore <- c(0,0,1,1)
prim_ci.unadj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################
# fit model
prim_lme.unadj.pctaa <- lmer(turnout ~ election*isBmore*(pct_aa) + (1|county_name/tract_id), dat = dat.p

# get model summary
prim_lme.unadj.pctaa.summary <- summary(prim_lme.unadj.pctaa)
ncoef <- nrow(prim_lme.unadj.pctaa.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pctaa<- confint.lme(prim_lme.unadj.pctaa, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.pctaa) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pctaa$variable <- c(rep("pct_aa", 4))
prim_ci.unadj.pctaa$bmore <- c(0,0,1,1)
prim_ci.unadj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################
# fit model
prim_lme.unadj.pcthisp <- lmer(turnout ~ election*isBmore*(pct_hisp) + (1|county_name/tract_id), dat = d
```

```r
# get model summary
prim_lme.unadj.pcthisp.summary <- summary(prim_lme.unadj.pcthisp)
ncoef <- nrow(prim_lme.unadj.pcthisp.summary$coefficients)

# preallocate indicator matrix for adding estimates
ind <- matrix(rep(0,ncoef*4), nrow = ncoef, ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,6),2] <- 1 # !bmore & 2022
ind[c(4,7),3] <- 1 # bmore & 2020
ind[c(4,6,7,8),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
prim_ci.unadj.pcthisp <- confint.lme(prim_lme.unadj.pcthisp, ind, 0.05) %>%
  as.data.frame()

colnames(prim_ci.unadj.pcthisp) <- c("Est", "Lower", "Upper")
prim_ci.unadj.pcthisp$variable <- c(rep("pct_hisp", 4))
prim_ci.unadj.pcthisp$bmore <- c(0,0,1,1)
prim_ci.unadj.pcthisp$year <- rep(c(2020,2022), 2)
```

I've created a table displaying the unadjusted parameter estimates below:

```r
prim_ci.unadj <- rbind(prim_ci.unadj.medinc, prim_ci.unadj.over65, prim_ci.unadj.pctaa, prim_ci.unadj.pc
                   prim_ci.unadj.pcths, prim_ci.unadj.pctmale, prim_ci.unadj.pcttransit, prim_ci.unadj.pc

prim_ci.unadj[,1:3] <- round(prim_ci.unadj[,1:3],3)
prim_ci.unadj
```

```
##       Est  Lower  Upper            variable bmore year
## 1   0.191  0.143  0.238  median_income < 75     0 2020
## 2   0.180  0.133  0.228  median_income < 75     0 2022
## 3   0.310  0.244  0.376  median_income < 75     1 2020
## 4   0.303  0.248  0.358  median_income < 75     1 2022
## 5   0.066  0.057  0.076 median_income >= 75     0 2020
## 6   0.067  0.058  0.077 median_income >= 75     0 2022
## 7   0.020 -0.021  0.062 median_income >= 75     1 2020
## 8   0.079  0.037  0.120 median_income >= 75     1 2022
## 9   0.392  0.346  0.438         pct_over_65     0 2020
## 10  0.538  0.492  0.584         pct_over_65     0 2022
## 11  0.552  0.434  0.671         pct_over_65     1 2020
## 12  0.394  0.276  0.513         pct_over_65     1 2022
## 13  0.054  0.035  0.072              pct_aa     0 2020
## 14 -0.073 -0.091 -0.055              pct_aa     0 2022
## 15 -0.067 -0.096 -0.039              pct_aa     1 2020
## 16 -0.137 -0.165 -0.108              pct_aa     1 2022
## 17 -0.364 -0.423 -0.305            pct_hisp     0 2020
## 18 -0.567 -0.626 -0.507            pct_hisp     0 2022
## 19 -0.809 -1.137 -0.480            pct_hisp     1 2020
## 20 -0.247 -0.575  0.082            pct_hisp     1 2022
## 21 -0.357 -0.399 -0.315              pct_hs     0 2020
## 22 -0.248 -0.290 -0.206              pct_hs     0 2022
## 23 -0.309 -0.383 -0.235              pct_hs     1 2020
## 24 -0.419 -0.493 -0.345              pct_hs     1 2022
## 25 -0.823 -0.978 -0.668            pct_male     0 2020
```

```
## 26  0.167  0.012  0.322           pct_male     0 2022
## 27 -0.602 -0.885 -0.320           pct_male     1 2020
## 28 -0.030 -0.313  0.252           pct_male     1 2022
## 29  0.064 -0.017  0.144        pct_transit     0 2020
## 30 -0.326 -0.407 -0.246        pct_transit     0 2022
## 31 -0.324 -0.404 -0.244        pct_transit     1 2020
## 32 -0.356 -0.436 -0.275        pct_transit     1 2022
## 33 -0.225 -0.350 -0.100     pct_unemployed     0 2020
## 34 -0.604 -0.729 -0.479     pct_unemployed     0 2022
## 35 -0.524 -0.681 -0.366     pct_unemployed     1 2020
## 36 -0.573 -0.730 -0.415     pct_unemployed     1 2022
```

Adjusted Model (LME)

```
# fit adjusted model for primary elections
prim_lme.adj <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + median_i

prim_lme.adj.summary <- summary(prim_lme.adj)
prim_lme.adj.summary
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: turnout ~ isBmore * election * (pct_aa + pct_hisp + pct_male +
##     median_income + median_income_sp1 + pct_over_65 + pct_unemployed +
##     pct_hs + pct_transit) + (1 | county_name/tract_id)
##    Data: dat.prim
##
##      AIC      BIC   logLik deviance df.resid
##  15293.0  15549.9  -7603.5  15207.0     2865
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7914 -0.4791  0.0004  0.4805  3.5620
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  tract_id:county_name (Intercept) 11.005   3.317
##  county_name          (Intercept)  5.976   2.445
##  Residual                          4.334   2.082
## Number of obs: 2908, groups:  tract_id:county_name, 1454; county_name, 24
##
## Fixed effects:
##                                                    Estimate Std. Error
## (Intercept)                                       2.194e+01  3.271e+00
## isBmoreTRUE                                       2.801e+01  7.331e+00
## electionprim_2022                                -2.710e+01  2.283e+00
## pct_aa                                            1.469e-01  7.658e-03
## pct_hisp                                         -1.284e-01  2.141e-02
## pct_male                                         -1.721e-01  6.236e-02
## median_income                                     1.519e-01  1.647e-02
## median_income_sp1                                -9.967e-02  1.731e-02
## pct_over_65                                       4.480e-01  1.717e-02
## pct_unemployed                                   -1.116e-01  3.999e-02
## pct_hs                                           -2.210e-01  1.661e-02
```

```
## pct_transit                                          5.962e-02  2.646e-02
## isBmoreTRUE:electionprim_2022                        8.999e+00  5.115e+00
## isBmoreTRUE:pct_aa                                  -1.449e-01  1.986e-02
## isBmoreTRUE:pct_hisp                                -8.353e-01  1.370e-01
## isBmoreTRUE:pct_male                                -4.245e-01  1.232e-01
## isBmoreTRUE:median_income                            9.388e-02  3.047e-02
## isBmoreTRUE:median_income_sp1                       -1.030e-01  3.597e-02
## isBmoreTRUE:pct_over_65                              8.652e-02  4.679e-02
## isBmoreTRUE:pct_unemployed                          -1.239e-02  7.036e-02
## isBmoreTRUE:pct_hs                                   2.321e-02  4.201e-02
## isBmoreTRUE:pct_transit                             -1.195e-01  4.129e-02
## electionprim_2022:pct_aa                            -1.120e-01  4.337e-03
## electionprim_2022:pct_hisp                          -1.469e-01  1.376e-02
## electionprim_2022:pct_male                           3.486e-01  4.483e-02
## electionprim_2022:median_income                      1.070e-02  1.156e-02
## electionprim_2022:median_income_sp1                 -2.959e-02  1.237e-02
## electionprim_2022:pct_over_65                        3.246e-02  1.183e-02
## electionprim_2022:pct_unemployed                    -3.491e-02  2.966e-02
## electionprim_2022:pct_hs                             1.010e-01  1.179e-02
## electionprim_2022:pct_transit                       -1.979e-02  1.777e-02
## isBmoreTRUE:electionprim_2022:pct_aa                 1.712e-02  1.444e-02
## isBmoreTRUE:electionprim_2022:pct_hisp               1.743e-01  1.027e-01
## isBmoreTRUE:electionprim_2022:pct_male              -1.344e-01  9.158e-02
## isBmoreTRUE:electionprim_2022:median_income         -6.782e-02  2.247e-02
## isBmoreTRUE:electionprim_2022:median_income_sp1  8.433e-02  2.674e-02
## isBmoreTRUE:electionprim_2022:pct_over_65           -7.641e-02  3.480e-02
## isBmoreTRUE:electionprim_2022:pct_unemployed         9.536e-02  5.267e-02
## isBmoreTRUE:electionprim_2022:pct_hs                -6.861e-02  3.131e-02
## isBmoreTRUE:electionprim_2022:pct_transit            5.886e-02  2.972e-02
##                                                            df t value Pr(>|t|)
## (Intercept)                                          1.799e+03   6.707 2.65e-11 ***
## isBmoreTRUE                                          8.615e+02   3.820 0.000143 ***
## electionprim_2022                                    1.454e+03 -11.870  < 2e-16 ***
## pct_aa                                               1.687e+03  19.186  < 2e-16 ***
## pct_hisp                                             1.770e+03  -5.996 2.44e-09 ***
## pct_male                                             1.864e+03  -2.759 0.005848 **
## median_income                                        1.846e+03   9.225  < 2e-16 ***
## median_income_sp1                                    1.871e+03  -5.760 9.84e-09 ***
## pct_over_65                                          1.835e+03  26.099  < 2e-16 ***
## pct_unemployed                                       1.884e+03  -2.790 0.005325 **
## pct_hs                                               1.861e+03 -13.305  < 2e-16 ***
## pct_transit                                          1.817e+03   2.253 0.024382 *
## isBmoreTRUE:electionprim_2022                        1.454e+03   1.759 0.078727 .
## isBmoreTRUE:pct_aa                                   1.870e+03  -7.296 4.35e-13 ***
## isBmoreTRUE:pct_hisp                                 1.886e+03  -6.096 1.32e-09 ***
## isBmoreTRUE:pct_male                                 1.882e+03  -3.446 0.000582 ***
## isBmoreTRUE:median_income                            1.890e+03   3.081 0.002090 **
## isBmoreTRUE:median_income_sp1                        1.889e+03  -2.862 0.004252 **
## isBmoreTRUE:pct_over_65                              1.887e+03   1.849 0.064590 .
## isBmoreTRUE:pct_unemployed                           1.886e+03  -0.176 0.860197
## isBmoreTRUE:pct_hs                                   1.884e+03   0.552 0.580733
## isBmoreTRUE:pct_transit                              1.865e+03  -2.895 0.003831 **
## electionprim_2022:pct_aa                             1.454e+03 -25.824  < 2e-16 ***
## electionprim_2022:pct_hisp                           1.454e+03 -10.669  < 2e-16 ***
```

```
## electionprim_2022:pct_male                         1.454e+03   7.775 1.42e-14 ***
## electionprim_2022:median_income                    1.454e+03   0.926 0.354700
## electionprim_2022:median_income_sp1                1.454e+03  -2.393 0.016844 *
## electionprim_2022:pct_over_65                      1.454e+03   2.743 0.006157 **
## electionprim_2022:pct_unemployed                   1.454e+03  -1.177 0.239479
## electionprim_2022:pct_hs                           1.454e+03   8.565  < 2e-16 ***
## electionprim_2022:pct_transit                      1.454e+03  -1.114 0.265611
## isBmoreTRUE:electionprim_2022:pct_aa               1.454e+03   1.185 0.236210
## isBmoreTRUE:electionprim_2022:pct_hisp             1.454e+03   1.697 0.089842 .
## isBmoreTRUE:electionprim_2022:pct_male             1.454e+03  -1.467 0.142527
## isBmoreTRUE:electionprim_2022:median_income        1.454e+03  -3.018 0.002585 **
## isBmoreTRUE:electionprim_2022:median_income_sp1    1.454e+03   3.154 0.001643 **
## isBmoreTRUE:electionprim_2022:pct_over_65          1.454e+03  -2.196 0.028264 *
## isBmoreTRUE:electionprim_2022:pct_unemployed       1.454e+03   1.810 0.070425 .
## isBmoreTRUE:electionprim_2022:pct_hs               1.454e+03  -2.191 0.028579 *
## isBmoreTRUE:electionprim_2022:pct_transit          1.454e+03   1.980 0.047855 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 40 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```r
# fit standardized adjusted model for primary elections
prim_lme.adj.st <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + medi

prim_lme.adj.st.summary <- summary(prim_lme.adj.st)
prim_lme.adj.st.summary
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: turnout ~ isBmore * election * (pct_aa + pct_hisp + pct_male +
##     median_income + median_income_sp1 + pct_over_65 + pct_unemployed +
##     pct_hs + pct_transit) + (1 | county_name/tract_id)
##    Data: dat.prim.st
##
##      AIC      BIC   logLik deviance df.resid
##   2070.7   2327.6   -992.4   1984.7     2865
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7914 -0.4791  0.0004  0.4805  3.5620
##
## Random effects:
##  Groups               Name        Variance Std.Dev.
##  tract_id:county_name (Intercept) 0.11665  0.3415
##  county_name          (Intercept) 0.06334  0.2517
##  Residual                         0.04595  0.2143
## Number of obs: 2908, groups:  tract_id:county_name, 1454; county_name, 24
##
## Fixed effects:
##                                                     Estimate Std. Error
## (Intercept)                                        1.051e+00  8.470e-02
## isBmoreTRUE                                        1.042e+00  2.956e-01
```

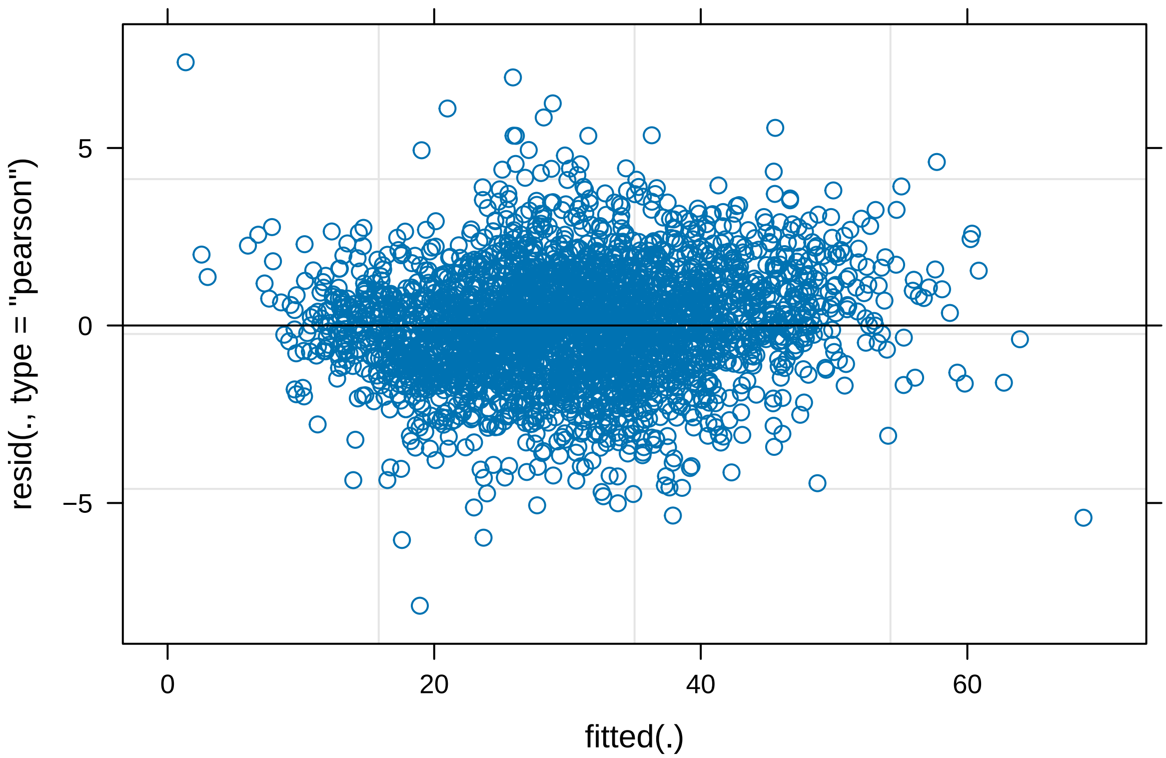
```
## electionprim_2022                               -1.177e+00  4.380e-02
## pct_aa                                           4.829e-01  2.517e-02
## pct_hisp                                        -9.042e-02  1.508e-02
## pct_male                                        -4.954e-02  1.795e-02
## median_income                                    7.273e-01  7.884e-02
## median_income_sp1                               -4.772e-01  8.285e-02
## pct_over_65                                      3.602e-01  1.380e-02
## pct_unemployed                                  -4.427e-02  1.587e-02
## pct_hs                                          -2.399e-01  1.803e-02
## pct_transit                                      4.829e-02  2.143e-02
## isBmoreTRUE:electionprim_2022                   -5.504e-01  1.069e-01
## isBmoreTRUE:pct_aa                              -4.764e-01  6.529e-02
## isBmoreTRUE:pct_hisp                            -5.883e-01  9.652e-02
## isBmoreTRUE:pct_male                            -1.222e-01  3.547e-02
## isBmoreTRUE:median_income                        4.494e-01  1.459e-01
## isBmoreTRUE:median_income_sp1                   -4.929e-01  1.722e-01
## isBmoreTRUE:pct_over_65                          6.957e-02  3.762e-02
## isBmoreTRUE:pct_unemployed                      -4.919e-03  2.792e-02
## isBmoreTRUE:pct_hs                               2.519e-02  4.560e-02
## isBmoreTRUE:pct_transit                         -9.682e-02  3.344e-02
## electionprim_2022:pct_aa                        -3.681e-01  1.425e-02
## electionprim_2022:pct_hisp                      -1.034e-01  9.695e-03
## electionprim_2022:pct_male                       1.004e-01  1.291e-02
## electionprim_2022:median_income                  5.123e-02  5.533e-02
## electionprim_2022:median_income_sp1             -1.417e-01  5.920e-02
## electionprim_2022:pct_over_65                    2.610e-02  9.515e-03
## electionprim_2022:pct_unemployed                -1.385e-02  1.177e-02
## electionprim_2022:pct_hs                         1.096e-01  1.279e-02
## electionprim_2022:pct_transit                   -1.603e-02  1.439e-02
## isBmoreTRUE:electionprim_2022:pct_aa             5.626e-02  4.748e-02
## isBmoreTRUE:electionprim_2022:pct_hisp           1.228e-01  7.232e-02
## isBmoreTRUE:electionprim_2022:pct_male          -3.869e-02  2.637e-02
## isBmoreTRUE:electionprim_2022:median_income     -3.247e-01  1.076e-01
## isBmoreTRUE:electionprim_2022:median_income_sp1  4.037e-01  1.280e-01
## isBmoreTRUE:electionprim_2022:pct_over_65       -6.143e-02  2.798e-02
## isBmoreTRUE:electionprim_2022:pct_unemployed     3.784e-02  2.090e-02
## isBmoreTRUE:electionprim_2022:pct_hs            -7.447e-02  3.398e-02
## isBmoreTRUE:electionprim_2022:pct_transit        4.767e-02  2.407e-02
##                                                         df t value Pr(>|t|)
## (Intercept)                                      1.055e+02  12.403  < 2e-16 ***
## isBmoreTRUE                                      3.088e+01   3.524 0.001348 **
## electionprim_2022                                1.454e+03 -26.881  < 2e-16 ***
## pct_aa                                           1.687e+03  19.186  < 2e-16 ***
## pct_hisp                                         1.770e+03  -5.996 2.44e-09 ***
## pct_male                                         1.864e+03  -2.759 0.005848 **
## median_income                                    1.846e+03   9.225  < 2e-16 ***
## median_income_sp1                                1.871e+03  -5.760 9.84e-09 ***
## pct_over_65                                      1.835e+03  26.099  < 2e-16 ***
## pct_unemployed                                   1.884e+03  -2.790 0.005325 **
## pct_hs                                           1.861e+03 -13.305  < 2e-16 ***
## pct_transit                                      1.817e+03   2.253 0.024382 *
## isBmoreTRUE:electionprim_2022                    1.454e+03  -5.146 3.02e-07 ***
## isBmoreTRUE:pct_aa                               1.870e+03  -7.296 4.35e-13 ***
## isBmoreTRUE:pct_hisp                             1.886e+03  -6.096 1.32e-09 ***
```

```
## isBmoreTRUE:pct_male                              1.882e+03  -3.446 0.000582 ***
## isBmoreTRUE:median_income                         1.890e+03   3.081 0.002090 **
## isBmoreTRUE:median_income_sp1                     1.889e+03  -2.862 0.004252 **
## isBmoreTRUE:pct_over_65                           1.887e+03   1.849 0.064590 .
## isBmoreTRUE:pct_unemployed                        1.886e+03  -0.176 0.860197
## isBmoreTRUE:pct_hs                                1.884e+03   0.552 0.580733
## isBmoreTRUE:pct_transit                           1.865e+03  -2.895 0.003831 **
## electionprim_2022:pct_aa                          1.454e+03 -25.824  < 2e-16 ***
## electionprim_2022:pct_hisp                        1.454e+03 -10.669  < 2e-16 ***
## electionprim_2022:pct_male                        1.454e+03   7.775 1.42e-14 ***
## electionprim_2022:median_income                   1.454e+03   0.926 0.354700
## electionprim_2022:median_income_sp1               1.454e+03  -2.393 0.016844 *
## electionprim_2022:pct_over_65                     1.454e+03   2.743 0.006157 **
## electionprim_2022:pct_unemployed                  1.454e+03  -1.177 0.239479
## electionprim_2022:pct_hs                          1.454e+03   8.565  < 2e-16 ***
## electionprim_2022:pct_transit                     1.454e+03  -1.114 0.265611
## isBmoreTRUE:electionprim_2022:pct_aa              1.454e+03   1.185 0.236210
## isBmoreTRUE:electionprim_2022:pct_hisp            1.454e+03   1.697 0.089842 .
## isBmoreTRUE:electionprim_2022:pct_male            1.454e+03  -1.467 0.142527
## isBmoreTRUE:electionprim_2022:median_income       1.454e+03  -3.018 0.002585 **
## isBmoreTRUE:electionprim_2022:median_income_sp1   1.454e+03   3.154 0.001643 **
## isBmoreTRUE:electionprim_2022:pct_over_65         1.454e+03  -2.196 0.028264 *
## isBmoreTRUE:electionprim_2022:pct_unemployed      1.454e+03   1.810 0.070425 .
## isBmoreTRUE:electionprim_2022:pct_hs              1.454e+03  -2.191 0.028579 *
## isBmoreTRUE:electionprim_2022:pct_transit         1.454e+03   1.980 0.047855 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 40 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

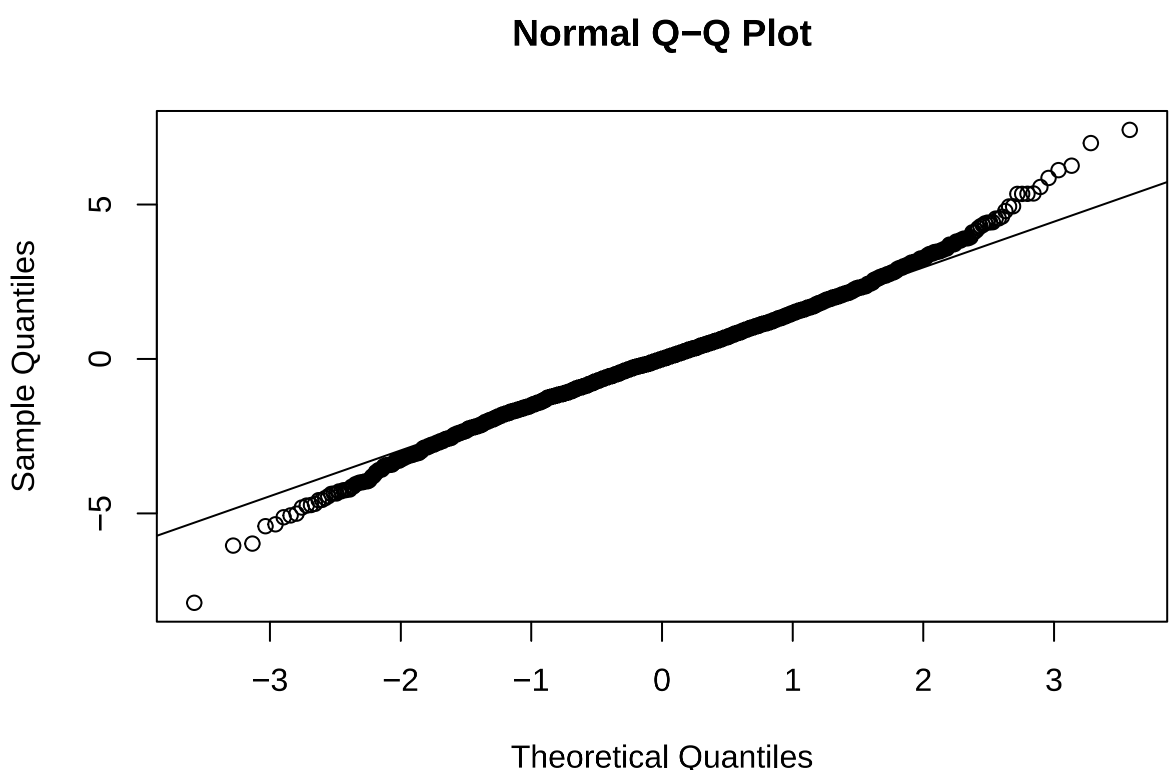
Diagnostics I want to evaluate the fit of the (standardized) LME model and assess its sensitivity to outliers and influential points.

Step 1) Plot residuals against fitted values from standardized adjusted model.

```r
plot(prim_lme.adj)
```




```r
qqnorm(resid(prim_lme.adj))
qqline(resid(prim_lme.adj))
```



Step 2) Identify census tracts with the largest residuals

```r
dat.prim.st$res <- prim_lme.adj.st.summary$residuals
dat.prim.st$fitted <- predict(prim_lme.adj.st, dat.prim.st, re.form = NULL)

dat.prim.st %>%
  select(tract_id, county_name, fitted, res) %>%
  arrange(desc(abs(res))) %>%
  head(15)
```

```
## # A tibble: 15 x 4
##    tract_id    county_name            fitted   res
##    <chr>       <chr>                   <dbl> <dbl>
##  1 24033805601 Prince George's County -1.19  -3.79
##  2 24033805601 Prince George's County -3.00   3.56
##  3 24005490605 Baltimore County       -0.475  3.36
##  4 24037875400 St. Mary's County      -0.168  3.01
##  5 24033804600 Prince George's County -0.980  2.94
##  6 24033801105 Prince George's County -1.33  -2.90
##  7 24031701221 Montgomery County      -0.702 -2.87
##  8 24037875502 St. Mary's County      -0.237  2.81
##  9 24027605505 Howard County           1.55   2.67
## 10 24510271503 Baltimore city          3.93  -2.60
## 11 24023000602 Garrett County          0.597  2.57
## 12 24037875400 St. Mary's County       0.760 -2.57
## 13 24043000800 Washington County      -0.471  2.57
## 14 24025302100 Harford County          0.107  2.57
## 15 24037875700 St. Mary's County      -0.452  2.57
```

Step 2) Use Cook's distance to identify influential data points.

```r
dat.prim.st$cooks_distance <- cooks.distance(prim_lme.adj.st)

# identify most influential points
dat.prim.st %>%
  arrange(desc(cooks_distance)) %>%
  select(tract_id, county_name, election, cooks_distance, fitted, res) %>%
  head()
```

```
## # A tibble: 6 x 6
##   tract_id    county_name            election  cooks_distance fitted   res
##   <chr>       <chr>                  <chr>              <dbl>  <dbl> <dbl>
## 1 24510271503 Baltimore city         prim_2020          0.558  3.93  -2.60
## 2 24033805601 Prince George's County prim_2020          0.537 -1.19  -3.79
## 3 24033805601 Prince George's County prim_2022          0.474 -3.00   3.56
## 4 24005490605 Baltimore County       prim_2020          0.430 -0.475  3.36
## 5 24037875400 St. Mary's County      prim_2022          0.287 -0.168  3.01
## 6 24033804600 Prince George's County prim_2022          0.275 -0.980  2.94
```

Step 3) Compare model excluding the most influential census tracts to the original model

I will now refit the standardized lme model excluding the 2 census tracts with cooks distance > 0.5, and compare it to the original model fit.

```r
tracts_to_exclude <- c(24510271503, 24033805601)
dat.prim.st.ex <- dat.prim.st %>%
  filter(!tract_id %in% tracts_to_exclude)

# refit model
```

```r
prim_lme.adj.st.ex <- lmer(turnout ~ isBmore*election*(pct_aa + pct_hisp + pct_male + median_income + me

prim_lme.adj.st.ex.summary <- summary(prim_lme.adj.st.ex)

# compare new coefficient estimates with original ones
prim_lme.adj.st.t <- round(cbind(prim_lme.adj.st.summary$coefficients[,c(1,5)],
                          prim_lme.adj.st.ex.summary$coefficients[,c(1,5)]),3)

prim_lme.adj.st.t
```

```
##                                                Estimate Pr(>|t|) Estimate
## (Intercept)                                       1.051    0.000    1.046
## isBmoreTRUE                                       1.042    0.001    1.080
## electionprim_2022                                -1.177    0.000   -1.168
## pct_aa                                            0.483    0.000    0.480
## pct_hisp                                         -0.090    0.000   -0.084
## pct_male                                         -0.050    0.006   -0.051
## median_income                                     0.727    0.000    0.721
## median_income_sp1                                -0.477    0.000   -0.469
## pct_over_65                                       0.360    0.000    0.360
## pct_unemployed                                   -0.044    0.005   -0.046
## pct_hs                                           -0.240    0.000   -0.239
## pct_transit                                       0.048    0.024    0.049
## isBmoreTRUE:electionprim_2022                    -0.550    0.000   -0.580
## isBmoreTRUE:pct_aa                               -0.476    0.000   -0.528
## isBmoreTRUE:pct_hisp                             -0.588    0.000   -0.617
## isBmoreTRUE:pct_male                             -0.122    0.001   -0.148
## isBmoreTRUE:median_income                         0.449    0.002    0.413
## isBmoreTRUE:median_income_sp1                    -0.493    0.004   -0.496
## isBmoreTRUE:pct_over_65                           0.070    0.065    0.121
## isBmoreTRUE:pct_unemployed                       -0.005    0.860   -0.010
## isBmoreTRUE:pct_hs                                0.025    0.581    0.012
## isBmoreTRUE:pct_transit                          -0.097    0.004   -0.098
## electionprim_2022:pct_aa                         -0.368    0.000   -0.365
## electionprim_2022:pct_hisp                       -0.103    0.000   -0.114
## electionprim_2022:pct_male                        0.100    0.000    0.103
## electionprim_2022:median_income                   0.051    0.355    0.063
## electionprim_2022:median_income_sp1              -0.142    0.017   -0.154
## electionprim_2022:pct_over_65                     0.026    0.006    0.027
## electionprim_2022:pct_unemployed                 -0.014    0.239   -0.012
## electionprim_2022:pct_hs                          0.110    0.000    0.108
## electionprim_2022:pct_transit                    -0.016    0.266   -0.016
## isBmoreTRUE:electionprim_2022:pct_aa              0.056    0.236    0.087
## isBmoreTRUE:electionprim_2022:pct_hisp            0.123    0.090    0.147
## isBmoreTRUE:electionprim_2022:pct_male           -0.039    0.143   -0.024
## isBmoreTRUE:electionprim_2022:median_income      -0.325    0.003   -0.310
## isBmoreTRUE:electionprim_2022:median_income_sp1   0.404    0.002    0.414
## isBmoreTRUE:electionprim_2022:pct_over_65        -0.061    0.028   -0.094
## isBmoreTRUE:electionprim_2022:pct_unemployed      0.038    0.070    0.039
## isBmoreTRUE:electionprim_2022:pct_hs             -0.074    0.029   -0.065
## isBmoreTRUE:electionprim_2022:pct_transit         0.048    0.048    0.048
##                                                Pr(>|t|)
## (Intercept)                                       0.000
## isBmoreTRUE                                       0.001
```

```
## electionprim_2022                                 0.000
## pct_aa                                            0.000
## pct_hisp                                          0.000
## pct_male                                          0.004
## median_income                                     0.000
## median_income_sp1                                 0.000
## pct_over_65                                       0.000
## pct_unemployed                                    0.004
## pct_hs                                            0.000
## pct_transit                                       0.023
## isBmoreTRUE:electionprim_2022                     0.000
## isBmoreTRUE:pct_aa                                0.000
## isBmoreTRUE:pct_hisp                              0.000
## isBmoreTRUE:pct_male                              0.000
## isBmoreTRUE:median_income                         0.005
## isBmoreTRUE:median_income_sp1                     0.004
## isBmoreTRUE:pct_over_65                           0.002
## isBmoreTRUE:pct_unemployed                        0.732
## isBmoreTRUE:pct_hs                                0.792
## isBmoreTRUE:pct_transit                           0.003
## electionprim_2022:pct_aa                          0.000
## electionprim_2022:pct_hisp                        0.000
## electionprim_2022:pct_male                        0.000
## electionprim_2022:median_income                   0.248
## electionprim_2022:median_income_sp1               0.008
## electionprim_2022:pct_over_65                     0.004
## electionprim_2022:pct_unemployed                  0.316
## electionprim_2022:pct_hs                          0.000
## electionprim_2022:pct_transit                     0.271
## isBmoreTRUE:electionprim_2022:pct_aa              0.071
## isBmoreTRUE:electionprim_2022:pct_hisp            0.040
## isBmoreTRUE:electionprim_2022:pct_male            0.370
## isBmoreTRUE:electionprim_2022:median_income       0.004
## isBmoreTRUE:electionprim_2022:median_income_sp1   0.001
## isBmoreTRUE:electionprim_2022:pct_over_65         0.001
## isBmoreTRUE:electionprim_2022:pct_unemployed      0.057
## isBmoreTRUE:electionprim_2022:pct_hs              0.052
## isBmoreTRUE:electionprim_2022:pct_transit         0.044
```

We can see that the estimates and their significance changed very little after excluding these census tracts. This suggests model inferences are not overly sensitive to the presence of these outliers.

Model Summary

To summarize the adjusted model, I will extract the coefficient estimates and their confidence intervals separately for election year and `isBmore` status.

```r
prim_lme.adj.coefs <- summary(prim_lme.adj)$coefficients %>%
  as.data.frame() %>%
  select(Estimate)

prim_lme.adj.coefs$idx <- 1:nrow(prim_lme.adj.coefs)
prim_lme.adj.coefs
```

Unstandardized adjusted model

```
##                                                    Estimate idx
## (Intercept)                                      21.93647116   1
## isBmoreTRUE                                      28.00631312   2
## electionprim_2022                               -27.09574636   3
## pct_aa                                            0.14692237   4
## pct_hisp                                         -0.12837543   5
## pct_male                                         -0.17207463   6
## median_income                                     0.15192340   7
## median_income_sp1                                -0.09967158   8
## pct_over_65                                       0.44801302   9
## pct_unemployed                                   -0.11155883  10
## pct_hs                                           -0.22103664  11
## pct_transit                                       0.05962066  12
## isBmoreTRUE:electionprim_2022                     8.99857464  13
## isBmoreTRUE:pct_aa                               -0.14493245  14
## isBmoreTRUE:pct_hisp                             -0.83530089  15
## isBmoreTRUE:pct_male                             -0.42445618  16
## isBmoreTRUE:median_income                         0.09387543  17
## isBmoreTRUE:median_income_sp1                    -0.10295594  18
## isBmoreTRUE:pct_over_65                           0.08652217  19
## isBmoreTRUE:pct_unemployed                       -0.01239370  20
## isBmoreTRUE:pct_hs                                0.02320698  21
## isBmoreTRUE:pct_transit                          -0.11954501  22
## electionprim_2022:pct_aa                         -0.11199250  23
## electionprim_2022:pct_hisp                       -0.14684959  24
## electionprim_2022:pct_male                        0.34858220  25
## electionprim_2022:median_income                   0.01070065  26
## electionprim_2022:median_income_sp1              -0.02959115  27
## electionprim_2022:pct_over_65                     0.03246397  28
## electionprim_2022:pct_unemployed                 -0.03490732  29
## electionprim_2022:pct_hs                          0.10095142  30
## electionprim_2022:pct_transit                    -0.01978836  31
## isBmoreTRUE:electionprim_2022:pct_aa              0.01711723  32
## isBmoreTRUE:electionprim_2022:pct_hisp            0.17427641  33
## isBmoreTRUE:electionprim_2022:pct_male           -0.13437656  34
## isBmoreTRUE:electionprim_2022:median_income      -0.06781967  35
## isBmoreTRUE:electionprim_2022:median_income_sp1   0.08433131  36
## isBmoreTRUE:electionprim_2022:pct_over_65        -0.07640489  37
## isBmoreTRUE:electionprim_2022:pct_unemployed      0.09535711  38
## isBmoreTRUE:electionprim_2022:pct_hs             -0.06861438  39
## isBmoreTRUE:electionprim_2022:pct_transit         0.05885773  40
```

```
################ median income ################

# isbmore and gen_2020
ind <- matrix(0,nrow=nrow(prim_lme.adj.coefs),ncol=8)
ind[7,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(7,26),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(7,17),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(7,17,26,35),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(7,8),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(7,8,26,27),6] <- 1 # median_income >= 75 & !bmore & 2022
ind[c(7,8,17,18),7] <- 1 # median_income >= 75 & bmore & 2020
```

```r
ind[c(7,8,17,18,26,27,35,36),8] <- 1 # median_income >= 75 & bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.medinc <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.medinc) <- c("Est", "Lower", "Upper")
ci.adj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
ci.adj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
ci.adj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[9,1] <- 1 # !bmore & 2020
ind[c(9,28),2] <- 1 # !bmore & 2022
ind[c(9,19),3] <- 1 # bmore & 2020
ind[c(9,19,28,37),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.over65 <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.over65) <- c("Est", "Lower", "Upper")
ci.adj.over65$variable <- c(rep("pct_over_65", 4))
ci.adj.over65$bmore <- c(0,0,1,1)
ci.adj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[6,1] <- 1 # !bmore & 2020
ind[c(6,25),2] <- 1 # !bmore & 2022
ind[c(6,16),3] <- 1 # bmore & 2020
ind[c(6,16,25,34),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctmale <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctmale) <- c("Est", "Lower", "Upper")
ci.adj.pctmale$variable <- c(rep("pct_male", 4))
ci.adj.pctmale$bmore <- c(0,0,1,1)
ci.adj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[10,1] <- 1 # !bmore & 2020
ind[c(10,29),2] <- 1 # !bmore & 2022
ind[c(10,20),3] <- 1 # bmore & 2020
ind[c(10,20,29,38),4] <- 1 # bmore & 2022
```

```r
# get confidence intervals for desired linear combination of estimates
ci.adj.pctunemp <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctunemp) <- c("Est", "Lower", "Upper")
ci.adj.pctunemp$variable <- c(rep("pct_unemployed", 4))
ci.adj.pctunemp$bmore <- c(0,0,1,1)
ci.adj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[11,1] <- 1 # !bmore & 2020
ind[c(11,30),2] <- 1 # !bmore & 2022
ind[c(11,21),3] <- 1 # bmore & 2020
ind[c(11,21,30,39),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcths <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcths) <- c("Est", "Lower", "Upper")
ci.adj.pcths$variable <- c(rep("pct_hs", 4))
ci.adj.pcths$bmore <- c(0,0,1,1)
ci.adj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[12,1] <- 1 # !bmore & 2020
ind[c(12,31),2] <- 1 # !bmore & 2022
ind[c(12,22),3] <- 1 # bmore & 2020
ind[c(12,22,31,40),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcttransit <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcttransit) <- c("Est", "Lower", "Upper")
ci.adj.pcttransit$variable <- c(rep("pct_transit", 4))
ci.adj.pcttransit$bmore <- c(0,0,1,1)
ci.adj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,23),2] <- 1 # !bmore & 2022
ind[c(4,14),3] <- 1 # bmore & 2020
ind[c(4,14,23,32),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctaa <- confint.lme(prim_lme.adj, ind, 0.05) %>%
```

```r
  as.data.frame()

colnames(ci.adj.pctaa) <- c("Est", "Lower", "Upper")
ci.adj.pctaa$variable <- c(rep("pct_aa", 4))
ci.adj.pctaa$bmore <- c(0,0,1,1)
ci.adj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.coefs), ncol = 4)
ind[5,1] <- 1 # !bmore & 2020
ind[c(5,24),2] <- 1 # !bmore & 2022
ind[c(5,15),3] <- 1 # bmore & 2020
ind[c(5,15,24,33),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcthisp <- confint.lme(prim_lme.adj, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcthisp) <- c("Est", "Lower", "Upper")
ci.adj.pcthisp$variable <- c(rep("pct_hisp", 4))
ci.adj.pcthisp$bmore <- c(0,0,1,1)
ci.adj.pcthisp$year <- rep(c(2020,2022), 2)

################ combine tables ################
prim_ci.adj <- rbind(ci.adj.medinc, ci.adj.over65, ci.adj.pctaa, ci.adj.pcthisp,
                     ci.adj.pcths, ci.adj.pctmale, ci.adj.pcttransit, ci.adj.pctunemp)

prim_ci.adj[,1:3] <- round(prim_ci.adj[,1:3],3)
prim_ci.adj
```

```
##       Est  Lower  Upper            variable bmore year
## 1   0.152  0.120  0.184  median_income < 75     0 2020
## 2   0.163  0.130  0.195  median_income < 75     0 2022
## 3   0.246  0.196  0.296  median_income < 75     1 2020
## 4   0.189  0.138  0.239  median_income < 75     1 2022
## 5   0.052  0.044  0.061 median_income >= 75     0 2020
## 6   0.033  0.025  0.042 median_income >= 75     0 2022
## 7   0.043  0.010  0.076 median_income >= 75     1 2020
## 8   0.041  0.008  0.074 median_income >= 75     1 2022
## 9   0.448  0.414  0.482         pct_over_65     0 2020
## 10  0.480  0.447  0.514         pct_over_65     0 2022
## 11  0.535  0.449  0.620         pct_over_65     1 2020
## 12  0.491  0.405  0.576         pct_over_65     1 2022
## 13  0.147  0.132  0.162              pct_aa     0 2020
## 14  0.035  0.020  0.050              pct_aa     0 2022
## 15  0.002 -0.034  0.038              pct_aa     1 2020
## 16 -0.093 -0.129 -0.057              pct_aa     1 2022
## 17 -0.128 -0.170 -0.086            pct_hisp     0 2020
## 18 -0.275 -0.317 -0.233            pct_hisp     0 2022
## 19 -0.964 -1.229 -0.698            pct_hisp     1 2020
## 20 -0.936 -1.202 -0.671            pct_hisp     1 2022
## 21 -0.221 -0.254 -0.188              pct_hs     0 2020
## 22 -0.120 -0.153 -0.088              pct_hs     0 2022
```

```
## 23 -0.198 -0.273 -0.122             pct_hs     1 2020
## 24 -0.165 -0.241 -0.090             pct_hs     1 2022
## 25 -0.172 -0.294 -0.050           pct_male     0 2020
## 26  0.177  0.054  0.299           pct_male     0 2022
## 27 -0.597 -0.805 -0.388           pct_male     1 2020
## 28 -0.382 -0.591 -0.174           pct_male     1 2022
## 29  0.060  0.008  0.111        pct_transit     0 2020
## 30  0.040 -0.012  0.092        pct_transit     0 2022
## 31 -0.060 -0.122  0.002        pct_transit     1 2020
## 32 -0.021 -0.083  0.041        pct_transit     1 2022
## 33 -0.112 -0.190 -0.033     pct_unemployed     0 2020
## 34 -0.146 -0.225 -0.068     pct_unemployed     0 2022
## 35 -0.124 -0.237 -0.010     pct_unemployed     1 2020
## 36 -0.064 -0.177  0.050     pct_unemployed     1 2022
```

```
# define controls for plotting model fits
bmore_dat <- dat.prim %>%
  filter(isBmore == 1, election == "prim_2020")

controls <- bmore_dat %>%
  summarize(med_median_income = median(median_income),
            max_median_income = max(median_income),
            min_median_income = min(median_income),
            med_pct_aa = median(pct_aa),
            max_pct_aa = max(pct_aa),
            min_pct_aa = min(pct_aa),
            med_pct_hisp = median(pct_hisp),
            max_pct_hisp = max(pct_hisp),
            min_pct_hisp = min(pct_hisp),
            med_pct_over_65 = median(pct_over_65),
            max_pct_over_65 = max(pct_over_65),
            min_pct_over_65 = min(pct_over_65),
            med_pct_male = median(pct_male),
            max_pct_male = max(pct_male),
            min_pct_male = min(pct_male),
            med_pct_unemployed = median(pct_unemployed),
            max_pct_unemployed = max(pct_unemployed),
            min_pct_unemployed = min(pct_unemployed),
            med_pct_hs = median(pct_hs),
            max_pct_hs = max(pct_hs),
            min_pct_hs = min(pct_hs),
            med_pct_transit = median(pct_transit),
            max_pct_transit = max(pct_transit),
            min_pct_transit = min(pct_transit))

controls
```

```
## # A tibble: 1 x 24
##   med_median_income max_median_income min_median_income med_pct_aa max_pct_aa
##               <dbl>             <dbl>             <dbl>      <dbl>      <dbl>
## 1              54.4              230.              13.0       79.9       93.8
## # i 19 more variables: min_pct_aa <dbl>, med_pct_hisp <dbl>,
## #   max_pct_hisp <dbl>, min_pct_hisp <dbl>, med_pct_over_65 <dbl>,
## #   max_pct_over_65 <dbl>, min_pct_over_65 <dbl>, med_pct_male <dbl>,
## #   max_pct_male <dbl>, min_pct_male <dbl>, med_pct_unemployed <dbl>,
```

```
## #   max_pct_unemployed <dbl>, min_pct_unemployed <dbl>, med_pct_hs <dbl>,
## #   max_pct_hs <dbl>, min_pct_hs <dbl>, med_pct_transit <dbl>,
## #   max_pct_transit <dbl>, min_pct_transit <dbl>
```

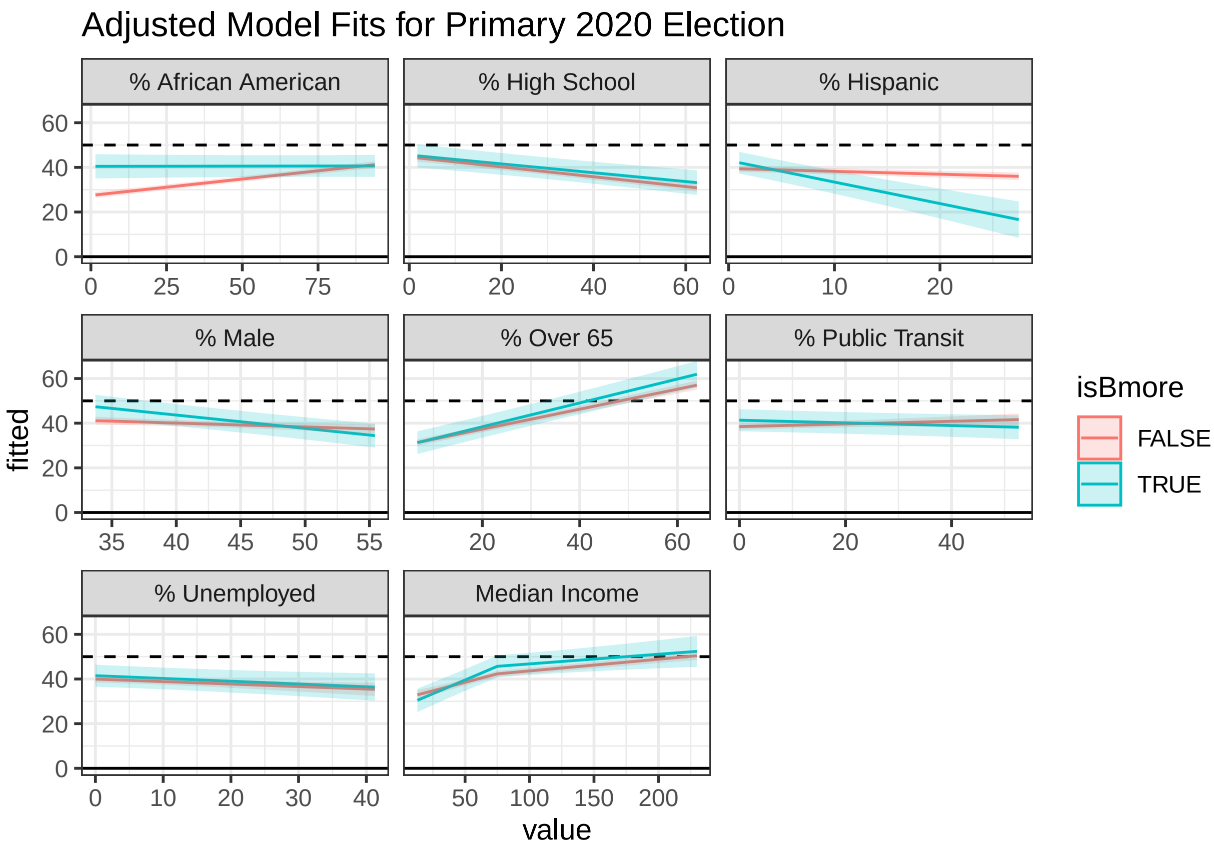
```r
# primary elections

################ median income ################
new_data.medinc <- get_new_data(controls, "median_income", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.medinc, re.form = NA, se.fit=TRUE)

new_data.medinc$fitted <- pred$fit
new_data.medinc$se <- pred$se.fit
new_data.medinc$variable <- "Median Income"
new_data.medinc <- new_data.medinc %>%
  mutate(value = median_income) %>%
  select(variable, isBmore, election, value, fitted, se)

################ over 65 ################
new_data.over65 <- get_new_data(controls, "pct_over_65", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.over65, re.form = NA, se.fit=TRUE)

new_data.over65$fitted <- pred$fit
new_data.over65$se <- pred$se.fit
new_data.over65$variable <- "% Over 65"
new_data.over65 <- new_data.over65 %>%
  mutate(value = pct_over_65) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent male ################
new_data.pctmale <- get_new_data(controls, "pct_male", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pctmale, re.form = NA, se.fit=TRUE)

new_data.pctmale$fitted <- pred$fit
new_data.pctmale$se <- pred$se.fit
new_data.pctmale$variable <- "% Male"
new_data.pctmale <- new_data.pctmale %>%
  mutate(value = pct_male) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent unemployed ################
new_data.pctunemp <- get_new_data(controls, "pct_unemployed", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pctunemp, re.form = NA, se.fit=TRUE)

new_data.pctunemp$fitted <- pred$fit
new_data.pctunemp$se <- pred$se.fit
new_data.pctunemp$variable <- "% Unemployed"
new_data.pctunemp <- new_data.pctunemp%>%
  mutate(value = pct_unemployed) %>%
  select(variable, isBmore, election, value, fitted, se)
```

```r
################ percent high school attainment ################
new_data.pcths <- get_new_data(controls, "pct_hs", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pcths, re.form = NA, se.fit=TRUE)

new_data.pcths$fitted <- pred$fit
new_data.pcths$se <- pred$se.fit
new_data.pcths$variable <- "% High School"
new_data.pcths <- new_data.pcths %>%
  mutate(value = pct_hs) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent relying on public transit ################
new_data.pcttransit <- get_new_data(controls, "pct_transit", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pcttransit, re.form = NA, se.fit=TRUE)

new_data.pcttransit$fitted <- pred$fit
new_data.pcttransit$se <- pred$se.fit
new_data.pcttransit$variable <- "% Public Transit"
new_data.pcttransit <- new_data.pcttransit %>%
  mutate(value = pct_transit) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent African American ################
new_data.pctaa <- get_new_data(controls, "pct_aa", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pctaa, re.form = NA, se.fit=TRUE)

new_data.pctaa$fitted <- pred$fit
new_data.pctaa$se <- pred$se.fit
new_data.pctaa$variable <- "% African American"
new_data.pctaa <- new_data.pctaa %>%
  mutate(value = pct_aa) %>%
  select(variable, isBmore, election, value, fitted, se)

################ percent hispanic ################
new_data.pcthisp <- get_new_data(controls, "pct_hisp", c("prim_2020", "prim_2022"))

pred <- predict(prim_lme.adj, new_data.pcthisp, re.form = NA, se.fit=TRUE)

new_data.pcthisp$fitted <- pred$fit
new_data.pcthisp$se <- pred$se.fit
new_data.pcthisp$variable <- "% Hispanic"
new_data.pcthisp <- new_data.pcthisp %>%
  mutate(value = pct_hisp) %>%
  select(variable, isBmore, election, value, fitted, se)

################ combine data frames ################
new_data <- rbind(new_data.pctaa, new_data.pcthisp, new_data.pcths,
                  new_data.pctmale, new_data.pcttransit, new_data.pctunemp,
                  new_data.medinc, new_data.over65)
```

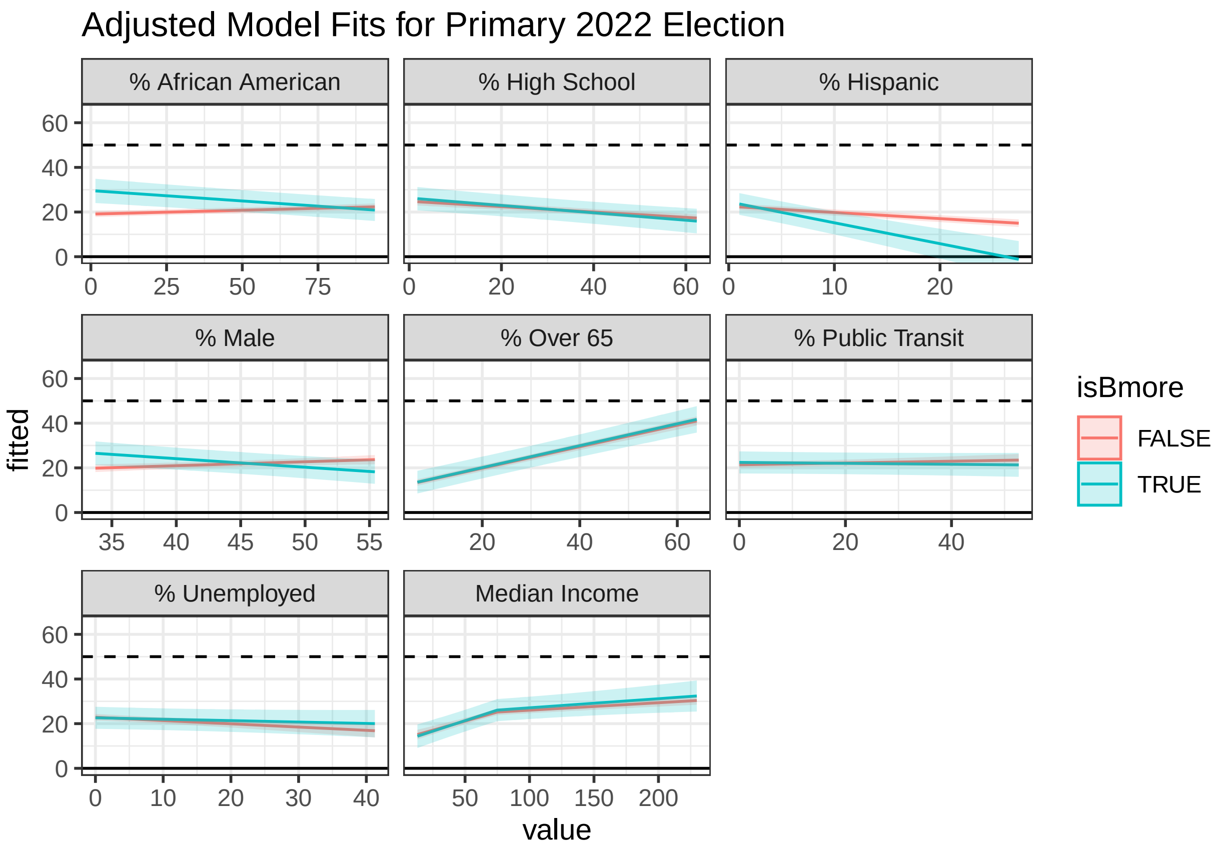
```r
# calculate confidence intervals
new_data <- new_data %>%
  mutate(lower = fitted - 1.96*se, upper = fitted + 1.96*se)
```

Plot model fits

```r
prim_lme.plot.2020 <- new_data %>%
  filter(election == "prim_2020") %>%
  ggplot(aes(x = value, y = fitted, color = isBmore)) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = isBmore, color = NULL), alpha = 0.2) +
  coord_cartesian(ylim = c(0, 65)) +
  facet_wrap(~variable, scale = "free_x") +
  labs(title = "Adjusted Model Fits for Primary 2020 Election") +
  theme_bw()

prim_lme.plot.2022 <- new_data %>%
  filter(election == "prim_2022") %>%
  ggplot(aes(x = value, y = fitted, color = isBmore)) +
  geom_hline(aes(yintercept = 0)) +
  geom_hline(aes(yintercept = 50), linetype = "dashed") +
  geom_line() +
  geom_ribbon(aes(ymin = lower, ymax = upper, fill = isBmore, color = NULL), alpha = 0.2) +
  coord_cartesian(ylim = c(0, 65)) +
  facet_wrap(~variable, scale = "free_x") +
  labs(title = "Adjusted Model Fits for Primary 2022 Election") +
  theme_bw()

prim_lme.plot.2020
```




```
prim_lme.plot.2022
```




```r
prim_lme.adj.st.coefs <- summary(prim_lme.adj.st)$coefficients %>%
  as.data.frame() %>%
  select(Estimate)

prim_lme.adj.st.coefs$idx <- 1:nrow(prim_lme.adj.st.coefs)
prim_lme.adj.st.coefs
```

Standardized Adjusted Model

```
##                                              Estimate idx
## (Intercept)                               1.050567362   1
## isBmoreTRUE                               1.041617595   2
## electionprim_2022                        -1.177473181   3
## pct_aa                                    0.482895343   4
## pct_hisp                                 -0.090421284   5
## pct_male                                 -0.049543644   6
## median_income                             0.727311026   7
## median_income_sp1                        -0.477163090   8
## pct_over_65                               0.360215397   9
## pct_unemployed                           -0.044272780  10
## pct_hs                                   -0.239902063  11
## pct_transit                               0.048287574  12
## isBmoreTRUE:electionprim_2022            -0.550362308  13
## isBmoreTRUE:pct_aa                       -0.476354978  14
## isBmoreTRUE:pct_hisp                     -0.588344515  15
## isBmoreTRUE:pct_male                     -0.122209216  16
```

```
## isBmoreTRUE:median_income                          0.449414874  17
## isBmoreTRUE:median_income_sp1                     -0.492886454  18
## isBmoreTRUE:pct_over_65                            0.069566323  19
## isBmoreTRUE:pct_unemployed                        -0.004918514  20
## isBmoreTRUE:pct_hs                                 0.025187690  21
## isBmoreTRUE:pct_transit                           -0.096821117  22
## electionprim_2022:pct_aa                          -0.368090008  23
## electionprim_2022:pct_hisp                        -0.103433567  24
## electionprim_2022:pct_male                         0.100363619  25
## electionprim_2022:median_income                    0.051227775  26
## electionprim_2022:median_income_sp1               -0.141663304  27
## electionprim_2022:pct_over_65                      0.026101970  28
## electionprim_2022:pct_unemployed                  -0.013853174  29
## electionprim_2022:pct_hs                           0.109567596  30
## electionprim_2022:pct_transit                     -0.016026859  31
## isBmoreTRUE:electionprim_2022:pct_aa               0.056259853  32
## isBmoreTRUE:electionprim_2022:pct_hisp             0.122751662  33
## isBmoreTRUE:electionprim_2022:pct_male            -0.038689635  34
## isBmoreTRUE:electionprim_2022:median_income       -0.324676712  35
## isBmoreTRUE:electionprim_2022:median_income_sp1  0.403723795  36
## isBmoreTRUE:electionprim_2022:pct_over_65         -0.061431739  37
## isBmoreTRUE:electionprim_2022:pct_unemployed       0.037843031  38
## isBmoreTRUE:electionprim_2022:pct_hs              -0.074470596  39
## isBmoreTRUE:electionprim_2022:pct_transit          0.047669666  40
```

```r
################ median income ################

# isbmore and gen_2020
ind <- matrix(0,nrow=nrow(prim_lme.adj.st.coefs),ncol=8)
ind[7,1] <- 1 # median_income < 75 & !bmore & 2020
ind[c(7,26),2] <- 1 # median_income < 75 & !bmore & 2022
ind[c(7,17),3] <- 1 # median_income < 75 & bmore & 2020
ind[c(7,17,26,35),4] <- 1 # median_income < 75 & bmore & 2022
ind[c(7,8),5] <- 1 # median_income >= 75 & !bmore & 2020
ind[c(7,8,26,27),6] <- 1 # median_income >= 75 & !bmore & 2022
ind[c(7,8,17,18),7] <- 1 # median_income >= 75 & bmore & 2020
ind[c(7,8,17,18,26,27,35,36),8] <- 1 # median_income >= 75 & bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.medinc <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.medinc) <- c("Est", "Lower", "Upper")
ci.adj.medinc$variable <- c(rep("median_income < 75", 4), rep("median_income >= 75", 4))
ci.adj.medinc$bmore <- c(0,0,1,1,0,0,1,1)
ci.adj.medinc$year <- rep(c(2020,2022), 4)

################ percent over 65 ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[9,1] <- 1 # !bmore & 2020
ind[c(9,28),2] <- 1 # !bmore & 2022
ind[c(9,19),3] <- 1 # bmore & 2020
ind[c(9,19,28,37),4] <- 1 # bmore & 2022
```

```r
# get confidence intervals for desired linear combination of estimates
ci.adj.over65 <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.over65) <- c("Est", "Lower", "Upper")
ci.adj.over65$variable <- c(rep("pct_over_65", 4))
ci.adj.over65$bmore <- c(0,0,1,1)
ci.adj.over65$year <- rep(c(2020,2022), 2)

################ percent male ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[6,1] <- 1 # !bmore & 2020
ind[c(6,25),2] <- 1 # !bmore & 2022
ind[c(6,16),3] <- 1 # bmore & 2020
ind[c(6,16,25,34),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctmale <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctmale) <- c("Est", "Lower", "Upper")
ci.adj.pctmale$variable <- c(rep("pct_male", 4))
ci.adj.pctmale$bmore <- c(0,0,1,1)
ci.adj.pctmale$year <- rep(c(2020,2022), 2)

################ percent unemployed ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[10,1] <- 1 # !bmore & 2020
ind[c(10,29),2] <- 1 # !bmore & 2022
ind[c(10,20),3] <- 1 # bmore & 2020
ind[c(10,20,29,38),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctunemp <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctunemp) <- c("Est", "Lower", "Upper")
ci.adj.pctunemp$variable <- c(rep("pct_unemployed", 4))
ci.adj.pctunemp$bmore <- c(0,0,1,1)
ci.adj.pctunemp$year <- rep(c(2020,2022), 2)

################ percent high school attainment ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[11,1] <- 1 # !bmore & 2020
ind[c(11,30),2] <- 1 # !bmore & 2022
ind[c(11,21),3] <- 1 # bmore & 2020
ind[c(11,21,30,39),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcths <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
```

```r
  as.data.frame()

colnames(ci.adj.pcths) <- c("Est", "Lower", "Upper")
ci.adj.pcths$variable <- c(rep("pct_hs", 4))
ci.adj.pcths$bmore <- c(0,0,1,1)
ci.adj.pcths$year <- rep(c(2020,2022), 2)

################ percent reliance on public transit ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[12,1] <- 1 # !bmore & 2020
ind[c(12,31),2] <- 1 # !bmore & 2022
ind[c(12,22),3] <- 1 # bmore & 2020
ind[c(12,22,31,40),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcttransit <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pcttransit) <- c("Est", "Lower", "Upper")
ci.adj.pcttransit$variable <- c(rep("pct_transit", 4))
ci.adj.pcttransit$bmore <- c(0,0,1,1)
ci.adj.pcttransit$year <- rep(c(2020,2022), 2)

################ percent African American ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[4,1] <- 1 # !bmore & 2020
ind[c(4,23),2] <- 1 # !bmore & 2022
ind[c(4,14),3] <- 1 # bmore & 2020
ind[c(4,14,23,32),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pctaa <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()

colnames(ci.adj.pctaa) <- c("Est", "Lower", "Upper")
ci.adj.pctaa$variable <- c(rep("pct_aa", 4))
ci.adj.pctaa$bmore <- c(0,0,1,1)
ci.adj.pctaa$year <- rep(c(2020,2022), 2)

################ percent hispanic ################
# preallocate indicator matrix for adding estimates
ind <- matrix(0, nrow = nrow(prim_lme.adj.st.coefs), ncol = 4)
ind[5,1] <- 1 # !bmore & 2020
ind[c(5,24),2] <- 1 # !bmore & 2022
ind[c(5,15),3] <- 1 # bmore & 2020
ind[c(5,15,24,33),4] <- 1 # bmore & 2022

# get confidence intervals for desired linear combination of estimates
ci.adj.pcthisp <- confint.lme(prim_lme.adj.st, ind, 0.05) %>%
  as.data.frame()
```

```r
colnames(ci.adj.pcthisp) <- c("Est", "Lower", "Upper")
ci.adj.pcthisp$variable <- c(rep("pct_hisp", 4))
ci.adj.pcthisp$bmore <- c(0,0,1,1)
ci.adj.pcthisp$year <- rep(c(2020,2022), 2)

################ combine tables ################
prim_ci.adj.st <- rbind(ci.adj.medinc, ci.adj.over65, ci.adj.pctaa, ci.adj.pcthisp,
                  ci.adj.pcths, ci.adj.pctmale, ci.adj.pcttransit, ci.adj.pctunemp)

prim_ci.adj.st[,1:3] <- round(prim_ci.adj.st[,1:3],3)
prim_ci.adj.st
```

```
##       Est  Lower  Upper           variable bmore year
## 1   0.727  0.573  0.882  median_income < 75     0 2020
## 2   0.779  0.624  0.933  median_income < 75     0 2022
## 3   1.177  0.936  1.417  median_income < 75     1 2020
## 4   0.903  0.663  1.144  median_income < 75     1 2022
## 5   0.250  0.209  0.292 median_income >= 75     0 2020
## 6   0.160  0.118  0.201 median_income >= 75     0 2022
## 7   0.207  0.050  0.363 median_income >= 75     1 2020
## 8   0.195  0.039  0.352 median_income >= 75     1 2022
## 9   0.360  0.333  0.387         pct_over_65     0 2020
## 10  0.386  0.359  0.413         pct_over_65     0 2022
## 11  0.430  0.361  0.498         pct_over_65     1 2020
## 12  0.394  0.326  0.463         pct_over_65     1 2022
## 13  0.483  0.434  0.532              pct_aa     0 2020
## 14  0.115  0.065  0.164              pct_aa     0 2022
## 15  0.007 -0.112  0.125              pct_aa     1 2020
## 16 -0.305 -0.423 -0.187              pct_aa     1 2022
## 17 -0.090 -0.120 -0.061            pct_hisp     0 2020
## 18 -0.194 -0.223 -0.164            pct_hisp     0 2022
## 19 -0.679 -0.866 -0.492            pct_hisp     1 2020
## 20 -0.659 -0.846 -0.473            pct_hisp     1 2022
## 21 -0.240 -0.275 -0.205              pct_hs     0 2020
## 22 -0.130 -0.166 -0.095              pct_hs     0 2022
## 23 -0.215 -0.297 -0.133              pct_hs     1 2020
## 24 -0.180 -0.262 -0.098              pct_hs     1 2022
## 25 -0.050 -0.085 -0.014            pct_male     0 2020
## 26  0.051  0.016  0.086            pct_male     0 2022
## 27 -0.172 -0.232 -0.112            pct_male     1 2020
## 28 -0.110 -0.170 -0.050            pct_male     1 2022
## 29  0.048  0.006  0.090         pct_transit     0 2020
## 30  0.032 -0.010  0.074         pct_transit     0 2022
## 31 -0.049 -0.099  0.002         pct_transit     1 2020
## 32 -0.017 -0.067  0.033         pct_transit     1 2022
## 33 -0.044 -0.075 -0.013      pct_unemployed     0 2020
## 34 -0.058 -0.089 -0.027      pct_unemployed     0 2022
## 35 -0.049 -0.094 -0.004      pct_unemployed     1 2020
## 36 -0.025 -0.070  0.020      pct_unemployed     1 2022
```

Plot standardized estimates

```r
prim_ci.adj.st %>%
  ggplot(aes(x = Est, y = variable)) +
  geom_vline(aes(xintercept = 0), linetype = "dashed") +
```

```
geom_point(aes(shape = factor(bmore), color = factor(bmore)),position = position_dodge(width = 0.5)) +
geom_errorbarh(aes(xmin = Lower, xmax = Upper, color = factor(bmore)), height = 0.3, position = positi
labs(title = "Adjusted Standardized Estimates\nfor 2020/2022 Primary Elections", color = "Is in Baltin
guides(shape = "none") +
facet_wrap(~year) +
theme_bw()
```