Gerrymandering Analysis: 1790-1860

Introduction

Gerrymandering is a hot topic in statehouses, academia and the Supreme Court. It was named for Elbridge Gerry, Massachusetts Governor from 1810-1812 and alleged gerrymandering mastermind, but as far as I know not much study has been done on gerrymandering in the pre-Civil War era. This analysis aims to scratch the surface and provide a glimpse of the prevalence of gerrymandering in that early stage in our history.

Summaries

Parties Represented:

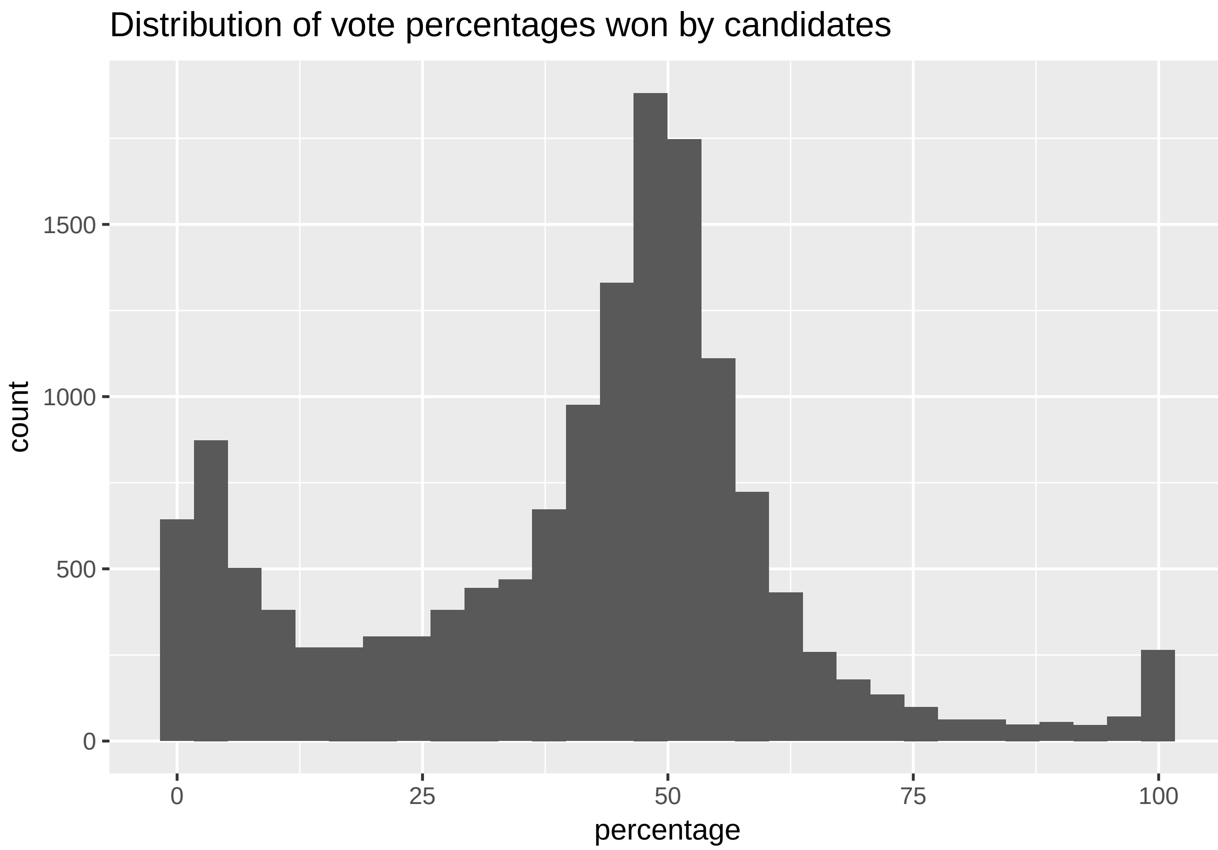
```r
data$party <- as.character(data$party)
data[,"party"][data[,"party"] == "D-R"] <- "DR"
data <- separate(data, party, into=c("party", "subparty"), sep="-", fill="right")
data[,"party"][data[,"party"] == "Ad"] <- "NR"
data[,"party"][data[,"party"] == "f"] <- "F"
data[,"party"][data[,"party"] == "a-f"] <- "AF"
data[,"party"][data[,"party"] == "J"] <- "D"
table(data$party)
```

```
##
##          a     A    Ab   AdD    Ag    AM   A M    AX    Bk   BnD   BrD     C    CC    CD
## 2606    63   608     7     1     3     1     1     1    91    19    18     7     4     1
##    ch    CI    Cl   Cnv    Co   CsD  Csts   Cts   CtU    CU   CuR     D   DgD    DR     F
##     6     6    83     4     1     3    10     2    24     1     2  4403     1  2710  2072
##    FD    FS     I    IA    ID   INR    IR    IU   IUD   IUW    IW     J    JR   LRf   Lty
##     2   226   130     1    55     2    22     3     1     1    29     5     6     7   365
##    Lw    NR   NRf    NV    OD   Opp    OS    Pe   PeD    Pt   ptm     Q     R   Rad    RU
##     3   558     1     4     1    36     1    47     1     1     5    16   488     6     2
##   ScD   ScW   SoR  SoRD    SR   StR     T   TfD   TyD     U    UD    Un    UR   USC   UUD
##     1     1    41    12     2     3     6     1     1   283    22    29    37     4     1
##   UVD    UW    VI    VT    VW     W    WA    WD   Wkm   Wks
##     1     7     2     1     1  2291     1     1     4     1
```

```r
ggplot(data, aes(x=percentage)) +
  geom_histogram() +
  ggtitle("Distribution of vote percentages won by candidates")
```

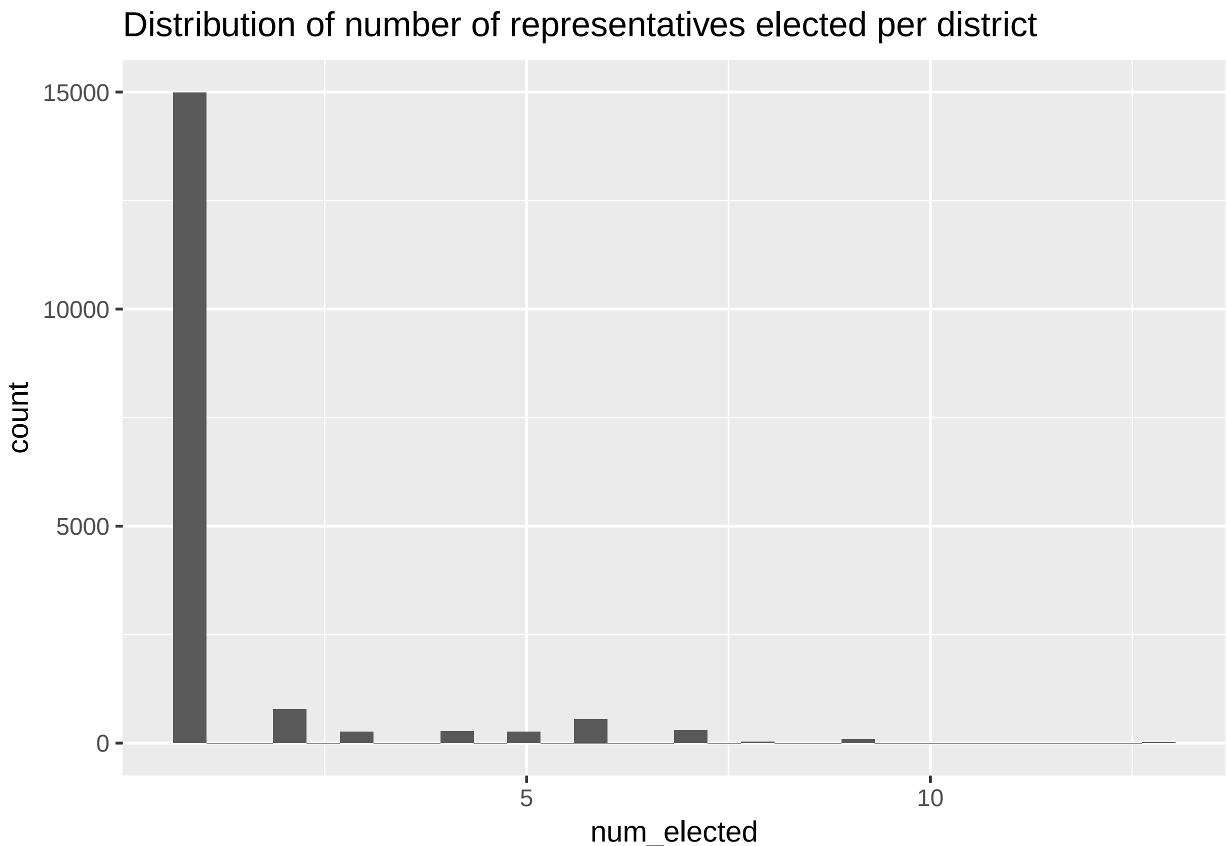
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2542 rows containing non-finite values (stat_bin).
```

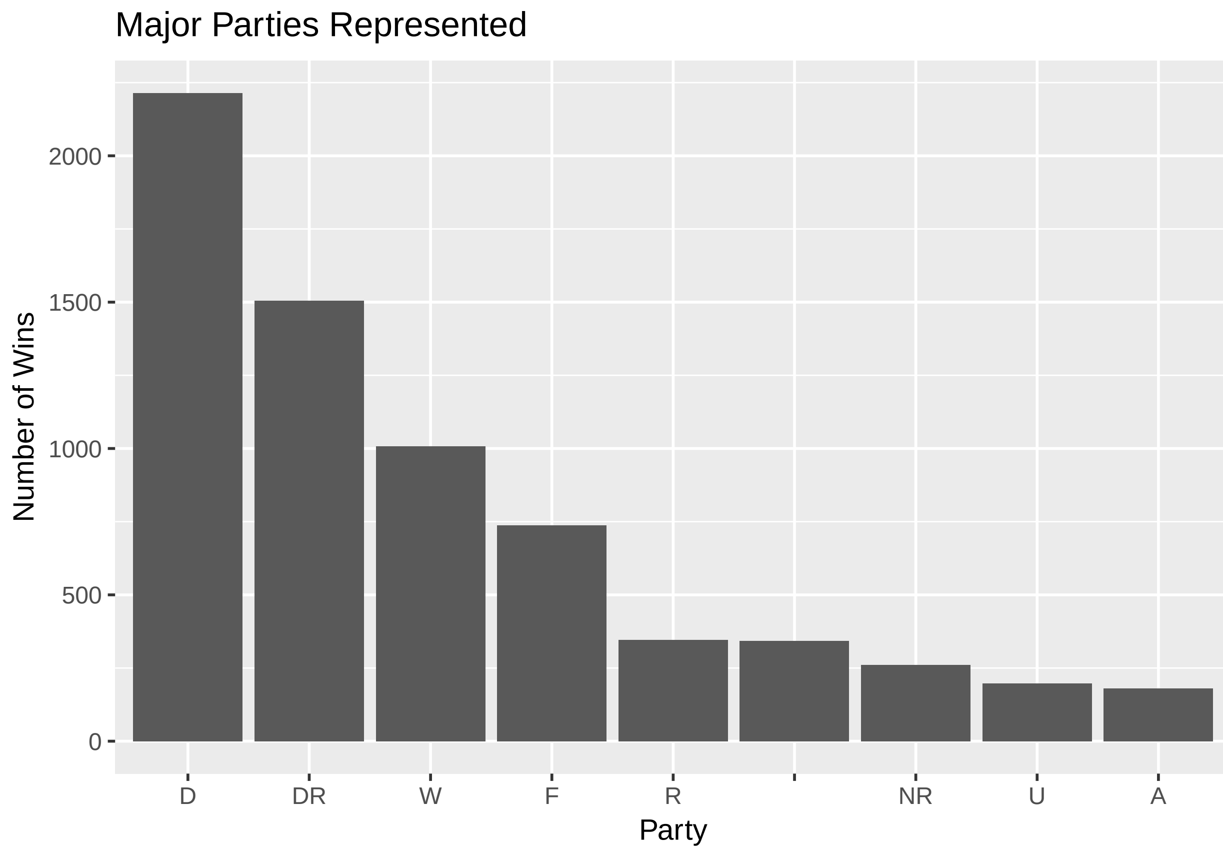



```r
ggplot(data, aes(x=num_elected)) +
  geom_histogram() +
  ggtitle("Distribution of number of representatives elected per district")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
party_summary <- data %>%
  group_by(party) %>%
  summarise(won=sum(result=="won", na.rm=TRUE)) %>%
  filter(won > sum(data$result == "won", na.rm=TRUE)*.1/8)

ggplot(party_summary, aes(x=reorder(party, -won), y = won)) +
  geom_bar(stat="identity") +
  ggtitle("Major Parties Represented") +
  xlab("Party") +
  ylab("Number of Wins")
```



Party Key:

D: Democrat

DR: Democratic-Republican

W: Whig

F: Federalist

R: Republican

(Blank): No party

NR: National Republican (also called Anti-Jacksonian)

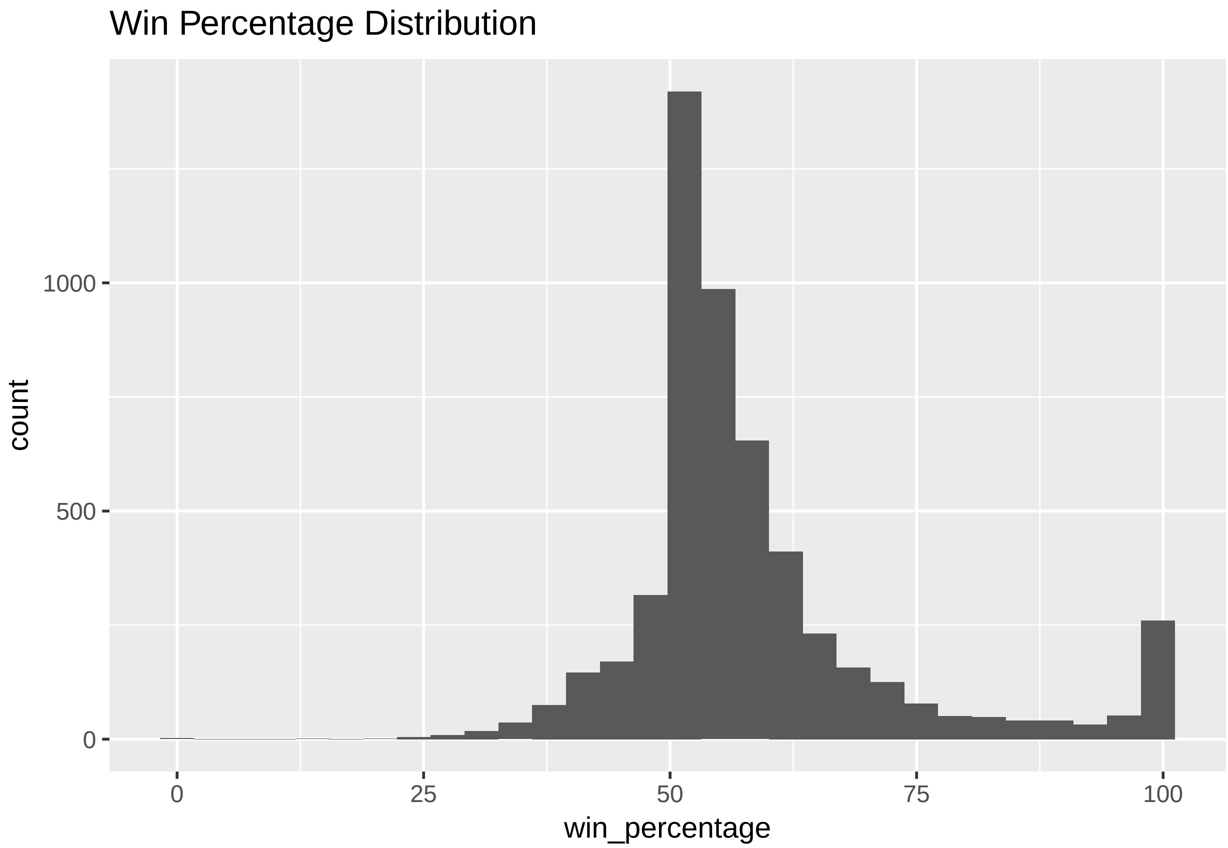
U: Union

A: American (also known as Know-Nothing)

```
data <- transform(data,
                  election_id = as.numeric(interaction(congress, trial,
                                                       year, state,
                                                       district, drop=TRUE)))

data$congress <- as.factor(data$congress)
summary <- data %>%
  group_by(election_id) %>%
  summarise(win_percentage=max(percentage))

ggplot(summary, aes(x=win_percentage)) +
  geom_histogram() +
```

```
  ggtitle("Win Percentage Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1252 rows containing non-finite values (stat_bin).
```




```
data <- data %>%
  group_by(election_id) %>%
  mutate(inferred_percentage=ifelse(!is.na(percentage) | num_elected > 1,
                                    percentage,
                                    votes/sum(votes)*100*num_elected))
```

```
data <- transform(data, state_id = as.numeric(interaction(congress, state, drop=TRUE)))

to_join <- data %>%
  group_by(state_id) %>%
  summarise(num_elected_state=sum(result=="won"))

data_joined <- inner_join(data, to_join, by="state_id")

grouped_states <- data_joined %>% group_by(state, congress) %>%
  mutate(sum_votes = sum(votes * (type=="StandardElections"),
                         na.rm=TRUE)) %>%
  group_by(state, congress) %>%
  mutate(num_districts = length(unique(district))) %>%
  group_by(state, congress, party) %>%
  summarise(theoretical = sum(votes * (type=="StandardElections"),
                              na.rm=TRUE)/mean(sum_votes),
```

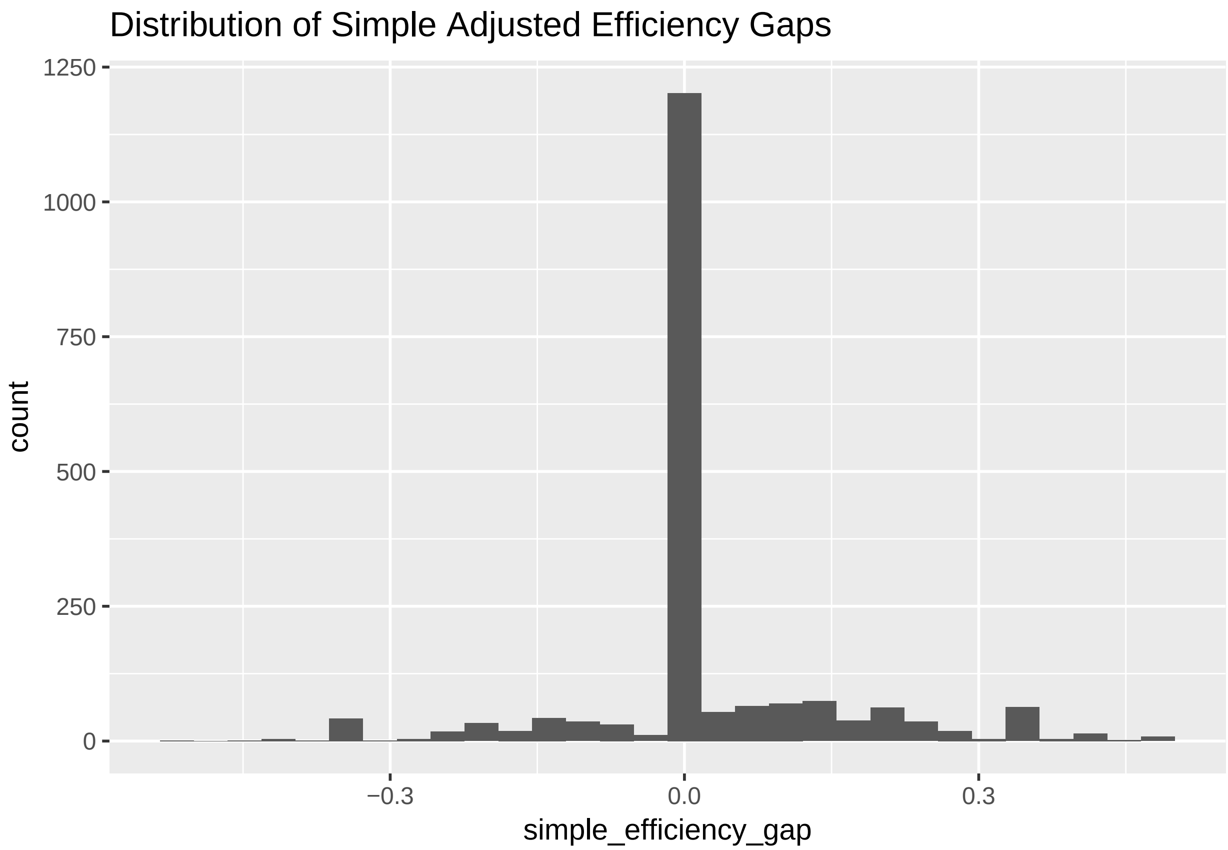
```
            actual=sum(result=="won")/max(num_elected_state),
            num_elected_state=max(num_elected_state),
            num_districts = max(num_districts)) %>%
  mutate(x= theoretical, theoretical = (theoretical-.5)*2+.5) %>%
  mutate(theoretical = ifelse(theoretical > 1, 1,
                              ifelse(theoretical < 0, 0, theoretical)),
         mean_elected_district=num_elected_state/num_districts)
```

Calculating the Efficiency Gap

First, I use the efficiency gap calculation proposed by Nicholas Stephanopoulos and Eric McGhee, specifically the simplified version described here. I calculate the efficiency gap for a given state, then floor it so that it represents a whole number of seats. For example, if the efficiency gap is 0.27 for a state with 4 seats, that gap represents 1.08 seats. This number is floored to one seat, to make a floored gap of 0.25.

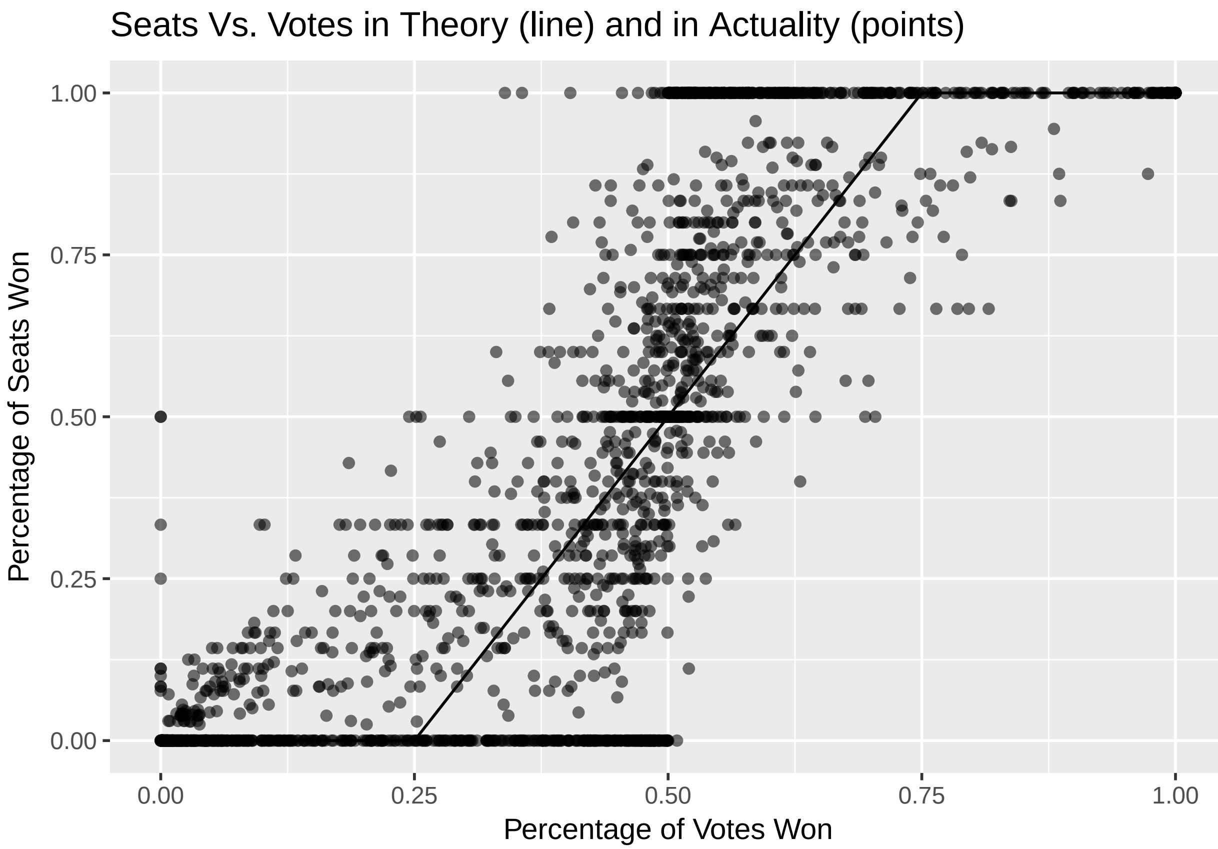
```
# party %in% c("D-R", "F", "J", "NR", "Ad")
filtered <- subset(grouped_states, party != "" &
                     !is.na(theoretical) &
                     !is.na(actual) &
                     num_districts>0)

filtered$sign <- ifelse(filtered$actual-filtered$theoretical > 0, 1, -1)
filtered$simple_efficiency_gap <- filtered$sign *
  floor(abs(filtered$actual-filtered$theoretical)*filtered$num_elected_state)/
  filtered$num_elected_state
ggplot(filtered, aes(x=simple_efficiency_gap)) +
  geom_histogram() +
  ggtitle("Distribution of Simple Adjusted Efficiency Gaps")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Below is a plot of comparing the percentages of seats won to the percentage of votes won, along with the prediction from the basic model.

```
ggplot(filtered, aes(x=x, y=actual)) +
  geom_point(aes(alpha=0.001)) +
  xlab("Percentage of Votes Won") +
  ylab("Percentage of Seats Won") +
  guides(alpha=FALSE) +
  geom_line(aes(x=x, y=theoretical)) +
  ggtitle("Seats Vs. Votes in Theory (line) and in Actuality (points)")
```



Next, I filter the data to include only frequent parties, and collect the data by party, state, and census.

```
filtered <- filtered %>%
  filter(party %in% c("D", "DR", "W", "F", "R", "NR","U", "A"))
data$census <- floor((as.integer(data$congress)+2)/5)
filtered$census <- floor((as.integer(filtered$congress)+2)/5)
filtered$seat_advantage <- filtered$simple_efficiency_gap*filtered$num_elected_state
```

In my observation, it appears that states with fewer districts typically have seat percentage outcomes that are closer to zero or one. I build a new model predicting seat percentage, taking this into account, and calculate efficiency gaps (difference between predicted and actual seat share) for both the traditional and modified predictions.

```
model <- lm(abs(filtered$actual-.5) ~ abs(filtered$x-.5) + log(filtered$num_districts))
summary(model)
```

```
##
## Call:
## lm(formula = abs(filtered$actual - 0.5) ~ abs(filtered$x - 0.5) +
##     log(filtered$num_districts))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49428 -0.10054  0.02301  0.10485  0.29023
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.383229   0.007133   53.72   <2e-16 ***
```

```
## abs(filtered$x - 0.5)        0.455273   0.023210   19.61   <2e-16 ***
## log(filtered$num_districts) -0.077570   0.003150  -24.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1379 on 1575 degrees of freedom
## Multiple R-squared:  0.4101, Adjusted R-squared:  0.4093
## F-statistic: 547.4 on 2 and 1575 DF,  p-value: < 2.2e-16
```

```r
new_predictions <- .5+ifelse(filtered$x < .5, -1, 1)*predict(model)
new_predictions <- ifelse(new_predictions < 0, 0, ifelse(new_predictions > 1, 1, new_predictions))
filtered$theoretical_2 <- new_predictions
model <- lm(actual ~ 0 + theoretical_2, filtered)
summary(model)
```

```
##
## Call:
## lm(formula = actual ~ 0 + theoretical_2, data = filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90272 -0.09961 -0.00760  0.09678  0.95549
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## theoretical_2 1.022013   0.007905   129.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1835 on 1577 degrees of freedom
## Multiple R-squared:  0.9138, Adjusted R-squared:  0.9137
## F-statistic: 1.672e+04 on 1 and 1577 DF,  p-value: < 2.2e-16
```

```r
summary(lm(actual ~ 0 + theoretical, filtered))
```

```
##
## Call:
## lm(formula = actual ~ 0 + theoretical, data = filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5297 -0.1026  0.0000  0.1429  0.8111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## theoretical   1.0590     0.0105   100.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2289 on 1577 degrees of freedom
## Multiple R-squared:  0.8659, Adjusted R-squared:  0.8658
## F-statistic: 1.018e+04 on 1 and 1577 DF,  p-value: < 2.2e-16
```

```r
filtered$sign <- ifelse(filtered$actual-filtered$theoretical_2 > 0, 1, -1)
filtered$modified_efficiency_gap <- filtered$sign *
  floor(abs(filtered$actual-filtered$theoretical_2)*filtered$num_elected_state)/
```

```
  filtered$num_elected_state

state_profile <- filtered %>%
  group_by(party, census, state) %>%
  summarise(mean_simple_gap=mean(simple_efficiency_gap, na.rm=TRUE),
            mean_modified_gap=mean(modified_efficiency_gap, na.rm=T),
            theoretical=mean(theoretical, na.rm=TRUE),
            actual=mean(actual, na.rm=TRUE),
            num_districts=mean(num_districts),
            seat_advantage=mean(seat_advantage))

write.csv(data_joined, "data_joined.csv")
```

```
states <- map_data("state")
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:plyr':
##
##     ozone
```

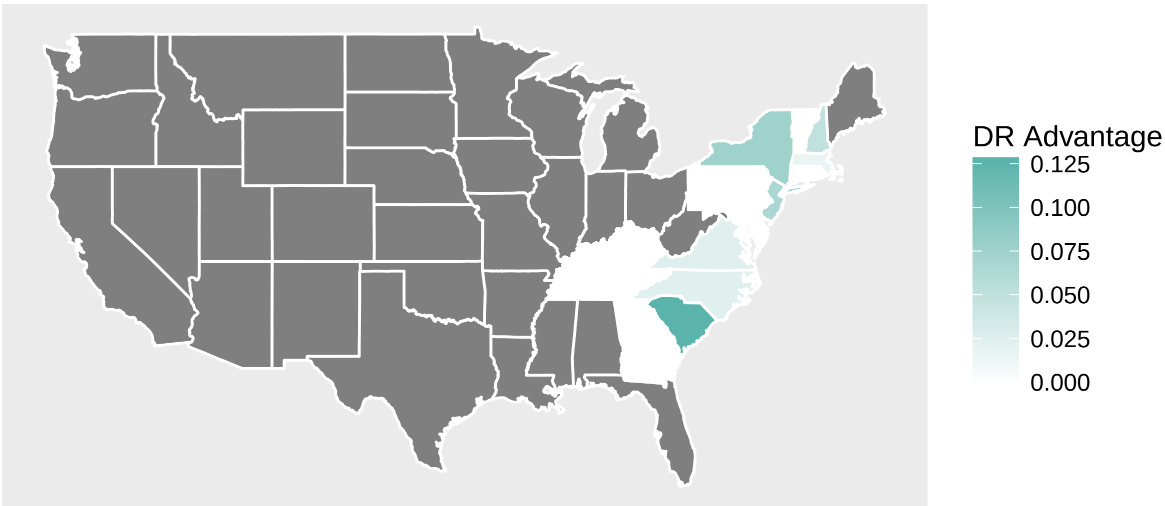
```
state_profile$region <- sapply(state_profile$state, tolower)
write.csv(filtered, "filtered.csv")
```

Below are maps representing the efficiency gaps from each party present at each census:

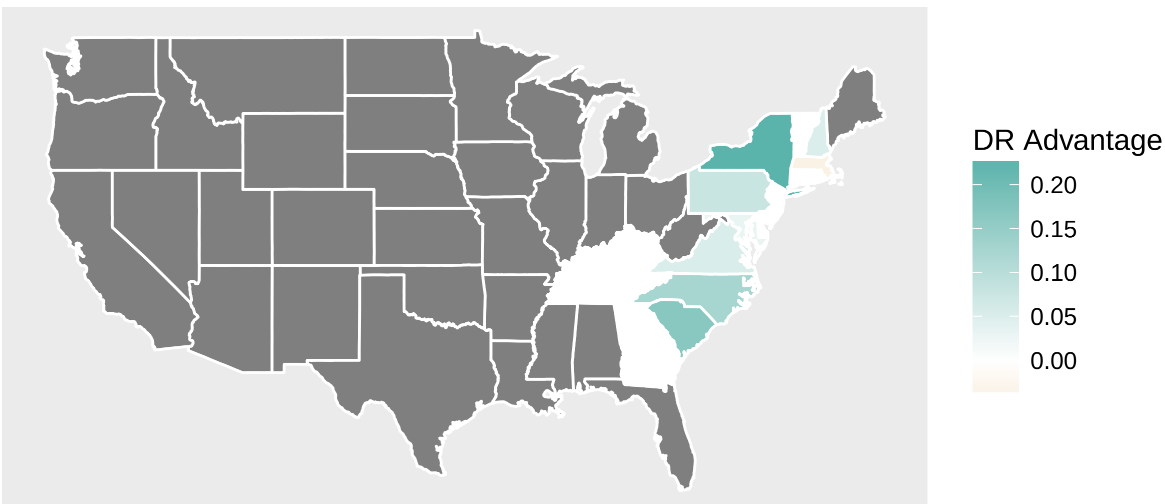
```
for (this_census in 1:7) {
  for (this_party in unique(state_profile$party)) {
    states_with_data <- left_join(states,
                                  subset(state_profile, census==this_census & party==this_party),
                                  by = "region")
    if (sum(!is.na(states_with_data$mean_modified_gap)) > 0) {
      plot.new()
      print(
        ggplot(data = states_with_data) +
          geom_polygon(aes(x = long, y = lat,
                          group = group, fill=mean_simple_gap),
                      color = "white") +
          coord_fixed(1.3) +
          scale_fill_gradient2(high="#5ab4ac", low="#d8b365") +
          theme(panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                axis.line=element_blank(),
                axis.text.x=element_blank(),
                axis.text.y=element_blank(),
                axis.ticks=element_blank()) +
          xlab("") +
          ylab("") +
          labs(fill= this_party %s+% " Advantage") +
          ggtitle("Census #" %s+% this_census %s+% " " %s+% this_party %s+% ": simple")
        )
      print(
        ggplot(data = states_with_data) +
          geom_polygon(aes(x = long, y = lat,
                          group = group, fill=mean_modified_gap),
```

```
                        color = "white") +
          coord_fixed(1.3) +
          scale_fill_gradient2(high="#5ab4ac", low="#d8b365") +
          theme(panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                axis.line=element_blank(),
                axis.text.x=element_blank(),
                axis.text.y=element_blank(),
                axis.ticks=element_blank()) +
          xlab("") +
          ylab("") +
          labs(fill= this_party %s+% " Advantage") +
          ggtitle("Census #" %s+% this_census %s+% " " %s+% this_party  %s+% ": modified")
        )
    }
  }
}
```

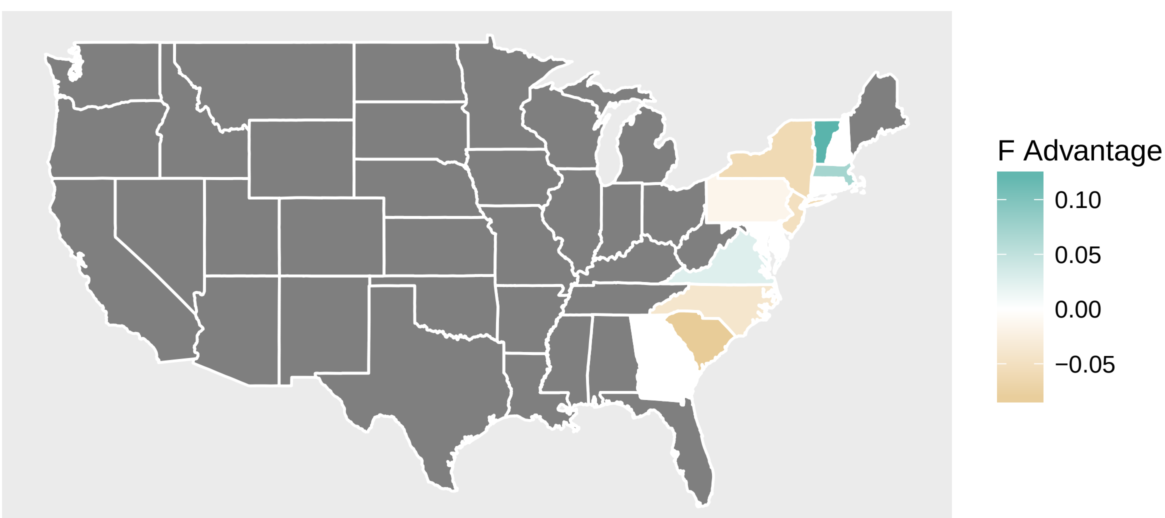
Census #1 DR: simple



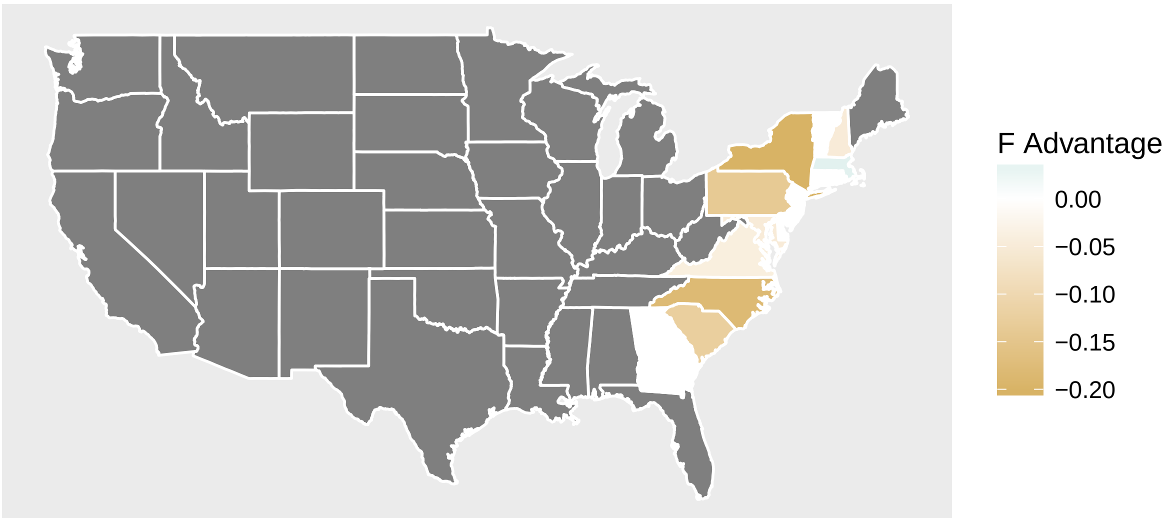
Census #1 DR: modified



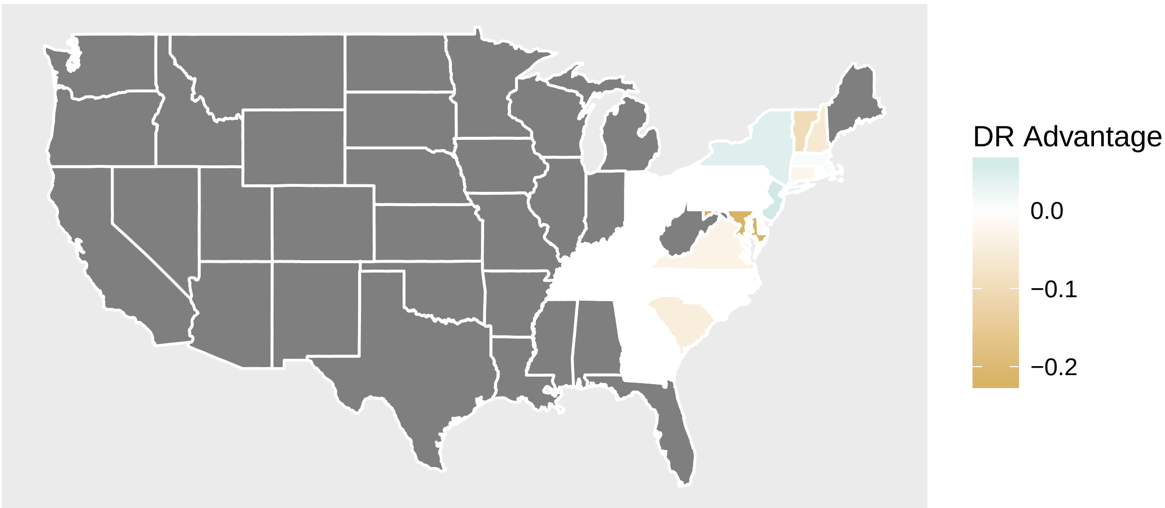
Census #1 F: simple



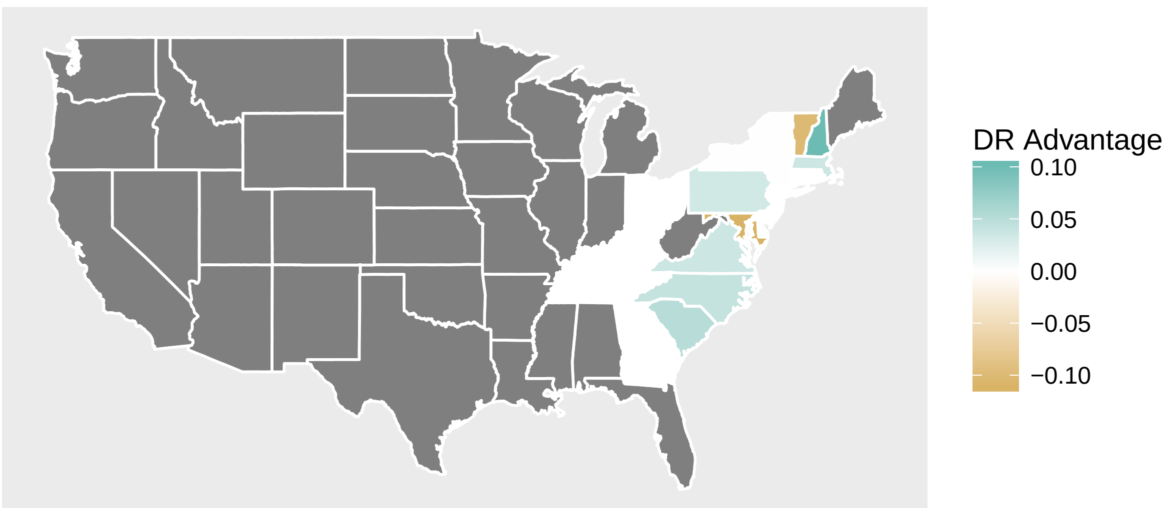
Census #1 F: modified



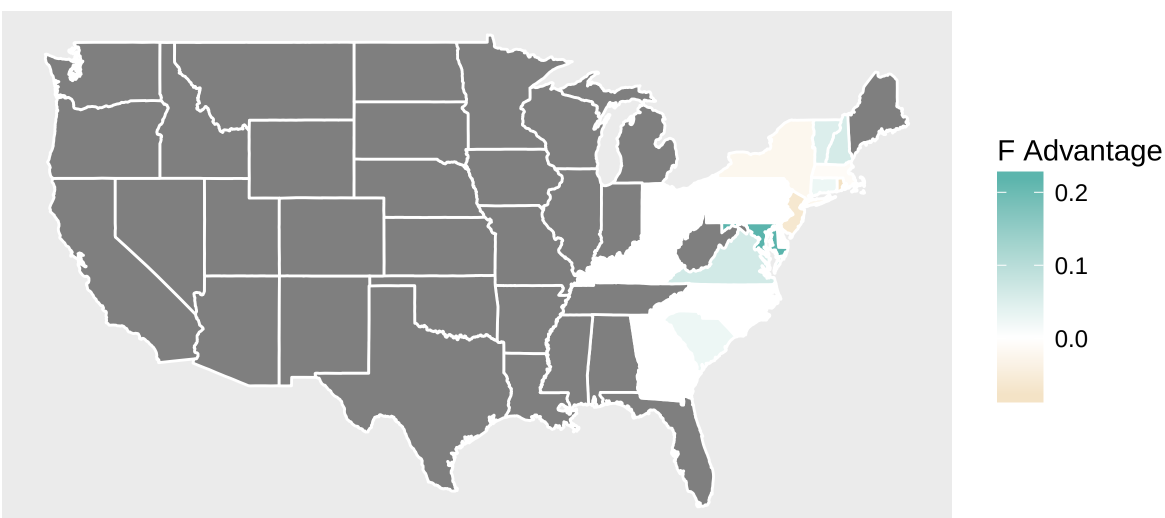
Census #2 DR: simple



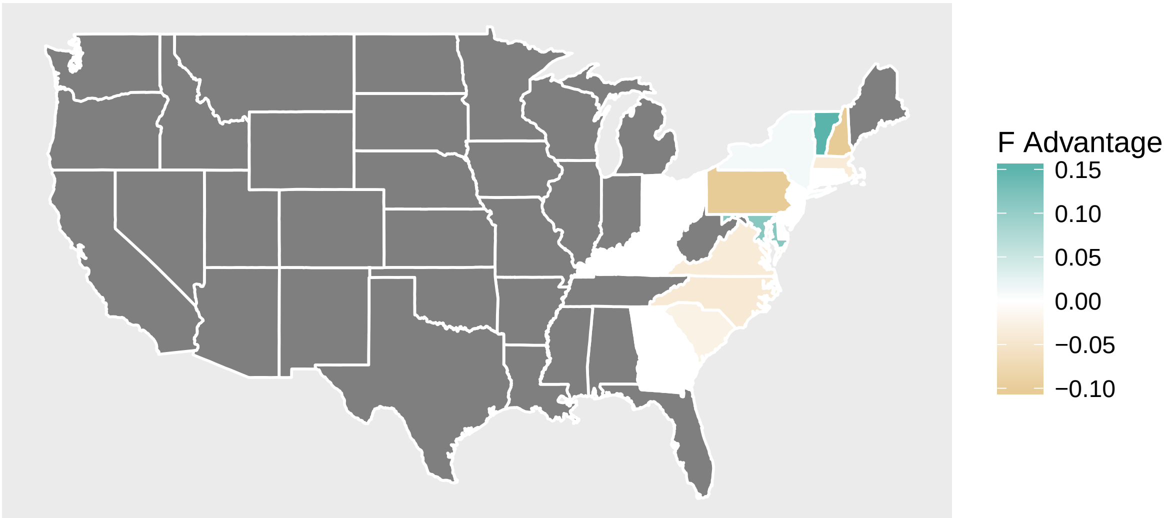
Census #2 DR: modified



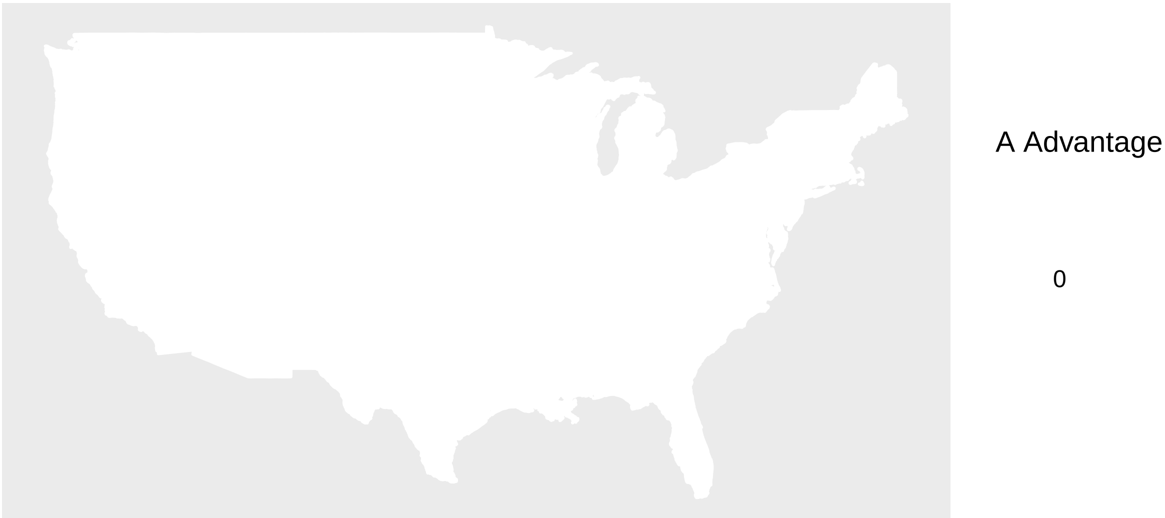
Census #2 F: simple



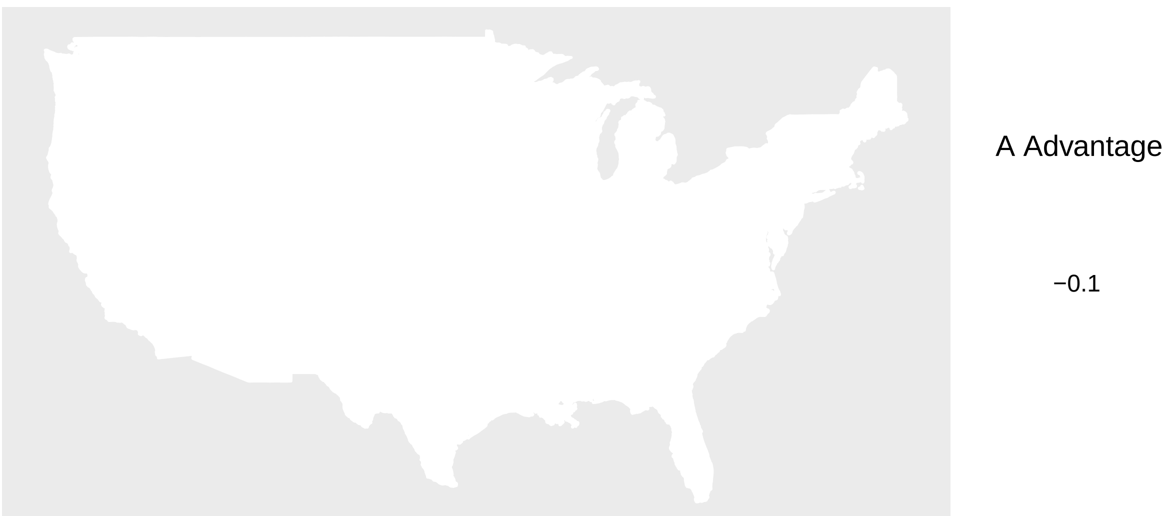
Census #2 F: modified



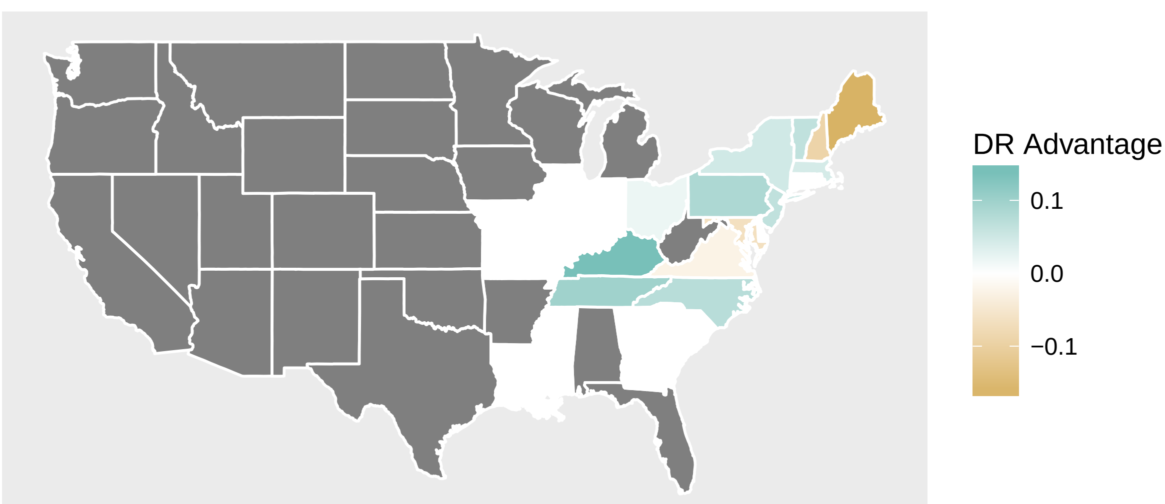
Census #3 A: simple



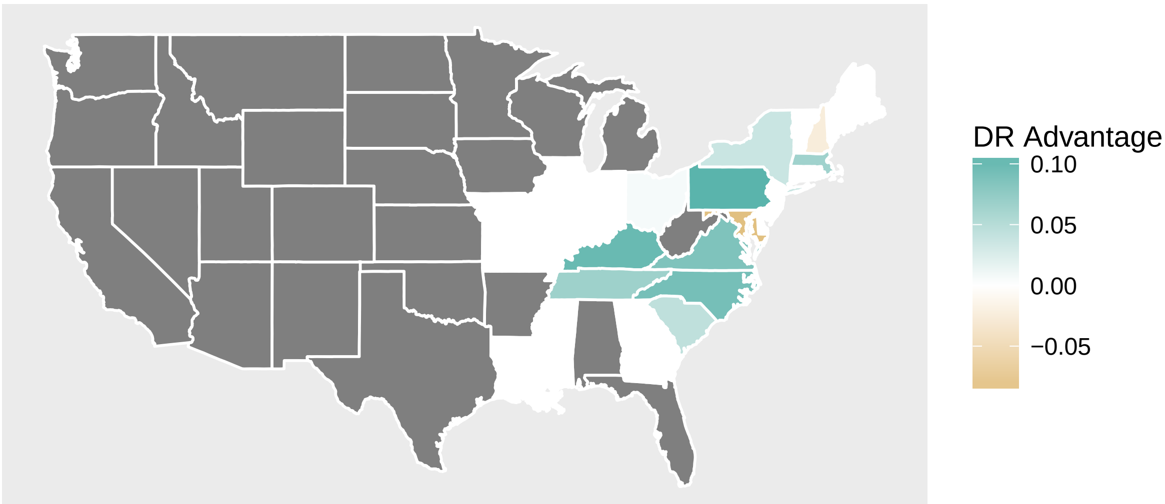
Census #3 A: modified



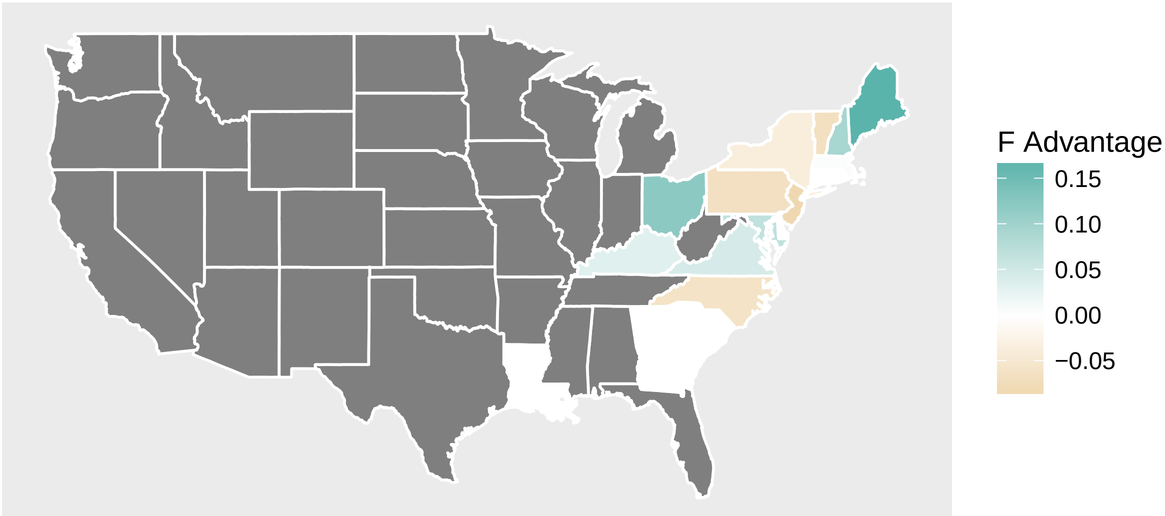
Census #3 DR: simple



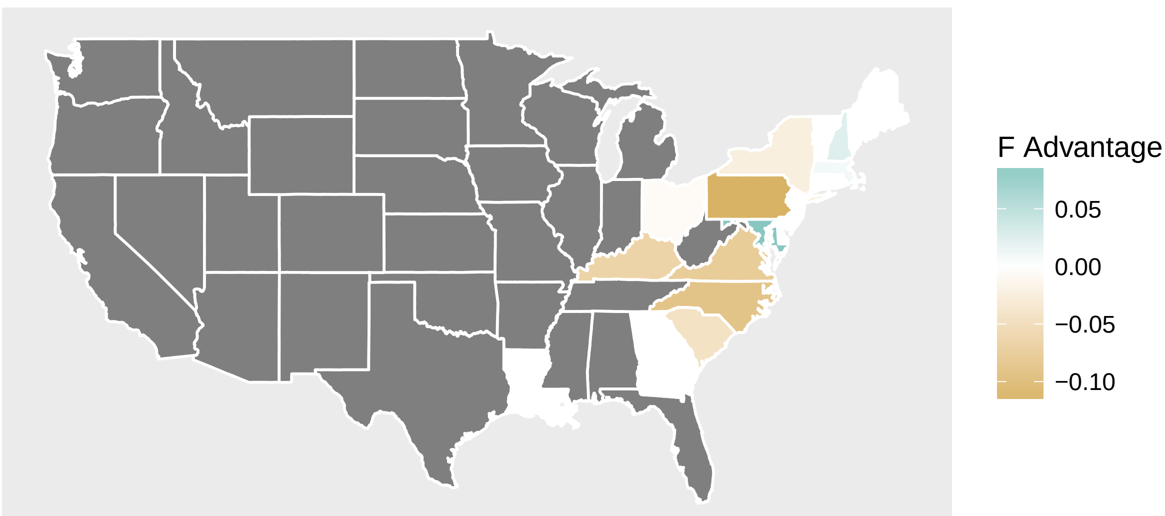
Census #3 DR: modified



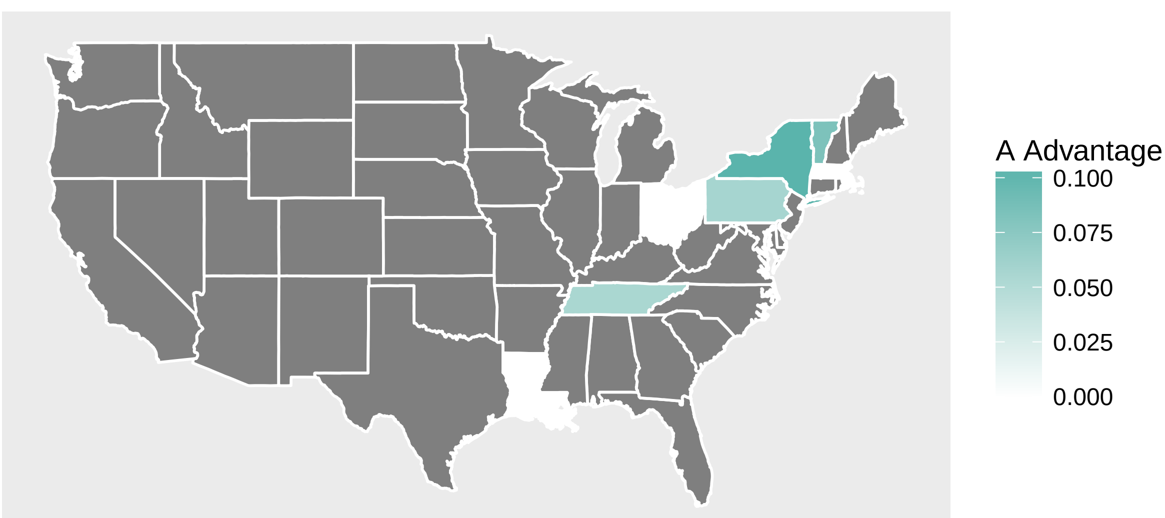
Census #3 F: simple



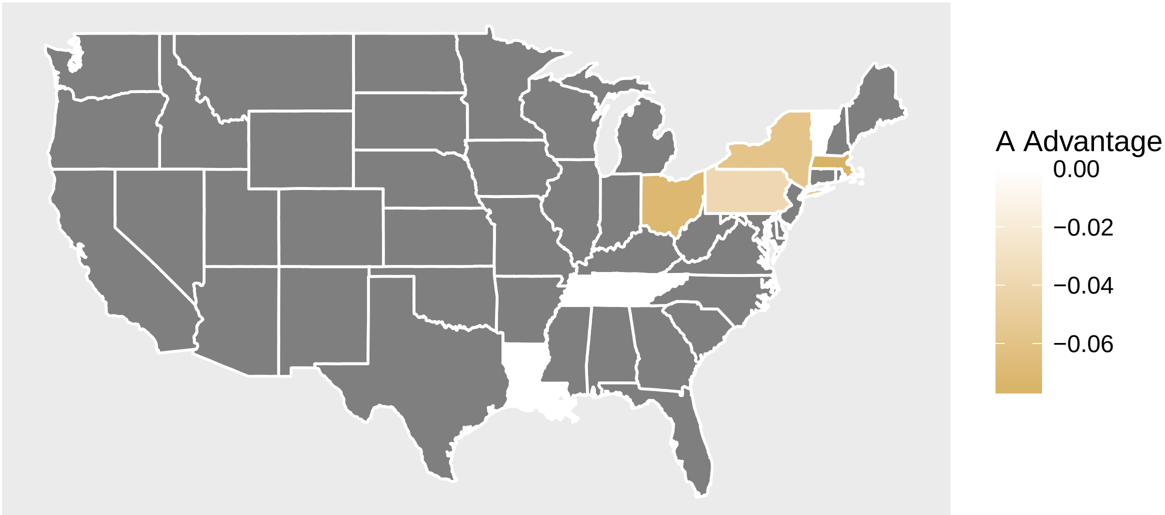
Census #3 F: modified



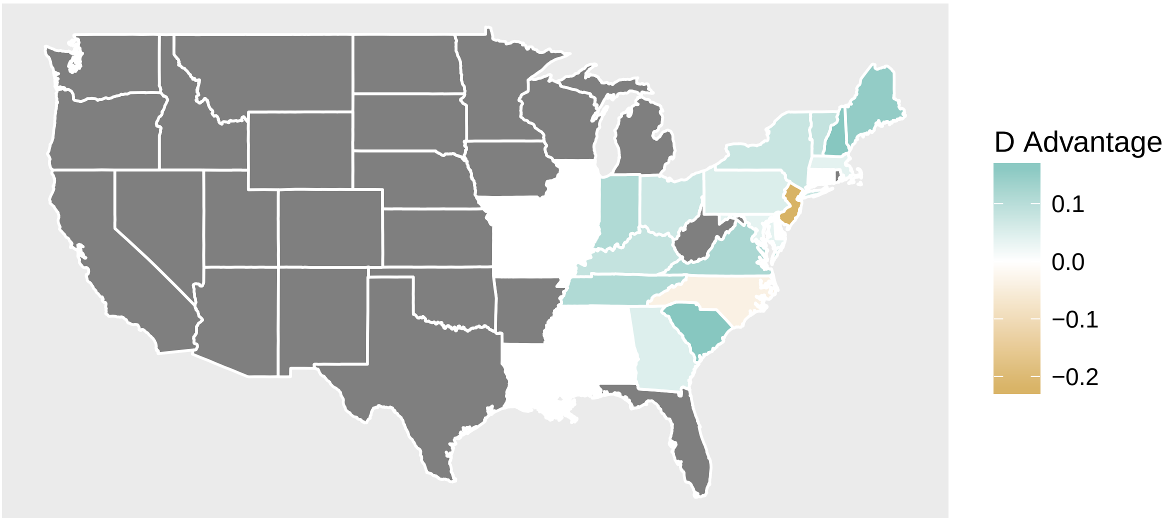
Census #4 A: simple



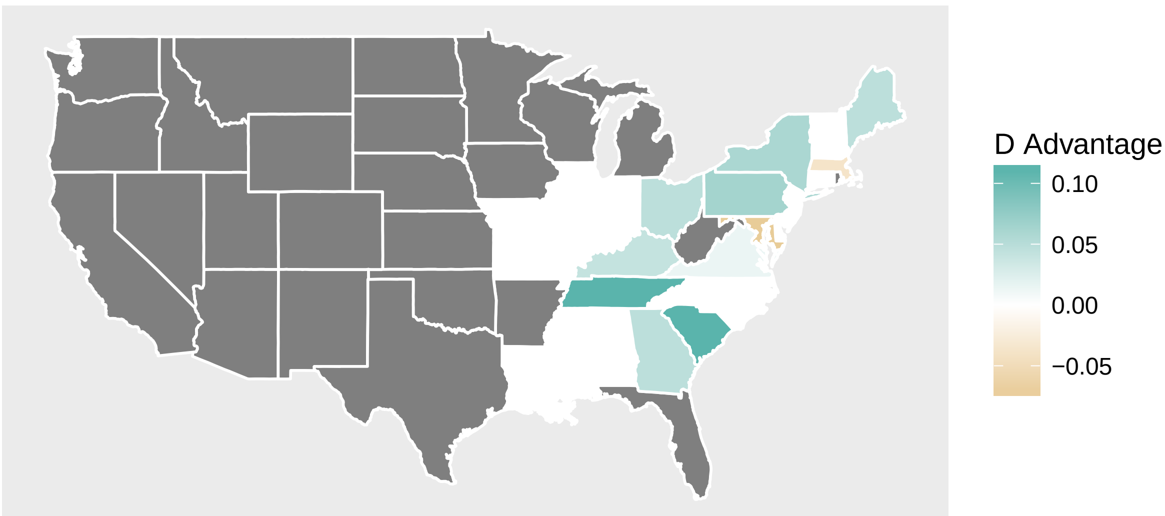
Census #4 A: modified



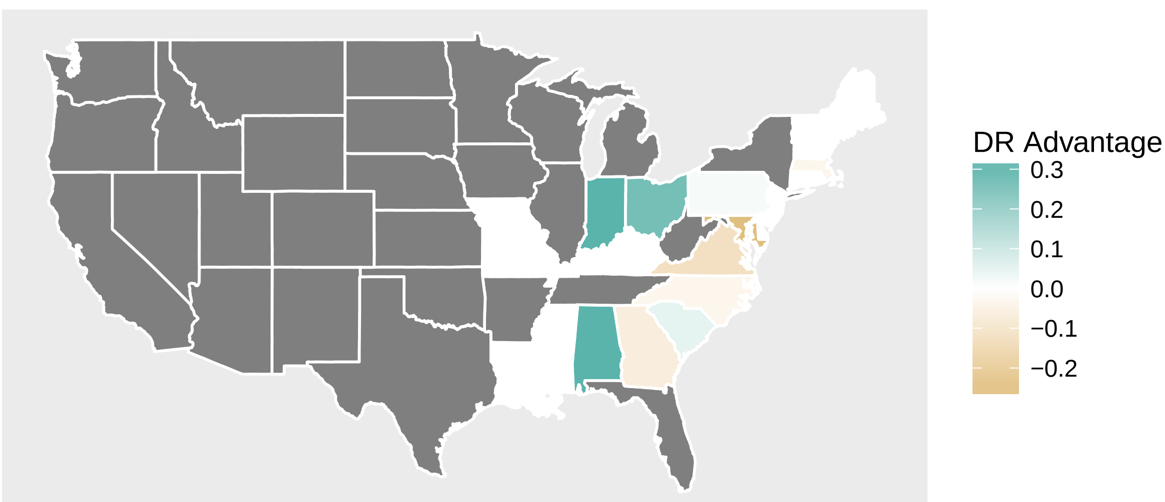
Census #4 D: simple



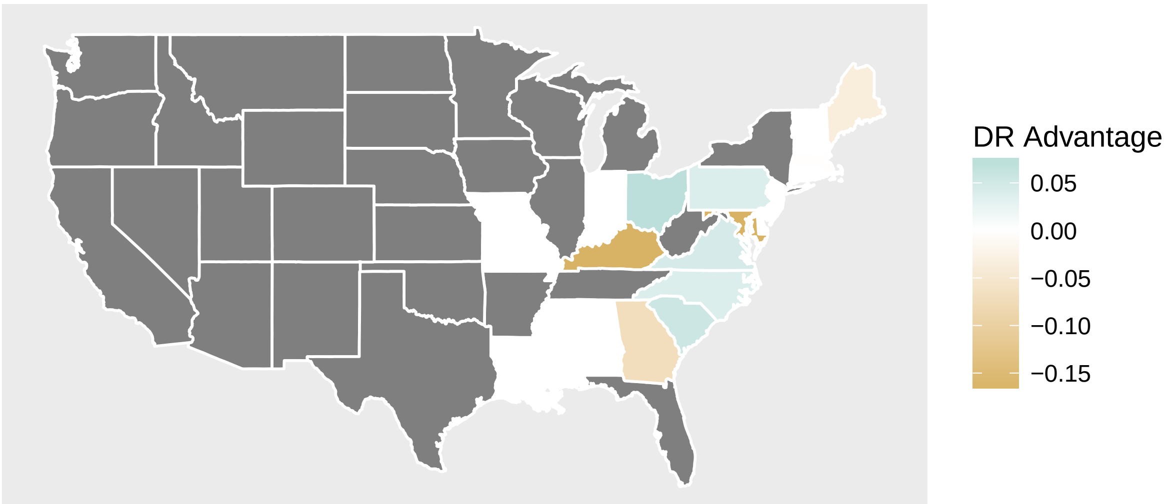
Census #4 D: modified



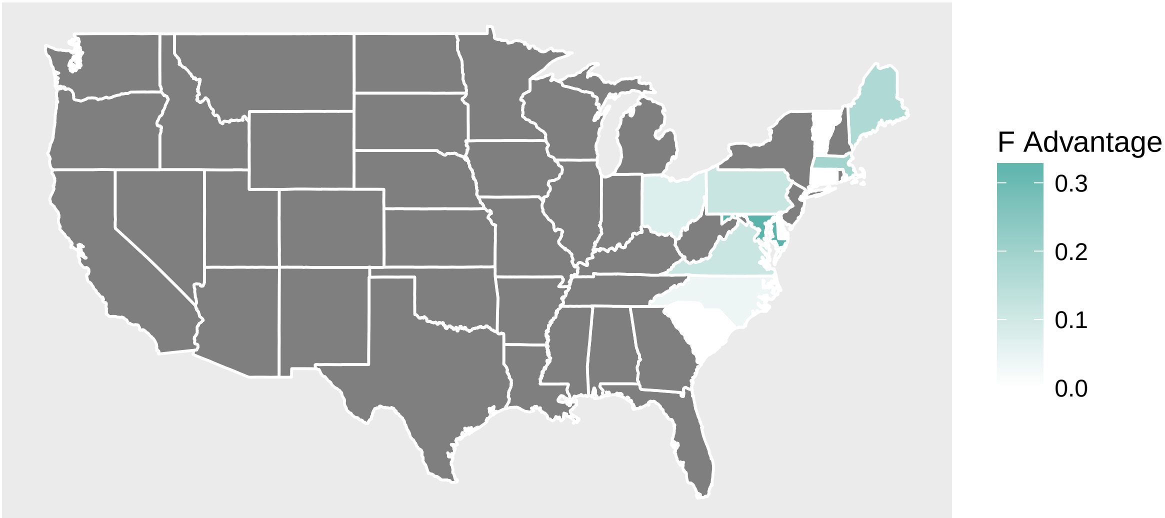
Census #4 DR: simple



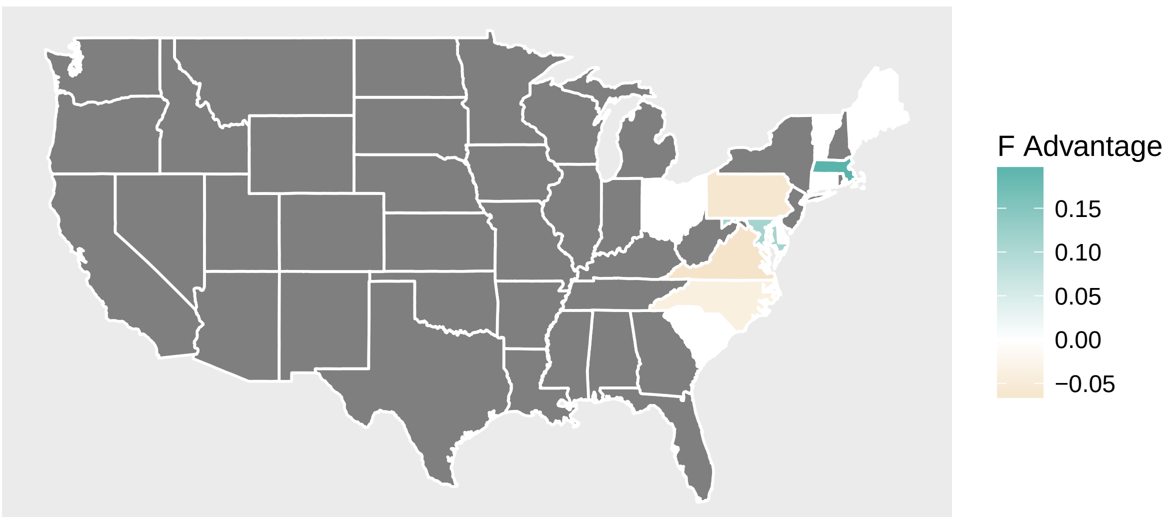
Census #4 DR: modified



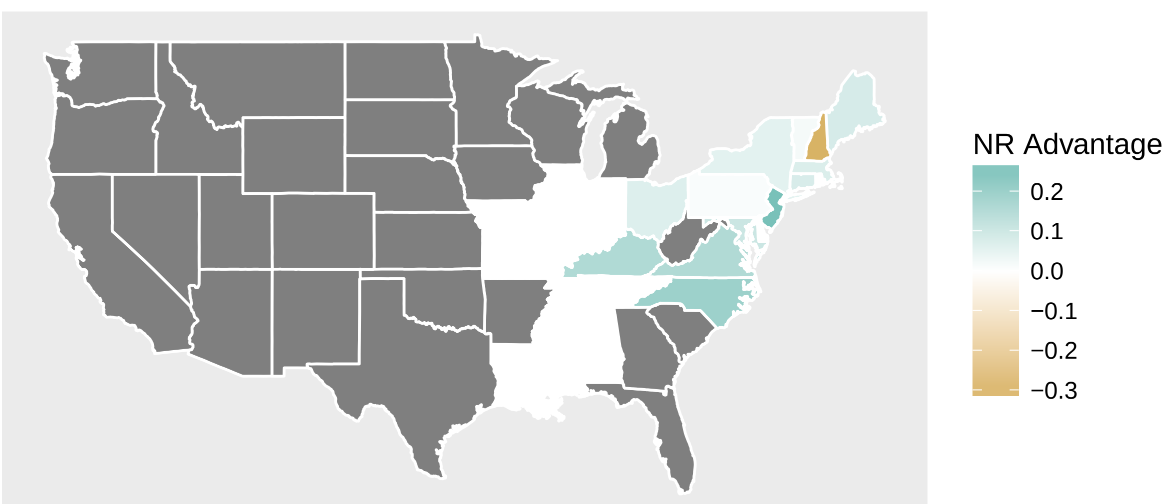
Census #4 F: simple



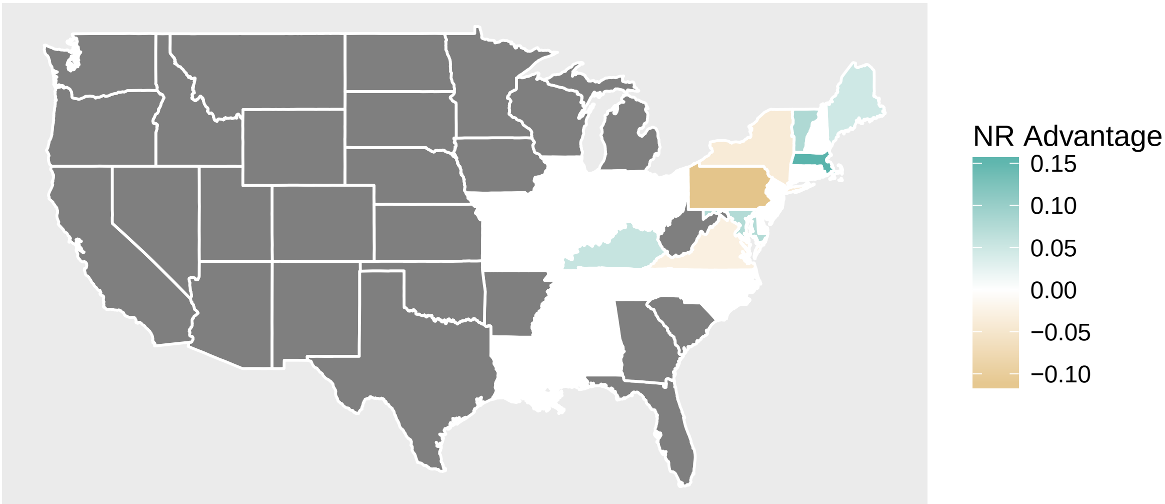
Census #4 F: modified



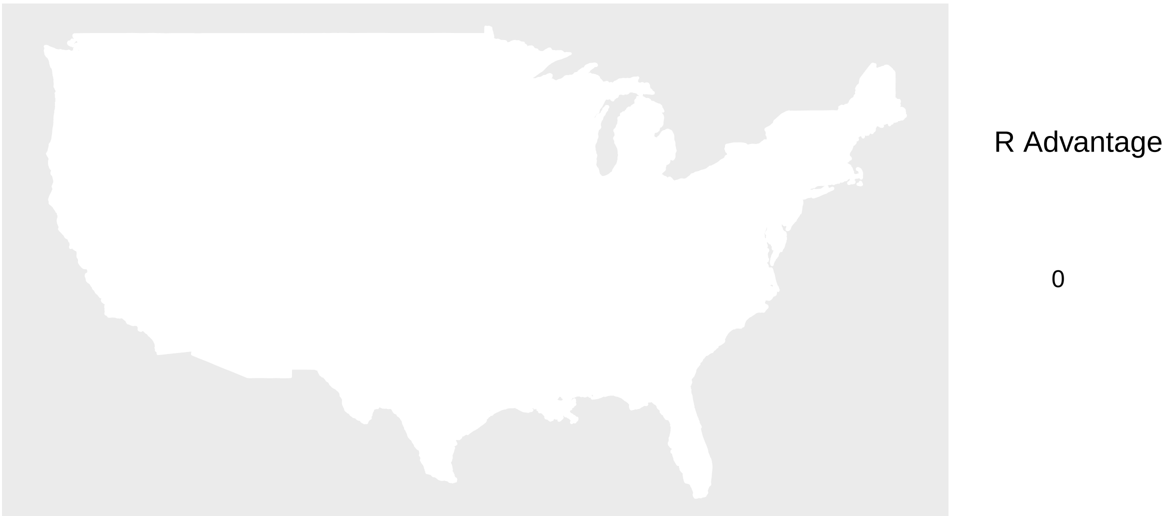
Census #4 NR: simple



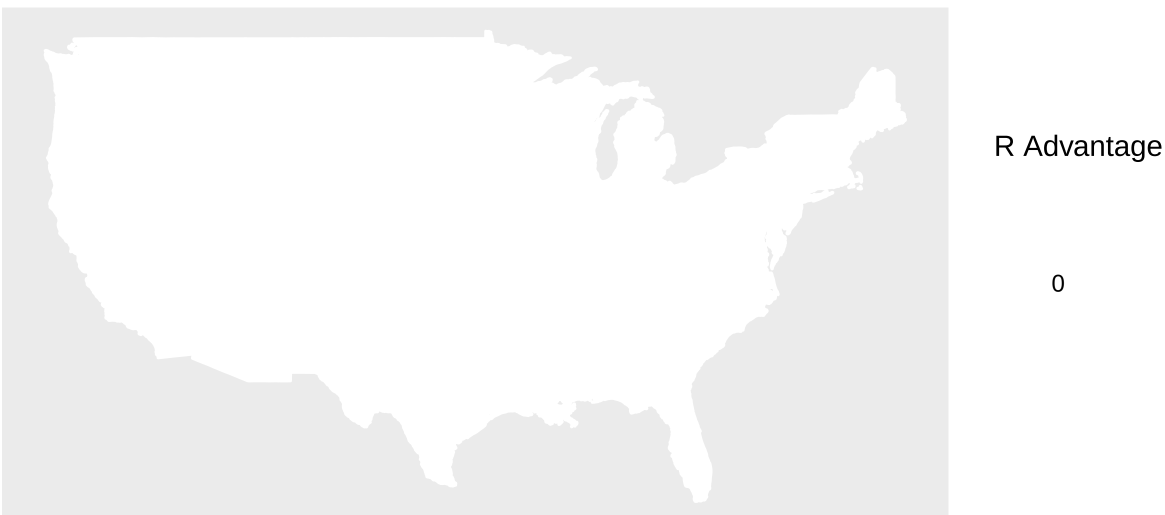
Census #4 NR: modified



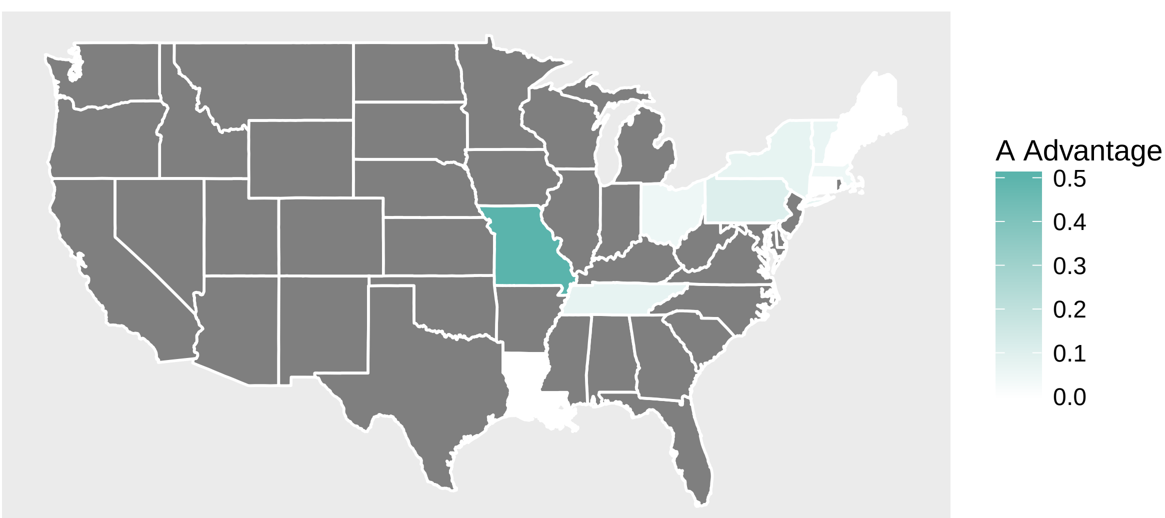
Census #4 R: simple



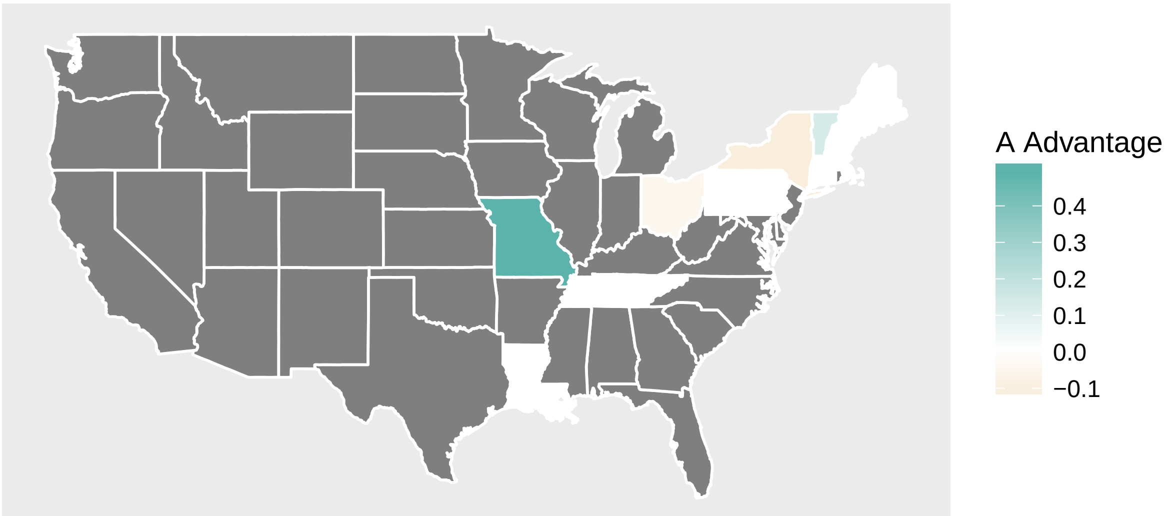
Census #4 R: modified



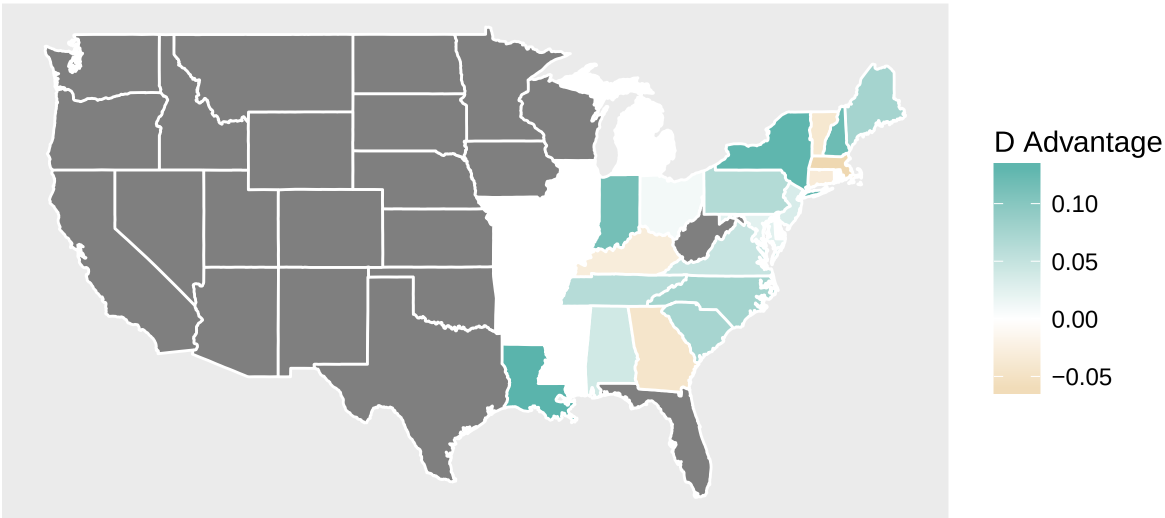
Census #5 A: simple



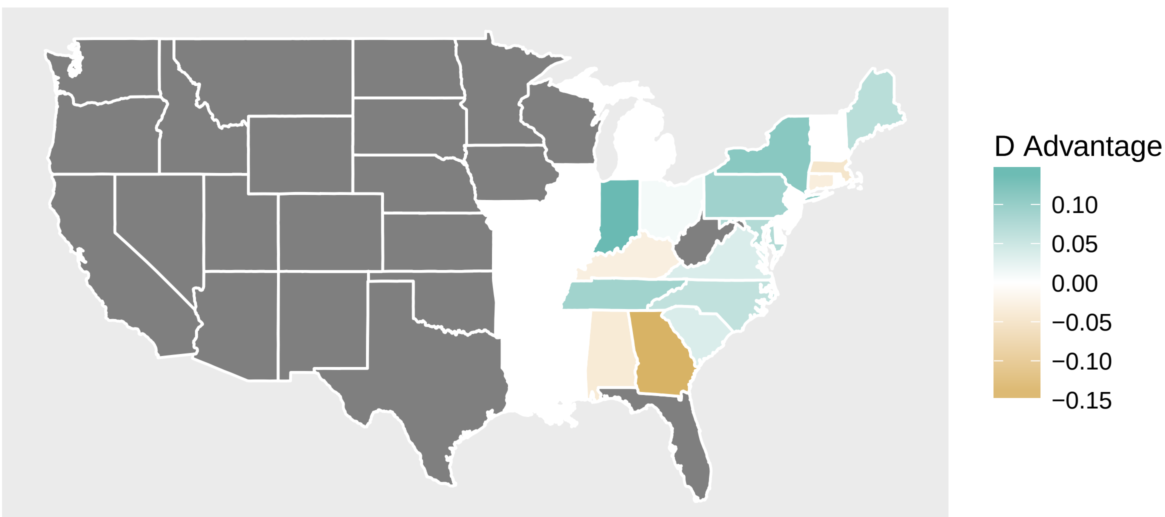
Census #5 A: modified



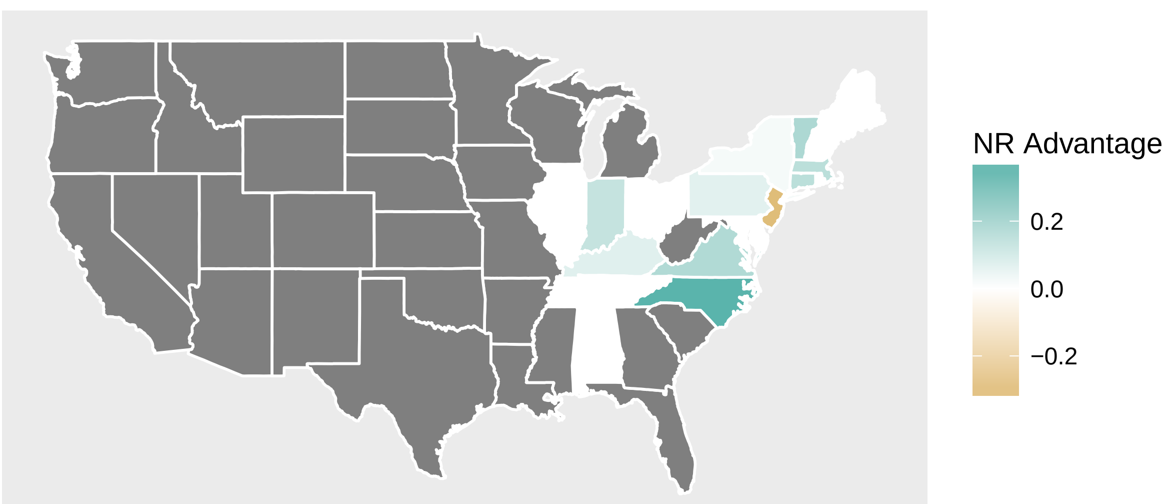
Census #5 D: simple



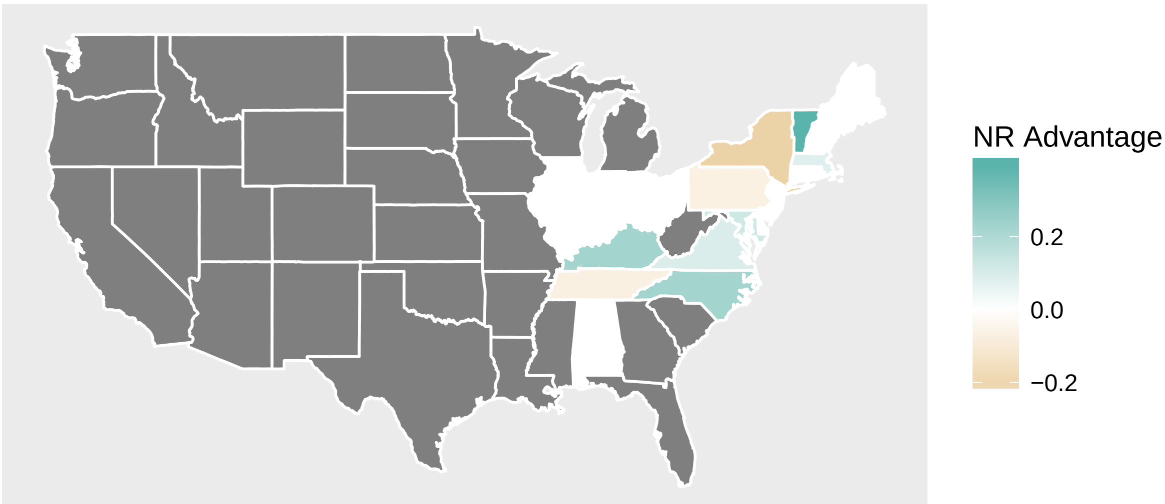
Census #5 D: modified



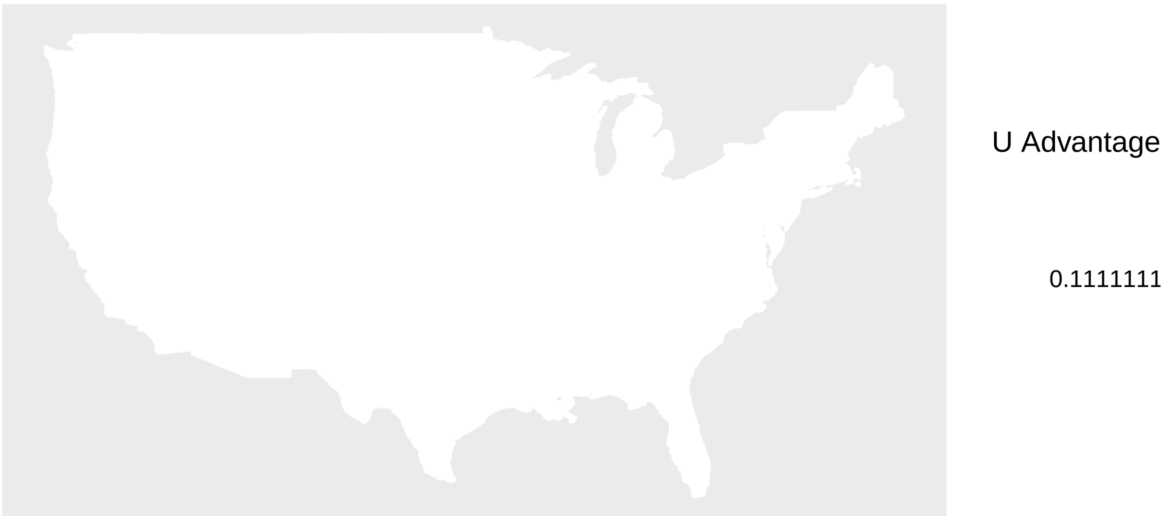
Census #5 NR: simple



Census #5 NR: modified

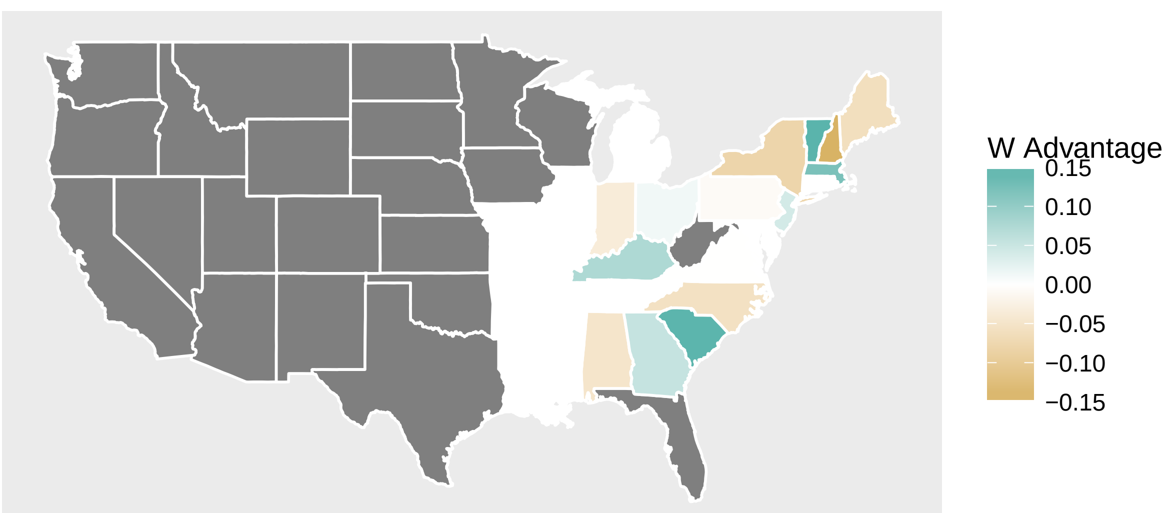


Census #5 U: simple

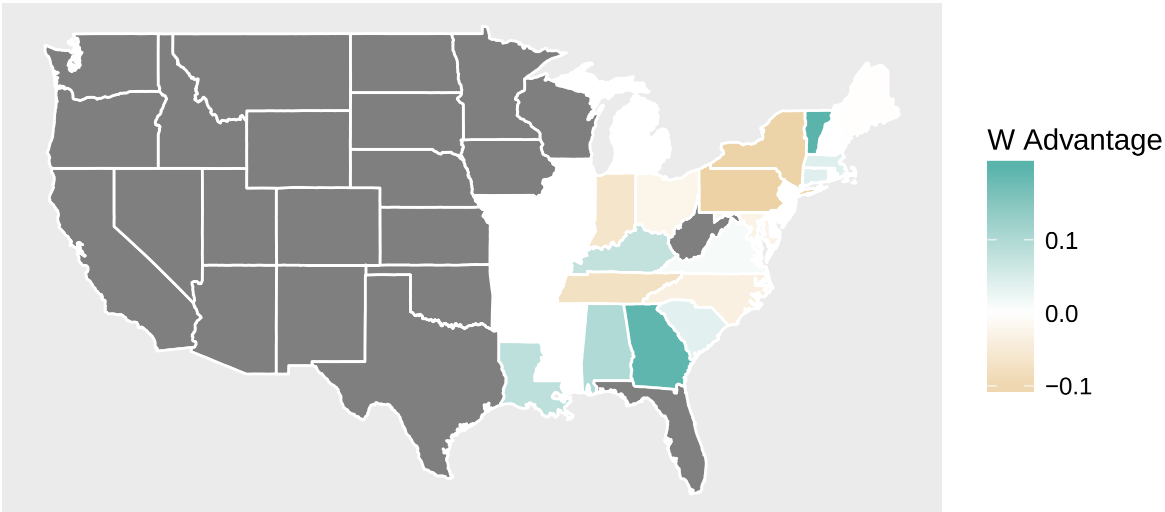


Census #5 U: modified

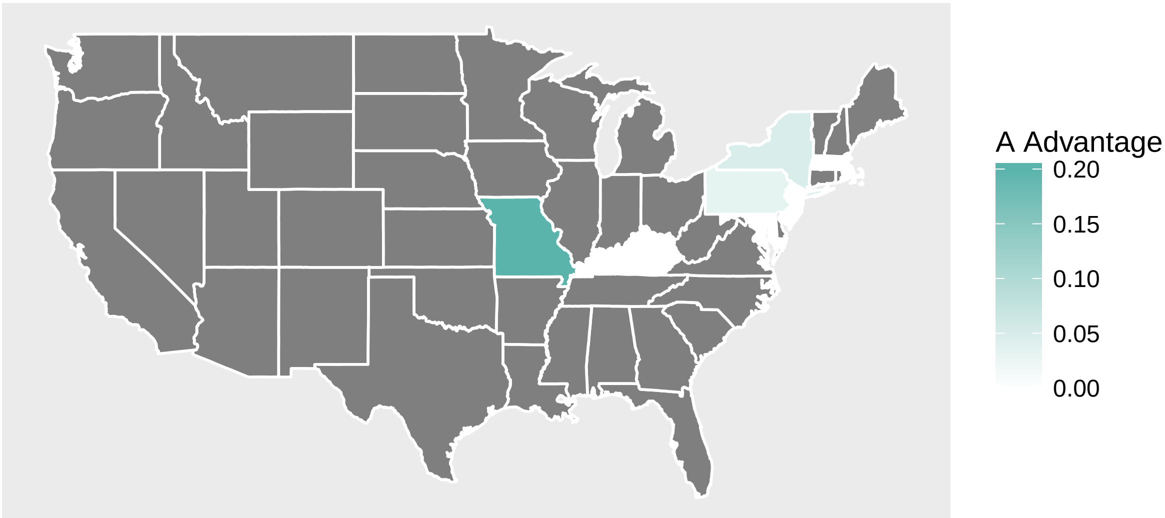
Census #5 W: simple



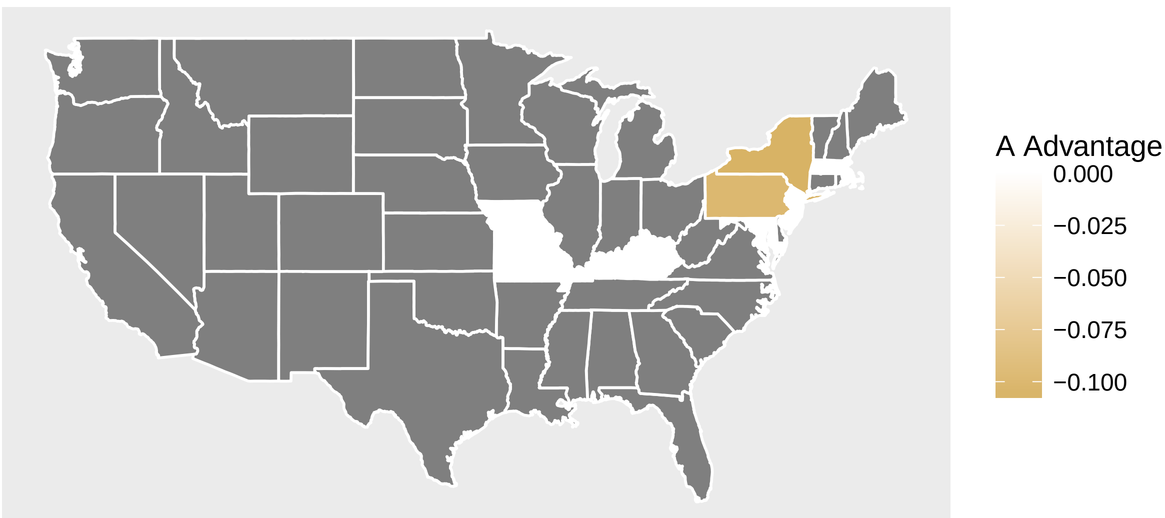
Census #5 W: modified



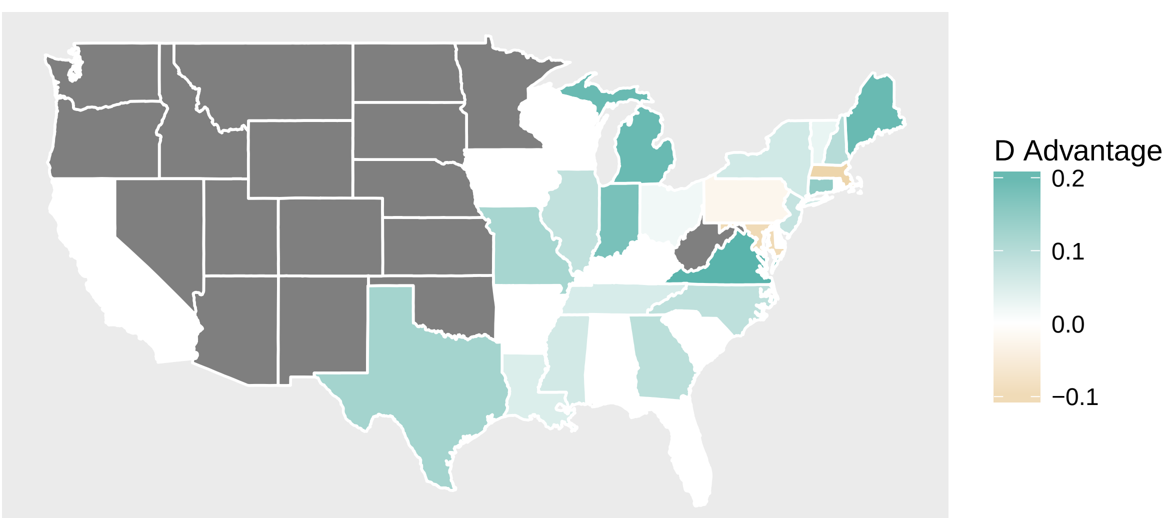
Census #6 A: simple



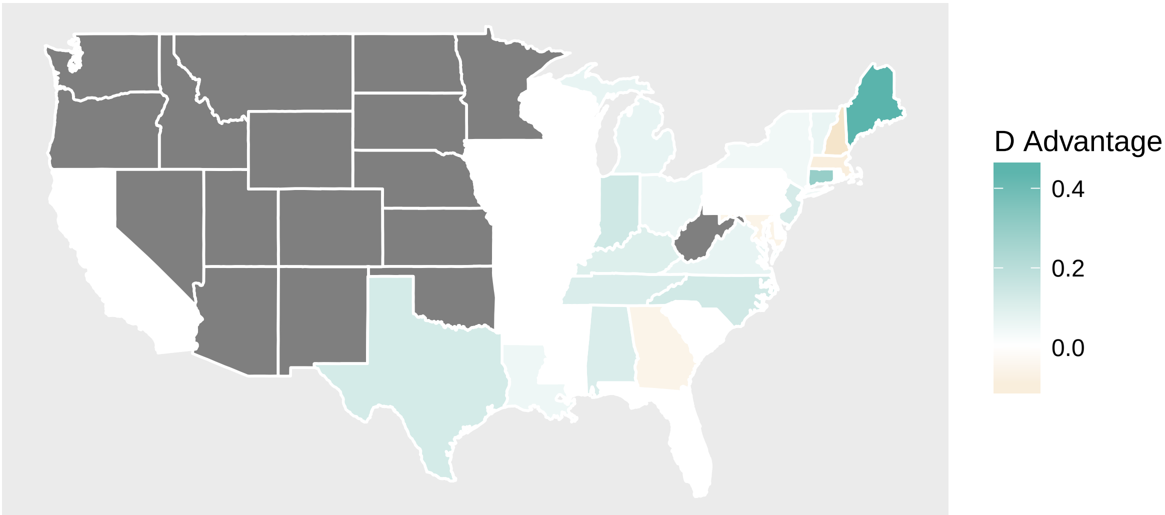
Census #6 A: modified



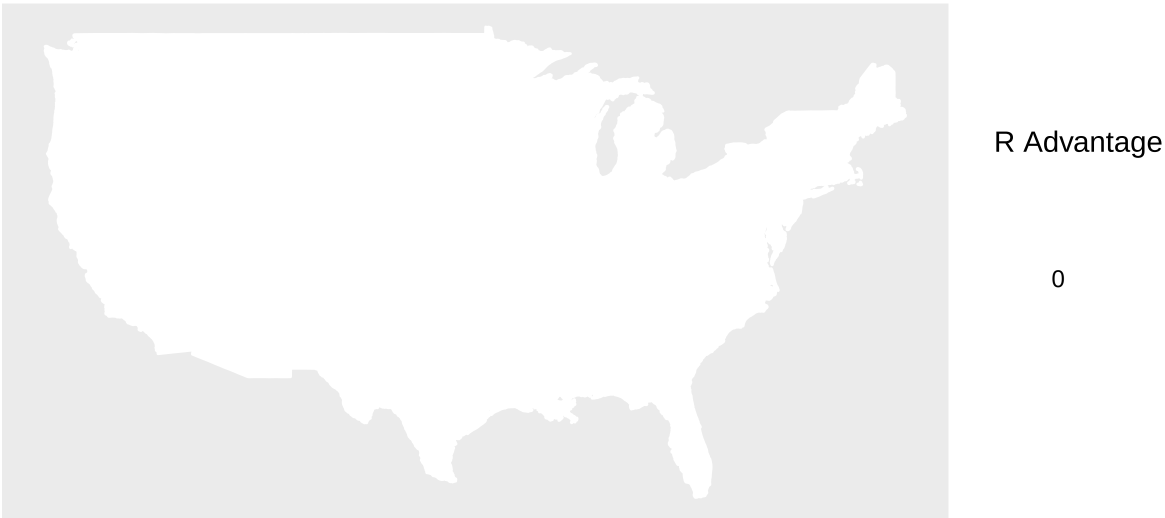
Census #6 D: simple



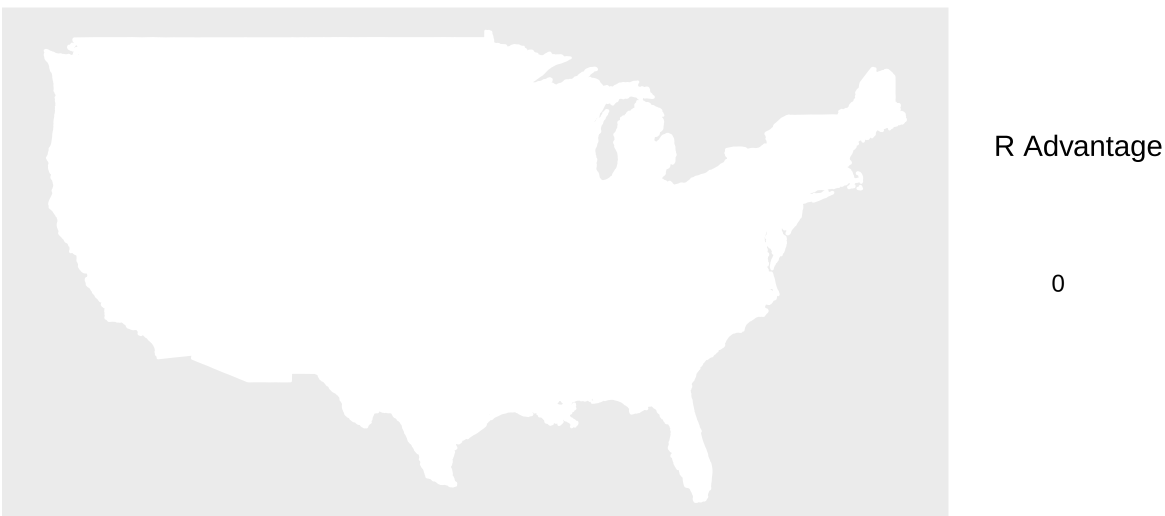
Census #6 D: modified



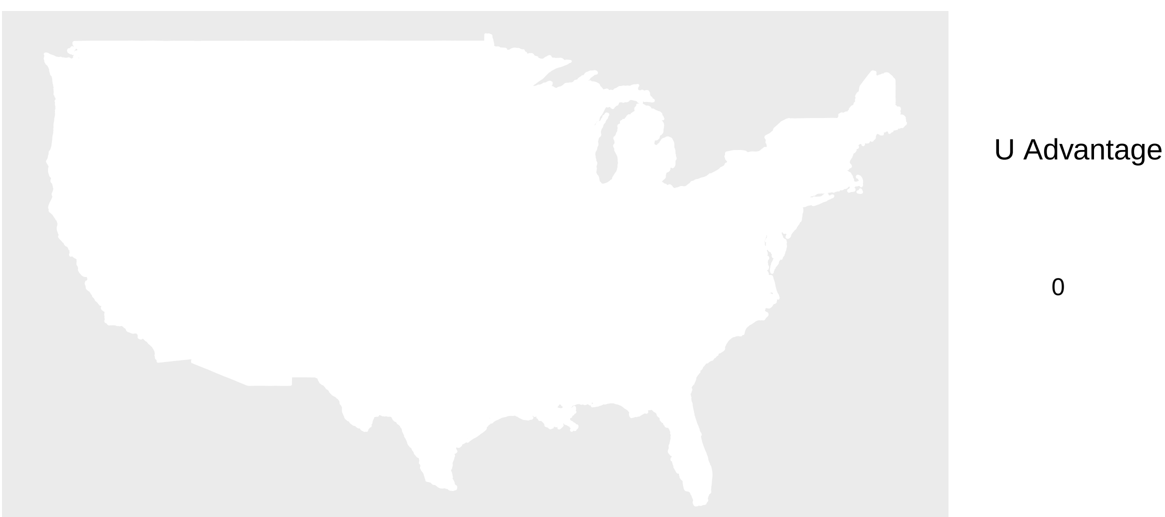
Census #6 R: simple



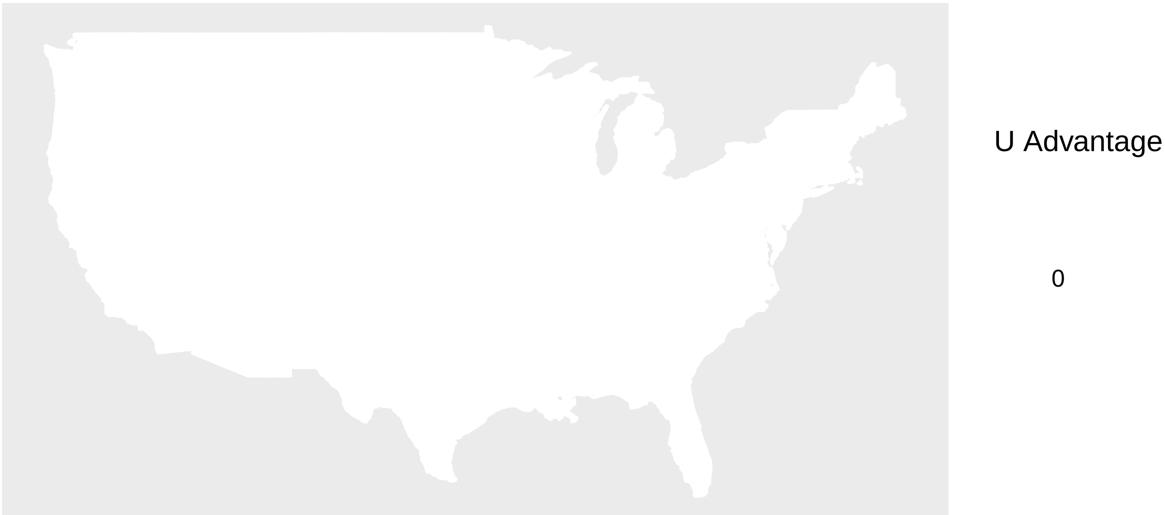
Census #6 R: modified



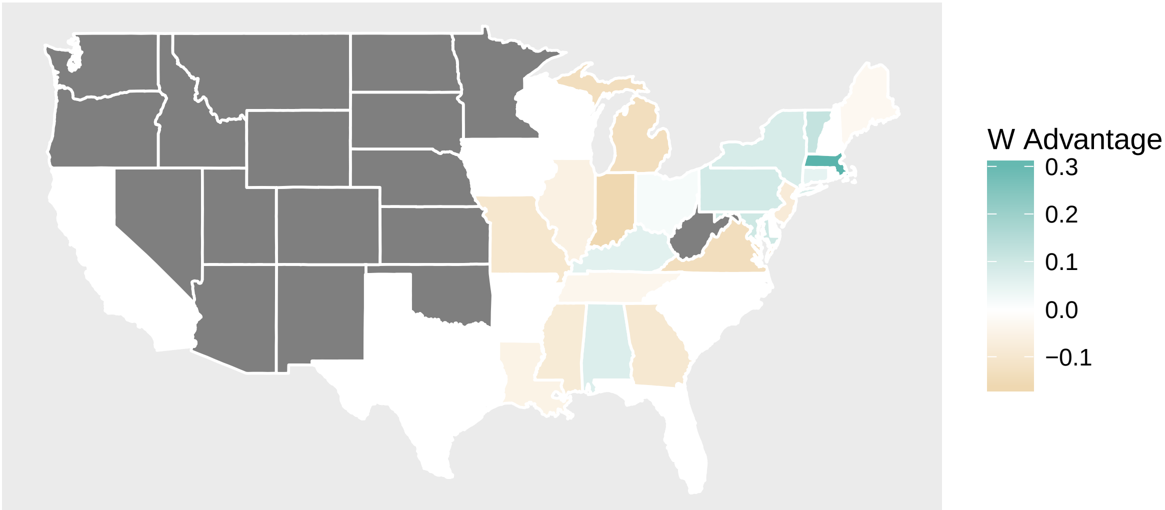
Census #6 U: simple



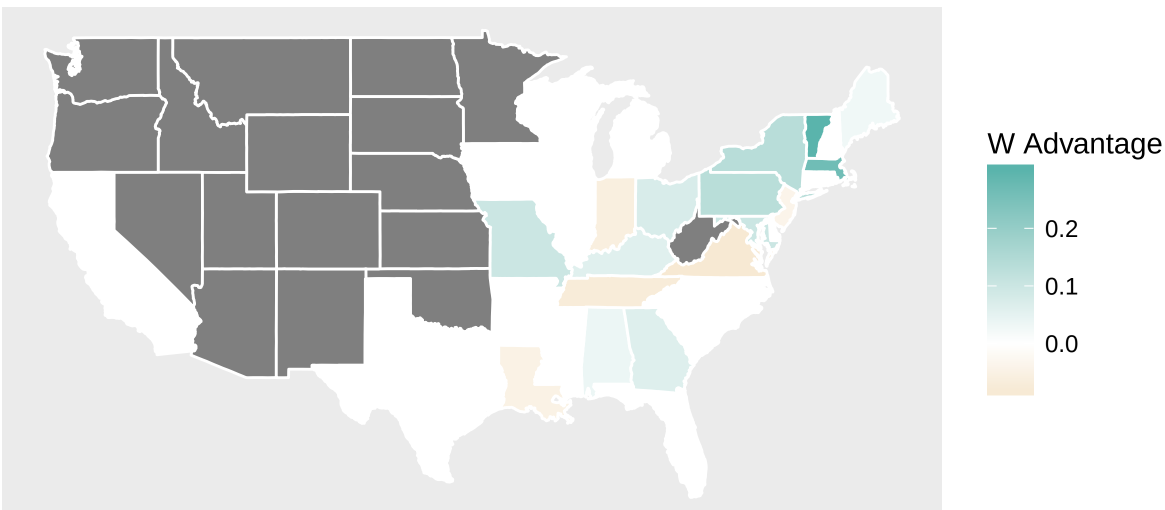
Census #6 U: modified



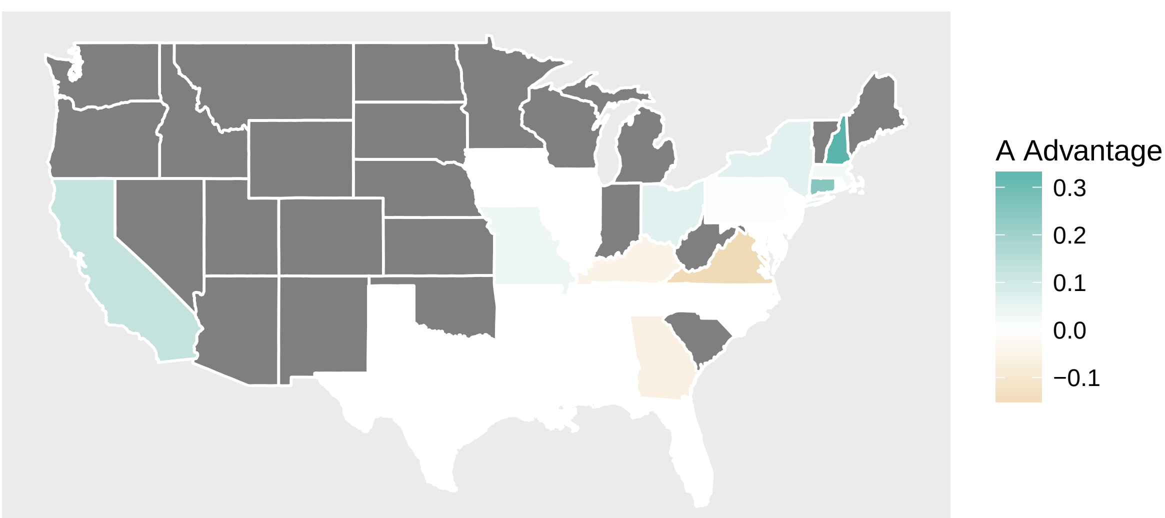
Census #6 W: simple



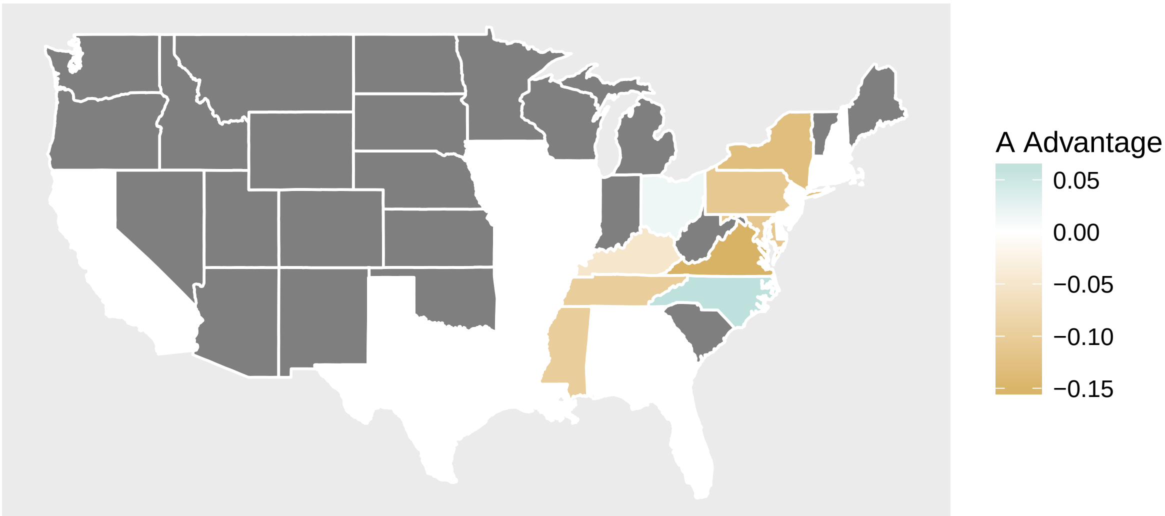
Census #6 W: modified



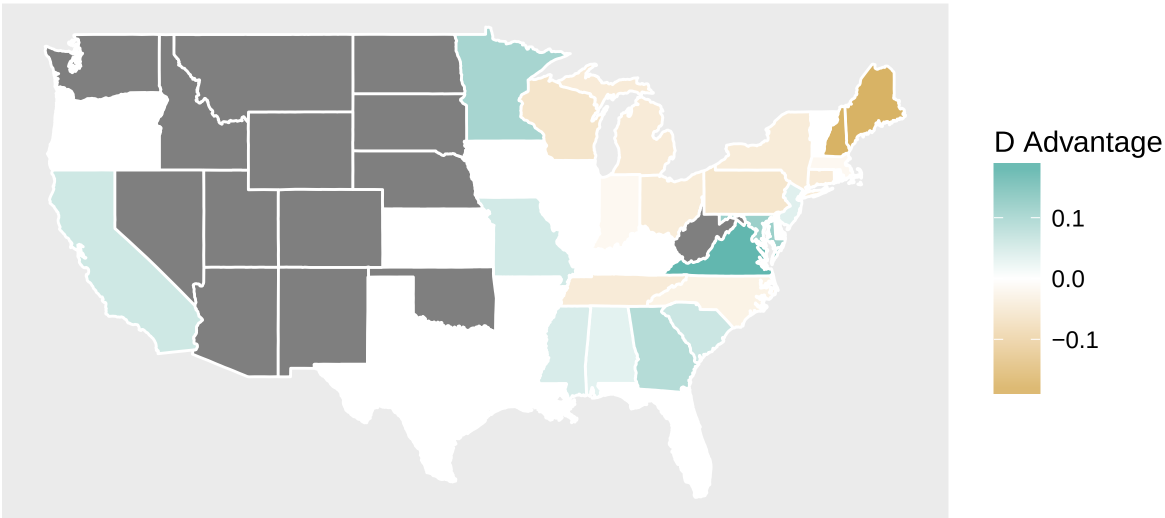
Census #7 A: simple



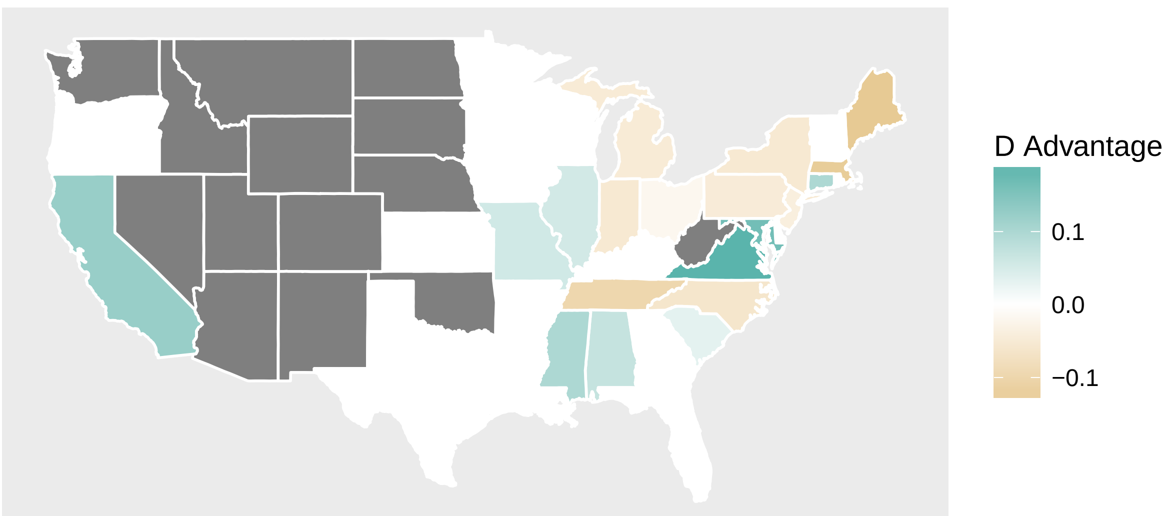
Census #7 A: modified



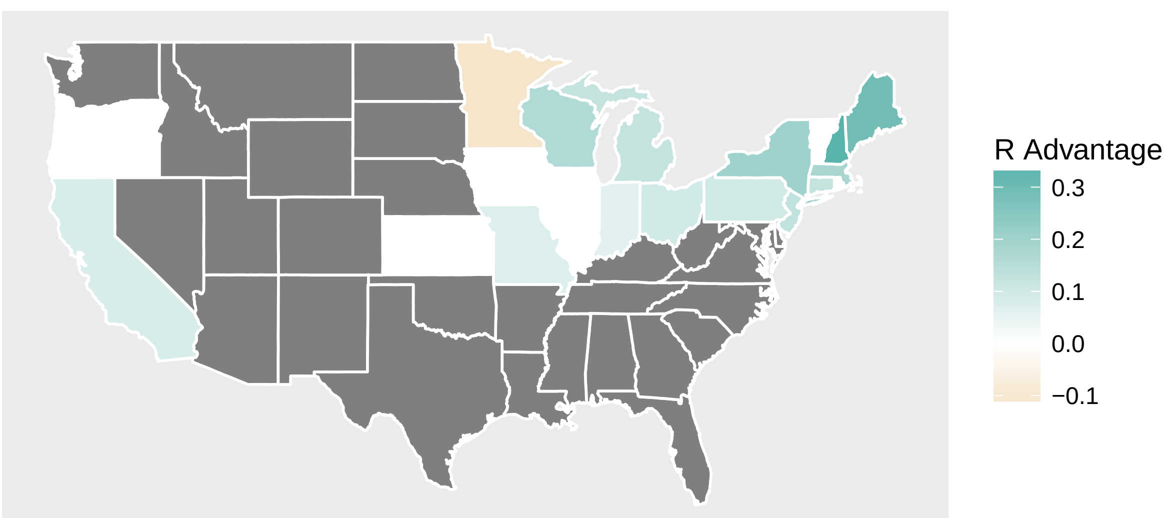
Census #7 D: simple

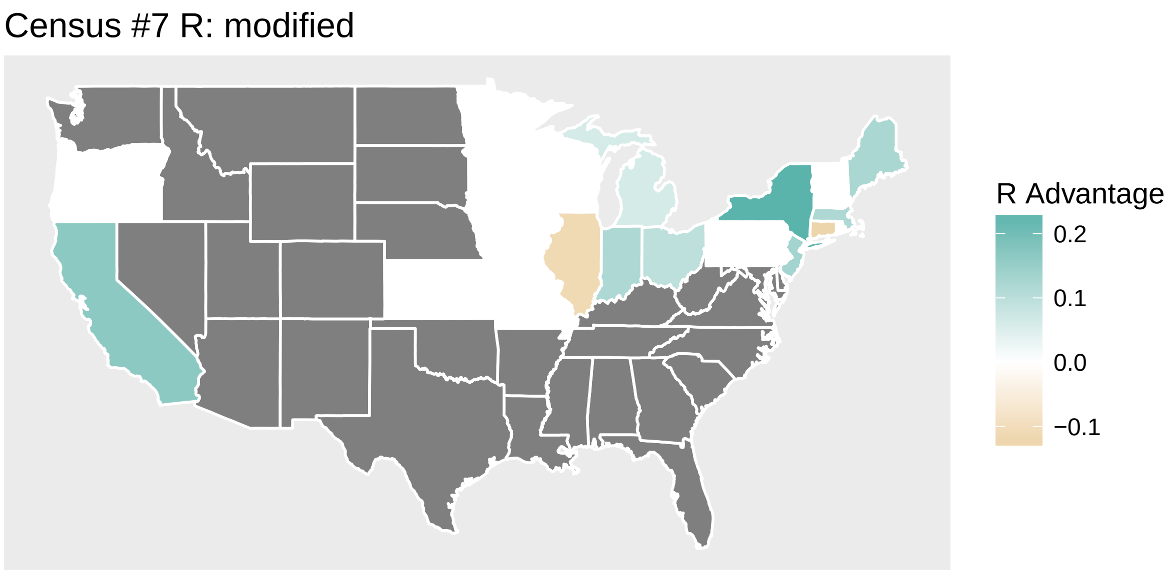


Census #7 D: modified

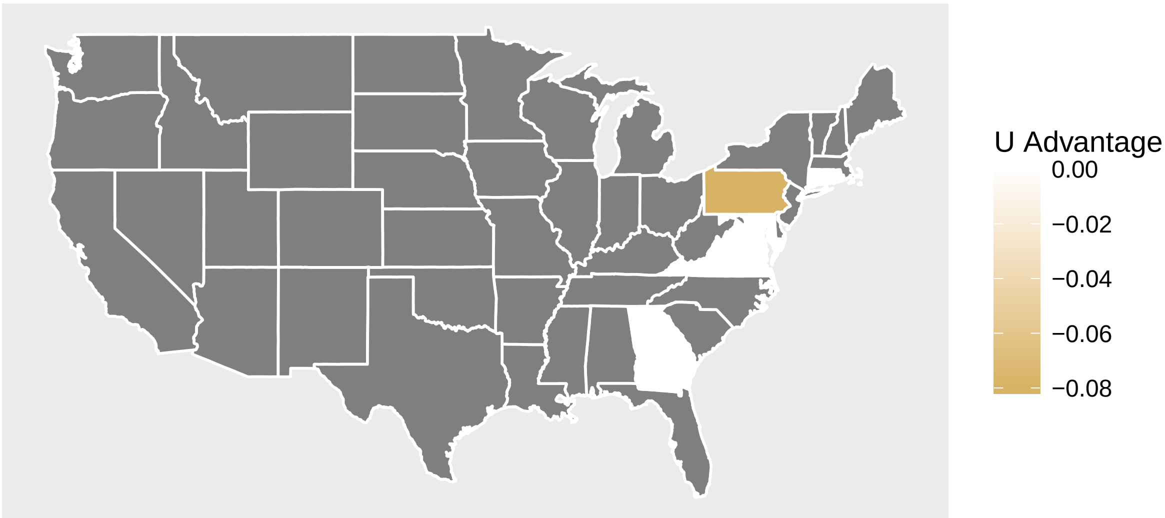


Census #7 R: simple

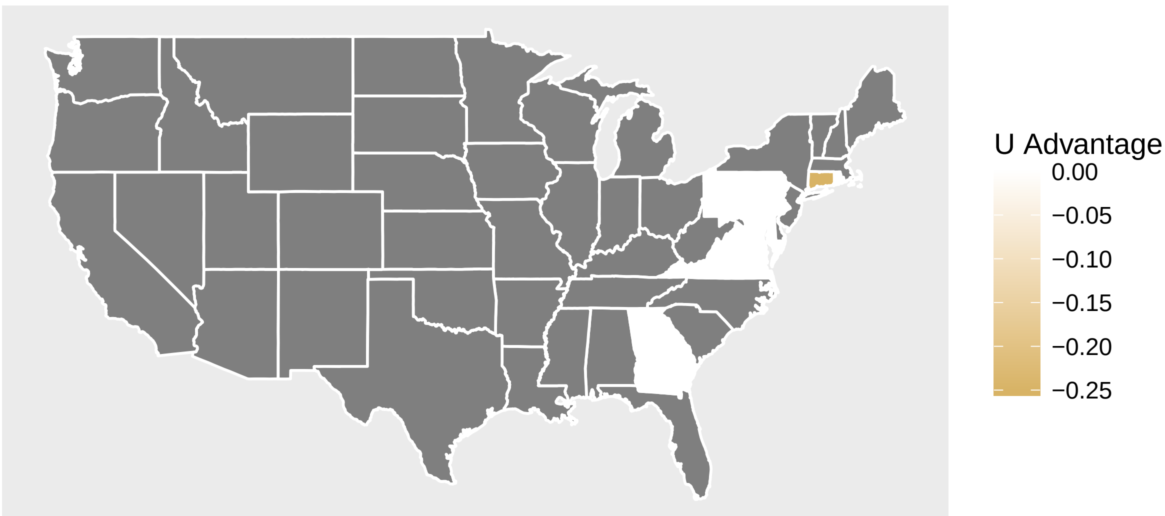



Census #7 R: modified
R Advantage
0.2
0.1
0.0
−0.1

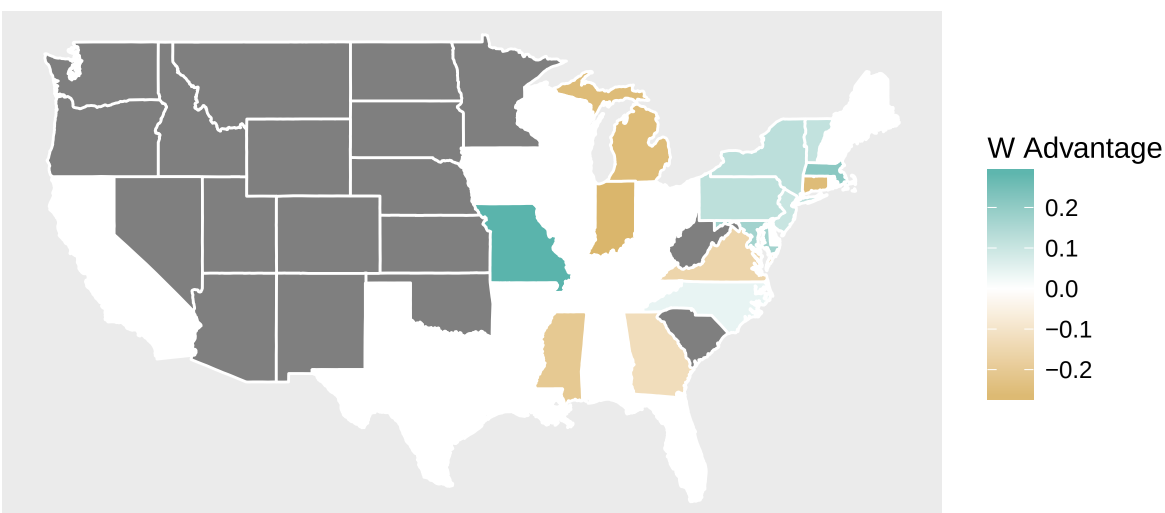
Census #7 U: simple



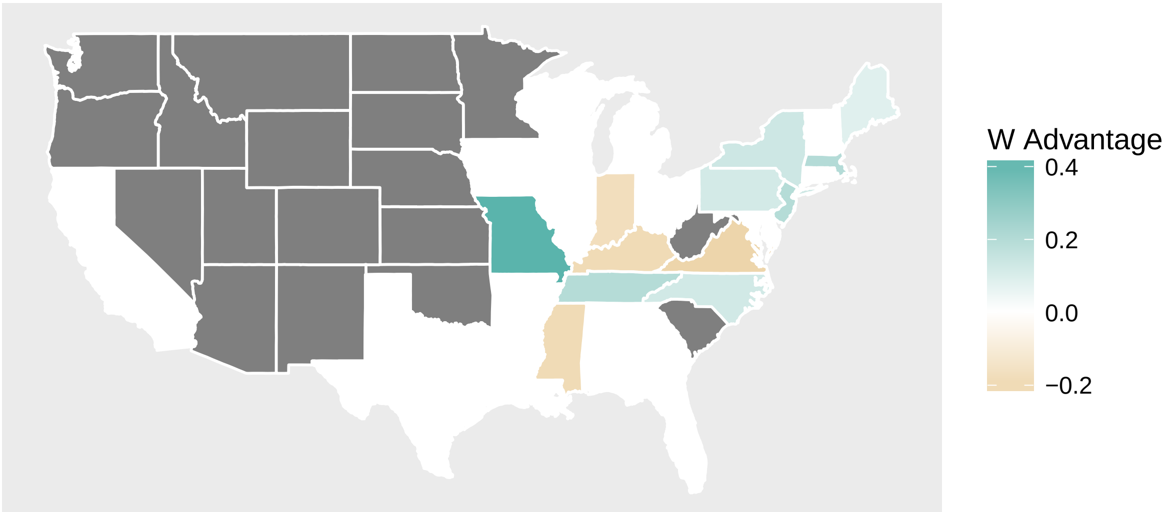
Census #7 U: modified



Census #7 W: simple



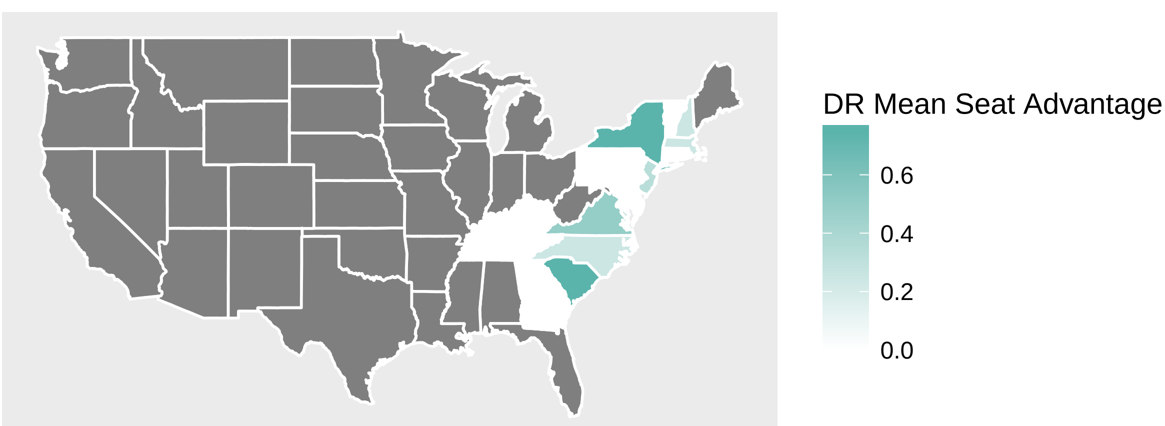
Census #7 W: modified



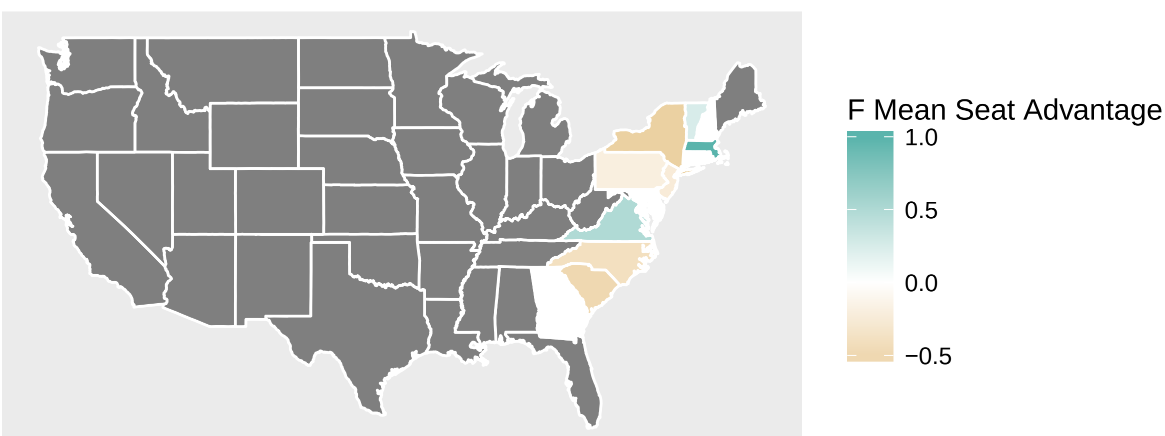
Below are similar maps, but representing the mean number of extra seats the party won as a result of the gap:

```
for (this_census in 1:7) {
  for (this_party in unique(state_profile$party)) {
    states_with_data <- left_join(states,
                                  subset(state_profile, census==this_census & party==this_party),
                                  by = "region")
    if (sum(!is.na(states_with_data$seat_advantage)) > 0) {
      plot.new()
      print(
        ggplot(data = states_with_data) +
          geom_polygon(aes(x = long, y = lat,
                           group = group, fill=seat_advantage),
                       color = "white") +
          coord_fixed(1.3) +
          scale_fill_gradient2(high="#5ab4ac", low="#d8b365") +
          theme(panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                axis.line=element_blank(),
                axis.text.x=element_blank(),
                axis.text.y=element_blank(),
                axis.ticks=element_blank()) +
          xlab("") +
          ylab("") +
          labs(fill= this_party %s+% " Mean Seat Advantage") +
          ggtitle("Census #" %s+% this_census %s+% " " %s+% this_party)
      )
    }
  }
}
```

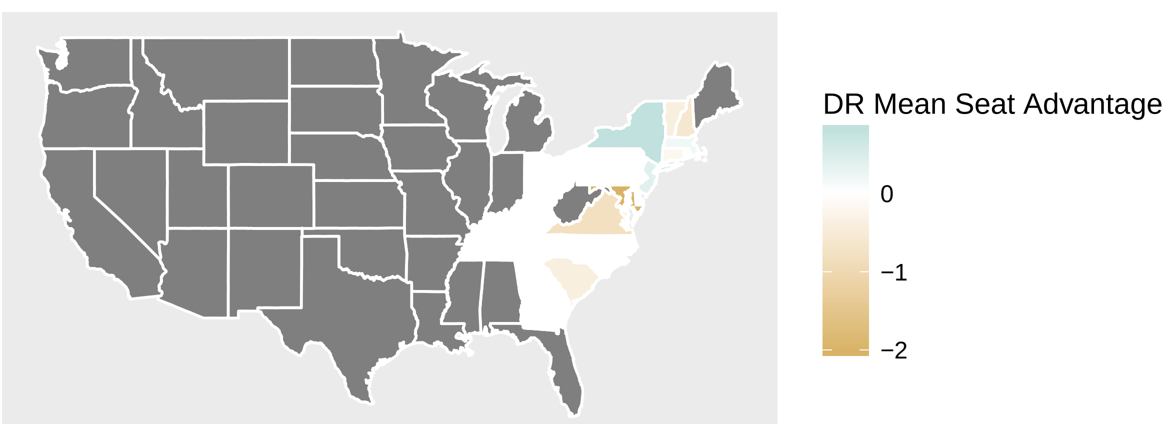
Census #1 DR



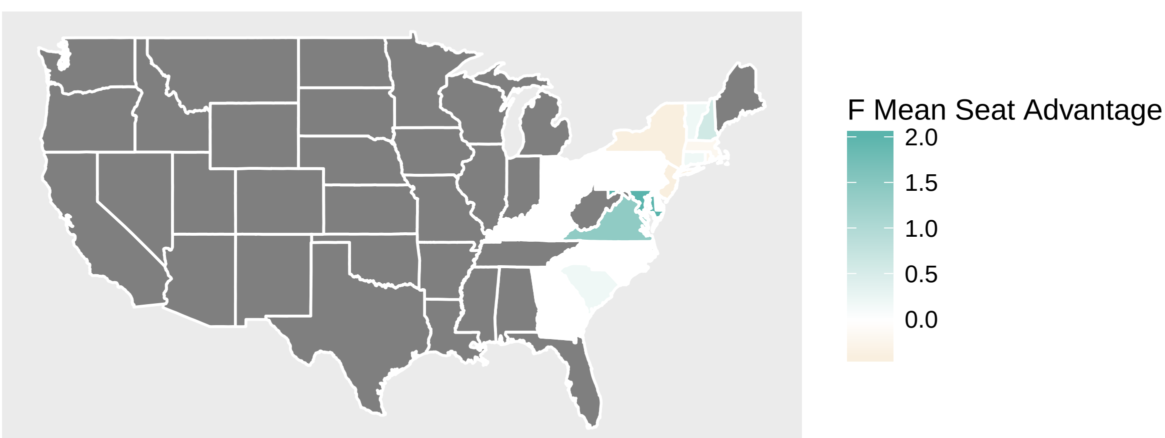
Census #1 F



Census #2 DR



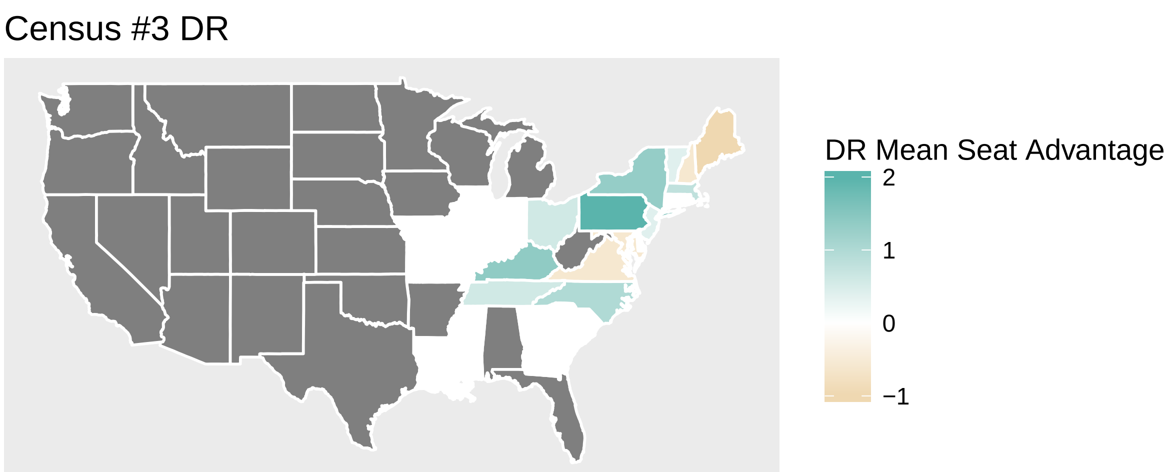
Census #2 F



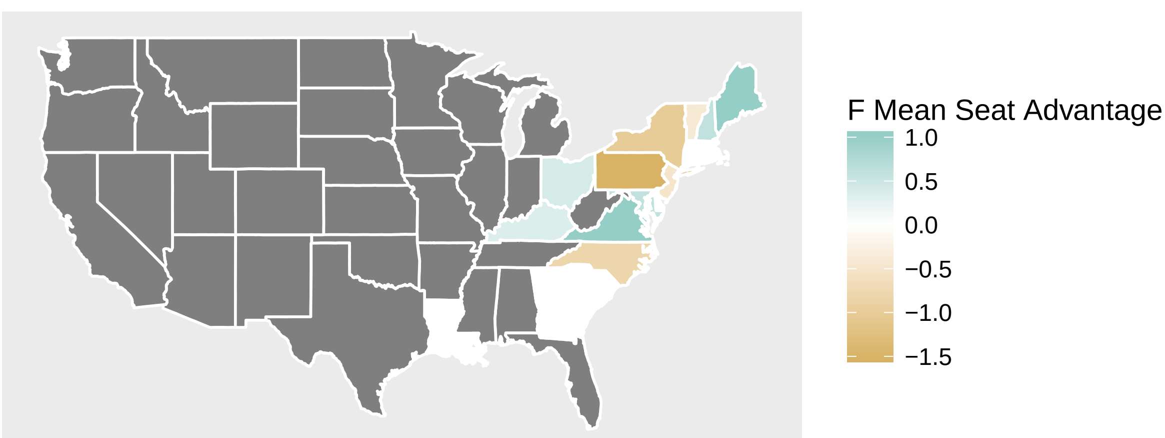
Census #3 A

A Mean Seat Advantage

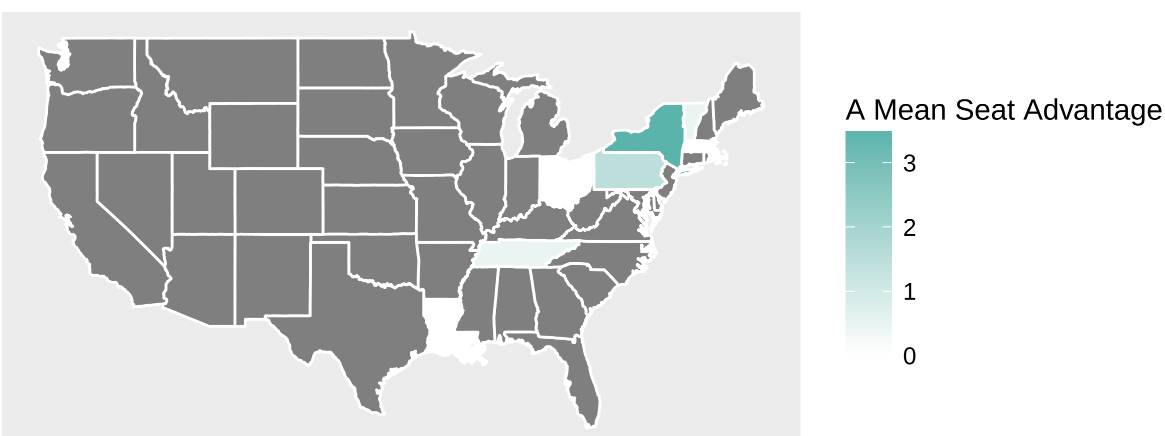
0


Census #3 DR
DR Mean Seat Advantage
2
1
0
−1

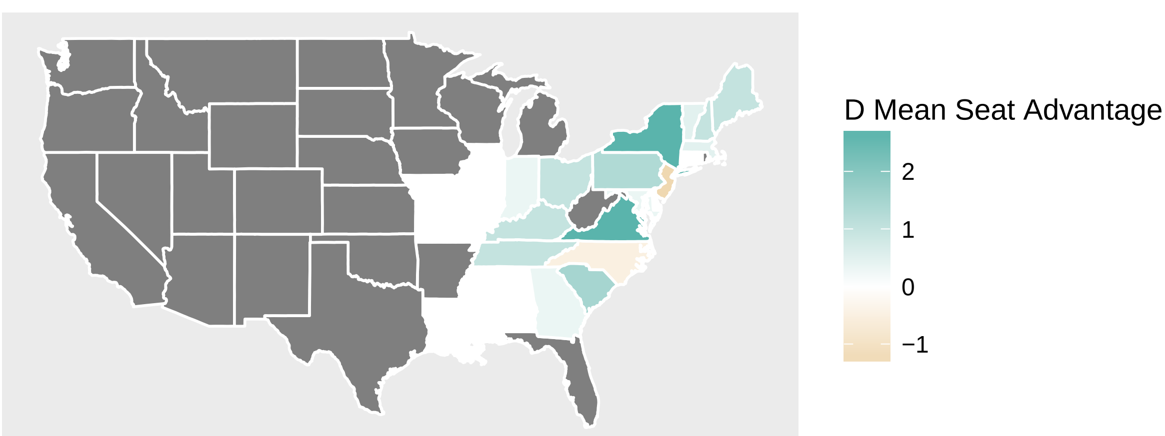
Census #3 F



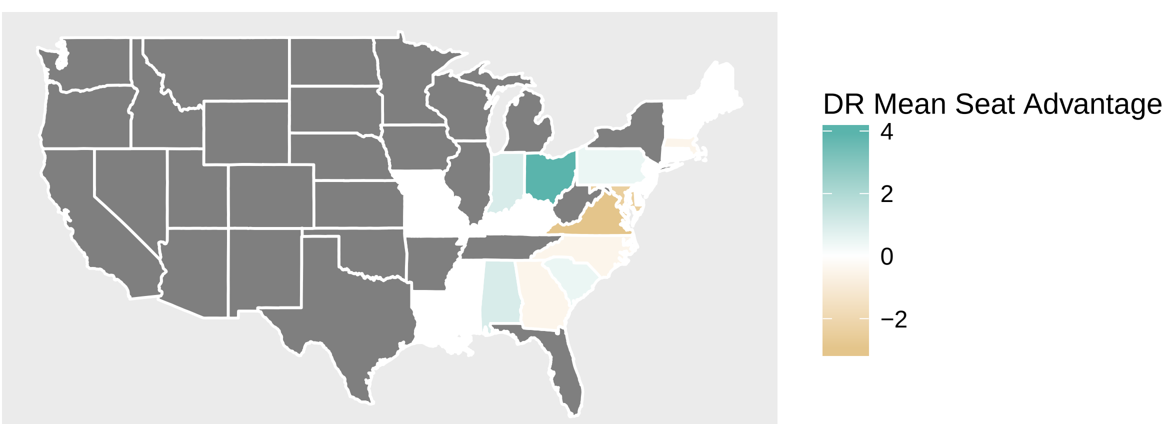
Census #4 A



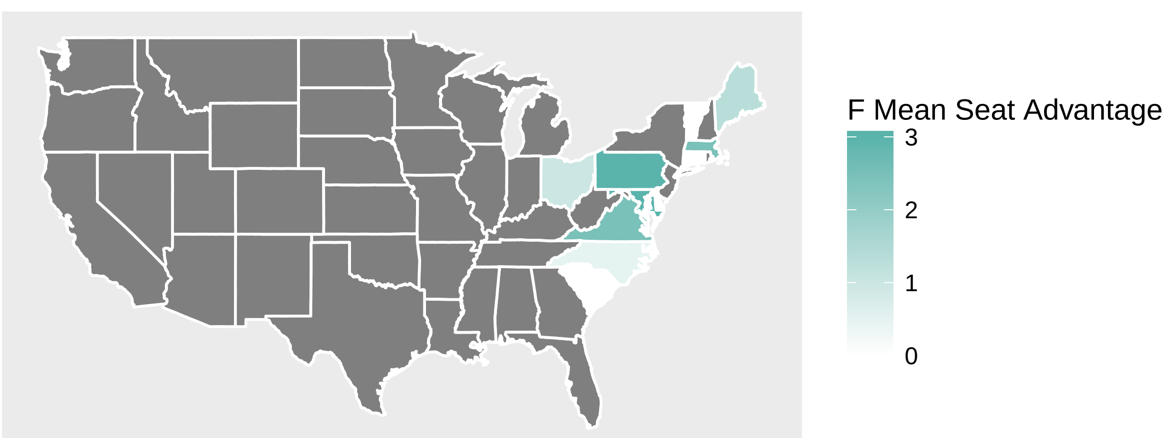
Census #4 D



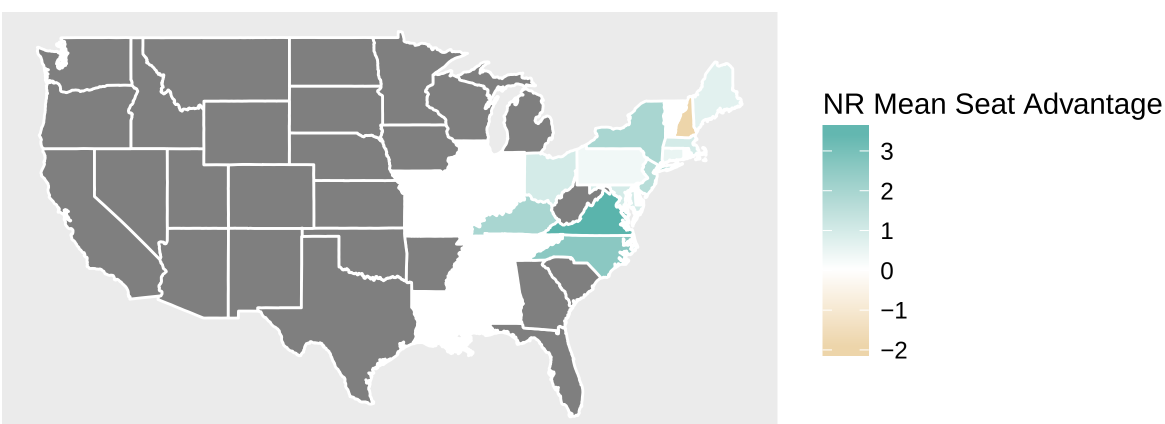
Census #4 DR



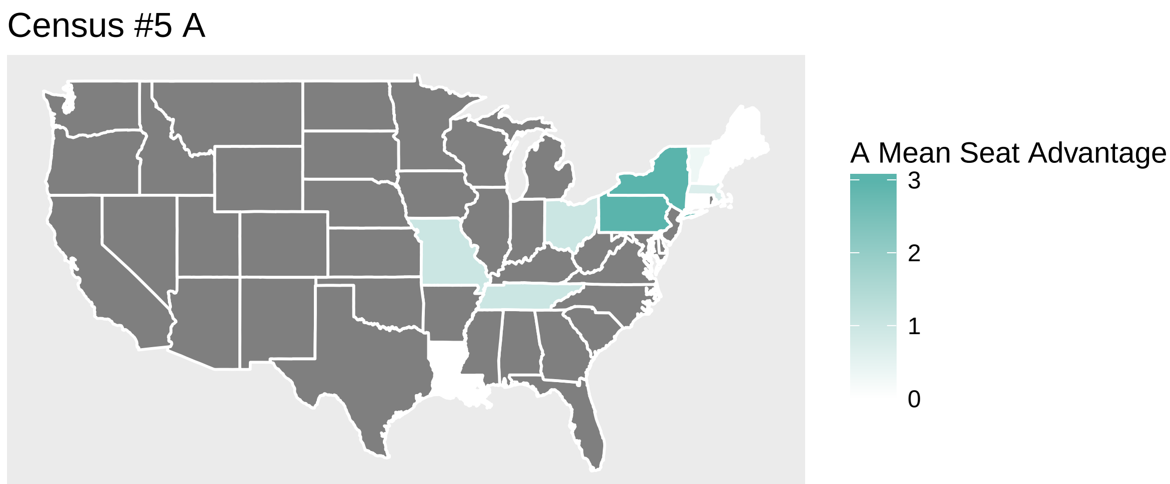
Census #4 F



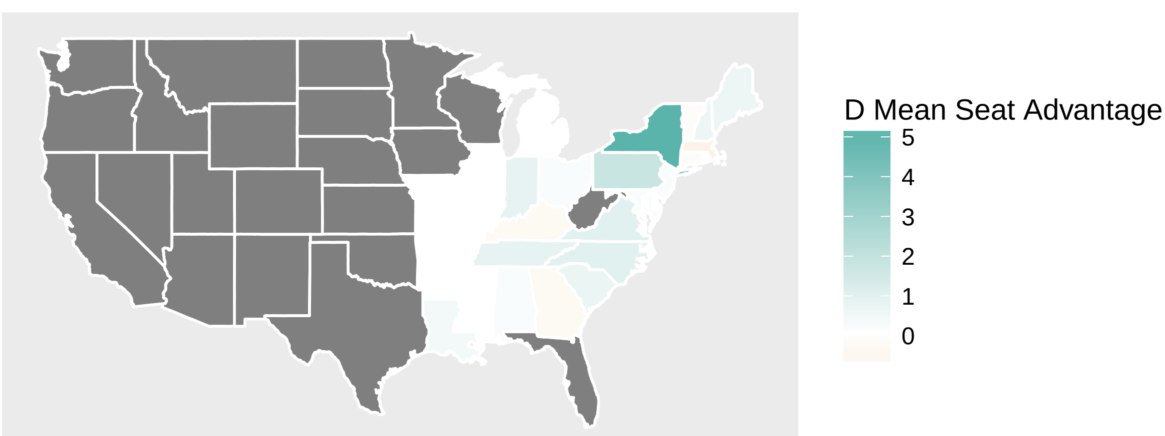
Census #4 NR

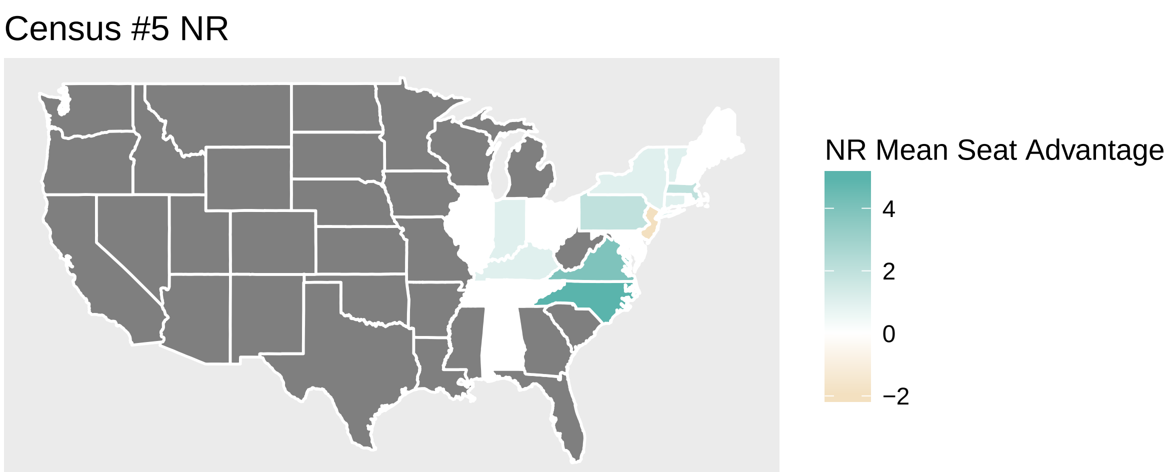


Census #4 R


Census #5 A
A Mean Seat Advantage
3
2
1
0

Census #5 D



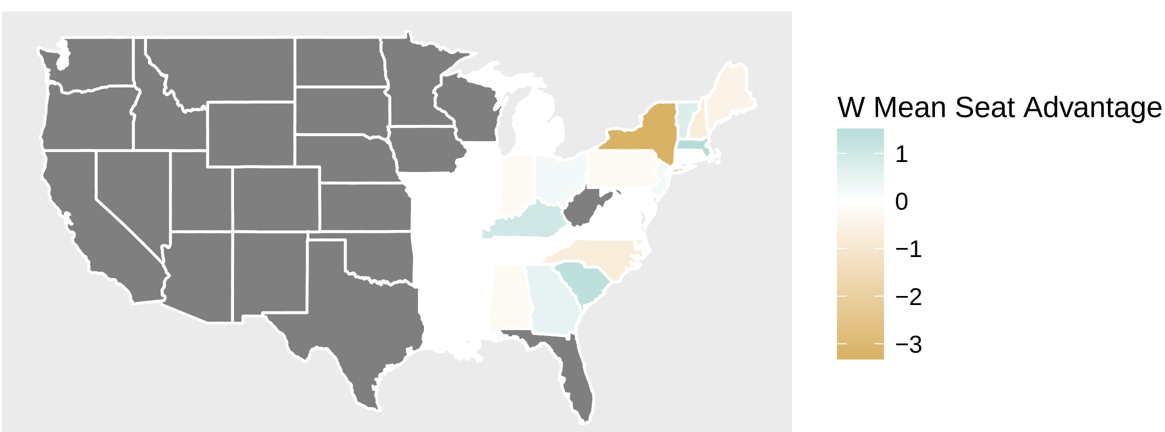

Census #5 NR
NR Mean Seat Advantage
4
2
0
−2

Census #5 U

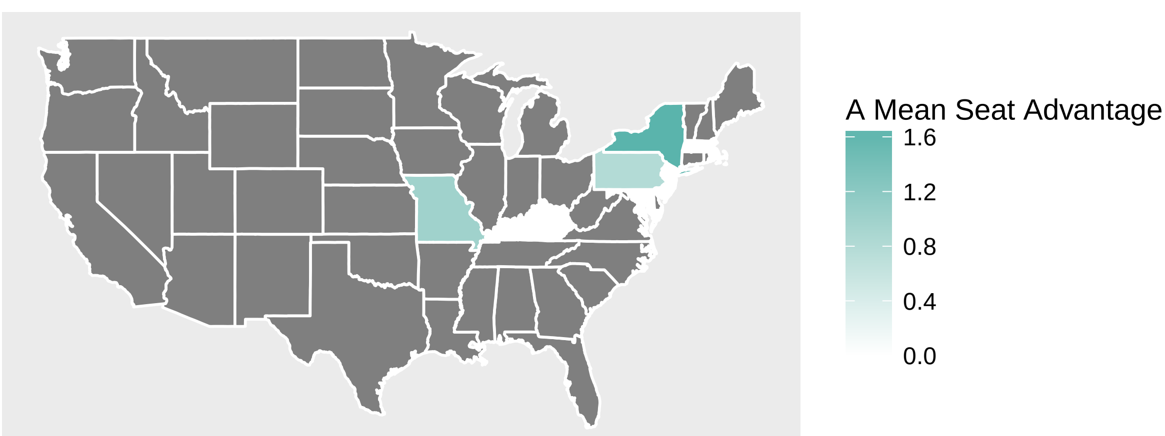
U Mean Seat Advantage

1

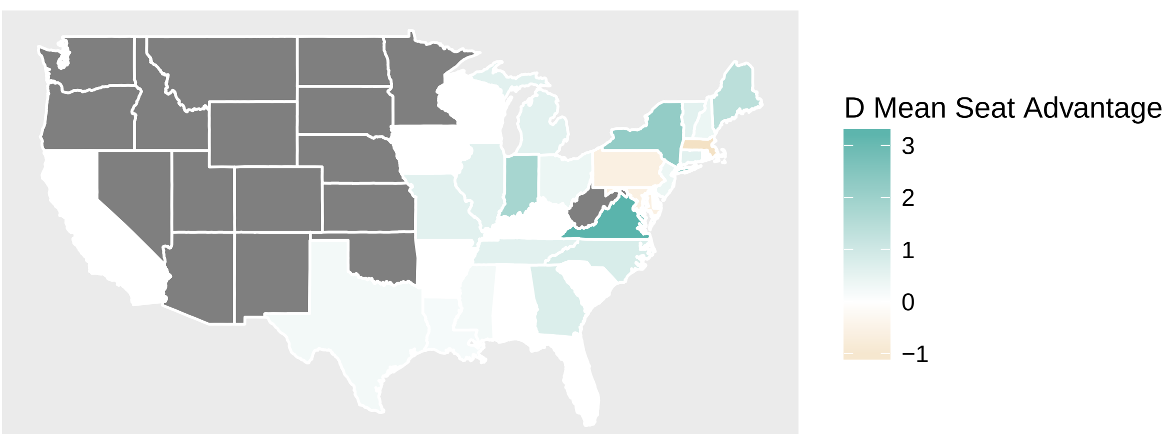
Census #5 W



Census #6 A



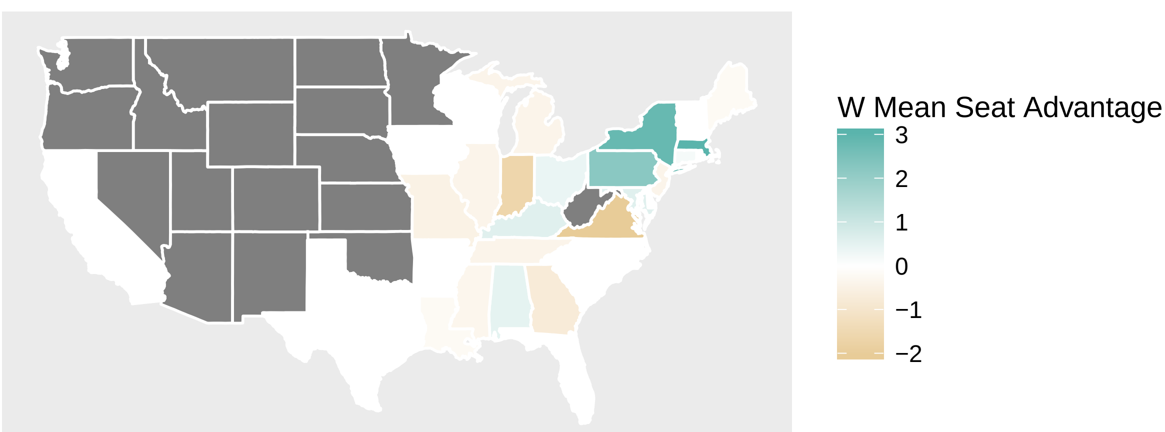
Census #6 D



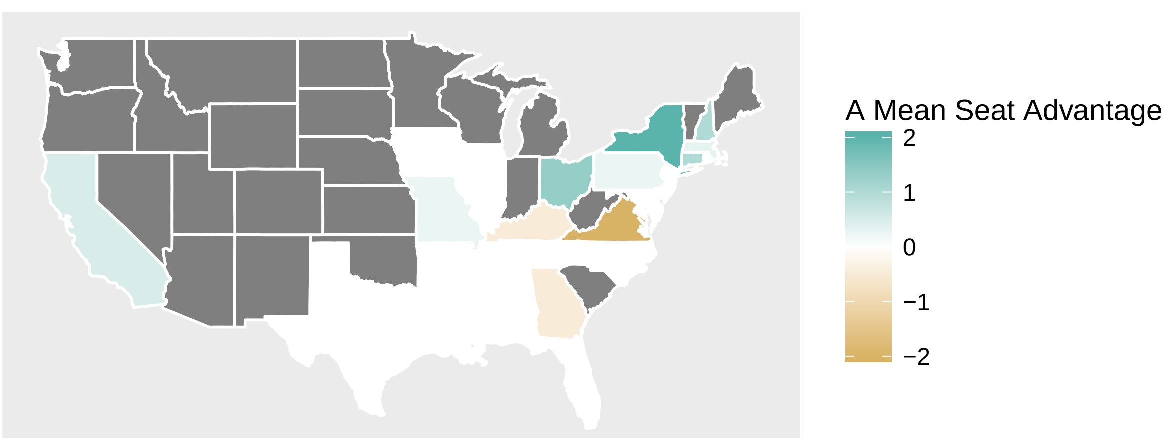
Census #6 R

Census #6 U

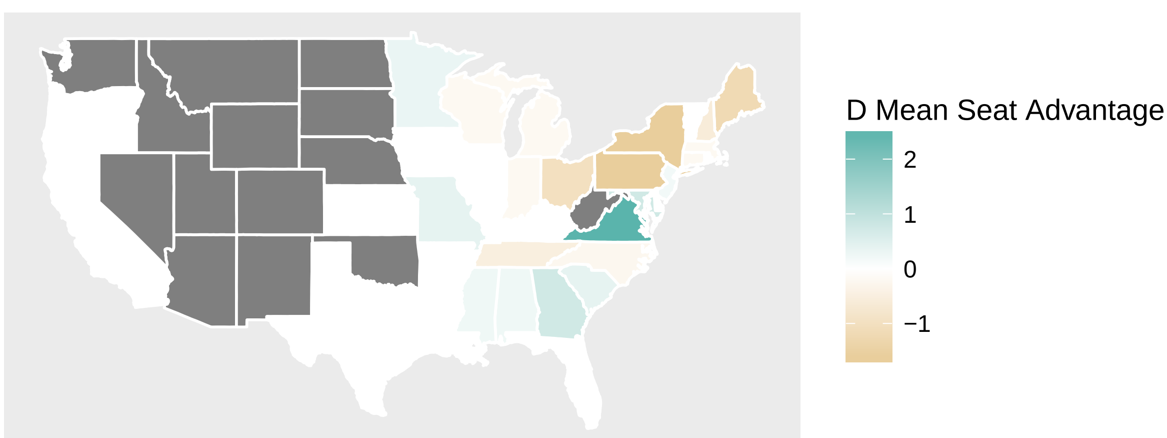
Census #6 W

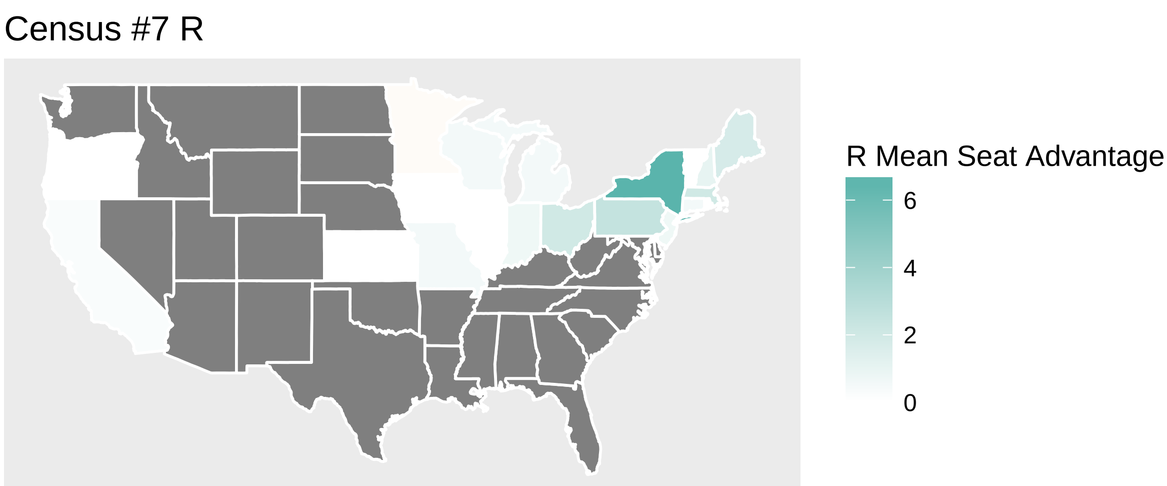


Census #7 A

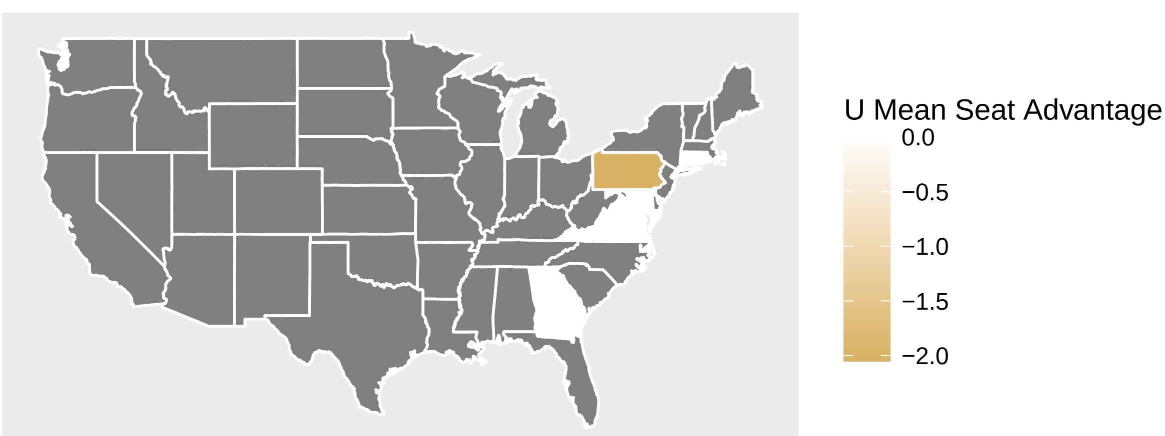


Census #7 D




Census #7 R
R Mean Seat Advantage
6
4
2
0

Census #7 U



Census #7 W