Geographic Representation and Requests for Federal Funds in the U.S. Senate (Main Text Analysis 2)

Jacyln Kaslovsky and Andrew R. Stone

Load the data

```
# Loading the data
load("county_level_request_data.RData")
```

Figure 2

```
# Loading in data from the census to make shapefiles
states_2010 <- states(cb = TRUE, year=2010) %>%
  shift_geometry()
```

```
##   |                                                                      |
```

```
counties_2020 <- counties(cb = TRUE, year=2020) %>%
  shift_geometry()
```

```
##   |                                                                      |
```

```
# A function to remove map axes
ditch_the_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
)

# Load the data to make a map of senator requests it to the county level
tomap <- plyr::ddply(senators_appropriations, plyr::.(GEOID), summarize, sum=sum(appropriation_sum, na.
tomap_final <- left_join(counties_2020,tomap, by="GEOID")
tomap_final$logsum <- log(tomap_final$sum +1)

# Creating the map
tomap_final[tomap_final$STATE_NAME!="Puerto Rico",] %>%
  ggplot() +
  geom_sf(aes(fill=logsum),lwd = 0.3, color="grey74") +
  theme_bw() + ditch_the_axes  +
  labs(fill = "Logged Sum of Appropriations Request") +
```

```
  theme(legend.position="bottom") + geom_sf(data=states_2010[states_2010$NAME!="Puerto Rico",] , fill=NA
  coord_sf(crs = st_crs(2163), xlim = c(-2500000, 2500000), ylim=c(-2300000, 730000))
```

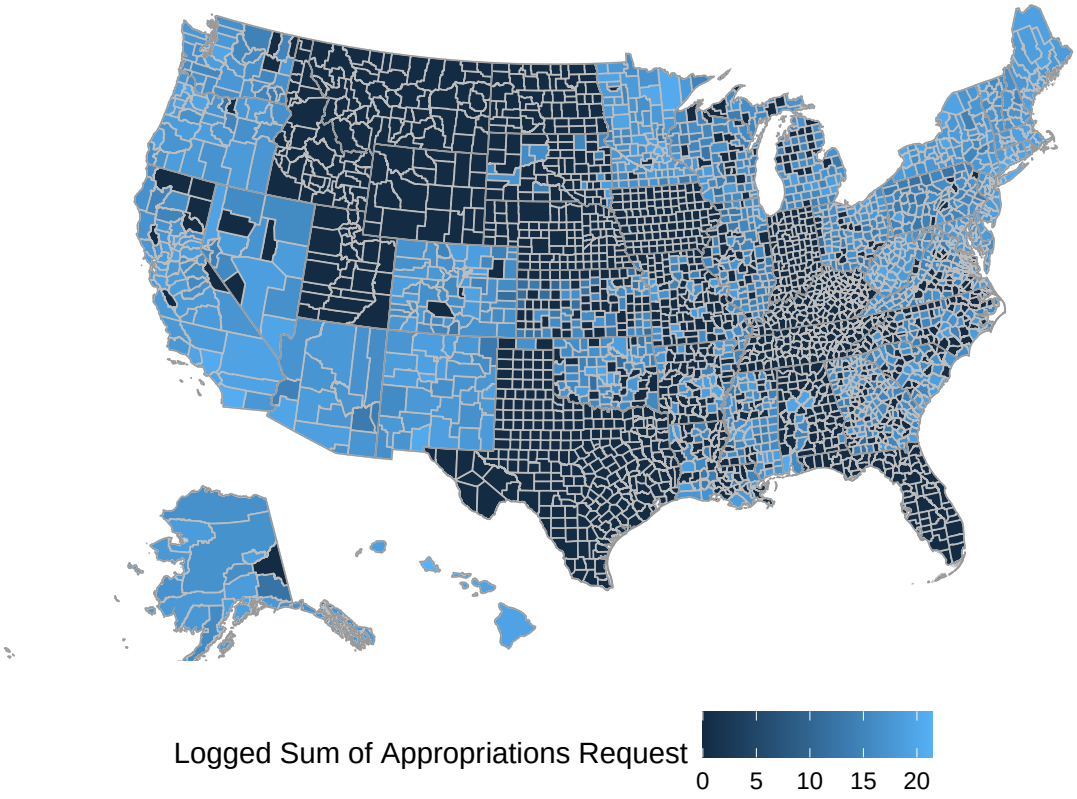


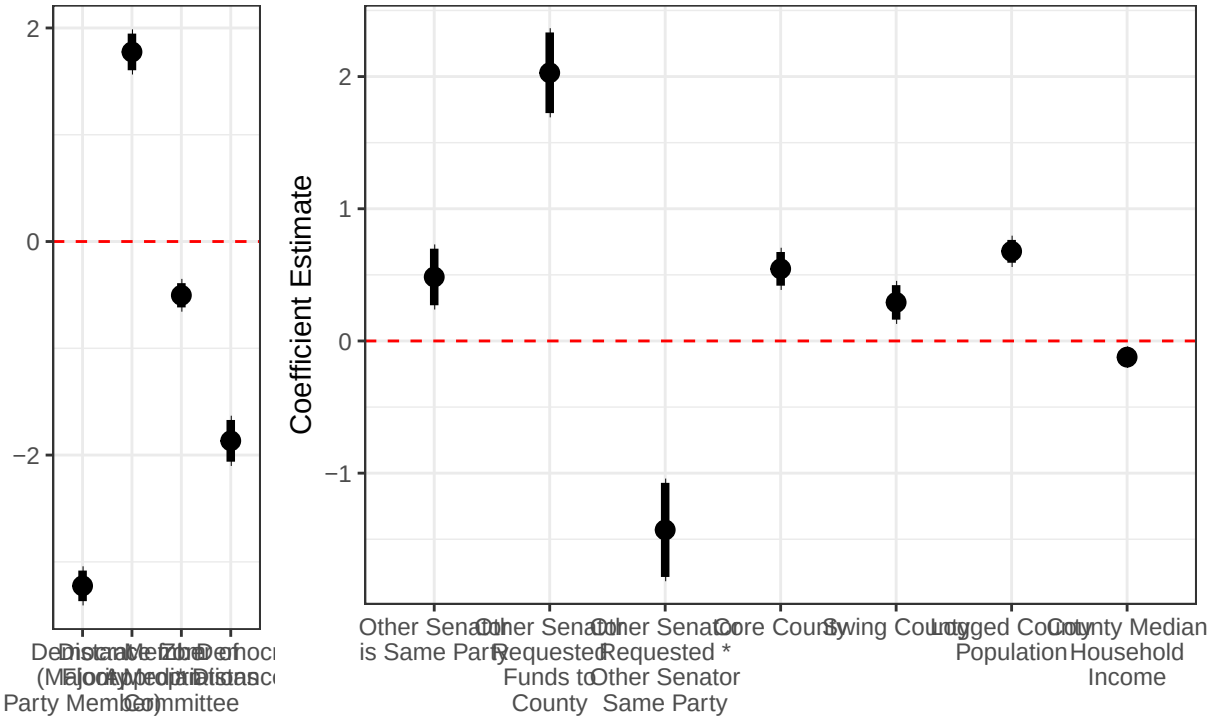
Figure 4 Analysis

```
# Running the zero inflated model
zim.f4 <- glmmTMB(log_sum ~ log_countypop_scaled  + other_sen_requested*other_sen_sameparty + on_appropr
                  core_county + swing_county + seniority_scaled + party_leader + meddist_scaled +
                  freshman + pct_urban_scaled + median_household_income_scaled + capital + factor(year)
                zi = ~  dem + on_appropriations + meddist_scaled + dem*meddist_scaled,
                data = senators_appropriations, family =  gaussian)
# Model summary and standard errors
summary_zim.f4 <- summary(zim.f4)
ses.f4 <- standard_error(zim.f4)

# Making Figure 4
# Vector of coefficients from conditional and zero-inflated stage
coefs_total.f4 <- c(summary_zim.f4$coefficients$cond[,1], summary_zim.f4$coefficients$zi[,1])
# Calculating confidence intervals
ses.f4$coefs <- coefs_total.f4
ses.f4$lb <- ses.f4$coefs - qnorm(.975)*ses.f4$SE
ses.f4$ub <- ses.f4$coefs + qnorm(.975)*ses.f4$SE

# Renaming variables with clearer descriptions for plotting
ses.f4$Parameter[ses.f4$Parameter=="dem"] <- "Democrat (Majority Party Member)"
ses.f4$Parameter[ses.f4$Parameter=="other_sen_requested"] <- "Other Senator Requested Funds to County"
```

```
ses.f4$Parameter[ses.f4$Parameter=="other_sen_sameparty"] <- "Other Senator is Same Party"
ses.f4$Parameter[ses.f4$Parameter=="freshman"] <- "Freshman Senator"
ses.f4$Parameter[ses.f4$Parameter=="female"] <- "Senator is a Woman"
ses.f4$Parameter[ses.f4$Parameter=="party_leader"] <- "Party Leader"
ses.f4$Parameter[ses.f4$Parameter=="meddist_scaled"] <- "Distance from Floor Median"
ses.f4$Parameter[ses.f4$Parameter=="core_county"] <- "Core County"
ses.f4$Parameter[ses.f4$Parameter=="swing_county"] <- "Swing County"
ses.f4$Parameter[ses.f4$Parameter=="log_countypop_scaled"] <- "Logged County Population"
ses.f4$Parameter[ses.f4$Parameter=="seniority_scaled"] <- "Seniority"
ses.f4$Parameter[ses.f4$Parameter=="on_appropriations"] <- "Member of Appropriations Committee"
ses.f4$Parameter[ses.f4$Parameter=="pct_urban_scaled"] <- "County Percent Urban Population"
ses.f4$Parameter[ses.f4$Parameter=="median_household_income_scaled"] <- "County Median Household Income
ses.f4$Parameter[ses.f4$Parameter=="other_sen_requested:other_sen_sameparty"] <- "Other Senator Request
ses.f4$Parameter[ses.f4$Parameter=="capital"] <- "County Includes Capital City"

ses.f4$Parameter[ses.f4$Parameter=="dem"] <- "ZI: Democrat (Majority Party Member)"
ses.f4$Parameter[ses.f4$Parameter=="on_appropriations"] <- "ZI: On Appropriations"
ses.f4$Parameter[ses.f4$Parameter=="meddist_scaled"] <- "ZI: Distance from Floor Median"
ses.f4$Parameter[ses.f4$Parameter=="dem:meddist_scaled"] <- "ZI: Democrat * Distance"

# Extracting coefficients to plot for zero-inflated and conditional stages
zi.f4 <- ses.f4[20:23,]
results.f4 <- ses.f4[-c(1,5:7,10:14,16:17,19:23),]
# Factorizing variable names to allow us to plot coefficients in desired order
results.f4$Parameter <- as.factor(results.f4$Parameter)
results.f4 <- results.f4 %>%
  mutate(Parameter = fct_relevel(Parameter,
                                 "Other Senator is Same Party","Other Senator Requested Funds to County
                                 "Other Senator Requested * Other Senator Same Party",
                                 "Core County", "Swing County", "Logged County Population"))

# Conditional stage plot
g1_full_f4 <- ggplot(results.f4, aes(x = Parameter, y = coefs)) +
  geom_point(size=3) +
  geom_errorbar(aes(ymin = lb, ymax = ub), linewidth=1.5, width = 0) +
  geom_hline(yintercept = 0, lty = 2, color="red") +
  xlab("") + theme_bw() +
  ylab("Coefficient Estimate") +
  labs(title="")  +
  theme(plot.title = element_text(hjust = 0.5)) + theme(legend.title=element_blank()) + theme(legend.pos
  scale_x_discrete(labels = wrap_format(14))

# Zero-inflated stage plot
g2_zi_f4 <- ggplot(zi.f4, aes(x = Parameter, y = coefs)) +
  geom_point(size=3) +
  geom_errorbar(aes(ymin = lb, ymax = ub), linewidth=1.5, width = 0) +
  geom_hline(yintercept = 0, lty = 2, color="red") +
  xlab("") + theme_bw() +
  ylab("") +
  labs(title="")  +
  theme(plot.title = element_text(hjust = 0.5)) + theme(legend.title=element_blank()) + theme(legend.pos
  scale_x_discrete(labels = wrap_format(14))
```

Figure 4 Output

```
full_v2_f4 <- grid.arrange(g2_zi_f4, g1_full_f4,ncol=2,widths=c(1,3), top="")
```



In-text discussion of results from the county-level requests model

```
### Interpreting the magnitude of the result on core and swing variables (log-level interpretation) ###
# summary_zim.f4
# One unit increase in core_county is associated with 72.6% increase in requested funds
round((exp(0.546)-1)*100,1)
```

```
## [1] 72.6
```

```
# One unit increase in swing_county is associated with 33.9% increase in requested funds
round((exp(0.292)-1)*100,1)
```

```
## [1] 33.9
```

```
### Predicting logged sum of appropriations for each observation, then exp() to unlog ###
senators_appropriations$prediction <- exp(predict(zim.f4,newdata=senators_appropriations, type="response

# Average predicted value for each of core/swing/opposition, other senator behavior, other senator party
summarized_predictions <- senators_appropriations %>%
  group_by(other_sen_requested, other_sen_sameparty,
           opposition_county, core_county, swing_county) %>%
  summarise(prediction=mean(prediction,na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'other_sen_requested',
## 'other_sen_sameparty', 'opposition_county', 'core_county'. You can override
```

```
## using the `.groups` argument.
```

```
summarized_predictions <- summarized_predictions[!is.na(summarized_predictions$opposition_county),]

# For states with split delegations, senators predicted to request most for swing that also received re
summarized_predictions[which(summarized_predictions$other_sen_sameparty == 0),c(1,3:6)]
```

```
## # A tibble: 6 x 5
## # Groups:   other_sen_requested, opposition_county, core_county [6]
##   other_sen_requested opposition_county core_county swing_county prediction
##                 <dbl>             <dbl>       <dbl>        <dbl>      <dbl>
## 1                   0                 0           0            1     21305.
## 2                   0                 0           1            0      9635.
## 3                   0                 1           0            0      8303.
## 4                   1                 0           0            1    122535.
## 5                   1                 0           1            0     62247.
## 6                   1                 1           0            0     55093.
```

```
# For states with two copartisans, senators predicted to request most for core that also received reque
summarized_predictions[which(summarized_predictions$other_sen_sameparty == 1),c(1,3:6)]
```

```
## # A tibble: 6 x 5
## # Groups:   other_sen_requested, opposition_county, core_county [6]
##   other_sen_requested opposition_county core_county swing_county prediction
##                 <dbl>             <dbl>       <dbl>        <dbl>      <dbl>
## 1                   0                 0           0            1      2953.
## 2                   0                 0           1            0       306.
## 3                   0                 1           0            0      5694.
## 4                   1                 0           0            1     24707.
## 5                   1                 0           1            0     24814.
## 6                   1                 1           0            0     13047.
```

```
### Percent of counties in a state members target ###
# For each senator in each year, how many counties do they have (num_counties), did they make requests
multiple_counties <- senators_appropriations %>%
  group_by(senator, state, year, county) %>%
  mutate(made_request=as.numeric(appropriation_sum >0),
         county_indicator=1) %>%
  group_by(senator, state, year) %>%
  summarize(num_counties = sum(county_indicator),
            num_requests = sum(made_request),
            pct_requests = num_requests/num_counties)
```

```
## `summarise()` has grouped output by 'senator', 'state'. You can override using
## the `.groups` argument.
```

```
# 27 senators across the two fiscal years make requests to all counties
length(which(multiple_counties$pct_requests == 1))
```

```
## [1] 27
```

```
# Mean number of counties requests made to among requesters
round(mean(multiple_counties$pct_requests[which(multiple_counties$num_requests > 0)]),3)
```

```
## [1] 0.682
```