Politics Are Afoot!

Da Qi Ren

The Setup

There is *a lot* of money that is spent in politics in Presidential election years. So far, estimates have the number at about $11,000,000,000 (11 billion USD). For context, in 2019 Twitter's annual revenue was about $3,500,000,000 (3.5 billion USD).

The work

Install the package, `fec16`.

```
## install.packages('fec16')
```

This package is a compendium of spending and results from the 2016 election cycle. In this dataset are 9 different datasets that cover:

- `candidates`: candidate attributes, like their name, a unique id of the candidate, the election year under consideration, the office they're running for, etc.
- `results_house`: race attributes, like the name of the candidates running in the election, a unique id of the candidate, the number of `general_votes` garnered by each candidate, and other information.
- `campaigns`: financial information for each house & senate campaign. This includes a unique candidate id, the total receipts (how much came in the doors), and total disbursements (the total spent by the campaign), the total contributed by party central committees, and other information.

Your task

Describe the relationship between spending on a candidate's behalf and the votes they receive.

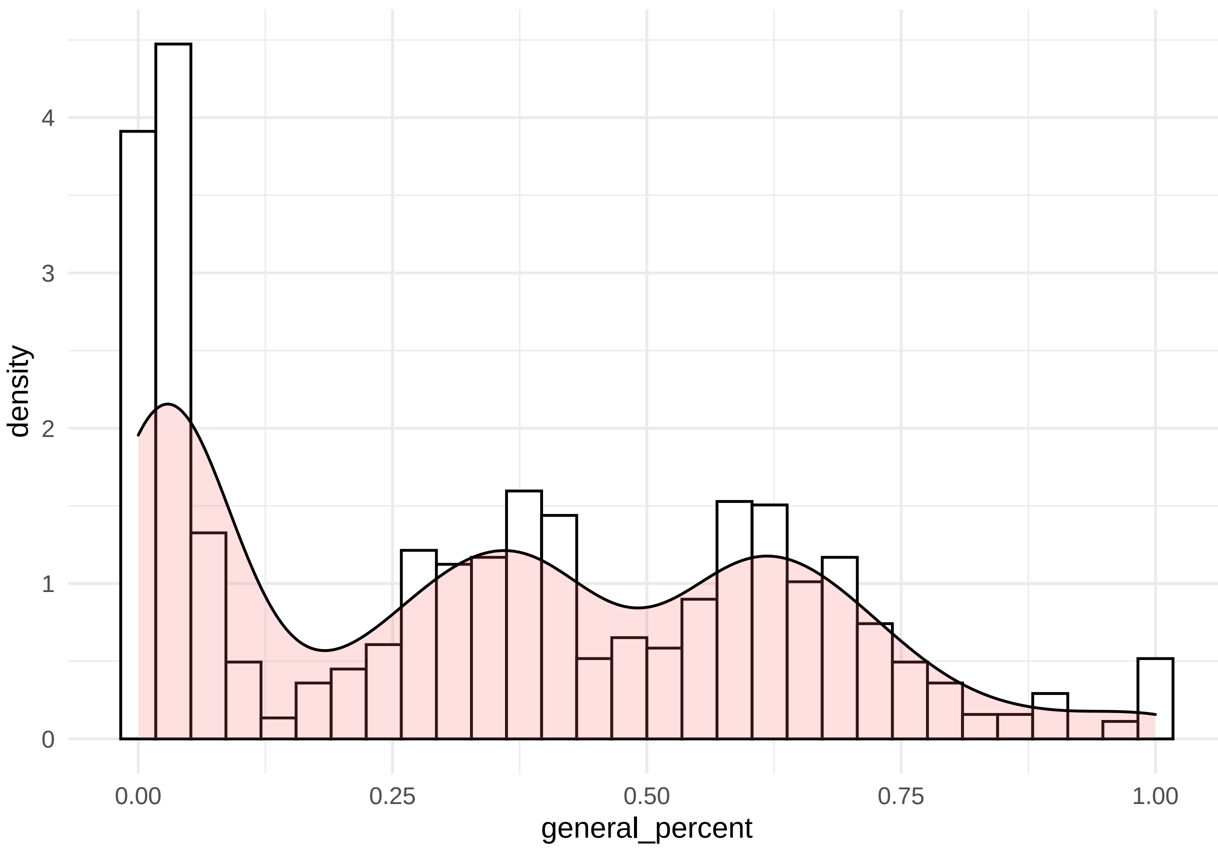
Your work

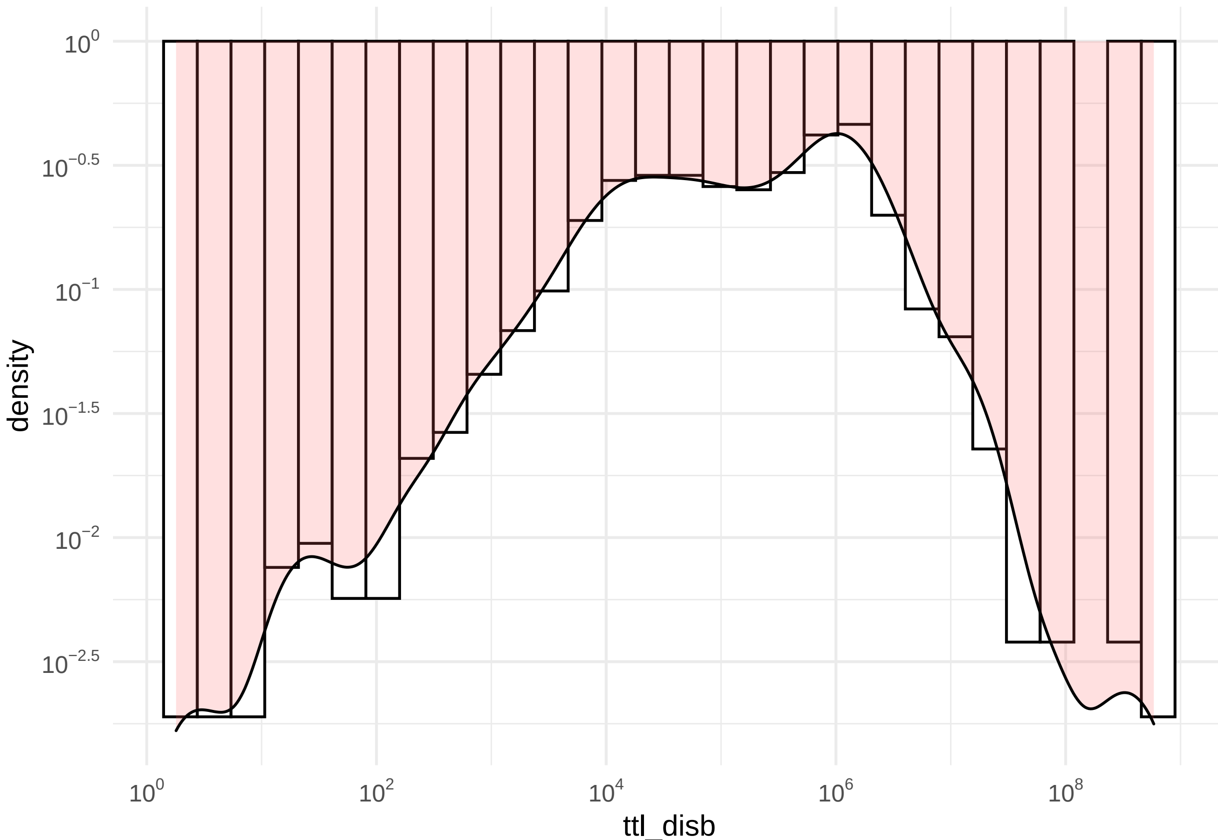
- We want to keep this work *relatively* constrained, which is why we're providing you with data through the `fec16` package. It is possible to gather all the information from current FEC reports, but it would require you to make a series of API calls that would pull us away from the core modeling tasks that we want you to focus on instead.
- Throughout this assignment, limit yourself to functions that are within the `tidyverse` family of packages: `dplyr`, `ggplot`, `patchwork`, and `magrittr` for wrangling and exploration and `base`, `stats`, `sandwich` and `lmtest` for modeling and testing. You do not *have* to use these packages; but try to limit yourself to using only these.

```r
candidates    <- fec16::candidates
results_house <- fec16::results_house
campaigns     <- fec16::campaigns
```

1. What does the distribution of votes and of spending look like?

1. (3 points) In separate histograms, show both the distribution of votes (measured in `results_house$general_percent` for now) and spending (measured in `ttl_disb`). Use a log transform if appropriate for each visualization. How would you describe what you see in these two plots?





2. Exploring the relationship between spending and votes.

2. (3 points) Create a new dataframe by joining `results_house` and `campaigns` using the `inner_join` function from `dplyr`. (We use the format `package::function` – so `dplyr::inner_join`.)

```
nrow(results_house)
```

```
## [1] 2110
```

```
nrow(campaigns)
```

```
## [1] 1898
```

```
d1 <- dplyr::inner_join(results_house, campaigns, by = NULL)
```

```
## Joining, by = "cand_id"
```

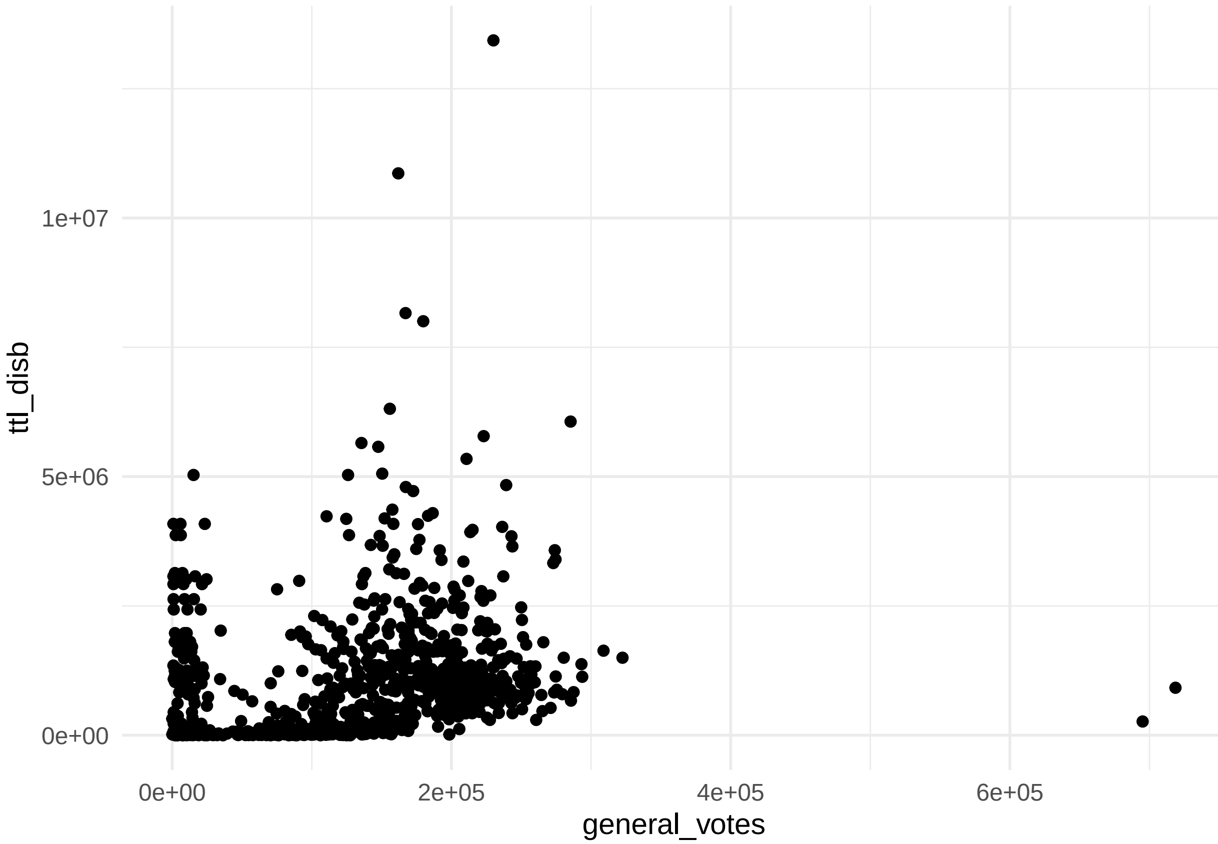
```
nrow(d1)
```

```
## [1] 1342
```

3. (3 points) Produce a scatter plot of `general_votes` on the y-axis and `ttl_disb` on the x-axis. What do you observe about the shape of the joint distribution?

```r
ggplot(d1, aes(x=general_votes, y=ttl_disb)) + geom_point()
```

```
## Warning: Removed 462 rows containing missing values (geom_point).
```

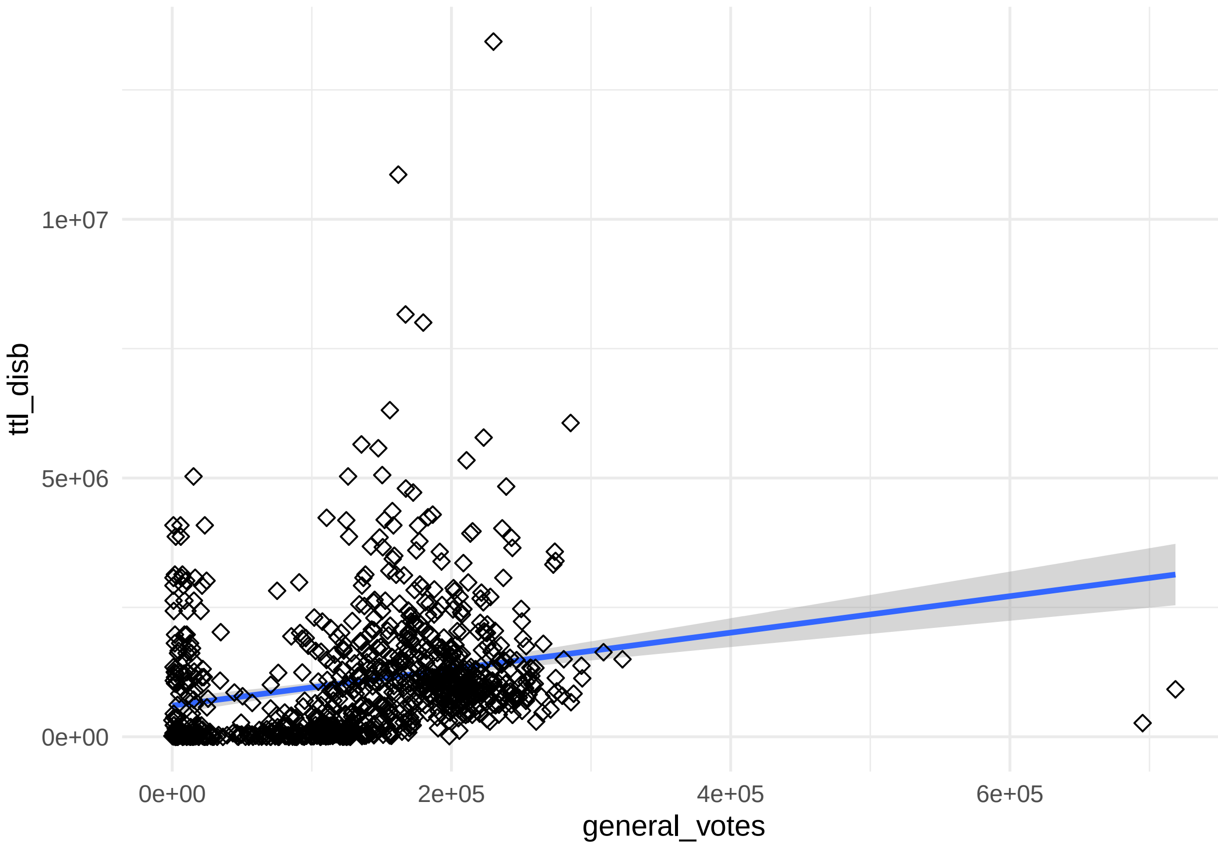



```r
sp <- ggplot(d1, aes(x=general_votes, y=ttl_disb  )) +
  geom_smooth(method=lm)+
  geom_point(size=2, shape=23)

sp
```

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

```
## Warning: Removed 462 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 462 rows containing missing values (geom_point).
```



4. (3 points) Create a new variable to indicate whether each individual is a "Democrat", "Republican" or "Other Party".

- Here's an example of how you might use `mutate` and `case_when` together to create a variable.

```
starwars %>%
  select(name:mass, gender, species) %>%
  mutate(
  type = case_when(
    height > 200 | mass > 200 ~ "large",
    species == "Droid"        ~ "robot",
    TRUE                      ~ "other"
    )
  )
```

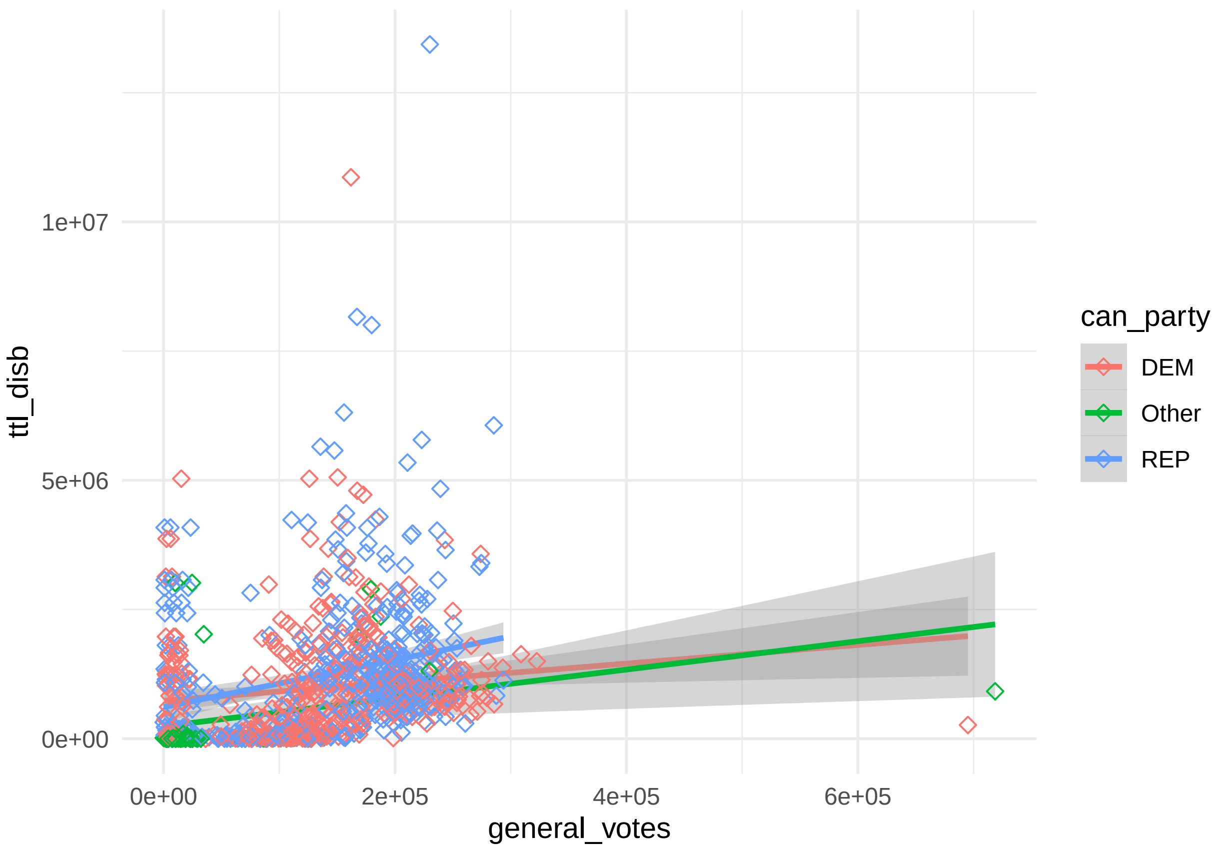
Once you've produced the new variable, plot your scatter plot again, but this time adding an argument into the `aes()` function that colors the points by party membership. What do you observe about the distribution of all three variables?

```
d2<-d1 %>%
  dplyr::select(cand_pty_affiliation, general_votes, ttl_disb, state) %>%
  na.omit() %>%
    mutate(
    can_party = case_when(
      cand_pty_affiliation=="REP" ~ "REP",
```

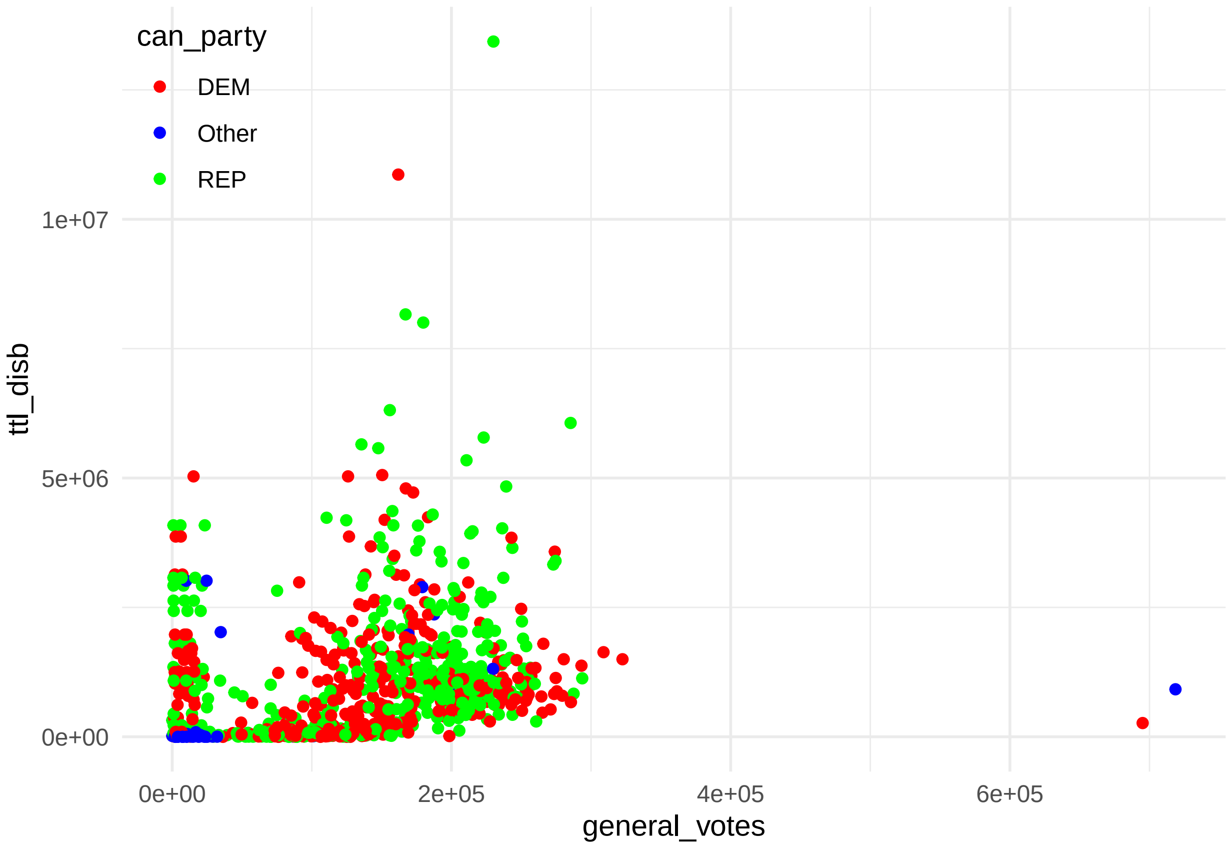
```
      cand_pty_affiliation=="DEM" ~ "DEM",
      TRUE ~ "Other"
    )
  )

d2<-d2 %>% dplyr::select(can_party, general_votes, ttl_disb, state)
```

```
sp <- ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_smooth(method=lm)+
  geom_point(size=2, shape=23)
sp
```

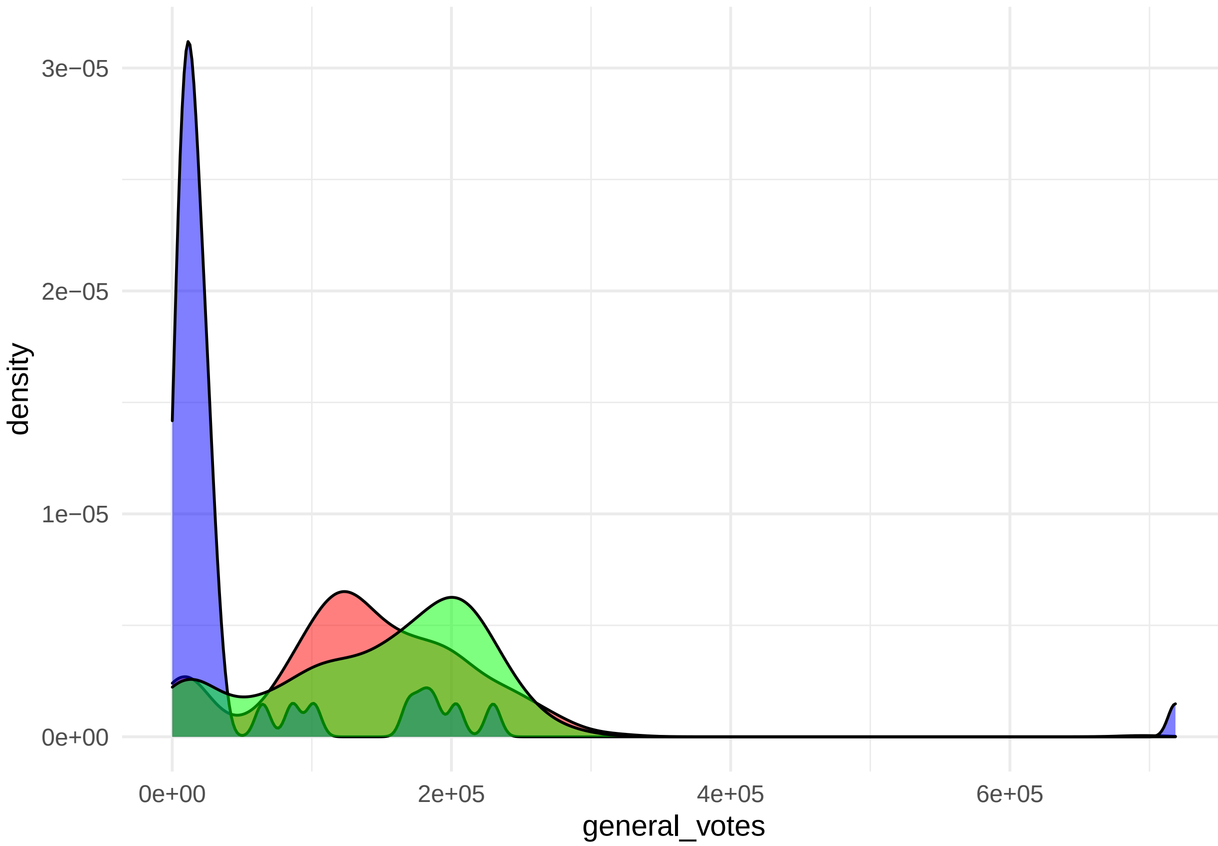
```
## `geom_smooth()` using formula 'y ~ x'
```



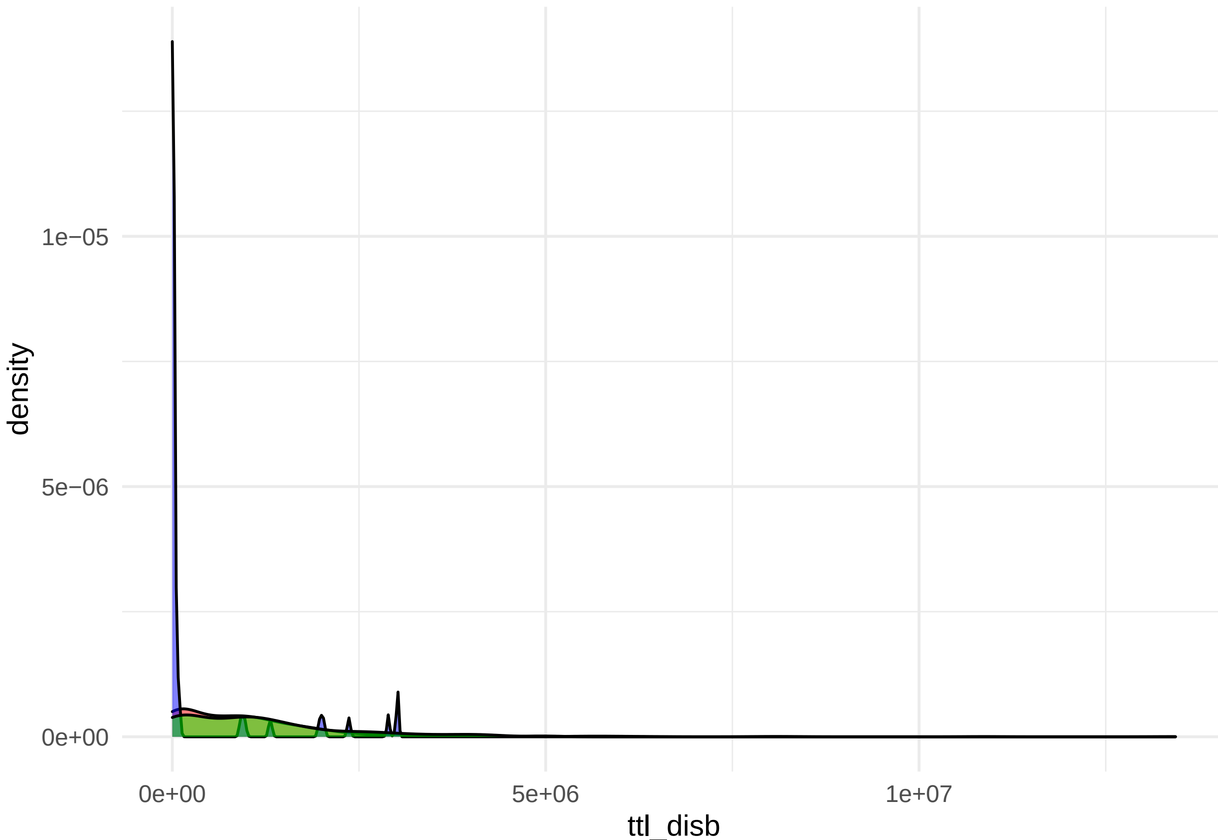

```
p1<-ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_point() +
  scale_color_manual(values = c("red", "blue", "green")) +
  theme(legend.position=c(0,1), legend.justification=c(0,1))
p1
```



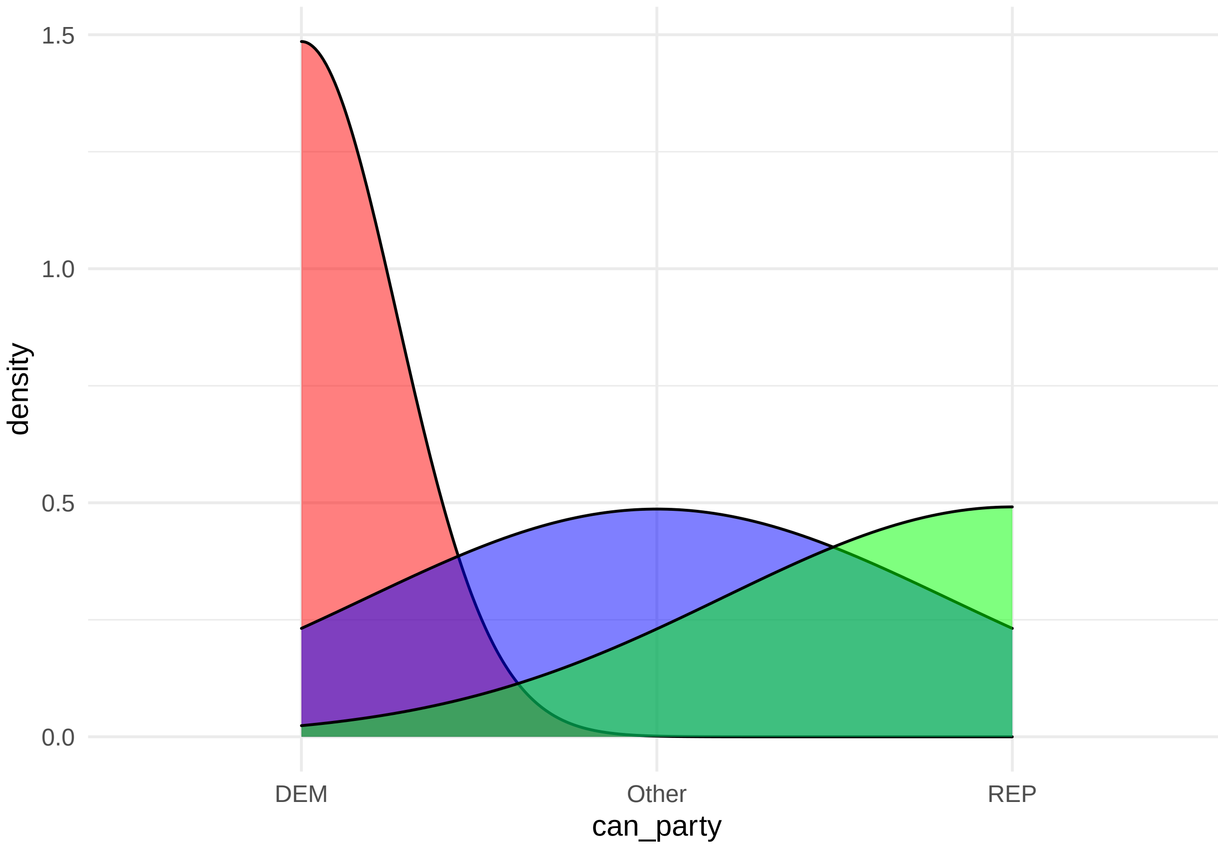

```
p2<-ggplot(d2, aes(x=general_votes, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values =  c("red", "blue", "green")) +
  theme(legend.position = "none")
p2
```




```
# Marginal density plot of y (right panel)
p3<-ggplot(d2, aes(x=ttl_disb, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values =  c("red", "blue", "green")) +
  theme(legend.position = "none")
p3
```



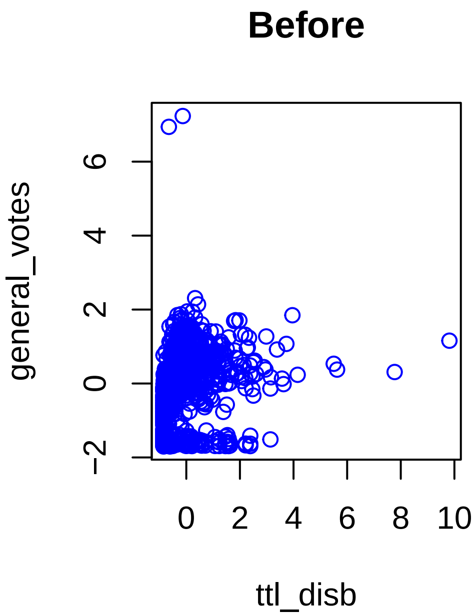

```
p3<-ggplot(d2, aes(x=can_party, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values =  c("red", "blue", "green")) +
  theme(legend.position = "none")
p3
```

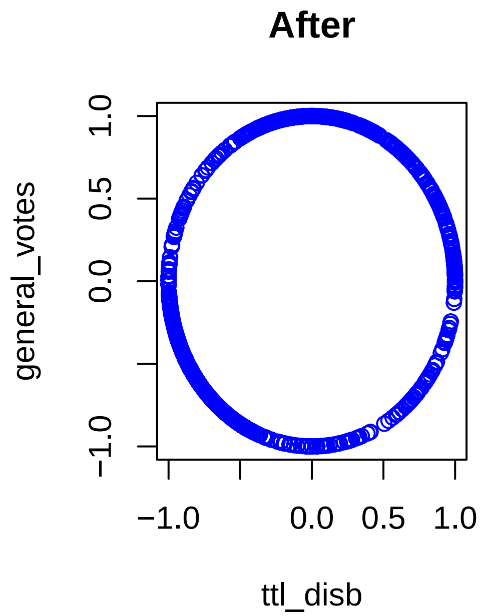


Produce a Descriptive Model

5. (5 Points) Given your observations, produce a linear model that you think does a good job at describing the relationship between candidate spending and votes they receive. You should decide what transformation to apply to spending (if any), what transformation to apply to votes (if any) and also how to include the party affiliation.

```
d2[d2 == -Inf] <- 0

sdat <- d2[, c("general_votes", "ttl_disb")]

imp <- preProcess(sdat, method = c("knnImpute"), k = 5)
sdat <- predict(imp, sdat)
transformed <- spatialSign(sdat)
transformed <- as.data.frame(transformed)
par(mfrow = c(1, 2), oma = c(2, 2, 2, 2))
plot(general_votes ~ ttl_disb, data = sdat, col = "blue", main = "Before")
plot(general_votes ~ ttl_disb, data = transformed, col = "blue", main = "After")
```






```r
d2$novotes<-transformed$"general_votes"
d2$nodisb<-transformed$"ttl_disb"

summary(d2)
```

```
##    can_party          general_votes       ttl_disb            state          
##  Length:880         Min.   :    55   Min.   :       0   Length:880        
##  Class :character   1st Qu.: 88229   1st Qu.:  102276   Class :character  
##  Mode  :character   Median :142597   Median :  830659   Mode  :character  
##                     Mean   :136932   Mean   : 1084565                     
##                     3rd Qu.:198290   3rd Qu.: 1527533                     
##                     Max.   :718591   Max.   :13433669                     
##     novotes             nodisb       
##  Min.   :-1.00000   Min.   :-1.0000  
##  1st Qu.:-0.65905   1st Qu.:-0.7263  
##  Median : 0.07400   Median :-0.2163  
##  Mean   : 0.07698   Mean   :-0.1272  
##  3rd Qu.: 0.90077   3rd Qu.: 0.4287  
##  Max.   : 1.00000   Max.   : 1.0000
```

```r
#d2<-transformed
```

```r
write.csv(d2, "d2.csv")
#summary(d2)
 # set the 'method' option
trans <- preProcess(d2, method = c("center", "scale"))
# use predict() function to get the final result
d3 <- predict(trans, d2)

d2$csvotes = d3$general_votes
d2$csdisb = d3$ttl_disb

write.csv(d2, "d2.csv")
summary(d2)
```

```
##   can_party         general_votes      ttl_disb           state
##  Length:880         Min.   :    55   Min.   :       0   Length:880
##  Class :character   1st Qu.: 88229   1st Qu.:  102276   Class :character
##  Mode  :character   Median :142597   Median :  830659   Mode  :character
##                     Mean   :136932   Mean   : 1084565
##                     3rd Qu.:198290   3rd Qu.: 1527533
##                     Max.   :718591   Max.   :13433669
##     novotes             nodisb           csvotes             csdisb
##  Min.   :-1.00000   Min.   :-1.0000   Min.   :-1.70236   Min.   :-0.8619
##  1st Qu.:-0.65905   1st Qu.:-0.7263   1st Qu.:-0.60573   1st Qu.:-0.7806
##  Median : 0.07400   Median :-0.2163   Median : 0.07045   Median :-0.2018
##  Mean   : 0.07698   Mean   :-0.1272   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.90077   3rd Qu.: 0.4287   3rd Qu.: 0.76311   3rd Qu.: 0.3520
##  Max.   : 1.00000   Max.   : 1.0000   Max.   : 7.23415   Max.   : 9.8139
```

```
write.csv(d3, "d3.csv")
```

```
#summary(d3)
write.csv(d3, "d3.csv")
#d2$disb <- log(d$tdisb)
#d2$votes <- log(d2$tvotes)


d2$logdisb <- log(d2$ttl_disb)
d2$logvotes <- log(d2$general_votes)
d2 <- na.omit(d2)
d2[d2 == -Inf] <- 0

#only original R2 = 0.5116
fit0 <- lm(d2$general_votes ~ d2$ttl_disb + d2$state + d2$can_party)
summary(fit0)
```

```
##
## Call:
## lm(formula = d2$general_votes ~ d2$ttl_disb + d2$state + d2$can_party)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -415756  -39794   -5242   36879  269903
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.139e+05  4.120e+04   2.763  0.00585 **
## d2$ttl_disb        1.518e-02  1.661e-03   9.138  < 2e-16 ***
## d2$stateAL         4.215e+04  4.471e+04   0.943  0.34607
## d2$stateAR         4.851e+04  4.867e+04   0.997  0.31926
## d2$stateAS        -1.112e+05  5.312e+04  -2.094  0.03660 *
## d2$stateAZ         1.136e+04  4.377e+04   0.260  0.79531
## d2$stateCA        -1.482e+02  4.153e+04  -0.004  0.99715
## d2$stateCO         5.012e+04  4.395e+04   1.140  0.25447
## d2$stateCT        -2.964e+04  4.378e+04  -0.677  0.49848
```

```
## d2$stateDC         1.442e+05  7.125e+04   2.024  0.04325 *
## d2$stateDE         7.554e+04  5.815e+04   1.299  0.19425
## d2$stateFL         2.945e+04  4.184e+04   0.704  0.48167
## d2$stateGA         5.646e+04  4.335e+04   1.302  0.19316
## d2$stateGU        -1.001e+05  5.817e+04  -1.721  0.08558 .
## d2$stateHI         2.308e+04  5.312e+04   0.434  0.66406
## d2$stateIA         4.883e+04  4.597e+04   1.062  0.28847
## d2$stateID         5.123e+04  5.041e+04   1.016  0.30988
## d2$stateIL         4.114e+04  4.244e+04   0.969  0.33260
## d2$stateIN         2.570e+04  4.350e+04   0.591  0.55475
## d2$stateKS        -9.865e+03  4.547e+04  -0.217  0.82828
## d2$stateKY         5.083e+04  4.471e+04   1.137  0.25585
## d2$stateLA        -4.898e+04  4.324e+04  -1.133  0.25764
## d2$stateMA         7.588e+04  4.365e+04   1.738  0.08254 .
## d2$stateMD         4.579e+04  4.396e+04   1.042  0.29784
## d2$stateME         4.369e+04  5.037e+04   0.867  0.38598
## d2$stateMI         3.268e+04  4.254e+04   0.768  0.44261
## d2$stateMN         5.922e+04  4.386e+04   1.350  0.17730
## d2$stateMO         6.812e+04  4.419e+04   1.541  0.12358
## d2$stateMP         6.315e+03  7.178e+04   0.088  0.92991
## d2$stateMS         3.385e+04  4.749e+04   0.713  0.47613
## d2$stateMT         6.441e+04  5.838e+04   1.103  0.27026
## d2$stateNC         5.234e+04  4.274e+04   1.225  0.22102
## d2$stateND         2.870e+04  5.317e+04   0.540  0.58947
## d2$stateNE         3.875e+04  5.037e+04   0.769  0.44191
## d2$stateNH        -9.872e+03  4.869e+04  -0.203  0.83938
## d2$stateNJ         1.872e+04  4.295e+04   0.436  0.66310
## d2$stateNM         5.723e+03  4.865e+04   0.118  0.90638
## d2$stateNV        -5.810e+03  4.546e+04  -0.128  0.89835
## d2$stateNY        -7.894e+04  4.146e+04  -1.904  0.05726 .
## d2$stateOH         4.231e+04  4.248e+04   0.996  0.31949
## d2$stateOK         4.615e+04  4.867e+04   0.948  0.34321
## d2$stateOR         8.356e+04  4.662e+04   1.792  0.07343 .
## d2$statePA         5.607e+04  4.251e+04   1.319  0.18754
## d2$statePR         4.313e+05  5.342e+04   8.074 2.42e-15 ***
## d2$stateRI        -2.106e+04  5.037e+04  -0.418  0.67599
## d2$stateSC        -1.508e+04  4.336e+04  -0.348  0.72814
## d2$stateSD         4.393e+04  5.815e+04   0.756  0.45016
## d2$stateTN         2.770e+04  4.397e+04   0.630  0.52899
## d2$stateTX         1.064e+04  4.187e+04   0.254  0.79956
## d2$stateUT        -3.352e+03  4.597e+04  -0.073  0.94188
## d2$stateVA         4.654e+04  4.303e+04   1.082  0.27975
## d2$stateVI        -1.045e+05  7.125e+04  -1.466  0.14300
## d2$stateVT         1.387e+05  7.124e+04   1.947  0.05184 .
## d2$stateWA         2.911e+04  4.361e+04   0.667  0.50465
## d2$stateWI         4.214e+04  4.350e+04   0.969  0.33301
## d2$stateWV        -8.468e+03  4.665e+04  -0.182  0.85599
## d2$stateWY        -8.747e+03  5.316e+04  -0.165  0.86935
## d2$can_partyOther -1.104e+05  9.287e+03 -11.891  < 2e-16 ***
## d2$can_partyREP    1.661e+03  4.157e+03   0.400  0.68957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58140 on 821 degrees of freedom
```

```
## Multiple R-squared:  0.5116, Adjusted R-squared:  0.477
## F-statistic: 14.82 on 58 and 821 DF,  p-value: < 2.2e-16
```

```r
#only no outlier data R2 = 0.4055
fit1 <- lm(d2$novotes ~ d2$nodisb + d2$state + d2$can_party)
summary(fit1)
```

```
##
## Call:
## lm(formula = d2$novotes ~ d2$nodisb + d2$state + d2$can_party)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42419 -0.43776 -0.09312  0.45242  1.65909
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.27065    0.41796  -0.648   0.5175
## d2$nodisb          0.37367    0.03213  11.632   <2e-16 ***
## d2$stateAL         0.67185    0.45416   1.479   0.1394
## d2$stateAR         0.94994    0.49453   1.921   0.0551 .
## d2$stateAS        -0.47494    0.53940  -0.880   0.3788
## d2$stateAZ         0.38926    0.44471   0.875   0.3817
## d2$stateCA         0.23880    0.42174   0.566   0.5714
## d2$stateCO         0.69086    0.44638   1.548   0.1221
## d2$stateCT         0.15191    0.44462   0.342   0.7327
## d2$stateDC         1.33633    0.72341   1.847   0.0651 .
## d2$stateDE         1.11124    0.59060   1.882   0.0603 .
## d2$stateFL         0.58173    0.42505   1.369   0.1715
## d2$stateGA         0.93794    0.44044   2.130   0.0335 *
## d2$stateGU        -0.48972    0.59077  -0.829   0.4074
## d2$stateHI         0.78311    0.53979   1.451   0.1472
## d2$stateIA         0.74443    0.46667   1.595   0.1111
## d2$stateID         0.77779    0.51222   1.518   0.1293
## d2$stateIL         0.58372    0.43094   1.355   0.1759
## d2$stateIN         0.56613    0.44171   1.282   0.2003
## d2$stateKS         0.23598    0.46165   0.511   0.6094
## d2$stateKY         0.76758    0.45422   1.690   0.0914 .
## d2$stateLA        -0.22768    0.43927  -0.518   0.6044
## d2$stateMA         0.83929    0.44350   1.892   0.0588 .
## d2$stateMD         0.60027    0.44641   1.345   0.1791
## d2$stateME         0.63680    0.51121   1.246   0.2132
## d2$stateMI         0.63358    0.43198   1.467   0.1428
## d2$stateMN         0.83033    0.44502   1.866   0.0624 .
## d2$stateMO         0.93168    0.44894   2.075   0.0383 *
## d2$stateMP         0.37485    0.72885   0.514   0.6072
## d2$stateMS         0.79287    0.48256   1.643   0.1008
## d2$stateMT         0.40596    0.59049   0.687   0.4920
## d2$stateNC         0.83001    0.43446   1.910   0.0564 .
## d2$stateND         0.49196    0.54010   0.911   0.3626
## d2$stateNE         0.59092    0.51128   1.156   0.2481
## d2$stateNH         0.18938    0.49423   0.383   0.7017
## d2$stateNJ         0.54196    0.43622   1.242   0.2144
## d2$stateNM         0.40963    0.49405   0.829   0.4073
```

```
## d2$stateNV          0.08137    0.46157   0.176   0.8601
## d2$stateNY         -0.32091    0.42087  -0.762   0.4460
## d2$stateOH          0.71703    0.43169   1.661   0.0971 .
## d2$stateOK          0.88984    0.49424   1.800   0.0722 .
## d2$stateOR          1.05039    0.47348   2.218   0.0268 *
## d2$statePA          0.74362    0.43173   1.722   0.0854 .
## d2$statePR          1.27125    0.54241   2.344   0.0193 *
## d2$stateRI         -0.08019    0.51242  -0.156   0.8757
## d2$stateSC          0.29949    0.44059   0.680   0.4969
## d2$stateSD          0.54419    0.59041   0.922   0.3569
## d2$stateTN          0.69502    0.44690   1.555   0.1203
## d2$stateTX          0.44547    0.42533   1.047   0.2952
## d2$stateUT          0.31092    0.46679   0.666   0.5055
## d2$stateVA          0.77208    0.43720   1.766   0.0778 .
## d2$stateVI         -0.52556    0.72346  -0.726   0.4678
## d2$stateVT          1.31569    0.72331   1.819   0.0693 .
## d2$stateWA          0.60705    0.44299   1.370   0.1710
## d2$stateWI          0.64472    0.44181   1.459   0.1449
## d2$stateWV          0.26609    0.47392   0.561   0.5746
## d2$stateWY          0.10441    0.53977   0.193   0.8467
## d2$can_partyOther -0.81900    0.09368  -8.742   <2e-16 ***
## d2$can_partyREP    0.05703    0.04226   1.350   0.1775
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5902 on 821 degrees of freedom
## Multiple R-squared:  0.4055, Adjusted R-squared:  0.3635
## F-statistic: 9.655 on 58 and 821 DF,  p-value: < 2.2e-16
```

```
#only original, log(spending) data R2 = 0.5534
fit2 <- lm(d2$logvotes ~ d2$logdisb + d2$state + d2$can_party)
summary(fit2)
```

```
##
## Call:
## lm(formula = d2$logvotes ~ d2$logdisb + d2$state + d2$can_party)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1414 -0.2226  0.0358  0.2405  2.4223
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         9.728436   0.661912  14.697  < 2e-16 ***
## d2$logdisb          0.150433   0.014935  10.072  < 2e-16 ***
## d2$stateAL          0.393952   0.683236   0.577  0.56437
## d2$stateAR          0.457498   0.743716   0.615  0.53862
## d2$stateAS         -3.456836   0.812874  -4.253 2.36e-05 ***
## d2$stateAZ          0.209046   0.668994   0.312  0.75476
## d2$stateCA          0.089661   0.634685   0.141  0.88769
## d2$stateCO          0.435558   0.671695   0.648  0.51688
## d2$stateCT         -0.744538   0.668925  -1.113  0.26602
## d2$stateDC          0.796079   1.088524   0.731  0.46478
## d2$stateDE          0.537825   0.888427   0.605  0.54510
```

```
## d2$stateFL         0.234287   0.639452   0.366  0.71417
## d2$stateGA         0.534132   0.662755   0.806  0.42052
## d2$stateGU        -1.740463   0.888881  -1.958  0.05056 .
## d2$stateHI         0.153200   0.811586   0.189  0.85032
## d2$stateIA         0.371649   0.702316   0.529  0.59683
## d2$stateID         0.537484   0.770411   0.698  0.48559
## d2$stateIL         0.241059   0.648423   0.372  0.71017
## d2$stateIN         0.256032   0.664709   0.385  0.70021
## d2$stateKS        -0.502246   0.694669  -0.723  0.46989
## d2$stateKY         0.464223   0.683298   0.679  0.49708
## d2$stateLA        -0.843382   0.660847  -1.276  0.20224
## d2$stateMA         0.500444   0.667004   0.750  0.45330
## d2$stateMD         0.308020   0.671664   0.459  0.64665
## d2$stateME         0.405274   0.769367   0.527  0.59850
## d2$stateMI         0.266784   0.650160   0.410  0.68167
## d2$stateMN         0.698374   0.669780   1.043  0.29740
## d2$stateMO         0.449121   0.675175   0.665  0.50611
## d2$stateMP        -0.511372   1.096675  -0.466  0.64113
## d2$stateMS         0.578777   0.726874   0.796  0.42611
## d2$stateMT         0.431390   0.888541   0.486  0.62745
## d2$stateNC         0.545448   0.653399   0.835  0.40408
## d2$stateND         0.374428   0.812575   0.461  0.64507
## d2$stateNE         0.306547   0.769509   0.398  0.69046
## d2$stateNH        -1.385793   0.743843  -1.863  0.06282 .
## d2$stateNJ         0.166043   0.656496   0.253  0.80039
## d2$stateNM         0.074438   0.743320   0.100  0.92026
## d2$stateNV        -0.025407   0.694668  -0.037  0.97083
## d2$stateNY        -1.962890   0.633398  -3.099  0.00201 **
## d2$stateOH         0.414593   0.649312   0.639  0.52332
## d2$stateOK         0.378501   0.743560   0.509  0.61086
## d2$stateOR         0.552123   0.712327   0.775  0.43851
## d2$statePA         0.407956   0.649539   0.628  0.53013
## d2$statePR         1.901748   0.816216   2.330  0.02005 *
## d2$stateRI         0.031556   0.770074   0.041  0.96732
## d2$stateSC        -0.634812   0.662818  -0.958  0.33847
## d2$stateSD         0.314075   0.888357   0.354  0.72377
## d2$stateTN         0.364948   0.672684   0.543  0.58760
## d2$stateTX         0.212269   0.640017   0.332  0.74023
## d2$stateUT         0.026838   0.702353   0.038  0.96953
## d2$stateVA         0.391433   0.657551   0.595  0.55181
## d2$stateVI        -2.059756   1.088585  -1.892  0.05882 .
## d2$stateVT         0.716088   1.088472   0.658  0.51080
## d2$stateWA         0.323453   0.666571   0.485  0.62763
## d2$stateWI         0.426686   0.664754   0.642  0.52114
## d2$stateWV        -0.033487   0.712746  -0.047  0.96254
## d2$stateWY        -0.009488   0.812715  -0.012  0.99069
## d2$can_partyOther -1.593954   0.148787 -10.713  < 2e-16 ***
## d2$can_partyREP   -0.100363   0.063257  -1.587  0.11299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8884 on 821 degrees of freedom
## Multiple R-squared:  0.5534, Adjusted R-squared:  0.5219
## F-statistic: 17.54 on 58 and 821 DF,  p-value: < 2.2e-16
```

```r
#only original, log(spending) data R2 = 0.6041
fit3 <- lm(d2$general_votes ~ d2$logdisb + d2$state + d2$can_party)
summary(fit3)
```

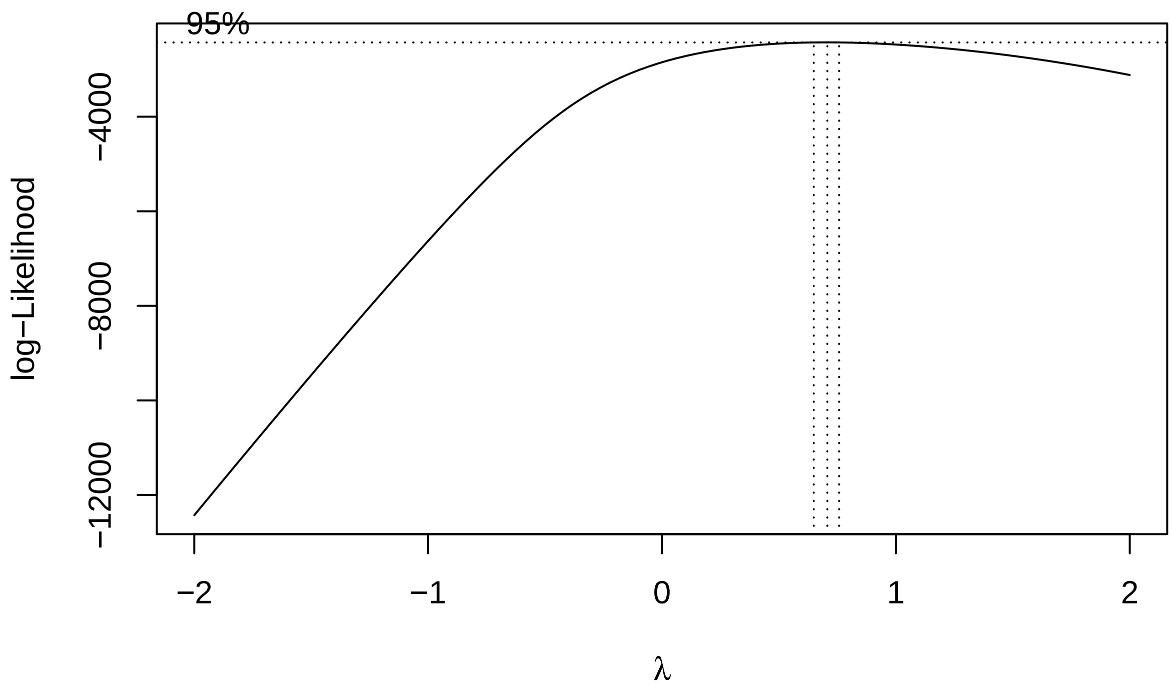
```
##
## Call:
## lm(formula = d2$general_votes ~ d2$logdisb + d2$state + d2$can_party)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -378949  -35379   -1422   30616  228002
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -78916.4    39004.1  -2.023   0.0434 *
## d2$logdisb           15114.2      880.1  17.174  < 2e-16 ***
## d2$stateAL           56260.9    40260.6   1.397   0.1627
## d2$stateAR           57802.7    43824.5   1.319   0.1876
## d2$stateAS          -72654.9    47899.7  -1.517   0.1297
## d2$stateAZ           29836.2    39421.4   0.757   0.4494
## d2$stateCA           17128.9    37399.7   0.458   0.6471
## d2$stateCO           64120.1    39580.5   1.620   0.1056
## d2$stateCT          -19862.7    39417.3  -0.504   0.6145
## d2$stateDC          146804.2    64142.8   2.289   0.0223 *
## d2$stateDE           81261.9    52351.8   1.552   0.1210
## d2$stateFL           47406.9    37680.6   1.258   0.2087
## d2$stateGA           76113.3    39053.8   1.949   0.0516 .
## d2$stateGU          -85174.9    52378.5  -1.626   0.1043
## d2$stateHI           24440.2    47823.8   0.511   0.6095
## d2$stateIA           56284.4    41384.9   1.360   0.1742
## d2$stateID           64117.5    45397.5   1.412   0.1582
## d2$stateIL           53099.1    38209.2   1.390   0.1650
## d2$stateIN           39081.9    39168.9   0.998   0.3187
## d2$stateKS           -5399.2    40934.3  -0.132   0.8951
## d2$stateKY           66803.2    40264.3   1.659   0.0975 .
## d2$stateLA          -36807.4    38941.3  -0.945   0.3448
## d2$stateMA           84732.1    39304.1   2.156   0.0314 *
## d2$stateMD           53793.9    39578.7   1.359   0.1745
## d2$stateME           59654.7    45336.0   1.316   0.1886
## d2$stateMI           50454.6    38311.6   1.317   0.1882
## d2$stateMN           67285.5    39467.7   1.705   0.0886 .
## d2$stateMO           74373.2    39785.7   1.869   0.0619 .
## d2$stateMP           -3586.2    64623.1  -0.055   0.9558
## d2$stateMS           70879.1    42832.0   1.655   0.0983 .
## d2$stateMT           94261.3    52358.5   1.800   0.0722 .
## d2$stateNC           72652.2    38502.5   1.887   0.0595 .
## d2$stateND           44297.4    47882.1   0.925   0.3552
## d2$stateNE           42828.1    45344.4   0.945   0.3452
## d2$stateNH          -13691.4    43832.0  -0.312   0.7548
## d2$stateNJ           38246.2    38684.9   0.989   0.3231
## d2$stateNM           12176.5    43801.2   0.278   0.7811
## d2$stateNV            1680.0    40934.3   0.041   0.9673
## d2$stateNY          -71680.4    37323.9  -1.920   0.0551 .
```

```
## d2$stateOH          59912.9    38261.6   1.566   0.1178
## d2$stateOK          52178.6    43815.3   1.191   0.2340
## d2$stateOR          91212.5    41974.8   2.173   0.0301 *
## d2$statePA          64973.1    38275.0   1.698   0.0900 .
## d2$statePR         434255.9    48096.7   9.029  < 2e-16 ***
## d2$stateRI           2223.8    45377.7   0.049   0.9609
## d2$stateSC           3311.5    39057.5   0.085   0.9325
## d2$stateSD          53147.5    52347.7   1.015   0.3103
## d2$stateTN          56307.3    39638.8   1.421   0.1558
## d2$stateTX          27554.8    37713.9   0.731   0.4652
## d2$stateUT           8442.1    41387.1   0.204   0.8384
## d2$stateVA          57819.4    38747.1   1.492   0.1360
## d2$stateVI         -98991.8    64146.4  -1.543   0.1232
## d2$stateVT         138293.3    64139.7   2.156   0.0314 *
## d2$stateWA          43843.3    39278.6   1.116   0.2647
## d2$stateWI          65675.6    39171.6   1.677   0.0940 .
## d2$stateWV             76.6    41999.6   0.002   0.9985
## d2$stateWY          16878.2    47890.3   0.352   0.7246
## d2$can_partyOther -72265.7     8767.5  -8.242 6.65e-16 ***
## d2$can_partyREP       683.6     3727.5   0.183   0.8545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52350 on 821 degrees of freedom
## Multiple R-squared:  0.6041, Adjusted R-squared:  0.5761
## F-statistic:  21.6 on 58 and 821 DF,  p-value: < 2.2e-16
```

```
#Y = d2$general_votes
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:patchwork':
##
##     area

## The following object is masked from 'package:dplyr':
##
##     select
```

```
b <- boxcox(general_votes ~ logdisb + state + can_party, data = d2)
```




```
#b
lambda <- b$x
lik <-b$y
bc<-cbind(lambda, lik)
bc[order(-lik),]
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'
```

```
##          lambda       lik
## [1,] -2.000000 -12428.90
## [2,] -1.959596 -12186.52
```

```
lambda<- 0.61
d2$lamvotes <- (d2$general_votes^lambda-1)/lambda


m1<-lm(lamvotes ~ logdisb + state + can_party, data = d2)
summary(m1)
```

```
##
## Call:
## lm(formula = lamvotes ~ logdisb + state + can_party, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2682.19  -350.59    -2.88   303.52  1895.70
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -11.634    434.772  -0.027  0.97866
```

```
## logdisb          157.107      9.810  16.015  < 2e-16 ***
## stateAL          524.181    448.779   1.168  0.24314
## stateAR          564.518    488.505   1.156  0.24818
## stateAS        -1372.188    533.931  -2.570  0.01035 *
## stateAZ          276.312    439.424   0.629  0.52965
## stateCA          138.621    416.889   0.333  0.73959
## stateCO          600.158    441.198   1.360  0.17411
## stateCT         -385.851    439.379  -0.878  0.38011
## stateDC         1293.716    714.990   1.809  0.07075 .
## stateDE          760.799    583.557   1.304  0.19269
## stateFL          415.572    420.020   0.989  0.32275
## stateGA          711.203    435.326   1.634  0.10270
## stateGU        -1237.859    583.855  -2.120  0.03429 *
## stateHI          237.590    533.085   0.446  0.65594
## stateIA          531.536    461.311   1.152  0.24956
## stateID          606.015    506.039   1.198  0.23143
## stateIL          491.092    425.912   1.153  0.24923
## stateIN          352.393    436.610   0.807  0.41984
## stateKS         -161.472    456.289  -0.354  0.72352
## stateKY          621.769    448.819   1.385  0.16632
## stateLA         -570.354    434.073  -1.314  0.18923
## stateMA          744.284    438.117   1.699  0.08973 .
## stateMD          480.307    441.178   1.089  0.27661
## stateME          566.643    505.354   1.121  0.26250
## stateMI          451.871    427.053   1.058  0.29031
## stateMN          703.826    439.941   1.600  0.11002
## stateMO          682.664    443.484   1.539  0.12411
## stateMP         -227.912    720.343  -0.316  0.75178
## stateMS          687.475    477.442   1.440  0.15027
## stateMT          799.726    583.632   1.370  0.17098
## stateNC          697.916    429.181   1.626  0.10430
## stateND          394.216    533.734   0.739  0.46036
## stateNE          408.381    505.447   0.808  0.41935
## stateNH         -298.081    488.588  -0.610  0.54197
## stateNJ          341.875    431.215   0.793  0.42811
## stateNM          109.207    488.245   0.224  0.82307
## stateNV           -9.471    456.288  -0.021  0.98345
## stateNY        -1082.044    416.043  -2.601  0.00947 **
## stateOH          553.497    426.496   1.298  0.19473
## stateOK          495.513    488.403   1.015  0.31062
## stateOR          829.843    467.887   1.774  0.07650 .
## statePA          592.915    426.645   1.390  0.16499
## statePR         3179.462    536.126   5.930 4.45e-09 ***
## stateRI            4.193    505.818   0.008  0.99339
## stateSC         -147.704    435.368  -0.339  0.73450
## stateSD          481.364    583.511   0.825  0.40964
## stateTN          524.139    441.848   1.186  0.23587
## stateTX          258.842    420.391   0.616  0.53825
## stateUT           59.425    461.336   0.129  0.89754
## stateVA          541.096    431.908   1.253  0.21063
## stateVI        -1423.225    715.029  -1.990  0.04687 *
## stateVT         1207.326    714.955   1.689  0.09166 .
## stateWA          421.892    437.833   0.964  0.33553
## stateWI          595.455    436.639   1.364  0.17303
```

```
## stateWV          -36.806    468.162  -0.079  0.93736
## stateWY           96.287    533.826   0.180  0.85691
## can_partyOther -1013.103     97.730 -10.366  < 2e-16 ***
## can_partyREP     -15.185     41.550  -0.365  0.71485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 583.5 on 821 degrees of freedom
## Multiple R-squared:  0.6166, Adjusted R-squared:  0.5895
## F-statistic: 22.76 on 58 and 821 DF,  p-value: < 2.2e-16
```

6. (3 points) Interpret the model coefficients you estimate.

- Tasks to keep in mind as you're writing about your model:
  - At the time that you're writing and interpreting your regression coefficients you'll be *deep* in the analysis. Nobody will know more about the data than you do, at that point. *So, although it will feel tedious, be descriptive and thorough in describing your observations.*
  - It can be hard to strike the balance between: on the one hand, writing enough of the technical underpinnings to know that your model meets the assumptions that it must; and, on the other hand, writing little enough about the model assumptions that the implications of the model can still be clear. We're starting this practice now, so that by the end of Lab 2 you will have had several chances to strike this balance.

```
#lm(d2$general_votes ~ b1*d2$ttl_disb + b2)
```