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.

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?

ANSWER:

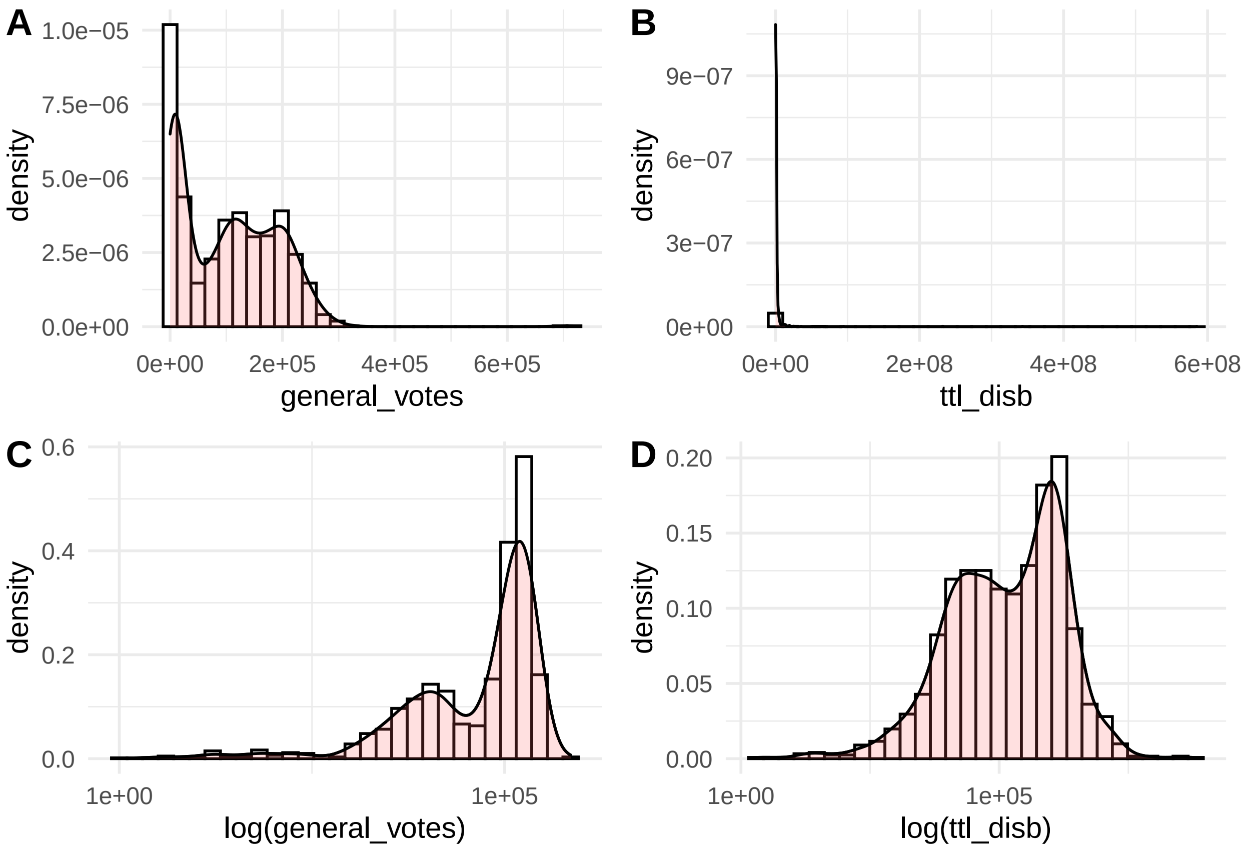
From my observation, the data `general_percent` and `ttl_disb` have the following problems:

- The original data of the 2 variables are not on the same scale (Fig. A-B) .
- Has skewness problems because the curve appears distorted and skewed to the left in a statistical distribution.
- The data are not centered.

At this stage, based on my finding, we need to perform data transforming including scaling, centering and skewness corrections.

I will perform Log transformation first, because log transform makes the data as "normal" as possible so that the statistical analysis results from this data become more valid, the log transformation reduces or removes the skewness of our original data. In detail I choose natural logarithm here for the purposes of linear modeling , i.e., using Log transformation replaces each variable x with a log(x). The results are shown in Fig. C-D, repectivelay. In C and D, after the transfermation, the vurves approcimately follows normal distribution, the graph appears symmetry, there are about as many data values on the left side of the median as on the right side.

I will do other data transformations later in the following questions. Data transformation can make our model working efficiently: distance based models perform well when data is pre-processed and transformed; having all features scaled it speeds up the model; better accuracy and more generalized model.


A
1.0e−05
7.5e−06
5.0e−06
2.5e−06
0.0e+00
density
0e+00
2e+05
4e+05
6e+05
general_votes
B
9e−07
6e−07
3e−07
0e+00
density
0e+00
2e+08
4e+08
6e+08
ttl_disb
C
0.6
0.4
0.2
0.0
density
1e+00
1e+05
log(general_votes)
D
0.20
0.15
0.10
0.05
0.00
density
1e+00
1e+05
log(ttl_disb)

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`.)

ANSWER:

Done the creation of new dataframe by joining `results_house` and `campaigns` using the `inner_join` function from `dplyr`. The new data frame is named "d1". A discription of "d1" is as the follows:

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

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

```
nrow(d1)
```

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

```
summary(d1)
```

```
##     state            district_id          cand_id            incumbent
##  Length:1342        Length:1342        Length:1342        Mode :logical
##  Class :character   Class :character   Class :character   FALSE:895
##  Mode  :character   Mode  :character   Mode  :character   TRUE :447
##
##
##
##
##     party            primary_votes     primary_percent    runoff_votes
##  Length:1342        Min.   :      1   Min.   :0.00015   Min.   : 1096
##  Class :character   1st Qu.:   8650   1st Qu.:0.19158   1st Qu.: 1464
##  Mode  :character   Median :  21299   Median :0.42257   Median : 8206
##                     Mean   :  32227   Mean   :0.48844   Mean   :11274
##                     3rd Qu.:  45638   3rd Qu.:0.78382   3rd Qu.:20082
##                     Max.   :326988   Max.   :1.00000   Max.   :25322
##                     NA's   :291      NA's   :292       NA's   :1330
##  runoff_percent   general_votes    general_percent     won
##  Min.   :0.3427   Min.   :    55   Min.   :0.0000   Mode :logical
##  1st Qu.:0.4624   1st Qu.: 88229   1st Qu.:0.3087   FALSE:850
##  Median :0.5000   Median :142597   Median :0.4773   TRUE :492
##  Mean   :0.5000   Mean   :136932   Mean   :0.4597
##  3rd Qu.:0.5376   3rd Qu.:198290   3rd Qu.:0.6406
##  Max.   :0.6573   Max.   :718591   Max.   :1.0000
##  NA's   :1330     NA's   :462      NA's   :463
##   footnotes          cand_name           cand_ici             pty_cd
##  Length:1342        Length:1342        Length:1342        Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:1.000
##  Mode  :character   Mode  :character   Mode  :character   Median :2.000
##                                                           Mean   :1.607
##                                                           3rd Qu.:2.000
##                                                           Max.   :3.000
##
```

```
##  cand_pty_affiliation  ttl_receipts      trans_from_auth       ttl_disb
##  Length:1342          Min.   :       0   Min.   :       0   Min.   :       0
##  Class :character     1st Qu.:   46612   1st Qu.:       0   1st Qu.:   46147
##  Mode  :character     Median :  398962   Median :       0   Median :  379570
##                       Mean   :  883177   Mean   :   26408   Mean   :  814754
##                       3rd Qu.: 1290266   3rd Qu.:       0   3rd Qu.: 1154148
##                       Max.   :19852221   Max.   :12374657   Max.   :13433669
##
##  trans_to_auth       coh_bop           coh_cop         cand_contrib
##  Min.   :     0   Min.   : -18681   Min.   : -32074   Min.   :       0
##  1st Qu.:     0   1st Qu.:      0   1st Qu.:      0   1st Qu.:       0
##  Median :     0   Median :      0   Median :   3881   Median :       0
##  Mean   :  7577   Mean   : 150271   Mean   : 218929   Mean   :   21879
##  3rd Qu.:     0   3rd Qu.:  85884   3rd Qu.: 170548   3rd Qu.:    1000
##  Max.   :766500   Max.   :3750024   Max.   :9098873   Max.   :13414225
##
##    cand_loans       other_loans     cand_loan_repay   other_loan_repay
##  Min.   :      0   Min.   :     0   Min.   :      0   Min.   :     0.0
##  1st Qu.:      0   1st Qu.:     0   1st Qu.:      0   1st Qu.:     0.0
##  Median :      0   Median :     0   Median :      0   Median :     0.0
##  Mean   :  56809   Mean   :  1049   Mean   :  12579   Mean   :   638.7
##  3rd Qu.:   9000   3rd Qu.:     0   3rd Qu.:      0   3rd Qu.:     0.0
##  Max.   :8050000   Max.   :350000   Max.   :1655854   Max.   :350000.0
##
##  debts_owed_by     ttl_indiv_contrib cand_office_st     cand_office_district
##  Min.   :  -1786   Min.   :      0   Length:1342        Length:1342
##  1st Qu.:      0   1st Qu.:  21310   Class :character   Class :character
##  Median :      0   Median : 207337   Mode  :character   Mode  :character
##  Mean   :  42528   Mean   : 464597
##  3rd Qu.:  12903   3rd Qu.: 638629
##  Max.   :2795000   Max.   :5975190
##
##  other_pol_cmte_contrib pol_pty_contrib   cvg_end_dt          indiv_refunds
##  Min.   :      0        Min.   :    0   Min.   :2015-08-10   Min.   : -1150
##  1st Qu.:      0        1st Qu.:    0   1st Qu.:2016-12-31   1st Qu.:     0
##  Median :  13700        Median :    0   Median :2016-12-31   Median :   200
##  Mean   : 305670        Mean   : 1230   Mean   :2016-11-30   Mean   :  6617
##  3rd Qu.: 506471        3rd Qu.:  150   3rd Qu.:2016-12-31   3rd Qu.:  5400
##  Max.   :3279747        Max.   :25400   Max.   :2017-01-31   Max.   :227497
##
##   cmte_refunds
##  Min.   :     0
##  1st Qu.:     0
##  Median :     0
##  Mean   :  1093
##  3rd Qu.:   250
##  Max.   :104758
##
```

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?

ANSWER:

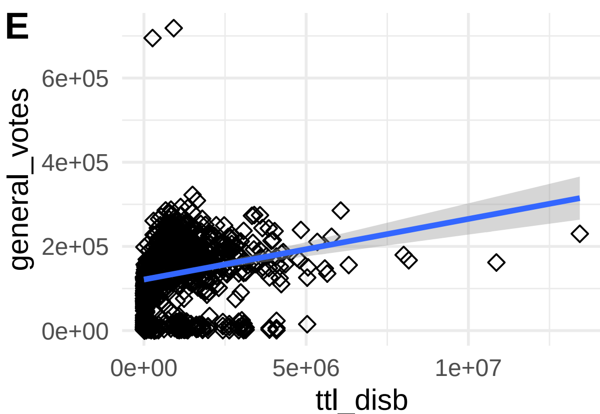
The scatter plot of `general_votes` on the y-axis and `ttl_disb` on the x-axis is shown below Fig.E. I also made a scatter plot using y = log(general_votes) and x= log(ttl_disb), as shown in Fig.F.

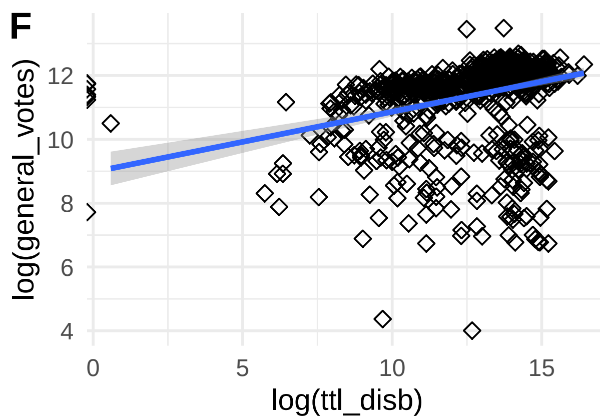
In general, a x-y scatter graph displays and compares values to show the numerical distribution of variables in a rectangular coordinate system. A two-dimensional scatter chart can show the data analysis of two variables to provide the relationship and correlation between the two. Scatter plots can provide three types of key information:

- Whether there is a quantitative correlation trend between variables;
- If there is a correlation trend, is it linear or non-linear;
- Observe whether there are outliers and analyze The influence of these outliers on the modeling analysis.

However, I couldn't find obvious correlation between variables since most of them look randomly distributed on the scatter plot. If there is a certain correlation, then most of the data points will be relatively dense and present in a certain trend, however I cannot figure it out it by simple observation.

By observing the distribution of data points on the scatter plot, I found there are some outliers.





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.

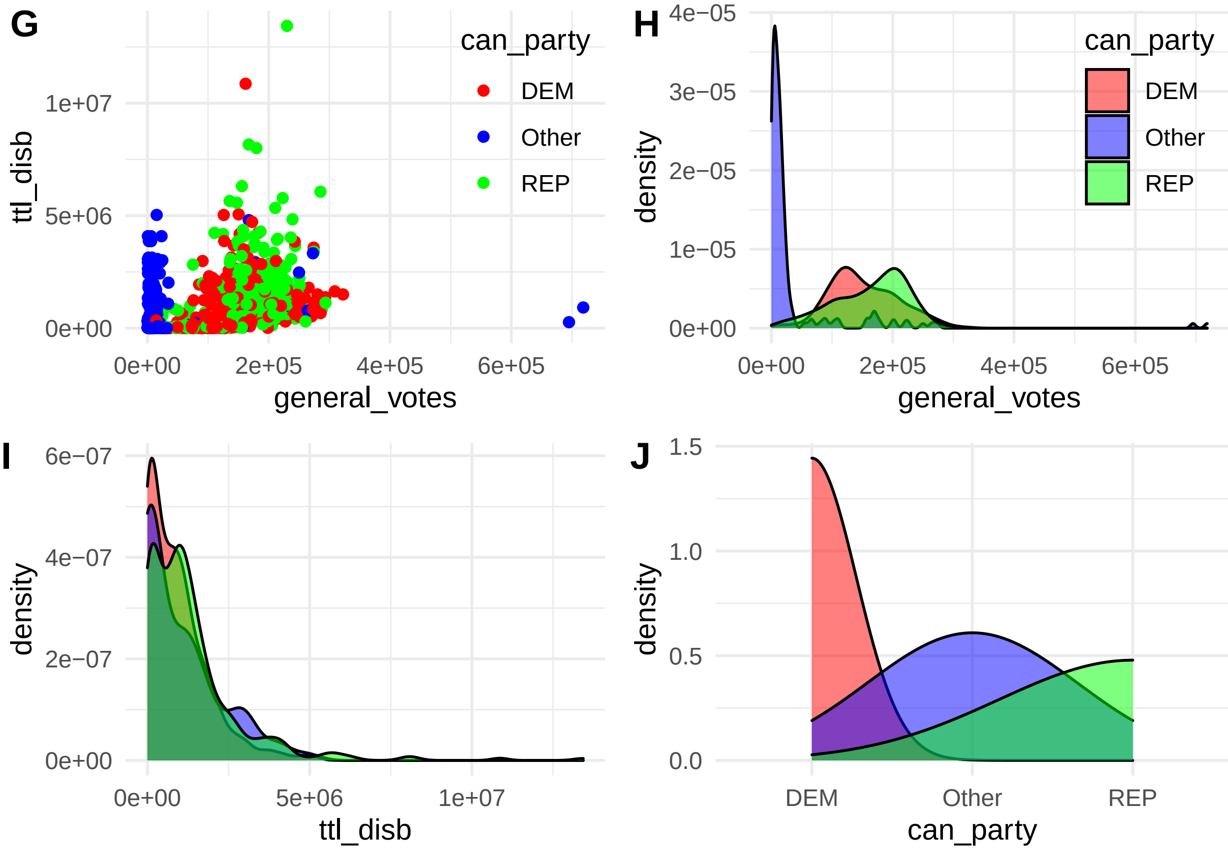
```
starwars %>%
  select(name:mass, gender, species) %>%
  mutate(
  type = case_when(
    height > 200 | mass > 200 ~ "large",
    species == "Droid"        ~ "robot",
    TRUE                      ~ "other"
    )
  )
```

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?

ANSWER:

The new variable has been produced, the new data frame is named "d2", a discription is as the follows :

```
##   can_party          general_votes       ttl_disb
##  Length:880          Min.   :    55   Min.   :       0
##  Class :character    1st Qu.: 88229   1st Qu.:  102276
##  Mode  :character    Median :142597   Median :  830659
##                      Mean   :136932   Mean   : 1084565
##                      3rd Qu.:198290   3rd Qu.: 1527533
##                      Max.   :718591   Max.   :13433669
```



From my observation in Fig H-J, the variable general_percent,ttl_disb and can_party have the following properties:

- The distribution of each of the three variables (i.e. can_party, ttl_disp, general_vote) are a combination of 3 different curves that are approximately following normal distributions.
- For each variable, the 3 curves in different color clustered by the 3 (i.e. DEM , REP, and Other) parties.
- Among the total 9 curves, each of the curves appears symmetry, there are about as many data values on the left side of the median as on the right side.
- Each of the curves has skewness problems because the curve appears distorted or skewed to the left or right in a statistical distribution.
- The data in each curve are not centered.

At this stage, based on my finding, the following decisions are made:

- A linear model can be created and fit the relationship between the general_votes and ttl_disb and can_party.
- Detailed analysis and pre-processing need to be done to the data using maths.
- further data transformations have to be performed.

Next, I will do the data pre-processing and model creation.

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.

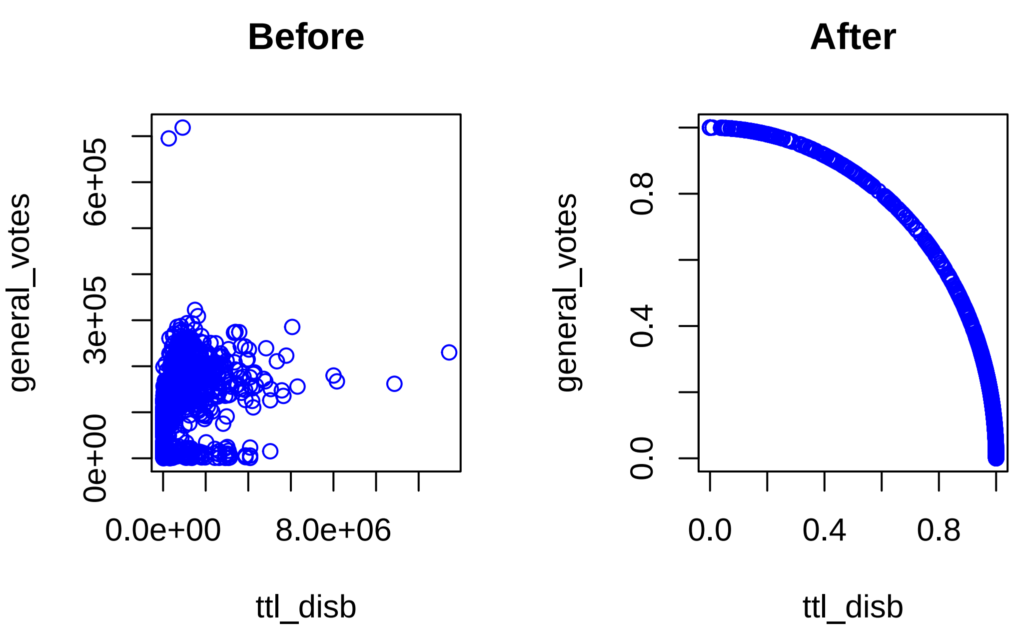
```
d5<-d2 %>%
  dplyr::select(can_party, general_votes, ttl_disb) %>%
  na.omit() %>%
    mutate(
    can_party = case_when(
      can_party=="REP" ~ 0,
      can_party=="DEM" ~ 1,
      TRUE ~ 2
    )
  )

d2<-d5 %>% dplyr::select(can_party, general_votes, ttl_disb)
```

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

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

imp <- preProcess(sdat, method = c("bagImpute"), 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"
d2$noparty<-transformed$"can_party"

#d2<-transformed
```

```r
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
d2$csparty = d3$can_party
```

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

fit0 <- lm(d2$general_votes ~ d2$ttl_disb  + d2$can_party)
summary(fit0)
```

```
##
## Call:
## lm(formula = d2$general_votes ~ d2$ttl_disb + d2$can_party)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -162812  -50839    -463   37128  645725
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.634e+05  4.080e+03  40.061  < 2e-16 ***
## d2$ttl_disb   1.163e-02  1.864e-03   6.238 6.88e-10 ***
## d2$can_party -5.062e+04  3.213e+03 -15.756  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69240 on 877 degrees of freedom
## Multiple R-squared:  0.2602, Adjusted R-squared:  0.2585
## F-statistic: 154.2 on 2 and 877 DF,  p-value: < 2.2e-16
```

```r
fit1 <- lm(d2$csvotes ~ d2$csdisb  + d2$csparty)
summary(fit1)
```

```
##
## Call:
## lm(formula = d2$csvotes ~ d2$csdisb + d2$csparty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0249 -0.6323 -0.0058  0.4618  8.0309
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.857e-16  2.903e-02   0.000        1
## d2$csdisb    1.820e-01  2.917e-02   6.238 6.88e-10 ***
## d2$csparty  -4.597e-01  2.917e-02 -15.756  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8611 on 877 degrees of freedom
## Multiple R-squared:  0.2602, Adjusted R-squared:  0.2585
## F-statistic: 154.2 on 2 and 877 DF,  p-value: < 2.2e-16
```

```
fit2 <- lm(d2$novotes ~ d2$nodisb  + d2$noparty)
summary(fit2)
```

```
##
## Call:
## lm(formula = d2$novotes ~ d2$nodisb + d2$noparty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25036 -0.09079 -0.01375  0.08107  0.24505
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.25036    0.01271   98.38   <2e-16 ***
## d2$nodisb     -1.09452    0.01426  -76.76   <2e-16 ***
## d2$noparty   -80.80898   79.99695   -1.01    0.313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.117 on 877 degrees of freedom
## Multiple R-squared:  0.8777, Adjusted R-squared:  0.8774
## F-statistic:  3146 on 2 and 877 DF,  p-value: < 2.2e-16
```

```
fit3 <- lm(d2$logvotes ~ d2$logdisb  + d2$logparty)
summary(fit3)
```

```
##
## Call:
## lm(formula = d2$logvotes ~ d2$logdisb + d2$logparty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3555 -0.1927  0.0741  0.2940  4.1067
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.60108    0.16135  65.704  < 2e-16 ***
## d2$logdisb   0.09767    0.01226   7.968 5.01e-15 ***
## d2$logparty -3.57205    0.10352 -34.507  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.809 on 877 degrees of freedom
## Multiple R-squared:  0.6044, Adjusted R-squared:  0.6035
## F-statistic: 669.9 on 2 and 877 DF,  p-value: < 2.2e-16
```

```
fit4 <- lm(d2$general_votes ~ d2$logdisb + d2$can_party)
summary(fit4)
```

```
##
## Call:
## lm(formula = d2$general_votes ~ d2$logdisb + d2$can_party)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -164084  -42521    2037   33117  627966
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20183.4    13565.1   1.488    0.137
## d2$logdisb    11935.7      999.7  11.939   <2e-16 ***
## d2$can_party -46716.3     3070.3 -15.215   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65620 on 877 degrees of freedom
## Multiple R-squared:  0.3354, Adjusted R-squared:  0.3339
## F-statistic: 221.3 on 2 and 877 DF,  p-value: < 2.2e-16
```

```r
attach(d2)

c1 <- lm(logvotes ~ logdisb + logparty)
summary(c1)
```

```
##
## Call:
## lm(formula = logvotes ~ logdisb + logparty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3555 -0.1927  0.0741  0.2940  4.1067
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.60108    0.16135  65.704  < 2e-16 ***
## logdisb      0.09767    0.01226   7.968 5.01e-15 ***
## logparty    -3.57205    0.10352 -34.507  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.809 on 877 degrees of freedom
## Multiple R-squared:  0.6044, Adjusted R-squared:  0.6035
## F-statistic: 669.9 on 2 and 877 DF,  p-value: < 2.2e-16
```

```r
e <- resid(c1)
c2 <- lm(e^2 ~ logdisb + logparty + I(logdisb^2) + I(logparty^2) + I(logdisb*logparty))
R2 <- summary(c2)$r.sq
n <- nrow(c2$model)
m <- ncol(c2$model)
W <- n*R2
P <- 1 - pchisq(W, m - 1)
c3 <- lm(logvotes ~  logdisb + logparty, weights = 1/abs(e))
summary(c3)
```

```
##
## Call:
## lm(formula = logvotes ~ logdisb + logparty, weights = 1/abs(e))
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3152 -0.4577  0.2486  0.5280  2.0255
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.597863   0.035366  299.67   <2e-16 ***
## logdisb      0.098488   0.002631   37.44   <2e-16 ***
## logparty    -3.579191   0.023330 -153.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6811 on 877 degrees of freedom
```

```
## Multiple R-squared:  0.9644, Adjusted R-squared:  0.9643
## F-statistic: 1.188e+04 on 2 and 877 DF,  p-value: < 2.2e-16
```