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.

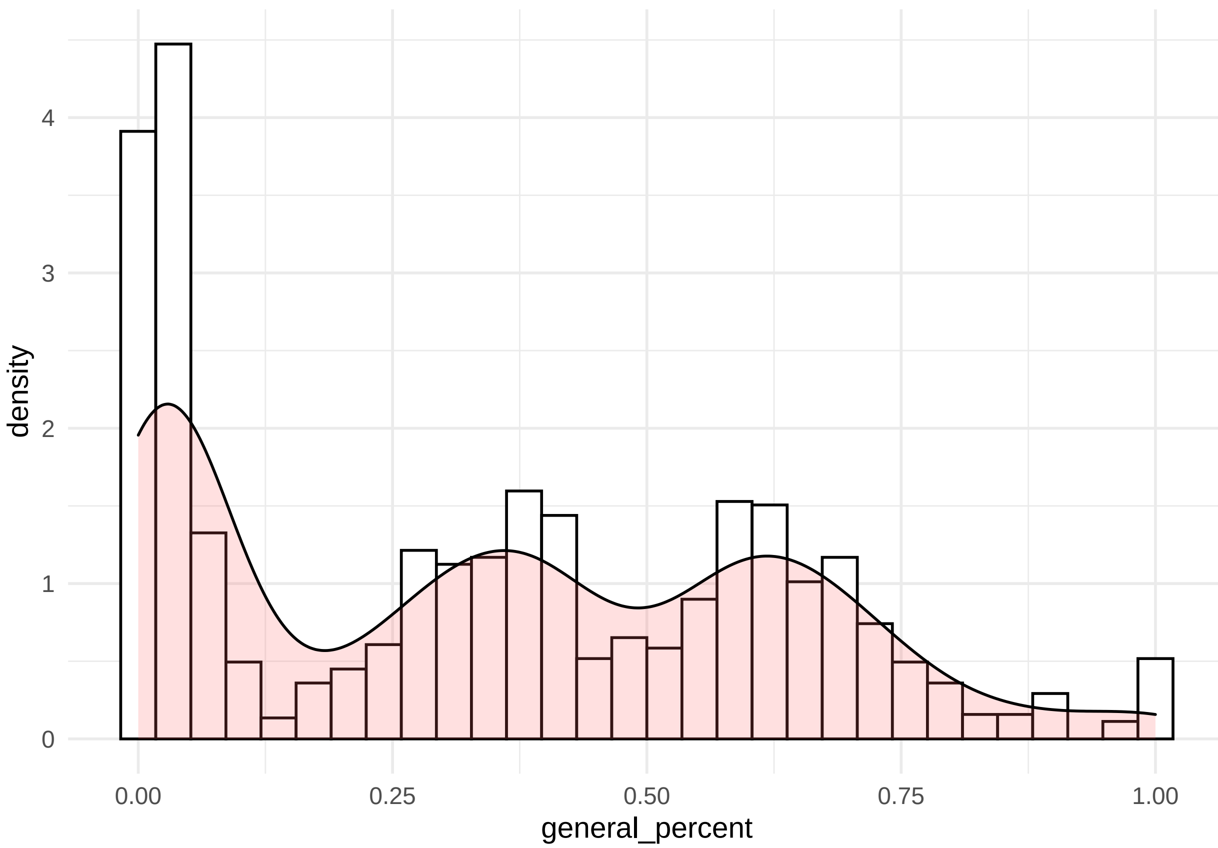
Your work

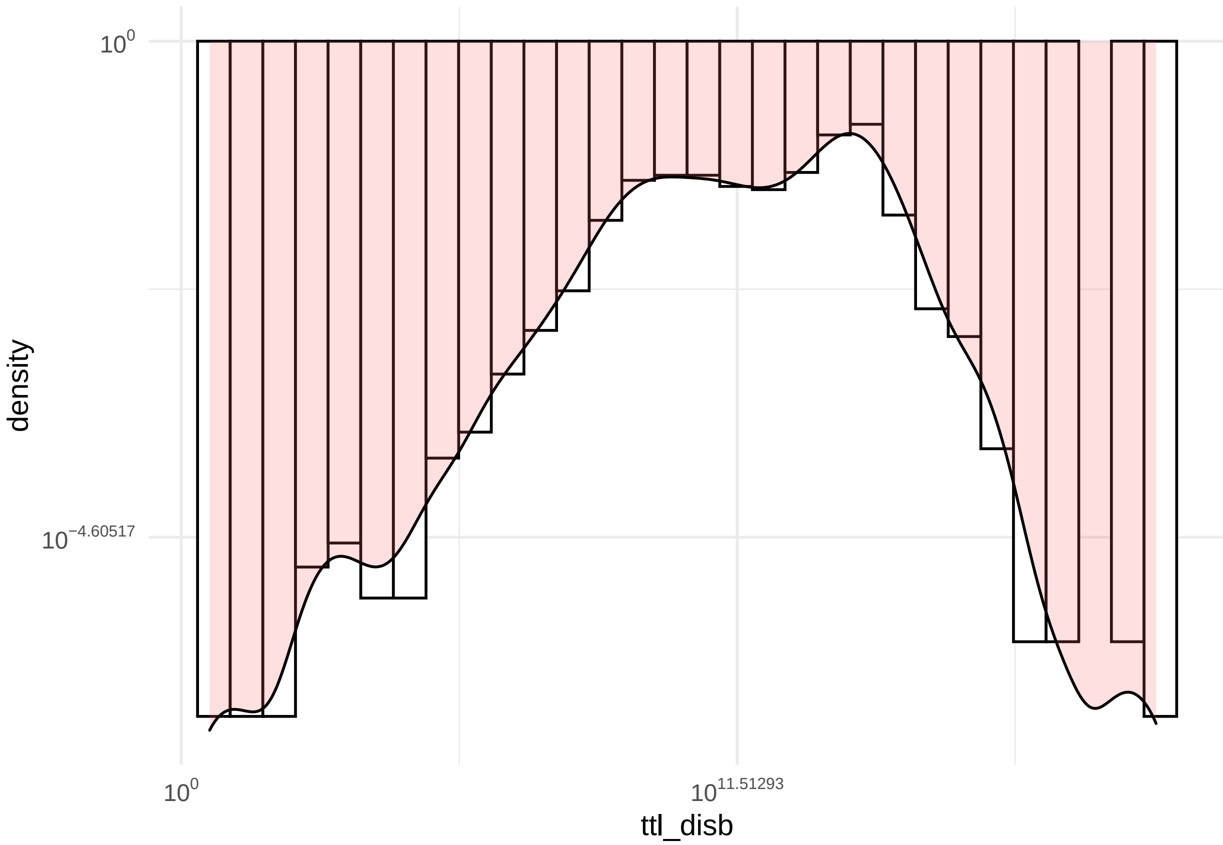
- We want to keep this work *relatively* constrained, which is why we're providing you with data through the `fec16` package. It is possible to gather all the information from current FEC reports, but it would require you to make a series of API calls that would pull us away from the core modeling tasks 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.

```
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)
```

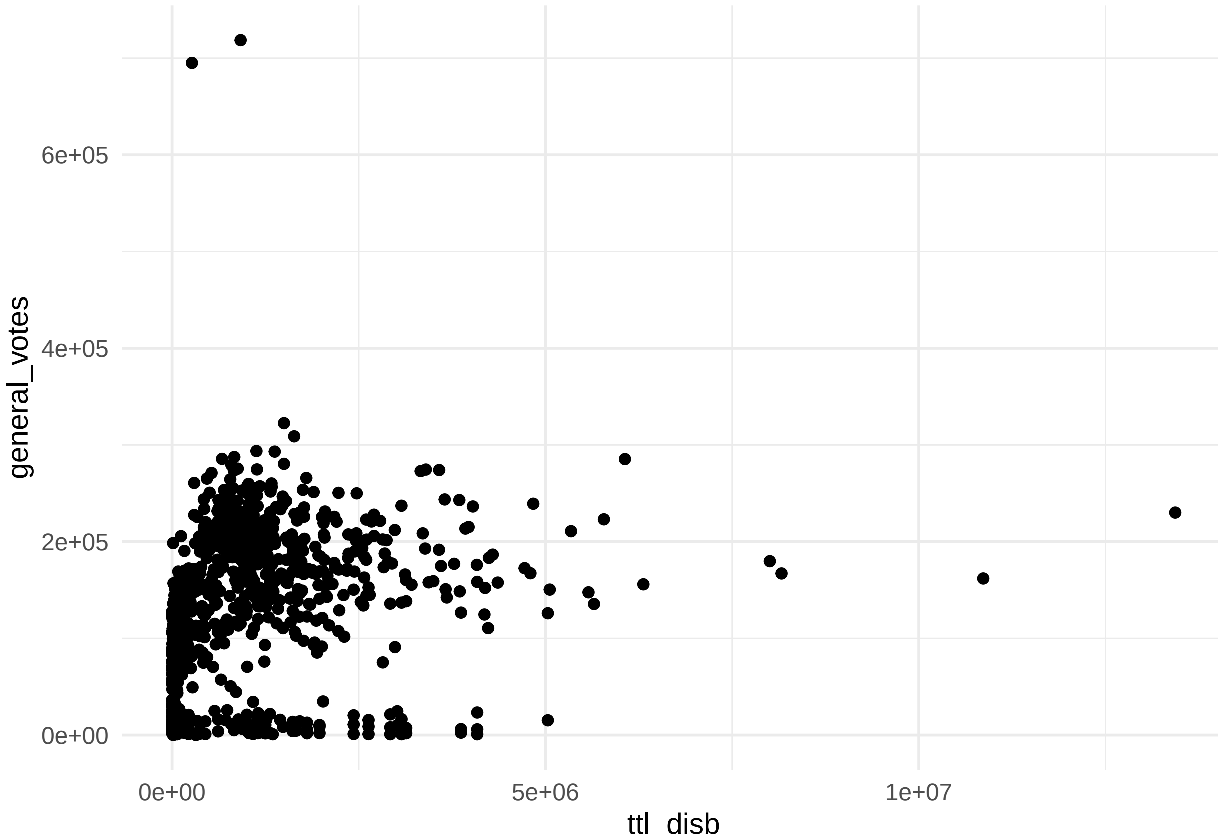
```
## Joining, by = "cand_id"
```

```
d1[d1 == -Inf] <- 0
#nrow(d1)
#summary(d1)
#write.csv(d1, "d1.csv")
```

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?

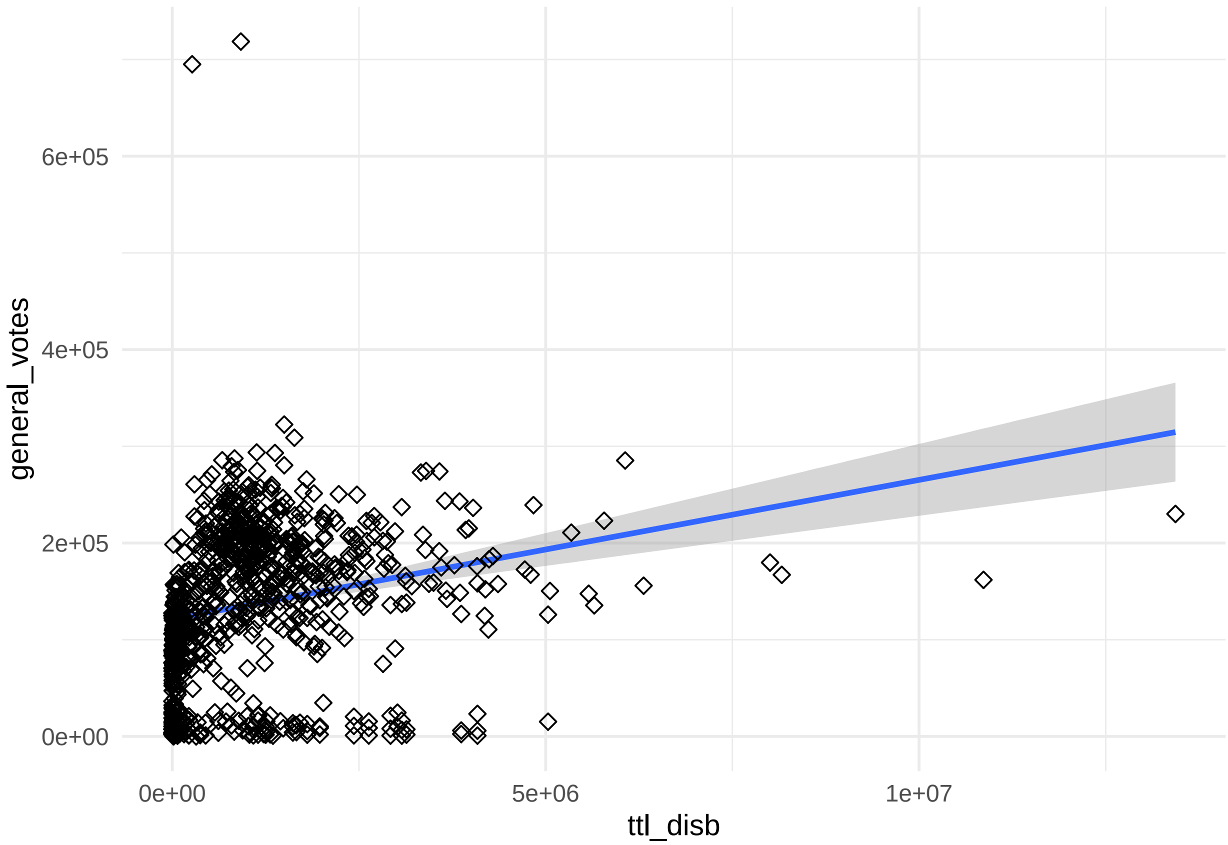
```
ggplot(d1, aes(y=general_votes, x=ttl_disb)) + geom_point()
```

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




```
sp <- ggplot(d1, aes(y=general_votes, x=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"
    )
  )
```

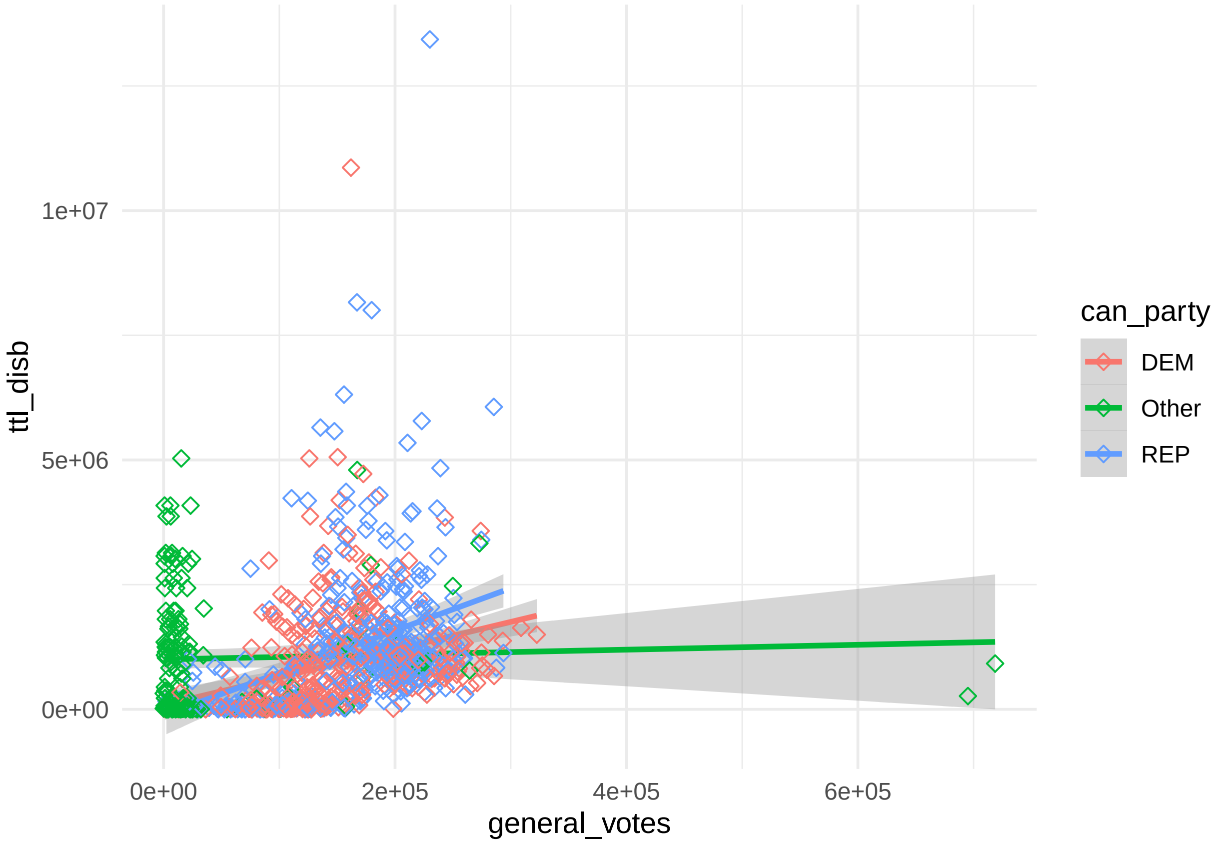
Once you've produced the new variable, plot your scatter plot again, but this time adding an argument into the `aes()` function that colors the points by party membership. What do you observe about the distribution of all three variables?

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

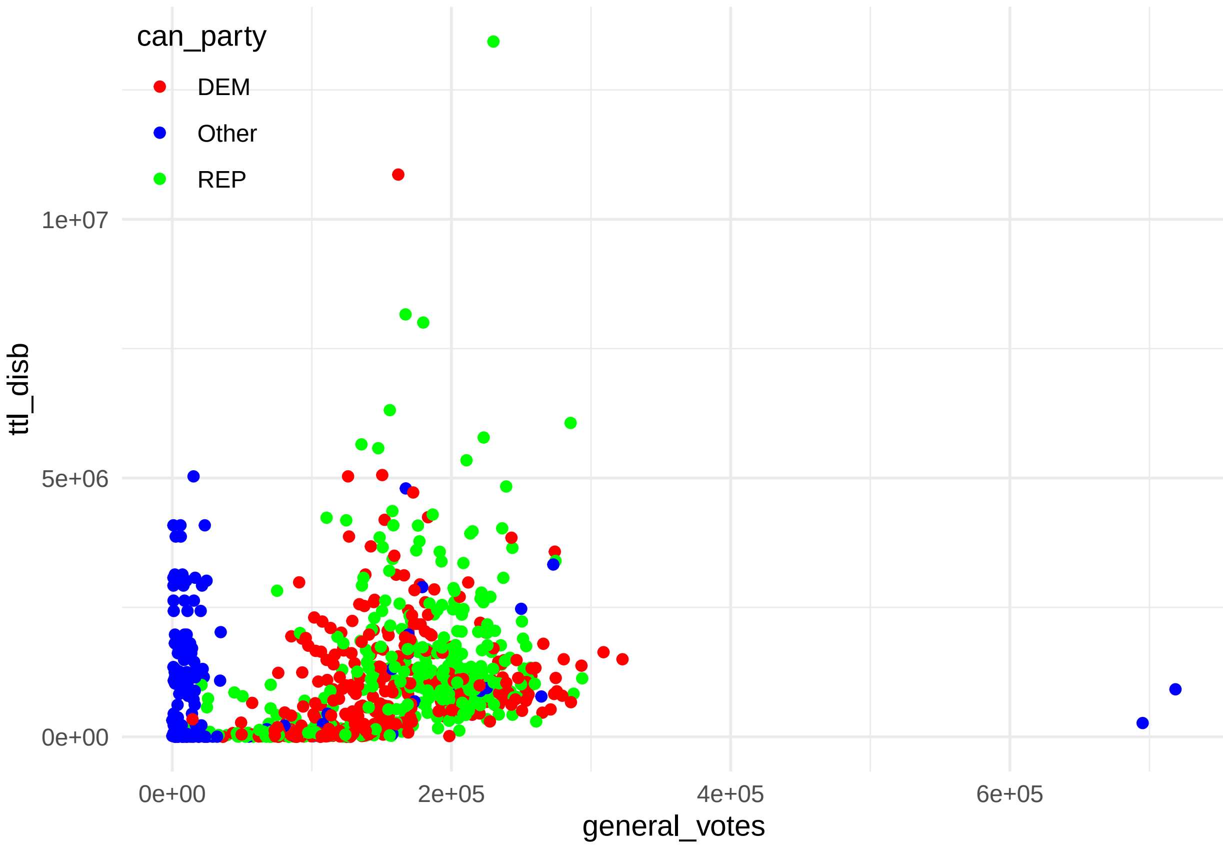
```
      party=="DEM" ~ "DEM",
      TRUE ~ "Other"
    )
  )

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
```

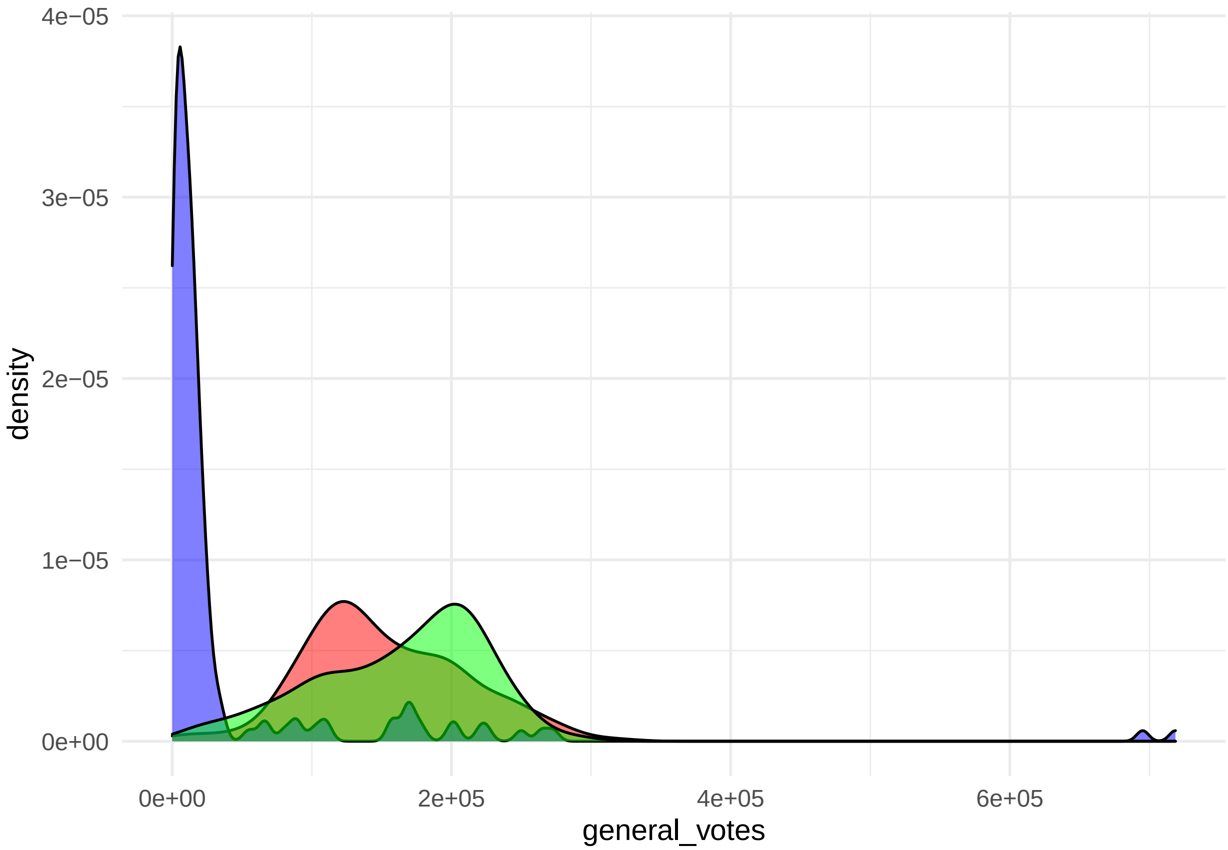
```
## `geom_smooth()` using formula 'y ~ x'
```



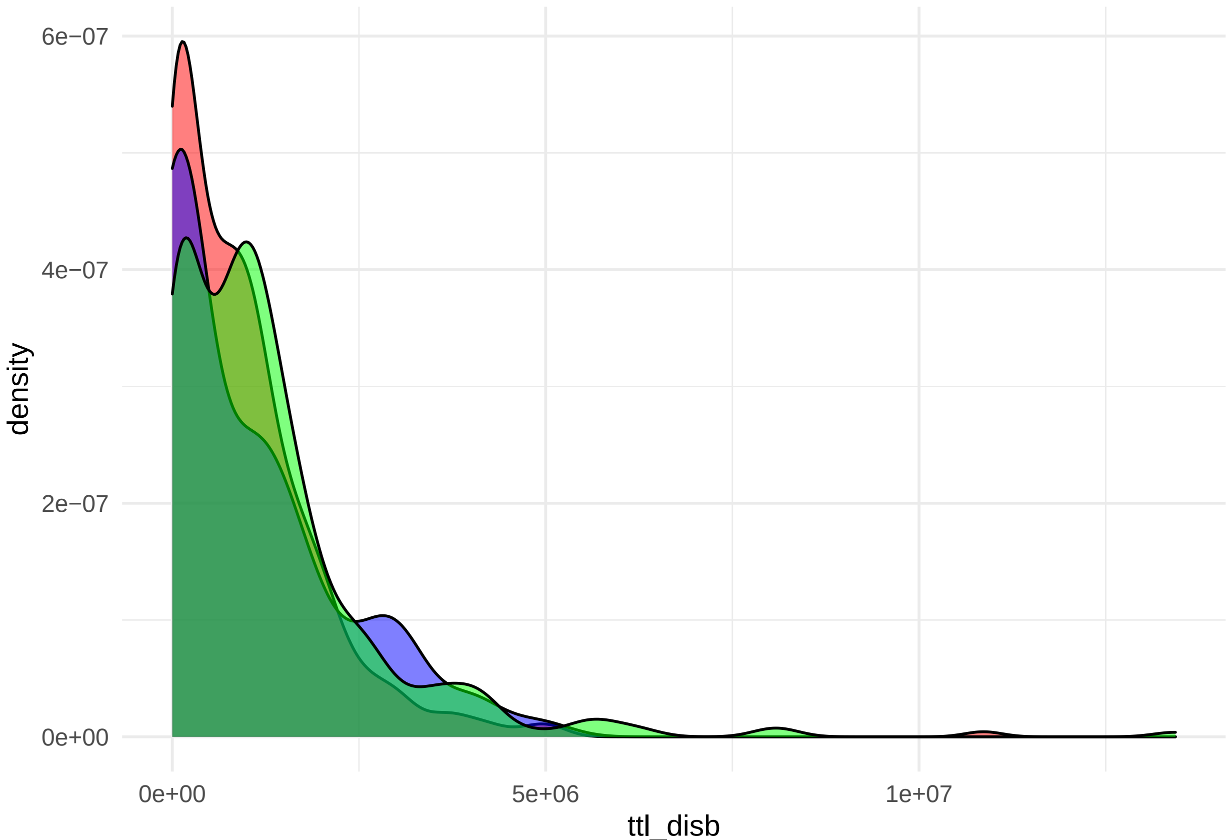

```
p1<-ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_point() +
  scale_color_manual(values = c("red", "blue", "green")) +
  theme(legend.position=c(0,1), legend.justification=c(0,1))
p1
```



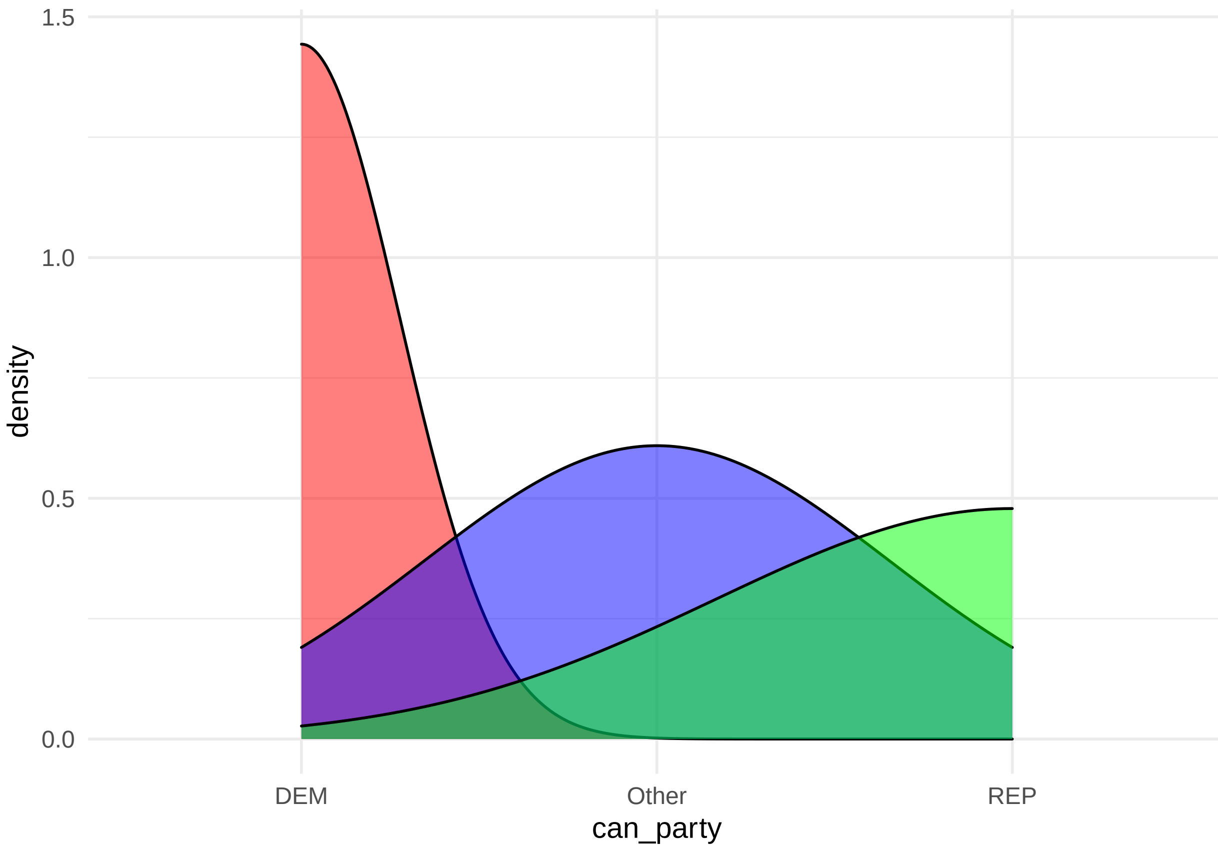

```r
p2<-ggplot(d2, aes(x=general_votes, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values =  c("red", "blue", "green")) +
  theme(legend.position = "none")
p2
```




```
# 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.

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


sdat <- lm(general_votes ~  ttl_disb + can_party + state, data = d2 )
bptest(sdat)
```

```
##
```

```
##  studentized Breusch-Pagan test
##
## data:  sdat
## BP = 472.23, df = 57, p-value < 2.2e-16
```

```r
#ncvTest(fit)

attach(d2)

c1 <- lm(general_votes ~  ttl_disb + can_party)
summary(c1)
```

```
##
## Call:
## lm(formula = general_votes ~ ttl_disb + 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 ***
## ttl_disb     1.163e-02  1.864e-03   6.238 6.88e-10 ***
## 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
e <- resid(c1)
c2 <- lm(e^2 ~ ttl_disb + can_party + I(ttl_disb^2) + I(can_party^2) + I(ttl_disb*can_party))
(R2 <- summary(c2)$r.sq)
```

```
## [1] 0.01738939
```

```r
(n <- nrow(c2$model))
```

```
## [1] 880
```

```r
(m <- ncol(c2$model))
```

```
## [1] 6
```

```r
(W <- n*R2)
```

```
## [1] 15.30266
```

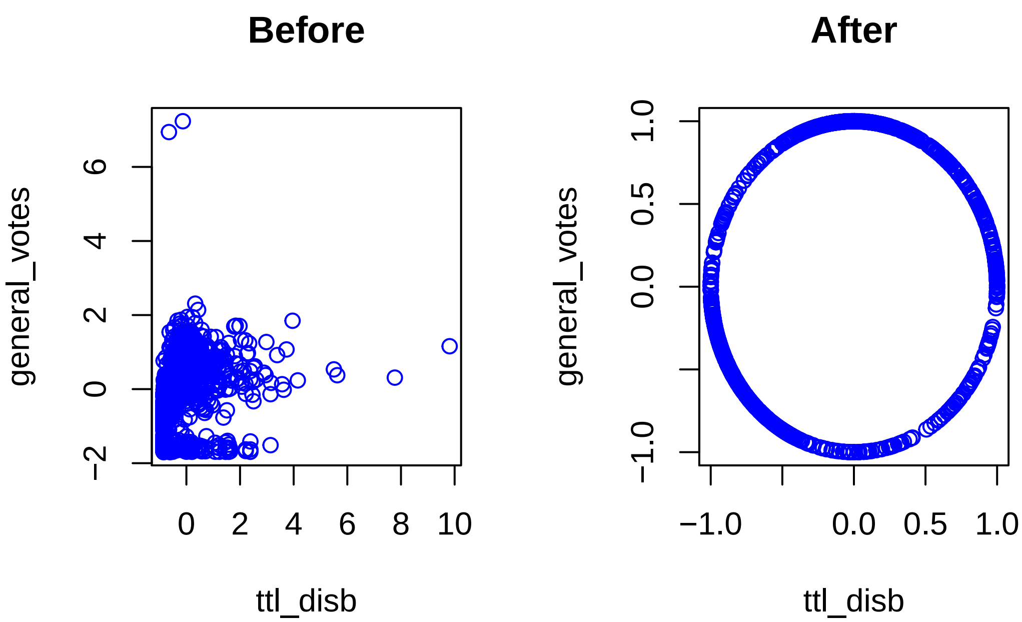
```r
(P <- 1 - pchisq(W, m - 1))
```

```
## [1] 0.009144436
```

```r
c3 <- lm(general_votes ~  ttl_disb + can_party, weights = 1/abs(e))

summary(c3)
```

```
##
## Call:
## lm(formula = general_votes ~ ttl_disb + can_party, weights = 1/abs(e))
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -404.99 -214.86   -1.48  194.43  808.03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.641e+05  1.112e+03  147.50   <2e-16 ***
## ttl_disb     1.155e-02  7.035e-04   16.41   <2e-16 ***
## can_party   -5.268e+04  1.149e+03  -45.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 226.1 on 877 degrees of freedom
## Multiple R-squared:  0.7956, Adjusted R-squared:  0.7951
## F-statistic:  1707 on 2 and 877 DF,  p-value: < 2.2e-16
```

```r
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")
```




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

summary(d2)
```

```
##    can_party      general_votes       ttl_disb           state
##  Min.   :0.0000   Min.   :    55   Min.   :       0   Length:880
##  1st Qu.:0.0000   1st Qu.: 88229   1st Qu.:  102276   Class :character
##  Median :1.0000   Median :142597   Median :  830659   Mode  :character
##  Mean   :0.7727   Mean   :136932   Mean   : 1084565
##  3rd Qu.:1.0000   3rd Qu.:198290   3rd Qu.: 1527533
##  Max.   :2.0000   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
```

```
#d2<-transformed
```

```
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
##  Min.   :0.0000   Min.   :    55   Min.   :       0   Length:880
##  1st Qu.:0.0000   1st Qu.: 88229   1st Qu.:  102276   Class :character
##  Median :1.0000   Median :142597   Median :  830659   Mode  :character
##  Mean   :0.7727   Mean   :136932   Mean   : 1084565
##  3rd Qu.:1.0000   3rd Qu.:198290   3rd Qu.: 1527533
##  Max.   :2.0000   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$logparty <- log(d2$can_party)
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)
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
#only no outlier data R2 = 0.4055
#fit1 <- lm(d2$novotes ~ d2$nodisb + d2$state + d2$can_party)
fit1 <- lm(d2$novotes ~ d2$nodisb  + d2$can_party)
summary(fit1)
```

```
##
## Call:
## lm(formula = d2$novotes ~ d2$nodisb + d2$can_party)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44064 -0.49643 -0.07907  0.54617  1.46145
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.45438    0.03129  14.521   <2e-16 ***
## d2$nodisb     0.27807    0.03266   8.515   <2e-16 ***
## d2$can_party -0.44263    0.02939 -15.062   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6349 on 877 degrees of freedom
## Multiple R-squared:  0.2653, Adjusted R-squared:  0.2636
## F-statistic: 158.3 on 2 and 877 DF,  p-value: < 2.2e-16
```

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

```
##
## 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
```

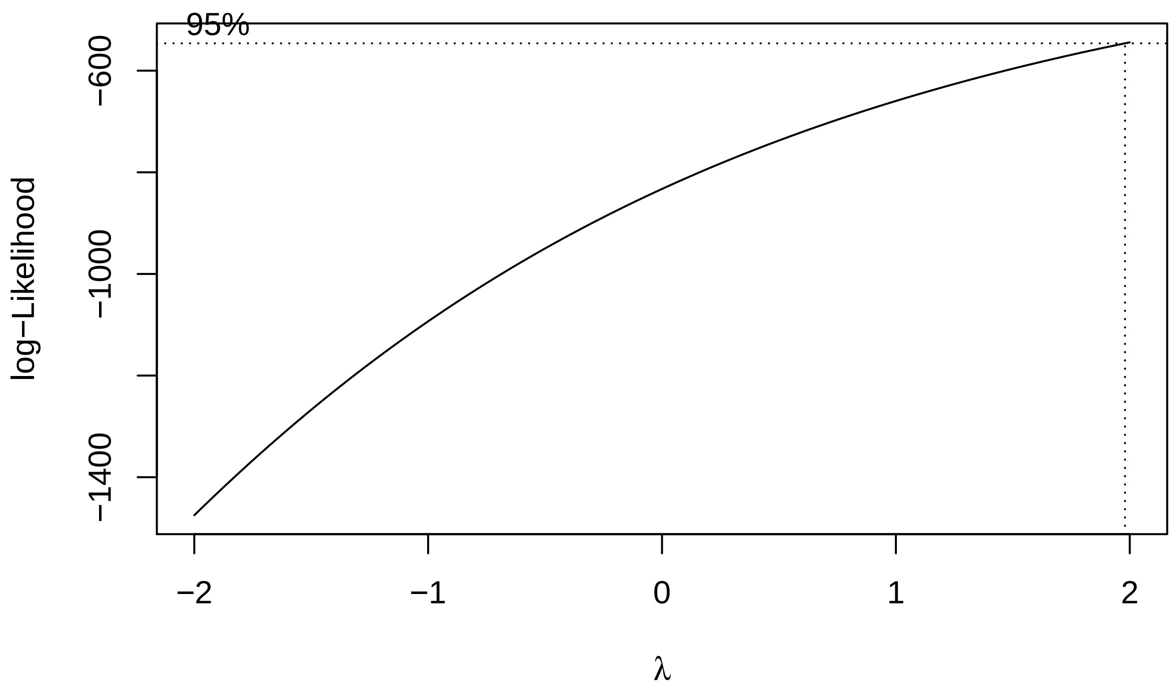
```r
#only original, log(spending) data R2 = 0.6041
#fit3 <- lm(d2$general_votes ~ d2$logdisb + d2$state + d2$can_party)
fit3 <- lm(d2$general_votes ~ d2$logdisb + d2$can_party)
summary(fit3)
```

```
##
## 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
#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
```

```r
b <- boxcox(logvotes ~ logdisb + logparty, 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 -1474.865
## [2,] -1.959596 -1456.833
```

```
lambda<- 2.4
d2$lamvotes <- (d2$logvotes^lambda-1)/lambda

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

```
##
## Call:
## lm(formula = lamvotes ~ logdisb + logparty, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.727   -6.848    1.412    8.771  119.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  118.737      4.067  29.192   <2e-16 ***
## logdisb        3.029      0.309   9.802   <2e-16 ***
```

```
## logparty     -91.757      2.610 -35.162   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.39 on 877 degrees of freedom
## Multiple R-squared:  0.6209, Adjusted R-squared:   0.62
## F-statistic: 718.1 on 2 and 877 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = lamvotes ~ logdisb + logparty, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.727   -6.848    1.412    8.771  119.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  118.737      4.067  29.192   <2e-16 ***
## logdisb        3.029      0.309   9.802   <2e-16 ***
## logparty     -91.757      2.610 -35.162   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.39 on 877 degrees of freedom
## Multiple R-squared:  0.6209, Adjusted R-squared:   0.62
## F-statistic: 718.1 on 2 and 877 DF,  p-value: < 2.2e-16
```

```
#bptest(sdat)

#ncvTest(fit)

attach(d2)
```

```
## The following objects are masked from d2 (pos = 4):
##
##     can_party, general_votes, state, ttl_disb
```

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

```
##
## Call:
## lm(formula = lamvotes ~ logdisb + logparty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.727   -6.848    1.412    8.771  119.849
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  118.737      4.067  29.192   <2e-16 ***
## logdisb        3.029      0.309   9.802   <2e-16 ***
## logparty     -91.757      2.610 -35.162   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.39 on 877 degrees of freedom
## Multiple R-squared:  0.6209, Adjusted R-squared:   0.62
## F-statistic: 718.1 on 2 and 877 DF,  p-value: < 2.2e-16
```

```
e <- resid(c1)
c2 <- lm(e^2 ~ logdisb + logparty + I(logdisb^2) + I(logparty^2) + I(logdisb*logparty))
(R2 <- summary(c2)$r.sq)
```

```
## [1] 0.1645108
```

```
(n <- nrow(c2$model))
```

```
## [1] 880
```

```
(m <- ncol(c2$model))
```

```
## [1] 6
```

```
(W <- n*R2)
```

```
## [1] 144.7695
```

```
(P <- 1 - pchisq(W, m - 1))
```

```
## [1] 0
```

```
c3 <- lm(lamvotes ~  logdisb + logparty, weights = 1/abs(e))

summary(c3)
```

```
##
## Call:
## lm(formula = lamvotes ~ logdisb + logparty, weights = 1/abs(e))
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -10.184  -2.701   1.057   2.873  10.958
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 118.10513    0.82784  142.67   <2e-16 ***
## logdisb       3.09811    0.06612   46.85   <2e-16 ***
```

```
## logparty    -92.24965    0.34882 -264.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.571 on 877 degrees of freedom
## Multiple R-squared:  0.9881, Adjusted R-squared:  0.9881
## F-statistic: 3.635e+04 on 2 and 877 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.