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`.

```r
## 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.

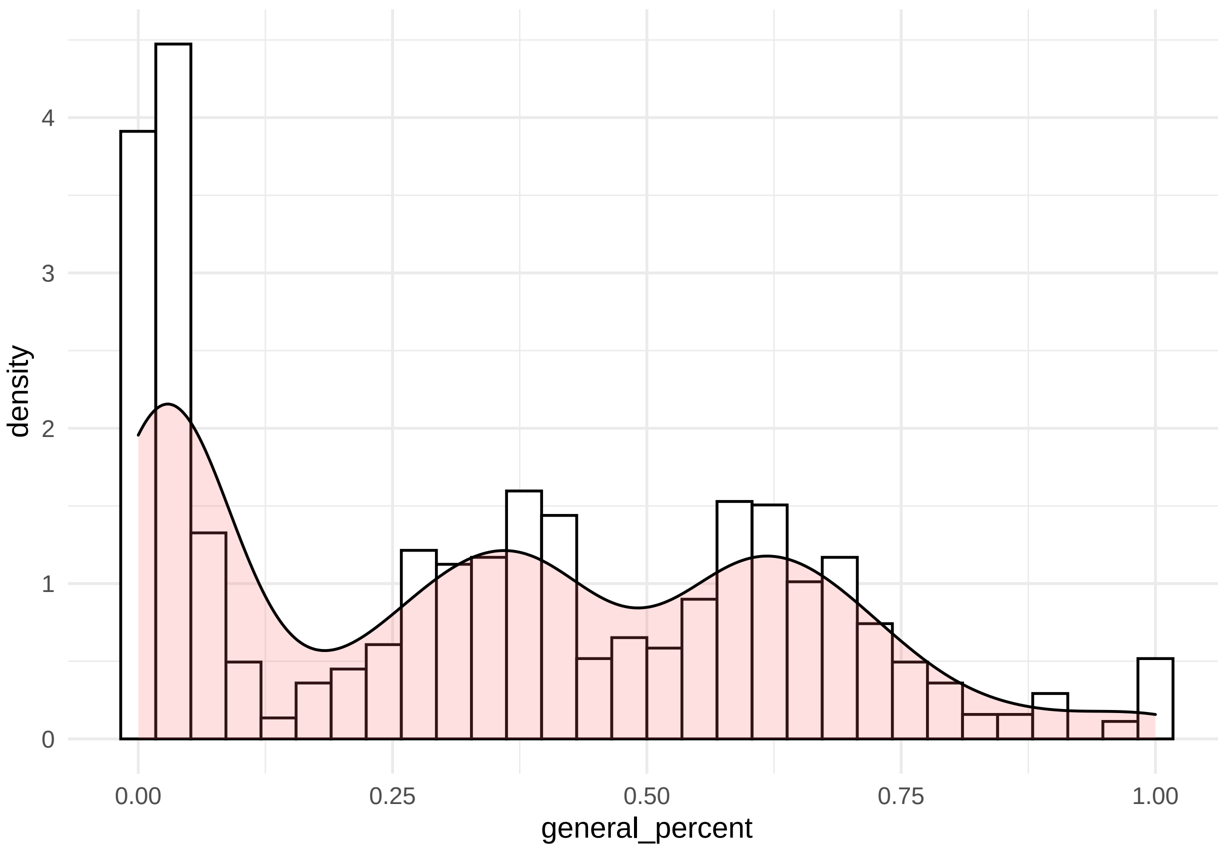
Your work

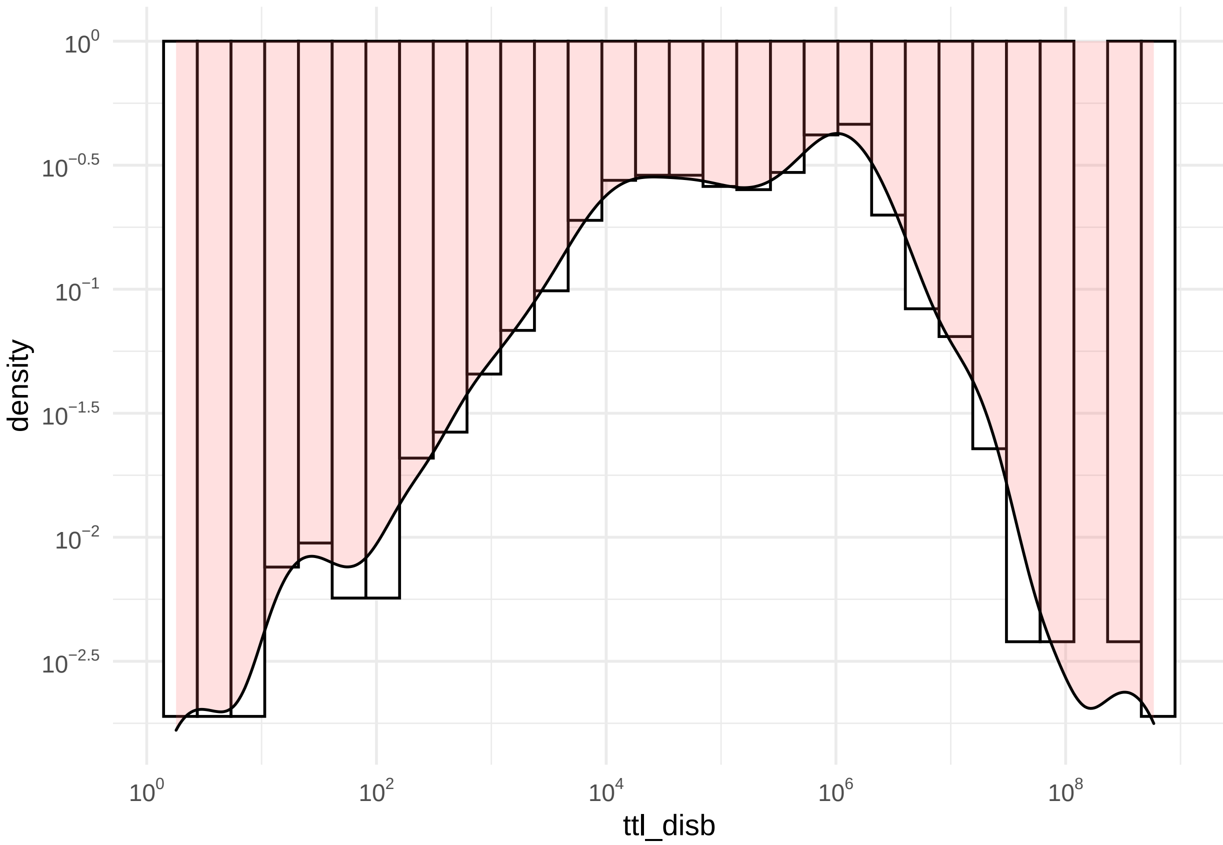
- We want to keep this work *relatively* constrained, which is why we're providing you with data through the `fec16` package. It is possible to gather all the information from current FEC reports, but it would require you to make a series of API calls that would pull us away from the core modeling tasks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

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

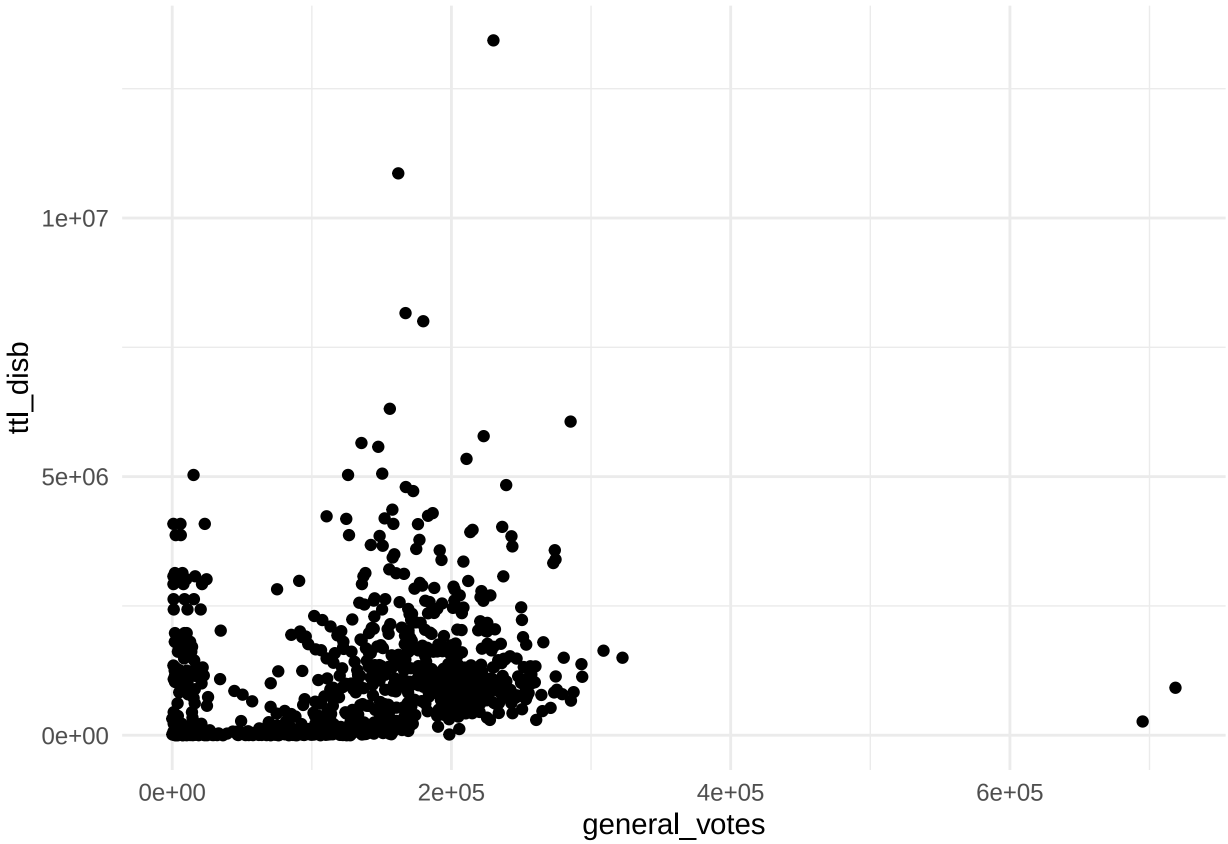
```
nrow(d1)
```

```
## [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).
```

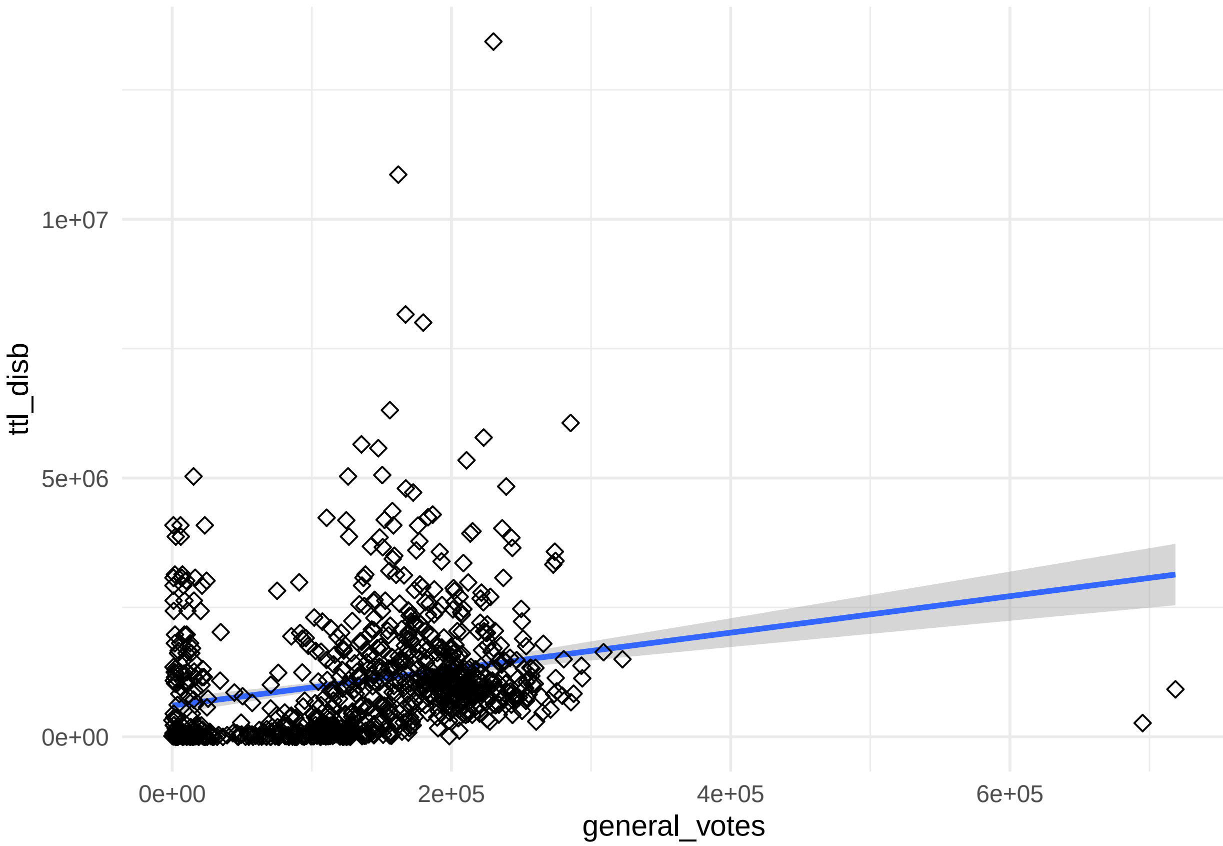



```r
sp <- ggplot(d1, aes(x=general_votes, y=ttl_disb  )) +
  geom_smooth(method=lm)+
  geom_point(size=2, shape=23)

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?

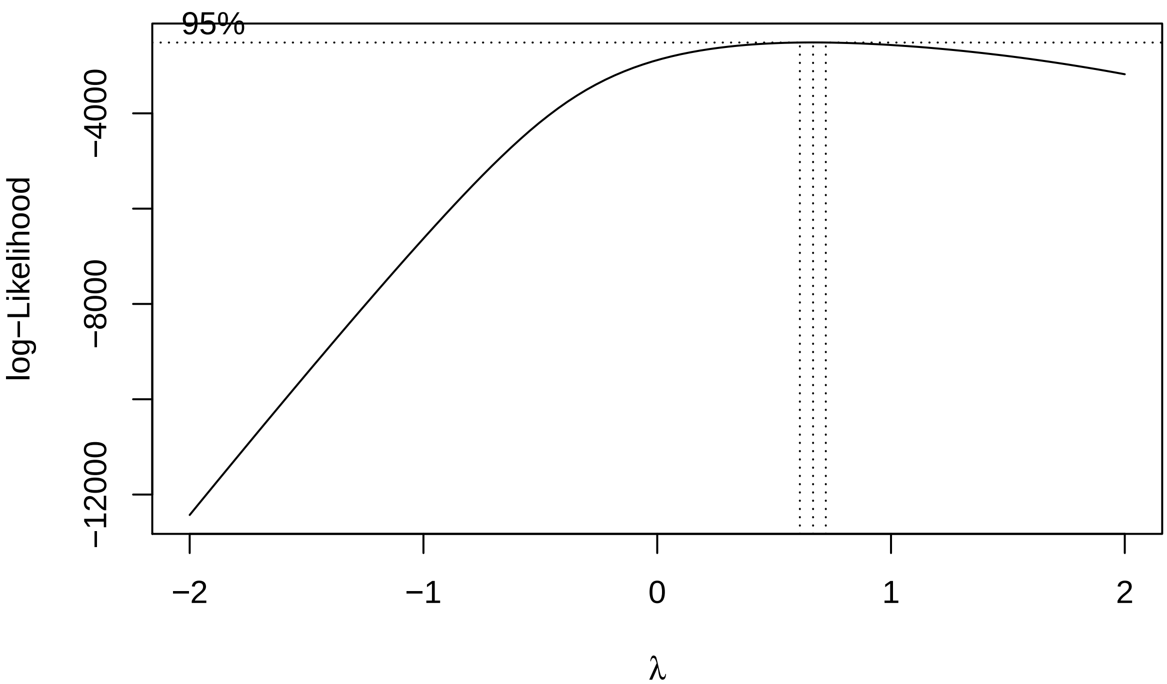
```
d2<-d1 %>%
  dplyr::select(cand_pty_affiliation, general_votes, ttl_disb, state) %>%
  na.omit() %>%
    mutate(
    can_party = case_when(
      cand_pty_affiliation=="REP" ~ "REP",
```

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

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

```
#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 ~ ttl_disb + state + can_party, data = d2)
```




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

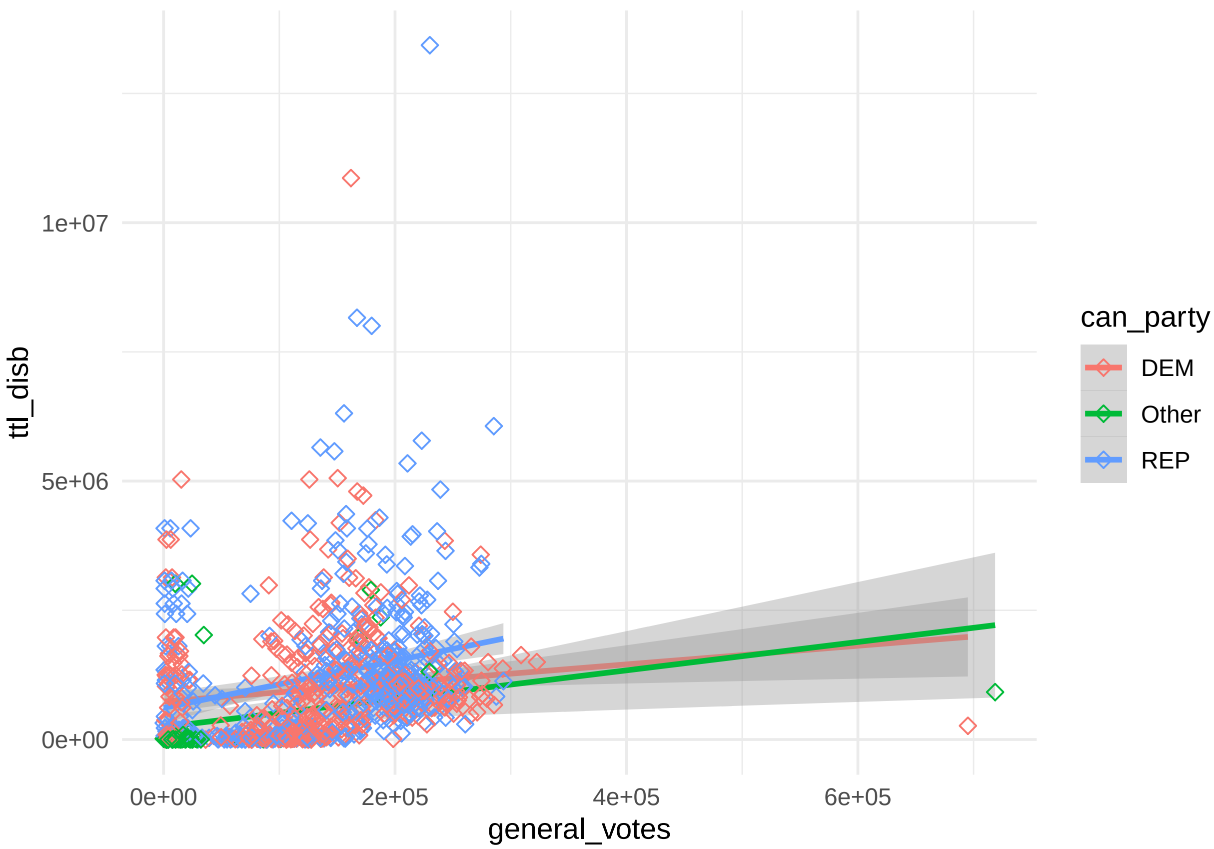
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'
```

```
##          lambda       lik
## [1,] -2.000000 -12428.47
## [2,] -1.959596 -12186.09
```

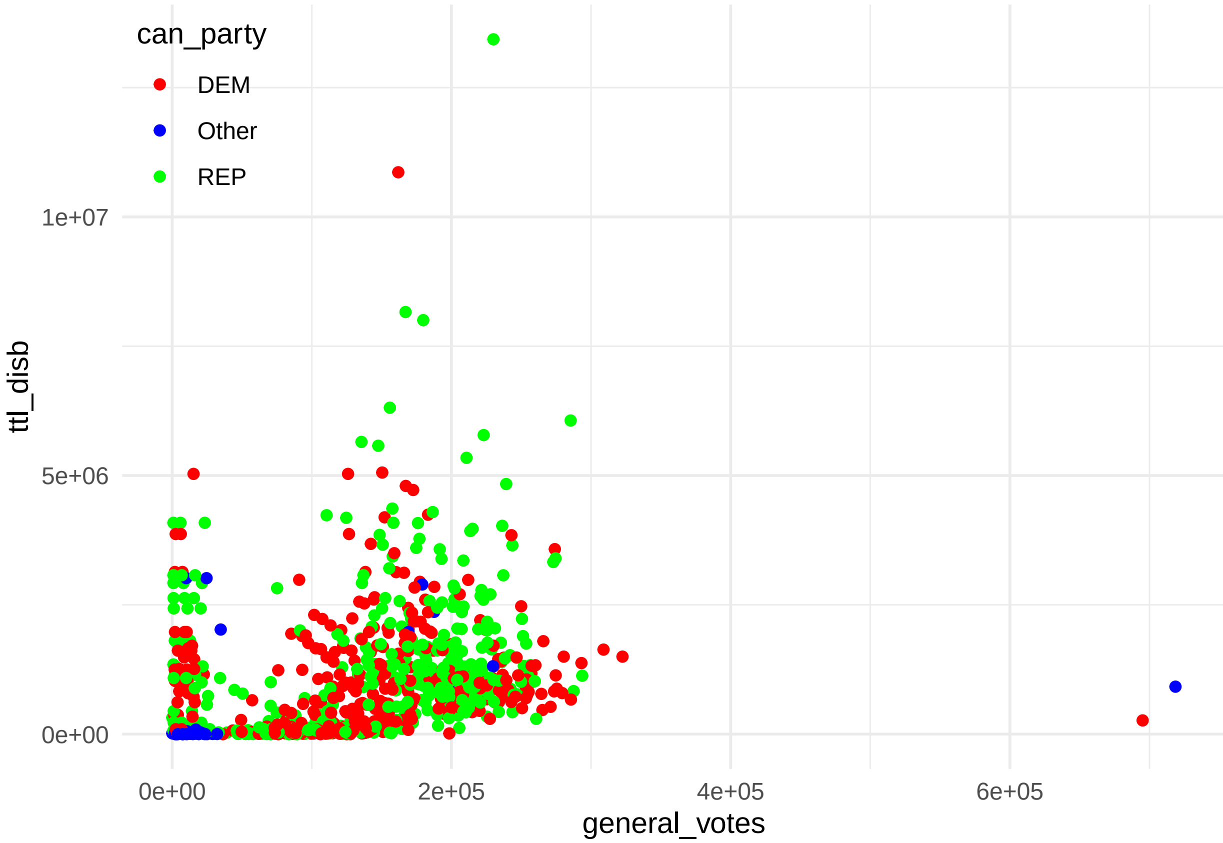
```
lambda<- 0.67
d2$lamvotes <- (d2$general_votes^lambda-1)/lambda


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

```
#d2$state <- as.numeric(d2$state)
#d2$can_party <- as.numeric(d2$can_party)

#d2

#write.csv(d2, "d2.csv")

#head(d2)

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

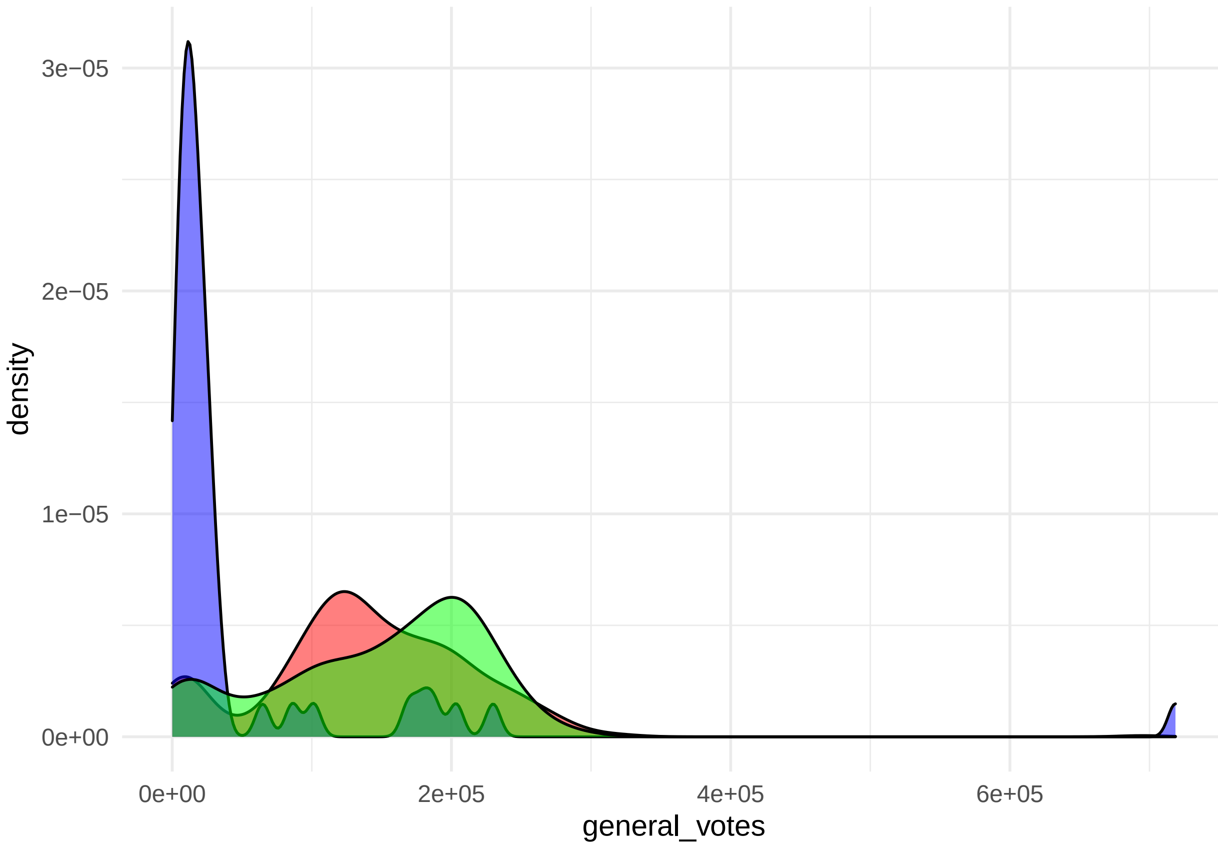
```
## `geom_smooth()` using formula 'y ~ x'
```



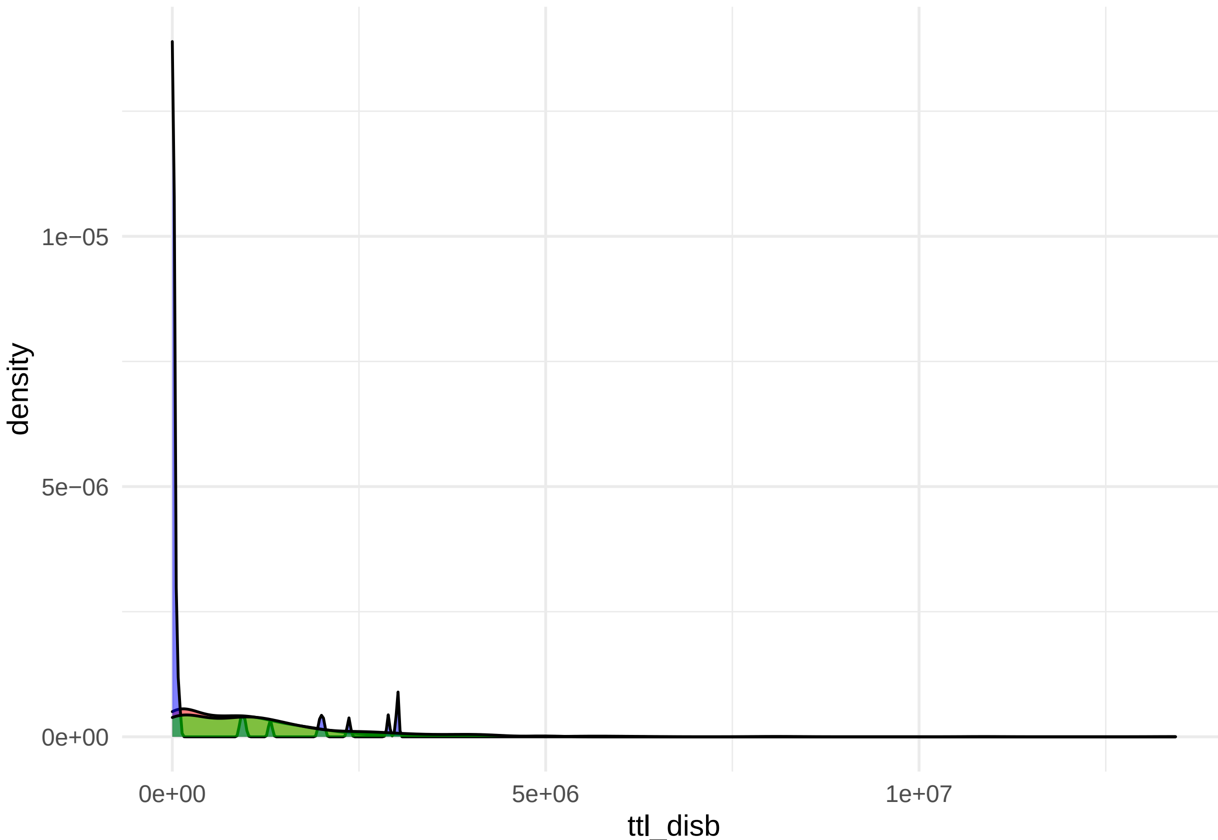
```
p1<-ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_point() +
  scale_color_manual(values = c("red", "blue", "green")) +
  theme(legend.position=c(0,1), legend.justification=c(0,1))
p1
```



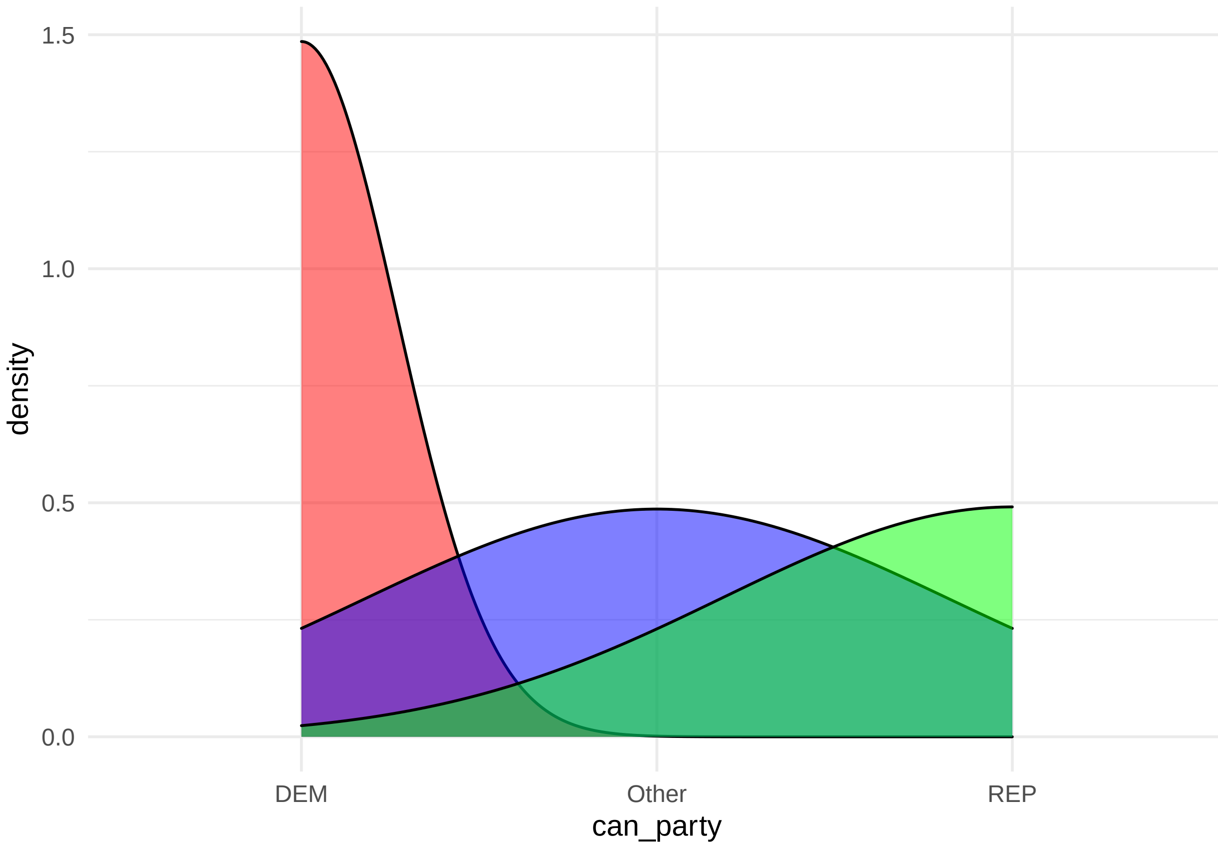

```
p2<-ggplot(d2, aes(x=general_votes, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values =  c("red", "blue", "green")) +
  theme(legend.position = "none")
p2
```




```r
# 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
```

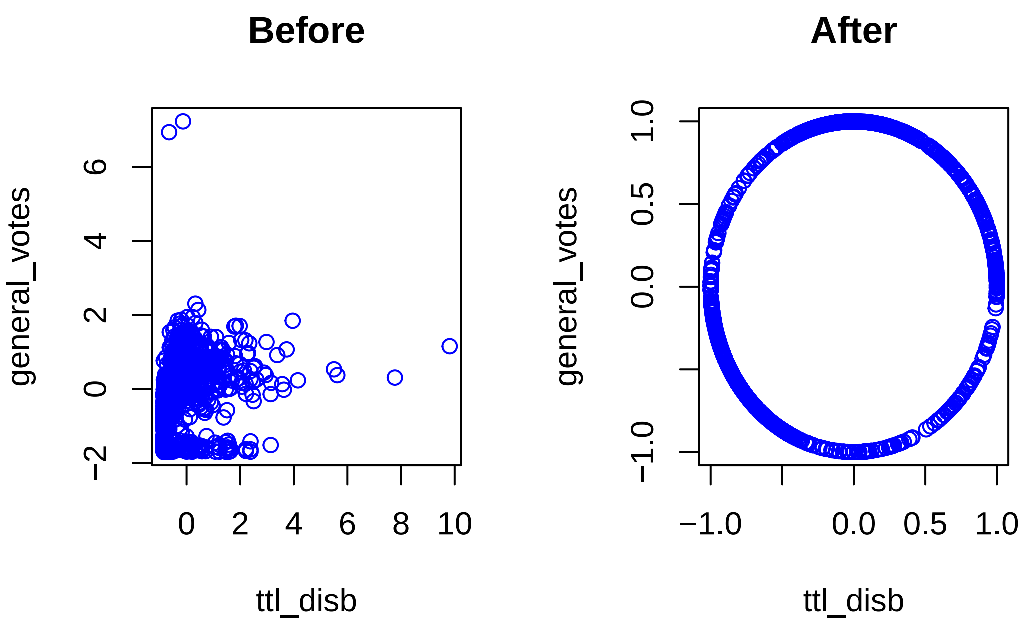



```
#sp + geom_density_2d()


#summary(d1)
```

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")
```




```
d2$tvotes<-transformed$"general_votes"
d2$tdisb<-transformed$"ttl_disb"

#summary(d2)

#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)

#summary(d3)

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

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


d3$logdisb <- log(d3$tdisb)
```

```
## Warning in log(d3$tdisb): NaNs produced
```

```
d3$logvotes <- log(d3$tvotes)
```

```
## Warning in log(d3$tvotes): NaNs produced
```

```r
#d3 <- na.omit(d3)


#d2[which(!is.finite(d2))] <- 0
#d2 <- d2[is.finite(rowSums(d2)),]
#d2[d2 == -Inf] <- 0
#d3[d3 == -Inf] <- 0
#data_new <- d2                                       # Duplicate data
#d2[is.na(d2$disb) | d2$disb == "Inf"] <- NA  # Replace NaN & Inf with NA
#d3 <- data_new
#head(d2)
#head(d2$disb)
#d3<-d3%>%na.omit()

#only center and scale R2 = 0.5116
fit0 <- lm(d3$general_votes ~ d3$ttl_disb + d3$state + d3$can_party)
#summary(fit0)

#only original data R2 = 0.5116
fit1 <- lm(d3$tvotes ~ d3$tdisb + d3$state + d3$can_party)
#summary(fit1)

#only original, log(spending) data R2 = 0.6041
fit2 <- lm(d3$logvotes ~ d3$logdisb + d3$state + d3$can_party)
#summary(fit2)


#only original, log(spending) data R2 = 0.6173
fit3 <- lm(d3$tvotes ~ d3$logdisb + d3$state + d3$can_party)
#summary(fit3)
```

```r
d2$disb <- log(d2$ttl_disb)
d2$votes <- log(d2$general_votes)

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

#d2[which(!is.finite(d2))] <- 0
#d2 <- d2[is.finite(rowSums(d2)),]
d2[d2 == -Inf] <- 0

#data_new <- d2                                       # Duplicate data

#d2[is.na(d2$disb) | d2$disb == "Inf"] <- NA  # Replace NaN & Inf with NA

#d3 <- data_new

head(d2)
```

```
## # A tibble: 6 x 9
##   can_party general_votes ttl_disb state lamvotes  tvotes   tdisb  disb votes
##   <chr>             <dbl>    <dbl> <chr>    <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1 REP              208083 1172750. AL       5458.  0.997   0.0789 14.0   12.2
## 2 REP              134886 1850536. AL       4082. -0.0418  0.999  14.4   11.8
```

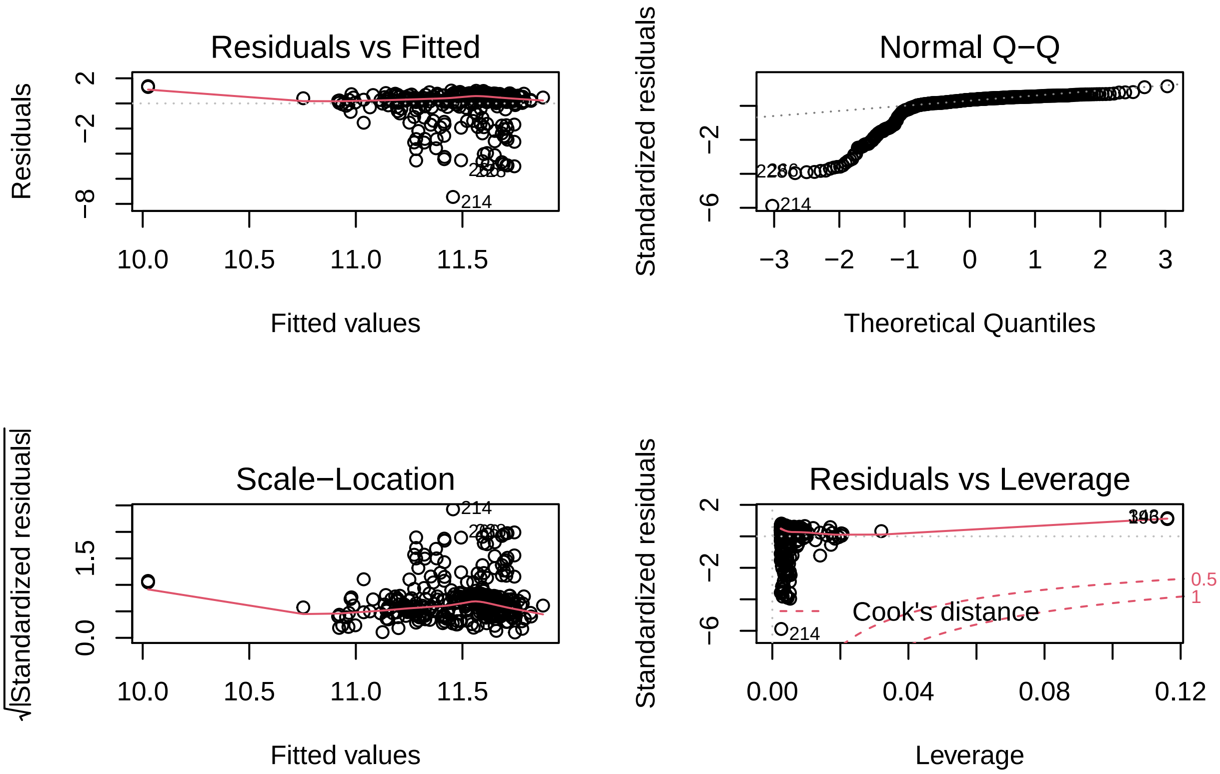
```
## 3 DEM              112089   36844   AL       3605. -0.348  -0.938  10.5   11.6
## 4 REP              192164 1071289. AL       5174.  1.00   -0.0154 13.9   12.2
## 5 DEM               94549    7348   AL       3217. -0.524  -0.852   8.90  11.5
## 6 REP              235925 1394461. AL       5937.  0.981   0.196  14.1   12.4
```

```
head(d2$disb)
```

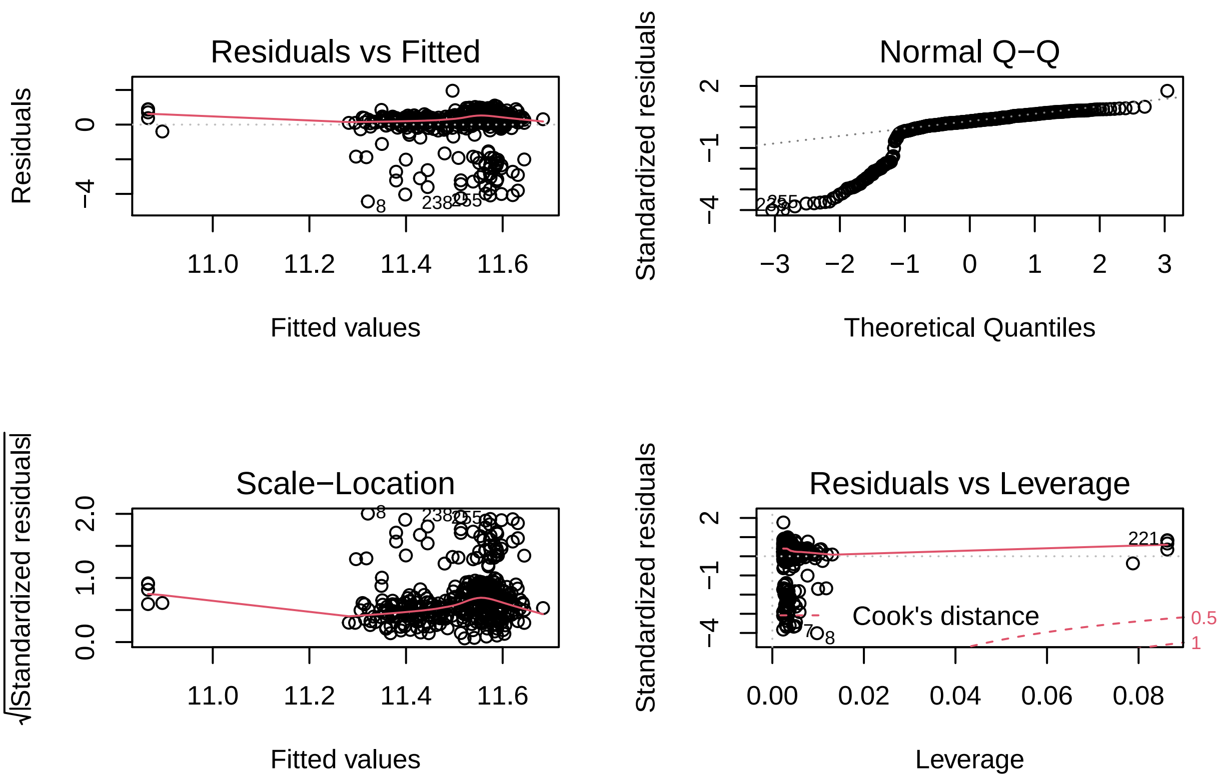
```
## [1] 13.974862 14.430986 10.514448 13.884374  8.902183 14.148019
```

```
#d3<-d3%>%na.omit()

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

fit1 <- lm(d2$general_votes ~ d2$disb + d2$state + d2$can_party)
#summary(fit1)
```

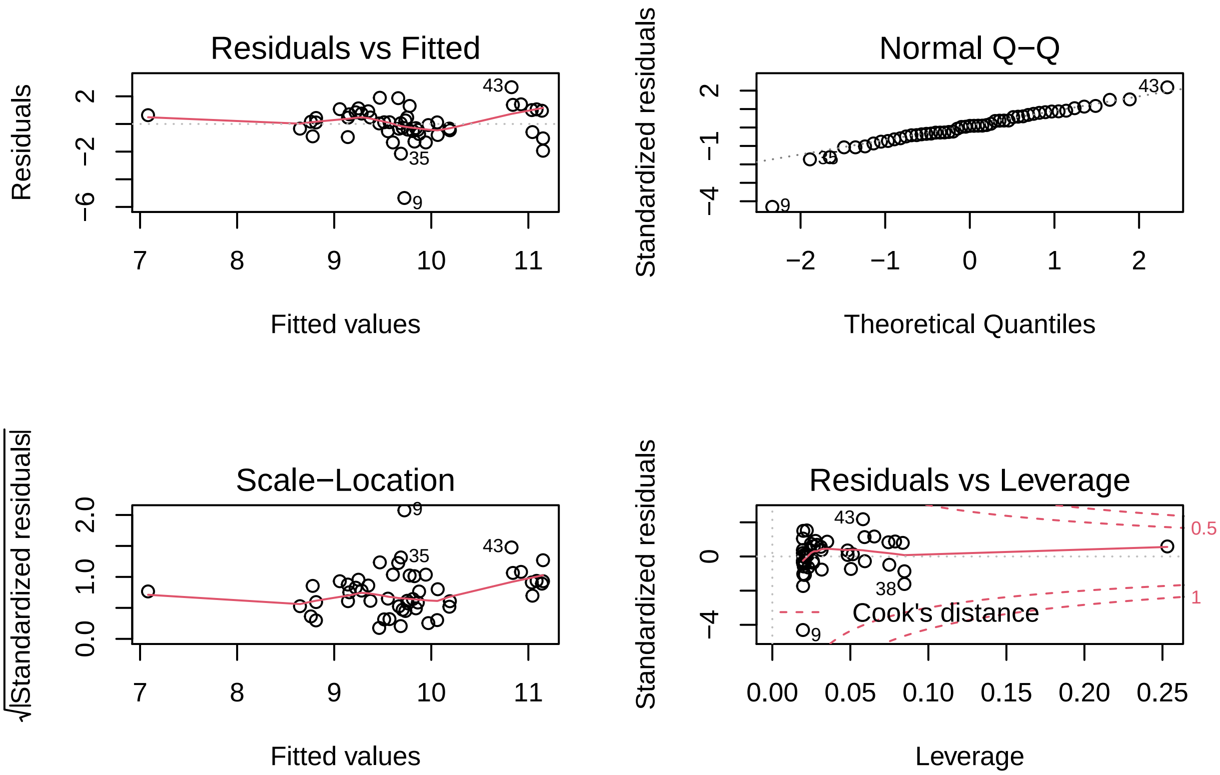
```
## boxcox test
#library(MASS)
#boxcox(general_votes~poly(disb,2),          data = d2)



g1 <- filter(d2, can_party == "REP")
g2 <- filter(d2, can_party == "DEM")
g3 <- filter(d2, can_party == "Other")


fit <- lm(g1$votes ~ g1$disb)
par(mfrow=c(2,2))
plot (fit)
```




```r
fit1 <- lm(g2$votes ~ g2$disb)
par(mfrow=c(2,2))
plot (fit1)
```

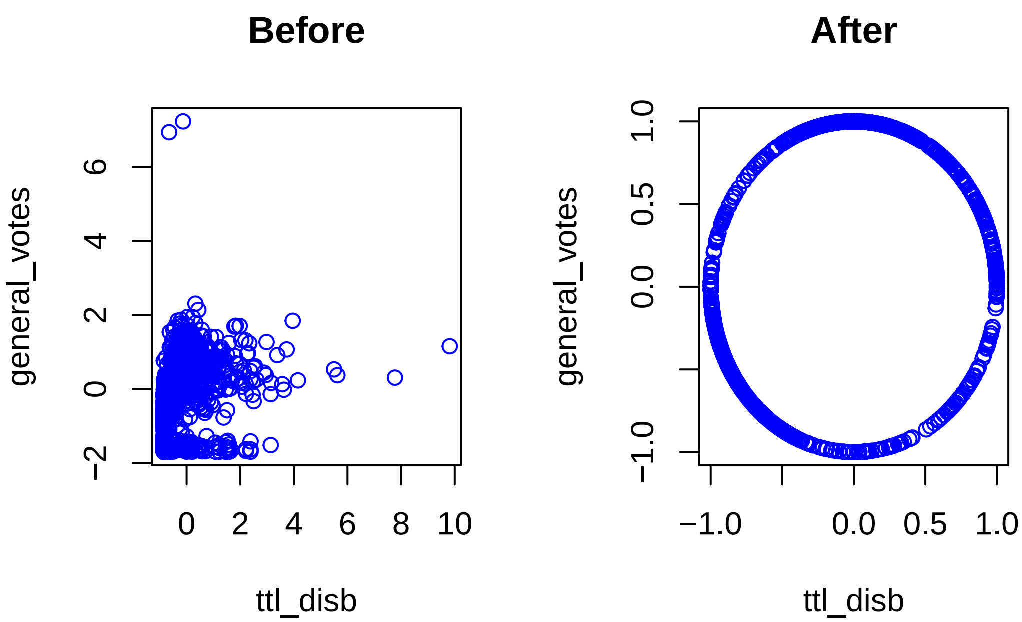


```
fit2 <- lm(g3$votes ~ g3$disb)
par(mfrow=c(2,2))
plot (fit2)
```




```
#summary(fit)
#summary(fit1)
#summary(fit2)
```

```
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")
```



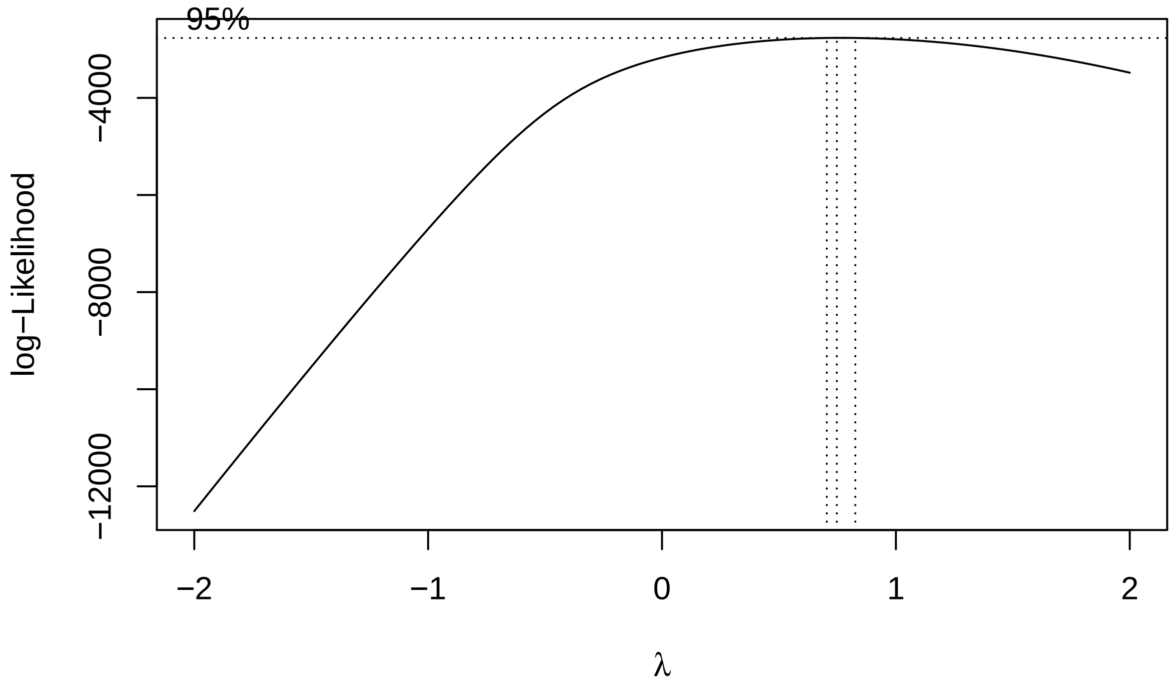

```r
#d2<-transformed

d2[d2 == -Inf] <- 0

head(d2)
```

```
## # A tibble: 6 x 9
##   can_party general_votes ttl_disb state lamvotes  tvotes   tdisb  disb votes
##   <chr>             <dbl>    <dbl> <chr>    <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1 REP              208083 1172750. AL       5458.  0.997   0.0789 14.0   12.2
## 2 REP              134886 1850536. AL       4082. -0.0418  0.999  14.4   11.8
## 3 DEM              112089   36844  AL       3605. -0.348  -0.938  10.5   11.6
## 4 REP              192164 1071289. AL       5174.  1.00   -0.0154 13.9   12.2
## 5 DEM               94549    7348  AL       3217. -0.524  -0.852   8.90  11.5
## 6 REP              235925 1394461. AL       5937.  0.981   0.196  14.1   12.4
```

```r
head(d2$disb)
```

```
## [1] 13.974862 14.430986 10.514448 13.884374  8.902183 14.148019
```

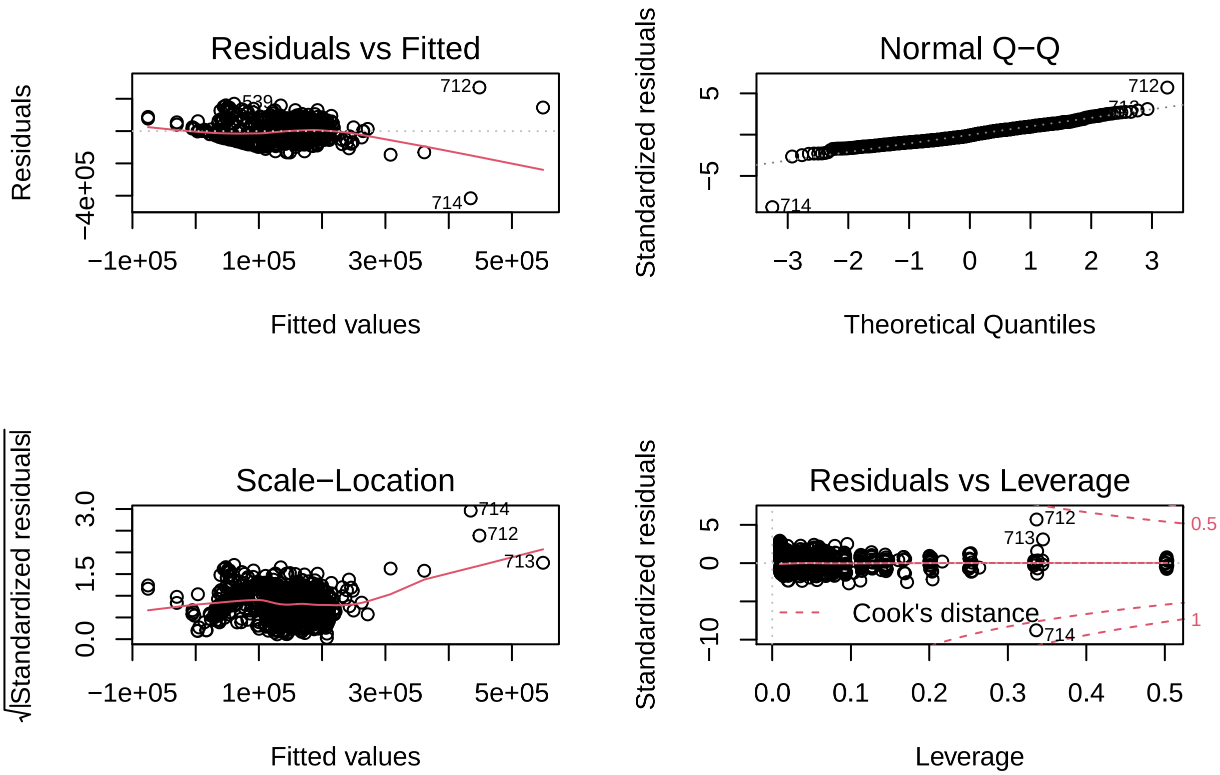
```r
#d3<-d3%>%na.omit()

fit <- lm(d2$general_votes ~ d2$disb)
#summary(fit)


## boxcox test
library(MASS)
boxcox(general_votes~poly(disb,2),
          data = d2)
```




```r
# g0 <- d2
# g0$votes <- log10(g0$general_votes)
# g0$disb <- log10(g0$ttl_disb)
# g0[g0 == -Inf] <- 0


g0 <- d2
g0$votes <- g0$general_votes
g0$disb <- g0$ttl_disb
g0[g0 == -Inf] <- 0

g1 <- filter(d2, can_party == "REP")
g1$votes <- g1$general_votes*g1$general_votes
g1$disb <- log(g1$ttl_disb)
g1[g1 == -Inf] <- 0


g2 <- filter(d2, can_party == "DEM")
g2$votes <- g2$general_votes*g2$general_votes
g2$disb <- log(g2$ttl_disb)
g2[g2 == -Inf] <- 0

g3 <- filter(d2, can_party == "Other")
g3$votes <- g3$general_votes
g3$disb <- log(g3$ttl_disb)
g3[g3 == -Inf] <- 0


write.csv(g1, "g1.csv")
write.csv(g2, "g2.csv")
write.csv(g3, "g3.csv")
```

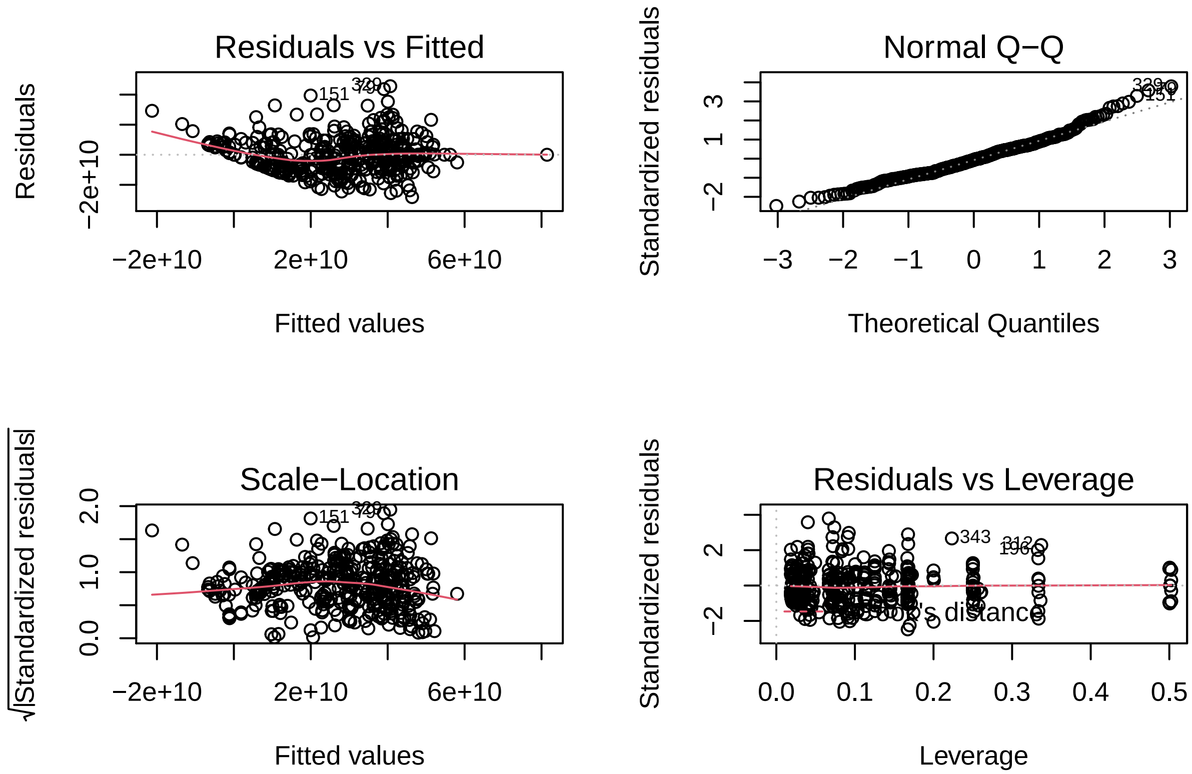
```r
fit0 <- lm(g0$votes ~ g0$disb + g0$state + g0$can_party )
par(mfrow=c(2,2))
plot (fit0)
```

```
## Warning: not plotting observations with leverage one:
##   168, 640, 815, 837
```



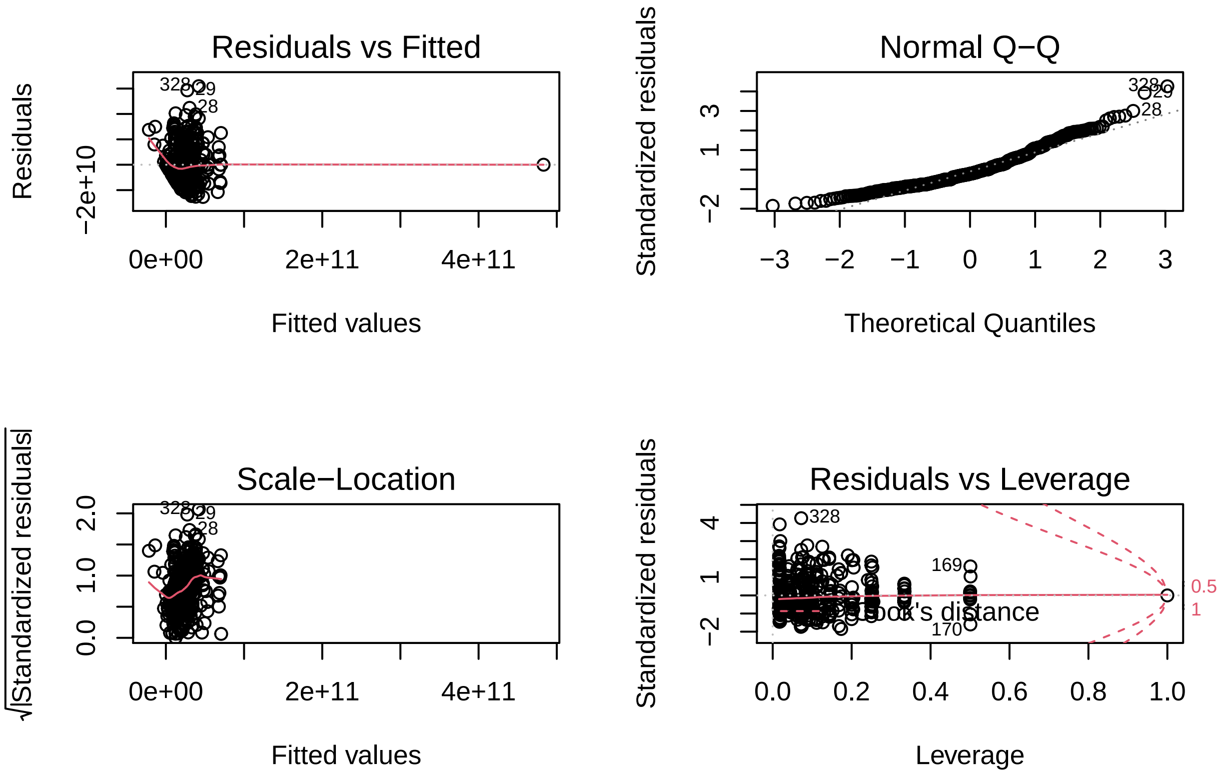

```r
fit1 <- lm(g1$votes ~ g1$disb + g1$state )
par(mfrow=c(2,2))
plot (fit1)
```

```
## Warning: not plotting observations with leverage one:
##   7, 8, 75, 113, 205, 293, 338, 406
```



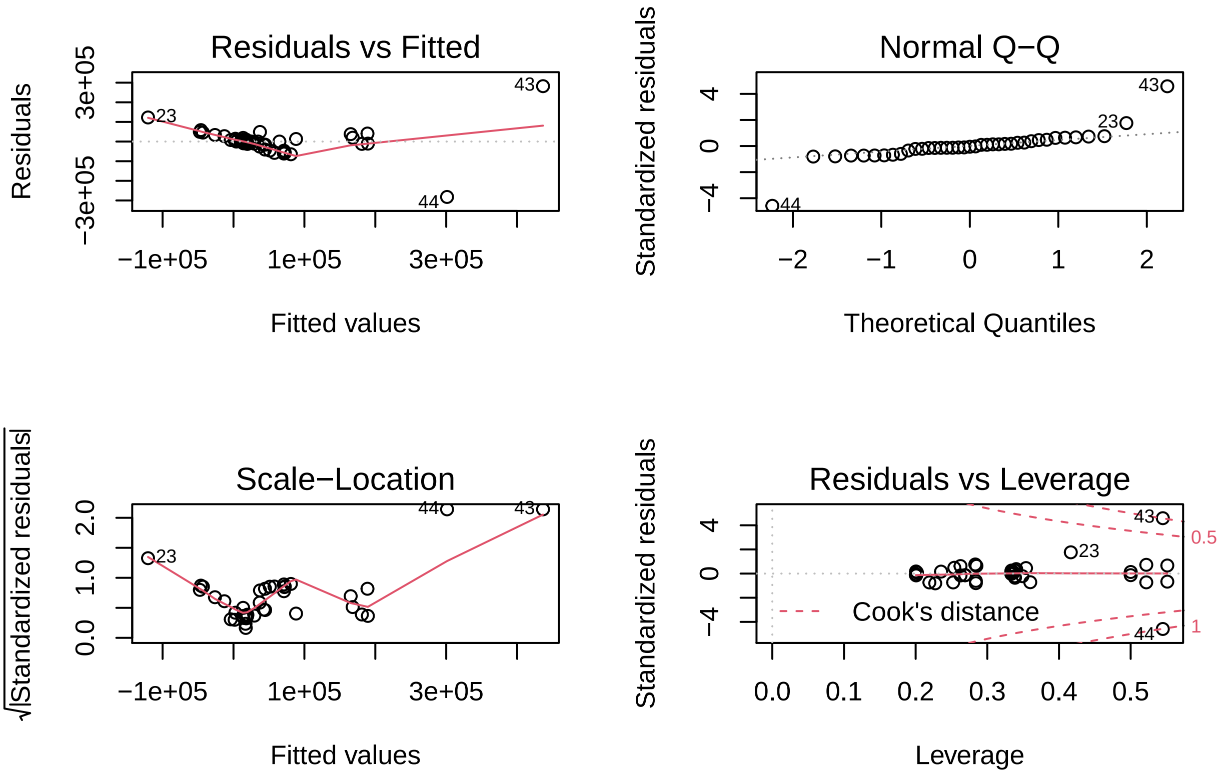

```
fit2 <- lm(g2$votes ~ g2$disb + g2$state )
par(mfrow=c(2,2))
plot (fit2)
```

```
## Warning: not plotting observations with leverage one:
##   6, 91, 92, 126, 130, 211, 212, 307, 322, 341, 356, 389, 401, 423

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```




```
fit3 <- lm(g3$votes ~ g3$disb + g3$state )
par(mfrow=c(2,2))
plot (fit3)
```

```
## Warning: not plotting observations with leverage one:
##   1, 7, 15, 16, 31, 32, 39, 40, 45, 46, 47, 51
```




```r
#summary(fit0)
#summary(fit1)
#summary(fit2)
#summary(fit3)
```

```r
#d2$disb <- log(d2$ttl_disb)
#d2$votes <- log(d2$general_votes)

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

#d2[which(!is.finite(d2))] <- 0
#d2 <- d2[is.finite(rowSums(d2)),]
d2[d2 == -Inf] <- 0

#data_new <- d2                                      # Duplicate data

#d2[is.na(d2$disb) | d2$disb == "Inf"] <- NA   # Replace NaN & Inf with NA

#d3 <- data_new

head(d2)
```

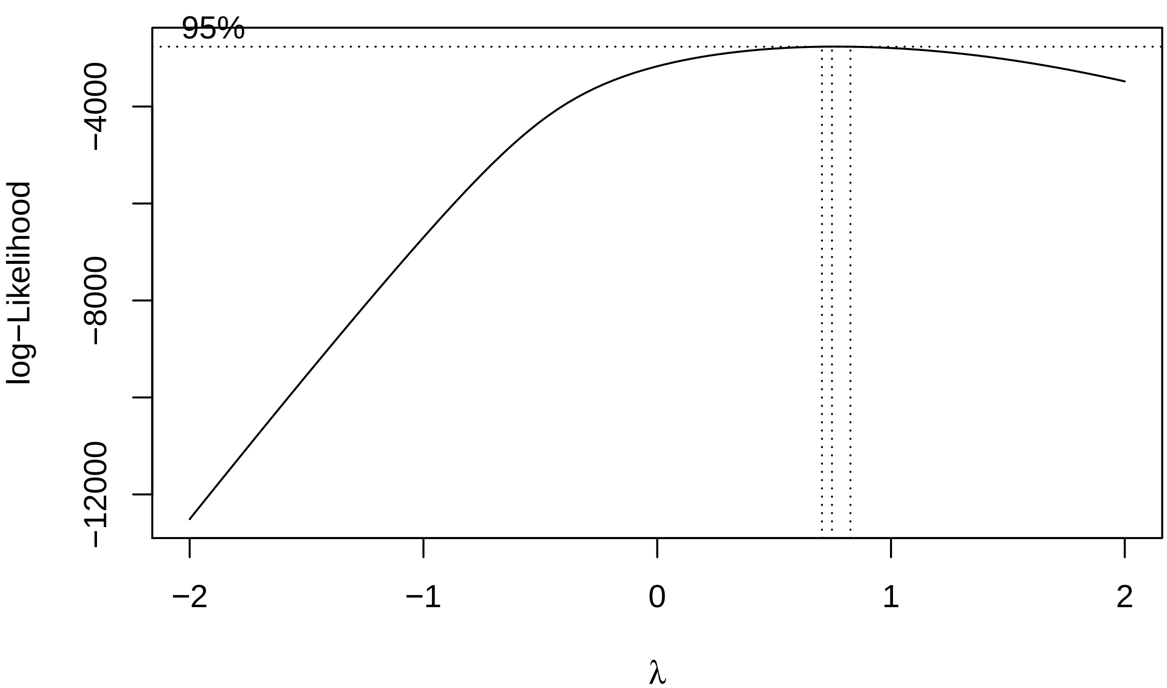
```
## # A tibble: 6 x 9
##   can_party general_votes ttl_disb state lamvotes  tvotes   tdisb  disb votes
##   <chr>             <dbl>    <dbl> <chr>    <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1 REP              208083 1172750. AL       5458.  0.997   0.0789 14.0   12.2
## 2 REP              134886 1850536. AL       4082. -0.0418  0.999  14.4   11.8
## 3 DEM              112089   36844  AL       3605. -0.348  -0.938  10.5   11.6
```

```
## 4 REP              192164 1071289. AL       5174.  1.00   -0.0154 13.9   12.2
## 5 DEM               94549    7348  AL       3217. -0.524  -0.852   8.90  11.5
## 6 REP              235925 1394461. AL       5937.  0.981   0.196  14.1   12.4
```

```
head(d2$disb)
```

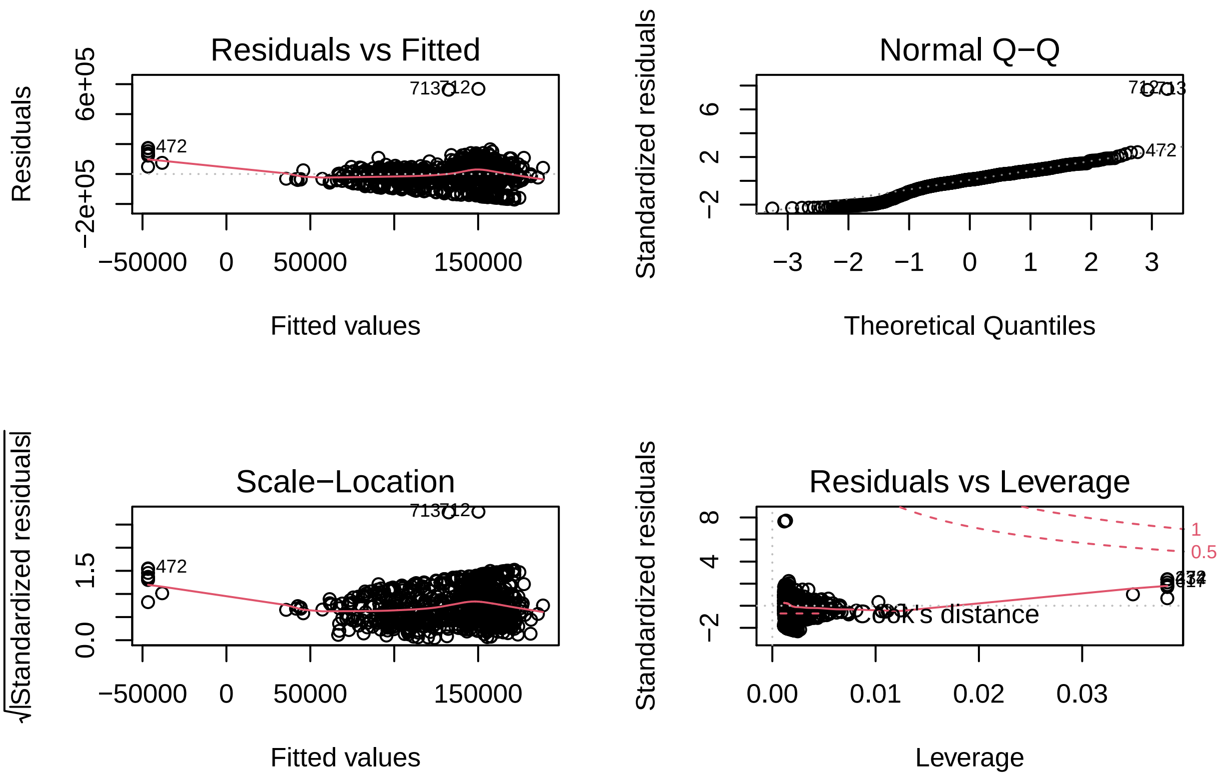
```
## [1] 13.974862 14.430986 10.514448 13.884374  8.902183 14.148019
```

```
#d3<-d3%>%na.omit()

fit <- lm(d2$general_votes ~ d2$disb)

#summary(fit)


## boxcox test
library(MASS)
boxcox(general_votes~poly(disb,2),
          data = d2)
```

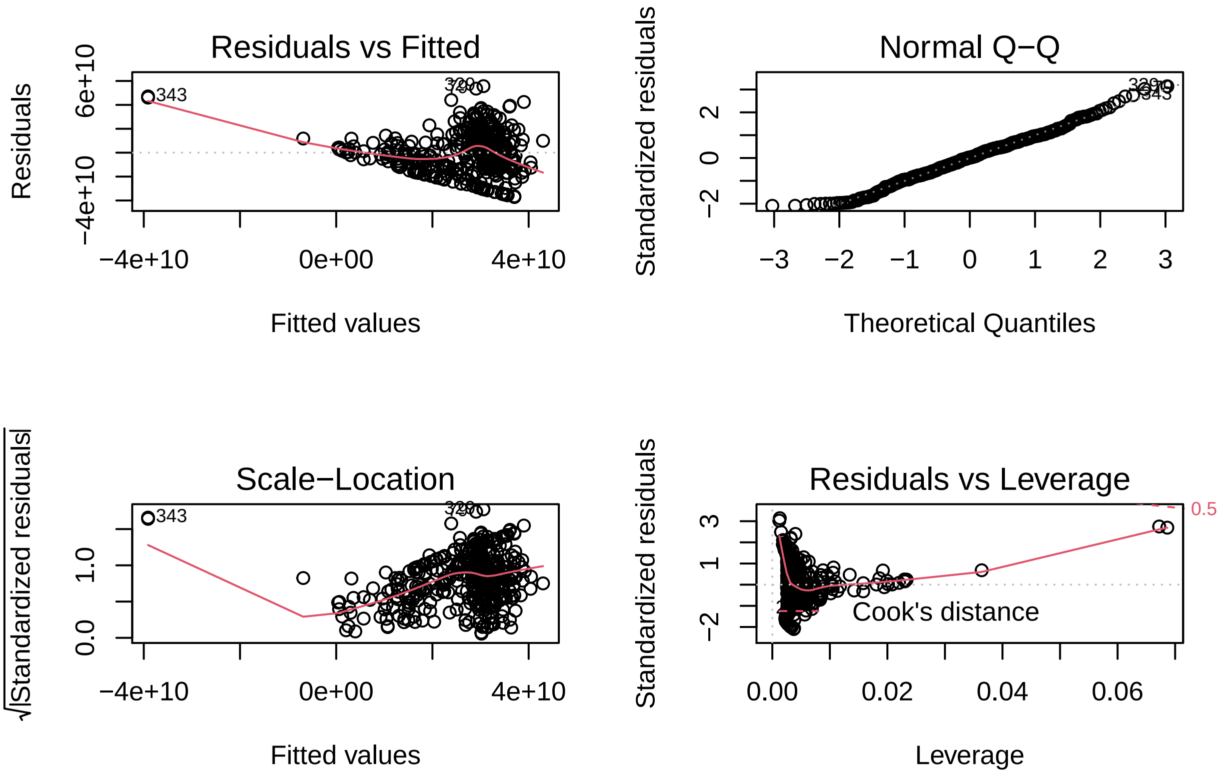



```
g0 <- d2
g0$votes <- log10(g0$general_votes)
g0$disb <- log10(g0$ttl_disb)
g0[g0 == -Inf] <- 0

g1 <- filter(d2, can_party == "REP")
g1$votes <- g1$general_votes*g1$general_votes
g1$disb <- log(g1$ttl_disb)
g1[g1 == -Inf] <- 0
```

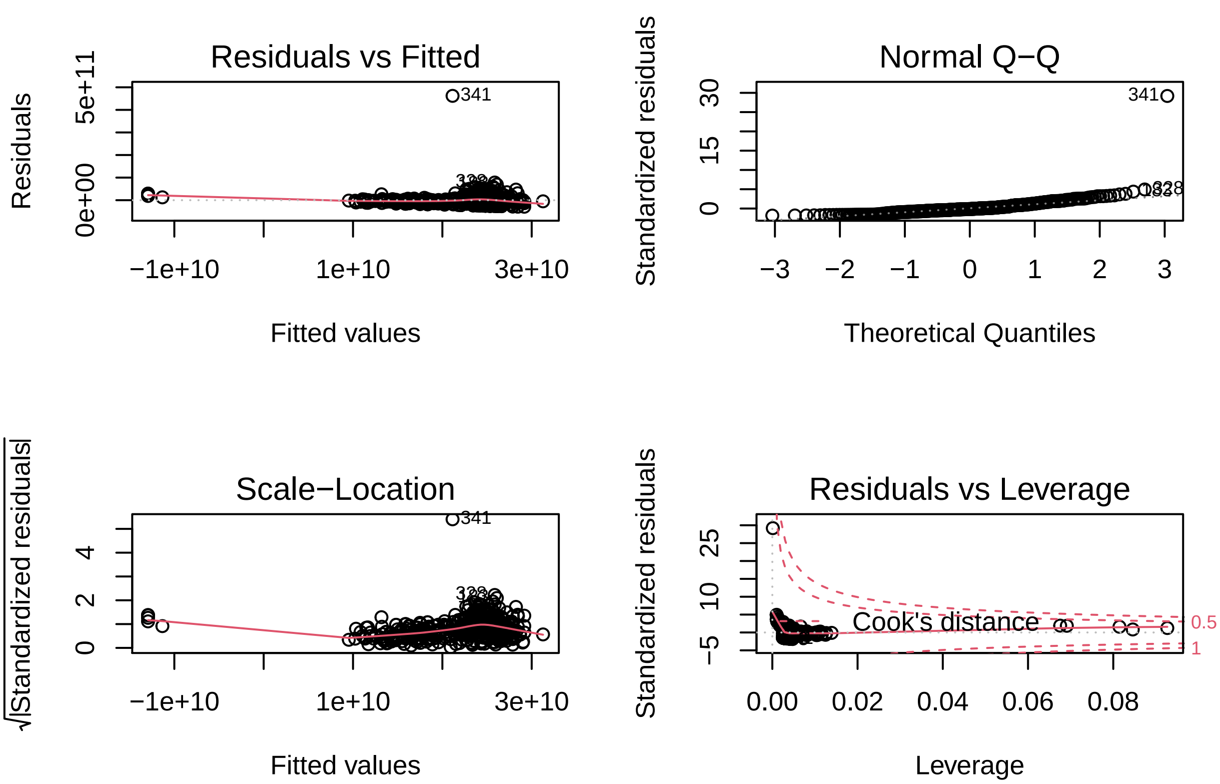
```
g2 <- filter(d2, can_party == "DEM")
g2$votes <- g2$general_votes*g2$general_votes
g2$disb <- log(g2$ttl_disb)
g2[g2 == -Inf] <- 0

g3 <- filter(d2, can_party == "Other")
g3$votes <- g3$general_votes
g3$disb <- log(g3$ttl_disb)
g3[g3 == -Inf] <- 0


write.csv(g1, "g1.csv")
write.csv(g2, "g2.csv")
write.csv(g3, "g3.csv")

fit0 <- rlm(g0$votes ~ g0$disb)
par(mfrow=c(2,2))
plot (fit)

fit1 <- rlm(g1$votes ~ g1$disb)
par(mfrow=c(2,2))
plot (fit)
```




```
fit2 <- rlm(g2$votes ~ g2$disb)
par(mfrow=c(2,2))
plot (fit1)
```




```
fit3 <- rlm(g3$votes ~ g3$disb)
par(mfrow=c(2,2))
plot (fit2)
```



```
#summary(fit0)
#summary(fit1)
#summary(fit2)
#summary(fit3)
```

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)
```