

# 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.

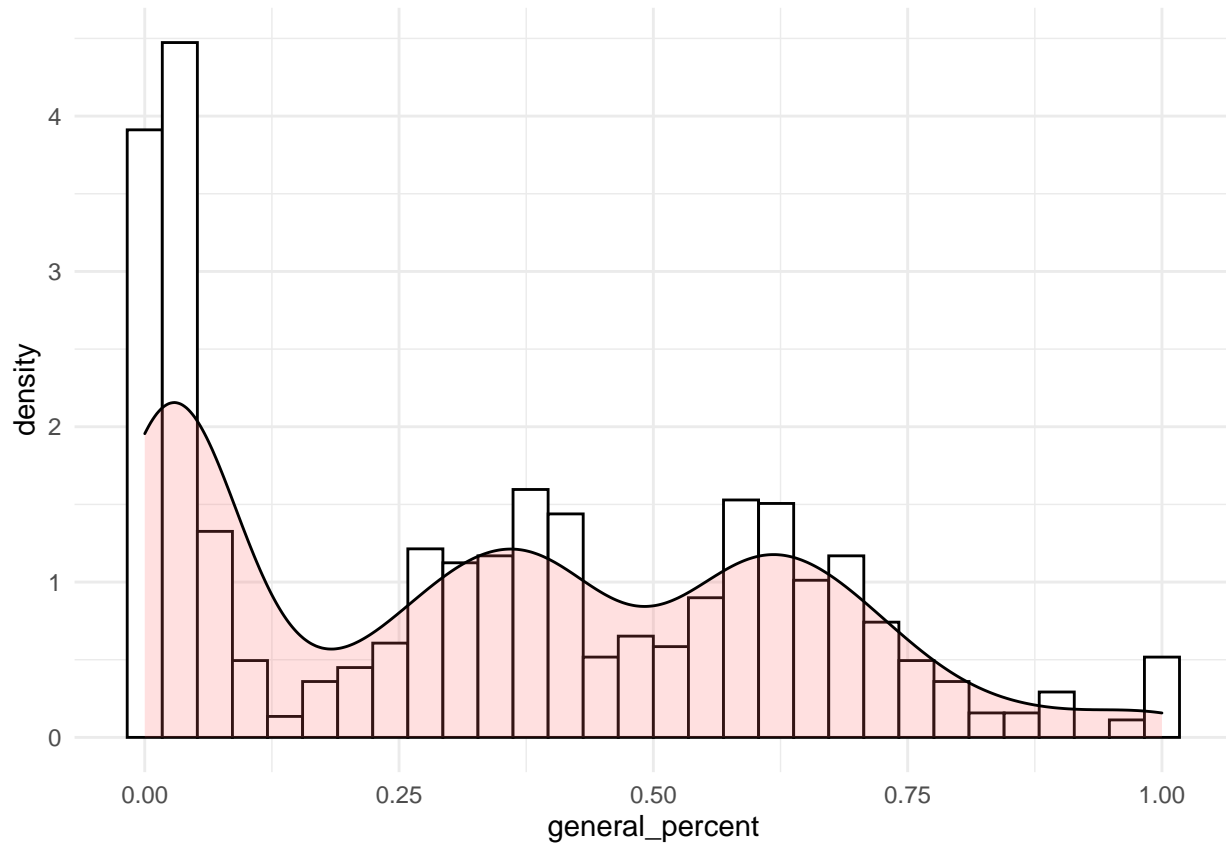
## Your work

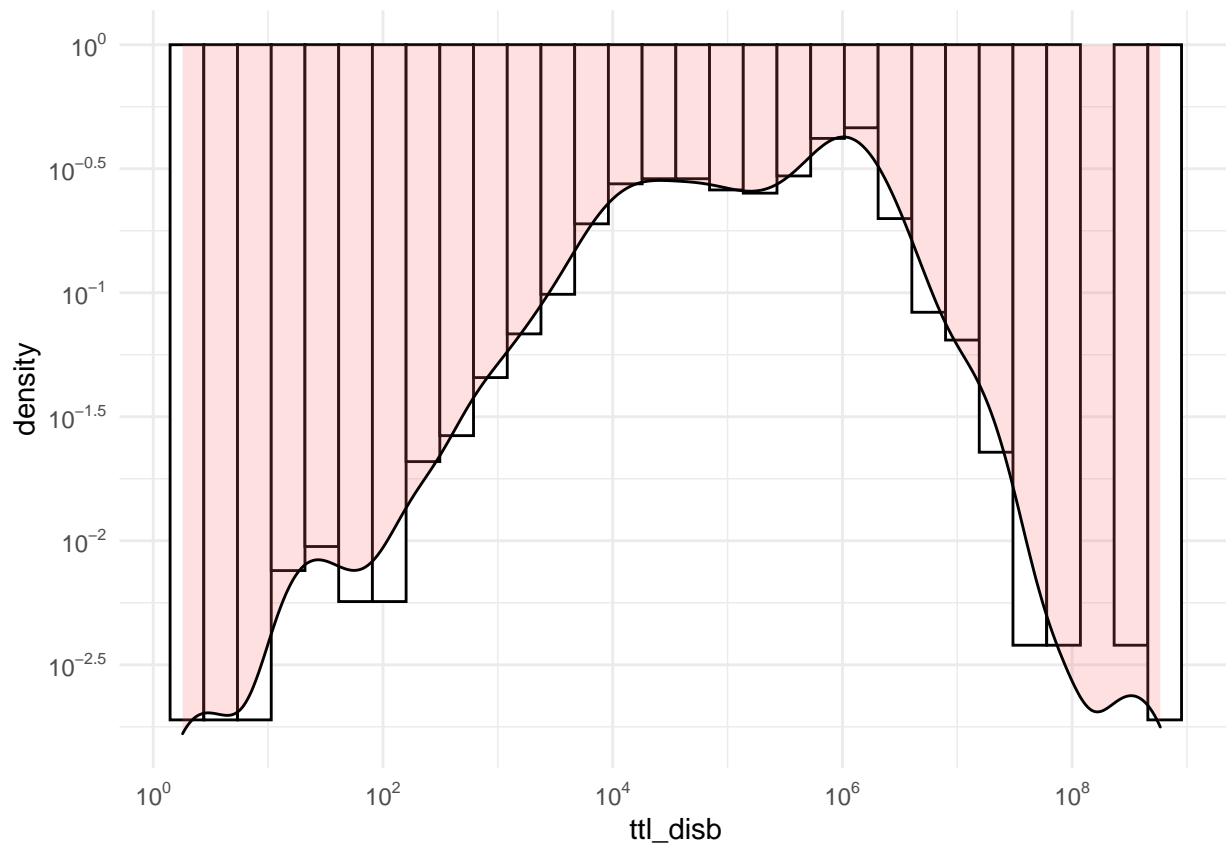
- We want to keep this work *relatively* constrained, which is why we're providing you with data through the `fec16` package. It is possible to gather all the information from current FEC reports, but it would require you to make a series of API calls that would pull us away from the core modeling tasks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

- (3 points) In separate histograms, show both the distribution of votes (measured in `results_house$general_percent` for now) and spending (measured in `t11_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"
```

```
#d1 <- merge(results_house, campaigns, by = "cand_id")
```

```
#d2 <- merge(results_house, campaigns)
```

```
nrow(d1)
```

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

```
#nrow(d2)

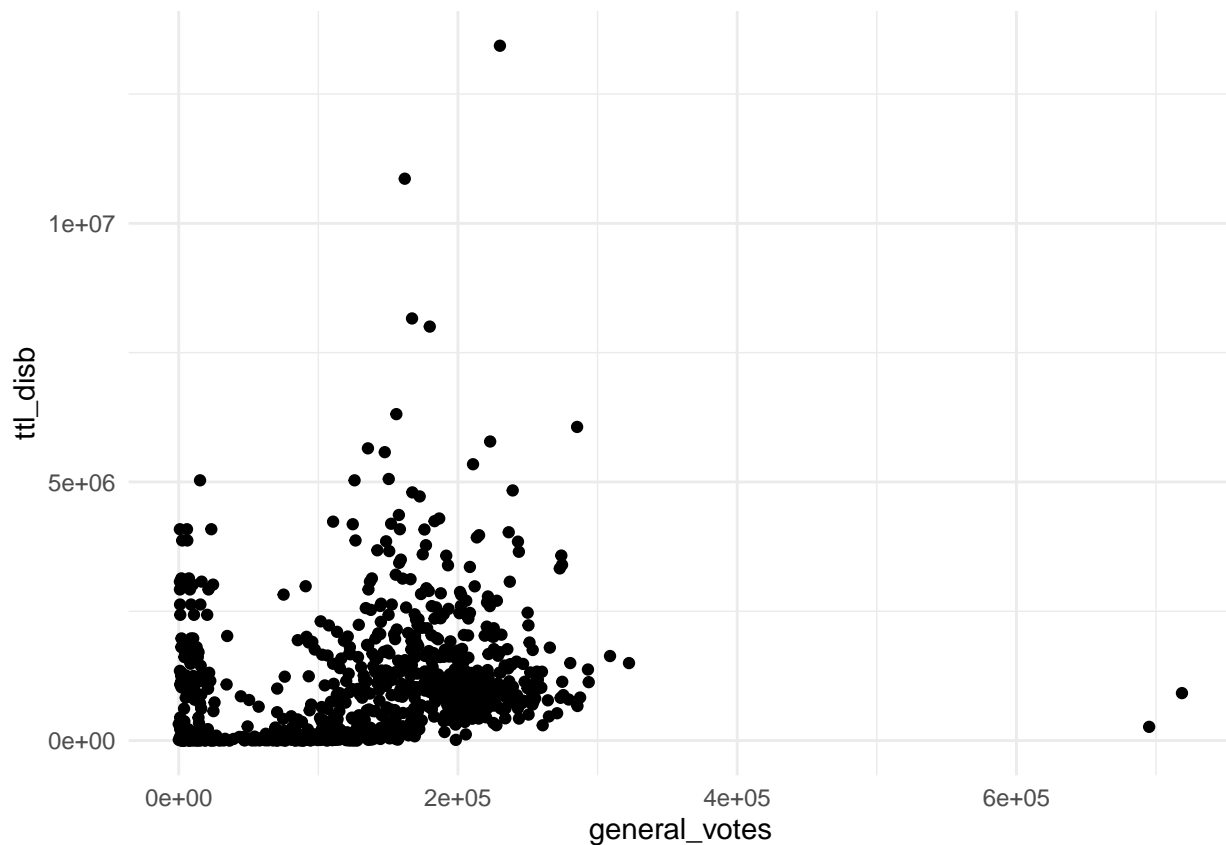
#comparison <- compare(d1,d2,allowAll=TRUE)
#comparison

#summary(d1)
#summary(d2)
```

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(x=general_votes, y=ttl_disb)) + geom_point()
```

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



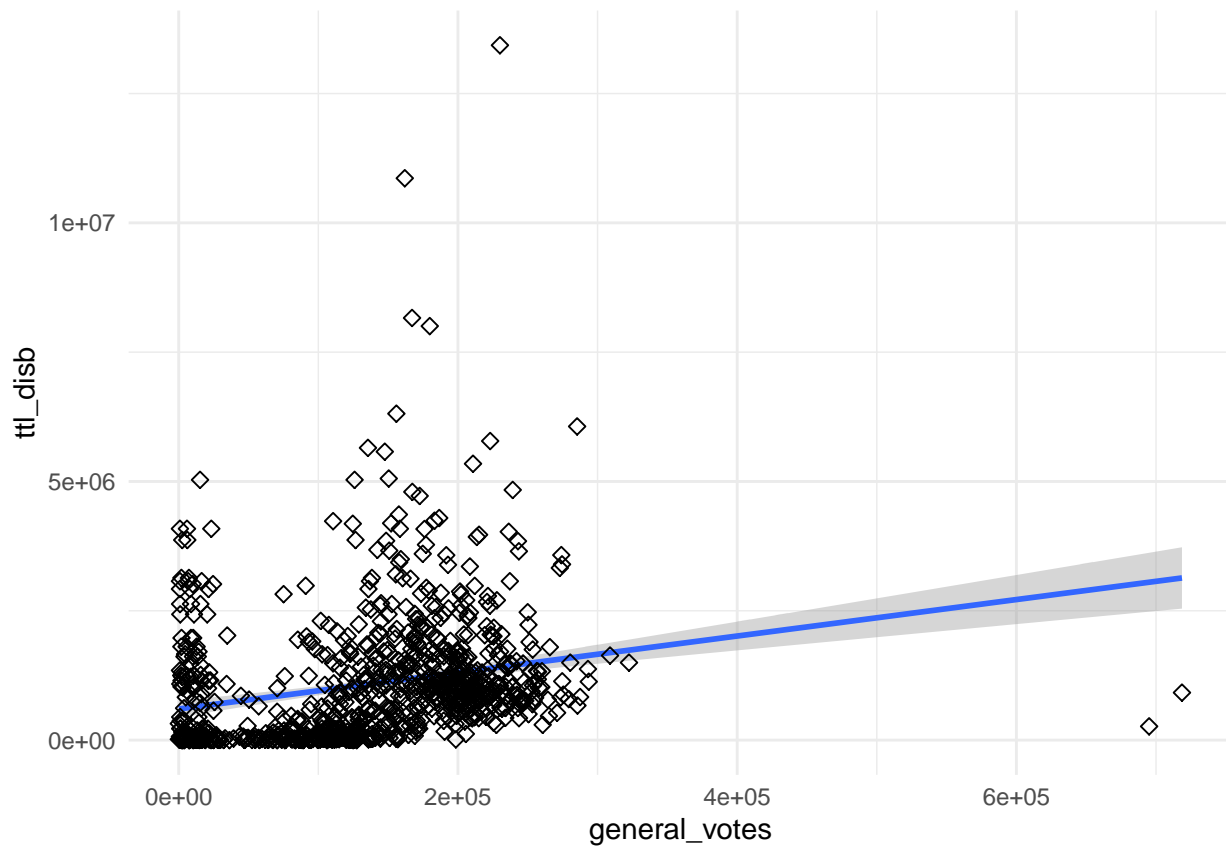
```
# Change the point size, and shape
sp <- ggplot(d1, aes(x=general_votes, y=ttl_disb )) +
  geom_smooth(method=lm)+
  geom_point(size=2, shape=23)

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).
```



```
#sp + geom_density_2d()
```

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 %>%
  select(cand_pty_affiliation, general_votes, ttl_disb) %>%
  na.omit() %>%
  mutate(
    can_party = case_when(
      cand_pty_affiliation=="REP" ~ "REP",
      cand_pty_affiliation=="DEM" ~ "DEM",
      TRUE ~ "Other"
    )
  )

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

#print(d2$general_votes)

#summary(d2)

#ggplot(d1, aes(x=general_votes, y=ttl_disb)) + geom_point()
# Change the point size, and shape

#d3 <- d2 %>% data.frame(can_party, general_votes, ttl_disb)

head(d2)

```

```

## # A tibble: 6 x 4
##   cand_pty_affiliation general_votes ttl_disb can_party
##   <chr>                <dbl>    <dbl> <chr>
## 1 REP                  208083  1172750. REP
## 2 REP                  134886  1850536. REP
## 3 DEM                  112089    36844. DEM
## 4 REP                  192164  1071289. REP
## 5 DEM                   94549     7348. DEM
## 6 REP                  235925  1394461. REP

```

```

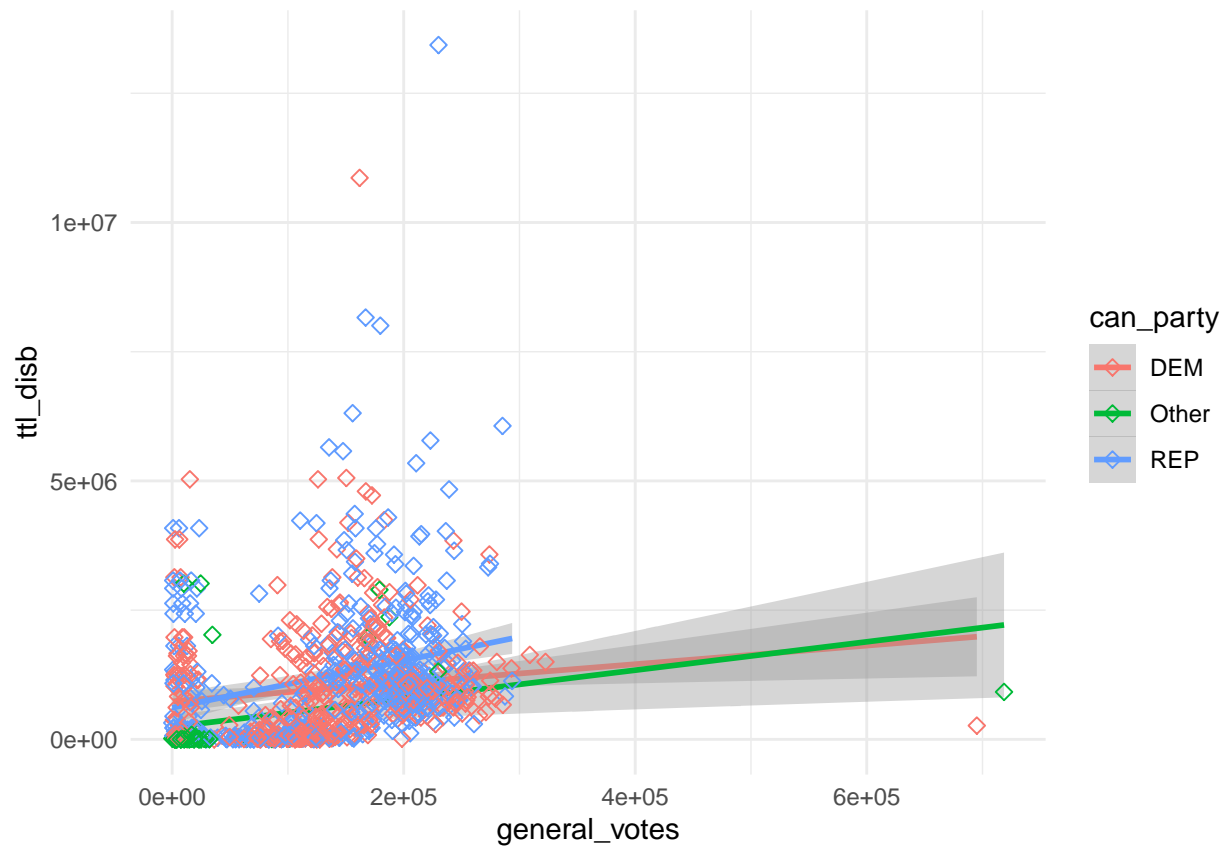
sp <- ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_smooth(method=lm)+
  geom_point(size=2, shape=23)
sp

```

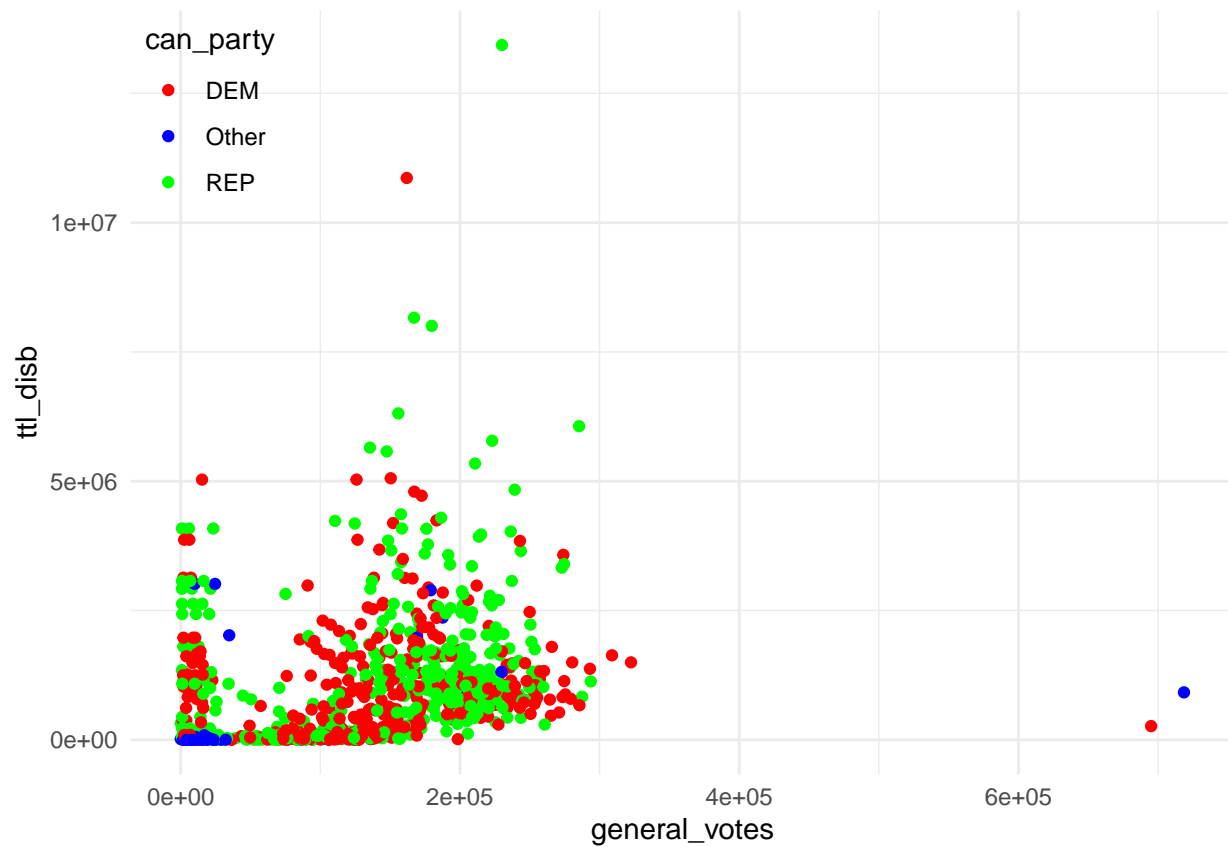
```

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

```

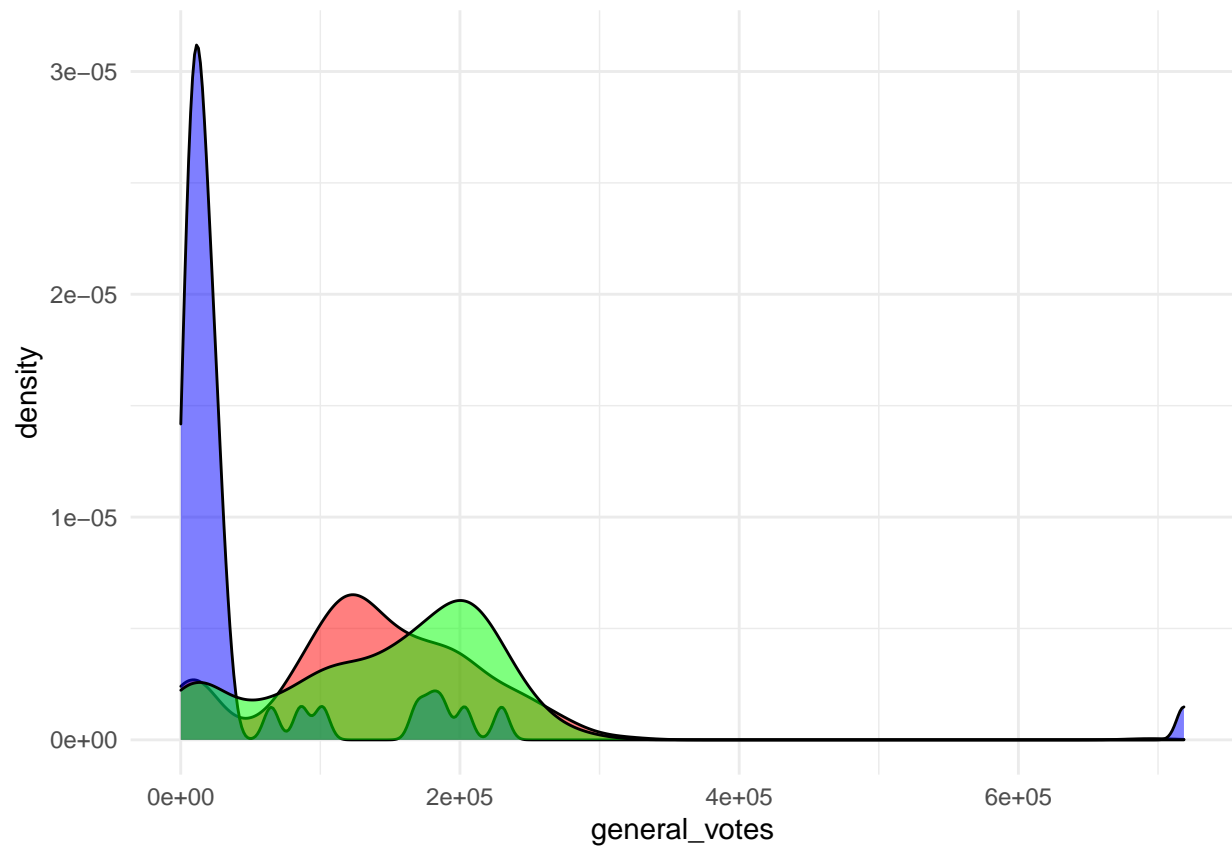


```
p1<-ggplot(d2, aes(x=general_votes, y=ttl_disb, color=can_party)) +
  geom_point() +
  scale_color_manual(values = c("red", "blue", "green")) +
  theme(legend.position=c(0,1), legend.justification=c(0,1))
p1
```

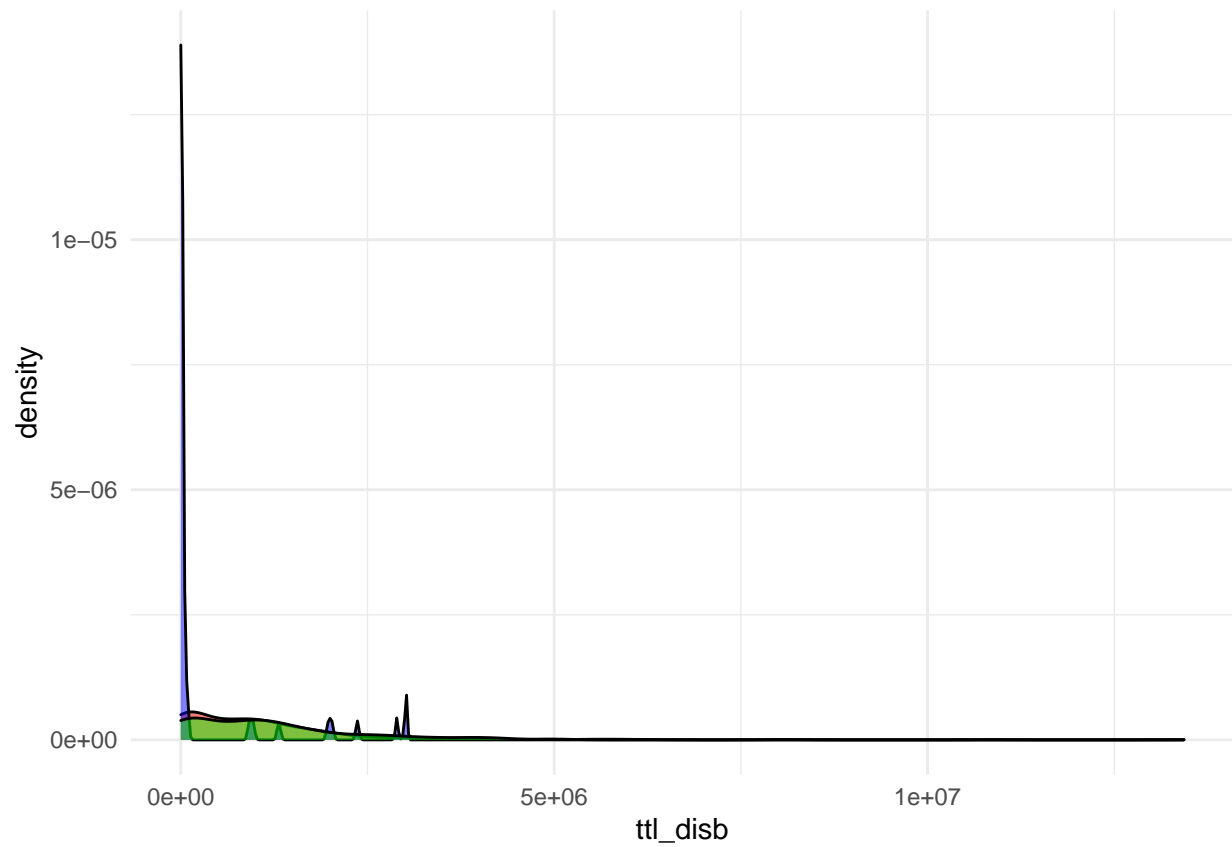


```
p2<-ggplot(d2, aes(x=general_votes, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values = c("red", "blue", "green")) +
  theme(legend.position = "none")
p2
```

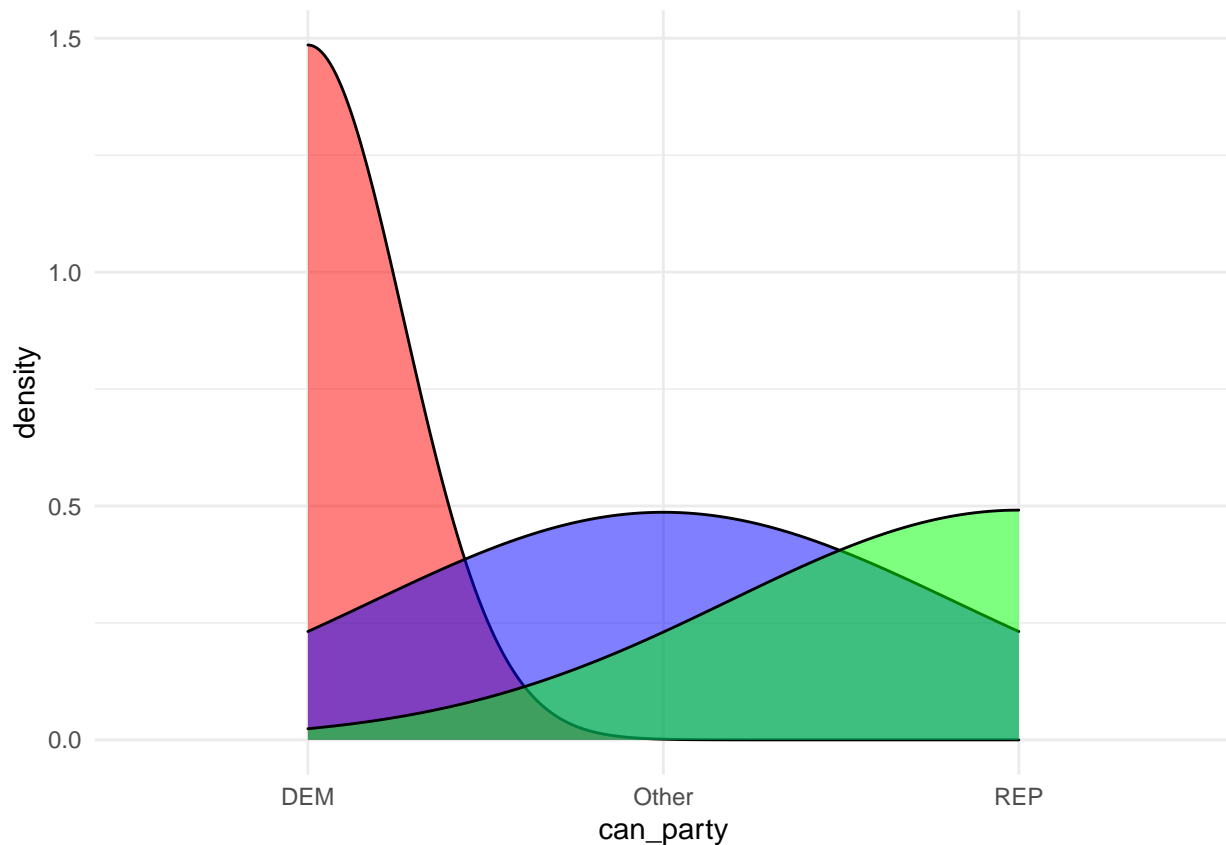




```
# Marginal density plot of y (right panel)
p3<-ggplot(d2, aes(x=ttl_disb, fill=can_party)) +
  geom_density(alpha=.5) +
  scale_fill_manual(values = c("red", "blue", "green")) +
  theme(legend.position = "none")
p3
```



```
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
```



```
#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$disb <- log(d2$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 6
##   cand_pty_affiliation general_votes ttl_disb can_party disb votes
##   <chr>                <dbl>    <dbl> <chr>    <dbl> <dbl>
## 1 REP                 208083 1172750. REP      14.0  12.2
## 2 REP                 134886 1850536. REP      14.4  11.8
## 3 DEM                 112089   36844  DEM      10.5  11.6
## 4 REP                 192164 1071289. REP      13.9  12.2
## 5 DEM                 94549    7348  DEM       8.90 11.5
## 6 REP                 235925 1394461. REP      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)
```

```
##
## Call:
## lm(formula = d2$general_votes ~ d2$disb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -170750  -34066    7653   45029  568412
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -46697     14420   -3.238  0.00125 **
## d2$disb         14339       1109  12.928 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73740 on 878 degrees of freedom
## Multiple R-squared:  0.1599, Adjusted R-squared:  0.159
## F-statistic: 167.1 on 1 and 878 DF, p-value: < 2.2e-16
```

```
## boxcox test
```

```
library(MASS)
```

```
##
```

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

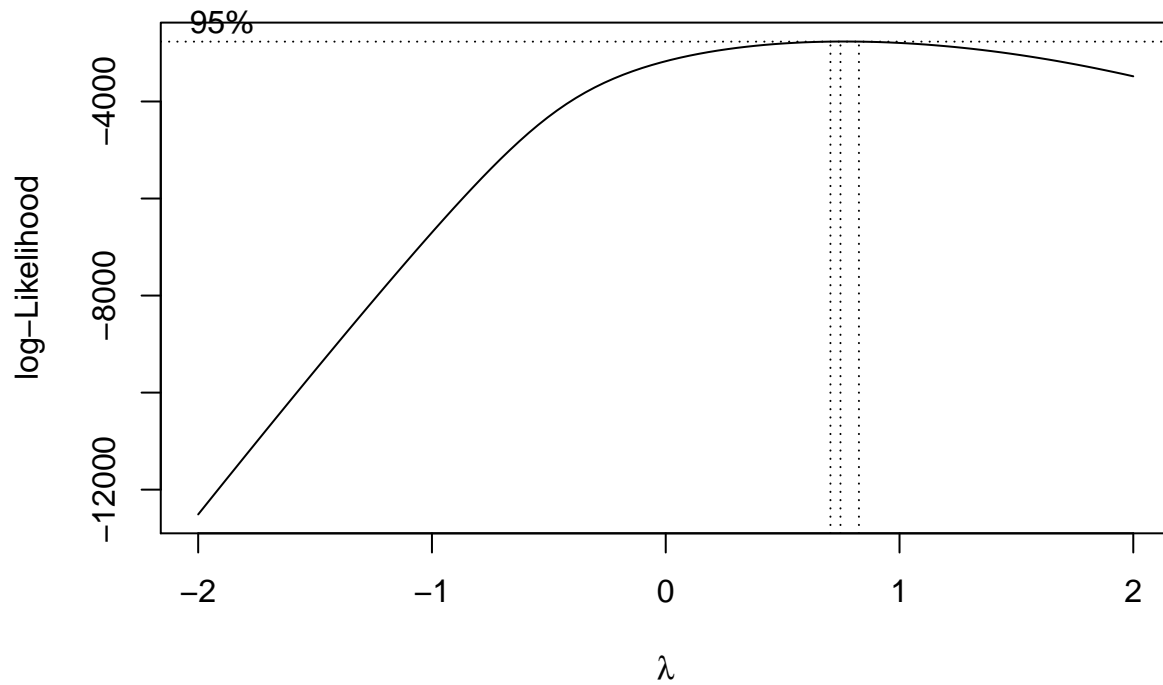
```
## The following object is masked from 'package:patchwork':
```

```
##
```

```
## area
```

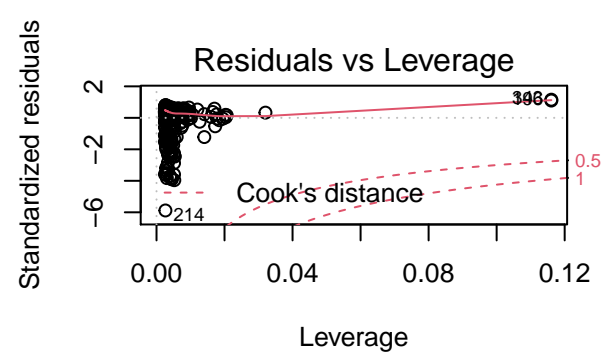
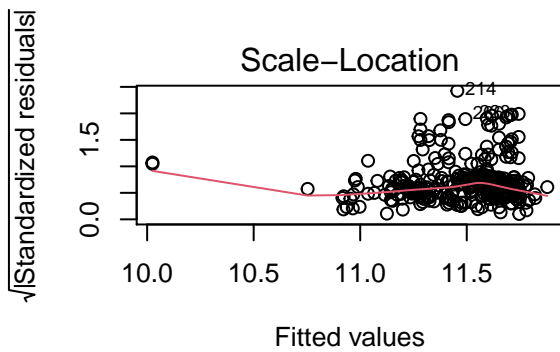
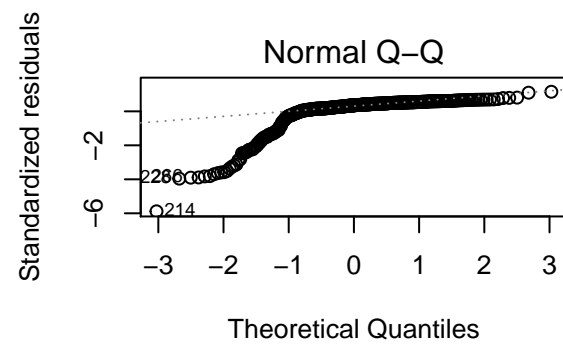
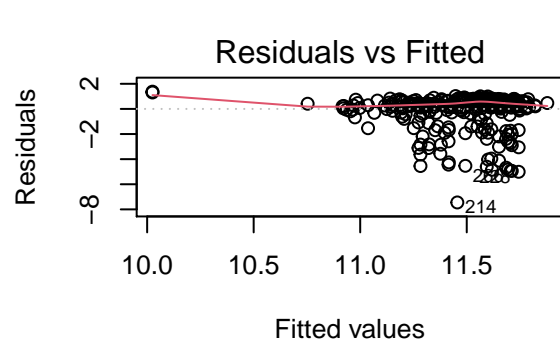
```
## The following object is masked from 'package:dplyr':  
##  
##      select
```

```
boxcox(general_votes~poly(disb,2),  
       data = d2)
```

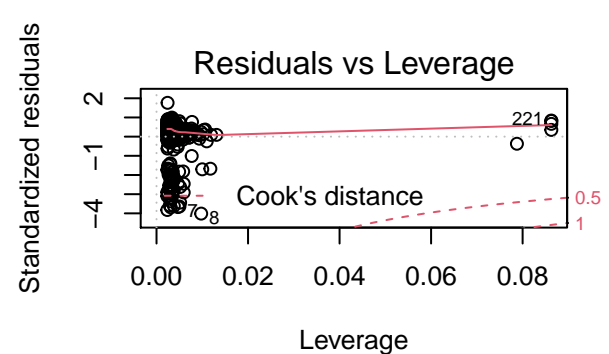
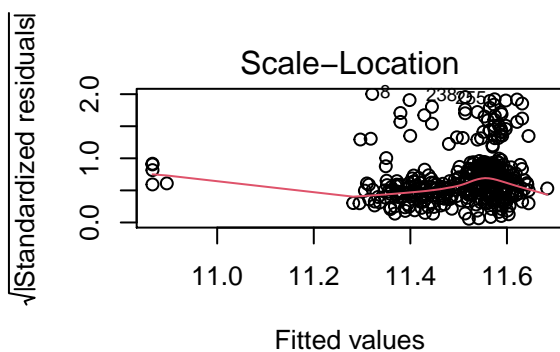
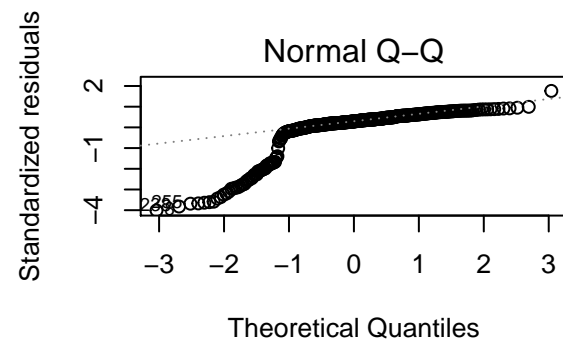
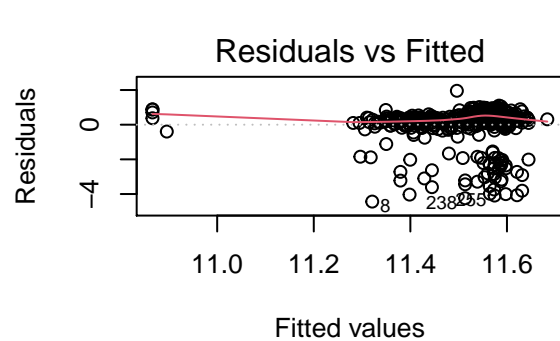


```
g1 <- filter(d2, can_party == "REP")  
g2 <- filter(d2, can_party == "DEM")  
g3 <- filter(d2, can_party == "Other")
```

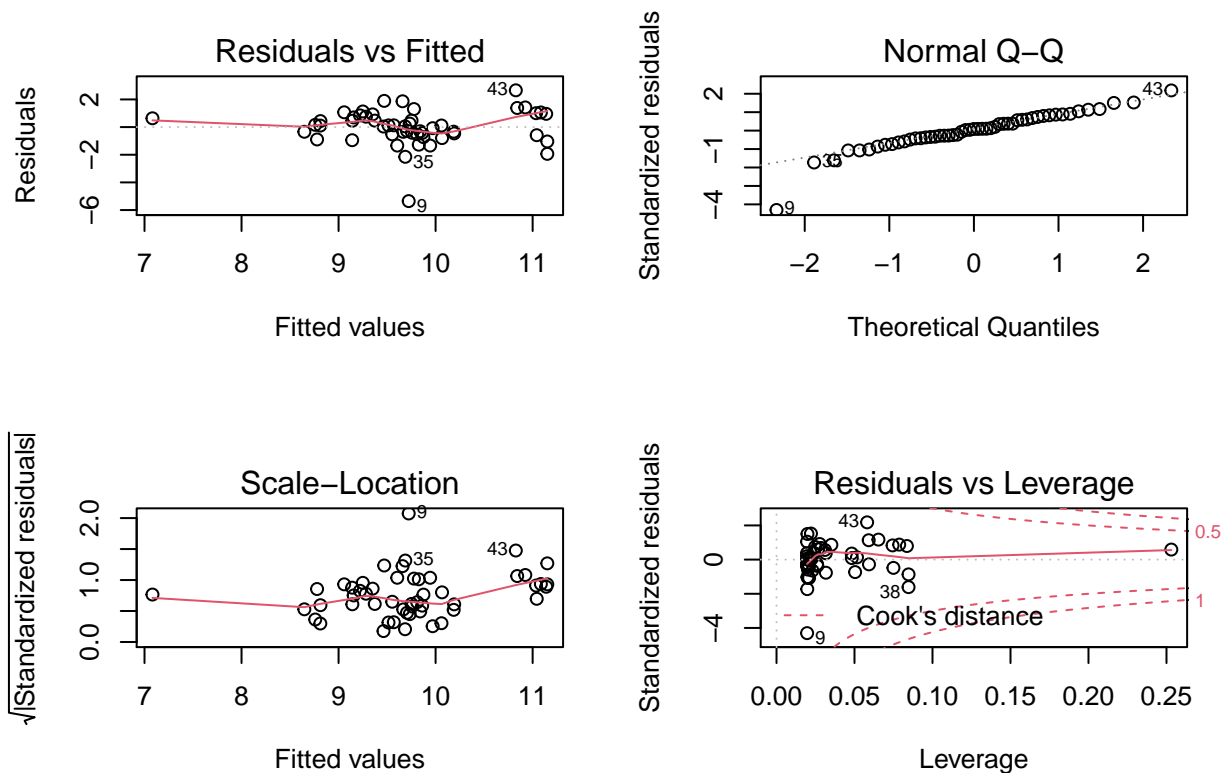
```
fit <- lm(g1$votes ~ g1$disb)  
par(mfrow=c(2,2))  
plot (fit)
```



```
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)
```

```
##
## Call:
## lm(formula = g1$votes ~ g1$disb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.4485  0.1288  0.4484  0.6419  1.3730
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.02502    0.43189  23.212  < 2e-16 ***
## g1$disb      0.11290    0.03245   3.479 0.000557 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.268 on 404 degrees of freedom
## Multiple R-squared:  0.0291, Adjusted R-squared:  0.02669
## F-statistic: 12.11 on 1 and 404 DF, p-value: 0.0005571
```

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$general_votes ~ b1*d2$t1l_disb + b2)
```