Unit 09: Programming Practice

Diogo Viveiros

2025-10-25

```
library(dplyr)
```

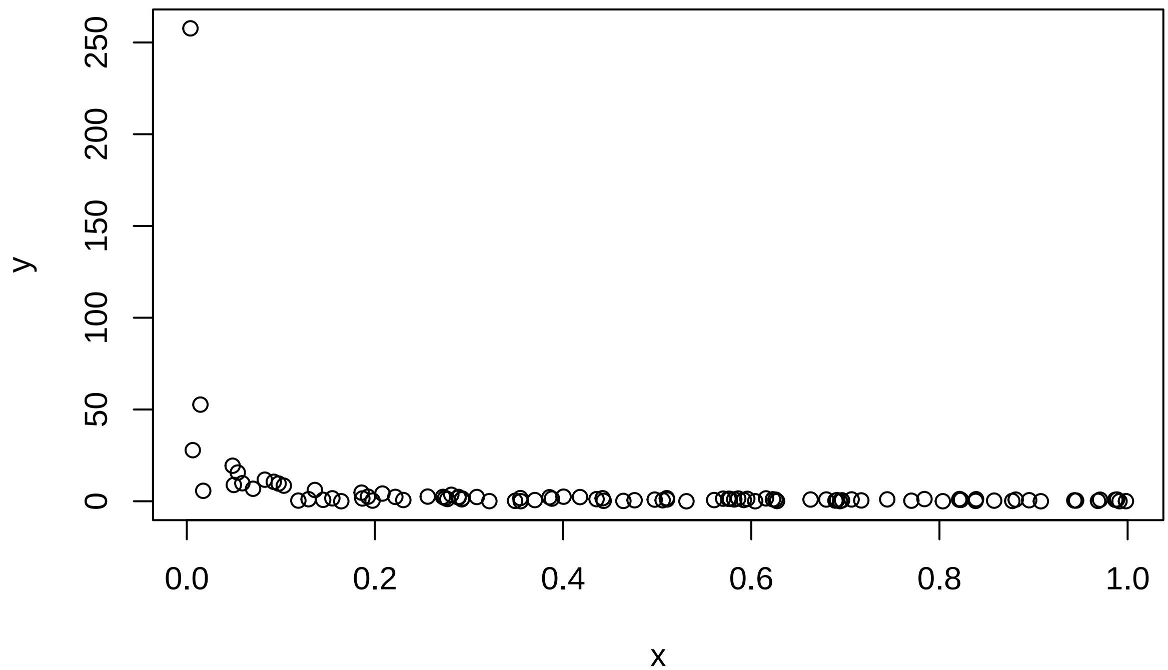
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

Question 3

Question 3.1

```
 rmystery <- function(n){
  x = runif(n)
  y = runif(n, min=0, max = 1/x)
  data.frame(x=x,y=y)
}
plot(rmystery(100))
```

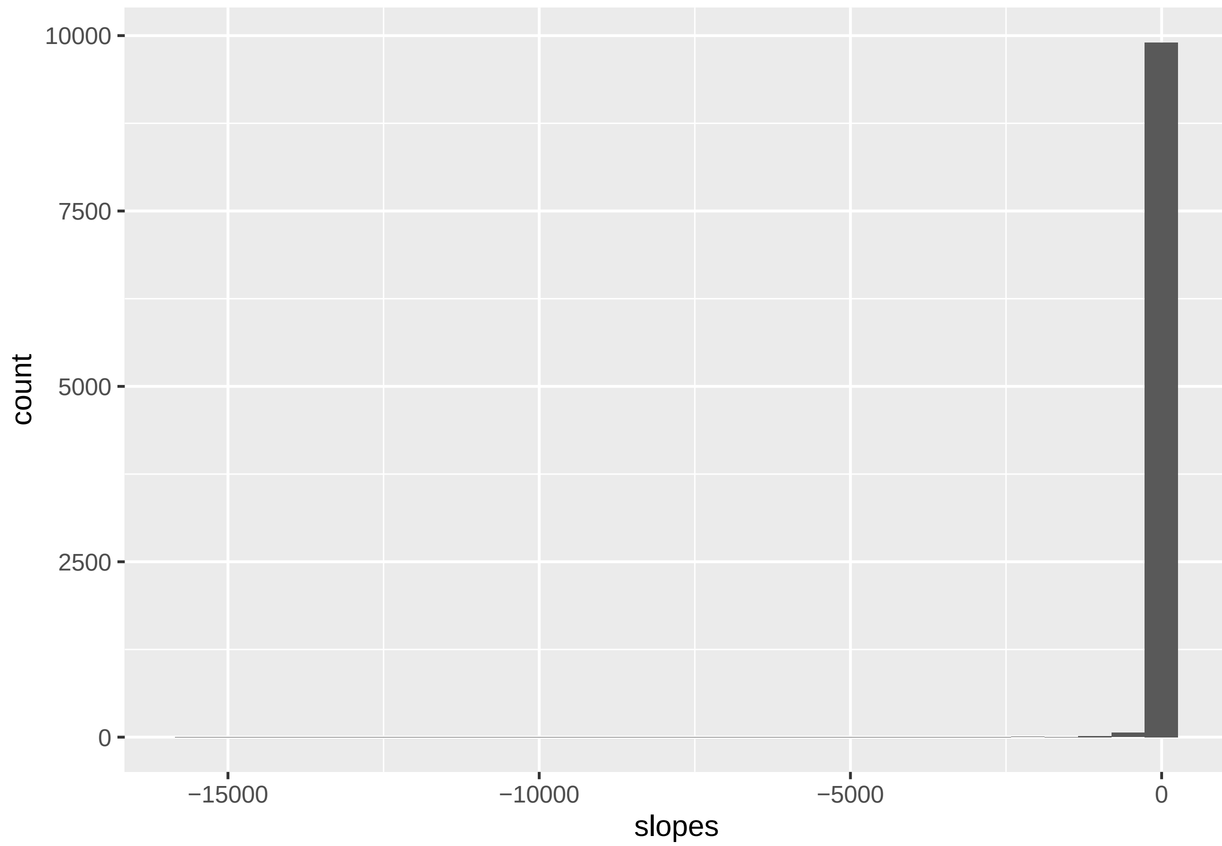



```
experiment_m <- function(){
  df <- rmystery(100)
  reg <- lm(y ~ x, data = df)
  slope <- coef(reg)[2]
  return(slope)
}
```

```
df_q3 <- data.frame()
```

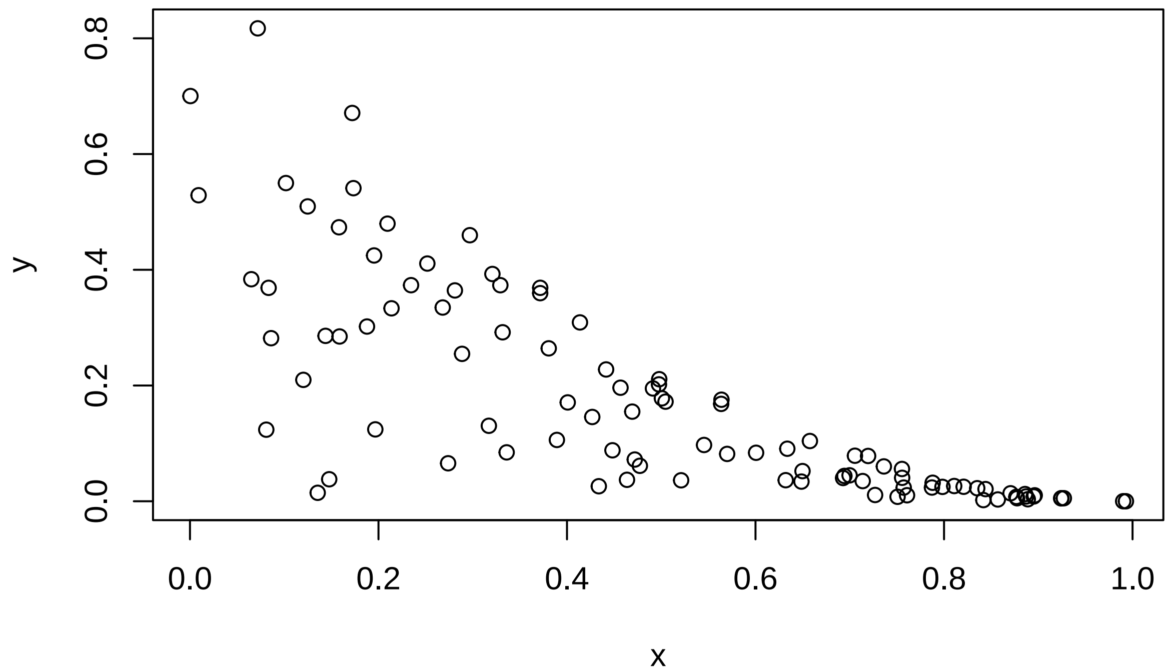
```
slopes_m <- replicate(10000, experiment_m())
df_q3.1 <- data.frame(slopes = slopes_m)
ggplot(data = df_q3.1, aes(x = slopes)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



Question 3.3

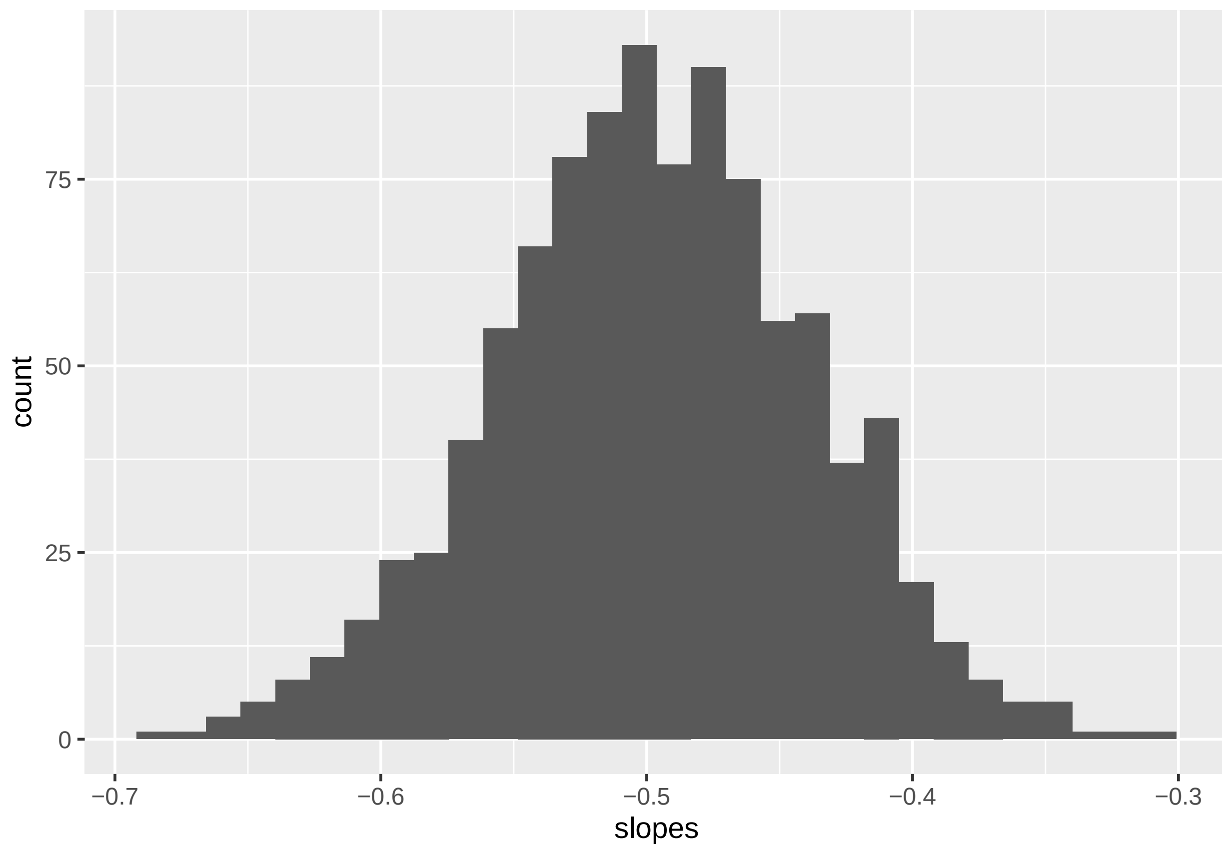
```
renigma <- function(n){
  x = runif(n)
  y = runif(n, min=0, max = (1-x)^2)
  data.frame(x=x,y=y)
}
plot(renigma(100))
```




```
experiment_e <- function(){
  df <- renigma(100)
  reg <- lm(y ~ x, data = df)
  slope <- coef(reg)[2]
  return(slope)
}
```

```
slopes_e <- replicate(1000, experiment_e())
df_q3.3 <- data.frame(slopes = slopes_e)
ggplot(data = df_q3.3, aes(x = slopes)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```




```r
#hist(slopes_e, breaks = seq(min(slopes_e), max(slopes_e))
```

Question 4

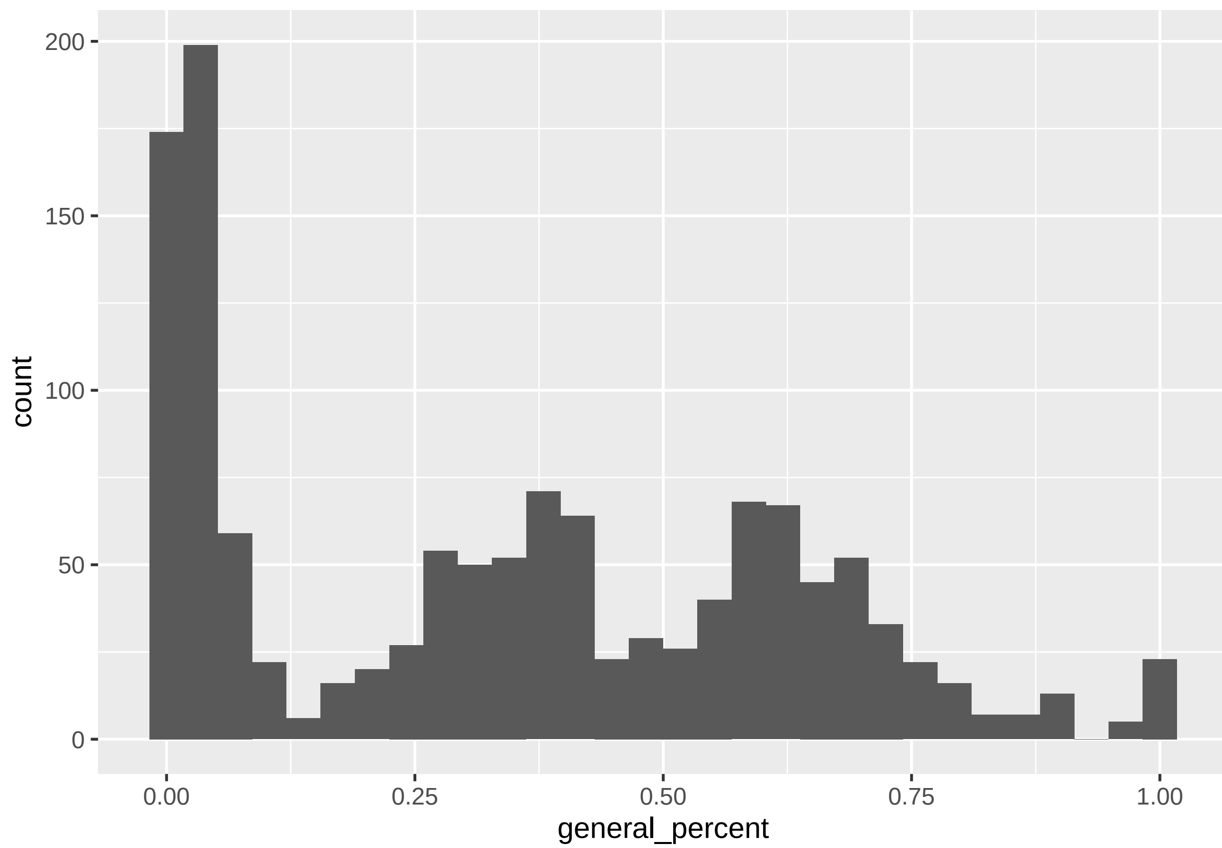
```r
library("fec16")
data("results_house")
data("campaigns")
```

Question 4.1

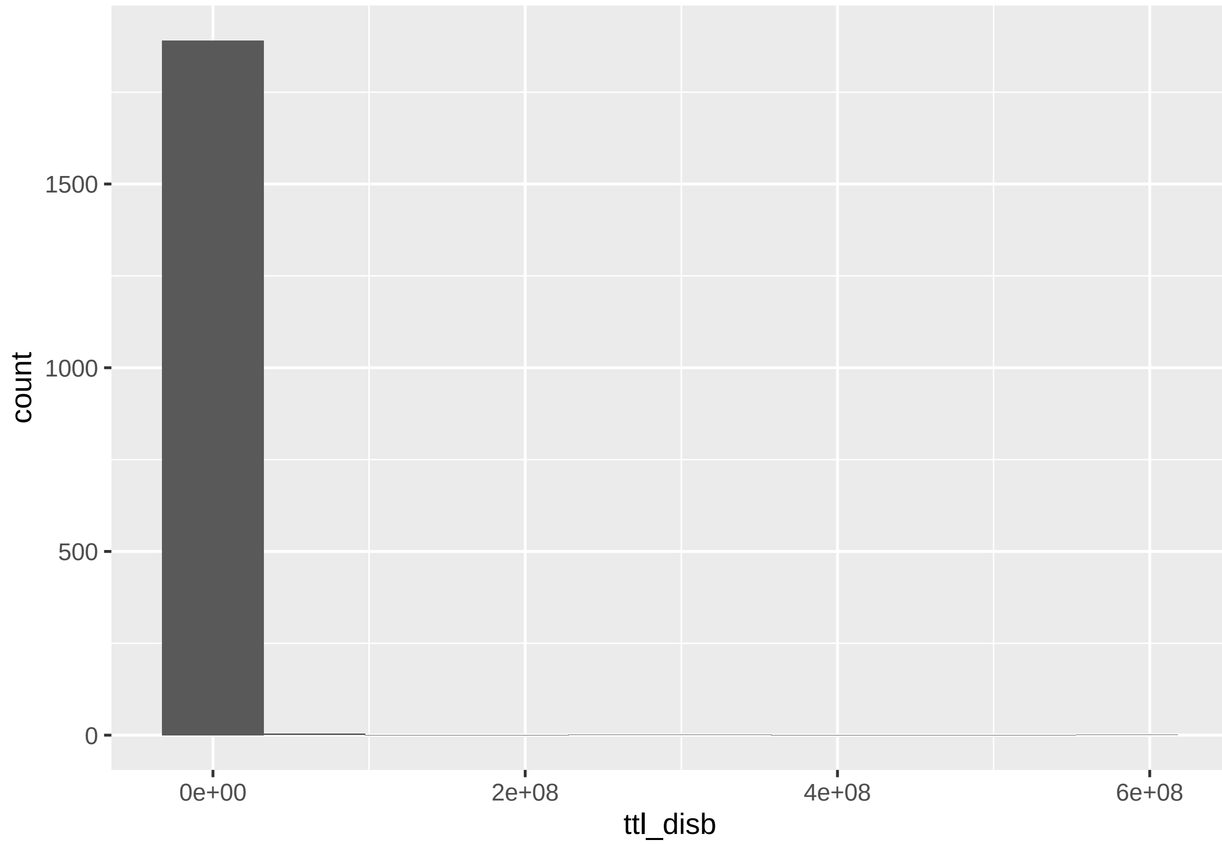
```r
ggplot(data = results_house, aes(x = general_percent)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```

```
## Warning: Removed 820 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```r
ggplot(data = campaigns, aes(x = ttl_disb)) +
  geom_histogram(bins = 10)
```



Questions 4.2/4.3

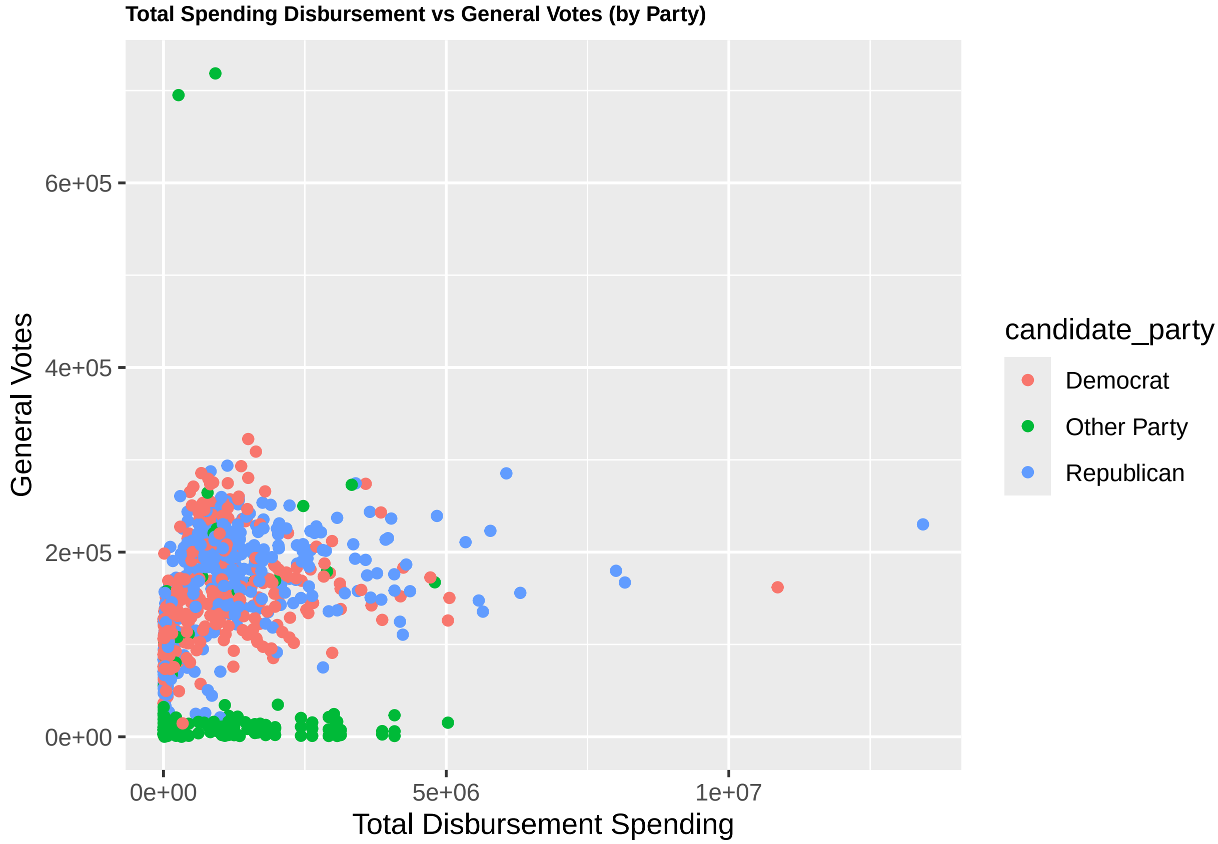
```
df_q4 <- inner_join(results_house, campaigns, by = "cand_id")
```

Question 4.4

```
df_q4 <- df_q4 %>%
  mutate(
    candidate_party = case_when(
      party == "DEM" ~ "Democrat",
      party == "REP" ~ "Republican",
      TRUE ~ "Other Party"
    )
  )
```

```
ggplot(data = df_q4, aes(x = ttl_disb, y=general_votes, color = candidate_party)) +
  geom_point() +
  labs(
       x = "Total Disbursement Spending",
       y = "General Votes")+
  ggtitle("Total Spending Disbursement vs General Votes (by Party)") +
  theme(plot.title = element_text(size = 8, face = "bold"))
```

```
## Warning: Removed 462 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Question 4.5

Large-Sample Assumptions

I.I.D. Data:

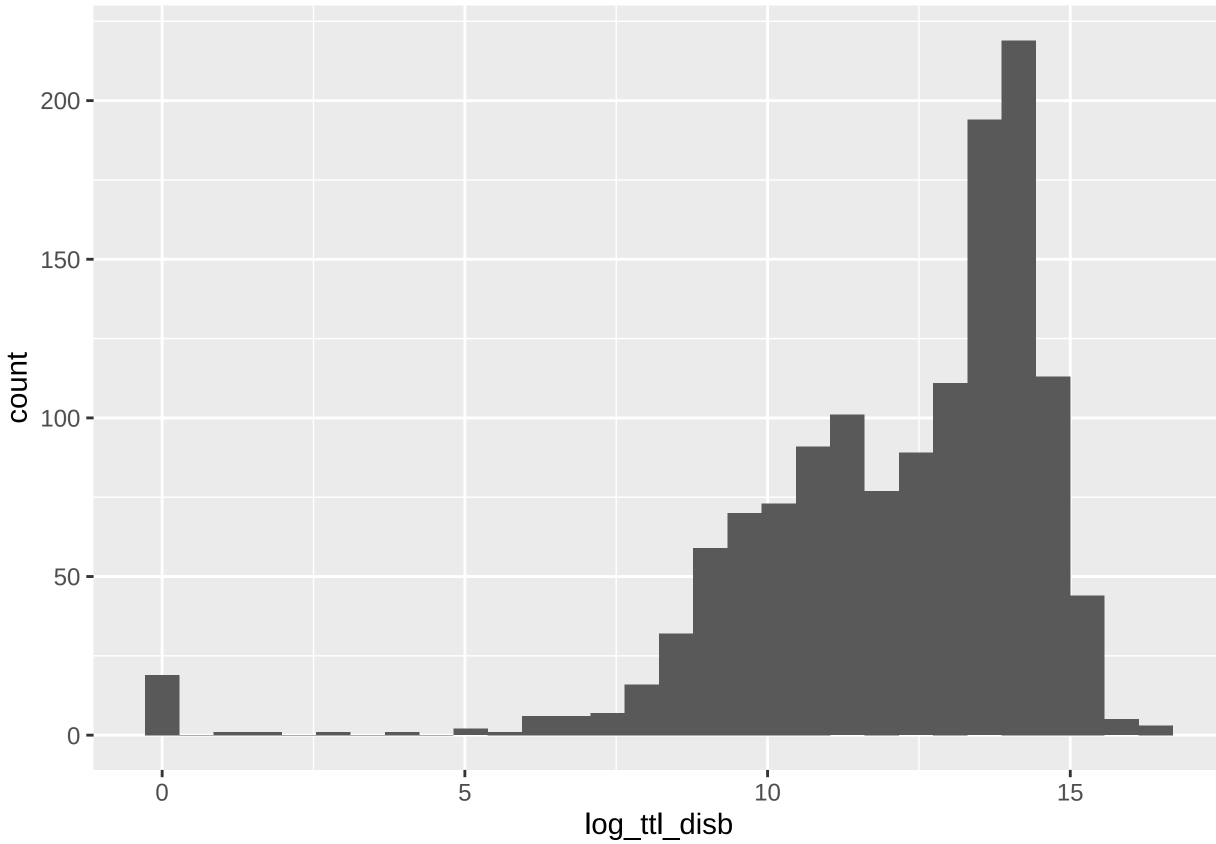
The data are independently and identically distributed, as each observation is drawn from the same underlying distribution of candidates. Each candidate's campaign information is independent of others, meaning that observing one campaign does not directly inform the outcomes of another.

Existence of the Best Linear Predictor (BLP):

The covariance terms need to be finite, so we should avoid heavy tails. However, based on the distribution observed in Question 4.1, the variable ttl_disb exhibits a very heavy tail. I am going to apply a log transformation ttl_disb in order to smooth out the tails and better satisfy the assumption that there are no infinite variances. There are a lot of values of 0 though, which would be undefined, so we're setting those to 1 with the log1p function.

```r
df_q4$log_ttl_disb <- log1p(df_q4$ttl_disb)
ggplot(data = df_q4, aes(x = log_ttl_disb)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



Uniqueness of the BLP:

There is no perfect collinearity among the regressors to make E[X^TX] invertible. In other words, no explanatory variable can be expressed as a linear combination of the others. To verify this, a correlation test was conducted between ttl_disb and general_votes, yielding a correlation coefficient of 0.40. This indicates that there is no perfect collinearity, so the log of ttl_disb cannot be written as a linear combination of general_votes, and vice versa.

```r
cor(df_q4$log_ttl_disb, df_q4$general_votes, use = "complete.obs")
```

```
## [1] 0.4000912
```

```r
model_1 <- lm(general_votes ~ log_ttl_disb + candidate_party , data = df_q4)
model_1
```

```
##
## Call:
## lm(formula = general_votes ~ log_ttl_disb + candidate_party,
##     data = df_q4)
##
## Coefficients:
##               (Intercept)                log_ttl_disb
##                     36.66                    12017.04
## candidate_partyOther Party   candidate_partyRepublican
##                -106471.52                     4917.22
```

Question 4.6

```r
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
robust_se <- coeftest(model_1, vcov = vcovHC(model_1))[, "Std. Error"]
```

```r
stargazer(
  model_1,
  type = 'latex',
  title = "Campaign Spending Effects on General Election Votes By Party",
  se = list(robust_se),
  covariate.labels = c(
    "Log Effect of Campaign Spending",
    "Vote Difference for Other Parties (vs. Democrats)",
    "Vote Difference for Republicans (vs. Democrats)",
    "Baseline Vote Count (Democrat) with No Campaign Spending"
  )
)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Wed, Oct 29, 2025 - 5:11:49 PM

Question 4.7

```r
model_2 <- lm(general_votes ~ ttl_disb , data = df_q4)
model_2
```

```
##
## Call:
## lm(formula = general_votes ~ ttl_disb, data = df_q4)
##
## Coefficients:
## (Intercept)     ttl_disb
##   1.213e+05    1.439e-02
```

```r
anova(model_2, model_1, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: general_votes ~ ttl_disb
```

Table 1: Campaign Spending Effects on General Election Votes By Party

	Dependent variable:
	general_votes
Log Effect of Campaign Spending	12,017.040***
	(1,072.881)
Vote Difference for Other Parties (vs. Democrats)	−106,471.500***
	(8,322.458)
Vote Difference for Republicans (vs. Democrats)	4,917.221
	(3,769.448)
Baseline Vote Count (Democrat) with No Campaign Spending	36.659
	(14,050.610)
Observations	880
R^2	0.426
Adjusted R^2	0.424
Residual Std. Error	61,033.930 (df = 876)
F Statistic	216.495*** (df = 3; 876)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
## Model 2: general_votes ~ log_ttl_disb + candidate_party
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1    878 5.3943e+12
## 2    876 3.2632e+12  2 2.1311e+12 286.04 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 4.8

```r
model_3 <- lm(general_votes ~ candidate_party , data = df_q4)
model_3
```

```
##
## Call:
## lm(formula = general_votes ~ candidate_party, data = df_q4)
##
## Coefficients:
##                (Intercept)  candidate_partyOther Party
##                     152439                     -111934
##  candidate_partyRepublican
##                      11003
```

```r
anova(model_3, model_1, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: general_votes ~ candidate_party
## Model 2: general_votes ~ log_ttl_disb + candidate_party
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
```

```
## 1    877 3.8839e+12
## 2    876 3.2632e+12  1 6.2072e+11 166.63 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model_1, vcov = vcovHC(model_1))
```

```
##
## t test of coefficients:
##
##                               Estimate  Std. Error  t value Pr(>|t|)
## (Intercept)                     36.659   14050.611   0.0026   0.9979
## log_ttl_disb                 12017.039    1072.881  11.2007   <2e-16 ***
## candidate_partyOther Party -106471.517    8322.458 -12.7933   <2e-16 ***
## candidate_partyRepublican     4917.221    3769.448   1.3045   0.1924
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Office Hours:

If worried about skewness for the statistic you're running, do a log-transform. (I think in Q4.5?)

Checking for collinearity: run cor(); remove one of the collinear variables and then compare the coefficiencts to check if they're the same after running one and removing the other.

Last two parts of question 4

Use coeftest(model_x, vcovHC(model_x)) in library(sandwich) and some other library to evaluate robust standard errors.

Set either Republican or Democrat as the baseline, not the "other" label

Run an f test that compares the last two models you create.