# STA2201_Lab2

Alice Huang

18/01/2023

## Downloading packages and data

```
library(opendatatoronto)
library(tidyverse)
library(stringr)
library(skimr) # EDA
library(visdat) # EDA
library(janitor)
library(lubridate)
library(ggrepel)

all_data <- list_packages(limit = 500)

res <- list_package_resources("996cfe8d-fb35-40ce-b569-698d51fc683b") # obtained code from searching da
res <- res %>% mutate(year = str_extract(name, "202.?"))
delay_2022_ids <- res %>% filter(year==2022) %>% select(id) %>% pull()

delay_2022 <- get_resource(delay_2022_ids)

# make the column names nicer to work with
delay_2022 <- clean_names(delay_2022)

# note: I obtained these codes from the 'id' column in the `res` object above
delay_codes <- get_resource("3900e649-f31e-4b79-9f20-4731bbfd94f7")
delay_data_codebook <- get_resource("ca43ac3d-3940-4315-889b-a9375e7b8aa4")
```

## Lab Exercises

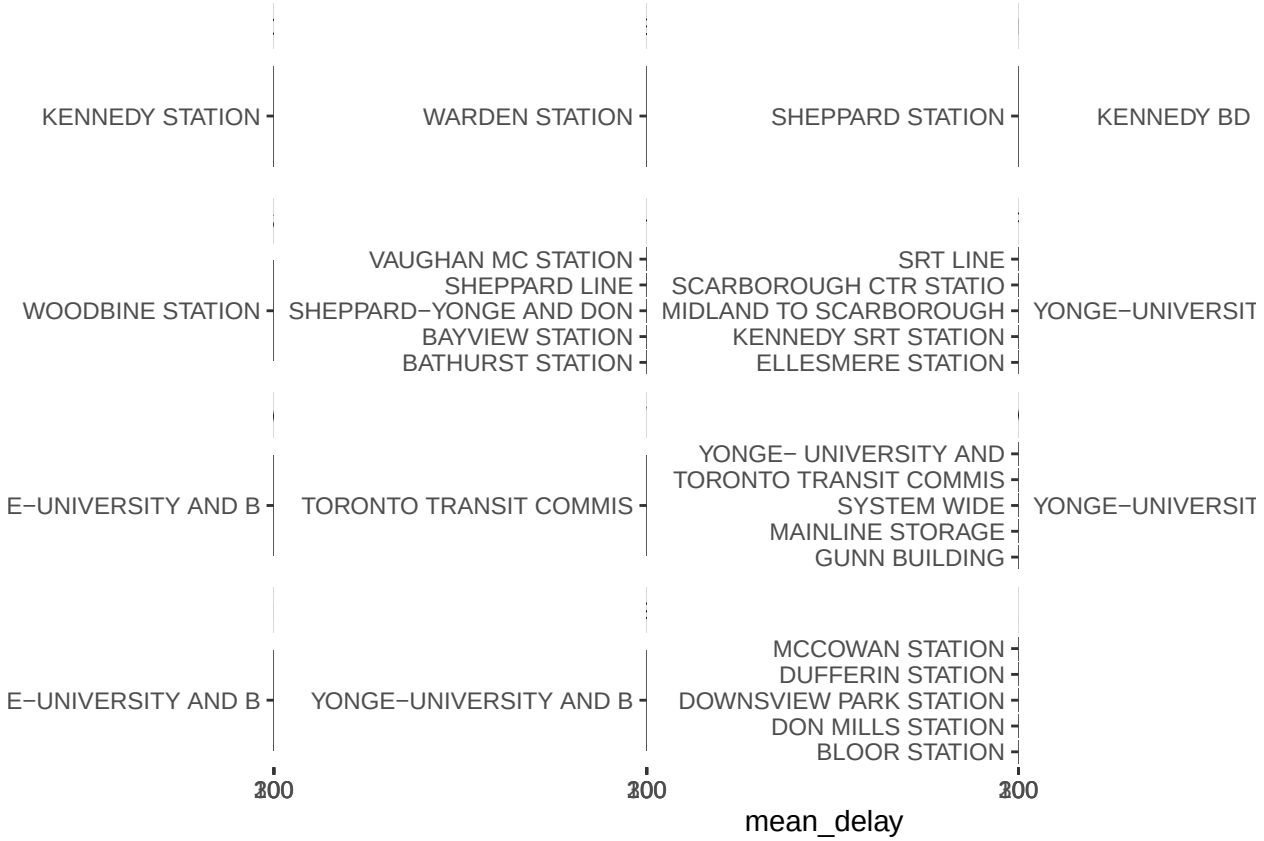
To be handed in via submission of quarto file (and rendered pdf) to GitHub.

1. Using the `delay_2022` data, plot the five stations with the highest mean delays. Facet the graph by `line`

```
delay_2022 %>%
  group_by(line, station) %>%
  summarise(mean_delay = mean(min_delay)) %>%
  arrange(-mean_delay) %>%
```

```
  slice(1:5) %>%
  ggplot(aes(x = station,
             y = mean_delay)) +
  geom_col() +
  facet_wrap(vars(line),
             scales = "free_y",
             nrow = 4) +
  coord_flip()
```

```
## `summarise()` has grouped output by 'line'. You can override using the
## `.groups` argument.
```




2. Using the `opendatatoronto` package, download the data on mayoral campaign contributions for 2014. Hints:
   - find the ID code you need for the package you need by searching for 'campaign' in the `all_data` tibble above
   - you will then need to `list_package_resources` to get ID for the data file
   - note: the 2014 file you will get from `get_resource` has a bunch of different campaign contributions, so just keep the data that relates to the Mayor election

```
all_data %>% filter(title=="Elections - Campaign Contributions - 2014 to 2017") %>%
  select(id) %>% pull -> all_data_id
```

```
dflist <- list_package_resources(all_data_id) # obtained code from searching data frame above

camp2014 <- get_resource("5b230e92-0a22-4a15-9572-0b19cc222985")
```

```
## New names:
## New names:
## New names:
## New names:
## New names:
## New names:
## New names:
## * `` -> `...2`
## * `` -> `...3`
```

```
mayorcamp2014 <- camp2014$`2_Mayor_Contributions_2014_election.xls`
```

3. Clean up the data format (fixing the parsing issue and standardizing the column names using `janitor`)

```
row_to_names(mayorcamp2014, 1, TRUE, TRUE) -> mayorcamp2014
clean_names(mayorcamp2014) -> mayorcamp2014
head(mayorcamp2014)
```

```
## # A tibble: 6 x 13
##   contributors~1 contr~2 contr~3 contr~4 contr~5 goods~6 contr~7 relat~8 presi~9
##   <chr>          <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
## 1 A D'Angelo, T~ <NA>    M6A 1P5 300     Moneta~ <NA>    Indivi~ <NA>    <NA>
## 2 A Strazar, Ma~ <NA>    M2M 3B8 300     Moneta~ <NA>    Indivi~ <NA>    <NA>
## 3 A'Court, K Su~ <NA>    M4M 2J8 36      Moneta~ <NA>    Indivi~ <NA>    <NA>
## 4 A'Court, K Su~ <NA>    M4M 2J8 100     Moneta~ <NA>    Indivi~ <NA>    <NA>
## 5 A'Court, K Su~ <NA>    M4M 2J8 100     Moneta~ <NA>    Indivi~ <NA>    <NA>
## 6 Aaron, Robert~ <NA>    M6B 1H7 250     Moneta~ <NA>    Indivi~ <NA>    <NA>
## # ... with 4 more variables: authorized_representative <chr>, candidate <chr>,
## #   office <chr>, ward <chr>, and abbreviated variable names
## #   1: contributors_name, 2: contributors_address, 3: contributors_postal_code,
## #   4: contribution_amount, 5: contribution_type_desc,
## #   6: goods_or_service_desc, 7: contributor_type_desc,
## #   8: relationship_to_candidate, 9: president_business_manager
```

4. Summarize the variables in the dataset. Are there missing values, and if so, should we be worried about them? Is every variable in the format it should be? If not, create new variable(s) that are in the right format.

```
summary(mayorcamp2014)
```

```
##  contributors_name  contributors_address contributors_postal_code
##  Length:10199       Length:10199         Length:10199
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##  contribution_amount contribution_type_desc goods_or_service_desc
##  Length:10199        Length:10199           Length:10199
```

```
##  Class :character    Class :character       Class :character
##  Mode  :character    Mode  :character       Mode  :character
##  contributor_type_desc relationship_to_candidate president_business_manager
##  Length:10199          Length:10199              Length:10199
##  Class :character      Class :character          Class :character
##  Mode  :character      Mode  :character          Mode  :character
##  authorized_representative  candidate            office
##  Length:10199              Length:10199       Length:10199
##  Class :character          Class :character   Class :character
##  Mode  :character          Mode  :character   Mode  :character
##      ward
##  Length:10199
##  Class :character
##  Mode  :character
```

```
skim(mayorcamp2014)
```

Table 1: Data summary

| | |
|---|---|
| Name | mayorcamp2014 |
| Number of rows | 10199 |
| Number of columns | 13 |
| Column type frequency: | |
| character | 13 |
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| contributors_name | 0 | 1 | 4 | 31 | 0 | 7545 | 0 |
| contributors_address | 10197 | 0 | 24 | 26 | 0 | 2 | 0 |
| contributors_postal_code | 0 | 1 | 7 | 7 | 0 | 5284 | 0 |
| contribution_amount | 0 | 1 | 1 | 18 | 0 | 209 | 0 |
| contribution_type_desc | 0 | 1 | 8 | 14 | 0 | 2 | 0 |
| goods_or_service_desc | 10188 | 0 | 11 | 40 | 0 | 9 | 0 |
| contributor_type_desc | 0 | 1 | 10 | 11 | 0 | 2 | 0 |
| relationship_to_candidate | 10166 | 0 | 6 | 9 | 0 | 2 | 0 |
| president_business_manager | 10197 | 0 | 13 | 16 | 0 | 2 | 0 |
| authorized_representative | 10197 | 0 | 13 | 16 | 0 | 2 | 0 |
| candidate | 0 | 1 | 9 | 18 | 0 | 27 | 0 |
| office | 0 | 1 | 5 | 5 | 0 | 1 | 0 |
| ward | 10199 | 0 | NA | NA | 0 | 0 | 0 |

There are 10197 missing values in `contributors_address` column. I wouldn't be worried about this as this information was probably hidden for privacy reasons.

There are 10188 missing values in the `goods_or_service_desc` column. There are 10166 missing values in the `relationship_to_candidate` column. There are 10197 missing values in the

`president_business_manager` column. There are 10197 missing values in the `authorized_representative` column and 10199 missing values in the `ward` column. This likely means that we will not be able to consider these variables with lots of missing data in our data analysis.

The contribution_amount variable is in character format, so we should change it to numeric format.

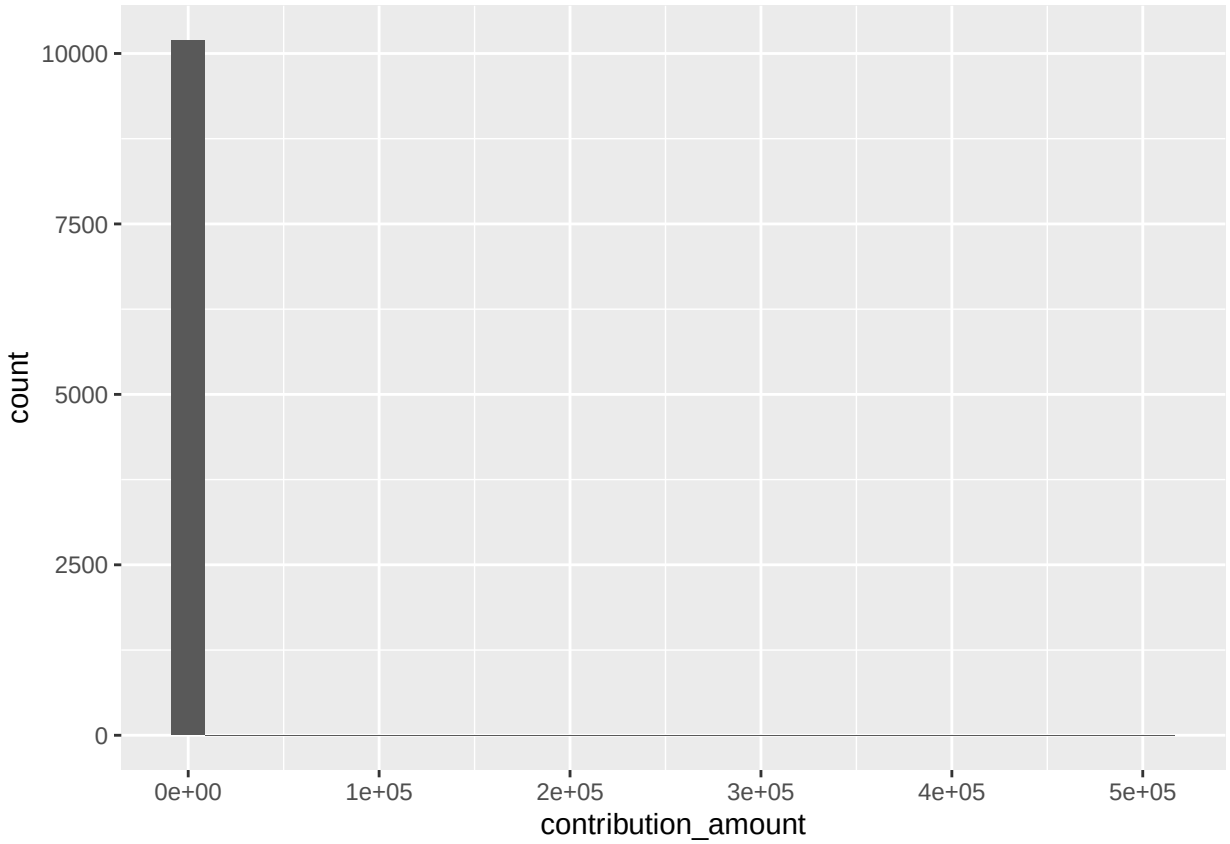
```
mayorcamp2014$contribution_amount <- as.numeric(mayorcamp2014$contribution_amount)
```

5. Visually explore the distribution of values of the contributions. What contributions are notable outliers? Do they share a similar characteristic(s)? It may be useful to plot the distribution of contributions without these outliers to get a better sense of the majority of the data.

Here is the distribution of contribution amounts.

```
mayorcamp2014 %>% ggplot(aes(x=contribution_amount)) + geom_histogram()
```

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



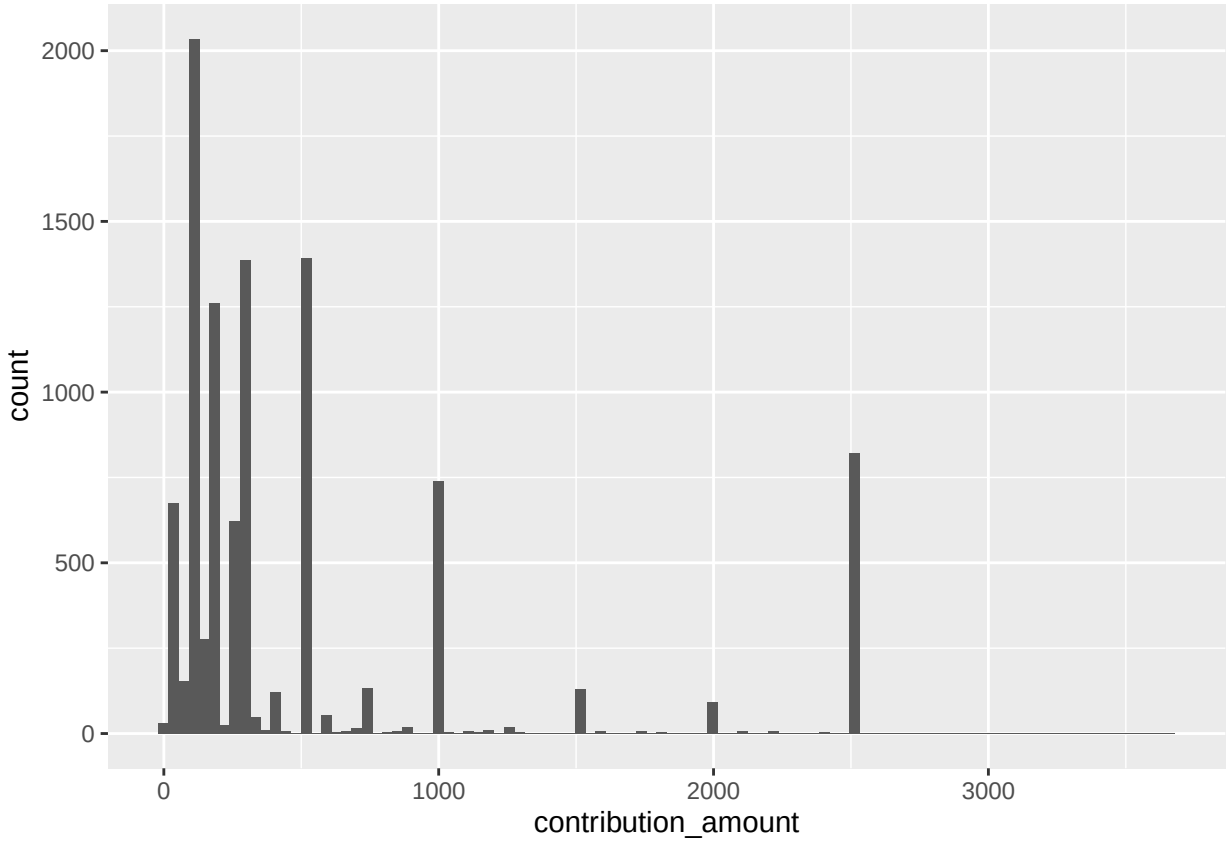

We notice that there are outliers in the contribution amount. Upon closer inspection of the data, it appears that the outliers are from candidates donating to their own campaign.

```
mayorcamp2014 %>% arrange(desc(contribution_amount)) %>%
  select(contributors_name, contribution_amount, relationship_to_candidate) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##    contributors_name contribution_amount relationship_to_candidate
##    <chr>                           <dbl> <chr>
##  1 Ford, Doug                    508225. Candidate
##  2 Ford, Rob                      78805. Candidate
##  3 Ford, Doug                     50000  Candidate
##  4 Ford, Rob                      50000  Candidate
##  5 Ford, Rob                      50000  Candidate
##  6 Goldkind, Ari                  23624. Candidate
##  7 Ford, Rob                      20000  Candidate
##  8 Ford, Rob                      12210  Candidate
##  9 Di Paola, Rocco                 6000  Candidate
## 10 Thomson, Sarah                  4426. Candidate
```

Let's see what distribution looks like if we filter out instances where candidates donated to their own campaign. The amounts that candidates' spouses donated were in the higher end, but they were not significantly higher than the rest of the donations, so I left them in there.

```
cand_rels <- mayorcamp2014 %>% select(relationship_to_candidate) %>% unique()
mayorcamp2014_nocand <- mayorcamp2014 %>% filter(relationship_to_candidate == "Spouse" | is.na(relations
mayorcamp2014_nocand %>% ggplot(aes(x=contribution_amount)) + geom_histogram(bins = 100)
```




6. List the top five candidates in each of these categories:
    - total contributions

- mean contribution
- number of contributions

```
mayorcamp2014 %>% group_by(candidate) %>%
  summarise(total_contributions = sum(contribution_amount, na.rm=TRUE)) %>%
  arrange(desc(total_contributions)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate     total_contributions
##   <chr>                       <dbl>
## 1 Tory, John                2767869.
## 2 Chow, Olivia              1638266.
## 3 Ford, Doug                 889897.
## 4 Ford, Rob                  387648.
## 5 Stintz, Karen              242805
```

```
mayorcamp2014 %>% group_by(candidate) %>%
  summarise(mean_contributions = mean(contribution_amount, na.rm=TRUE)) %>%
  arrange(desc(mean_contributions)) %>%
  slice(1:5)
```

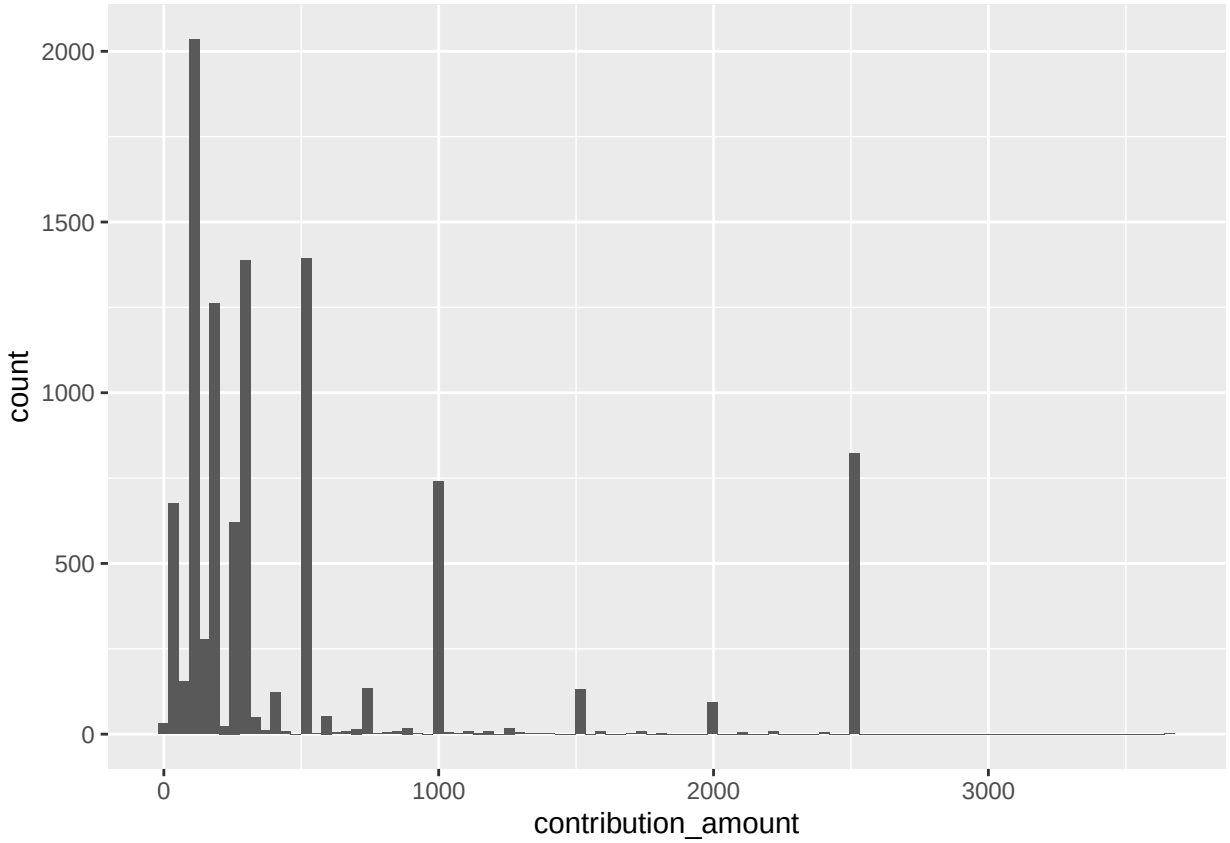
```
## # A tibble: 5 x 2
##   candidate        mean_contributions
##   <chr>                         <dbl>
## 1 Sniedzins, Erwin              2025
## 2 Syed, Hïmy                    2018
## 3 Ritch, Carlie                 1887.
## 4 Ford, Doug                    1456.
## 5 Clarke, Kevin                 1200
```

```
mayorcamp2014 %>% group_by(candidate) %>%
  summarise(number_contributions = n()) %>%
  arrange(desc(number_contributions)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate       number_contributions
##   <chr>                          <int>
## 1 Chow, Olivia                    5708
## 2 Tory, John                      2602
## 3 Ford, Doug                       611
## 4 Ford, Rob                        538
## 5 Soknacki, David                  314
```

7. Repeat 5 but without contributions from the candidates themselves.

```
cand_rels <- mayorcamp2014 %>% select(relationship_to_candidate) %>% unique()
mayorcamp2014_nocand <- mayorcamp2014 %>% filter(relationship_to_candidate == "Spouse" | is.na(relations
mayorcamp2014_nocand %>% ggplot(aes(x=contribution_amount)) + geom_histogram(bins = 100)
```




```
mayorcamp2014_nocand %>% group_by(candidate) %>%
  summarise(total_contributions = sum(contribution_amount, na.rm=TRUE)) %>%
  arrange(desc(total_contributions)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate     total_contributions
##   <chr>                       <dbl>
## 1 Tory, John                2765369.
## 2 Chow, Olivia              1635766.
## 3 Ford, Doug                 331173.
## 4 Stintz, Karen              242805
## 5 Ford, Rob                  174510.
```

```
mayorcamp2014_nocand %>% group_by(candidate) %>%
  summarise(mean_contributions = mean(contribution_amount, na.rm=TRUE)) %>%
  arrange(desc(mean_contributions)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate         mean_contributions
##   <chr>                          <dbl>
## 1 Ritch, Carlie                  1887.
## 2 Sniedzins, Erwin               1867.
## 3 Tory, John                     1063.
```

```
## 4 Gardner, Norman                1000
## 5 Tiwari, Ramnarine              1000
```

```
mayorcamp2014_nocand %>% group_by(candidate) %>%
  summarise(number_contributions = n()) %>%
  arrange(desc(number_contributions)) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate       number_contributions
##   <chr>                          <int>
## 1 Chow, Olivia                    5707
## 2 Tory, John                      2601
## 3 Ford, Doug                       608
## 4 Ford, Rob                        531
## 5 Soknacki, David                  314
```

8. How many contributors gave money to more than one candidate?

```
mayorcamp2014 %>% group_by(contributors_name) %>%
  summarise(n_candidates = n_unique(candidate)) %>%
  filter(n_candidates > 1) %>%
  summarise(num_contributors = n())
```

```
## # A tibble: 1 x 1
##   num_contributors
##              <int>
## 1              184
```