Lab 2

Roxanne Li

2024-01-18

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

Q1

Downloading the data on TTC subway delays in 2022.

```
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)
delay_codes <- get_resource("3900e649-f31e-4b79-9f20-4731bbfd94f7")
```

Preprocessing the data according to **2__eda__dataviz__additions.qmd**:

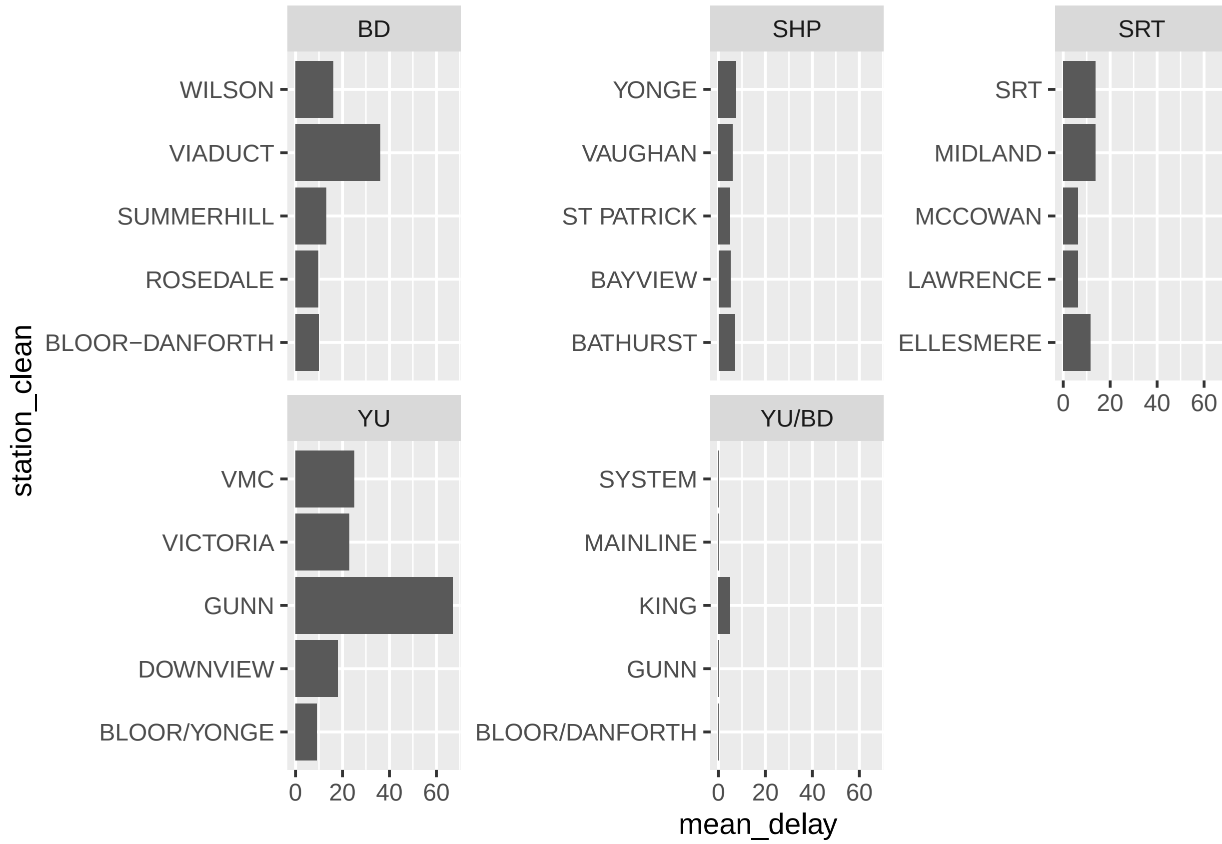
```
## remove duplicates
delay_2022 <- delay_2022 |> distinct()

## cleaning up the YU/BD line
delay_2022 <- delay_2022 |>
  mutate(contains_yu_bd = str_detect(str_to_lower(line), "bd")&str_detect(str_to_lower(line), "yu")) |>
  mutate(line = ifelse(contains_yu_bd, ifelse(line=="YU/BD", line, "YU/BD"), line))  |>
  select(-contains_yu_bd)

## removing non-standardized lines
delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT", "YU/BD"))

## left-joining delay codes
delay_2022 <- delay_2022 |>
  left_join(delay_codes |> rename(code = `SUB RMENU CODE`, code_desc = `CODE DESCRIPTION...3`) |> selec
```

```r
delay_2022 <- delay_2022 |>
  mutate(code_srt = ifelse(line=="SRT", code, "NA")) |>
  left_join(delay_codes |> rename(code_srt = `SRT RMENU CODE`, code_desc_srt = `CODE DESCRIPTION...7`)
  mutate(code = ifelse(code_srt=="NA", code, code_srt),
         code_desc = ifelse(is.na(code_desc_srt), code_desc, code_desc_srt)) |>
  select(-code_srt, -code_desc_srt)

## cleaning up station names
delay_2022 <- delay_2022 |>
  mutate(station_clean = ifelse(str_starts(station, "ST"), word(station, 1,2), word(station, 1)))

head(delay_2022)
```

```
## # A tibble: 6 x 12
##   date                time  day      station code  min_delay min_gap bound line
##   <dttm>              <chr> <chr>    <chr>   <chr>     <dbl>   <dbl> <chr> <chr>
## 1 2022-01-01 00:00:00 15:59 Saturday LAWREN~ SRDP          0       0 N     SRT
## 2 2022-01-01 00:00:00 02:23 Saturday SPADIN~ MUIS          0       0 <NA>  BD
## 3 2022-01-01 00:00:00 22:00 Saturday KENNED~ MRO           0       0 <NA>  SRT
## 4 2022-01-01 00:00:00 02:28 Saturday VAUGHA~ MUIS          0       0 <NA>  YU
## 5 2022-01-01 00:00:00 02:34 Saturday EGLINT~ MUATC         0       0 S     YU
## 6 2022-01-01 00:00:00 05:40 Saturday QUEEN ~ MUNCA         0       0 <NA>  YU
## # i 3 more variables: vehicle <dbl>, code_desc <chr>, station_clean <chr>
```

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

```r
delay_2022 |>
  group_by(line, station_clean) |>
  summarise(mean_delay = mean(min_delay)) |>
  arrange(line, desc(mean_delay))|>
  slice(1:5) |>
  ggplot(aes(x = station_clean, y = mean_delay)) +
    geom_col() +
    facet_wrap(vars(line), scales = "free_y") +
    coord_flip()
```



Q2

Restrict the `delay_2022` to delays that are greater than 0 and to only have delay reasons that appear in the top 50% of most frequent delay reasons. Perform a regression to study the association between delay minutes, and two covariates: line and delay reason. It's up to you how to specify the model, but make sure it's appropriate to the data types. Comment briefly on the results, including whether results generally agree with the exploratory data analysis above.

Preparing the data:

```r
# get the top 50% of most frequent delay reasons (exclude NAs)
most_frequent_delay_reasons <- delay_2022 |>
  filter(!is.na(code_desc), min_delay>0) |>
  group_by(code_desc) |>
  summarise(n_obs = n()) |>
  arrange(-n_obs) |>
  slice(1:(n() / 2))

# filter data based on the most_frequent_delay_reasons
data <- delay_2022 |>
  filter(min_delay>0 & code_desc %in% most_frequent_delay_reasons$code_desc)

# shorten the code names
data <- data |>
  mutate(code_red = case_when(
```

```
    str_starts(code_desc, "No") ~ word(code_desc, 1, 2),
    str_starts(code_desc, "Operator") ~ word(code_desc, 1,2),
    TRUE ~ word(code_desc,1))
         )
```

Using PCA to create the principal components that represent information about the delay reasons:

```
dwide <- data |>
  group_by(line, station_clean, code_red) |>
  summarise(n_delay = n()) |>
  pivot_wider(names_from = code_red, values_from = n_delay) |>
  mutate(
    across(everything(), ~ replace_na(.x, 0))
  )

delay_pca <- prcomp(dwide[,3:ncol(dwide)])

df_out <- as_tibble(delay_pca$x)
df_out <- bind_cols(dwide |> select(line, station_clean), df_out)
PCA_data <- df_out[, 1:4]
```

Merging two principal components representing the delay reasons with the original data:

```
merged_df <- merge(data, PCA_data, by = c("line", "station_clean"))
```

Fitting a linear regression model to the line variable as a factor and the two principal components of delay reasons:

$$Y_i = \beta_0 + \beta_1 Z_i + \beta_2 X_{1i} + \beta_3 X_{2i}$$

where Y_i is the delay minutes, Z_i is the factor variable of line, X_{1i} and X_{2i} are the principal components.

```
merged_df$line <- as.factor(merged_df$line)

model <- lm(min_delay ~ line + PC1 + PC2, data=merged_df)
summary(model)
```

```
##
## Call:
## lm(formula = min_delay ~ line + PC1 + PC2, data = merged_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -7.77  -4.45  -2.56  -0.02 440.21
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.546158   0.268394  31.842  < 2e-16 ***
## lineSHP     -0.654947   0.779781  -0.840  0.40098
## lineSRT      2.016967   0.699843   2.882  0.00396 **
## lineYU      -0.561222   0.371017  -1.513  0.13040
## lineYU/BD   -3.775776  13.894903  -0.272  0.78583
## PC1         -0.011737   0.002113  -5.555 2.86e-08 ***
```

```
## PC2         -0.011327   0.002574  -4.401 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.89 on 8535 degrees of freedom
## Multiple R-squared:  0.009766,   Adjusted R-squared:  0.00907
## F-statistic: 14.03 on 6 and 8535 DF,  p-value: 5.902e-16
```

The model did not fit well as the adjusted R-squared value is very low. However, the results show that the two principal components containing information about delay reasons are significant. Besides, the variable for line SRT is significant and it agrees with the EDA that the SRT line in general has longer delays.

Q3

Using the `opendatatoronto` package, download the data on mayoral campaign contributions for 2014 and clean it up. 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 + clean up the data format (fixing the parsing issue and standardizing the column names using `janitor`)

```r
list_package_resources("e869d365-2c15-4893-ad2a-744ca867be3b") # obtained code from searching data fram
```

```
## # A tibble: 4 x 4
##   name                               id                         format last_modified
##   <chr>                              <chr>                      <chr>  <date>
## 1 Campaign Contributions 2018 Data   5f54ab3d-44d7-4e5c-9c~ ZIP    2023-04-26
## 2 Campaign Contributions 2018 Readme eea9eecd-75ba-4a27-9f~ XLSX   2023-04-26
## 3 Campaign Contributions 2014 Data   8b42906f-c894-4e93-a9~ ZIP    2023-04-26
## 4 Campaign Contributions 2014 Readme 10158522-4f3b-4957-9f~ XLS    2023-04-26
```

Obtaining the mayor contribution data:

```r
camp_2014 <- get_resource("8b42906f-c894-4e93-a98e-acac200f34a4")
mayor <- camp_2014$"2_Mayor_Contributions_2014_election.xls"
colnames(mayor) <- as.character(mayor[1, ])
mayor <- mayor[-1, ]
mayor <- clean_names(mayor)
head(mayor)
```

```
## # A tibble: 6 x 13
##   contributors_name  contributors_address contributors_postal_code
##   <chr>              <chr>                <chr>
## 1 A D'Angelo, Tullio <NA>                 M6A 1P5
## 2 A Strazar, Martin  <NA>                 M2M 3B8
## 3 A'Court, K Susan   <NA>                 M4M 2J8
## 4 A'Court, K Susan   <NA>                 M4M 2J8
## 5 A'Court, K Susan   <NA>                 M4M 2J8
## 6 Aaron, Robert B    <NA>                 M6B 1H7
## # i 10 more variables: contribution_amount <chr>, contribution_type_desc <chr>,
## #   goods_or_service_desc <chr>, contributor_type_desc <chr>,
## #   relationship_to_candidate <chr>, president_business_manager <chr>,
## #   authorized_representative <chr>, candidate <chr>, office <chr>, ward <chr>
```

Q4

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.

The table below shows the number and percentage of missing values in each column.

```r
skim(mayor)
```

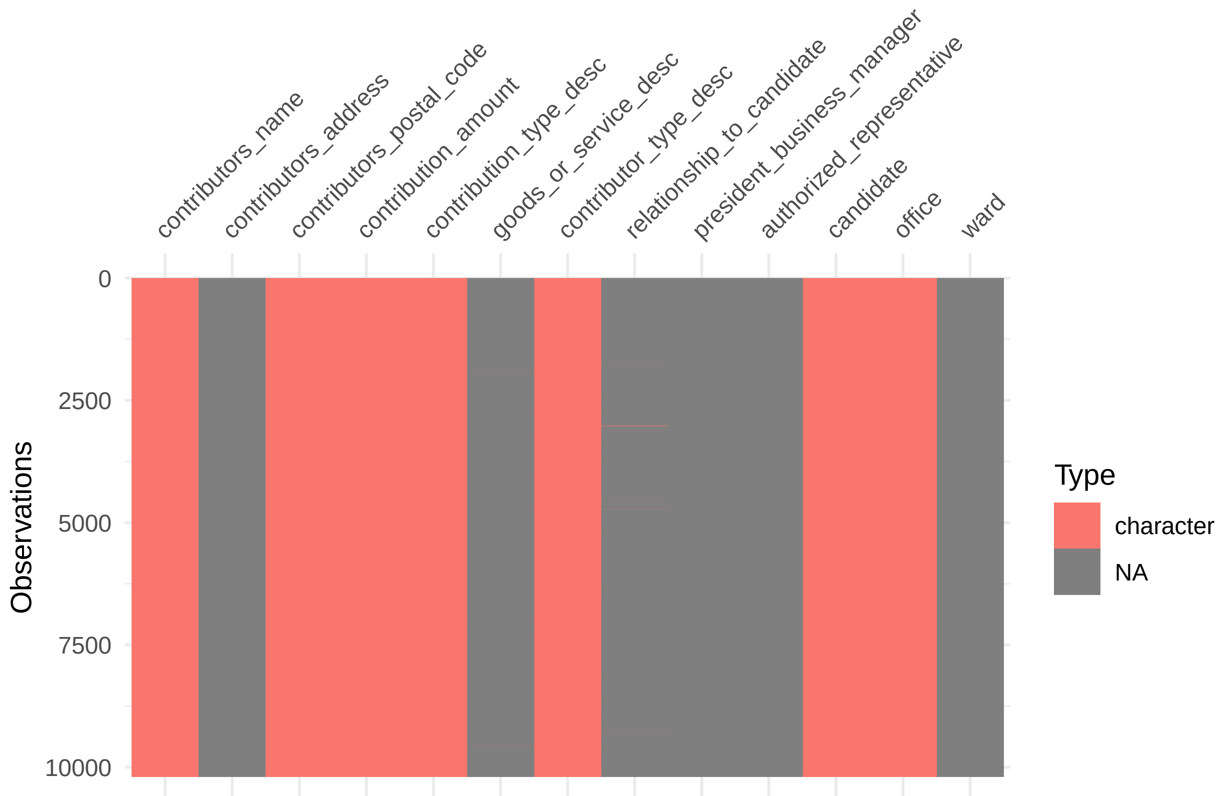
Table 1: Data summary

Name	mayor
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

```r
vis_dat(mayor)
```



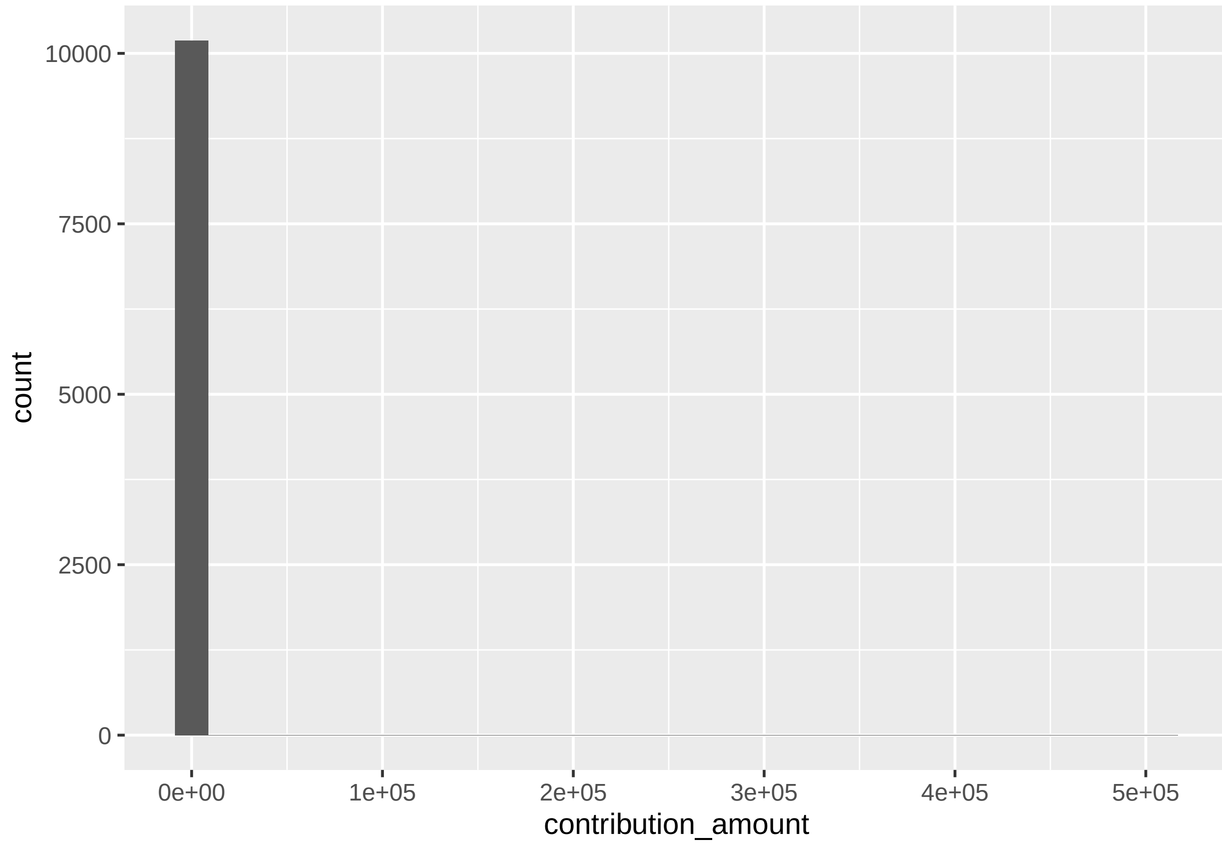
There are 6 columns in the dataset that have almost all the values missing, which is worrisome if we want to analyze for example, the address of the contributors and the relationship to candidate. Not all columns are in the correct data type.

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

Q5

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.

```
mayor |>
  ggplot(aes(x = contribution_amount)) +
   geom_histogram()
```

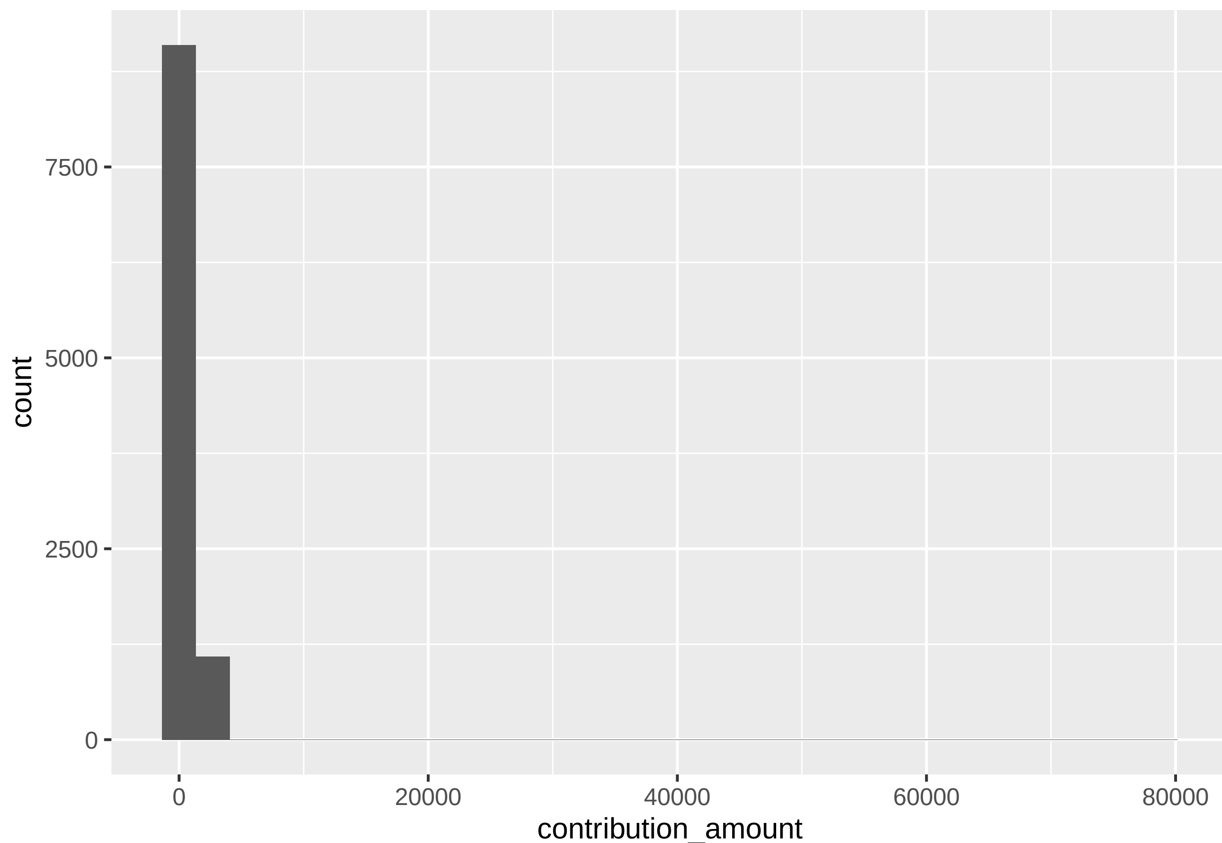



```
mayor |>
  filter(contribution_amount > 1e+05)
```

```
## # A tibble: 1 x 13
##   contributors_name contributors_address contributors_postal_code
##   <chr>             <chr>                <chr>
## 1 Ford, Doug        <NA>                 M9A 2C3
## # i 10 more variables: contribution_amount <dbl>, contribution_type_desc <chr>,
## #   goods_or_service_desc <chr>, contributor_type_desc <chr>,
## #   relationship_to_candidate <chr>, president_business_manager <chr>,
## #   authorized_representative <chr>, candidate <chr>, office <chr>, ward <chr>
```

The contribution by Doug Ford is an outlier so we exclude it from the data and plot the histogram again.

```
mayor |>
  filter(contribution_amount <= 1e+05) |>
  ggplot(aes(x = contribution_amount)) +
   geom_histogram()
```



There are still outliers, so we try to set the threshold at 4000.

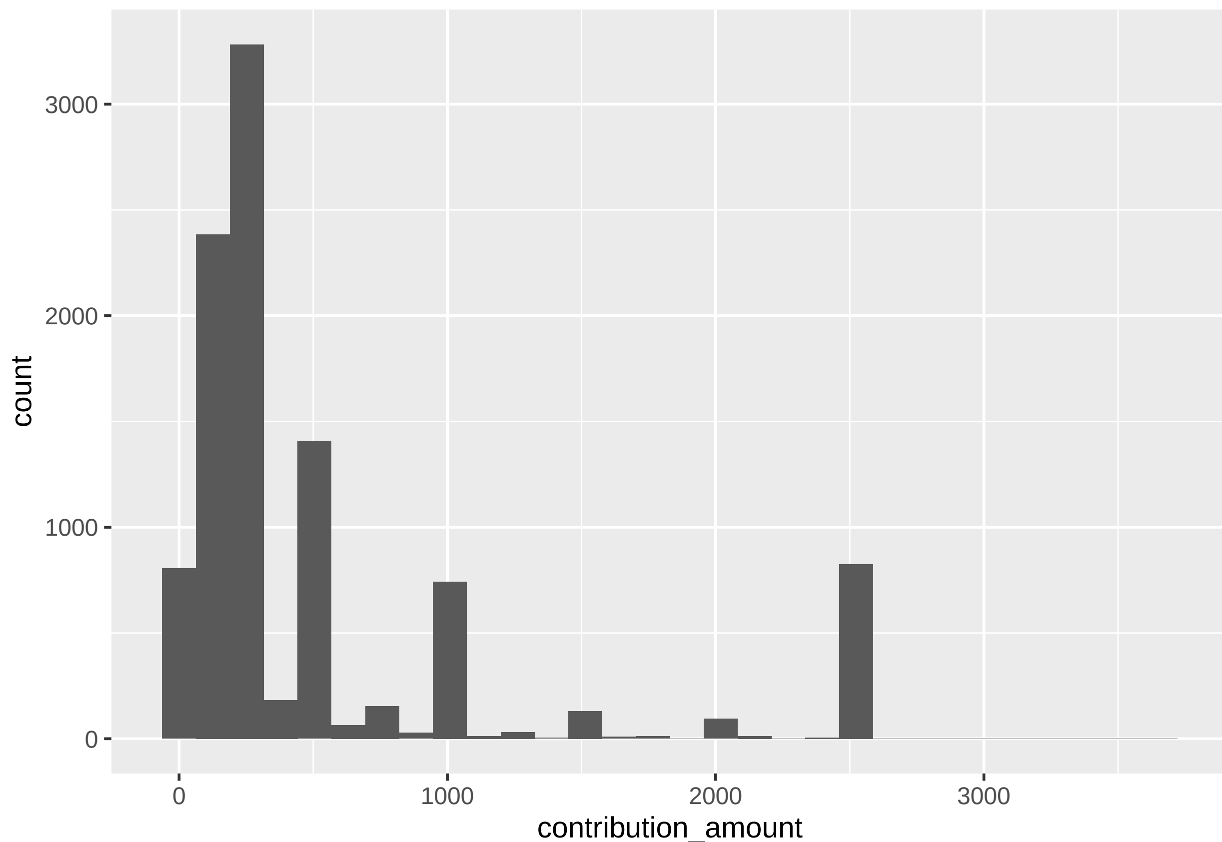
```
mayor |>
  filter(contribution_amount > 4000)
```

```
## # A tibble: 10 x 13
##    contributors_name contributors_address contributors_postal_code
##    <chr>             <chr>                <chr>
##  1 Di Paola, Rocco   <NA>                 M3H 2T1
##  2 Ford, Doug        <NA>                 M9A 2C3
##  3 Ford, Doug        <NA>                 M9A 2C3
##  4 Ford, Rob         <NA>                 M9A 3G9
##  5 Ford, Rob         <NA>                 M9A 3G9
##  6 Ford, Rob         <NA>                 M9A 3G9
##  7 Ford, Rob         <NA>                 M9A 3G9
##  8 Ford, Rob         <NA>                 M9A 3G9
##  9 Goldkind, Ari     <NA>                 M5P 1P5
## 10 Thomson, Sarah    <NA>                 M4W 2X6
## # i 10 more variables: contribution_amount <dbl>, contribution_type_desc <chr>,
## #   goods_or_service_desc <chr>, contributor_type_desc <chr>,
## #   relationship_to_candidate <chr>, president_business_manager <chr>,
## #   authorized_representative <chr>, candidate <chr>, office <chr>, ward <chr>
```

These outliers have the same characteristic that the monetary contribution comes from the mayor candidate himself/herself.

We exclude any amounts above 4000 and plot the histogram again:

```
mayor |>
  filter(contribution_amount <= 4000) |>
  ggplot(aes(x = contribution_amount)) +
   geom_histogram()
```



Excluding outliers with contribution amount above 4000, the majority of the contribution amounts are below 1000.

Q6

List the top five candidates in each of these categories: + total contributions + mean contribution + number of contributions

Top five candidates by total contributions:

```
mayor |>
  group_by(candidate) |>
  summarise(total_contribution = sum(contribution_amount)) |>
  arrange(-total_contribution) |>
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate      total_contribution
##   <chr>                       <dbl>
```

```
## 1 Tory, John              2767869.
## 2 Chow, Olivia            1638266.
## 3 Ford, Doug               889897.
## 4 Ford, Rob                387648.
## 5 Stintz, Karen            242805
```

Top five candidates by mean contribution:

```
mayor |>
  group_by(candidate) |>
  summarise(mean_contribution = sum(contribution_amount) / n()) |>
  arrange(-mean_contribution) |>
  slice(1:5)
```

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

Top five candidates by number of contributions:

```
mayor |>
  group_by(candidate) |>
  summarise(n_contribution = n()) |>
  arrange(-n_contribution) |>
  slice(1:5)
```

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

Q7

Repeat 6 but without contributions from the candidates themselves.

Top five candidates by total contributions:

```
mayor |>
  mutate(self_contribute = ifelse(contributors_name==candidate, 1, 0)) |>
  filter(self_contribute != 1) |>
  group_by(candidate) |>
  summarise(total_contribution = sum(contribution_amount)) |>
  arrange(-total_contribution) |>
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate     total_contribution
##   <chr>                      <dbl>
## 1 Tory, John              2765369.
## 2 Chow, Olivia            1634766.
## 3 Ford, Doug               331173.
## 4 Stintz, Karen            242805
## 5 Ford, Rob                174510.
```

Top five candidates by mean contribution:

```
mayor |>
  mutate(self_contribute = ifelse(contributors_name==candidate, 1, 0)) |>
  filter(self_contribute != 1) |>
  group_by(candidate) |>
  summarise(mean_contribution = sum(contribution_amount) / n()) |>
  arrange(-mean_contribution) |>
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate         mean_contribution
##   <chr>                         <dbl>
## 1 Ritch, Carlie                 1887.
## 2 Sniedzins, Erwin              1867.
## 3 Tory, John                    1063.
## 4 Gardner, Norman               1000
## 5 Tiwari, Ramnarine             1000
```

Top five candidates by number of contributions:

```
mayor |>
  mutate(self_contribute = ifelse(contributors_name==candidate, 1, 0)) |>
  filter(self_contribute != 1) |>
  group_by(candidate) |>
  summarise(n_contribution = n()) |>
  arrange(-n_contribution) |>
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate       n_contribution
##   <chr>                    <int>
## 1 Chow, Olivia              5706
## 2 Tory, John                2601
## 3 Ford, Doug                 608
## 4 Ford, Rob                  531
## 5 Soknacki, David            314
```

Q8

How many contributors gave money to more than one candidate?

```r
mutiple_cands <- mayor |>
  group_by(contributors_name) |>
  summarise(n_candidate = length(unique(candidate))) |>
  filter(n_candidate > 1)
nrow(mutiple_cands)
```

```
## [1] 184
```

184 contributors gave money to more than one candidate.