# Week 2 Lab Exercises

Kelly McConvey

23/01/23

## Table of contents



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

## 1 TTC subway delays

This package provides an interface to all data available on the Open Data Portal provided by the City of Toronto.

Use the `list_packages` function to look whats available look at what's available

```
all_data <- list_packages(limit = 500)
head(all_data)
```

```
# A tibble: 6 x 11
  title     id    topics civic~1 publi~2 excerpt datas~3 num_r~4 formats refre~5
  <chr>     <chr> <chr>  <chr>   <chr>   <chr>   <chr>     <int> <chr>   <chr>
```

```
1 Traffic ~ a330~ Trans~ <NA>    Transp~ This d~ Map          12 GPKG,S~ As ava~
2 Developm~ 0aa7~ <NA>   <NA>    City P~ This d~ Table         4 JSON,C~ Monthly
3 Resident~ 4a65~ Locat~ Mobili~ Transp~ Legall~ Table        4 JSON,C~ Weekly
4 Chemical~ ae8e~ Publi~ <NA>    Toront~ This d~ Table         6 CSV,XL~ Daily
5 Daily Sh~ 21c8~ Commu~ Afford~ Shelte~ Daily ~ Table        12 JSON,C~ Daily
6 Building~ 9425~ Devel~ Climat~ Toront~ Green ~ Table        5 JSON,X~ Weekly
# ... with 1 more variable: last_refreshed <date>, and abbreviated variable
#   names 1: civic_issues, 2: publisher, 3: dataset_category, 4: num_resources,
#   5: refresh_rate
```

Let's download the data on TTC subway delays in 2022.

```
res <- list_package_resources("996cfe8d-fb35-40ce-b569-698d51fc683b") # obtained code from
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")
```

```
New names:
* `` -> `...1`
* `CODE DESCRIPTION` -> `CODE DESCRIPTION...3`
* `` -> `...4`
* `` -> `...5`
* `CODE DESCRIPTION` -> `CODE DESCRIPTION...7`
```

```
delay_data_codebook <- get_resource("ca43ac3d-3940-4315-889b-a9375e7b8aa4")
```

```
head(delay_2022)
```

```
# A tibble: 6 x 10
  date                time  day      station   code  min_d~1 min_gap bound line
  <dttm>              <chr> <chr>    <chr>     <chr>   <dbl>   <dbl> <chr> <chr>
1 2022-01-01 00:00:00 15:59 Saturday LAWRENCE~ SRDP        0       0 N     SRT
2 2022-01-01 00:00:00 02:23 Saturday SPADINA ~ MUIS        0       0 <NA>  BD
3 2022-01-01 00:00:00 22:00 Saturday KENNEDY ~ MRO         0       0 <NA>  SRT
```

```
4 2022-01-01 00:00:00 02:28 Saturday VAUGHAN ~ MUIS        0       0 <NA>  YU
5 2022-01-01 00:00:00 02:34 Saturday EGLINTON~ MUATC       0       0 S     YU
6 2022-01-01 00:00:00 05:40 Saturday QUEEN ST~ MUNCA       0       0 <NA>  YU
# ... with 1 more variable: vehicle <dbl>, and abbreviated variable name
#   1: min_delay
```

```
## Removing the observations that have non-standardized lines
delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
```

```
delay_2022 <- delay_2022 |>
  left_join(delay_codes |> rename(code = `SUB RMENU CODE`, code_desc = `CODE DESCRIPTION..
```

```
Joining, by = "code"
```

```
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 DESCR
  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)
```

```
Joining, by = "code_srt"
```

```
delay_2022 <- delay_2022 |>
  mutate(station_clean = ifelse(str_starts(station, "ST"), word(station, 1,2), word(statio
```

## 2 Lab Exercises

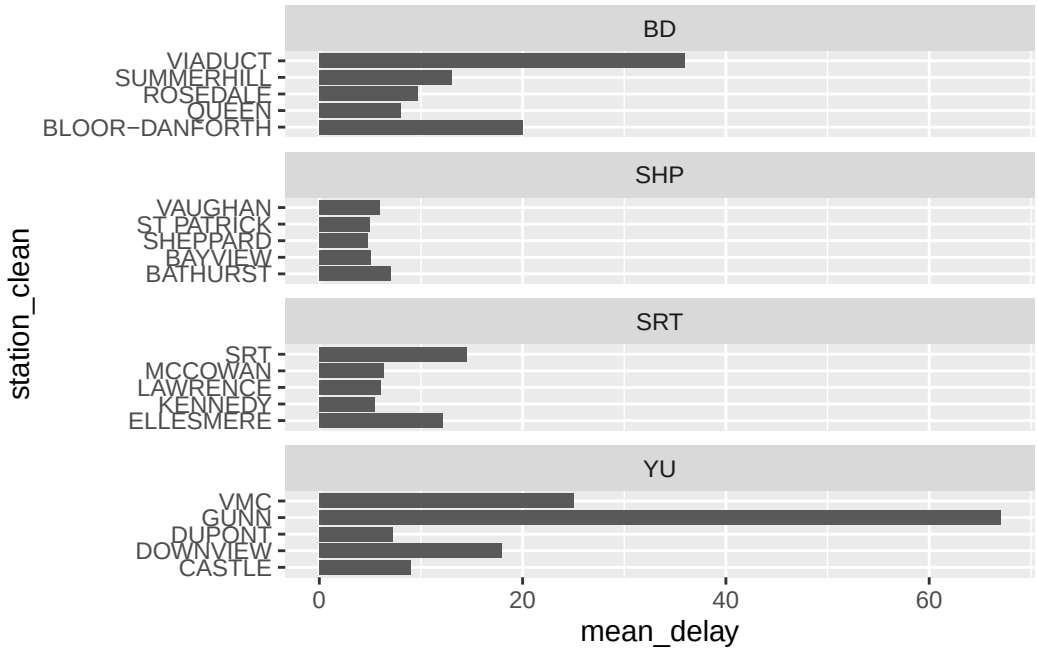
To be handed in via submission of quarto file (and rendered pdf) to GitHub.

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_clean) |>
  summarise(mean_delay = mean(min_delay)) |>
  arrange(-mean_delay) |>
```

```
  slice(1:5) |>
  ggplot(aes(x = station_clean,
             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 <- list_packages(limit = 500)
res2 <- list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c")
campaign_2014=get_resource("5b230e92-0a22-4a15-9572-0b19cc222985")
```

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

```
mayor=campaign_2014$"2_Mayor_Contributions_2014_election.xls"
head(mayor)
```

```
# A tibble: 6 x 13
  2014 Munic~1 ...2  ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11 ...12
  <chr>        <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 Contributor~ Cont~ Cont~ Cont~ Cont~ Good~ Cont~ Rela~ Pres~ Auth~ Cand~ Offi~
2 A D'Angelo,~ <NA>  M6A ~ 300   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Ford~ Mayor
3 A Strazar, ~ <NA>  M2M ~ 300   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Ford~ Mayor
4 A'Court, K ~ <NA>  M4M ~ 36    Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~ Mayor
5 A'Court, K ~ <NA>  M4M ~ 100   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~ Mayor
6 A'Court, K ~ <NA>  M4M ~ 100   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~ Mayor
# ... with 1 more variable: ...13 <chr>, and abbreviated variable name
#   1: `2014 Municipal Election - List of Contributors to Mayoralty Candidates`
```

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

```
mayor1 <- mayor |>
  row_to_names(row_number = 1) |>
  clean_names()

head(mayor1)
```

```
# 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.

```
skim(mayor1)
```

Table 1: Data summary

| | |
|---|---|
| Name | mayor1 |
| 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 |

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| 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 |

All of the variables are of the type character, so we'll create a new vairable for contritbution amount as type numeric.

```
mayor2 <- mayor1 |>
  mutate(contribution_amount_new=as.numeric(contribution_amount))
```

Check to make sure it worked:

```
skim(mayor2)
```

Table 3: Data summary

| | |
|---|---|
| Name | mayor2 |
| Number of rows | 10199 |
| Number of columns | 14 |
| _____ | |
| Column type frequency: | |
| character | 13 |
| numeric | 1 |
| _____ | |
| 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 |

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| 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 |

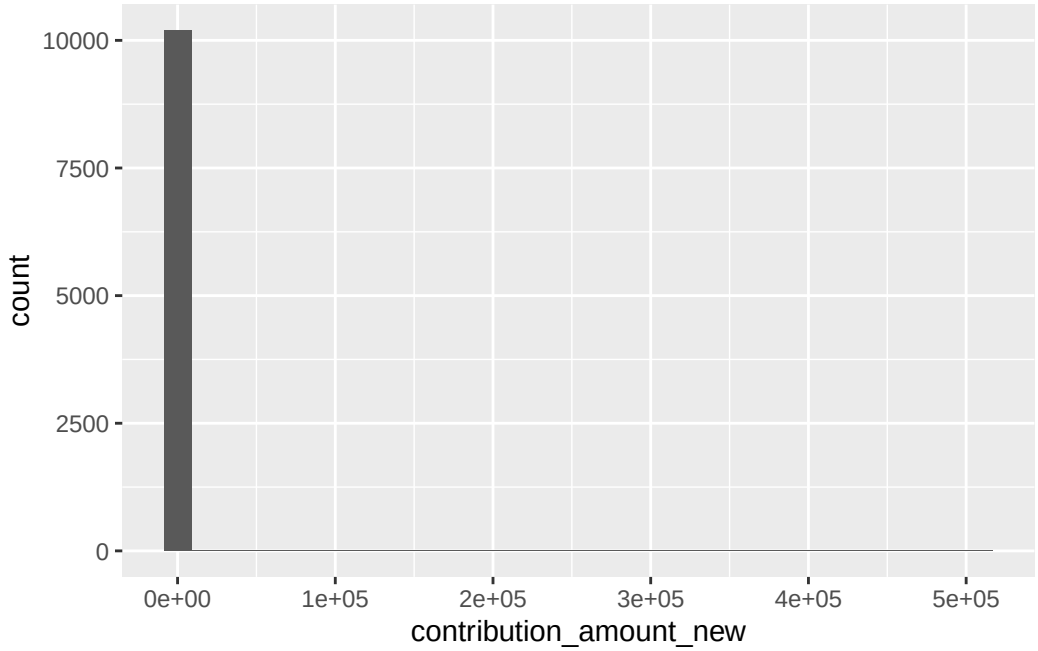
**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| contribution_amount_new | 0 | 1 | 607.95 | 5211.31 | 1 | 100 | 300 | 500 | 508224.7 | |

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.

```
ggplot(data = mayor2) +
  geom_histogram(aes(x = contribution_amount_new))
```

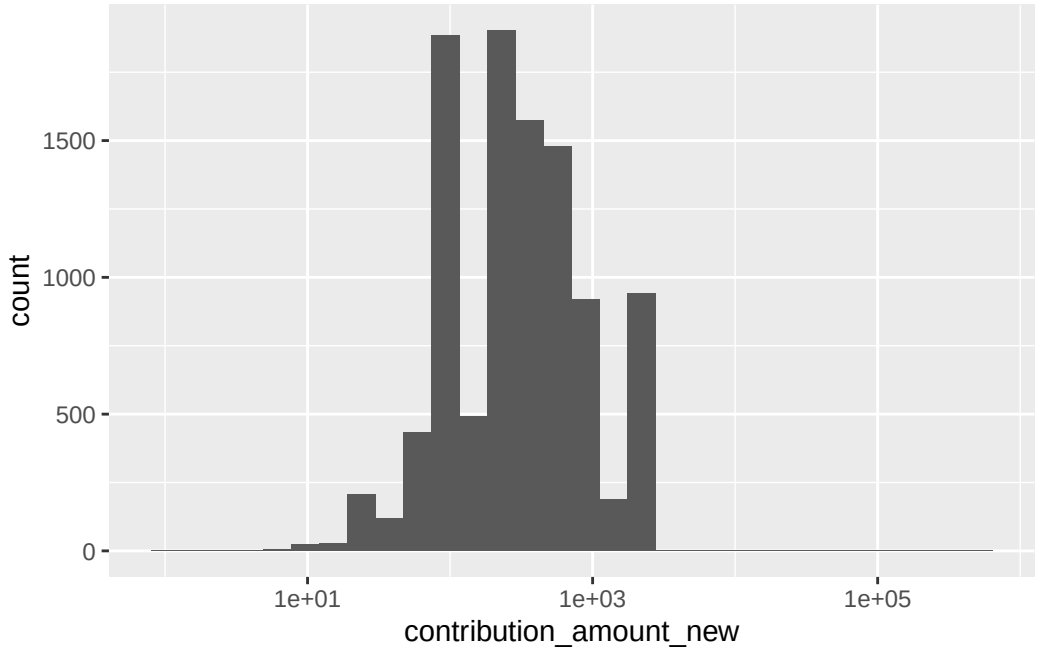
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




The outliers make it impossible to read. Let's try it with a log scale:

```
ggplot(data = mayor2) +
  geom_histogram(aes(x = contribution_amount_new)) +
  scale_x_log10()
```

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




Looks like some significant outliers at the high end:

```
mayor2 |>
  arrange(-contribution_amount_new)
```

```
# A tibble: 10,199 x 14
   contributor~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 Ford, Doug    <NA>    M9A 2C3 508224~ Moneta~ <NA>    Indivi~ Candid~ <NA>
 2 Ford, Rob     <NA>    M9A 3G9 78804.~ Moneta~ <NA>    Indivi~ Candid~ <NA>
 3 Ford, Doug    <NA>    M9A 2C3 50000   Moneta~ <NA>    Indivi~ Candid~ <NA>
 4 Ford, Rob     <NA>    M9A 3G9 50000   Moneta~ <NA>    Indivi~ Candid~ <NA>
 5 Ford, Rob     <NA>    M9A 3G9 50000   Moneta~ <NA>    Indivi~ Candid~ <NA>
 6 Goldkind, Ari <NA>    M5P 1P5 23623.~ Moneta~ <NA>    Indivi~ Candid~ <NA>
 7 Ford, Rob     <NA>    M9A 3G9 20000   Moneta~ <NA>    Indivi~ Candid~ <NA>
 8 Ford, Rob     <NA>    M9A 3G9 12210   Moneta~ <NA>    Indivi~ Candid~ <NA>
 9 Di Paola, Ro~ <NA>    M3H 2T1 6000    Moneta~ <NA>    Indivi~ Candid~ <NA>
10 Thomson, Sar~ <NA>    M4W 2X6 4425.5~ Moneta~ <NA>    Indivi~ Candid~ <NA>
# ... with 10,189 more rows, 5 more variables: authorized_representative <chr>,
#   candidate <chr>, office <chr>, ward <chr>, contribution_amount_new <dbl>,
#   and abbreviated variable names 1: contributors_name,
#   2: contributors_address, 3: contributors_postal_code,
```

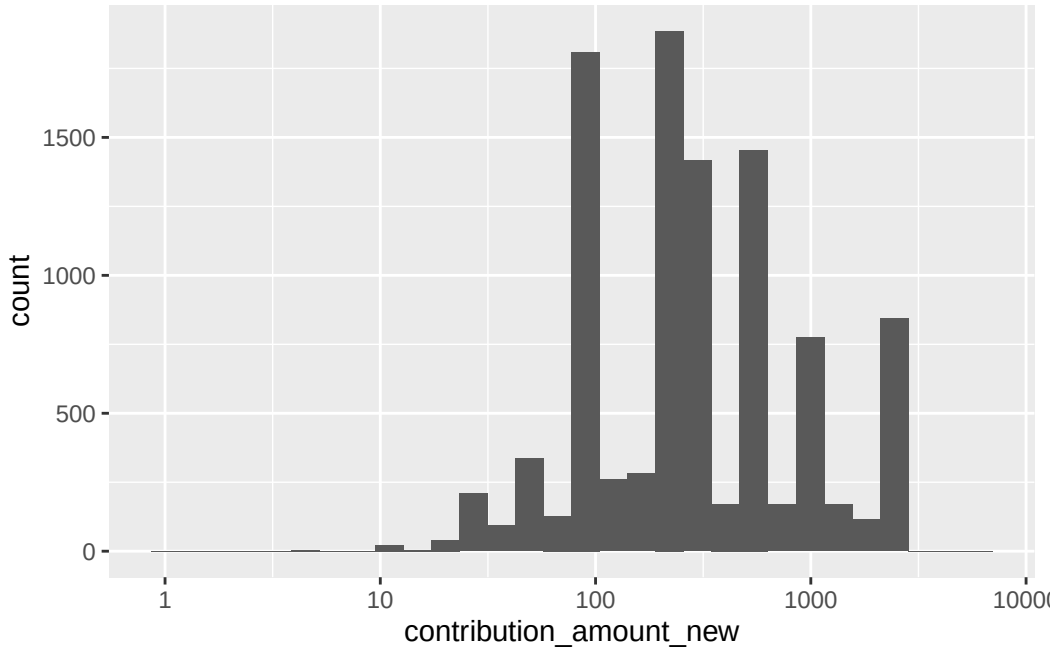
```
#   4: contribution_amount, 5: contribution_type_desc,
#   6: goods_or_service_desc, 7: contributor_type_desc,
#   8: relationship_to_candidate, 9: president_business_manager
```

Interestingly, all of the notable outliers are from candidates donating to their own campaigns! Doug Ford is by far the worst offender as the only candidate to donate amounts over $10,000. We'll drop those and plot it again:

```
mayor3 <- mayor2 |>
  filter(contribution_amount_new<=10000)

ggplot(data = mayor3) +
  geom_histogram(aes(x = contribution_amount_new)) +
  scale_x_log10()
```

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




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

   - total contributions
   - mean contribution
   - number of contributions

Total Contributions:

```
mayor2 |>
  group_by(candidate) |>
  summarize(total_contribution=sum(contribution_amount_new)) |>
  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
```

Mean Contributions:

```
mayor2 |>
  group_by(candidate) |>
  summarize(mean_contribution=mean(contribution_amount_new)) |>
  arrange(-mean_contribution) |>
  slice(1:5)
```

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

Number of Contributions:

```
mayor2 |>
  group_by(candidate) |>
  summarize(num_contribution=n()) |>
  arrange(-num_contribution) |>
  slice(1:5)
```

```
# A tibble: 5 x 2
  candidate       num_contribution
  <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.

Total contributions:

```
mayor2 |>
  filter(contributors_name != candidate) |>
  group_by(candidate) |>
  summarize(total_contribution=sum(contribution_amount_new)) |>
  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.
```

Mean Contributions:

```
mayor2 |>
  filter(contributors_name != candidate) |>
  group_by(candidate) |>
  summarize(mean_contribution=mean(contribution_amount_new)) |>
  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
```

Number of Contributions:

```
mayor2 |>
  filter(contributors_name != candidate) |>
  group_by(candidate) |>
  summarize(num_contribution=n()) |>
  arrange(-num_contribution) |>
  slice(1:5)
```

```
# A tibble: 5 x 2
  candidate       num_contribution
  <chr>                      <int>
1 Chow, Olivia                5706
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?

```
mayor2 |>
  group_by(contributors_name) |>
  summarize(num_candidates=n_distinct(candidate)) |>
  filter(num_candidates>1) |>
  count()
```

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

184 contributors gave money to more than one candidate.