Lab2

Qiaoyu Liang

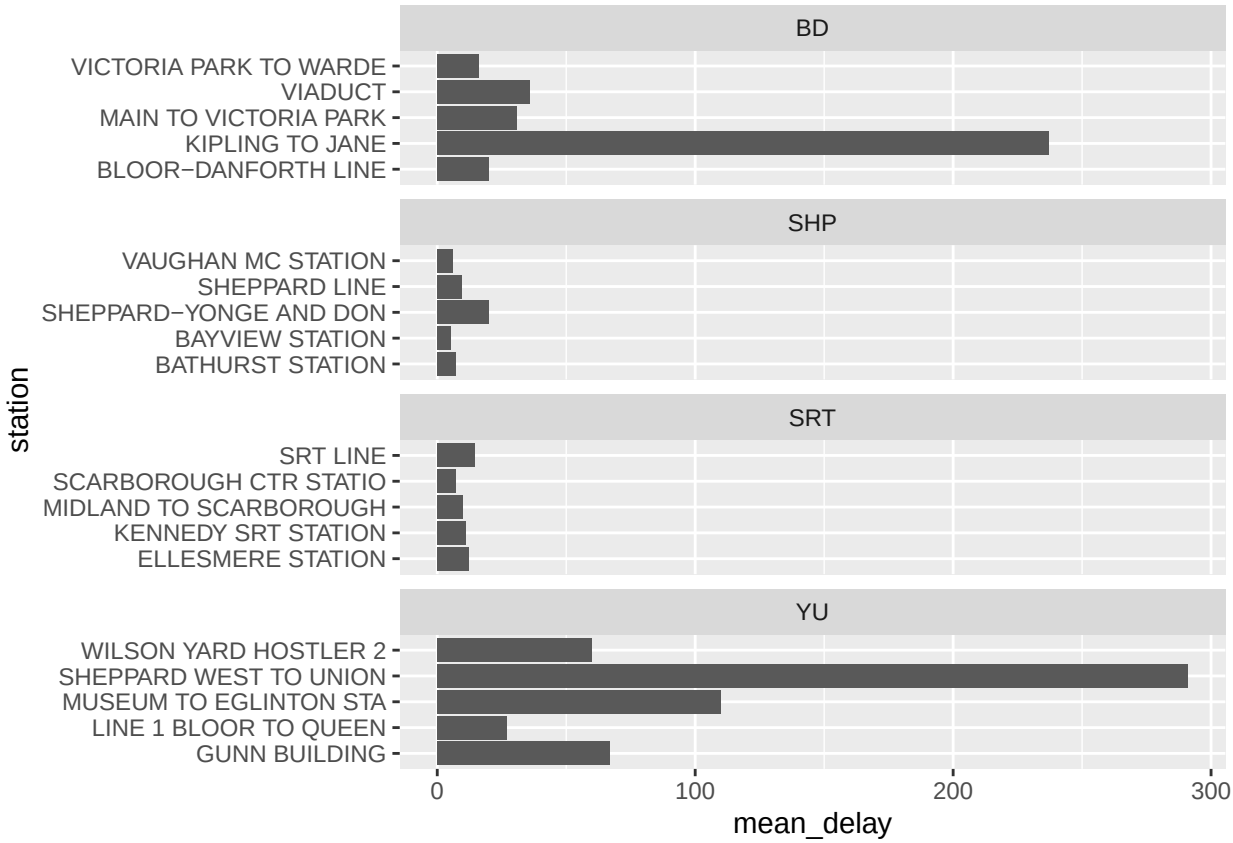
```
library(opendatatoronto)
library(tidyverse)
library(stringr)
# EDA
library(skimr)
# EDA
library(visdat)
library(janitor)
library(lubridate)
library(ggrepel)
```

```
res <- list_package_resources("996cfe8d-fb35-40ce-b569-698d51fc683b")
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_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
```

Lab Exercises

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()
```



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)
list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c")
```

```
## # A tibble: 2 x 4
##   name                                id                   format last_mod~1
##   <chr>                               <chr>                <chr>  <date>
## 1 campaign-contributions-2014-data    5b230e92-0a22-4a15-9~ ZIP    2019-07-23
## 2 campaign-contributions-2014-readme-xls aaf736f4-7468-4bda-9~ XLS    2019-07-23
## # ... with abbreviated variable name 1: last_modified
```

```
camps <- get_resource("5b230e92-0a22-4a15-9572-0b19cc222985")
Mayor2014 <- data.frame(camps[2])
```

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

```
Mayor2014 <- Mayor2014 %>%
  janitor::row_to_names(1) %>%
  janitor::clean_names()
```

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(Mayor2014)
```

Table 1: Data summary

Name	Mayor2014
Number of rows	10199
Number of columns	13
Column type frequency:	
character	13
Group variables	None

Variable type: character

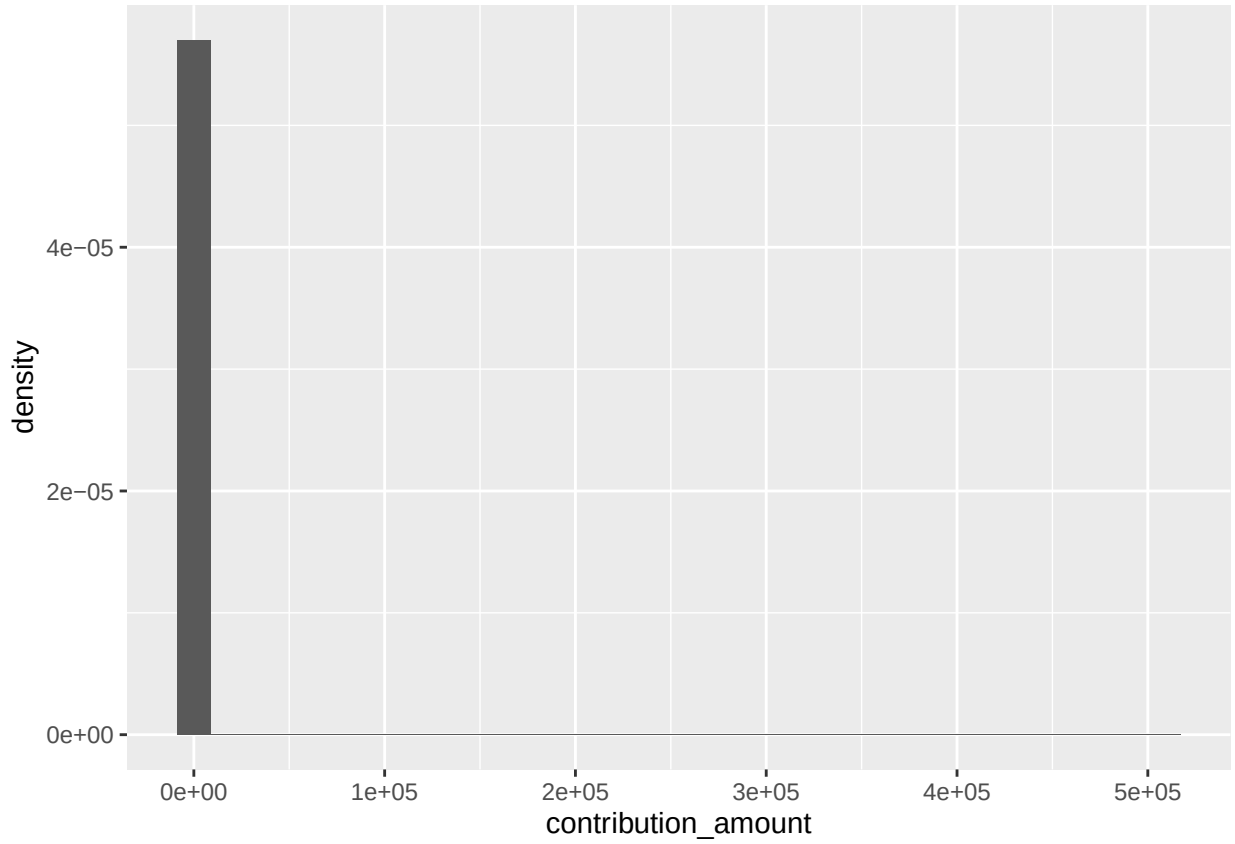
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
contributors_name	0	1	4	31	0	7545	0
contributors_address	10197	0	24	26	0	2	0
contributors_postal_code	0	1	7	7	0	5284	0
contribution_amount	0	1	1	18	0	209	0
contribution_type_desc	0	1	8	14	0	2	0
goods_or_service_desc	10188	0	11	40	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

```
Mayor2014 <- Mayor2014 %>%
  mutate(contribution_amount = as.numeric(contribution_amount))
```

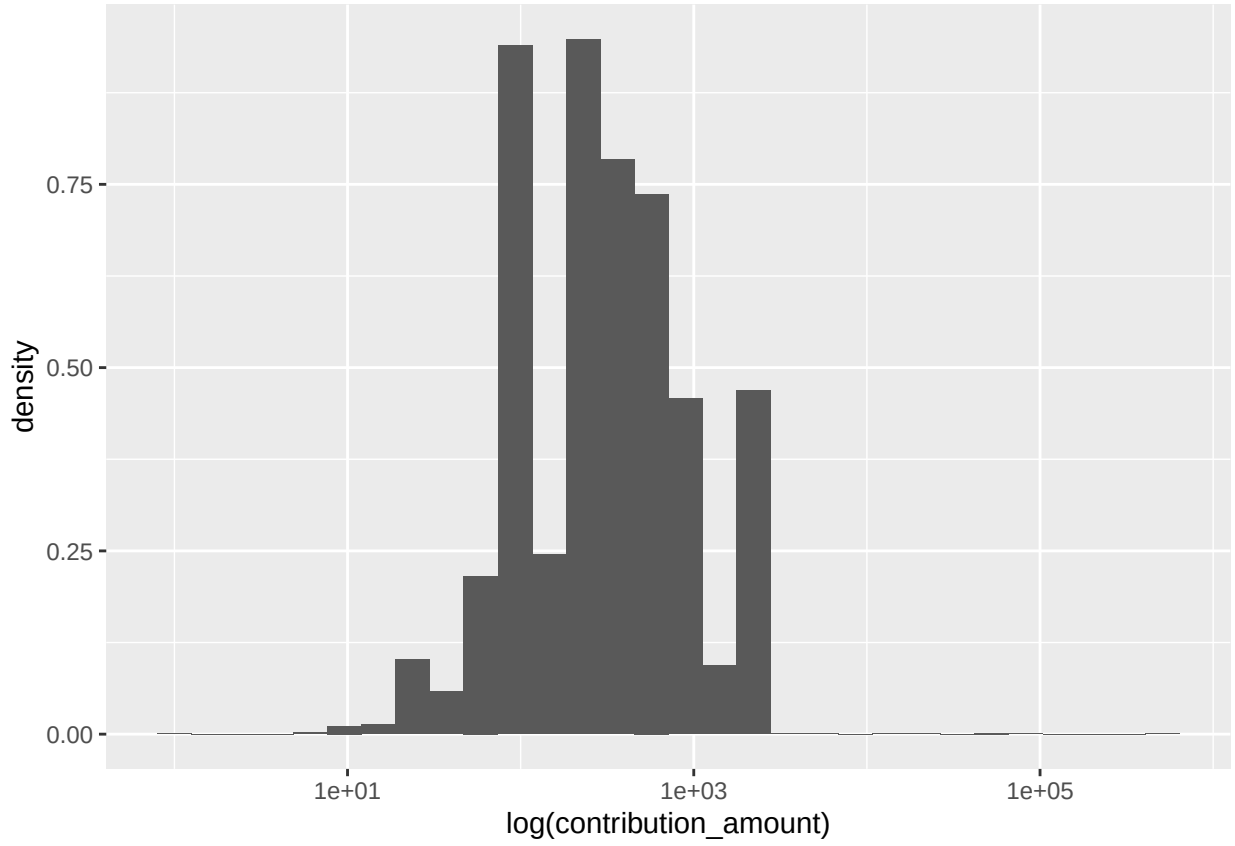
Yes, there are missing values in the dataset. Specifically, there are missing values for variables contributors_address, goods_or_service_desc, relationship_to_candidate, president_business_manager, authorized_representative and ward. Based on the purpose of this study, we can still do the analysis where variables with missing values are excluded. Thus, we should not be worried about the missing values in this case. Notice not every variable is in the format it should be. We notice contribution_amount is originally in character format so we change it in numeric format.

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.

```
Mayor2014 %>%
  ggplot(aes(x = contribution_amount, y = ..density..)) +
  geom_histogram()
```




```
Mayor2014 %>%
  ggplot(aes(x = contribution_amount, y = ..density..)) +
  geom_histogram() +
  scale_x_log10() +
  labs(x= "log(contribution_amount)")
```

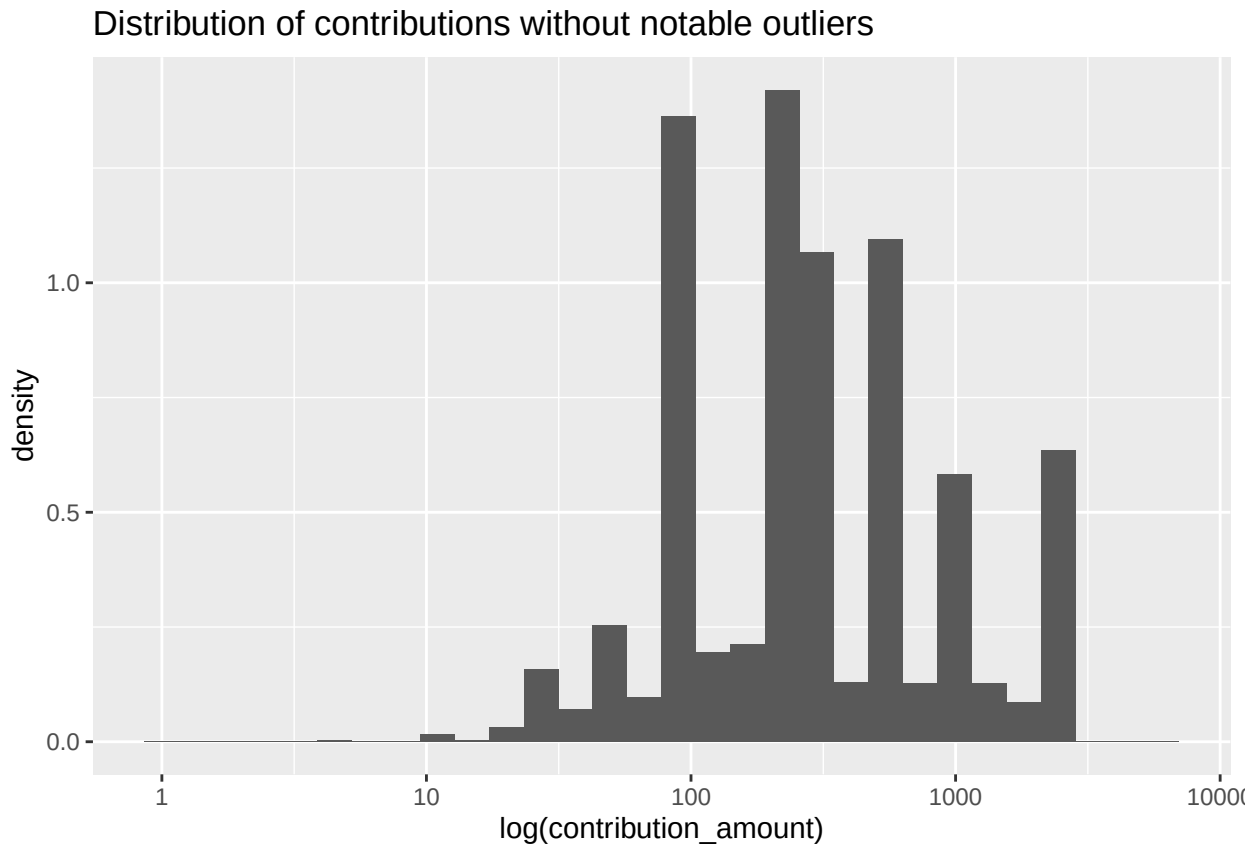



```
Mayor2014 %>% filter(contribution_amount >= 10000)
```

```
##   contributors_name contributors_address contributors_postal_code
## 1        Ford, Doug                <NA>                  M9A 2C3
## 2        Ford, Doug                <NA>                  M9A 2C3
## 3         Ford, Rob                <NA>                  M9A 3G9
## 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     Goldkind, Ari                <NA>                  M5P 1P5
##   contribution_amount contribution_type_desc goods_or_service_desc
## 1           508224.73               Monetary                  <NA>
## 2            50000.00               Monetary                  <NA>
## 3            20000.00               Monetary                  <NA>
## 4            50000.00               Monetary                  <NA>
## 5            50000.00               Monetary                  <NA>
## 6            78804.80               Monetary                  <NA>
## 7            12210.00               Monetary                  <NA>
## 8            23623.63               Monetary                  <NA>
##   contributor_type_desc relationship_to_candidate president_business_manager
## 1            Individual                Candidate                       <NA>
## 2            Individual                Candidate                       <NA>
## 3            Individual                Candidate                       <NA>
## 4            Individual                Candidate                       <NA>
```

```
## 5            Individual                Candidate                      <NA>
## 6            Individual                Candidate                      <NA>
## 7            Individual                Candidate                      <NA>
## 8            Individual                Candidate                      <NA>
##   authorized_representative     candidate office ward
## 1                      <NA>    Ford, Doug  Mayor <NA>
## 2                      <NA>    Ford, Doug  Mayor <NA>
## 3                      <NA>     Ford, Rob  Mayor <NA>
## 4                      <NA>     Ford, Rob  Mayor <NA>
## 5                      <NA>     Ford, Rob  Mayor <NA>
## 6                      <NA>     Ford, Rob  Mayor <NA>
## 7                      <NA>     Ford, Rob  Mayor <NA>
## 8                      <NA> Goldkind, Ari  Mayor <NA>
```

```
Mayor2014 %>%
  filter(contribution_amount <= 10000) %>%
  ggplot(aes(x = contribution_amount, y = ..density..)) +
  geom_histogram() +scale_x_log10() +
  labs(title = "Distribution of contributions without notable outliers",
       x= "log(contribution_amount)")
```



Contributions that exceed 10000 can be considered as potential notable outliers. The similar characteristics are that those contributions are contributed by candidates themselves and most of them come from the Ford family.

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

- total contributions
- mean contribution
- number of contributions

```
# top five candidates in total contributions
Mayor2014 %>%
  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 in mean contributions
Mayor2014 %>%
  group_by(candidate) %>%
  summarise(mean_contribution = mean(contribution_amount)) %>%
  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
```

```
# top five candidates in number of contributions
Mayor2014 %>%
  group_by(candidate) %>%
  tally() %>%
  arrange(-n) %>%
  slice(1:5)
```

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

7. Repeat 6 but without contributions from the candidates themselves.

```
Mayor2014_7 <- Mayor2014 %>%
  filter(contributors_name!=candidate)

# top five candidates in total contributions
Mayor2014_7 %>%
  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 in mean contributions
Mayor2014_7 %>%
  group_by(candidate) %>%
  summarise(mean_contribution = mean(contribution_amount)) %>%
  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 in number of contributions
Mayor2014_7 %>%
  group_by(candidate) %>%
  tally() %>%
  arrange(-n) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   candidate           n
##   <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?

```
Mayor2014 %>%
  group_by(contributors_name) %>%
  distinct(candidate) %>%
  tally() %>%
  filter(n > 1) %>%
  nrow()
```

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

184 contributors gave money to more than one candidate.