Lab 2

Fred Haochen Song

Lab Exercise:

```
options(warn=-1)
library(opendatatoronto)
library(tidyverse)
library(stringr)
library(skimr) # EDA
library(visdat) # EDA
library(janitor)
library(lubridate)
library(ggrepel)
```

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‘

Let’s reload the data first:

```
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)

## Removing the observations that have non-standardized lines

delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
```

```
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")

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"
```

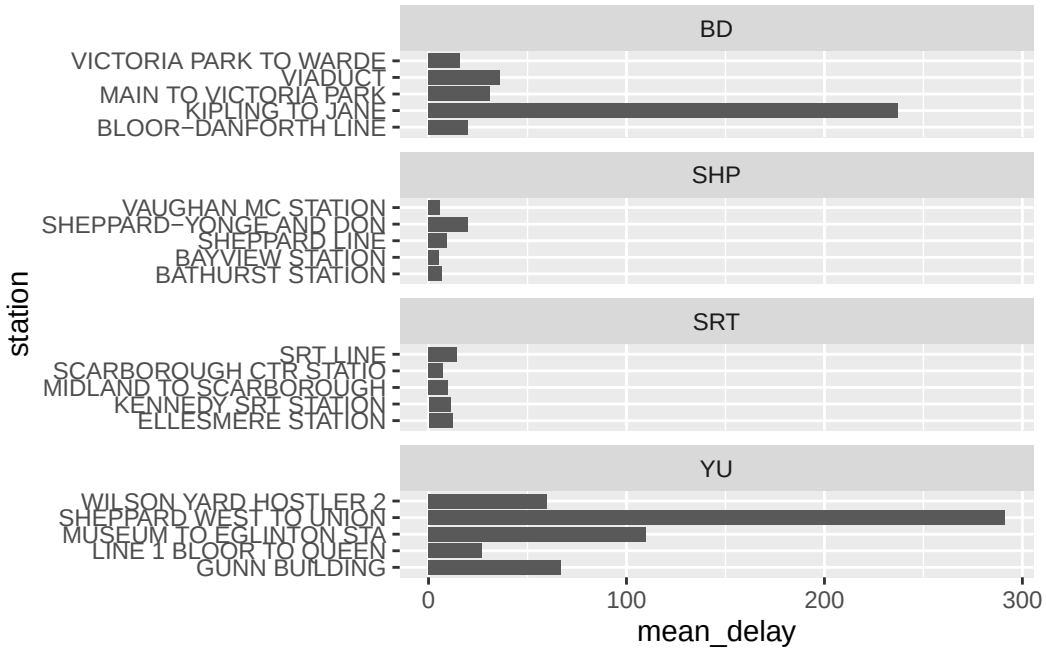
```
delay_2022 <- delay_2022 |>
  mutate(station_clean = ifelse(str_starts(station, "ST"), word(station, 1,2), word(statio

delay_2022 <- delay_2022 |>
  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))
         )
```

And do the plot below:

```
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: The ID I found was: f6651a40-2f52-46fc-9e04-b760c16edd5c

+ 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

```
cap <- list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c")
cap_2014 <- get_resource('5b230e92-0a22-4a15-9572-0b19cc222985')
```

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

```
dat <- cap_2014$'2_Mayor_Contributions_2014_election.xls'
head(dat)
```

```
# A tibble: 6 x 13
  `2014 Municipal ~` ...2  ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11
  <chr>              <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 Contributor's Name Cont~ Cont~ Cont~ Cont~ Good~ Cont~ Rela~ Pres~ Auth~ Cand~
2 A D'Angelo, Tullio <NA>  M6A ~ 300   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Ford~
3 A Strazar, Martin  <NA>  M2M ~ 300   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Ford~
4 A'Court, K Susan   <NA>  M4M ~ 36    Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~
5 A'Court, K Susan   <NA>  M4M ~ 100   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~
6 A'Court, K Susan   <NA>  M4M ~ 100   Mone~ <NA>  Indi~ <NA>  <NA>  <NA>  Chow~
# ... with 2 more variables: ...12 <chr>, ...13 <chr>
```

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

Hi Professor! I'm not so sure if I understand this question...so I tried to fix the basic issue of the data where the first column is not displayed correctly, and the column names were not assigned...so sorry if I misinterpret it...

```r
dat <- dat |>
  row_to_names(1)
head(dat)
```

```
# A tibble: 6 x 13
  `Contributor's Name` `Contributor's Address` `Contributor's~` `Contribution ~`
  <chr>                <chr>                  <chr>            <chr>
1 A D'Angelo, Tullio   <NA>                   M6A 1P5          300
2 A Strazar, Martin    <NA>                   M2M 3B8          300
3 A'Court, K Susan     <NA>                   M4M 2J8          36
4 A'Court, K Susan     <NA>                   M4M 2J8          100
5 A'Court, K Susan     <NA>                   M4M 2J8          100
6 Aaron, Robert B      <NA>                   M6B 1H7          250
# ... with 9 more variables: `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>
```

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.

a. is there missing values?

```r
dat|>
  summarize(across(everything(), ~ sum(is.na(.x))))
```

```
# A tibble: 1 x 13
  `Contributor's Name` `Contributor's Address` `Contributor's~` `Contribution ~`
                 <int>                  <int>            <int>            <int>
1                    0                  10197                0                0
# ... with 9 more variables: `Contribution Type Desc` <int>,
#   `Goods or Service Desc` <int>, `Contributor Type Desc` <int>,
#   `Relationship to Candidate` <int>, `President/ Business Manager` <int>,
#   `Authorized Representative` <int>, Candidate <int>, Office <int>,
#   Ward <int>
```

Answer: There are missing values but we don't need to worry about them, because those columns with Missing values are of really large scale, therefore we can just not use those columns at all.

b. is every variable in the format it should be?

Answer: No, the contribution amount should be of type dbl or int, let's change it to dbl.

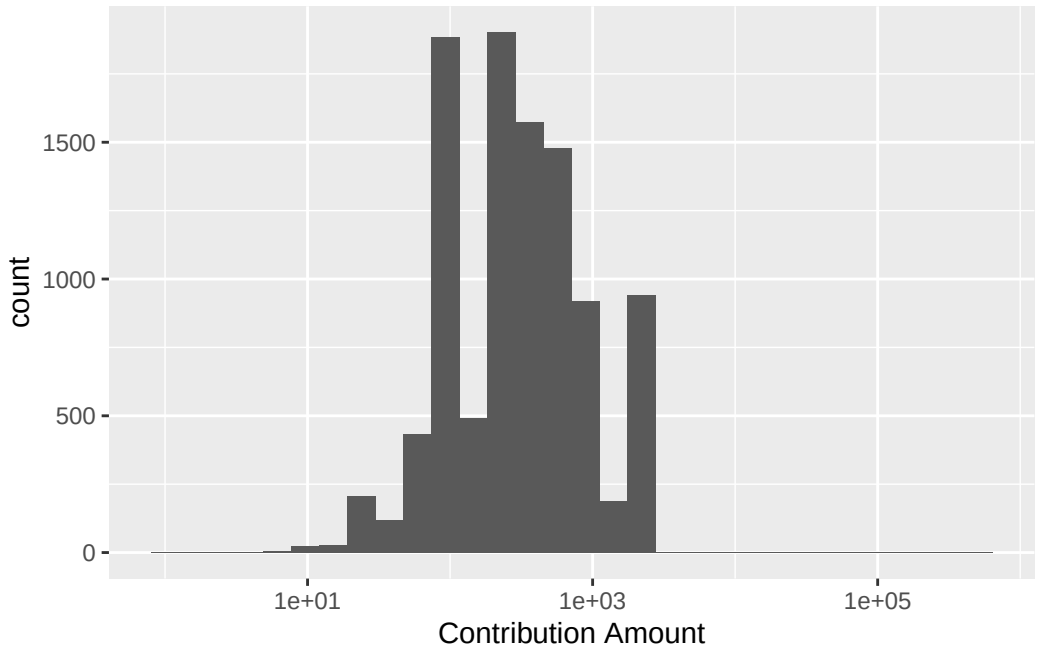
```
dat$`Contribution Amount` <- as.numeric(dat$`Contribution Amount`)
head(dat)
```

```
# A tibble: 6 x 13
  `Contributor's Name` `Contributor's Address` `Contributor's~` `Contribution ~`
  <chr>                <chr>                   <chr>                       <dbl>
1 A D'Angelo, Tullio   <NA>                    M6A 1P5                       300
2 A Strazar, Martin    <NA>                    M2M 3B8                       300
3 A'Court, K Susan     <NA>                    M4M 2J8                        36
4 A'Court, K Susan     <NA>                    M4M 2J8                       100
5 A'Court, K Susan     <NA>                    M4M 2J8                       100
6 Aaron, Robert B      <NA>                    M6B 1H7                       250
# ... with 9 more variables: `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>
```

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 = dat) +
  geom_histogram(aes(x = `Contribution Amount`))+
  scale_x_log10()
```

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



it can be seen that contribution amount > 10000 are clearly outliers, let's take a look at them:

```
dat |>
  filter(`Contribution Amount` > 10000) |>
  select(`Contributor's Name`,`Contribution Amount`, Candidate)
```

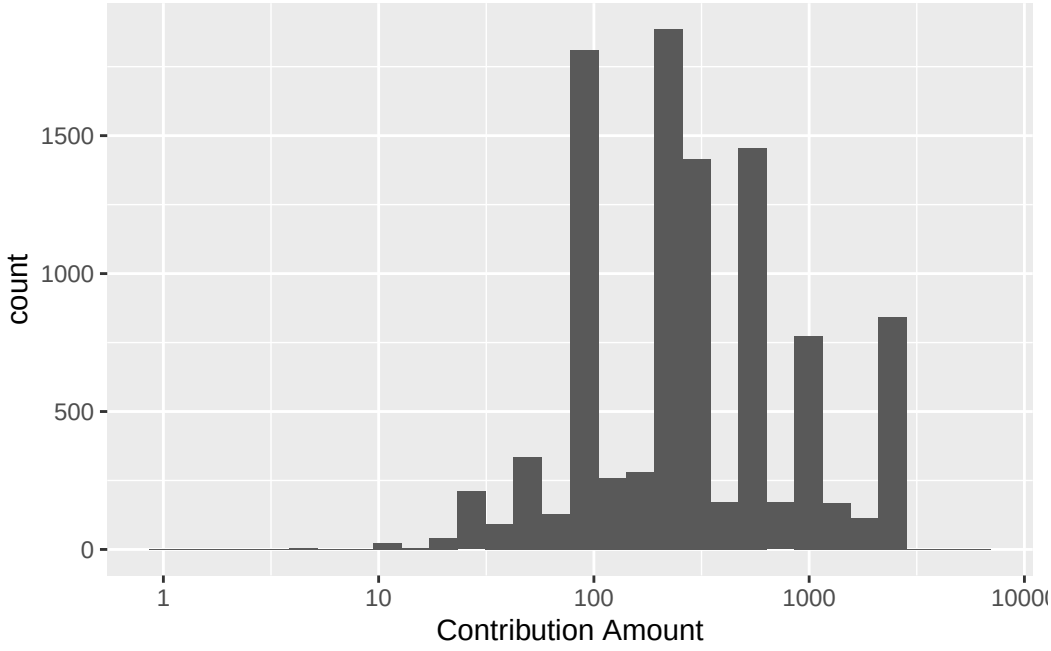
```
# A tibble: 8 x 3
  `Contributor's Name` `Contribution Amount` Candidate
  <chr>                                <dbl> <chr>
1 Ford, Doug                         508225. Ford, Doug
2 Ford, Doug                          50000  Ford, Doug
3 Ford, Rob                           20000  Ford, Rob
4 Ford, Rob                           50000  Ford, Rob
5 Ford, Rob                           50000  Ford, Rob
6 Ford, Rob                           78805. Ford, Rob
7 Ford, Rob                           12210  Ford, Rob
8 Goldkind, Ari                       23624. Goldkind, Ari
```

It can be seen that they all come from the candidate themselves, i.e. contributor's name = candidate.

let's remove the outliers:

```r
dat |>
  filter(`Contribution Amount` <= 10000) |>
  ggplot(aes(x = `Contribution Amount`))+
  geom_histogram()+
  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

a. Total contributions:

```r
dat |>
  group_by(Candidate) |>
  summarise(tot_contribution = sum(`Contribution Amount`)) |>
  arrange(-tot_contribution) |>
  slice(1:5)
```

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

b. Mean contributions:

```
dat |>
  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
```

c. Number of Contributions:

```
dat |>
  group_by(Candidate) |>
  summarise(cnt_contribution = length(`Contribution Amount`)) |>
  arrange(-cnt_contribution) |>
  slice(1:5)
```

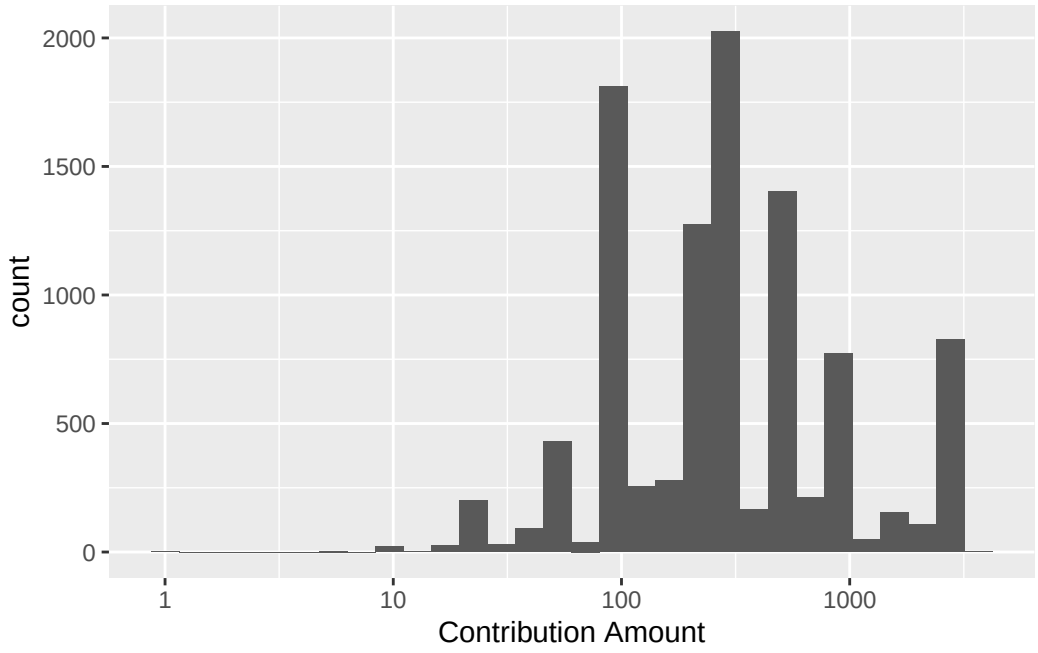
```
# A tibble: 5 x 2
  Candidate       cnt_contribution
  <chr>                      <int>
1 Chow, Olivia                5708
2 Tory, John                  2602
3 Ford, Doug                   611
4 Ford, Rob                    538
5 Soknacki, David              314
```

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

let's plot them here:

```
dat |>
  filter(`Contributor's Name`!= Candidate) |>
  ggplot(aes(x = `Contribution Amount`))+
  geom_histogram()+
  scale_x_log10()
```

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



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

```
dat |>
  group_by(`Contributor's Name`, Candidate) |>
  summarise(unique_candidate = unique(Candidate)) |>
  filter (length(unique_candidate) >1) |>
  summarise(num_candidate = length(unique_candidate)) |>
  select(`Contributor's Name`, num_candidate)
```

```
`summarise()` has grouped output by 'Contributor's Name'. You can override
using the `.groups` argument.
```

```
# A tibble: 184 x 2
   `Contributor's Name` num_candidate
   <chr>                        <int>
 1 Abadi, Babak                     2
 2 Adams, Michael                   2
 3 Anga, John                       2
 4 Argyris, Katerina                2
 5 Atkinson, Tom                    2
 6 Aziz, Peter                      2
 7 Bachir, Salah                    2
 8 Bajwa, Joginder                  2
 9 Baker, Norma                     2
10 Banwait, Rav                     2
# ... with 174 more rows
```

There are in total 184 contributors gave money more than one candidate.