Lab 2 Toronto TTC delay and Mayor Contribution Analyses

Hainan Xu

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

We import the data that is preprocessed in the lab2:

```r
delay_2022 <- read_csv("labs/ttcdelay_2022.csv")
```

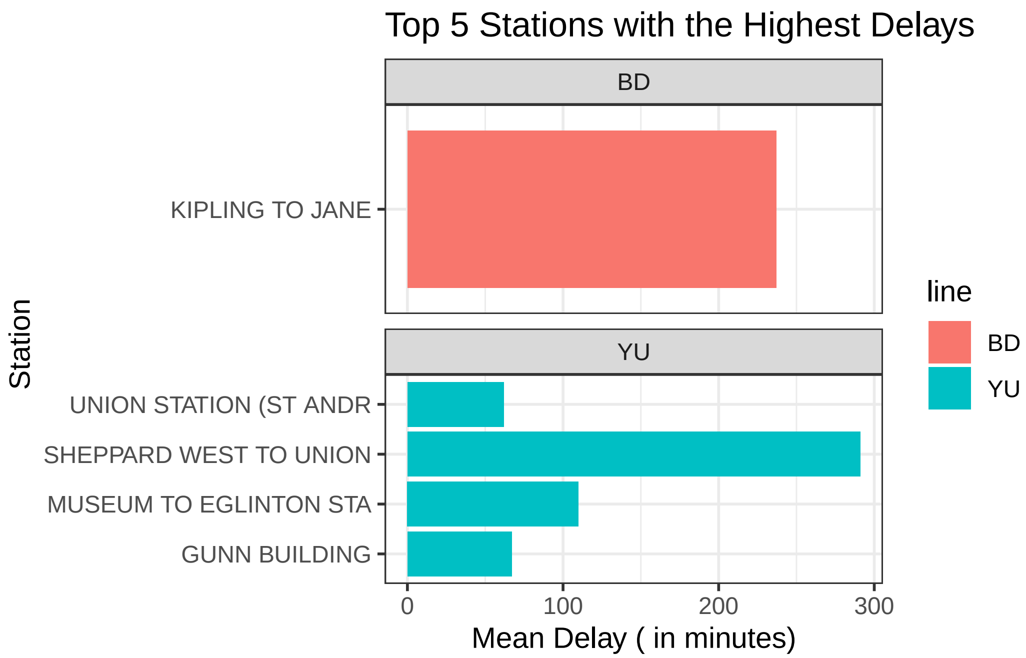
1. Using the `delay_2022` data, plot the five stations with the highest mean delays. Facet the graph by `line` .

The station variable does not always suggest only one station, it might also suggest the route between one station to another, that could be one of the reasons why some of the delays are really long.

```r
delay_2022 |>
  group_by(station,line) |>
  summarize(mean_delay=mean(min_delay))|>#,.groups = 'drop'
  arrange(-mean_delay)|>
  head(5)|>
  ggplot(aes(station, mean_delay, fill = line)) +
  geom_col() +
  facet_wrap(vars(line),  # Facet by the 'line' variable
             scales = "free_y",
             nrow = 4) +
```

```
  coord_flip()+
  labs(title = "Top 5 Stations with the Highest Delays",x="Station",y="Mean Delay ( in min
  theme_bw()
```

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



2. 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.

Below are the top 50% of the most frequent delay reasons.

```
#top 50% of most frequent delay reasons

delay_2022|>
  group_by(code_red)|>
  summarise(frequency=n()) |>
  arrange(-frequency) |>
```

```
  mutate(rank=cumsum(frequency)/sum(frequency)) |>
  slice(1:5)
```

```
# A tibble: 5 x 3
  code_red      frequency  rank
  <chr>             <int> <dbl>
1 Injured            3689 0.190
2 Passenger          2461 0.316
3 Disorderly         1936 0.416
4 Miscellaneous      1451 0.490
5 OPTO               1143 0.549
```

Since the min_delay variable is continuous, we fit a linear regression:

```
top_delay_reasons<- delay_2022|>
  group_by(code_red)|>
  summarise(frequency=n(),na.rm=TRUE) |>
  arrange(-frequency) |>
  mutate(rank=cumsum(frequency)/sum(frequency)) |>
  filter(rank<= 0.5) |>
  select(code_red)

#filter delay_2022
q2<-delay_2022|>filter(min_delay>0, code_red %in% top_delay_reasons$code_red)
#fit the model
model<-lm(min_delay~line + code_red, data=q2)
summary(model)
```

```

Call:
lm(formula = min_delay ~ line + code_red, data = q2)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.437  -3.603  -2.350   0.650 149.194 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)             6.7708     0.2909  23.277  < 2e-16 ***
lineSHP                 0.6321     0.8358   0.756    0.450    
lineSRT                 6.7821     0.8445   8.031 1.42e-15 ***
```

```
lineYU                  -0.4211     0.2915  -1.445    0.149
code_redInjured          1.8841     0.3694   5.100 3.63e-07 ***
code_redMiscellaneous   -0.4969     0.5028  -0.988    0.323
code_redPassenger        0.2530     0.3368   0.751    0.453
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.226 on 2751 degrees of freedom
Multiple R-squared:  0.03589,   Adjusted R-squared:  0.03379
F-statistic: 17.07 on 6 and 2751 DF,  p-value: < 2.2e-16
```

From the output above, with baseline lineBD and code being disorderly, the average estimated late time is around 6.7 minutes. If the line is SRT, the average estimated delay time will increase by 6.78 minutes. This does noe match what we observed in previos EDA, where we found the top 5 delayed stations are either lineYU or lineBD. The model is poorly fitted. More covariates need to be incorporated into this model, and we may need to consider performing some transformations or other analyses methods.

3. **Using the `opendatatoronto` package, download the data on mayoral campaign contributions for 2014 and clean it up**.

```r
all_data <- search_packages("campaign")
campaign_data_ids <- all_data$id
resources <- list_package_resources(campaign_data_ids[1])
mayor_campaign_data <- get_resource('8b42906f-c894-4e93-a98e-acac200f34a4')
mayor_contributions <- mayor_campaign_data$`2_Mayor_Contributions_2014_election.xls`
colnames(mayor_contributions) <- as.character(mayor_contributions[1, ])
mayor_contributions <- mayor_contributions[-1, ]
rownames(mayor_contributions) <- NULL
clean_mayor_contributions <- mayor_contributions |>
  clean_names()
head(clean_mayor_contributions )
```

```
# 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**.

The data are being summarized as the output as follows. There are missing values in the data in `contributors_address`, `goods_or_service_desc`,`relationship_to_candidate`,`president_business_manag` `authorized_representative` and `ward`. however, we may not need to worry about them since over 99% of the data of those vatiables are missing, and we will not analyse those variables. Based on the output, all the variables are characters. We modified the `contribution_amount` to be numeric.

```
skim(clean_mayor_contributions)
```

Table 1: Data summary

Name	clean_mayor_contributions
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

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
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

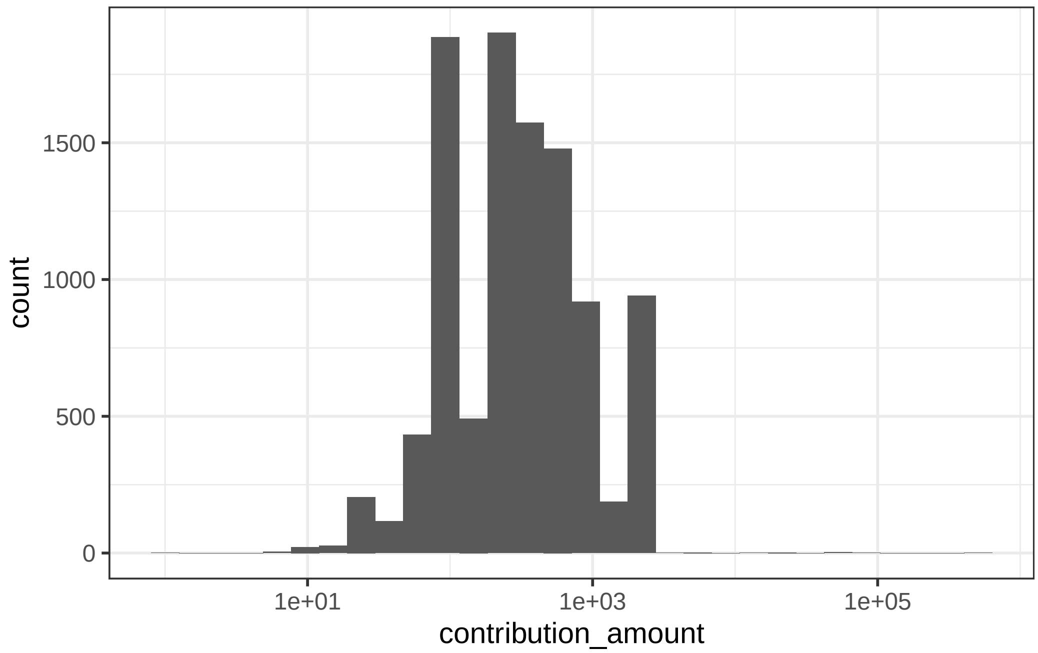
```
data<- clean_mayor_contributions|> mutate(contribution_amount=as.numeric(contribution_amou
```

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.**

We plot the histogram of contribution amount with log10 transformed x-axis. We observe that there are some large contributions.

```
#histogram
ggplot(data=data)+
 geom_histogram(aes(x=contribution_amount))+
 scale_x_log10()+theme_bw()
```

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



We plot the boxplot and get the outliers. The contributions over 1100 are considered as outliers for the boxplot. The threshold is calculated by 75% quantile of the data * 1.5 IQR.

```
quantile(data$contribution_amount,0.75)+1.5*IQR(data$contribution_amount)
```

```
 75%
1100
```

Some notable outliers are :2210.00 ,20000.00, 23623.63, 50000.00, 78804.80 and 508224.73. The contributors of the contributions that are over $4000 are contributed by the candidates themselves.

```
data|> filter(contribution_amount>=1100) |> arrange(-contribution_amount) |> select(contri
```

```
# A tibble: 1,147 x 3
   contributors_name candidate        contribution_amount
   <chr>             <chr>                          <dbl>
 1 Ford, Doug        Ford, Doug                   508225.
 2 Ford, Rob         Ford, Rob                     78805.
 3 Ford, Doug        Ford, Doug                    50000
 4 Ford, Rob         Ford, Rob                     50000
```

```
 5 Ford, Rob         Ford, Rob                     50000
 6 Goldkind, Ari     Goldkind, Ari                 23624.
 7 Ford, Rob         Ford, Rob                     20000
 8 Ford, Rob         Ford, Rob                     12210
 9 Di Paola, Rocco   Di Paola, Rocco                6000
10 Thomson, Sarah    Thomson, Sarah                 4426.
# ... with 1,137 more rows
```

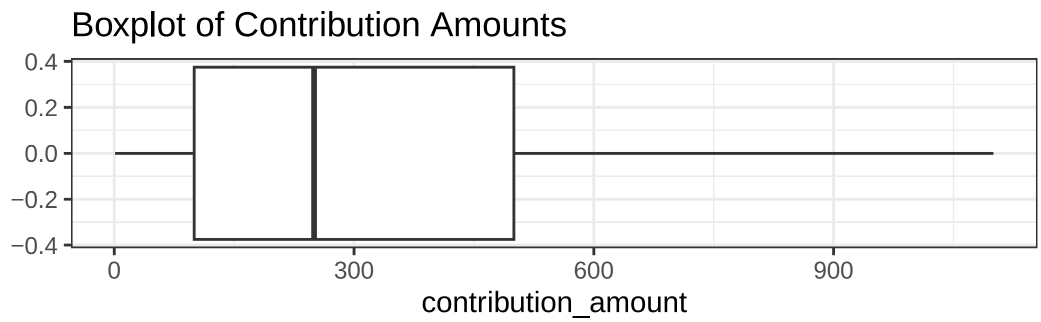
```
#data|> filter(contribution_amount>=1100) |> arrange(-contribution_amount) |> select(contr
```

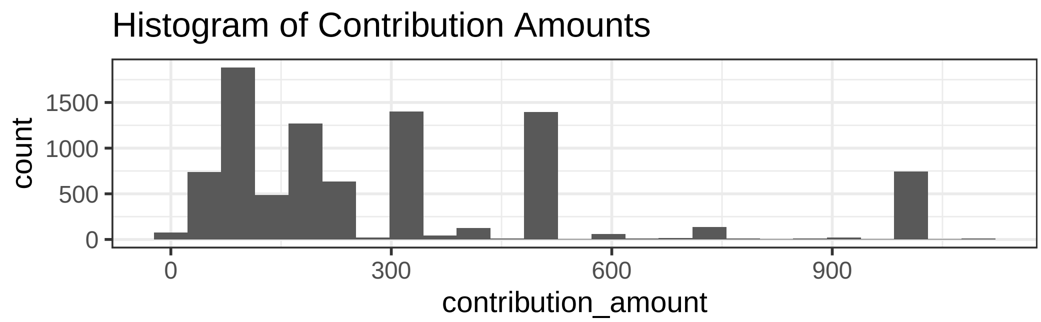
We only plot the values that are smaller or equal to 1100 to get a better sense of the data.

```
filtered_data <-data |> filter(contribution_amount <= 1100)
library(gridExtra)
```

```
Attaching package: 'gridExtra'


The following object is masked from 'package:dplyr':

    combine
```

```
a<-ggplot(data=filtered_data)+
   geom_boxplot(aes(x=contribution_amount))+
   theme_bw()+labs(title="Boxplot of Contribution Amounts")
b<-ggplot(filtered_data) + geom_histogram(aes(x=contribution_amount),bins = 25) +theme_bw(

grid.arrange(a, b, ncol = 1)
```





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

```
#top5 total contributions
total_contributions<- data |> group_by(candidate) |> summarise(total_contributions=sum(con
total_contributions
```

```
# 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
```

```
#top5 mean contributions
mean_contributions<- data|> group_by(candidate)|>
  summarise(mean_contributions=mean(contribution_amount)) |> arrange(-mean_contributions)
mean_contributions
```

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

```
#top5 number of countributions
number_of_contributions<-data|> group_by(candidate)|>
  summarise(frequency=n()) |>
  arrange(-frequency)|>
  slice(1:5)
number_of_contributions
```

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

```
#remove the contributions from the candidates themselves
data2<- data |> filter(contributors_name!=candidate)

#top5 total contributions
total_contributions<- data2 |> group_by(candidate) |> summarise(total_contributions=sum(co
total_contributions
```

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

```r
#top5 mean contributions
mean_contributions<- data2|> group_by(candidate)|>
  summarise(mean_contributions=mean(contribution_amount)) |> arrange(-mean_contributions)
mean_contributions
```

```
# 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
```

```r
#top5 number of countributions
number_of_contributions<-data2|> group_by(candidate)|>
  summarise(frequency=n()) |>
  arrange(-frequency)|>
  slice(1:5)
number_of_contributions
```

```
# A tibble: 5 x 2
  candidate       frequency
  <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?**

There are 184 contributors gave money to more than one candidate.

```r
data|> group_by(contributors_name) |> summarise(num_contribution=n_distinct(candidate)) |>
```

```
[1] 184
```