Lab Exercises 2

```r
library(opendatatoronto)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.1     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(stringr)
library(skimr) # EDA
library(visdat) # EDA
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(lubridate)
library(ggrepel)
```

```r
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)
```

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

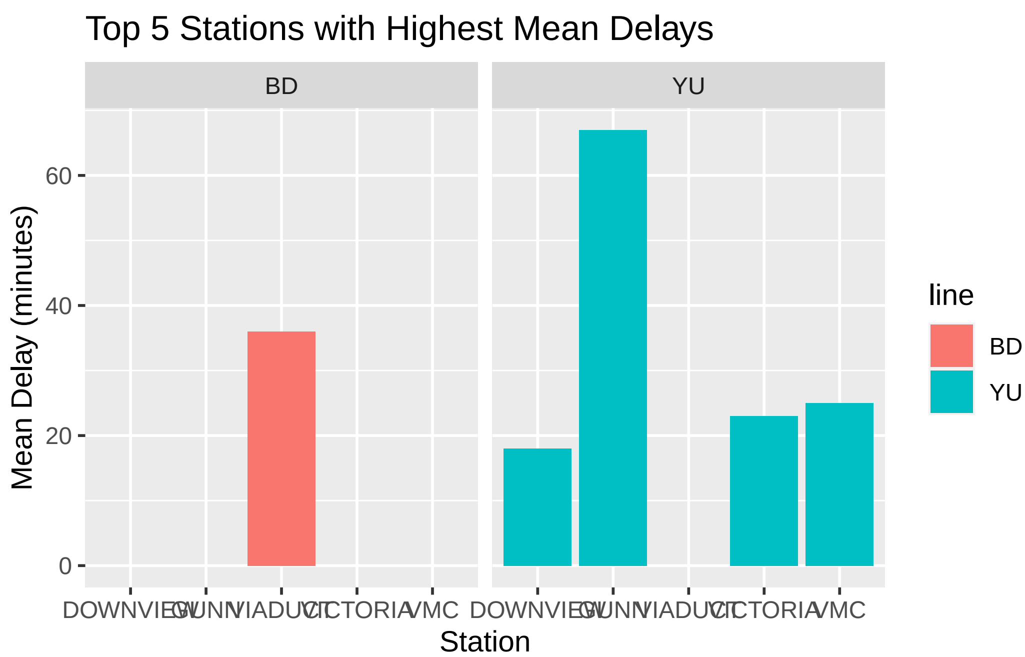
Question 1

```r
# Calculate the mean delay for each station and line
mean_delays <- delay_2022 |>
  group_by(station_clean, line) |>
  summarise(mean_delay = mean(min_delay, na.rm = TRUE)) |>
  ungroup()
```

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

```r
# Find the top five stations with the highest mean delays
top_stations <- mean_delays |>
  arrange(desc(mean_delay)) |>
  slice_max(order_by = mean_delay, n = 5)

# Plot the data, faceting by line
ggplot(top_stations, aes(x = station_clean, y = mean_delay, fill = line)) +
  geom_col() +
  facet_wrap(~ line) +
  labs(title = "Top 5 Stations with Highest Mean Delays",
       x = "Station",
       y = "Mean Delay (minutes)")
```



Question 2

```r
top_50 <- delay_2022 |>
  filter(min_delay > 0) |>
  group_by(code) |>
  summarise(count = length(code)) |>
  arrange(-count) |>
  mutate(cumulative_sum = cumsum(count))|>
  filter(cumulative_sum <= tail(cumulative_sum,1)/2) |>
  select(code)

top_50
```

```
# A tibble: 8 x 1
  code
  <chr>
1 SUDP
2 PUOPO
3 MUATC
4 MUPAA
5 SUUT
```

```
6 TUNOA
7 SUO
8 MUIR
```

```r
filtered_data <- delay_2022 |>
  filter(min_delay > 0 & (code %in% top_50$code))

filtered_data
```

```
# A tibble: 4,407 x 11
   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 08:12 Saturd~ FINCH ~ TUNOA         6      12 S     YU
 2 2022-01-01 00:00:00 09:51 Saturd~ FINCH ~ TUNOA         6      12 S     YU
 3 2022-01-01 00:00:00 12:01 Saturd~ DAVISV~ SUDP          3       8 S     YU
 4 2022-01-01 00:00:00 12:14 Saturd~ RUNNYM~ SUUT         20      25 W     BD
 5 2022-01-01 00:00:00 18:20 Saturd~ EGLINT~ MUATC         3      10 S     YU
 6 2022-01-01 00:00:00 18:59 Saturd~ EGLINT~ MUATC         3      10 S     YU
 7 2022-01-01 00:00:00 19:13 Saturd~ HIGHWA~ PUOPO         5      12 S     YU
 8 2022-01-01 00:00:00 23:37 Saturd~ KENNED~ SUDP          7      14 W     BD
 9 2022-01-02 00:00:00 08:14 Sunday  SHEPPA~ PUOPO         6      12 N     YU
10 2022-01-02 00:00:00 08:59 Sunday  EGLINT~ TUNOA         6      12 N     YU
# i 4,397 more rows
# i 2 more variables: vehicle <dbl>, station_clean <chr>
```

```r
model <- lm(min_delay~as.factor(line) + as.factor(code), data=filtered_data)
```

```r
summary(model)
```

```
Call:
lm(formula = min_delay ~ as.factor(line) + as.factor(code), data = filtered_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.475  -2.450  -1.072   0.890 227.525 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)             5.7698     0.3485  16.554  < 2e-16 ***
as.factor(line)SHP      1.3899     0.5828   2.385 0.017132 *
as.factor(line)YU      -0.3203     0.2521  -1.270 0.204022
as.factor(code)MUIR     1.5470     0.4432   3.491 0.000486 ***
as.factor(code)MUPAA   -1.6602     0.3741  -4.438  9.3e-06 ***
as.factor(code)PUOPO   -0.9396     0.3405  -2.759 0.005814 **
as.factor(code)SUDP     0.9928     0.3344   2.969 0.003003 **
as.factor(code)SUO      5.1117     0.4381  11.667  < 2e-16 ***
as.factor(code)SUUT     7.7057     0.4069  18.938  < 2e-16 ***
as.factor(code)TUNOA   -1.3775     0.3954  -3.484 0.000499 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.38 on 4396 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.1668,    Adjusted R-squared:  0.1651
F-statistic:  97.8 on 9 and 4396 DF,  p-value: < 2.2e-16
```

Based on the fitting result, most of the coefficients are statistically significant, but only lineYU is not. The r-squared and adjusted r-squared both are around 16.5% which is very low.

The result from the Question 1 shows that every station other than GUNN in YU line has a smaller mean delay time than the ones in BD. This consequence is aligned with the negative coefficient of lineYU.

Question 3

```r
# Step 1: Find the ID code for the package related to 'campaign'
package_results <- search_packages("campaign")
campaign_package_id <- package_results$id[1]  # Assuming the first result is the correct o

# Step 2: Get the ID for the specific data file
resources <- list_package_resources(campaign_package_id)
mayoral_campaign_resource_id <- resources$id[3]

# Step 3: Download the data file
mayoral_campaign_data <- get_resource(mayoral_campaign_resource_id)[[2]]
```

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

```r
colnames(mayoral_campaign_data) <- as.character(mayoral_campaign_data[1,])
mayoral_campaign_data <- mayoral_campaign_data[-1,]

rownames(mayoral_campaign_data) <- NULL
mayoral_campaign_data <- clean_names(mayoral_campaign_data)

mayoral_campaign_data
```

```
# A tibble: 10,199 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
 7 Abadi, Babak       <NA>                 M5S 2W7
 8 Abadi, Babak       <NA>                 M5S 2W7
 9 Abadi, David       <NA>                 M5S 2W7
10 Abate, Frank       <NA>                 L4H 2K7
# i 10,189 more rows
# 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>
```

Question 4

There are some variable containing a bunch of missing values which can make the model distorted. After dropping the variables with the missing values, the resulting data set involves

7 columns as a result.

```
noMissing <- function(x) all(!is.na(x))

mayoral_campaign_data <- mayoral_campaign_data |>
  select(where(noMissing))

mayoral_campaign_data
```

```
# A tibble: 10,199 x 7
   contributors_name  contributors_postal_code contribution_amount
   <chr>              <chr>                    <chr>
 1 A D'Angelo, Tullio M6A 1P5                  300
 2 A Strazar, Martin  M2M 3B8                  300
 3 A'Court, K Susan   M4M 2J8                  36
 4 A'Court, K Susan   M4M 2J8                  100
 5 A'Court, K Susan   M4M 2J8                  100
 6 Aaron, Robert B    M6B 1H7                  250
 7 Abadi, Babak       M5S 2W7                  500
 8 Abadi, Babak       M5S 2W7                  500
 9 Abadi, David       M5S 2W7                  300
10 Abate, Frank       L4H 2K7                  150
# i 10,189 more rows
# i 4 more variables: contribution_type_desc <chr>,
#   contributor_type_desc <chr>, candidate <chr>, office <chr>
```

The contributor_type_desc and contributon_type_desc should be a categorical variable, so we need to change the format to a factor, instead of just character. The contribution_amount should be a numerical variable, so we need to change the format to a numeric, instead of character.

```
mayoral_campaign_data$contributor_type_desc <- as.factor(mayoral_campaign_data$contributor

mayoral_campaign_data$contribution_type_desc <- as.factor(mayoral_campaign_data$contributi

mayoral_campaign_data$contribution_amount <- as.numeric(mayoral_campaign_data$contribution

mayoral_campaign_data
```

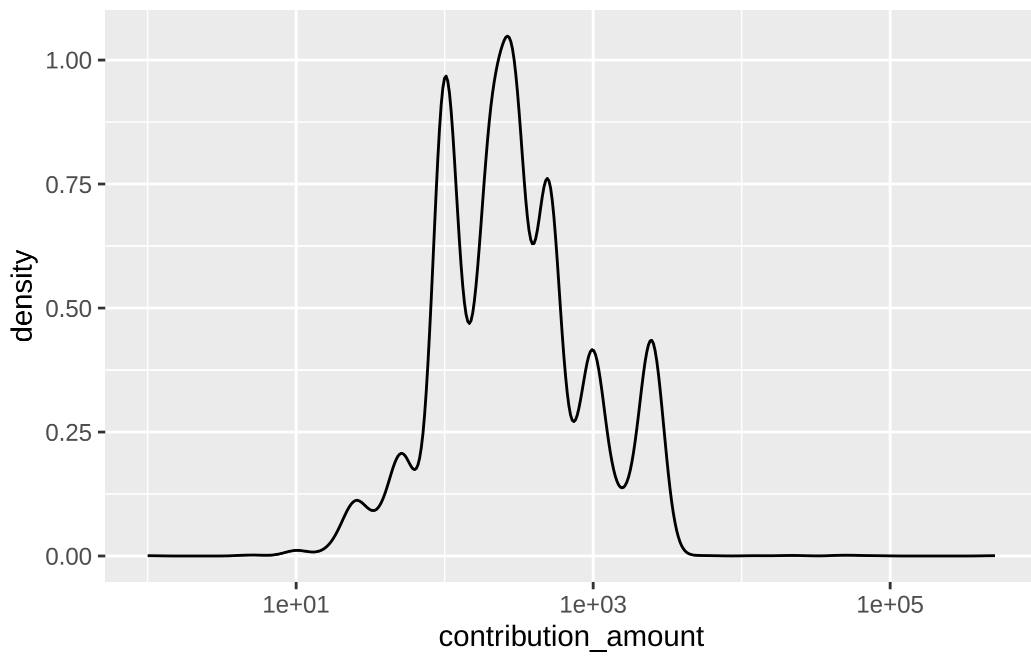
```
# A tibble: 10,199 x 7
   contributors_name  contributors_postal_code contribution_amount
```

```
   <chr>              <chr>                                      <dbl>
 1 A D'Angelo, Tullio M6A 1P5                                      300
 2 A Strazar, Martin  M2M 3B8                                      300
 3 A'Court, K Susan   M4M 2J8                                       36
 4 A'Court, K Susan   M4M 2J8                                      100
 5 A'Court, K Susan   M4M 2J8                                      100
 6 Aaron, Robert B    M6B 1H7                                      250
 7 Abadi, Babak       M5S 2W7                                      500
 8 Abadi, Babak       M5S 2W7                                      500
 9 Abadi, David       M5S 2W7                                      300
10 Abate, Frank       L4H 2K7                                      150
# i 10,189 more rows
# i 4 more variables: contribution_type_desc <fct>,
#   contributor_type_desc <fct>, candidate <chr>, office <chr>
```

Question 5

The amount of contribution is gathered in the middle, and does not seem having too many outliers.

```
mayoral_campaign_data |>
  ggplot() +
  geom_density(aes(x = contribution_amount), bw = .08) +
  scale_x_log10()
```



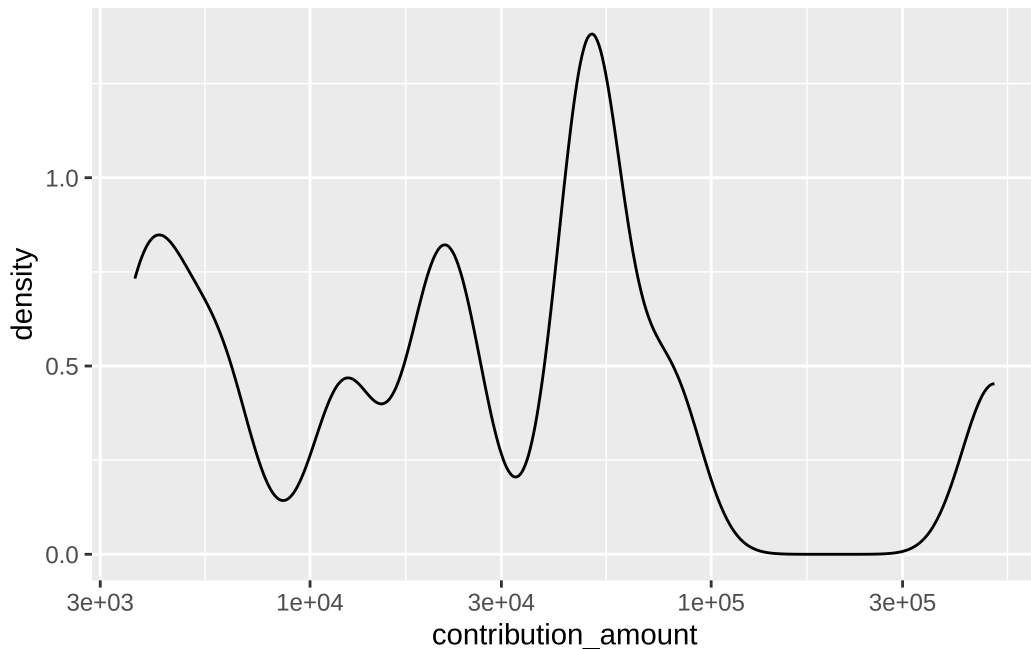
To explore the extreme values area, we need to sort the contribution amount.

```
mayoral_campaign_data |>
  arrange(-contribution_amount)
```

```
# A tibble: 10,199 x 7
   contributors_name contributors_postal_code contribution_amount
   <chr>             <chr>                                  <dbl>
 1 Ford, Doug        M9A 2C3                              508225.
 2 Ford, Rob         M9A 3G9                               78805.
 3 Ford, Doug        M9A 2C3                               50000
 4 Ford, Rob         M9A 3G9                               50000
 5 Ford, Rob         M9A 3G9                               50000
 6 Goldkind, Ari     M5P 1P5                               23624.
 7 Ford, Rob         M9A 3G9                               20000
 8 Ford, Rob         M9A 3G9                               12210
 9 Di Paola, Rocco   M3H 2T1                                6000
10 Thomson, Sarah    M4W 2X6                                4426.
# i 10,189 more rows
# i 4 more variables: contribution_type_desc <fct>,
#   contributor_type_desc <fct>, candidate <chr>, office <chr>
```

Here is the density of the contribution amount over 2,500.

```
mayoral_campaign_data |>
  filter(contribution_amount>2500) |>
  ggplot() +
  geom_density(aes(x = contribution_amount), bw = .08) +
  scale_x_log10()
```



There a couple of donors who contributed multiple times, such as Ford Doug or Ford Rob. In addition, most of the cases are the monetary contribution and individual donors.

```
mayoral_campaign_data |>
  filter(contribution_amount>2500)
```

```
# A tibble: 11 x 7
   contributors_name contributors_postal_code contribution_amount
   <chr>             <chr>                                  <dbl>
 1 Di Paola, Rocco   M3H 2T1                                 6000
 2 Ford, Doug        M9A 2C3                               508225.
 3 Ford, Doug        M9A 2C3                                50000
 4 Ford, Rob         M9A 3G9                                20000
```

```
 5 Ford, Rob         M9A 3G9                               50000
 6 Ford, Rob         M9A 3G9                               50000
 7 Ford, Rob         M9A 3G9                               78805.
 8 Ford, Rob         M9A 3G9                               12210
 9 Goldkind, Ari     M5P 1P5                               23624.
10 kindred's Muze    M6H 2W7                                3660
11 Thomson, Sarah    M4W 2X6                                4426.
# i 4 more variables: contribution_type_desc <fct>,
#   contributor_type_desc <fct>, candidate <chr>, office <chr>
```

Question 6

```r
candidate_contribution <- mayoral_campaign_data |>
  group_by(candidate) |>
  summarise(
  total = sum(contribution_amount, na.rm = TRUE),
  mean = mean(contribution_amount, na.rm = TRUE),
  count = n()
)
```

```r
candidate_contribution |>
  arrange(-total) |>
  select(candidate, total) |>
  head(5)
```

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

```r
candidate_contribution |>
  arrange(-mean) |>
  select(candidate, mean) |>
  head(5)
```

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

```
candidate_contribution |>
  arrange(-count) |>
  select(candidate, count) |>
  head(5)
```

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

Question 7

```
non_candidate_contribution <- mayoral_campaign_data |>
  filter(contributors_name != candidate)

non_candidate_contribution <- non_candidate_contribution |>
  group_by(candidate) |>
  summarise(
  total = sum(contribution_amount, na.rm = TRUE),
  mean = mean(contribution_amount, na.rm = TRUE),
  count = n()
  )
```

```
non_candidate_contribution |>
  arrange(-total) |>
  select(candidate, total) |>
  head(5)
```

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

```r
non_candidate_contribution |>
  arrange(-mean) |>
  select(candidate, mean) |>
  head(5)
```

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

```r
non_candidate_contribution |>
  arrange(-count) |>
  select(candidate, count) |>
  head(5)
```

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

Question 8

```r
multiple_contribution <- mayoral_campaign_data |>
  group_by(contributors_name) |>
  summarise(unique_candidates = n_distinct(candidate))
```

```r
sum(multiple_contribution$unique_candidates > 1)
```

```
[1] 184
```