Lab 2: EDA and data visualization

Toufic Ayoub

21/01/23

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

Lab Exercises

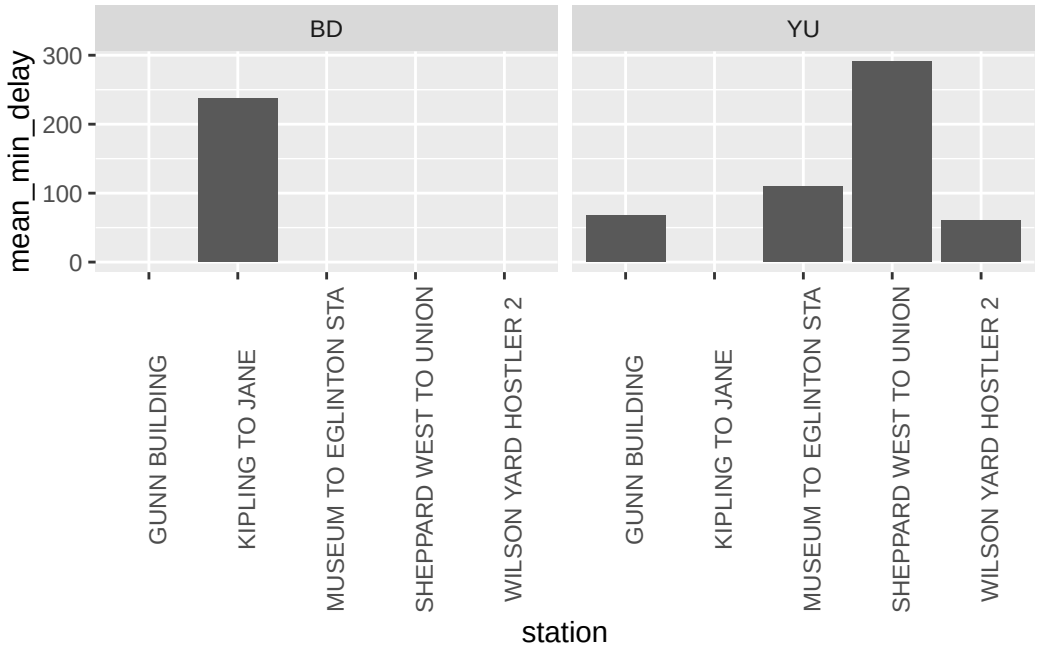
```
res <- list_package_resources("996cfe8d-fb35-40ce-b569-698d51fc683b")
res <- res |> mutate(year = str_extract(name, "202.?"))
delay_2022_ids <- res |> filter(year==2022) |> select(id) |> pull()

delay_2022 <- get_resource(delay_2022_ids)
delay_2022 <- clean_names(delay_2022)
delay_2022 <- delay_2022 |> distinct()
delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
```

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

```
stationDelays <- delay_2022 |>
                  group_by(station) |>
                  summarize(mean_min_delay = mean(min_delay, na.rm = T), line) |>
                  arrange(desc(mean_min_delay))
```

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

```
ggplot(data = stationDelays[1:5,], aes(x=station, y=mean_min_delay)) +
  geom_bar(stat = 'identity') + facet_grid(~line) +
  theme(axis.text.x = element_text(angle = 90))
```



2. Using the `opendatatoronto` package, download the data on mayoral campaign contributions for 2014.

```
res <- list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c")
mayor_2014_ids <- res |> filter(name=="campaign-contributions-2014-data") |>
                    select(id) |>
                    pull()

mayor_2014 <- get_resource(mayor_2014_ids)[[2]]
```

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

```
colnames(mayor_2014) = mayor_2014[1,]
mayor_2014 = mayor_2014[-1,]
mayor_2014 = clean_names(mayor_2014)
```

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.

There are many missing values, for example, contributors' addresses and their relationship to the candidate are predominantly missing all of their values. It would be nice to have these values for easier filtering but the meat and bones of the data is still there even if these columns are completely ignored.

The contribution amount attribute is encoded as a character vector. We obviously would like this to be numeric, so this change is made.

```
skim(mayor_2014)
```

Table 1: Data summary

Name	mayor_2014
Number of rows	10199
Number of columns	13
Column type frequency:	
character	13

Table 1: Data summary

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

```
mayor_2014 = mayor_2014 |>
              mutate(numeric_contribution_amount = as.double(contribution_amount))
```

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.

```
findoutlier <- function(x) {
  return(x < quantile(x, .25) - 1.5*IQR(x) | x > quantile(x, .75) + 1.5*IQR(x))
}
```

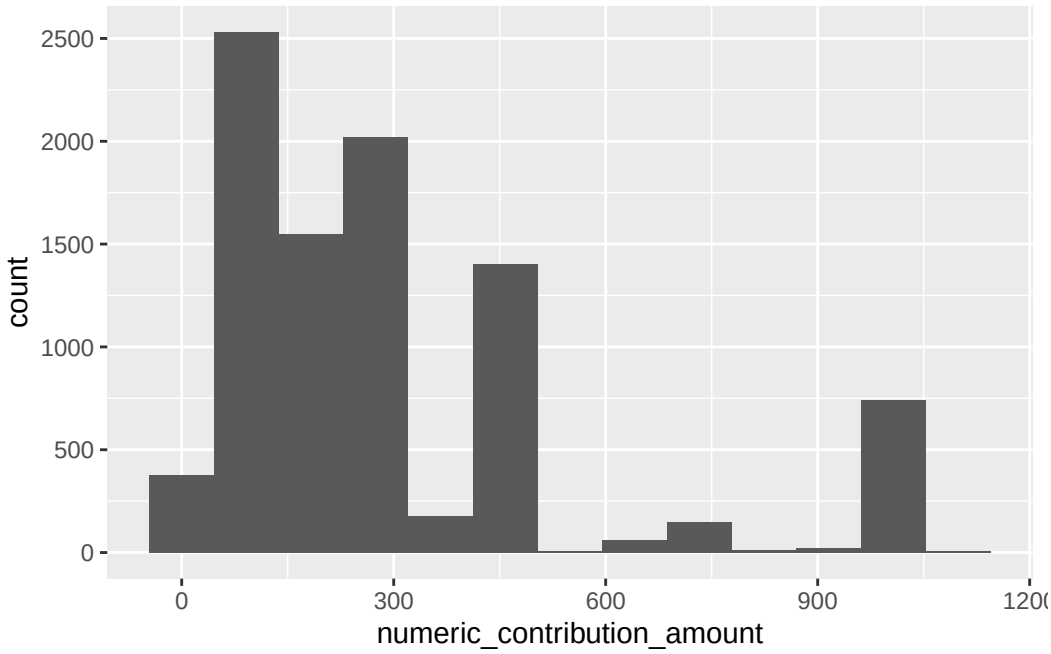
```
mayor_2014_outlier = mayor_2014 |>
                        mutate(outlier = ifelse(findoutlier(numeric_contribution_amount),
                                                numeric_contribution_amount, NA))
```

```r
mayor_2014_outlier |>
  filter(!is.na(outlier)) |>
  group_by(candidate) |>
  summarize(outlier_count = length(outlier))
```

```
# A tibble: 15 x 2
   candidate        outlier_count
   <chr>                    <int>
 1 Billard, Jeff                1
 2 Chow, Olivia               135
 3 Clarke, Kevin                1
 4 Di Paola, Rocco              2
 5 Ford, Doug                  67
 6 Ford, Rob                   33
 7 Gardner, Norman              1
 8 Goldkind, Ari                4
 9 Ritch, Carlie                2
10 Sniedzins, Erwin             3
11 Soknacki, David             29
12 Stintz, Karen               82
13 Syed, Hïmy                  1
14 Thomson, Sarah              8
15 Tory, John                770
```

Around 67% of the outliers are associated with large contributions made towards John Tory. Mr. Tory was the winner of the 2014 election.

```r
mayor_2014_outlier |>
  filter(is.na(outlier)) |>
  ggplot(aes(x=numeric_contribution_amount)) + geom_histogram(bins = 13)
```



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

- total contributions

- mean contribution

- number of contributions

```
mayor_2014_summary = mayor_2014 |>
                     group_by(contributors_name) |>
                     summarize( total_cont = sum(numeric_contribution_amount),
                       mean_cont = mean(numeric_contribution_amount), num_cont =
                     length(numeric_contribution_amount))


mayor_2014_summary_TC = mayor_2014_summary[,c(1,2)] |>
                        arrange(desc(total_cont))

mayor_2014_summary_MC = mayor_2014_summary[,c(1,3)] |>
                          arrange(desc(mean_cont))

mayor_2014_summary_NC = mayor_2014_summary[,c(1,4)] |>
```

```
                        arrange(desc(num_cont))
```

```
head(mayor_2014_summary_TC)
```

```
# A tibble: 6 x 2
  contributors_name  total_cont
  <chr>                   <dbl>
1 Ford, Doug            561225.
2 Ford, Rob             213139.
3 Goldkind, Ari          23624.
4 Thomson, Sarah          6926.
5 Pappalardo, Victor      6300
6 Di Paola, Rocco         6000
```

```
head(mayor_2014_summary_MC)
```

```
# A tibble: 6 x 2
  contributors_name mean_cont
  <chr>                <dbl>
1 Ford, Doug          140306.
2 Ford, Rob            30448.
3 Goldkind, Ari        23624.
4 Di Paola, Rocco       6000
5 kindred's Muze        3660
6 Thomson, Sarah        3463.
```

```
head(mayor_2014_summary_NC)
```

```
# A tibble: 6 x 2
  contributors_name    num_cont
  <chr>                   <int>
1 Italiano, Rob              12
2 Cranston, Jacqueline       10
3 Henery, Marjorie            8
4 Martin, Martha              8
5 Quin, Derek                 8
6 Stewart, Carol              8
```

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

```
mayor_2014_summary_b = mayor_2014 |>
                        group_by(contributors_name) |>
                        filter(contributors_name != candidate) |>
                        summarize( total_cont = sum(numeric_contribution_amount),
                          mean_cont = mean(numeric_contribution_amount), num_cont =
                          length(numeric_contribution_amount))

mayor_2014_summary_TCb = mayor_2014_summary_b[,c(1,2)] |>
                        arrange(desc(total_cont))

mayor_2014_summary_MCb = mayor_2014_summary_b[,c(1,3)] |>
                         arrange(desc(mean_cont))

mayor_2014_summary_NCb = mayor_2014_summary_b[,c(1,4)] |>
                         arrange(desc(num_cont))
```

```
head(mayor_2014_summary_TCb)
```

```
# A tibble: 6 x 2
  contributors_name    total_cont
  <chr>                     <dbl>
1 Pappalardo, Victor         6300
2 Block, Sheila              5500
3 Gazzola, Vern              5300
4 Bachir, Salah              5000
5 Corke, Lawrence            5000
6 Etherington, William       5000
```

```
head(mayor_2014_summary_MCb)
```

```
# A tibble: 6 x 2
  contributors_name   mean_cont
  <chr>                   <dbl>
1 kindred's Muze           3660
2 Achber, Vernon           2500
3 Adam, Michael            2500
4 Aghaei, Saeid            2500
```

```
5 Al Zaibak, Mohammad      2500
6 Allan, David G. P.       2500
```

```
head(mayor_2014_summary_NCb)
```

```
# A tibble: 6 x 2
  contributors_name    num_cont
  <chr>                   <int>
1 Italiano, Rob              12
2 Cranston, Jacqueline       10
3 Henery, Marjorie            8
4 Martin, Martha              8
5 Quin, Derek                 8
6 Stewart, Carol              8
```

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

```
mayor_2014_unique = mayor_2014 |>
                      group_by(contributors_name) |>
                      unique() |>
                      summarize(unique_contributions = length(candidate)) |>
                      filter(unique_contributions > 1)

numMoreThanOne = dim(mayor_2014_unique)[1]
```

There were 1416 contributors who gave money to more than one candidate.