EDA and data visualization

Kishore Basu

22/01/23

Table of contents


```
library(opendatatoronto)
library(tidyverse)
library(stringr)
library(skimr) # EDA
library(visdat) # EDA
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)
delay_2022 <- clean_names(delay_2022)
delay_2022 <- delay_2022 |> distinct()
delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
```

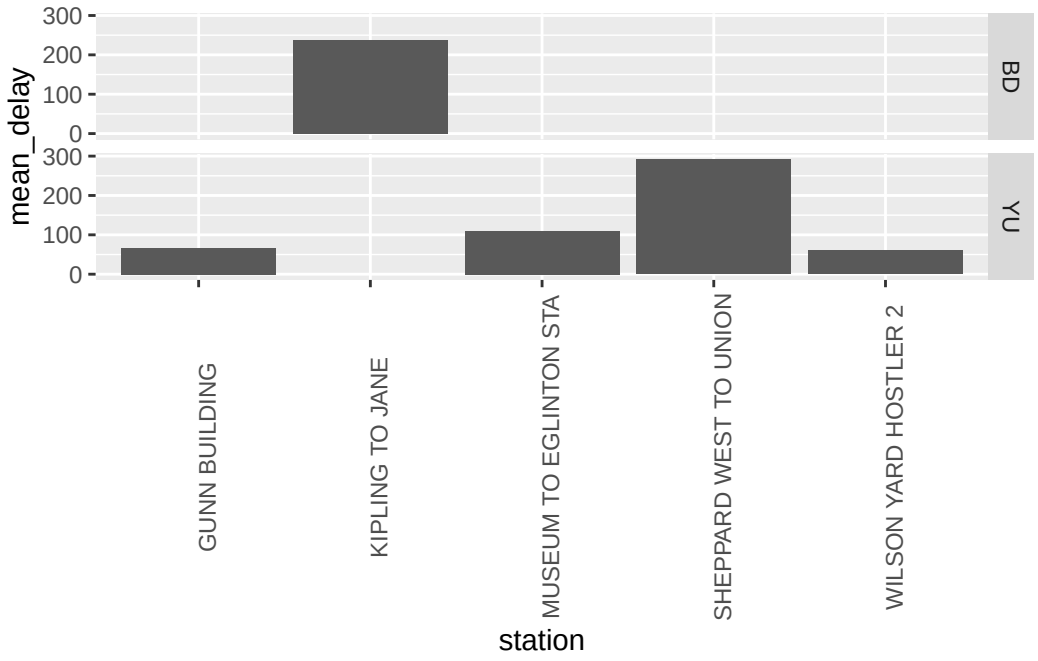
1 Lab Exercises

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`

```
delay_2022 %>%
  group_by(station) %>%
  summarize(station, mean_delay = mean(min_delay, na.rm = T), line) %>%
  arrange(-mean_delay) %>%
  head(5)%>%
  ggplot(aes(x = station, y = mean_delay)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90))+
  facet_grid(vars(line))
```

```
`summarise()` has grouped output by 'station'. 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
   - 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

```
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]]
```

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

```
df <- mayor_2014
```

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

```
names(df) <- df[1,]
```

```
Warning: The `value` argument of `names<-` must be a character vector as of tibble
3.0.0.
```

```
df <- df[2:dim(df)[1],1:dim(df)[2]]
```

```
df <- clean_names(df)
```

```
head(df)
```

```
# 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.

```
skim(df)
```

Table 1: Data summary

Name	df
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

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
office	0	1	5	5	0	1	0
ward	10199	0	NA	NA	0	0	0

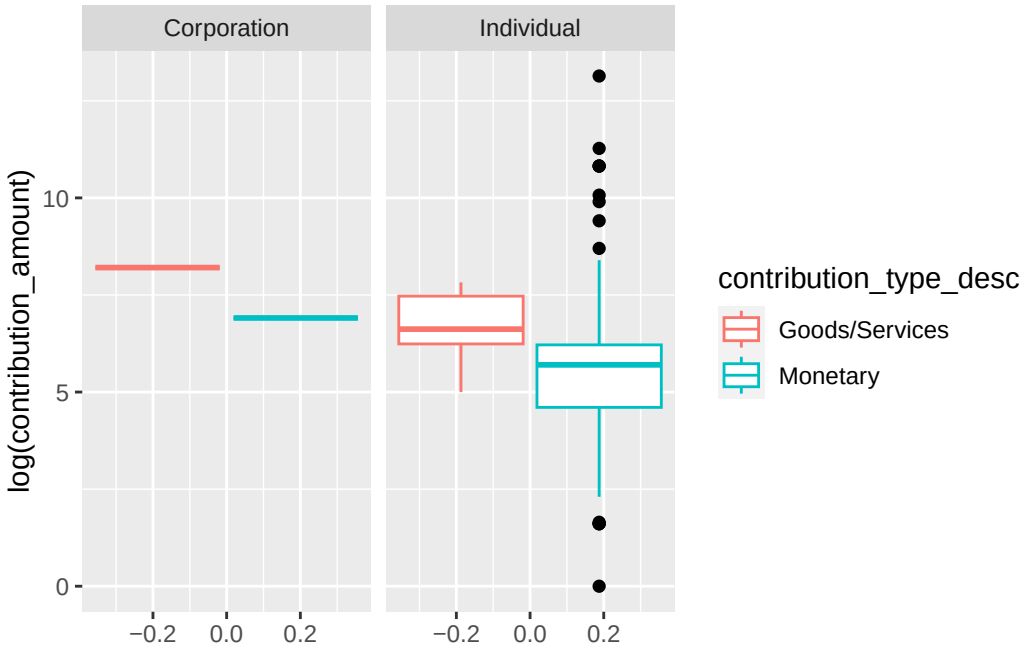
As we can see there are many missing values in the dataset. This is very worrying, as some relationships such as `relationship_to_candidate` might be very influential but we are not able to account for this influence due to a dearth of data. Note that contribution amount should be in floating point precision, so we change that.

```
df['contribution_amount'] <- as.double(df$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.

First, let's look at outliers on a log-scale.

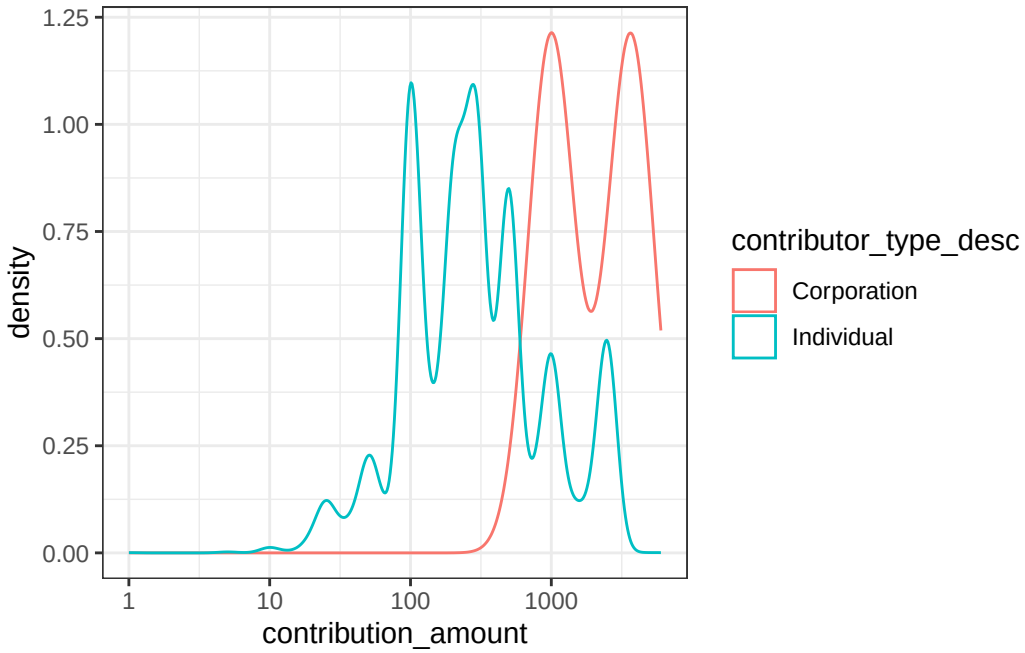
```
df %>%
  ggplot(aes(y = log(contribution_amount), color = contribution_type_desc)) +
  geom_boxplot(outlier.color = 'black', outlier.shape = 16, outlier.size = 2, notch = FALS
  facet_wrap(~contributor_type_desc)
```



There are a lot! Notice that all of these appear to be donated by individuals rather than corporations. This could be because corporations are limited by how much they can legally donate (so they might have large contributions but not outlying large contributions). In addition, they are all monetary donations rather than goods and services.

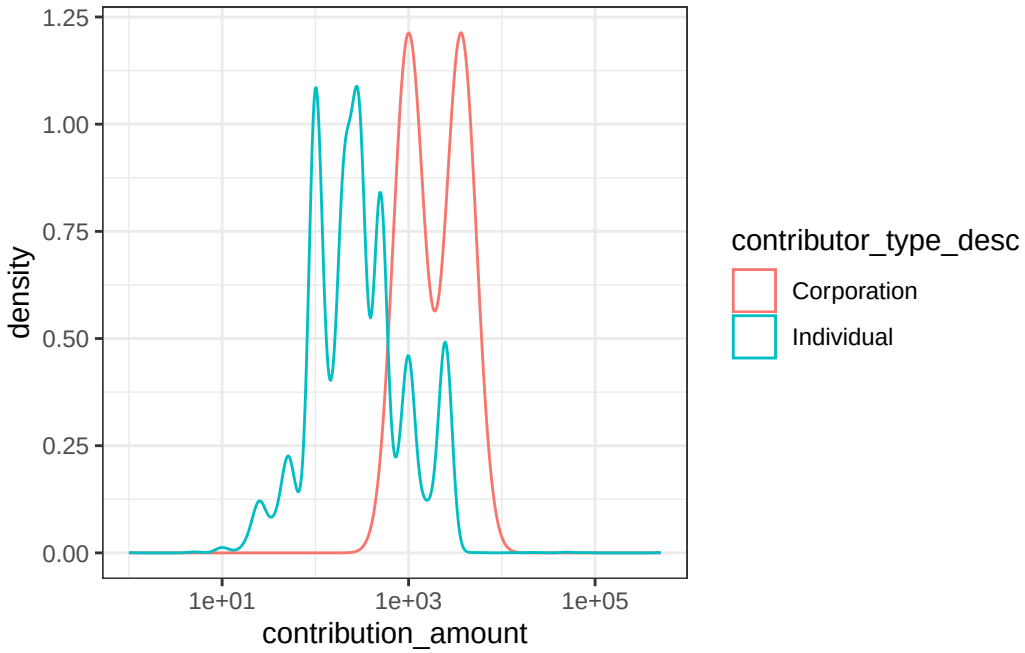
Let's plot the contribution amount without these outliers. We see that corporations tend to contribute more on average!

```
df %>%
  filter(between(contribution_amount, mean(contribution_amount, na.rm=TRUE) - (1.5 * sd(co
  ggplot() +
  geom_density(aes(x = contribution_amount, color = contributor_type_desc)) +
  scale_x_log10() +
  theme_bw()
```



For context, here is without outlier removal.

```
df %>%
  ggplot() +
  geom_density(aes(x = contribution_amount, color = contributor_type_desc)) +
  scale_x_log10() +
  theme_bw()
```



In addition, we can make a new function to find outliers for us using the interquartile range rather than the SD:

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

```
df_outlier <- df %>%
  mutate(outlier = ifelse(findoutlier(contribution_amount), contribution_amount, NA))
```

```
df_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
```

Notice that the vast majority of outliers are to jogn Tory, but others such as Olivia Chow got a lot of donations too.

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

```
df %>%
  group_by(contributors_name) %>%
  summarize(total_contr = sum(contribution_amount), mean_contr = mean(contribution_amount)
  arrange(-total_contr) %>%
  head(5)
```

```
# A tibble: 5 x 4
  contributors_name  total_contr mean_contr num_contr
  <chr>                    <dbl>      <dbl>     <int>
1 Ford, Doug             561225.    140306.         4
2 Ford, Rob              213139.     30448.         7
3 Goldkind, Ari           23624.     23624.         1
4 Thomson, Sarah           6926.      3463.         2
5 Pappalardo, Victor       6300       2100          3
```

```
df %>%
  group_by(contributors_name) %>%
  summarize(total_contr = sum(contribution_amount), mean_contr = mean(contribution_amount)
  arrange(-mean_contr) %>%
  head(5)
```

```
# A tibble: 5 x 4
  contributors_name total_contr mean_contr num_contr
  <chr>                   <dbl>      <dbl>     <int>
1 Ford, Doug            561225.    140306.         4
2 Ford, Rob             213139.     30448.         7
3 Goldkind, Ari          23624.     23624.         1
4 Di Paola, Rocco         6000       6000          1
5 kindred's Muze          3660       3660          1
```

```
df %>%
  group_by(contributors_name) %>%
  summarize(total_contr = sum(contribution_amount), mean_contr = mean(contribution_amount)
  arrange(-num_contr) %>%
  head(5)
```

```
# A tibble: 5 x 4
  contributors_name    total_contr mean_contr num_contr
  <chr>                      <dbl>      <dbl>     <int>
1 Italiano, Rob                751       62.6        12
2 Cranston, Jacqueline        2718      272.         10
3 Henery, Marjorie             900      112.          8
4 Martin, Martha               900      112.          8
5 Quin, Derek                 1350      169.          8
```

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

Group by total contribution:

```
df1 <- df %>%
  group_by(contributors_name) %>%
  filter(contributors_name != candidate) %>%
  summarize(total_cont = sum(contribution_amount))
head(df1[,c(1,2)] %>%  arrange(desc(total_cont)))
```

```
# 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
```

and by mean:

```
df2 <- df %>%
  group_by(contributors_name) %>%
  filter(contributors_name != candidate) %>%
  summarize(mean_cont = mean(contribution_amount))
head(df2[,c(1,2)] %>%  arrange(desc(mean_cont)))
```

```
# 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
```

and by length:

```
df3 <- df %>%
  group_by(contributors_name) %>%
  filter(contributors_name != candidate) %>%
  summarize(num_cont = length(contribution_amount))
head(df3[,c(1,2)] %>%  arrange(desc(num_cont)))
```

```
# 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?

```r
df %>%
  group_by(contributors_name) %>%
  unique() %>%
  summarize(num_donation = length(candidate)) %>%
  filter(num_donation > 1) %>%
  dim()
```

```
[1] 1416    2
```

So 1416 contributors gave money to more than one candidate.