STA2201H Assignment 1

J. Arturo Esquivel

20/01/23

Table of contents


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

1 Overdispersion

a) First, let's recall that if $X \sim Poisson(\lambda)$, $E[X] = Var(X) = \lambda$. Thus, $E[Y \mid \theta] = Var(Y \mid \theta) = \mu\theta$. Now, by the law of total expectation and assuming μ to be a constant

$$E[Y] = E[E[Y \mid \theta]] = E[\mu\theta] = \mu E[\theta] = \mu.$$

Similarly, by the law of total variance

$$\begin{aligned} Var(Y) &= E[Var(Y \mid \theta)] + Var(E[Y \mid \theta]) \\ &= E[\mu\theta] + Var(\mu\theta) \\ &= \mu E[\theta] + \mu^2 Var(\theta) \\ &= \mu + \mu^2\sigma^2 = \mu(1 + \mu\sigma^2). \end{aligned}$$

b) Considering the Gamma distribution in terms of the shape parameter α and the rate parameter $\eta = 1/\beta$, the distribution of θ is given by

$$f(\theta; \alpha, \eta) = \frac{1}{\Gamma(\alpha)} \theta^{\alpha-1} \eta^{\alpha} \exp(-\theta\eta).$$

Thus,

$$\begin{aligned} f(Y, \theta; \alpha, \eta) &= f(Y \mid \theta) f(\theta; \alpha, \eta) \\ &= \frac{\theta^{Y} e^{-\theta}}{Y!} \frac{1}{\Gamma(\alpha)} \theta^{\alpha-1} \eta^{\alpha} \exp(-\theta\eta). \end{aligned}$$

And to get the marginal distribution of Y we need to integrate over all the space of θ.

$$\begin{aligned} f(Y; \alpha, \eta) &= \int_0^{\infty} \frac{\theta^{Y} e^{-\theta}}{Y!} \frac{1}{\Gamma(\alpha)} \theta^{\alpha-1} \eta^{\alpha} e^{-\theta\eta} d\theta \\ &= \int_0^{\infty} \frac{\theta^{Y+\alpha-1} e^{-\theta(\eta+1)} \eta^{\alpha}}{Y! \Gamma(\alpha)} d\theta. \end{aligned}$$

And the function inside the integral corresponds to the kernel of a $Gamma(Y + \alpha, \eta + 1)$. We can complete the integral so that it integrates to 1 as

$$f(Y; \alpha, \eta) = \frac{\Gamma(Y+\alpha)\eta^{\alpha}}{Y!\Gamma(\alpha)(\eta+1)^{Y+\alpha}} \int_0^{\infty} \frac{\theta^{Y+\alpha-1} e^{-\theta(\eta+1)} (\eta+1)^{Y+\alpha}}{\Gamma(Y+\alpha)} d\theta$$

$$= \frac{\Gamma(Y+\alpha)\eta^{\alpha}}{Y!\Gamma(\alpha)(\eta+1)^{Y+\alpha}}$$

$$= \frac{\Gamma(Y+\alpha)}{Y!\Gamma(\alpha)(\eta+1)^{Y}} \left(\frac{\eta}{\eta+1}\right)^{\alpha}, \qquad \text{since } \Gamma(n) = (n-1)!$$

$$= \frac{(Y+\alpha-1)!}{Y!(\alpha-1)!(\eta+1)^{Y}} \left(\frac{\eta}{(\eta+1)}\right)^{\alpha}$$

$$= \binom{Y+\alpha-1}{Y-1} \left(\frac{1}{(\eta+1)}\right)^{Y} \left(\frac{\eta}{\eta+1}\right)^{\alpha}.$$

And substituting $\eta = 1/\beta$

$$f(Y;\alpha,\beta) = \binom{Y+\alpha-1}{Y-1}\left(\frac{1}{\frac{1}{\beta}+1}\right)^{Y}\left(\frac{\frac{1}{\beta}}{\frac{1}{\beta}+1}\right)^{\alpha}$$

$$= \binom{Y+\alpha-1}{Y-1}\left(\frac{\beta}{\beta+1}\right)^{Y}\left(\frac{1}{\beta+1}\right)^{\alpha}.$$

Which corresponds to $Y \sim NB(\alpha, 1/(\beta+1))$.

c) α and β have to be such that the mean and variance of θ are as defined in a). I.e., $E[\theta] = 1$ and $Var(\theta) = \sigma^2$. If θ follows a $Gamma(\alpha,\beta)$ distribution, $E[\theta] = \alpha\beta$ and $Var(\theta) = \alpha\beta^2$. Hence, α and β are the solution to the system

$$\alpha\beta = 1$$
$$\alpha\beta^2 = \sigma^2.$$

Solving the system yields that $\alpha = 1/\sigma^2$ and $\beta = \sigma^2$.

```
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)
# note: I obtained these codes from the 'id' column in the `res` object above
delay_codes <- get_resource("3900e649-f31e-4b79-9f20-4731bbfd94f7")
delay_data_codebook <- get_resource("ca43ac3d-3940-4315-889b-a9375e7b8aa4")

delay_2022 <- delay_2022 |> distinct()

## Removing the observations that have non-standardized lines

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

delay_2022 <- delay_2022 |>
  left_join(delay_codes |> rename(code = `SUB RMENU CODE`, code_desc = `CODE DESCRIPTION..


delay_2022 <- delay_2022 |>
  mutate(code_srt = ifelse(line=="SRT", code, "NA")) |>
```

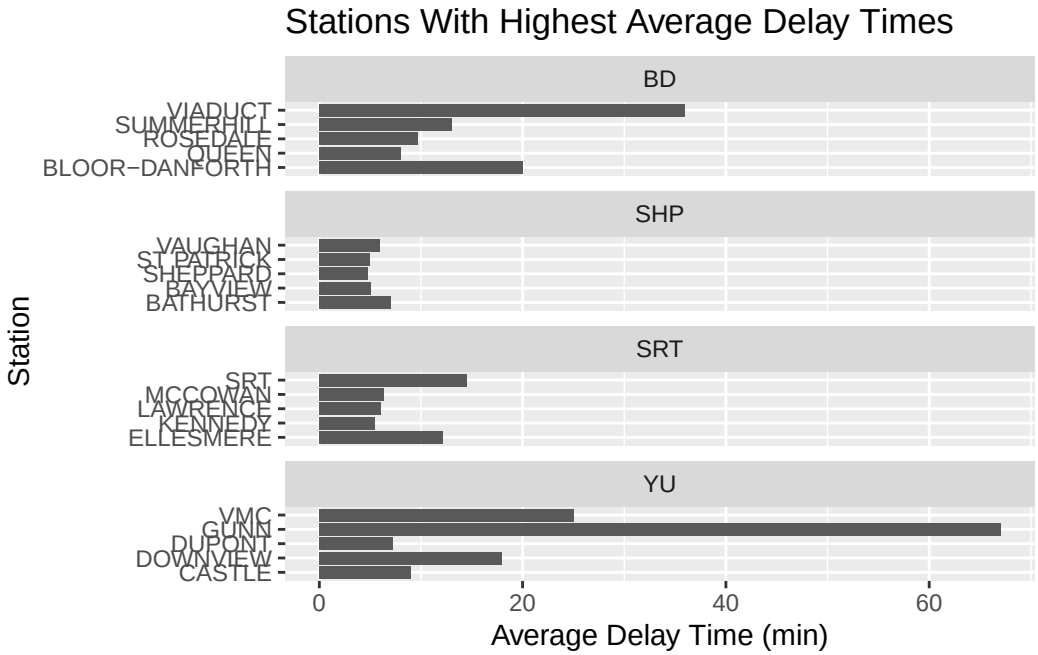
```
  left_join(delay_codes |> rename(code_srt = `SRT RMENU CODE`, code_desc_srt = `CODE DESCR
  mutate(code = ifelse(code_srt=="NA", code, code_srt),
         code_desc = ifelse(is.na(code_desc_srt), code_desc, code_desc_srt)) |>
  select(-code_srt, -code_desc_srt)

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

2 Lab Exercises

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

```
delay_2022 |>
  group_by(line, station_clean) |>
  summarise(mean_delay = mean(min_delay)) |>
  arrange(-mean_delay) |>
  slice(1:5) |>
  ggplot(aes(x = station_clean, y = mean_delay)) +
  geom_col() +
  facet_wrap(vars(line),
             scales = "free_y",
             nrow = 4) +
  coord_flip() +
  labs(title = "Stations With Highest Average Delay Times",
       x = "Station",
       y = "Average Delay Time (min)")
```



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

```
res2 <- list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c")
camp_data <- get_resource("5b230e92-0a22-4a15-9572-0b19cc222985")[[2]]
```

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

```
colnames(camp_data) <- camp_data[1, ]
camp_data <- clean_names(camp_data) |> slice(-1)
```

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 multiple variables with missing values. However, most of them are not worrying or surprising. E.g., the `contributors_address` is missing in most cases. But according

with the "read me" file it is only included when the contribution comes from an organization. Almost all contributions correspond to individuals and so there is no corresponding address in the data. Variables relevant for most analyses of interest, such as `contribution_amount`, `contributors_postal_code`, or `contributors_name` have no missing data.

```
skim(camp_data)
```

Table 1: Data summary

Name	camp_data
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
office	0	1	5	5	0	1	0
ward	10199	0	NA	NA	0	0	0

The only variable in incorrect format is `contribution_amount` which should be numerical and is stored as character. It is corrected below.

```
head(camp_data)
```

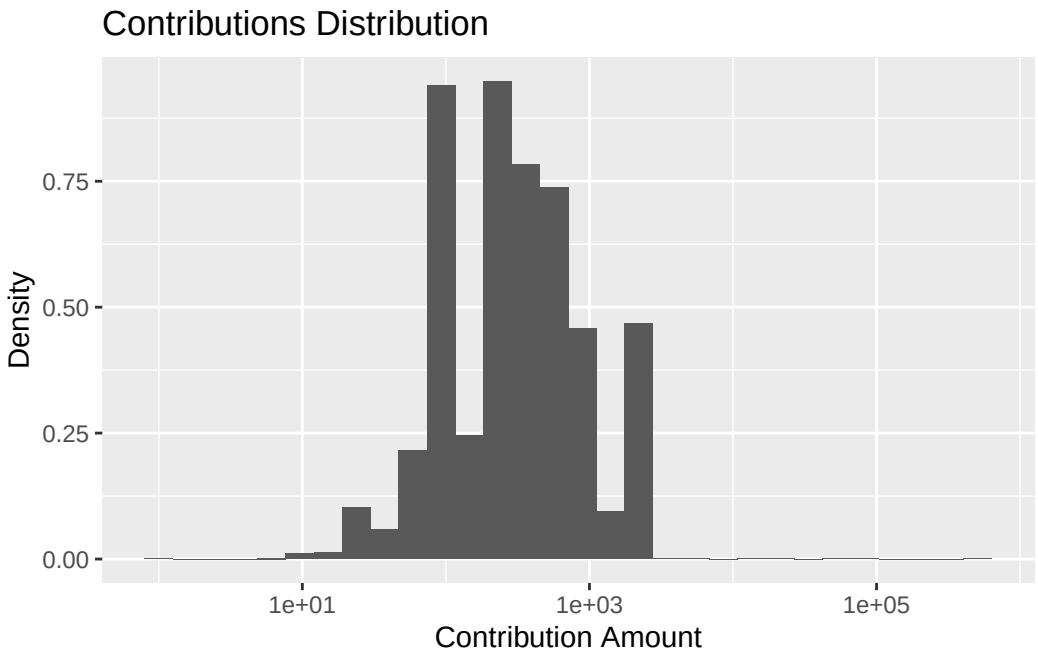
```
# A tibble: 6 x 13
  contributors~1 contr~2 contr~3 contr~4 contr~5 goods~6 contr~7 relat~8 presi~9
  <chr>          <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
1 A D'Angelo, T~ <NA>    M6A 1P5 300     Moneta~ <NA>    Indivi~ <NA>    <NA>
2 A Strazar, Ma~ <NA>    M2M 3B8 300     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
```

```
camp_data$contribution_amount <- as.numeric(camp_data$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.

Looking at the range of the plot, and at the density outside the center, it is clear that there exists some entries with outlying (abnormally large) contribution amounts.

```
ggplot(data = camp_data) +
  geom_histogram(aes(x = contribution_amount, y = after_stat(density)),
                 position = 'dodge') +
  scale_x_log10() +
  labs(title = "Contributions Distribution",
       x = "Contribution Amount",
       y = "Density")
```



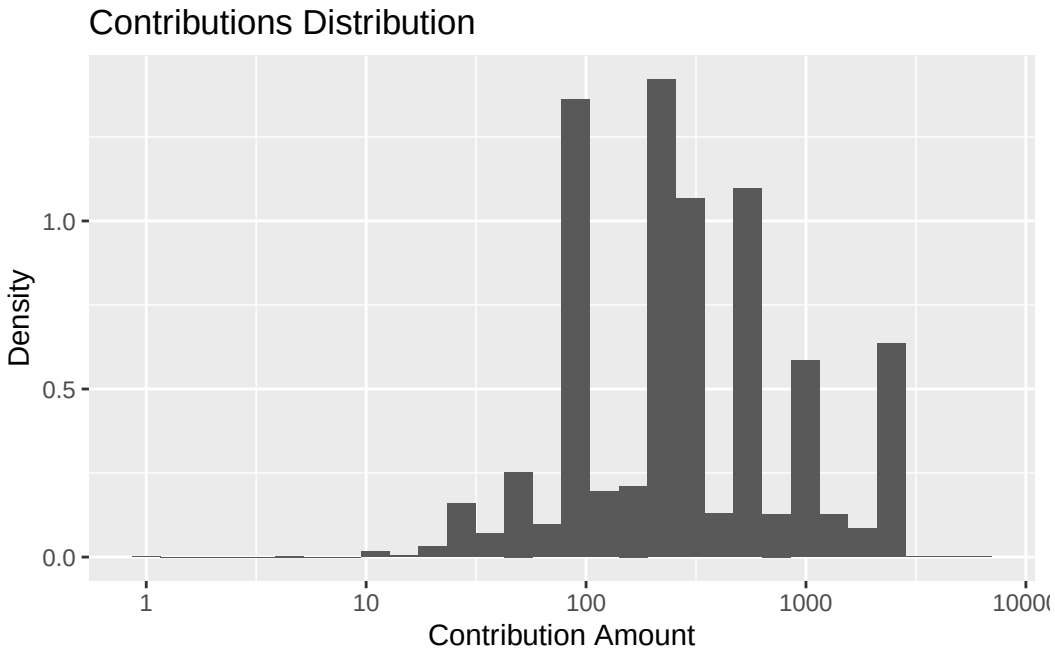
We can look at the top 8 largest contributions in the data (all other contributions were of $6000 or less). The first thing to notice is that all of these contributions were made by the candidates themselves. Also, all were made by Doug Ford, Rob Ford, or Ari Goldkind. The contributions they made to their own campaigns are clearly atypical.

```
camp_data |>
  select(contribution_amount, contributors_name, candidate) |>
  arrange(-contribution_amount) |>
  slice(1:8)
```

```
# A tibble: 8 x 3
  contribution_amount contributors_name candidate
                <dbl> <chr>             <chr>
1              508225. Ford, Doug        Ford, Doug
2               78805. Ford, Rob         Ford, Rob
3               50000  Ford, Doug        Ford, Doug
4               50000  Ford, Rob         Ford, Rob
5               50000  Ford, Rob         Ford, Rob
6               23624. Goldkind, Ari     Goldkind, Ari
7               20000  Ford, Rob         Ford, Rob
8               12210  Ford, Rob         Ford, Rob
```

The distribution of contribution amounts is shown below, excluding the 8 cases mentioned above.

```
camp_data |>
  filter(contribution_amount <12000) |>
ggplot() +
  geom_histogram(aes(x = contribution_amount, y = after_stat(density)),
                 position = 'dodge') +
  scale_x_log10() +
  labs(title = "Contributions Distribution",
       x = "Contribution Amount",
       y = "Density")
```



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

- total contributions

```
camp_data |>
  group_by(candidate) |>
  summarise(total_contributions = sum(contribution_amount)) |>
  arrange(-total_contributions) |>
  slice(1:5)
```

```
# 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
```

- mean contribution

```
camp_data |>
  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
```

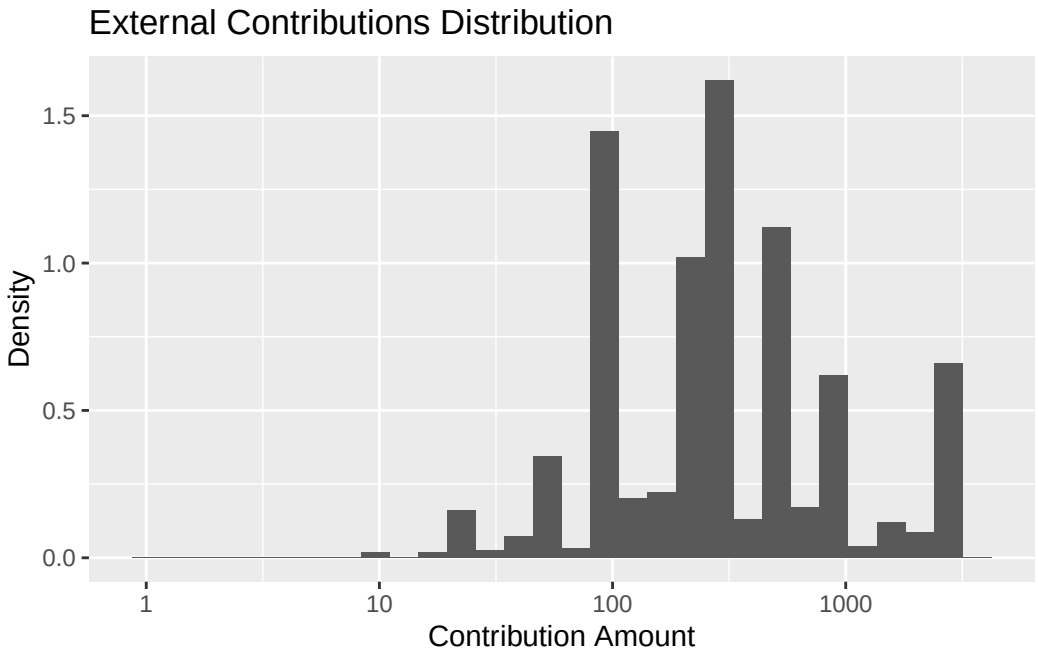
- number of contributions

```
camp_data |>
  count(candidate, sort = TRUE, name = "Number of Contributions") |>
  slice(1:5)
```

```
# A tibble: 5 x 2
  candidate       `Number of Contributions`
  <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.

```
camp_data |>
  filter(contributors_name != candidate) |>
  ggplot() +
  geom_histogram(aes(x = contribution_amount, y = after_stat(density)),
                 position = 'dodge') +
  scale_x_log10() +
  labs(title = "External Contributions Distribution",
       x = "Contribution Amount",
       y = "Density")
```



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

```
contributions <- camp_data |>
  count(contributors_name, candidate, sort = TRUE, name = "number_of_contributions")

mult_cons <- get_dupes(contributions[ , 1]) |> distinct() |> nrow()
```

There were 184 contributors supporting more than one candidate.