EDA and data visualization

Harriet Ware

January 22 2021

Load in all the packages we need:

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

1 Lab Exercises

1. 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")
# obtained code from searching data frame above
campaign_2014 <- get_resource("d99bb1f3-949a-4497-bb96-c93bbd203130")
# obtained code from searching data frame above
mayor_2014<-campaign_2014[2][[1]]
```

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

```
mayor_2014<-mayor_2014 %>%
  row_to_names(row_number = 1)
mayor_2014<-clean_names(mayor_2014)
```

3. 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(mayor_2014)
```

Table 1: Data summary

Name	mayor_2014
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

We can make the contribution amount variable numeric instead of character.

```r
mayor_2014<-mayor_2014%>%mutate(contribution_amount = as.numeric(contribution_amount))

skim(mayor_2014)
```

Table 3: Data summary

Name	mayor_2014
Number of rows	10199
Number of columns	13
Column type frequency:	
character	12
numeric	1
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_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

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
contribution_amount	0	1	607.95	5211.31	1	100	300	500	508224.7	

Check values of variables.

```r
unique(mayor_2014$contribution_type_desc)
```

```
## [1] "Monetary"       "Goods/Services"
```

```r
unique(mayor_2014$contributor_type_desc)
```

```
## [1] "Individual"  "Corporation"
```

```r
unique(mayor_2014$candidate)
```

```
##  [1] "Ford, Rob"          "Chow, Olivia"       "Tory, John"
##  [4] "Stintz, Karen"      "Ford, Doug"         "Soknacki, David"
##  [7] "Underhill, Richard" "Thomson, Sarah"     "Goldkind, Ari"
## [10] "Baskin, Morgan"     "Tiwari, Ramnarine"  "Di Paola, Rocco"
## [13] "Clarke, Kevin"      "Emond, Ryan"        "French, James"
## [16] "Billard, Jeff"      "Ritch, Carlie"      "Gardner, Norman"
## [19] "Kalevar, Chai"      "Khomenko, Klim"     "Lee, Dewitt"
## [22] "Mernagh, Matt"      "Ruel, Jim"          "Sniedzins, Erwin"
## [25] "Syed, Hïmy"         "Walker, Daniel"     "Lam, Steven"
```

```r
unique(mayor_2014$office)
```

```
## [1] "Mayor"
```

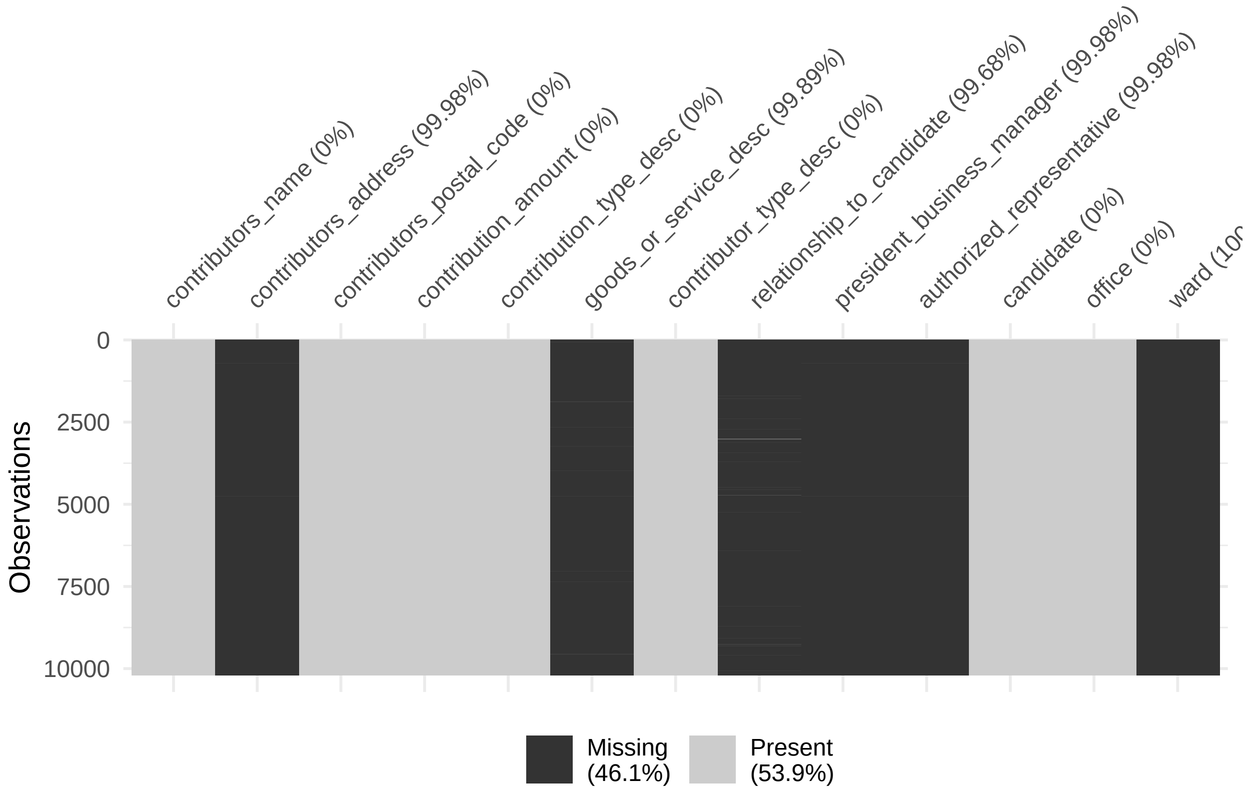
```r
unique(mayor_2014$relationship_to_candidate)
```

```
## [1] NA          "Candidate" "Spouse"
```

Everything looks good above.

Now look to see how many NAs by variable.

```
vis_miss(mayor_2014)
```

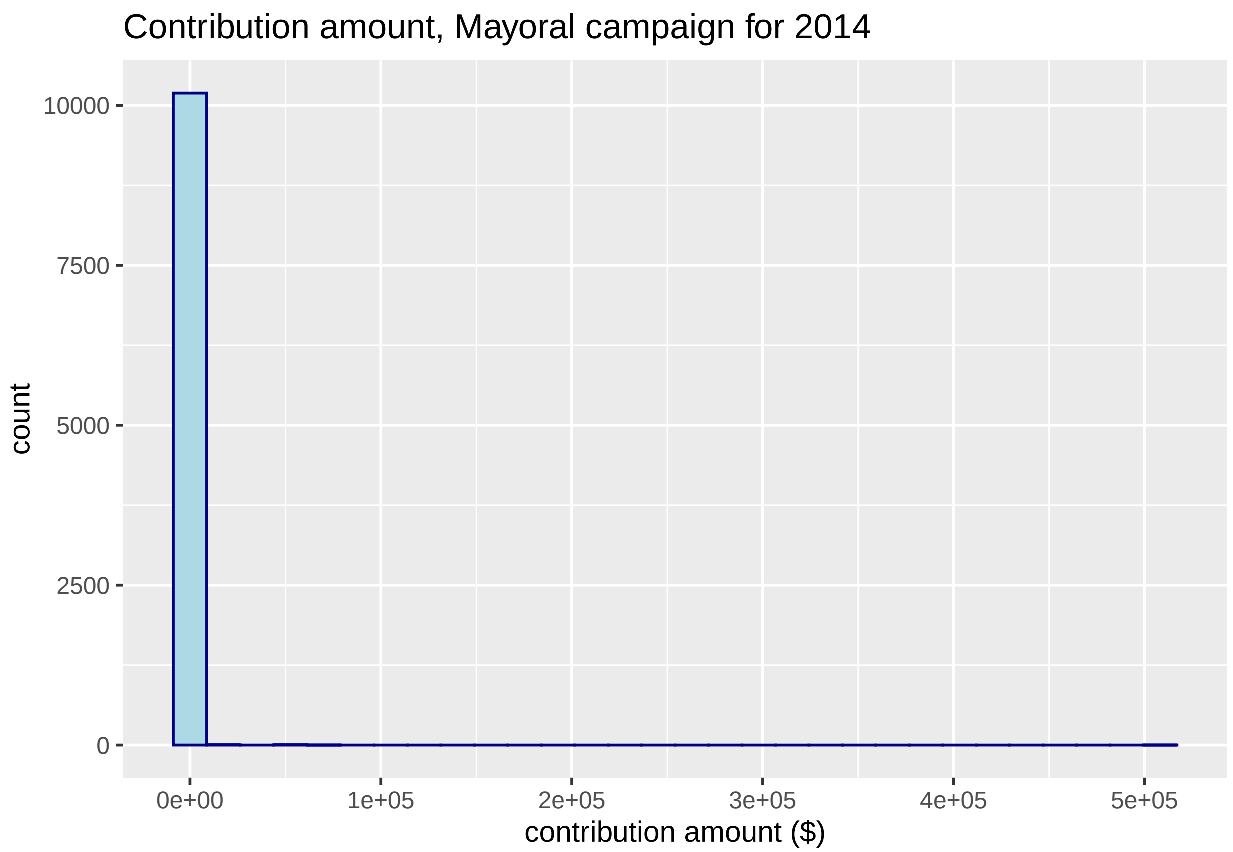


It seems like the key variables are all populated, so we don't need to worry about missing values.

4. 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 plot the distribution of all observations.

```
ggplot(data = mayor_2014, aes(contribution_amount)) +
  geom_histogram(fill = "lightblue", color = "navy") +
  ggtitle("Contribution amount, Mayoral campaign for 2014") +
  xlab("contribution amount ($)")
```



We can summarize the percentiles of the distribution to get a sense of the outliers with large contribution amounts.

```
mayor_2014 %>%
  summarize(contribution_amount_q = quantile(contribution_amount,
            probs=c(0,.25,.5,.75,.99,1),na.rm = T))
```

```
## # A tibble: 6 x 1
##   contribution_amount_q
##                   <dbl>
## 1                     1
## 2                   100
## 3                   300
## 4                   500
## 5                  2500
## 6                508225.
```

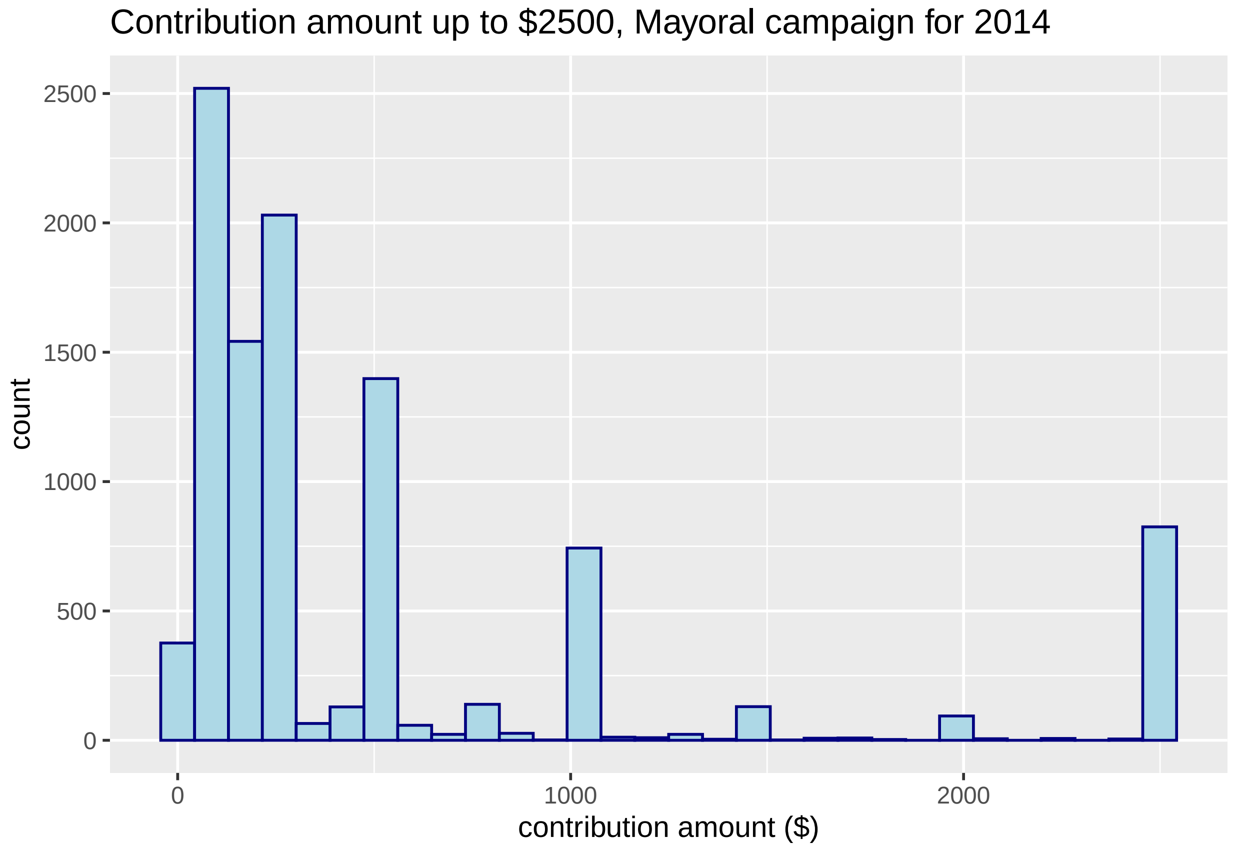
The 99% percentile of the distribution of contribution amounts is $2500. We can look more closely at the top 1%.

```
mayor_2014 %>%
  filter(contribution_amount>2500)
```

```
## # A tibble: 11 x 13
##    contributors_na~ contributors_ad~ contributors_po~ contribution_am~
##    <chr>             <chr>             <chr>                        <dbl>
##  1 Di Paola, Rocco  <NA>              M3H 2T1                       6000
##  2 Ford, Doug       <NA>              M9A 2C3                     508225.
##  3 Ford, Doug       <NA>              M9A 2C3                      50000
##  4 Ford, Rob        <NA>              M9A 3G9                      20000
##  5 Ford, Rob        <NA>              M9A 3G9                      50000
##  6 Ford, Rob        <NA>              M9A 3G9                      50000
##  7 Ford, Rob        <NA>              M9A 3G9                      78805.
##  8 Ford, Rob        <NA>              M9A 3G9                      12210
##  9 Goldkind, Ari    <NA>              M5P 1P5                      23624.
## 10 kindred's Muze   723 Dovercourt ~ M6H 2W7                       3660
## 11 Thomson, Sarah   <NA>              M4W 2X6                       4426.
## # ... with 9 more variables: 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>
```

We see that all but one of the contributions greater than $2500 come from the candidates themselves. The remaining one comes from photography services. We can plot the distribution of contributions without these 11 outliers to get a better sense of the majority of the data.

```
mayor_2014_to_plot<-mayor_2014 %>%
  filter(contribution_amount<=2500)
ggplot(data = mayor_2014_to_plot, aes(contribution_amount)) +
  geom_histogram(fill = "lightblue", color = "navy") +
  ggtitle("Contribution amount up to $2500, Mayoral campaign for 2014") +
  xlab("contribution amount ($)")
```



Now we see that the distribution is right-skewed with the median contribution around $300.

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

```
mayor_2014 %>% group_by(candidate) %>%
  summarize(total_contribution = sum(contribution_amount, na.rm = T)) %>%
  slice_max(total_contribution,n=5)
```

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

```
mayor_2014 %>% group_by(candidate) %>%
  summarize(mean_contribution = mean(contribution_amount, na.rm = T)) %>%
  slice_max(mean_contribution,n=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
```

```
mayor_2014 %>% group_by(candidate) %>%
  summarize(num_contribution = n()) %>%
  slice_max(num_contribution,n=5)
```

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

6. *Repeat 5 but without contributions from the candidates themselves.*

```
mayor_2014 %>% filter(relationship_to_candidate!="Candidate"|is.na(relationship_to_candidate)) %>%
  group_by(candidate) %>%
  summarize(total_contribution = sum(contribution_amount, na.rm = T)) %>%
  slice_max(total_contribution,n=5)
```

```
## # A tibble: 5 x 2
##   candidate     total_contribution
##   <chr>                      <dbl>
## 1 Tory, John              2765369.
## 2 Chow, Olivia            1635766.
## 3 Ford, Doug               331173.
## 4 Stintz, Karen            242805
## 5 Ford, Rob                174510.
```

```
mayor_2014 %>% filter(relationship_to_candidate!="Candidate"|is.na(relationship_to_candidate)) %>%
  group_by(candidate) %>%
  summarize(mean_contribution = mean(contribution_amount, na.rm = T)) %>%
  slice_max(mean_contribution,n=5)
```

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

```
mayor_2014 %>% filter(relationship_to_candidate!="Candidate"|is.na(relationship_to_candidate)) %>%
  group_by(candidate) %>%
  summarize(num_contribution = n()) %>%
  slice_max(num_contribution,n=5)
```

```
## # A tibble: 5 x 2
##   candidate       num_contribution
##   <chr>                      <int>
## 1 Chow, Olivia                5707
## 2 Tory, John                  2601
## 3 Ford, Doug                   608
## 4 Ford, Rob                    531
## 5 Soknacki, David              314
```

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

```
mayor_2014 %>%
  group_by(contributors_name) %>%
  summarise(num_candidate=n_distinct(candidate)) %>%
  filter(num_candidate>1)
```

```
## # A tibble: 184 x 2
##    contributors_name num_candidate
##    <chr>                     <int>
##  1 Abadi, Babak                  2
##  2 Adams, Michael                2
##  3 Anga, John                    2
##  4 Argyris, Katerina             2
##  5 Atkinson, Tom                 2
##  6 Aziz, Peter                   2
##  7 Bachir, Salah                 2
##  8 Bajwa, Joginder               2
##  9 Baker, Norma                  2
## 10 Banwait, Rav                  2
## # ... with 174 more rows
```

184 contributors gave money to more than one candidate.