

lab2

Kerry Hu

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'.

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

```
-- Attaching packages ----- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.5.0
v readr   2.1.3      v forcats 0.5.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
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

```
library(lubridate)
```

Loading required package: timechange

Attaching package: 'lubridate'

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

date, intersect, setdiff, union

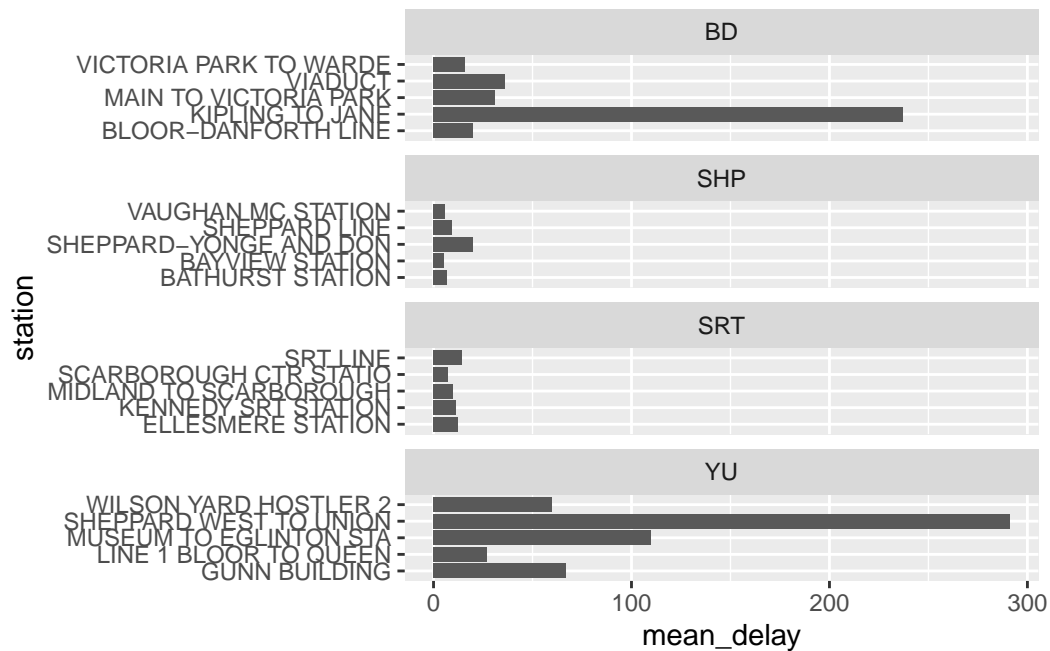
```
library(ggrepel)
```

```
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)
delay_2022 <- delay_2022 |> filter(line %in% c("BD", "YU", "SHP", "SRT"))
delay_2022 |>
  group_by(line, station) |>
  summarise(mean_delay = mean(min_delay)) |>
  arrange(-mean_delay) |>
  slice(1:5) |>
  ggplot(aes(x = station,
             y = mean_delay)) +
  geom_col() + facet_wrap(vars(line),
                          scales = "free_y",
                          nrow = 4)+ coord_flip()
```

`summarise()` has grouped output by 'line'. 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.

```
library(opendatatoronto)
all_data <- list_packages(limit = 500) |> filter(str_detect(title,"Campaign"))

res <- list_package_resources("f6651a40-2f52-46fc-9e04-b760c16edd5c") # obtained code from
res <- res |> mutate(year = str_extract(name, "2014"))

campaign_2014_ids <- res |> filter(year==2014) |> select(id) |> pull()
campaign1_2014<- get_resource(campaign_2014_ids[1])
```

New names:
New names:
New names:
New names:

New names:

New names:

New names:

* `` -> `...2`

* `` -> `...3`

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

```
library(janitor)
head(campaign1_2014[2]`$`2_Mayor_Contributions_2014_election.xls`)
```

```
# A tibble: 6 x 13
  2014 Munic~1 ...2 ...3 ...4 ...5 ...6 ...7 ...8 ...9 ...10 ...11 ...12
  <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 Contributor~ Cont~ Cont~ Cont~ Cont~ Good~ Cont~ Rela~ Pres~ Auth~ Cand~ Offi~
2 A D'Angelo,~ <NA> M6A ~ 300 Mone~ <NA> Indi~ <NA> <NA> <NA> Ford~ Mayor
3 A Strazar, ~ <NA> M2M ~ 300 Mone~ <NA> Indi~ <NA> <NA> <NA> Ford~ Mayor
4 A'Court, K ~ <NA> M4M ~ 36 Mone~ <NA> Indi~ <NA> <NA> <NA> Chow~ Mayor
5 A'Court, K ~ <NA> M4M ~ 100 Mone~ <NA> Indi~ <NA> <NA> <NA> Chow~ Mayor
6 A'Court, K ~ <NA> M4M ~ 100 Mone~ <NA> Indi~ <NA> <NA> <NA> Chow~ Mayor
# ... with 1 more variable: ...13 <chr>, and abbreviated variable name
# 1: `2014 Municipal Election - List of Contributors to Mayoralty Candidates`
```

```
campaign2_2014 <-campaign1_2014[2]`$`2_Mayor_Contributions_2014_election.xls`

colnames(campaign2_2014) <- campaign2_2014[1,]
campaign2_2014 <- campaign2_2014[-1, ]

# make the column names nicer to work with
names(campaign2_2014)<-janitor::make_clean_names(names(campaign2_2014))
campaign_2014<-campaign2_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 13 variables in the dataset containing contributors__name,contributors__address,contributors__postal_code,authorized__representative,candidate,office,ward.Number of records was 10199.

There are missing values but we did not worry about them because their name, postal code,contribution candidate are not missing.Thus we could find their address from other database. Other missing variables did not matter due to not our interesting events.

There is not one variable in the format it should be: `president_business_manager` should be switched into `president_or_business_manager`.

```
library(skimr)
skim(campaign_2014)
```

Table 1: Data summary

Name	campaign_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

```
colnames(campaign2_2014)[9] <- 'president_or_business_manager'
campaign_2014<- campaign2_2014
campaign_2014|>
  summarize(across(everything(), ~ sum(is.na(.x))))
```

A tibble: 1 x 13

```

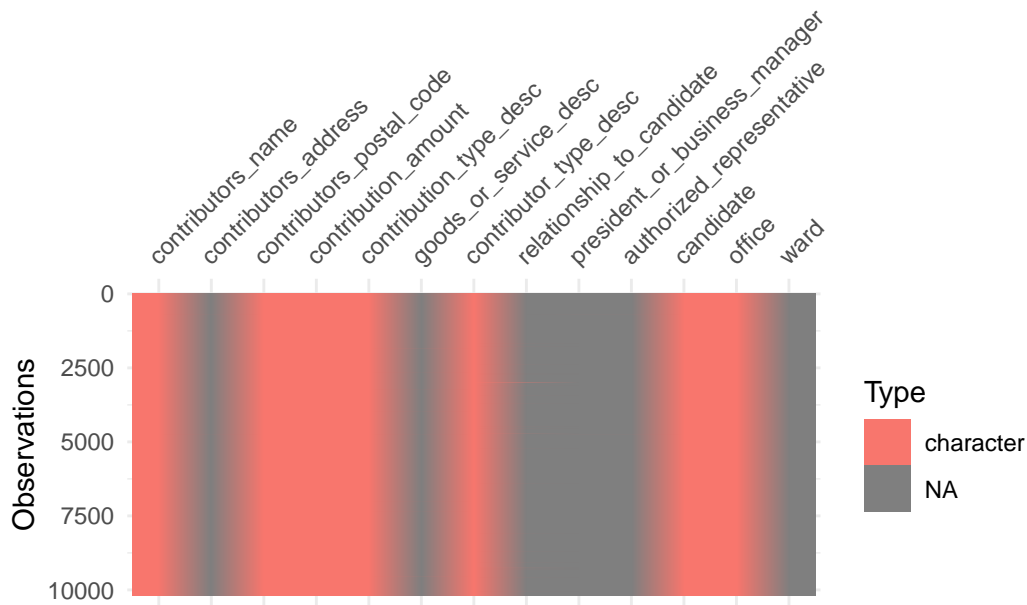
  contributors~1 contr~2 contr~3 contr~4 contr~5 goods~6 contr~7 relat~8 presi~9
            <int>   <int>   <int>   <int>   <int>   <int>   <int>   <int>   <int>
1             0   10197         0         0         0   10188         0   10166   10197
# ... with 4 more variables: authorized_representative <int>, candidate <int>,
#   office <int>, ward <int>, 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_or_business_manager

```

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.

```
vis_dat(campaign_2014)
```

Warning: `gather()` was deprecated in tidyr 1.2.0.
 i Please use `gather()` instead.
 i The deprecated feature was likely used in the visdat package.
 Please report the issue at <<https://github.com/ropensci/visdat/issues>>.



```
class(campaign_2014[4])
```

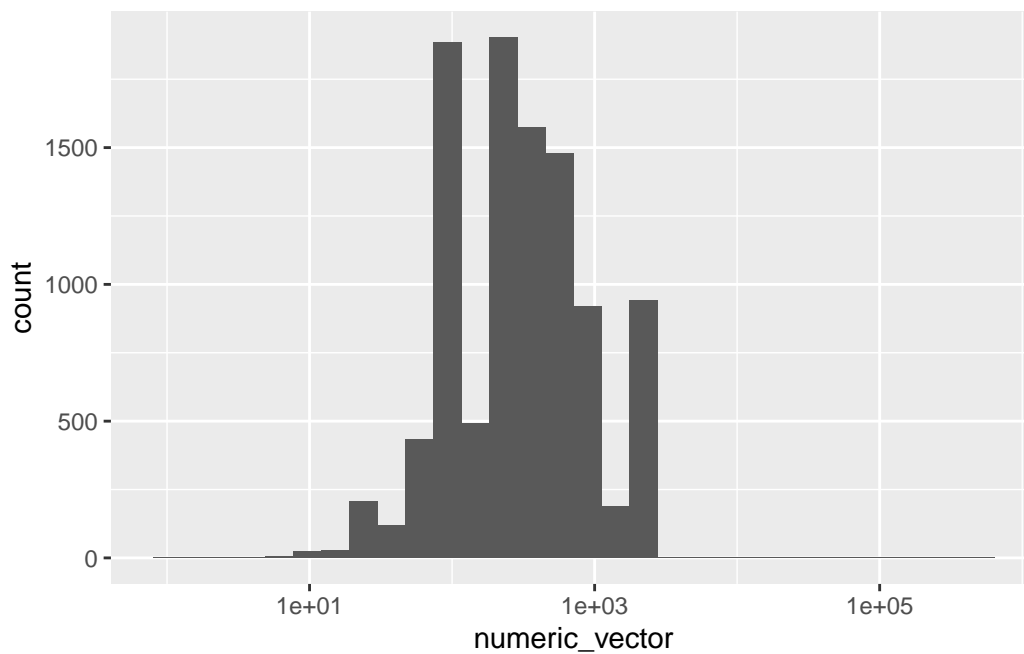
```
[1] "tbl_df"      "tbl"        "data.frame"
```

```
numeric_vector <- as.numeric(unlist(campaign_2014[4]))  
summary(numeric_vector)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	100	300	608	500	508225

```
#min=1.00      100      300      608      500  max=508224.73  
ggplot(data = campaign_2014) +  
  geom_histogram(aes(x = numeric_vector))+scale_x_log10()
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



1.00 CAD and 508224.73 CAD are outliers.

```
campaign2_2014$contribution_amount<-round(numeric_vector,2)
campaign_2014 <- campaign2_2014
campaign_2014 |> filter(campaign_2014$contribution_amount==1.00|campaign_2014$contribution
```

```
# A tibble: 2 x 13
  contributors~1 contr~2 contr~3 contr~4 contr~5 goods~6 contr~7 relat~8 presi~9
  <chr>          <chr>   <chr>    <dbl> <chr>   <chr>   <chr>   <chr>   <chr>
1 Ford, Doug    <NA>    M9A 2C3 508225. Moneta~ <NA>    Indivi~ Candid~ <NA>
2 Italiano, Rob <NA>    M3A 1W1      1 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_or_business_manager
```

They did not share a similar characteristic(s).

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

- total contributions
- mean contribution
- number of contributions

```
library(janitor)
campaign_2014<-campaign2_2014
class(campaign_2014$contribution_amount)
```

```
[1] "numeric"
```

```
campaign_2014 |> group_by(candidate) |> summarize(sum_cont=sum(contribution_amount)) |> ar
```

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



```
campaign_2014 |> group_by(candidate) |> summarize(mean_cont=mean(contribution_amount)) |>
```

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

```
campaign_2014 |> group_by(candidate) |> summarize(num_contribution=length(contributors_name)) |>
```

```
# 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
```

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

```
library(janitor)
campaign_2014<-campaign2_2014
campaign_2014 |> group_by(candidate)|>filter(contributors_name!=candidate) |>summarize(sum_cont=
```

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

```
campaign_2014 |> group_by(candidate)|>filter(contributors_name!=candidate) |> summarize(me
```

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

```
campaign_2014 |> group_by(candidate)|>filter(contributors_name!=candidate)|> summarize(num
```

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

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

```
library(janitor)
library(dplyr)
campaign_2014<-campaign2_2014

camp2<- campaign_2014 |> group_by(contributors_name)|>select(contributors_name,candidate)
  filter(num_candidates>1)
nrow(camp2)
```

```
[1] 184
```

```
length(camp2$contributors_name)
```

```
[1] 184
```

```
#camp1<-campaign_2014 |> group_by(contributors_name)|>select(contributors_name,candidate)
#camp1
```

```
#sum(table(camp1$contributors_name)-1)
```

184 contributors gave money to more than one candidate.