Tutorial 2 - Reproducible Workflows

Jimmy Luc

01/11/2024

```r
#### Workspace setup ####
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```r
install.packages("janitor", repos = "http://cran.us.r-project.org")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```r
install.packages("opendatatoronto", repos = "http://cran.us.r-project.org")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```r
library("janitor")
```

```

Attaching package: 'janitor'

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

    chisq.test, fisher.test
```

```r
library("knitr")
library("lubridate")
```

```

Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

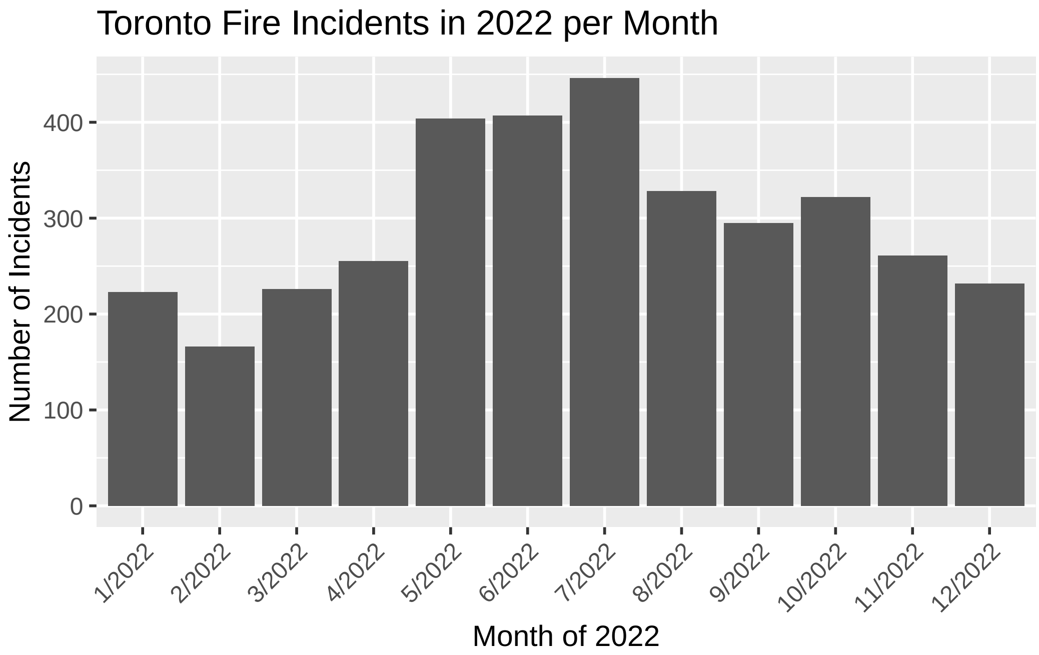
```r
library("opendatatoronto")
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr   1.1.4     v readr   2.1.4
v forcats 1.0.0     v stringr 1.5.1
v ggplot2 3.4.4     v tibble  3.2.1
v purrr   1.0.2     v tidyr   1.3.0

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
#### Read in the data ####
# retrieve package id
dataset <-
  search_packages("fire incidents") |> first()

# retrieve resource id
dataset <-
  list_package_resources(dataset$id) |> first()

# retrieve dataset
raw_fire_incidents_data <- get_resource(dataset$id)

write_csv(
  x = raw_fire_incidents_data,
  file = "toronto_fire_incidents.csv"
)
```

```
#### Basic cleaning ####
raw_fire_incidents_data <-
  read_csv(
    file = "toronto_fire_incidents.csv",
    show_col_types = FALSE
  )

# clean the names
cleaned_fire_incidents_data <-
  clean_names(raw_fire_incidents_data)

# select the column of interest
cleaned_fire_incidents_data <-
  cleaned_fire_incidents_data |>
  select(
    tfs_alarm_time
  )

# filter for data in 2022
cleaned_fire_incidents_data <-
  cleaned_fire_incidents_data[format(cleaned_fire_incidents_data$tfs_alarm_time, "%Y") ==

# ensure dates are in descending order
cleaned_fire_incidents_data <-
  cleaned_fire_incidents_data[order(as.Date(cleaned_fire_incidents_data$tfs_alarm_time, fo

# format date
cleaned_fire_incidents_data$tfs_alarm_time <-
  as.factor(
    paste(
      cleaned_fire_incidents_data$tfs_alarm_time |> month(),
      cleaned_fire_incidents_data$tfs_alarm_time |> year(),
      sep="/"
      )
    )

# save cleaned dataset
write_csv(
  x = cleaned_fire_incidents_data,
  file = "cleaned_fire_incidents_data.csv"
)
```

```
#### Read in the data ####
cleaned_fire_incidents_data <-
  read_csv(
    file = "cleaned_fire_incidents_data.csv",
    show_col_types = FALSE
  )

# generate graph
# Graph depicts the total number of fire incidents each month in Toronto, 2022
cleaned_fire_incidents_data |>
  ggplot(aes(factor(tfs_alarm_time, levels = unique(tfs_alarm_time)))) +
  geom_bar() +
  labs(x = "Month of 2022", y = "Number of Incidents", title = "Toronto Fire Incidents in
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Toronto Fire Incidents in 2022 per Month
Number of Incidents
1/2022
2/2022
3/2022
4/2022
5/2022
6/2022
7/2022
8/2022
9/2022
10/2022
11/2022
12/2022
Month of 2022