tut_week_2_pdf

Boxuan Yi

Preamble

Purpose: Read in data about Daily Shelter & Overnight Service Occupancy in Toronto 2023 to make a graph of the average number of shelters used every month. Author: Boxuan Yi Email: boxuan.yi@mail.utoronto.ca Date: 15 January 2024 Prerequisites: Know where to get data about the use of shelters in Toronto

```
library(knitr)
library(janitor)
```

```
Attaching package: 'janitor'

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

    chisq.test, fisher.test
```

```
library(lubridate)
```

```
Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

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

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr   1.1.4     v readr   2.1.5
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 becor
```

```
library(dplyr)
```

Acquire the dataset from opendatatoronto, and only use the 2023 dataset. Save it as "daily_shelters.csv"

```
daily_shelters <-
  list_package_resources("21c83b32-d5a8-4106-a54f-010dbe49f6f2") |>
  filter(name ==
    "daily-shelter-overnight-service-occupancy-capacity-2023.csv") |>
  get_resource()

write_csv(
  x = daily_shelters,
  file = "daily_shelters.csv"
)

head(daily_shelters)
```

```
# A tibble: 6 x 32
   X_id OCCUPANCY_DATE      ORGANIZATION_ID ORGANIZATION_NAME         SHELTER_ID
  <int> <chr>                         <int> <chr>                          <int>
1     1 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
2     2 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
3     3 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
4     4 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
5     5 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
6     6 2023-01-01T00:00:00              14 Christie Ossington Neigh~         22
# i 27 more variables: SHELTER_GROUP <chr>, LOCATION_ID <int>,
#   LOCATION_NAME <chr>, LOCATION_ADDRESS <chr>, LOCATION_POSTAL_CODE <chr>,
```

```
#   LOCATION_CITY <chr>, LOCATION_PROVINCE <chr>, PROGRAM_ID <int>,
#   PROGRAM_NAME <chr>, SECTOR <chr>, PROGRAM_MODEL <chr>,
#   OVERNIGHT_SERVICE_TYPE <chr>, PROGRAM_AREA <chr>, SERVICE_USER_COUNT <int>,
#   CAPACITY_TYPE <chr>, CAPACITY_ACTUAL_BED <int>, CAPACITY_FUNDING_BED <int>,
#   OCCUPIED_BEDS <int>, UNOCCUPIED_BEDS <int>, UNAVAILABLE_BEDS <int>, ...
```

Clean the dataset and save it as "cleaned_toronto_shelters.csv"

```
toronto_shelters_clean <-
  clean_names(daily_shelters)

write_csv(
  x = toronto_shelters_clean,
  file = "cleaned_toronto_shelters.csv"
)
head(toronto_shelters_clean)
```

```
# A tibble: 6 x 32
   x_id occupancy_date      organization_id organization_name         shelter_id
  <int> <chr>                         <int> <chr>                          <int>
1     1 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
2     2 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
3     3 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
4     4 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
5     5 2023-01-01T00:00:00              24 COSTI Immigrant Services          40
6     6 2023-01-01T00:00:00              14 Christie Ossington Neigh~         22
# i 27 more variables: shelter_group <chr>, location_id <int>,
#   location_name <chr>, location_address <chr>, location_postal_code <chr>,
#   location_city <chr>, location_province <chr>, program_id <int>,
#   program_name <chr>, sector <chr>, program_model <chr>,
#   overnight_service_type <chr>, program_area <chr>, service_user_count <int>,
#   capacity_type <chr>, capacity_actual_bed <int>, capacity_funding_bed <int>,
#   occupied_beds <int>, unoccupied_beds <int>, unavailable_beds <int>, ...
```

Read the file

```
toronto_shelters_clean <-
  read_csv(
    "cleaned_toronto_shelters.csv",
    show_col_types = FALSE
  )
```

Create a new column named occupancy_month based on the occupancy_date column. I used the full name of the month and its abbreviated name.

```
toronto_shelters_clean <- toronto_shelters_clean |>
  mutate(occupancy_month = month(
    occupancy_date,
    label = TRUE,
    abbr = TRUE
  ))
head(toronto_shelters_clean)
```

```
# A tibble: 6 x 33
   x_id occupancy_date      organization_id organization_name         shelter_id
  <dbl> <dttm>                        <dbl> <chr>                          <dbl>
1     1 2023-01-01 00:00:00              24 COSTI Immigrant Services          40
2     2 2023-01-01 00:00:00              24 COSTI Immigrant Services          40
3     3 2023-01-01 00:00:00              24 COSTI Immigrant Services          40
4     4 2023-01-01 00:00:00              24 COSTI Immigrant Services          40
5     5 2023-01-01 00:00:00              24 COSTI Immigrant Services          40
6     6 2023-01-01 00:00:00              14 Christie Ossington Neigh~         22
# i 28 more variables: shelter_group <chr>, location_id <dbl>,
#   location_name <chr>, location_address <chr>, location_postal_code <chr>,
#   location_city <chr>, location_province <chr>, program_id <dbl>,
#   program_name <chr>, sector <chr>, program_model <chr>,
#   overnight_service_type <chr>, program_area <chr>, service_user_count <dbl>,
#   capacity_type <chr>, capacity_actual_bed <dbl>, capacity_funding_bed <dbl>,
#   occupied_beds <dbl>, unoccupied_beds <dbl>, unavailable_beds <dbl>, ...
```

1. Unique() gives a vector containing the unique values in the occupancy_month column. Use this to test if the months are correct.
2. To see if all the corresponding dates start with "2023"

```
toronto_shelters_clean$occupancy_month |> unique()
```

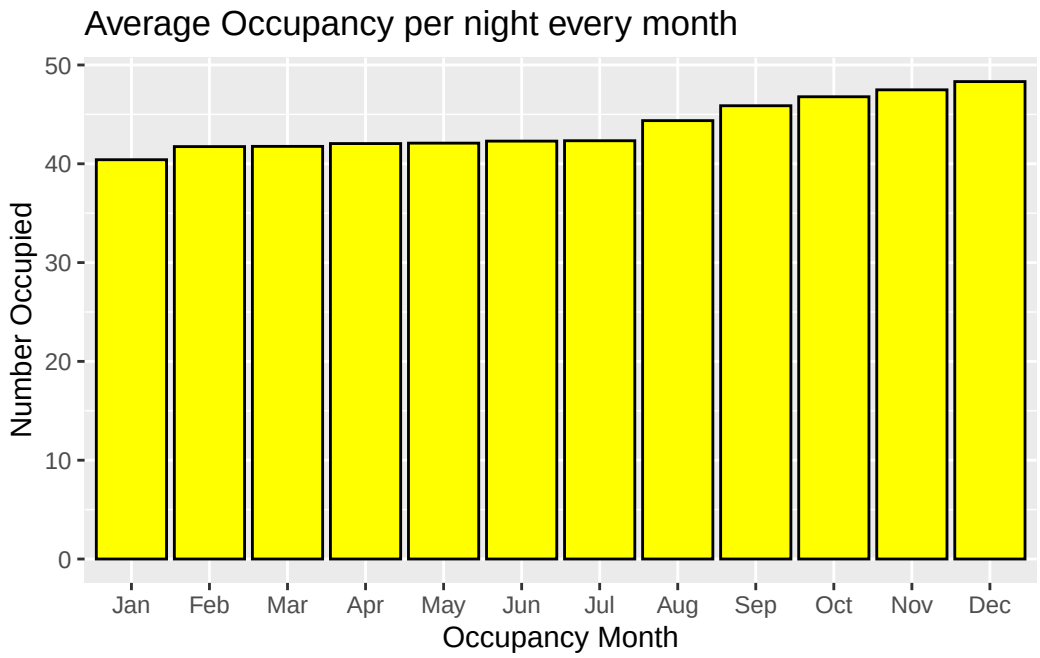
```
 [1] Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < ... < Dec
```

```
all(substr(toronto_shelters_clean$occupancy_date, 1, 4) == "2023")
```

```
[1] TRUE
```

Only keep the relevant data. Create a new data frame called number_occupied classified by month, which is the mean of occupied_beds every month. For visualization, use ggplot to draw a 12-column bar plot with x-axis representing month and y-axis representing the average occupancy per night.

```
toronto_shelters_clean |>
  arrange(month(occupancy_date)) |>
  drop_na(occupied_beds) |>
  summarise(number_occupied = mean(occupied_beds),
            .by = occupancy_month) |>
  ggplot(aes(x = occupancy_month, y = number_occupied)) +
  geom_bar(stat = "identity", fill = "yellow", color = "black") +
  labs(title = "Average Occupancy per night every month",
       x = "Occupancy Month",
       y = "Number Occupied")
```



Again, only keep the data with useful information. Create a new data frame called number_occupied_sum classified by month, which is the total number of occupied_beds every month. Use ggplot to draw a 12-column bar plot for visualization. X represents month and Y represents the total number.

```r
toronto_shelters_clean |>
  arrange(month(occupancy_date)) |>
  drop_na(occupied_beds) |>
  summarise(number_occupied_sum = sum(occupied_beds),
            .by = occupancy_month) |>
  ggplot(aes(x = occupancy_month, y = number_occupied_sum)) +
  geom_bar(stat = "identity", fill = "orange", color = "black") +
  labs(title = "Occupancy night every month",
       x = "Occupancy Month",
       y = "Total Number Occupied")
```