p8105_hw2_mc5698.Rmd

2024-09-27

#Question 1

```
#loading necessary packages
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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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 become errors
```

```
library(readxl)
```

```
#clean the dataset
nyc_t =
  read_csv(
    "/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/NYC_Transit_Subway_Entrance_And_Exit_Data.csv
    col_types = cols(Route8 = "c", Route9 = "c", Route10 = "c", Route11 = "c")) |>
  janitor::clean_names() |>
  select(
    line, station_name, station_latitude, station_longitude, route1, route2, route3, route4, route5, rou
  mutate(
    entry = ifelse(entry == "YES", TRUE, FALSE))
```

The dataset contains line, station_name, station_latitude, station_longitude, route1, route2, route3, route4, route5, route6, route7, route8, route9, route10, route11, entry, vending, entrance_type, ada. For the data cleaning, I removed unnecessary columns and convert the entry variable from character to a logical variable by using `case_match` function. The dimension of the resulting dataset is 1868, 19. These data are mostly tidy but we could pivot different route columns into one variable.

```
distinct_stations=
  nyc_t|>
  distinct(station_name,line)
nrow(distinct_stations)
```

```
## [1] 465
```

```
ada_stations=
  nyc_t|>
  filter(ada==TRUE)|>
  distinct(station_name, line)

nrow(ada_stations)
```

```
## [1] 84
```

```
no_vending=
  nyc_t |>
  filter(vending == "NO") |>
  pull(entry)

proportion_entry= mean(no_vending)
proportion_entry
```

```
## [1] 0.3770492
```

There are 465 distinct stations. 84 stations are ADA compliant.The proportion of station entrances/exits without vending allow entrance is 0.3770492.

```
transfrom_ent=
  nyc_t |>
  pivot_longer(
    route1:route11,
    names_to = "route_num",
    values_to = "route")

A_stations=
  transfrom_ent|>
  filter(route == "A") |>
  distinct(station_name, line)


nyc_t |>
  pivot_longer(
    route1:route11,
    names_to = "route_num",
    values_to = "route") |>
  filter(route == "A", ada == TRUE) |>
  distinct(station_name, line)
```

```
## # A tibble: 17 x 2
##    station_name                  line
##    <chr>                         <chr>
##  1 14th St                       8 Avenue
##  2 168th St - Washington Heights 8 Avenue
##  3 175th St                      8 Avenue
##  4 34th St                       8 Avenue
##  5 42nd St                       8 Avenue
##  6 59th St                       8 Avenue
```

```
##  7 Inwood - 207th St             8 Avenue
##  8 West 4th St                   8 Avenue
##  9 World Trade Center            8 Avenue
## 10 Times Square-42nd St          Broadway
## 11 59th St-Columbus Circle       Broadway-7th Ave
## 12 Times Square                  Broadway-7th Ave
## 13 8th Av                        Canarsie
## 14 Franklin Av                   Franklin
## 15 Euclid Av                     Fulton
## 16 Franklin Av                   Fulton
## 17 Howard Beach                  Rockaway
```

There are 60 distinct stations serve the A train and 17 stations serve the A train and ADA compliant.

#Question 2

```
#clean the datasets
mr_trash_wheel =
  readxl::read_excel("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/202409TrashWheelCollectionDa
  filter(!is.na(Dumpster)) |>
  mutate(Sports_Balls = as.integer(round(`Sports Balls`)),
         Year = as.character(Year),
         Trash_Wheel = "Mr. Trash Wheel")

professor_trash_wheel =
  readxl::read_excel("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/202409TrashWheelCollectionDa
  filter(!is.na(Dumpster)) |>
  mutate(Year = as.character(Year),
         Trash_Wheel = "Professor Trash Wheel")

gwynnda_trash_wheel =
  readxl::read_excel("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/202409TrashWheelCollectionDa
  filter(!is.na(Dumpster)) |>
  mutate(Year = as.character(Year),
         Trash_Wheel = "Gwynnda Trash Wheel")

combined_data =
  bind_rows(mr_trash_wheel, professor_trash_wheel, gwynnda_trash_wheel)
combined_data
```

```
## # A tibble: 1,033 x 16
##    Dumpster Month Year  Date                `Weight (tons)`
##       <dbl> <chr> <chr> <dttm>                        <dbl>
##  1        1 May   2014  2014-05-16 00:00:00            4.31
##  2        2 May   2014  2014-05-16 00:00:00            2.74
##  3        3 May   2014  2014-05-16 00:00:00            3.45
##  4        4 May   2014  2014-05-17 00:00:00            3.1
##  5        5 May   2014  2014-05-17 00:00:00            4.06
##  6        6 May   2014  2014-05-20 00:00:00            2.71
##  7        7 May   2014  2014-05-21 00:00:00            1.91
##  8        8 May   2014  2014-05-28 00:00:00            3.7
##  9        9 June  2014  2014-06-05 00:00:00            2.52
## 10       10 June  2014  2014-06-11 00:00:00            3.76
## # i 1,023 more rows
```

```
## # i 11 more variables: `Volume (cubic yards)` <dbl>, `Plastic Bottles` <dbl>,
## #   Polystyrene <dbl>, `Cigarette Butts` <dbl>, `Glass Bottles` <dbl>,
## #   `Plastic Bags` <dbl>, Wrappers <dbl>, `Sports Balls` <dbl>,
## #   `Homes Powered*` <dbl>, Sports_Balls <int>, Trash_Wheel <chr>
```

By reading and cleaning the datasets, I combined the three datasets from `Mr. Trash Wheel`, `Professor Trash Wheel` and `Gwynnda Trash Wheel`. There are 1033 observations in the combined dataset. This dataset includes key variables such as`Dumpster`, which shows the the number of dumpster filled by trash, and `Cigarette Butts` which means the number of cigarette they collected. It also includes the specific time of the trash such as `Year`, `Date`, `Month` and `Trash_Wheel` indicates different trash types correspond to the different trash wheel. Moreover, it provides the detailed volumn and types for each trash wheel.

```
tw_professor =
  combined_data |>
  filter(Trash_Wheel == "Professor Trash Wheel") |>
  summarise(total_weight = sum(`Weight (tons)`, na.rm = TRUE))
```

```
cb_gwynnda_june2022 =
  combined_data |>
  filter(Trash_Wheel == "Gwynnda Trash Wheel", Year == "2022", Month == "June") |>
  summarise(total_cig_butts = sum(`Cigarette Butts`, na.rm = TRUE))
```

The total weight of trash collected by Professor Trash Wheel wad 246.74. The total number of cigarette butts collected by Gwynnda in June of 2022 was 1.812×10^4.

#Question 3

```
#read and clean the datasets
bakers_data =
  read_csv("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/gbb_datasets/bakers.csv") |>
  janitor::clean_names() %>%
  mutate(source = "bakers")


bakes_data =
  read_csv("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/gbb_datasets/bakes.csv") |>
  janitor::clean_names() %>%
  mutate(source = "bakes")

results_data =
  read_csv("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/gbb_datasets/results.csv", skip = 2) |
  janitor::clean_names()

viewers_data =
  read_csv("/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/gbb_datasets/viewers.csv") |>
  janitor::clean_names()
```

```
#rename the column names
bakers_data <- bakers_data %>%
  rename(baker = baker_name)

# Extract the first word
bakers_data$baker <- sapply(strsplit(bakers_data$baker, " "), function(x) x[1])
```

```r
#check for completeness and correctness across datasets
missing_bakers <- anti_join(bakes_data, bakers_data, by = c("baker"))

missing_bakes <- anti_join(results_data, bakes_data, by = c("series", "episode", "baker"))
```

```r
# merge the datasets
final_dataset =
  merge(bakers_data, bakes_data, all= TRUE)

final_dataset =
  merge(final_dataset,results_data, all= TRUE)
```

```r
write_csv(final_dataset, "/Users/nicolechen/Downloads/p8105_hw2_mc5698/dataset/gbb_datasets/final_datas
```

For this project, I cleaned and organized data from `bakers.csv`, `bakes.csv`, `results.csv`, and `viewers.csv`. Firstly, I renamed the `Baker Name` column to `Baker` in the `bakers.csv` file to make it easier to match with other datasets. Then, I noticed that the baker names had different formats, so I used a function to convert all the names to lowercase and remove extra spaces.I also transformed the `Series` and `Episode` · columns to numeric for easier merging. For `viewers.csv`, I reshaped the data from wide to long format to align with the other files. After checking for missing bakers between the files, I merged the datasets step by step: first, results with `bakers`, then with `bakes`, and finally with `viewers`. I organized the data by series and episode to make it more readable. The final dataset contains all relevant information, including bakers' details, results, and viewership.

```r
#Create a reader-friendly table showing the star baker or winner of each episode in Seasons 5 through 1
star_baker =
  final_dataset %>%
  filter(series >= 5 & series <= 10, result %in% c("STAR BAKER", "WINNER")) %>%
  select(series, episode , baker, result) %>%
  arrange(series, episode)
star_baker
```

```
##    series episode     baker     result
## 1       5       1     Nancy STAR BAKER
## 2       5       2   Richard STAR BAKER
## 3       5       3      Luis STAR BAKER
## 4       5       4   Richard STAR BAKER
## 5       5       5      Kate STAR BAKER
## 6       5       6    Chetna STAR BAKER
## 7       5       7   Richard STAR BAKER
## 8       5       8   Richard STAR BAKER
## 9       5       9   Richard STAR BAKER
## 10      5      10     Nancy     WINNER
## 11      6       1     Marie STAR BAKER
## 12      6       2       Ian STAR BAKER
## 13      6       3       Ian STAR BAKER
## 14      6       4       Ian STAR BAKER
## 15      6       5    Nadiya STAR BAKER
## 16      6       6       Mat STAR BAKER
## 17      6       7     Tamal STAR BAKER
## 18      6       8    Nadiya STAR BAKER
## 19      6       9    Nadiya STAR BAKER
```

```
## 20      6      10    Nadiya     WINNER
## 21      7       1      Jane STAR BAKER
## 22      7       2   Candice STAR BAKER
## 23      7       3       Tom STAR BAKER
## 24      7       4 Benjamina STAR BAKER
## 25      7       5   Candice STAR BAKER
## 26      7       6       Tom STAR BAKER
## 27      7       7    Andrew STAR BAKER
## 28      7       8   Candice STAR BAKER
## 29      7       9    Andrew STAR BAKER
## 30      7      10   Candice     WINNER
## 31      8       1    Steven STAR BAKER
## 32      8       2    Steven STAR BAKER
## 33      8       3     Julia STAR BAKER
## 34      8       4      Kate STAR BAKER
## 35      8       5    Sophie STAR BAKER
## 36      8       6      Liam STAR BAKER
## 37      8       7    Steven STAR BAKER
## 38      8       8    Stacey STAR BAKER
## 39      8       9    Sophie STAR BAKER
## 40      8      10    Sophie     WINNER
## 41      9       1     Manon STAR BAKER
## 42      9       2     Rahul STAR BAKER
## 43      9       3     Rahul STAR BAKER
## 44      9       4       Dan STAR BAKER
## 45      9       5   Kim-Joy STAR BAKER
## 46      9       6    Briony STAR BAKER
## 47      9       7   Kim-Joy STAR BAKER
## 48      9       8      Ruby STAR BAKER
## 49      9       9      Ruby STAR BAKER
## 50      9      10     Rahul     WINNER
## 51     10       1  Michelle STAR BAKER
## 52     10       2     Alice STAR BAKER
## 53     10       3   Michael STAR BAKER
## 54     10       4     Steph STAR BAKER
## 55     10       5     Steph STAR BAKER
## 56     10       6     Steph STAR BAKER
## 57     10       7     Henry STAR BAKER
## 58     10       8     Steph STAR BAKER
## 59     10       9     Alice STAR BAKER
## 60     10      10     David     WINNER
```

From the table, I found that some people such as Richard become star bakers or winners in multiple episodes, which might make their overall success predictable.

```
#import, clean, tidy, and organize the viewership data
head(viewers_data, 10)
```

```
## # A tibble: 10 x 11
##    episode series_1 series_2 series_3 series_4 series_5 series_6 series_7
##      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1       1     2.24     3.1      3.85     6.6      8.51     11.6     13.6
##  2       2     3        3.53     4.6      6.65     8.79     11.6     13.4
```

```
##  3       3     3        3.82     4.53     7.17     9.28     12.0     13.0
##  4       4     2.6      3.6      4.71     6.82    10.2      12.4     13.3
##  5       5     3.03     3.83     4.61     6.95     9.95     12.4     13.1
##  6       6     2.75     4.25     4.82     7.32    10.1      12       13.1
##  7       7    NA        4.42     5.1      7.76    10.3      12.4     13.4
##  8       8    NA        5.06     5.35     7.41     9.02     11.1     13.3
##  9       9    NA       NA        5.7      7.41    10.7      12.6     13.4
## 10      10    NA       NA        6.74     9.45    13.5      15.0     15.9
## # i 3 more variables: series_8 <dbl>, series_9 <dbl>, series_10 <dbl>
```

```
season_1 = mean(viewers_data$series_1, na.rm = TRUE)

season_5 = mean(viewers_data$series_5, na.rm = TRUE)
```

The average viewership in Season 1 is 2.77, and the average viewership in Season 5 is approximately 10.0393, showing the growth in the show's popularity.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this: