Untitled

Tai Yue

2024-09-27

problem1:

```
data <- read.csv('NYC_Transit_Subway_Entrance_And_Exit_Data.csv')

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
cleaned_data <- data %>%
  select(Line, `Station.Name`, `Station.Latitude`, `Station.Longitude`,
         Route1, Route2, Route3, Route4, Entry, Vending, `Entrance.Type`, ADA) %>%
  mutate(Entry = ifelse(Entry == "YES", TRUE, FALSE))

dim(cleaned_data)
```

```
## [1] 1868   12
```

the data are generally tidy. In tidy data, each variable forms a column, each observation forms a row, and each type of observational unit forms a table.However, there are four separate columns (Route1, Route2, Route3, and Route4) representing different subway routes served by each station.

```
library(dplyr)
library(tidyr)

cleaned_data <- data %>%
  select(Line, `Station.Name`, `Station.Latitude`, `Station.Longitude`,
         Route1, Route2, Route3, Route4, Entry, Vending, `Entrance.Type`, ADA) %>%
  mutate(Entry = ifelse(Entry == "YES", TRUE, FALSE))
```

```r
distinct_stations <- cleaned_data %>%
  distinct(`Station.Name`, Line) %>%
  nrow()


ada_compliant_stations <- cleaned_data %>%
  filter(ADA == TRUE) %>%
  distinct(`Station.Name`, Line) %>%
  nrow()


proportion_without_vending_allows_entry <- cleaned_data %>%
  filter(Vending == "NO") %>%
  summarise(proportion = mean(Entry)) %>%
  pull(proportion)


routes_melted <- cleaned_data %>%
  pivot_longer(cols = starts_with("Route"), names_to = "Route Number", values_to = "Route Name") %>%
  filter(!is.na(`Route Name`))

stations_serving_A_train <- routes_melted %>%
  filter(`Route Name` == "A") %>%
  distinct(`Station.Name`, Line) %>%
  nrow()

ada_compliant_A_train_stations <- routes_melted %>%
  filter(`Route Name` == "A", ADA == TRUE) %>%
  distinct(`Station.Name`, Line) %>%
  nrow()

list(
  distinct_stations = distinct_stations,
  ada_compliant_stations = ada_compliant_stations,
  proportion_without_vending_allows_entry = proportion_without_vending_allows_entry,
  stations_serving_A_train = stations_serving_A_train,
  ada_compliant_A_train_stations = ada_compliant_A_train_stations
)
```

```
## $distinct_stations
## [1] 465
##
## $ada_compliant_stations
## [1] 84
##
## $proportion_without_vending_allows_entry
## [1] 0.3770492
##
## $stations_serving_A_train
## [1] 60
##
## $ada_compliant_A_train_stations
## [1] 17
```

problem2:

```
library(readxl)
library(dplyr)


trash_wheel_data <- read_excel("202409 Trash Wheel Collection Data.xlsx",
                               sheet = "Mr. Trash Wheel", skip = 1)
```

```
## New names:
## * `` -> `...15`
## * `` -> `...16`
```

```
names(trash_wheel_data) <- c("Dumpster", "Month", "Year", "Date", "Weight_Tons",
                             "Volume_Cubic_Yards", "Plastic_Bottles", "Polystyrene",
                             "Cigarette_Butts", "Glass_Bottles", "Plastic_Bags",
                             "Wrappers", "Sports_Balls", "Homes_Powered", "Extra1", "Extra2")


trash_wheel_data <- trash_wheel_data %>%
  select(-Extra1, -Extra2)


trash_wheel_data <- trash_wheel_data %>%
  filter(!is.na(Dumpster))


trash_wheel_data <- trash_wheel_data %>%
  mutate(Sports_Balls = as.integer(round(Sports_Balls, 0)))


head(trash_wheel_data)
```

```
## # A tibble: 6 x 14
##   Dumpster Month Year  Date                Weight_Tons Volume_Cubic_Yards
##      <dbl> <chr> <chr> <dttm>                    <dbl>              <dbl>
## 1        1 May   2014  2014-05-16 00:00:00        4.31                 18
## 2        2 May   2014  2014-05-16 00:00:00        2.74                 13
## 3        3 May   2014  2014-05-16 00:00:00        3.45                 15
## 4        4 May   2014  2014-05-17 00:00:00        3.1                  15
## 5        5 May   2014  2014-05-17 00:00:00        4.06                 18
## 6        6 May   2014  2014-05-20 00:00:00        2.71                 13
## # i 8 more variables: Plastic_Bottles <dbl>, Polystyrene <dbl>,
## #   Cigarette_Butts <dbl>, Glass_Bottles <dbl>, Plastic_Bags <dbl>,
## #   Wrappers <dbl>, Sports_Balls <int>, Homes_Powered <dbl>
```

```
library(readxl)
library(dplyr)


clean_trash_wheel_data <- function(file_path, sheet_name, wheel_name) {
  data <- read_excel(file_path, sheet = sheet_name, skip = 1)
```

```
  col_names <- c("Dumpster", "Month", "Year", "Date", "Weight_Tons",
                 "Volume_Cubic_Yards", "Plastic_Bottles", "Polystyrene",
                 "Cigarette_Butts", "Glass_Bottles", "Plastic_Bags",
                 "Wrappers", "Sports_Balls", "Homes_Powered", "Extra1", "Extra2")


  names(data) <- col_names[1:ncol(data)]


  data <- data %>%
    select(-starts_with("Extra"))


  data <- data %>%
    filter(!is.na(Dumpster))

  if ("Sports_Balls" %in% names(data)) {
    data <- data %>%
      mutate(Sports_Balls = as.integer(round(Sports_Balls, 0)))
  }


  data <- data %>%
    mutate(Trash_Wheel = wheel_name)

  return(data)
}


file <- "202409 Trash Wheel Collection Data.xlsx"


mr_trash_wheel <- clean_trash_wheel_data(file, "Mr. Trash Wheel", "Mr. Trash Wheel")
```

```
## New names:
## * `` -> `...15`
## * `` -> `...16`
```

```
professor_trash_wheel <- clean_trash_wheel_data(file, "Professor Trash Wheel", "Professor Trash Wheel")
gwynnda_trash_wheel <- clean_trash_wheel_data(file, "Gwynnda Trash Wheel", "Gwynnda Trash Wheel")


common_cols <- intersect(names(mr_trash_wheel), intersect(names(professor_trash_wheel), names(gwynnda_t


mr_trash_wheel <- mr_trash_wheel %>% select(all_of(common_cols))
professor_trash_wheel <- professor_trash_wheel %>% select(all_of(common_cols))
gwynnda_trash_wheel <- gwynnda_trash_wheel %>% select(all_of(common_cols))


mr_trash_wheel <- mr_trash_wheel %>%
  mutate(Year = as.character(Year),
```

```
         Date = as.Date(Date))

professor_trash_wheel <- professor_trash_wheel %>%
  mutate(Year = as.character(Year),
         Date = as.Date(Date))

gwynnda_trash_wheel <- gwynnda_trash_wheel %>%
  mutate(Year = as.character(Year),
         Date = as.Date(Date))


combined_trash_wheel_data <- bind_rows(mr_trash_wheel, professor_trash_wheel, gwynnda_trash_wheel)


head(combined_trash_wheel_data)
```

```
## # A tibble: 6 x 13
##   Dumpster Month Year  Date       Weight_Tons Volume_Cubic_Yards Plastic_Bottles
##      <dbl> <chr> <chr> <date>           <dbl>              <dbl>           <dbl>
## 1        1 May   2014  2014-05-16        4.31                 18            1450
## 2        2 May   2014  2014-05-16        2.74                 13            1120
## 3        3 May   2014  2014-05-16        3.45                 15            2450
## 4        4 May   2014  2014-05-17        3.1                  15            2380
## 5        5 May   2014  2014-05-17        4.06                 18             980
## 6        6 May   2014  2014-05-20        2.71                 13            1430
## # i 6 more variables: Polystyrene <dbl>, Cigarette_Butts <dbl>,
## #   Glass_Bottles <dbl>, Plastic_Bags <dbl>, Wrappers <dbl>, Trash_Wheel <chr>
```

The combined dataset contains data collected by multiple trash wheels, including Mr. Trash Wheel, Professor Trash Wheel, and Gwynnda Trash Wheel. The dataset includes key variables such as the weight of trash collected, volume, and the number of specific types of waste like plastic bottles, cigarette butts, and polystyrene.

For example, Professor Trash Wheel collected a total of approximately 246.74 tons of trash based on the available data. Additionally, Gwynnda Trash Wheel collected 16,720 cigarette butts in June of 2022. This dataset provides an insightful overview of the types and quantities of waste being removed by each trash wheel.

problem3:

```
library(dplyr)
library(readr)


bakers <- read_csv("bakers.csv")
```

```
## Rows: 120 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Baker Name, Baker Occupation, Hometown
## dbl (2): Series, Baker Age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
bakes <- read_csv("bakes.csv")
```

```
## Rows: 548 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Baker, Signature Bake, Show Stopper
## dbl (2): Series, Episode
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
results <- read_csv("results.csv")
```

```
## New names:
## Rows: 1138 Columns: 5
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (5): ...1, ...2, ...3, ...4, IN = stayed in; OUT = Eliminated; STAR BAKE...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
```

```r
colnames(bakers)
```

```
## [1] "Baker Name"       "Series"           "Baker Age"        "Baker Occupation"
## [5] "Hometown"
```

```r
colnames(bakes)
```

```
## [1] "Series"         "Episode"        "Baker"          "Signature Bake"
## [5] "Show Stopper"
```

```r
colnames(results)
```

```
## [1] "...1"
## [2] "...2"
## [3] "...3"
## [4] "...4"
## [5] "IN = stayed in; OUT = Eliminated; STAR BAKER = Star Baker; WINNER = Series Winner; Runner-up = S
```

```r
bakers <- bakers %>%
  rename(Baker = `Baker Name`)


results_cleaned <- results[-1, ]  # Remove the first row with headers inside the data
colnames(results_cleaned) <- c("Series", "Episode", "Baker", "Technical", "Result")
```

```r
bakers <- bakers %>%
  mutate(Baker = trimws(Baker))

bakes <- bakes %>%
  mutate(Baker = trimws(Baker))

results_cleaned <- results_cleaned %>%
  mutate(Baker = trimws(Baker))

bakers <- bakers %>%
  mutate(Series = as.character(Series))

bakes <- bakes %>%
  mutate(Series = as.character(Series))

results_cleaned <- results_cleaned %>%
  mutate(Series = as.character(Series))


bakes <- bakes %>%
  mutate(Episode = as.character(Episode))

results_cleaned <- results_cleaned %>%
  mutate(Episode = as.character(Episode))


merged_data <- bakes %>%
  left_join(bakers, by = c("Series", "Baker")) %>%
  left_join(results_cleaned, by = c("Series", "Episode", "Baker"))

print(head(merged_data))
```

```
## # A tibble: 6 x 10
##   Series Episode Baker     `Signature Bake`           `Show Stopper` `Baker Age`
##   <chr>  <chr>   <chr>     <chr>                      <chr>                <dbl>
## 1 1      1       Annetha   Light Jamaican Black Cake~ Red, White & ~          NA
## 2 1      1       David     Chocolate Orange Cake      Black Forest ~          NA
## 3 1      1       Edd       Caramel Cinnamon and Bana~ N/A                     NA
## 4 1      1       Jasminder Fresh Mango and Passion F~ N/A                     NA
## 5 1      1       Jonathan  Carrot Cake with Lime and~ Three Tiered ~          NA
## 6 1      1       Lea       Cranberry and Pistachio C~ Raspberries a~          NA
## # i 4 more variables: `Baker Occupation` <chr>, Hometown <chr>,
## #   Technical <chr>, Result <chr>
```

```r
write_csv(merged_data, "final_bake_off_data.csv")


summary(merged_data)
```

```
##     Series             Episode             Baker           Signature Bake
##  Length:548         Length:548         Length:548         Length:548
##  Class :character   Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Show Stopper         Baker Age   Baker Occupation     Hometown
##  Length:548         Min.   : NA   Length:548         Length:548
##  Class :character   1st Qu.: NA   Class :character   Class :character
##  Mode  :character   Median : NA   Mode  :character   Mode  :character
##                     Mean   :NaN
##                     3rd Qu.: NA
##                     Max.   : NA
##                     NA's   :548
##   Technical            Result
##  Length:548         Length:548
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

```
colnames(merged_data)
```

```
##  [1] "Series"           "Episode"          "Baker"            "Signature Bake"
##  [5] "Show Stopper"     "Baker Age"        "Baker Occupation" "Hometown"
##  [9] "Technical"        "Result"
```

The first step was to inspect the column names in all three datasets to ensure consistency across them. In the case of the bakers dataset, I identified that the column containing baker names was labeled as Baker Name, while in the other datasets, it was just Baker. To solve this, I renamed the column in the bakers dataset to Baker to maintain consistency across the merges.The initial error indicated that the Series column had a mismatch in data types between datasets .To fix this, I converted the Series and Episode columns in all datasets to character type. This ensured a smooth join process without data type conflicts. Another potential issue that could affect merging is whitespace in the Baker column. To ensure no trailing or leading spaces affected the merge, I used trimws() on the Baker columns in all datasets to make the entries consistent for the join.After cleaning the datasets, I merged them using left_join() based on the Series, Baker, and Episode columns. This step combines information from the bakes, bakers, and results datasets into a single dataset. After merging the data, I analyzed the final dataset to ensure that the join was successful and no important data was missing. The final dataset displayed relevant details such as Series, Episode, and Baker, as well as other columns.

```
library(dplyr)
library(readr)


results <- read_csv("results.csv", skip = 1)  # Skipping the first row as it contains extra headers
```

```
## New names:
## Rows: 1137 Columns: 5
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
```

```
## (5): ...1, ...2, ...3, ...4, ...5
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
```

```
colnames(results) <- c("Series", "Episode", "Baker", "Technical", "Result")


results <- results %>%
  mutate(Series = as.character(Series))


filtered_results <- results %>%
  filter(Series %in% c("5", "6", "7", "8", "9", "10"), Result %in% c("STAR BAKER", "WINNER"))


filtered_results <- filtered_results %>%
  select(Series, Episode, Baker, Result)


print(filtered_results)
```

```
## # A tibble: 60 x 4
##    Series Episode Baker   Result
##    <chr>  <chr>   <chr>   <chr>
##  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
## # i 50 more rows
```

```
write_csv(filtered_results, "star_bakers_and_winners_season_5_to_10.csv")
```

Richard had an excellent performance during the seasons, earning Star Baker five times. Given this, we could expect him to be the predictable winner. Despite Richard's good performance, Nancy won the competition in the final episode. This is surprising given Richard's consistent performance. Nancy only earned Star Baker once in the first episode but ultimately won the finale. This outcome suggests that the final episode carries significant weight in determining the winner.

```
library(dplyr)
library(readr)


viewers <- read_csv("viewers.csv")
```

```
## Rows: 10 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (11): Episode, Series 1, Series 2, Series 3, Series 4, Series 5, Series ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(viewers)
```

```
## # A tibble: 6 x 11
##   Episode `Series 1` `Series 2` `Series 3` `Series 4` `Series 5` `Series 6`
##     <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1       1       2.24       3.1        3.85       6.6        8.51       11.6
## 2       2       3          3.53       4.6        6.65       8.79       11.6
## 3       3       3          3.82       4.53       7.17       9.28       12.0
## 4       4       2.6        3.6        4.71       6.82      10.2        12.4
## 5       5       3.03       3.83       4.61       6.95       9.95       12.4
## 6       6       2.75       4.25       4.82       7.32      10.1        12
## # i 4 more variables: `Series 7` <dbl>, `Series 8` <dbl>, `Series 9` <dbl>,
## #   `Series 10` <dbl>
```

```
colnames(viewers) <- c("Episode", "Series_1", "Series_2", "Series_3", "Series_4", "Series_5", "Series_6
                       "Series_7", "Series_8", "Series_9", "Series_10")


first_10_rows <- head(viewers, 10)
print(first_10_rows)
```

```
## # 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>
```

```
avg_viewership_season_1 <- mean(viewers$Series_1, na.rm = TRUE)
avg_viewership_season_5 <- mean(viewers$Series_5, na.rm = TRUE)


cat("Average viewership for Season 1: ", avg_viewership_season_1, "million\n")
```

```
## Average viewership for Season 1:  2.77 million
```

```r
cat("Average viewership for Season 5: ", avg_viewership_season_5, "million\n")
```

```
## Average viewership for Season 5:  10.0393 million
```