hw2

Zitao Zhang

2024-10-01

Problem 1

```r
# Read data
nyc_transit_data <- read.csv("hw2_data/NYC_Transit_Subway_Entrance_And_Exit_Data.csv")

# Retain columns
nyc_transit_cleaned <- nyc_transit_data %>%
  select(Line, Station.Name, Station.Latitude, Station.Longitude,
         Route1, Route2, Route3, Route4, Route5, Route6, Route7, Route8,
         Entry, Vending, Entrance.Type, ADA) %>%
  clean_names()

# Convert "entry"
nyc_transit_cleaned <- nyc_transit_cleaned %>%
  mutate(entry = ifelse(entry == "YES", TRUE, FALSE))
```

The cleaned dataset contains 16 columns and 1,868 rows. The variables include:

line: The subway line serving the station
station_name: The name of the station
station_latitude / station_longitude: Geographical coordinates of the station
route1 to route8: The subway routes serving the station
entry: Indicates whether entry is allowed at this entrance (converted to a logical variable: True for "YES," False for "NO")
vending: Describes whether vending machines are present
entrance_type: Type of entrance
ada: Indicates if the station is ADA (Americans with Disabilities Act) compliant

Data Cleaning Steps

1. Selected relevant columns.

2. Renamed columns to follow a consistent lowercase format with underscores.

3. Converted the "entry" variable from "YES"/"NO" to a logical (True/False) variable.

The resulting dataset is tidy since each variable is in its own column, and each observation occupies a single row.

1. How many distinct stations are there?

```r
distinct_stations <- nyc_transit_cleaned %>%
  distinct(line, station_name) %>%
  nrow()
print(distinct_stations)
```

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

There are 465 distinct stations.

2. How many stations are ADA compliant?

```r
ada_compliant_stations <- nyc_transit_cleaned %>%
  filter(ada == TRUE) %>%
  distinct(line, station_name) %>%
  nrow()
print(ada_compliant_stations)
```

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

There are 84 stations that are ADA compliant.

3. What proportion of station entrances/exits without vending allow entry?

```r
proportion_no_vending_entry <- nyc_transit_cleaned %>%
  filter(vending == "NO") %>%
  summarise(proportion = mean(entry, na.rm = TRUE)) %>%
  pull(proportion)
print(proportion_no_vending_entry)
```

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

There is the proportion of 0.3770492 station entrances / exits without vending allow entrance.

```r
# Make all route columns are of the same data type (character)
nyc_transit_cleaned <- nyc_transit_cleaned %>%
  mutate(across(route1:route8, as.character))

# Reformat data
nyc_transit_routes =
  pivot_longer(
    nyc_transit_cleaned,
    route1:route8,
    names_to = "route_number",
    values_to = "route_name")
```

How many distinct stations serve the A train?

```r
stations_serving_A <- nyc_transit_routes %>%
  filter(route_name == "A") %>%
  distinct(line, station_name) %>%
  nrow()
print(stations_serving_A)
```

```
## [1] 60
```

There are 60 distinct stations serve the A train.

Of the stations that serve the A train, how many are ADA compliant?

```r
ada_compliant_stations_A <- nyc_transit_routes %>%
  filter(route_name == "A", ada == TRUE) %>%
  distinct(line, station_name) %>%
  nrow()
print(ada_compliant_stations_A)
```

```
## [1] 17
```

There are 17 that are ADA compliant of the stations that serve the A train.

Problem 2

Mrs Trash Wheel

```r
mr_trash_wheel <- read_excel("hw2_data/202409 Trash Wheel Collection Data.xlsx",
                             sheet = "Mr. Trash Wheel") %>%
  clean_names() %>%
  select(-starts_with("X")) %>%
  filter(!is.na(dumpster)) %>%
  mutate(sports_balls = as.integer(round(as.numeric(sports_balls), 0)),
         trash_wheel = "Mr. Trash Wheel")
```

Professor Trash Wheel

```r
prof_trash_wheel <- read_excel("hw2_data/202409 Trash Wheel Collection Data.xlsx",
                               sheet = "Professor Trash Wheel") %>%
  clean_names() %>%
  filter(!is.na(dumpster)) %>%
  mutate(year = as.character(year),
         trash_wheel = "Professor Trash Wheel")
```

Gwynnda Trash Wheel

```r
gwynnda_trash_wheel <- read_excel("hw2_data/202409 Trash Wheel Collection Data.xlsx",
                                  sheet = "Gwynnda Trash Wheel") %>%
  clean_names() %>%
  filter(!is.na(dumpster)) %>%
  mutate(year = as.character(year),
         trash_wheel = "Gwynnda Trash Wheel")
```

Combined dataset

```r
combined_trash_wheel_data <- bind_rows(mr_trash_wheel, prof_trash_wheel, gwynnda_trash_wheel)
print(combined_trash_wheel_data)
```

```
## # A tibble: 1,033 x 15
##    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
##  7        7 May   2014  2014-05-21 00:00:00        1.91                  8
##  8        8 May   2014  2014-05-28 00:00:00        3.7                  16
##  9        9 June  2014  2014-06-05 00:00:00        2.52                 14
## 10       10 June  2014  2014-06-11 00:00:00        3.76                 18
## # i 1,023 more rows
## # i 9 more variables: plastic_bottles <dbl>, polystyrene <dbl>,
## #   cigarette_butts <dbl>, glass_bottles <dbl>, plastic_bags <dbl>,
## #   wrappers <dbl>, sports_balls <int>, homes_powered <dbl>, trash_wheel <chr>
```

The combined trash wheel dataset contains a total of 1033 observations, merging data from Mr. Trash Wheel, Professor Trash Wheel, and Gwynnda Trash Wheel. The dataset consists of 651 observations from Mr. Trash Wheel, 119 from Professor Trash Wheel, and 263 from Gwynnda Trash Wheel. Key variables in this dataset include `year`, `month`, and `date`, which indicate the time period when each observation was recorded, and `dumpster`, which represents the dumpster number used for each trash collection; `weight_tons`, showing the total weight of trash collected in tons; and `plastic_bottles`, which indicates the number of plastic bottles collected during each trash collection. Additionally, `cigarette_butts` counts the total number of cigarette butts collected, while `sports_balls` captures the number of sports balls found. Another significant variable is `homes_powered`, which estimates the number of homes that could be powered by the energy generated from the collected trash.

```r
# Total weight by Professor Trash Wheel
total_weight_professor <- combined_trash_wheel_data %>%
  filter(trash_wheel == "Professor Trash Wheel") %>%
  summarise(total_weight = sum(as.numeric(weight_tons), na.rm = TRUE)) %>%
  pull(total_weight)

print(total_weight_professor)
```

```
## [1] 246.74
```

```r
# Total number of cigarette butts by Gwynnda
total_cig_butts_gwynnda_june <- combined_trash_wheel_data %>%
  filter(trash_wheel == "Gwynnda Trash Wheel", month == "June", year == "2022") %>%
  summarise(total_cig_butts = sum(as.numeric(cigarette_butts), na.rm = TRUE)) %>%
  pull(total_cig_butts)

print(total_cig_butts_gwynnda_june)
```

```
## [1] 18120
```

According to the data, the total weight of trash collected by Professor Trash Wheel was 246.74 tons. Gwynnda Trash Wheel collected 1.812×10^4 cigarette butts in June of 2022.

Problem 3

```r
# Load data
bakers <- read.csv("hw2_data/gbb_datasets/bakers.csv")
bakes <- read.csv("hw2_data/gbb_datasets/bakes.csv")
results <- read.csv("hw2_data/gbb_datasets/results.csv", skip = 2)

# Clean data
bakers_clean <- bakers %>% clean_names()
bakes_clean <- bakes %>% clean_names()
results_clean <- results %>% clean_names()
```

```r
# Check bakers in 'results' not present in 'bakers'
missing_bakers_in_results <- anti_join(results_clean, bakers_clean,
                                       by = c("baker" = "baker_name"))
# print(missing_bakers_in_results)

# Check bakers in 'bakers' not present in 'results'
missing_bakers_in_bakers <- anti_join(bakers_clean, results_clean,
                                      by = c("baker_name" = "baker"))
# print(missing_bakers_in_bakers)

# Check bakes that do not have corresponding entries in 'results'
missing_bakes_in_results <-
  anti_join(bakes_clean, results_clean, by = c("baker" = "baker",
                                   "series" = "series",
                                   "episode" = "episode"))
# print(missing_bakes_in_results)

# Check bakers in 'bakes' that do not have corresponding entries in 'bakers'
missing_bakers_in_bakes <- anti_join(bakes_clean, bakers_clean,
                                     by = c("baker" = "baker_name"))
# print(missing_bakers_in_bakes)
```

```r
# Fix "name" problem
bakers_clean <- bakers_clean %>%
  mutate(first_name = word(`baker_name`, 1))
```

```r
# Merge data
combined_data <- merge(bakers_clean, results_clean, by.x = c("first_name", "series"),
                       by.y = c("baker", "series"), all.x = TRUE)

final_combined_data <- merge(combined_data, bakes_clean,
                             by.x = c("first_name", "series", "episode"),
                             by.y = c("baker", "series", "episode"), all.x = TRUE)

# Remove unnecessary columns
final_combined_data <- final_combined_data %>% select(-first_name)

# Export the final cleaned dataset
write_csv(final_combined_data, "hw2_data/gbb_datasets/final_combined_data.csv")
```

Data Cleaning Process

I started by importing the `bakers`, `bakes`, and `results` datasets and then cleaned them using the `clean_names()` function from the `janitor` package to convert all column names to a consistent format (lowercase and underscores).

During the data validation process using `anti_join()`, I identified discrepancies between the datasets:

All bakers listed in `results` were not present in the `bakers` dataset, and vice versa.
Similarly, all bakers in `bakes` were missing in the `bakers` dataset.

One major challenge I encountered was that the `baker_name` column in the `bakers` dataset contained full names, while the `results` and `bakes` datasets only used first names. To address this, I used `mutate()` and `word()` to extract the first name from `baker_name` and create a new `first_name` column in the `bakers` dataset, which allowed for accurate merging.

Next, I merged `bakers_clean` with `results_clean` using the `first_name` and `series` columns. Afterward, I merged the resulting dataset with `bakes_clean` using `first_name`, `series`, and `episode` as the merging keys.

Finally, I removed any unnecessary columns and exported the clean, merged dataset as final_combined_data.csv.

Discussion of the Final Dataset

The final dataset effectively combines information about each baker, their performances, and their bakes across all episodes. It includes details such as the full name of each baker, the series and episode they participated in, their technical challenge results, and the specifics of their bakes. Despite the challenge of dealing with inconsistent name formats, the cleaning and merging process resulted in a comprehensive, well-structured dataset ready for analysis.

```r
# Filter for Seasons 5 through 10 and for Star Baker or Winner results
star_bakers_winners <- final_combined_data %>%
  filter(series %in% c(5, 6, 7, 8, 9, 10) & grepl("STAR BAKER|WINNER", result, ignore.case = TRUE)) %>%
  select(series, episode, baker_name, result) %>%
  arrange(series, episode)

# Display the resulting table
print(star_bakers_winners)
```

```
##    series episode          baker_name     result
## 1       5       1    Nancy Birtwhistle STAR BAKER
## 2       5       2        Richard Burr STAR BAKER
## 3       5       3        Luis Troyano STAR BAKER
## 4       5       4        Richard Burr STAR BAKER
## 5       5       5          Kate Henry STAR BAKER
## 6       5       6        Chetna Makan STAR BAKER
## 7       5       7        Richard Burr STAR BAKER
## 8       5       8        Richard Burr STAR BAKER
## 9       5       9        Richard Burr STAR BAKER
## 10      5      10    Nancy Birtwhistle     WINNER
## 11      6       1      Marie Campbell STAR BAKER
## 12      6       2         Ian Cumming STAR BAKER
## 13      6       3         Ian Cumming STAR BAKER
## 14      6       4         Ian Cumming STAR BAKER
## 15      6       5      Nadiya Hussain STAR BAKER
## 16      6       6           Mat Riley STAR BAKER
## 17      6       7           Tamal Ray STAR BAKER
## 18      6       8      Nadiya Hussain STAR BAKER
## 19      6       9      Nadiya Hussain STAR BAKER
## 20      6      10       Nadiya Hussain     WINNER
## 21      7       1          Jane Beedle STAR BAKER
## 22      7       2        Candice Brown STAR BAKER
## 23      7       3        Tom Gilliford STAR BAKER
## 24      7       4     Benjamina Ebuehi STAR BAKER
## 25      7       5        Candice Brown STAR BAKER
## 26      7       6        Tom Gilliford STAR BAKER
## 27      7       7         Andrew Smyth STAR BAKER
## 28      7       8        Candice Brown STAR BAKER
## 29      7       9         Andrew Smyth STAR BAKER
## 30      7      10        Candice Brown     WINNER
## 31      8       1 Steven Carter-Bailey STAR BAKER
## 32      8       2 Steven Carter-Bailey STAR BAKER
## 33      8       3   Julia Chernogorova STAR BAKER
## 34      8       4            Kate Lyon STAR BAKER
## 35      8       5         Sophie Faldo STAR BAKER
## 36      8       6         Liam Charles STAR BAKER
## 37      8       7 Steven Carter-Bailey STAR BAKER
## 38      8       8          Stacey Hart STAR BAKER
## 39      8       9         Sophie Faldo STAR BAKER
## 40      8      10         Sophie Faldo     WINNER
## 41      9       1        Manon Lagrave STAR BAKER
## 42      9       2         Rahul Mandal STAR BAKER
## 43      9       3         Rahul Mandal STAR BAKER
## 44      9       4  Dan Beasley-Harling STAR BAKER
## 45      9       5      Kim-Joy Hewlett STAR BAKER
## 46      9       6      Briony Williams STAR BAKER
## 47      9       7      Kim-Joy Hewlett STAR BAKER
## 48      9       8          Ruby Bhogal STAR BAKER
## 49      9       9          Ruby Bhogal STAR BAKER
## 50      9      10         Rahul Mandal     WINNER
## 51     10       1 Michelle Evans-Fecci STAR BAKER
## 52     10       2       Alice Fevronia STAR BAKER
## 53     10       3  Michael Chakraverty STAR BAKER
## 54     10       4      Steph Blackwell STAR BAKER
## 55     10       5      Steph Blackwell STAR BAKER
## 56     10       6      Steph Blackwell STAR BAKER
## 57     10       7           Henry Bird STAR BAKER
## 58     10       8      Steph Blackwell STAR BAKER
## 59     10       9       Alice Fevronia STAR BAKER
## 60     10      10       David Atherton     WINNER
```

Comment on the Table

Predictable Overall Winners: Contestants who consistently earned the "Star Baker" title across multiple episodes demonstrated strong performance and could have been expected to win the overall competition.

Surprises: There were some bakers who didn't earn the "Star Baker" title at all throughout the competition but still managed to win the overall title, making their victory quite unexpected.

```r
# Import the viewership data
viewers <- read_csv("hw2_data/gbb_datasets/viewers.csv", show_col_types = FALSE)

# Clean the column names
```

```
viewers <- viewers %>% clean_names()

# Display the first 10 rows of the cleaned dataset
head(viewers, 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>
```

```
# Calculate the average viewership for Season 1 and Season 5
avg_viewership_season1 <- mean(viewers$series_1, na.rm = TRUE)
avg_viewership_season5 <- mean(viewers$series_5, na.rm = TRUE)

# Display the average viewership results
avg_viewership_season1
```

```
## [1] 2.77
```

```
avg_viewership_season5
```

```
## [1] 10.0393
```

The average viewership in Season 1 is 2.77 and the average viewership in Season 5 is 10.0393.