p8105_hw2_dl3757

Dang Lin dl3757

2024-10-02

```
# Import the libraries
library(tidyverse)
library(readxl)
```

Problem 1

```
# Read the csv file and clean the dataset
nyc_transit <- read_csv("./NYC_Transit_Subway_Entrance_And_Exit_Data.csv",
                        na = c("NA", ".", "")) %>%
  janitor::clean_names() %>%
  select(line, station_name, station_latitude,
         station_longitude, route1:route11, entry, vending,
         entrance_type, ada) %>%
  mutate(across(starts_with("route"), as.character)) %>%
  mutate(entry = case_match(entry, "YES" ~ TRUE, "NO" ~ FALSE))
```

Firstly, we read the CSV file and applied the janitor::clean_names() function to make all the variable names tidy and consistent in format. Next, we used the select() function to choose the necessary variables from the original dataset and rearranged the position of each column. After that, we converted all the route columns to character variables. Finally, we transformed the "entry" variable from a character to a logical variable using the case_match() function. After these data-cleaning steps, the dataset contains 1,868 observations and 19 variables, meaning the cleaned dataset has 1,868 rows and 19 columns. For instance, variables such as "station_name," "line," "station_latitude," "station_longitude," and different routes are included. This dataset is not entirely tidy because it contains a lot of missing values, which we may need to address by removing some of them depending on the analysis. Additionally, some route variables were not initially in the correct format, so converting them to character variables was necessary before performing further analysis.

```
# The number of distinct stations
distinct_station <- nyc_transit %>%
  distinct(station_name, line)

nrow(distinct_station)
```

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

There are 465 distinct stations.

```
# The number of distinct ADA-compliant stations
ada_compliant_station <- nyc_transit %>%
  filter(ada == TRUE) %>%
  distinct(station_name, line)
```

```r
nrow(ada_compliant_station)
```

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

There are 84 ADA-compliant stations.

```r
# The proportion of stations entrances/exits without vending allow entrance
nyc_transit %>%
  filter(vending == "NO") %>%
  pull(entry) %>%
  mean()
```

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

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

```r
# The number of distinct stations serve the A train
distinct_A <- nyc_transit %>%
  pivot_longer(
    route1:route11,
    names_to = "route_num",
    values_to = "route") %>%
  filter(route == "A") %>%
  distinct(station_name, line)

nrow(distinct_A)
```

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

Sixty distinct stations serve the A train.

```r
# The number of distinct stations that are ADA-compliant serve the A train
distinct_A_ada <- nyc_transit %>%
  pivot_longer(
    route1:route11,
    names_to = "route_num",
    values_to = "route") %>%
  filter(route == "A", ada == TRUE) %>%
  distinct(station_name, line)

nrow(distinct_A_ada)
```

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

Of these 60 distinct stations that serve the A train, 17 stations are ADA-compliant.

Problem 2

```r
# Load and clean the dataset "Mr. Trash Wheel"
mr_trash_wheel <- read_excel("./202409 Trash Wheel Collection Data.xlsx",
                             sheet = "Mr. Trash Wheel", range = "A2:N655") %>%
  janitor::clean_names() %>%
  drop_na(dumpster) %>%
  mutate(sports_balls = as.integer(round(sports_balls, 0))) %>%
  mutate(year = as.character(year)) %>%
  mutate(sheet = "Mr.")
```

```r
# Load and clean the dataset "Professor Trash Wheel"
professor_trash_wheel <- read_excel("./202409 Trash Wheel Collection Data.xlsx",
                          sheet = "Professor Trash Wheel", range = "A2:M123") %>%
  janitor::clean_names() %>%
  drop_na(dumpster) %>%
  mutate(year = as.character(year)) %>%
  mutate(sheet = "Professor")
```

```r
# Load and clean the dataset "Gwynnda Trash Wheel"
gwynnda_trash_wheel <- read_excel("./202409 Trash Wheel Collection Data.xlsx",
                          sheet = "Gwynnda Trash Wheel", range = "A2:L266") %>%
  janitor::clean_names() %>%
  drop_na(dumpster) %>%
  mutate(year = as.character(year)) %>%
  mutate(sheet = "Gwynnda")
```

```r
# Produce a single tidy dataset by merging different datasets
tidy_dataset =
  bind_rows(mr_trash_wheel, professor_trash_wheel,
            gwynnda_trash_wheel) %>%
  janitor::clean_names() %>%
  select(sheet, everything())
```

After a series of data cleaning steps, the Mr. Trash Wheel dataset contains 651 observations and 15 variables, the Professor Trash Wheel dataset incorporates 119 observations and 14 variables, and the Gwynnda Trash Wheel dataset includes 263 observations and 13 variables. It is worth mentioning that most of the values in the "Wrappers" column of the Gwynnda Trash Wheel dataset are listed as "NA". However, we did not omit the rows without values for "Wrappers" because doing so would result in the loss of many informative observations. Subsequently, we combined the three datasets into one to perform a more comprehensive analysis. The resulting tidy dataset contains 1033 observations and 15 variables, with each observation labeled according to its source sheet. This combined dataset includes important variables such as "weight_tons", "volume_cubic_yards", "plastic_bottles", and others, with data collected between 2014 and 2024. The total weight of trash collected by Professor Trash Wheel is 246.74 based on available data. Moreover, the total number of cigarette butts collected by Gwynnda in June of 2022 is 1.812×10^4.

Problem 3

```r
# Load and clean the dataset
bakers <- read_csv("/Users/apple/Desktop/P8105_hw2_dl3757/gbb_datasets/bakers.csv") %>%
  janitor::clean_names() %>%
  mutate(baker = sub(" .*", "", baker_name)) %>%
  select(baker, everything(), -baker_name) %>%
  arrange(baker)

bakes <- read_csv("/Users/apple/Desktop/P8105_hw2_dl3757/gbb_datasets/bakes.csv") %>%
  janitor::clean_names() %>%
  select(baker, series, everything()) %>%
  arrange(baker)

results <- read_csv("/Users/apple/Desktop/P8105_hw2_dl3757/gbb_datasets/results.csv",
                    skip = 2) %>%
  janitor::clean_names() %>%
```

```r
  arrange(baker)
```

```r
# Check for completeness and correctness
bakers_bakes <- anti_join(bakers, bakes, by = "baker")

bakes_results <- anti_join(bakes, results, by = c("series", "episode"))
```

```r
# Produce a single tidy dataset by merging different datasets
baker_tidy_1 <- left_join(bakes, bakers) %>%
  janitor::clean_names() %>%
  select(baker, series, episode, everything())

baker_tidy_2 <- right_join(bakers, results) %>%
  janitor::clean_names() %>%
  select(baker, series, episode, result, everything()) %>%
  arrange(baker)

baker_tidy_3 <- right_join(baker_tidy_1, baker_tidy_2) %>%
  janitor::clean_names() %>%
  select(baker, series, episode, result, everything()) %>%
  arrange(baker)

# Export the CSV
write_csv(baker_tidy_3,
          "/Users/apple/Desktop/P8105_hw2_dl3757/gbb_datasets/bake_tidy_data.csv")
```

Firstly, the three datasets—"bakers," "bakes," and "results"—were imported into R Studio using the read_csv() function. Then, the janitor::clean_names() function was applied to convert the variable names to a tidy and consistent format. For the "baker_name" column in the "baker" dataset, we created a new variable called "baker" that contains only the first names of the bakers, making it easier to merge the datasets in later steps. After creating the new variable, we used the select() function to rearrange the column positions and the arrange() function to order the rows by the bakers' names. Next, we checked the completeness and correctness of the datasets but chose not to remove any potentially overlapping rows and unmatched records. Subsequently, we used the left_join() and right_join() functions to combine the three datasets into a single tidy dataset. Finally, we saved this newly created merged dataset as a CSV file. The final dataset contains 1,136 observations and 10 variables, including "baker," "series," "episode," "results," and other relevant information for each baker. It is worth noting that the final dataset contains many missing values. However, we chose not to remove these values, as most of them will not affect our analysis.

```r
# Create a table showing the star baker or winner of each episode in Seasons 5 through 10
star_baker <- baker_tidy_3 %>%
  janitor::clean_names() %>%
  filter(result == c("STAR BAKER"),
         series >= 5 & series <= 10) %>%
  arrange(series, episode) %>%
  select(series, episode, baker, result)

winner <- baker_tidy_3 %>%
  janitor::clean_names() %>%
  filter(result == c("WINNER"),
         series >= 5 & series <= 10) %>%
  arrange(series, episode) %>%
  select(series, episode, baker, result)

table <- bind_rows(winner, star_baker)
```

```
knitr::kable(table, caption = "Winner and Star Bakers")
```

Table 1: Winner and Star Bakers

series	episode	baker	result
5	10	Nancy	WINNER
6	10	Nadiya	WINNER
7	10	Candice	WINNER
8	10	Sophie	WINNER
9	10	Rahul	WINNER
10	10	David	WINNER
5	1	Nancy	STAR BAKER
5	2	Richard	STAR BAKER
5	3	Luis	STAR BAKER
5	4	Richard	STAR BAKER
5	5	Kate	STAR BAKER
5	6	Chetna	STAR BAKER
5	7	Richard	STAR BAKER
5	8	Richard	STAR BAKER
5	9	Richard	STAR BAKER
6	1	Marie	STAR BAKER
6	2	Ian	STAR BAKER
6	3	Ian	STAR BAKER
6	4	Ian	STAR BAKER
6	5	Nadiya	STAR BAKER
6	6	Mat	STAR BAKER
6	7	Tamal	STAR BAKER
6	8	Nadiya	STAR BAKER
6	9	Nadiya	STAR BAKER
7	1	Jane	STAR BAKER
7	2	Candice	STAR BAKER
7	3	Tom	STAR BAKER
7	4	Benjamina	STAR BAKER
7	5	Candice	STAR BAKER
7	6	Tom	STAR BAKER
7	7	Andrew	STAR BAKER
7	8	Candice	STAR BAKER
7	9	Andrew	STAR BAKER
8	1	Steven	STAR BAKER
8	2	Steven	STAR BAKER
8	3	Julia	STAR BAKER
8	4	Kate	STAR BAKER
8	5	Sophie	STAR BAKER
8	6	Liam	STAR BAKER
8	7	Steven	STAR BAKER
8	8	Stacey	STAR BAKER
8	9	Sophie	STAR BAKER
9	1	Manon	STAR BAKER
9	2	Rahul	STAR BAKER
9	3	Rahul	STAR BAKER
9	4	Dan	STAR BAKER
9	5	Kim-Joy	STAR BAKER
9	6	Briony	STAR BAKER

series	episode	baker	result
9	7	Kim-Joy	STAR BAKER
9	8	Ruby	STAR BAKER
9	9	Ruby	STAR BAKER
10	1	Michelle	STAR BAKER
10	2	Alice	STAR BAKER
10	3	Michael	STAR BAKER
10	4	Steph	STAR BAKER
10	5	Steph	STAR BAKER
10	6	Steph	STAR BAKER
10	7	Henry	STAR BAKER
10	8	Steph	STAR BAKER
10	9	Alice	STAR BAKER

If a baker frequently earns STAR BAKER across episodes within a specific series, they are more likely to become the final WINNER of that series. From this table, we can observe that Nadiya and Rahul were predictable winners because they consistently achieved STAR BAKER during their series. However, despite Richard, Ian, and Steph earning STAR BAKER the most times in their respective series, they unfortunately did not win the final title.

```
# Import the viewership data
viewers <- read_csv("/Users/apple/Desktop/P8105_hw2_dl3757/gbb_datasets/viewers.csv") %>%
  janitor::clean_names()

# View the first 10 rows of the dataset
viewers_10 <- head(viewers, 10)
knitr::kable(viewers_10)
```

episode	series_1	series_2	series_3	series_4	series_5	series_6	series_7	series_8	series_9	series_10
1	2.24	3.10	3.85	6.60	8.510	11.62	13.58	9.46	9.55	9.62
2	3.00	3.53	4.60	6.65	8.790	11.59	13.45	9.23	9.31	9.38
3	3.00	3.82	4.53	7.17	9.280	12.01	13.01	8.68	8.91	8.94
4	2.60	3.60	4.71	6.82	10.250	12.36	13.29	8.55	8.88	8.96
5	3.03	3.83	4.61	6.95	9.950	12.39	13.12	8.61	8.67	9.26
6	2.75	4.25	4.82	7.32	10.130	12.00	13.13	8.61	8.91	8.70
7	NA	4.42	5.10	7.76	10.280	12.35	13.45	9.01	9.22	8.98
8	NA	5.06	5.35	7.41	9.023	11.09	13.26	8.95	9.69	9.19
9	NA	NA	5.70	7.41	10.670	12.65	13.44	9.03	9.50	9.34
10	NA	NA	6.74	9.45	13.510	15.05	15.90	10.04	10.34	10.05

```
# Calculate the average viewership for Season 1 and 5
viewership_avg <- viewers %>%
  summarise(`Average Viewership Season 1` = mean(series_1, na.rm = TRUE),
            `Average Viewership Season 5` = mean(series_5, na.rm = TRUE))

knitr::kable(viewership_avg,
             caption = "Average Viewership in Season 1 and 5")
```

Table 3: Average Viewership in Season 1 and 5

Average Viewership Season 1	Average Viewership Season 5
2.77	10.0393

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