P3 Darkkhaki

AUTHOR
Zhang XiangHui, Lin WeiChen, Priscilla Thung, Tang Guan You, Arshad Bin Mazlan, Wong Jing Yong Shawn

1 Introduction

1.1 Data and Library Import

1.2 Data Import and Cleaning

```
# Read the CSV file and remove all address with NA
restaurants <- read_csv("updated_restaurant_list.csv") |>
  drop_na(full_address)
```

```
Rows: 63469 Columns: 12
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (5): name, category, price_range, full_address, zip_code
dbl (7): id, position, score, ratings, lat, lng, Year Partnered

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.
```

```
# Extract the state abbreviation using a regular expression and clean it up
restaurants <- restaurants %>%
  mutate(state = str_extract(full_address, ",\\s*([A-Z]{2})\\s*,") %>%
                   str_replace_all("^,\\s*|\\s*,$", ""))

# Select the state and Year Partnered columns
filtered_restaurants <- restaurants %>%
  select(id, state, `Year Partnered`)

# Remove all records with empty state
filtered_restaurants <- filtered_restaurants |>
  drop_na(state)

filtered_restaurants
```

```
# A tibble: 62,845 × 3
      id state `Year Partnered`
   <dbl> <chr>            <dbl>
 1     1 AL                2022
 2     2 AL                2018
 3     3 AL                2022
```

```
  5     5 AL                2018
  6     6 AL                2018
  7     7 AL                2024
  8     8 AL                2018
  9     9 AL                2022
 10    10 AL                2018
# ℹ 62,835 more rows
```

```
# Check for NA rows
#na_count <- sum(is.na(filtered_restaurants$state))
#na_count

# Print records with NA in the state column
#records_with_na <- filtered_restaurants %>%
#  filter(is.na(state))
#records_with_na
```

1.3 Group each state by the number of restaurants for each year

```
# Group the data by state and year
grouped_restaurants <- filtered_restaurants |>
  group_by(state, `Year Partnered`) |>
  summarise(count = n())
```

```
`summarise()` has grouped output by 'state'. You can override using the
`.groups` argument.
```

```
grouped_restaurants
```

```
# A tibble: 139 × 3
# Groups:   state [22]
   state `Year Partnered` count
   <chr>            <dbl> <int>
 1 AL                2018   159
 2 AL                2019   156
 3 AL                2020   163
 4 AL                2021   137
 5 AL                2022   144
 6 AL                2023   179
 7 AL                2024   167
 8 AR                2018     2
 9 AR                2019     4
10 AR                2020     4
# ℹ 129 more rows
```

```
# Calculate the percentage increase after each year
grouped_restaurants <- grouped_restaurants |>
  group_by(state) |>
  mutate(percentage_increase = (count - lag(count)) / lag(count) * 100)

grouped_restaurants
```

```
# A tibble: 139 × 4
# Groups:   state [22]
   state `Year Partnered` count percentage_increase
   <chr>            <dbl> <int>               <dbl>
 1 AL                2018   159              NA
 2 AL                2019   156              -1.89
 3 AL                2020   163               4.49
 4 AL                2021   137             -16.0
 5 AL                2022   144               5.11
 6 AL                2023   179              24.3
 7 AL                2024   167              -6.70
 8 AR                2018     2              NA
 9 AR                2019     4             100
10 AR                2020     4               0
# ℹ 129 more rows
```

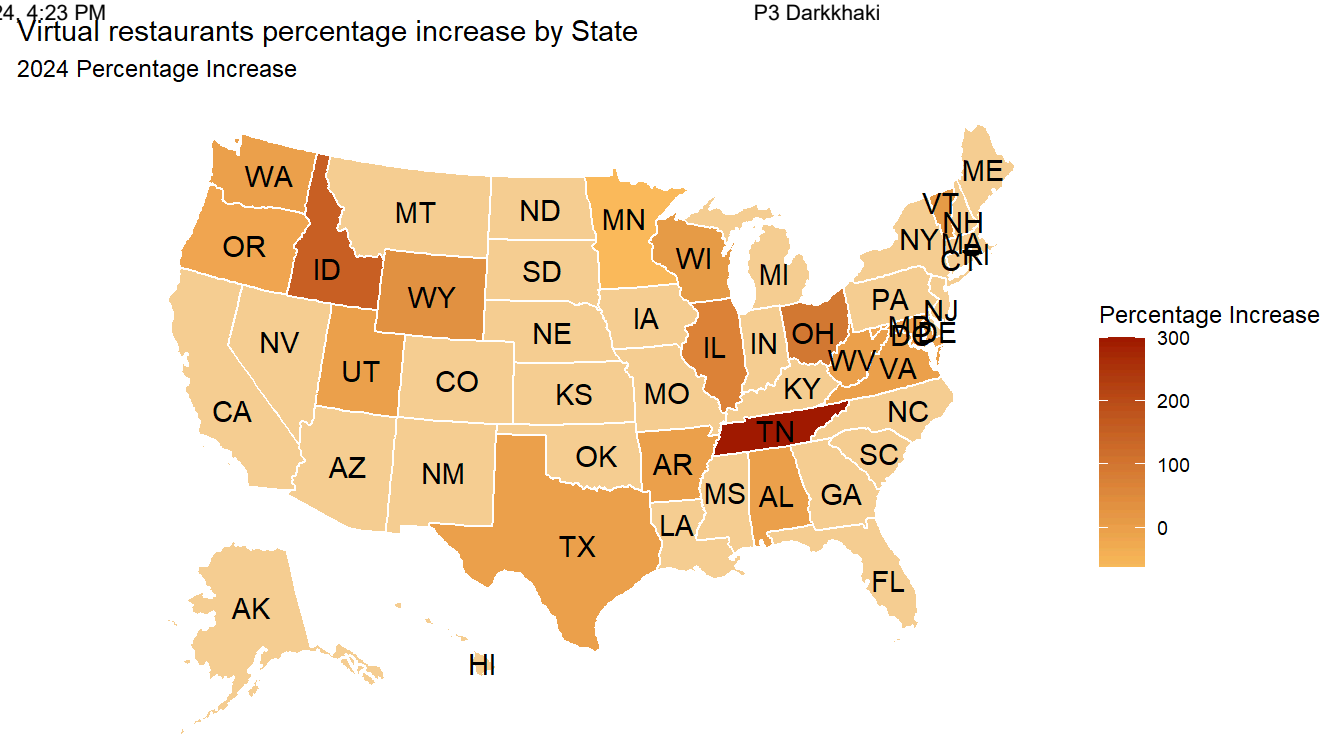
1.5 Map visualisation of USA

```
#data(World)
#ggplot(World, aes(fill = continent)) + geom_sf()

#zaf <-
#  filter(World, name == "United States")
#ggplot(zaf) + geom_sf(fill = "lightgreen")

# Prepare the latest year's percentage increase for each state
latest_year_data <- grouped_restaurants %>%
  group_by(state) %>%
  filter(`Year Partnered` == max(`Year Partnered`)) %>%
  ungroup()

# Plotting the US map with the percentage increase
plot_usmap(data = latest_year_data, values = "percentage_increase", color = "white", lab
  scale_fill_continuous(low = "#fabb5c", high = "#a11902", na.value = "#f7cf92", name =
  theme(legend.position = "right") +
  labs(title = "Virtual restaurants percentage increase by State", subtitle = "2024 Perc
```




```
        # Checking the states that are included in the data
        unique_states <- grouped_restaurants %>%
          distinct(state)
        unique_states
```

```
# A tibble: 22 × 1
# Groups:   state [22]
   state
   <chr>
 1 AL
 2 AR
 3 DC
 4 ID
 5 IL
 6 MD
 7 MN
 8 NE
 9 NW
10 OH
# ℹ 12 more rows
```