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_v2.csv") |>
  drop_na(full_address)

# 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                2023
 2     2 AL                2022
 3     3 AL                2021
 4     4 AL                2020
 5     5 AL                2021
 6     6 AL                2021
 7     7 AL                2023
 8     8 AL                2023
 9     9 AL                2022
10    10 AL                2023
# 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())

grouped_restaurants
```

```
# A tibble: 95 × 3
# Groups:   state [22]
   state `Year Partnered` count
   <chr>            <dbl> <int>
 1 AL                2019    37
 2 AL                2020    69
 3 AL                2021   155
 4 AL                2022   291
 5 AL                2023   553
 6 AR                2020     1
 7 AR                2021     3
 8 AR                2022     9
 9 AR                2023    15
10 DC                2019    53
# ℹ 85 more rows
```

1.4 Percentage increase after each year

```
# 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: 95 × 4
# Groups:   state [22]
   state `Year Partnered` count percentage_increase
   <chr>           <dbl> <int>           <dbl>
 1 AL               2019    37            NA
 2 AL               2020    69            86.5
 3 AL               2021   155           125.
 4 AL               2022   291            87.7
 5 AL               2023   553            90.0
 6 AR               2020     1            NA
 7 AR               2021     3           200
 8 AR               2022     9           200
 9 AR               2023    15            66.7
10 DC               2019    53            NA
# ℹ 85 more rows
```

1.5 Map visualisation of USA

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

# Get centroids for each state
centroid_labels <- usmapdata::centroid_labels("states")

# Rename the column in centroid_labels to match latest_year_data
centroid_labels <- centroid_labels %>%
  rename(state = abbr)

# Join centroids to data
state_labels <- merge(latest_year_data, centroid_labels, by = "state")

# Extract x and y coordinates from geom column using stringr and add 2 new column lon an
state_labels <- state_labels %>%
  mutate(
    lon_lat = str_extract_all(geom, "-?\\d+\\.?\\d*"),
    lon = as.numeric(sapply(lon_lat, function(x) x[1])),
    lat = as.numeric(sapply(lon_lat, function(x) x[2]))
  ) %>%
  select(-lon_lat)
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `lon_lat = str_extract_all(geom, "-?\\d+\\.?\\d*")`.
Caused by warning in `stri_extract_all_regex()`:
! argument is not an atomic vector; coercing
```

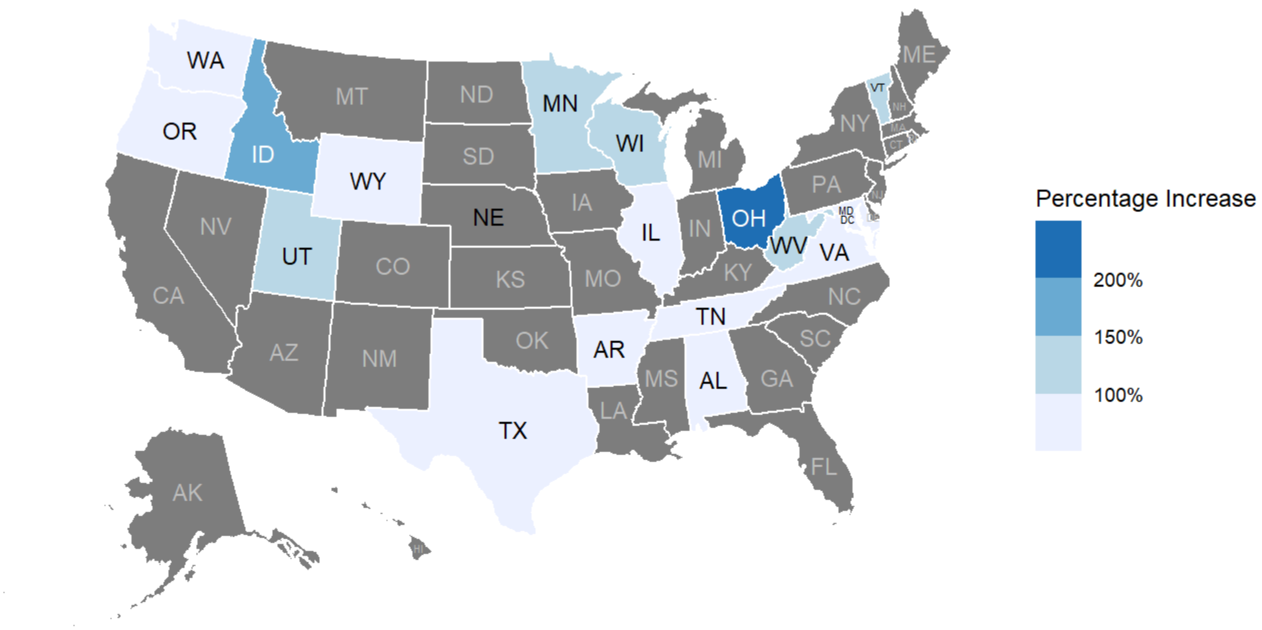
```
# Extract x and y coordinates from geom column using stringr and add 2 new column lon an
centroid_labels <- centroid_labels %>%
  mutate(
    lon_lat = str_extract_all(geom, "-?\\d+\\.?\\d*"),
    lon = as.numeric(sapply(lon_lat, function(x) x[1])),
    lat = as.numeric(sapply(lon_lat, function(x) x[2]))
  ) %>%
  select(-lon_lat)
```

```
Warning: There was 1 warning in `stopifnot()`.
ℹ In argument: `lon_lat = str_extract_all(geom, "-?\\d+\\.?\\d*")`.
Caused by warning in `stri_extract_all_regex()`:
! argument is not an atomic vector; coercing
```

```
# missing_states is used for plotting the label of the states with no data
all_states <- unique(centroid_labels$state)
plotted_states <- unique(latest_year_data$state)
missing_states <- setdiff(all_states, plotted_states)

# Define a list of states with smaller areas (For a smaller size of the label text)
small_area_states <- c("CT", "DE", "DC", "HI", "MD", "MA", "NH", "NJ", "RI", "VT")

# Plotting the US map with the percentage increase
p <- plot_usmap(data = latest_year_data, values = "percentage_increase", color = "white"
  theme(legend.position = "right") +
  scale_fill_fermenter(palette = "Blues", name = "Percentage Increase", label = percent_
  labs(title = "Virtual restaurants percentage increase by State", subtitle = "2024 Perc
  theme(
    plot.caption = element_markdown()
  ) +
  # First geom_text is used for plotting states with data
  geom_text(data = state_labels, aes(
    x = lon, y = lat,
    label = state,
  ), color = ifelse(is.na(state_labels$percentage_increase) | state_labels$percentage_in
  size = ifelse(state_labels$state %in% small_area_states, 1.5, 3)) +
  # Second geom_text is used for plotting states with no data
  geom_text(data = centroid_labels, aes(
    x = lon, y = lat,
    label = ifelse(state %in% missing_states, state, "")),
    size = ifelse(centroid_labels$state %in% small_area_states, 1.5, 3), color = "gray"
  )

# Set label font size for usmap library
#p$layers[[2]]$aes_params$size <- 2

p
```

Virtual restaurants percentage increase by State

2024 Percentage Increase



Source: Kaggle Dataset of Uber Eats USA Restaurants

```
# 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
```