Data Processing

Tong Zhou

2024-04-22

Contents

TOPIC

- Analyze concert information in Virginia for the upcoming two weeks (March 17th to March 30th).

DATA

- The data I collected come from two websites:
  - Concert Archives, where I gathered the concert information.
  - LatLong.net, where I collected the location information for places in Virginia.
- The data was accessed on March 27th. Concert Archives originally serves to record the history of concerts for singers/bands; LatLong.net is used to search for the longitude and latitude of places.
- I have created four datasets for different visualizations:
  - The first dataset contains variables 'Genre' and 'Count' to determine the relationship between genre and concert frequency.

Genre	Count
Rock	42

  - The second dataset contains variables 'Genre', 'Count', and 'Weekday' to analyze the relationship between weekdays and concert frequency for each genre.

Weekday	Genre	Count
Sunday	Folk	1

  - The third dataset also includes 'Genre', 'Count', and 'Weekday' but is used to analyze the overall relationship between weekdays and concert frequency across all genres. Here, the 'Genre' variable does not hold meaningful value, primarily for creating a heatmap

Weekday	Genre	Count
Sunday	All genres	20

- The fourth dataset contains variables ‘Place’, ‘Count’, ‘Lon’, and ‘Lat’ to examine the relationship between location and concert frequency.

Place	Count	Lon	Lat
Alexandria	5	-77.0	38.8

Load packages

```
library(tidyverse)
library(patchwork)
library(ggmap)
library(maps)
library(ggrepel)
```

Read in data

```
# Read in the concert_data using read_csv
concert_raw = read_csv("con_infor.csv",show_col_types = FALSE)
head(concert_raw)
```

```
## # A tibble: 6 x 6
##   Performer                             Genre       Place         Day Month  Year
##   <chr>                                 <chr>       <chr>       <dbl> <chr> <dbl>
## 1 Haley Heynderickx                     Folk        Charlotte~     22 March  2024
## 2 Hermanos Gutiérrez                    Rock        Charlotte~     19 March  2024
## 3 Too Many Zooz                         Pop         Charlotte~     17 March  2024
## 4 Kane Brown / Tyler Hubbard / Parmalee Country/Pop Charlotte~     28 March  2024
## 5 Tony Trischka                         Bluegrass   Charlotte~     28 March  2024
## 6 The Zombies                           Rock        Charlotte~     29 March  2024
```

```
# Add longitude and latitude value for places included
lon_values <- c(-77.0469, -77.1073, -78.4767, -77.3064, -78.8597,
                -77.5636, -79.1423, -76.5280, -76.2859, -77.4360,
                -79.9414, -75.9779, -77.2653)

lat_values <- c(38.8048, 38.8816, 38.0293, 38.8462, 38.4496,
                39.1157, 37.4138, 36.9784, 36.8508, 37.5407,
                37.2707, 36.8529, 38.9012)
```

Review/clean datasets

- conduct data cleaning processes
- provide code analyzing the structure and layout of datasets

```r
# Separate rows with more than one genres
con_gen_sep <- concert_raw %>%
  separate_rows(Genre, sep = "/")

head(con_gen_sep)
```

```
## # A tibble: 6 x 6
##   Performer                             Genre     Place          Day Month  Year
##   <chr>                                 <chr>     <chr>        <dbl> <chr> <dbl>
## 1 Haley Heynderickx                     Folk      Charlottesv~    22 March  2024
## 2 Hermanos Gutiérrez                    Rock      Charlottesv~    19 March  2024
## 3 Too Many Zooz                         Pop       Charlottesv~    17 March  2024
## 4 Kane Brown / Tyler Hubbard / Parmalee Country   Charlottesv~    28 March  2024
## 5 Kane Brown / Tyler Hubbard / Parmalee Pop       Charlottesv~    28 March  2024
## 6 Tony Trischka                         Bluegrass Charlottesv~    28 March  2024
```

```r
# Add Weekday and parse the format for existing date as day-month-year
con_clean <- con_gen_sep %>%
  mutate(Date = paste(Day, Month, year(Sys.Date()), sep = " "),
         Date = dmy(Date))%>%
         mutate(Weekday = weekdays(Date))%>%
         select(-c(Day,Month, Year))

head(con_clean)
```

```
## # A tibble: 6 x 5
##   Performer                             Genre     Place        Date       Weekday
##   <chr>                                 <chr>     <chr>        <date>     <chr>
## 1 Haley Heynderickx                     Folk      Charlottes~ 2024-03-22 Friday
## 2 Hermanos Gutiérrez                    Rock      Charlottes~ 2024-03-19 Tuesday
## 3 Too Many Zooz                         Pop       Charlottes~ 2024-03-17 Sunday
## 4 Kane Brown / Tyler Hubbard / Parmalee Country   Charlottes~ 2024-03-28 Thursd~
## 5 Kane Brown / Tyler Hubbard / Parmalee Pop       Charlottes~ 2024-03-28 Thursd~
## 6 Tony Trischka                         Bluegrass Charlottes~ 2024-03-28 Thursd~
```

```r
# Create dataset type_counts for determining the relationship between Genre and Count
type_counts <- con_clean %>%
  group_by(Genre) %>%
  summarize(Count = n())%>%
  arrange(desc(Count))

# Reorder type_counts in descending order based on Count
type_counts$Genre <- factor(type_counts$Genre, levels = type_counts$Genre[order(type_counts$Count)])

head(type_counts)
```

```
## # A tibble: 6 x 2
##   Genre  Count
##   <fct>  <int>
## 1 Rock      42
## 2 Pop       23
## 3 Indie     11
## 4 Folk       9
## 5 Jazz       8
## 6 Comedy     7
```

```
write_csv(type_counts, "type_counts.csv")
```

```
# Create variable weekdays_ordered to determine the order of weekday
weekdays_ordered <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

# Create a complete grid with all combinations of Weekday and Genre
complete_grid <- expand_grid(Weekday = weekdays_ordered,
                             Genre = unique(con_clean$Genre))

# Create dataset weekday_counts for determining the relationship between Weekday and Count for each gen
weekday_counts <- con_clean %>%
  group_by(Weekday, Genre) %>%
  summarize(Count = n(), .groups = 'drop')

# Ensure dataset includes all weekday value for all genre by replacing NAs with 0
weekday_counts <- complete_grid %>%
  left_join(weekday_counts, by = c("Weekday", "Genre")) %>%
  replace_na(list(Count = 0))

# Reorder type_counts based on weekdays_ordered
weekday_counts$Weekday <- factor(weekday_counts$Weekday, levels = weekdays_ordered)

head(weekday_counts)
```

```
## # A tibble: 6 x 3
##   Weekday Genre     Count
##   <fct>   <chr>     <int>
## 1 Sunday  Folk          1
## 2 Sunday  Rock          6
## 3 Sunday  Pop           6
## 4 Sunday  Country       0
## 5 Sunday  Bluegrass     1
## 6 Sunday  Christian     0
```

```
write_csv(weekday_counts, "weekday_counts.csv")
```

```
# Create dataset weekday_total_counts for determining the relationship between Weekday and Count for ge
weekday_total_counts <- con_clean %>%
  group_by(Weekday) %>%
  summarize(TotalCount = n(), .groups = 'drop')


# Create data frame single_genre that has a single Genre value to aggregating all genres later
single_genre <- tibble(
  Weekday = weekdays_ordered,
  Genre = "All Genres",
  Count = 0
)
```

```
# Merge single_genre into weekday_total_counts
weekday_total_counts <- single_genre %>%
  left_join(weekday_total_counts, by = "Weekday") %>%
  mutate(Count = ifelse(is.na(TotalCount), 0, TotalCount)) %>%
  select(-TotalCount)

# Reorder type_counts based on weekdays_ordered
weekday_total_counts$Weekday <- factor(weekday_total_counts$Weekday, levels = weekdays_ordered)

head(weekday_total_counts)
```

```
## # A tibble: 6 x 3
##   Weekday   Genre      Count
##   <fct>     <chr>      <int>
## 1 Sunday    All Genres    20
## 2 Monday    All Genres     6
## 3 Tuesday   All Genres    14
## 4 Wednesday All Genres    19
## 5 Thursday  All Genres    18
## 6 Friday    All Genres    32
```

```
write_csv(weekday_total_counts, "weekday_total_counts.csv")
```

```
# Create dataset locations_with_counts for determining the relationship between Location and Count
locations_with_counts <- con_clean %>%
  group_by(Place) %>%
  summarize(Count = n())

# Add variable Lon and Lat into locations_with_counts
locations_with_counts <- locations_with_counts%>%
  mutate(Lon = lon_values,
         Lat = lat_values)

# Set the colors for points
point_colors <- c("Alexandria" = "#6a6f51", "Arlington" = "#6a6f51", "Charlottesville" = "#6a6f51",
                  "Fairfax" = "#6a6f51", "Harrisonburg" = "#6a6f51", "Leesburg" = "#6a6f51",
                  "Lynchburg" = "#6a6f51", "Newport" = "#6a6f51", "Norfolk" = "#6a6f51", "Roanoke" = "#


# Get the map data for Virginia
virginia_map <- map_data("state", region = "virginia")
write_csv(virginia_map, "virginia_map.csv")

head(locations_with_counts)
```

```
## # A tibble: 6 x 4
##   Place           Count   Lon   Lat
##   <chr>           <int> <dbl> <dbl>
## 1 Alexandria          5 -77.0  38.8
## 2 Arlington           1 -77.1  38.9
## 3 Charlottesville     8 -78.5  38.0
## 4 Fairfax             1 -77.3  38.8
## 5 Harrisonburg        1 -78.9  38.4
## 6 Leesburg            6 -77.6  39.1
```

```
write_csv(locations_with_counts, "locations_with_counts.csv")
```