San Francisco Bike Share Network

Mohammed Sharif

Understanding the San Francisco Bike Share Network: An Analytical Exploration

```
if (!require("lubridate")) {
    install.packages("lubridate")
    library(lubridate)
}
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
if (!require("ggplot2")) {
    install.packages("ggplot2e")
    library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```
if (!require("dplyr")) {
    install.packages("dplyr")
    library(dplyr)
}
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
if (!require("tidyr")) {
    install.packages("tidyr")
    library(tidyr)
}
```

```
## Loading required package: tidyr
```

```r
if (!require("scales")) {
    install.packages("scales")
    library(scales)
}
```

```
## Loading required package: scales
```

Introduction

The San Francisco Bike Share network is a vital urban mobility solution, offering valuable insights into commuter behavior, network utilization, and areas for improvement. This analysis focuses on two data sets from July 2014 and July 2015 to explore network evolution, identify critical stations and routes, and uncover opportunities for optimization. Comparing these two consecutive Julys provides insights into how user behavior and network performance evolve over time, offering a unique perspective on its operational dynamics.

Objectives to analyse :

- Evaluate the organization of the network through trip data and station metadata.
- Compare usage trends between 2014 and 2015 to identify patterns of growth or decline.
- Analyze spatial dynamics to pinpoint high-traffic stations and popular areas.
- Assess route popularity, station connectivity, and user behavior.
- Provide actionable insights and recommendations to improve the network.

Scope of the Assignment

- **Data Cleaning and Preparation**: Ensuring consistency, removing duplicates, and enriching datasets with station metadata.
- **Data Aggregation and Analysis**: Summarizing trip data to uncover patterns in route popularity, station traffic, and temporal trends.
- **Visualization**: Using visualizations to highlight spatial dynamics, route popularity, and trip patterns.
- **Insights and Recommendations**: Identifying critical findings and proposing actionable strategies for improvement.

Data Cleaning and Preparation

The raw data sets include trip-level data for two weeks in July 2014 and July 2015, as well as metadata about the bike share stations. Cleaning and preparing the data ensures consistency, reliability, and enrichment for further analysis.

Key Steps:

- **Consistency**: Standardized station names, IDs, and formats for compatibility across datasets.
- **Temporal Standardization**: Converted date columns into a consistent format for easier analysis.
- **Integrity Checks**: Removed duplicate records and verified completeness by checking for missing values.
- **Data Enrichment**: Mapped station metadata (e.g., latitude, longitude, dock count) to trip data for spatial analysis.

```
# Read the datasets
july_2014 <- read.csv("/Users/sharif/Desktop/2740/A10/SF-bikeshare-1-week-2014-07.csv")
july_2015 <- read.csv("/Users/sharif/Desktop/2740/A10/SF-bikeshare-1-week-2015-07.csv")
station_info <- read.csv("/Users/sharif/Desktop/2740/A10/SF-bikeshare-station-info.csv")

# Step 1: Ensure consistency in station names and IDs
# Rename columns in station_info for clarity and compatibility
station_info <- station_info %>%
  rename(
    station_id = id,
    station_name = name,
    latitude = lat,
    longitude = long,
    installation_date = installation_date
  )

# Step 2: Convert date columns to a consistent format
# Convert trip start and end dates to Date format for analysis
july_2014 <- july_2014 %>%
  mutate(
    start_date = as.Date(start_date_yyyymmdd),
    end_date = as.Date(end_date_yyyymmdd)
  ) %>%
  select(-start_date_yyyymmdd, -end_date_yyyymmdd)  # Drop original columns

july_2015 <- july_2015 %>%
  mutate(
    start_date = as.Date(start_date_yyyymmdd),
    end_date = as.Date(end_date_yyyymmdd)
  ) %>%
  select(-start_date_yyyymmdd, -end_date_yyyymmdd)

# Convert installation_date in station_info to Date format
station_info <- station_info %>%
  mutate(installation_date = as.Date(installation_date, format = "%m/%d/%Y"))
```

```r
# Step 3: Check for missing data
# Summarize missing values in each dataset to ensure completeness
missing_2014 <- colSums(is.na(july_2014))
missing_2015 <- colSums(is.na(july_2015))
missing_station_info <- colSums(is.na(station_info))

cat("Missing values in July 2014 dataset:\n")
```

```
## Missing values in July 2014 dataset:
```

```r
print(missing_2014)
```

```
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##           duration         start_date           end_date
##                  0                  0                  0
```

```r
cat("\nMissing values in July 2015 dataset:\n")
```

```
##
## Missing values in July 2015 dataset:
```

```r
print(missing_2015)
```

```
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##           duration         start_date           end_date
##                  0                  0                  0
```

```r
cat("\nMissing values in Station Info dataset:\n")
```

```
##
## Missing values in Station Info dataset:
```

```r
print(missing_station_info)
```

```
##        station_id      station_name          latitude         longitude
##                 0                 0                 0                 0
##        dock_count              city installation_date
##                 0                 0                 0
```

```r
# Step 4: Remove duplicate rows
# Ensuring data integrity by removing duplicate trip records
july_2014 <- july_2014 %>% distinct()
july_2015 <- july_2015 %>% distinct()

# Step 5: Enrich trip datasets with station metadata
# Add station metadata (e.g., latitude, longitude) to the trip datasets
july_2014 <- july_2014 %>%
```

```r
  left_join(station_info, by = c("start_station_id" = "station_id"))

july_2015 <- july_2015 %>%
  left_join(station_info, by = c("start_station_id" = "station_id"))

# Step 6: Summarize the cleaned data
# Calculate summary statistics to validate the cleaned datasets
summary_2014 <- july_2014 %>%
  summarize(
    total_trips = n(),
    unique_stations = n_distinct(station_name),
    date_range = paste(min(start_date), "to", max(start_date))
  )

summary_2015 <- july_2015 %>%
  summarize(
    total_trips = n(),
    unique_stations = n_distinct(station_name),
    date_range = paste(min(start_date), "to", max(start_date))
  )

cat("\nSummary of July 2014 dataset:\n")
```

```
##
## Summary of July 2014 dataset:
```

```r
print(summary_2014)
```

```
##   total_trips unique_stations               date_range
## 1        6875              69 2014-07-07 to 2014-07-13
```

```r
cat("\nSummary of July 2015 dataset:\n")
```

```
##
## Summary of July 2015 dataset:
```

```r
print(summary_2015)
```

```
##   total_trips unique_stations               date_range
## 1        7342              70 2015-07-06 to 2015-07-12
```

```r
# Step 7: Save cleaned datasets for further analysis
write.csv(july_2014, "Cleaned_July_2014.csv", row.names = FALSE)
write.csv(july_2015, "Cleaned_July_2015.csv", row.names = FALSE)

# Step 8: Output the structure of the cleaned data for validation
cat("\nStructure of cleaned July 2014 dataset:\n")
```

```
##
## Structure of cleaned July 2014 dataset:
```

```r
str(july_2014)
```

```
## 'data.frame':    6875 obs. of  13 variables:
##  $ start_station_name: chr  "Powell at Post (Union Square)" "Market at 4th" "Grant Avenue at Columbus
##  $ start_station_id  : int  71 76 73 50 2 61 75 28 71 60 ...
##  $ end_station_name  : chr  "Embarcadero at Bryant" "Market at 10th" "Powell at Post (Union Square)"
##  $ end_station_id    : int  54 67 71 63 4 54 57 32 39 46 ...
##  $ duration          : int  667 401 470 421 221 233 455 559 1386 602 ...
##  $ start_date        : Date, format: "2014-07-13" "2014-07-13" ...
##  $ end_date          : Date, format: "2014-07-13" "2014-07-13" ...
##  $ station_name      : chr  "Powell at Post (Union Square)" "Market at 4th" "Grant Avenue at Columbus
##  $ latitude          : num  37.8 37.8 37.8 37.8 37.3 ...
##  $ longitude         : num  -122 -122 -122 -122 -122 ...
##  $ dock_count        : int  19 19 15 23 27 27 19 23 19 15 ...
##  $ city              : chr  "San Francisco" "San Francisco" "San Francisco" "San Francisco" ...
##  $ installation_date : Date, format: "2013-08-23" "2013-08-25" ...
```

```r
cat("\nStructure of cleaned July 2015 dataset:\n")
```

```
##
## Structure of cleaned July 2015 dataset:
```

```r
str(july_2015)
```

```
## 'data.frame':    7342 obs. of  13 variables:
##  $ start_station_name: chr  "Howard at 2nd" "Temporary Transbay Terminal (Howard at Beale)" "San Jose
##  $ start_station_id  : int  63 55 10 41 77 42 16 16 76 50 ...
##  $ end_station_name  : chr  "Market at Sansome" "Powell Street BART" "SJSU - San Salvador at 9th" "W
##  $ end_station_id    : int  77 39 16 46 73 49 10 16 69 56 ...
##  $ duration          : int  121 444 444 166 624 363 570 82 402 5868 ...
##  $ start_date        : Date, format: "2015-07-12" "2015-07-12" ...
##  $ end_date          : Date, format: "2015-07-12" "2015-07-12" ...
##  $ station_name      : chr  "Howard at 2nd" "Temporary Transbay Terminal (Howard at Beale)" "San Jose
##  $ latitude          : num  37.8 37.8 37.3 37.8 37.8 ...
##  $ longitude         : num  -122 -122 -122 -122 -122 ...
##  $ dock_count        : int  19 23 15 15 27 15 15 15 19 23 ...
##  $ city              : chr  "San Francisco" "San Francisco" "San Jose" "San Francisco" ...
##  $ installation_date : Date, format: "2013-08-22" "2013-08-20" ...
```

Code Summary

- Renamed columns in `station_info` for clarity.
- Converted trip dates and installation dates to `Date` format.
- Checked for missing values with `colSums(is.na(...))`.
- Removed duplicates using `distinct()`.
- Merged station metadata into trip data sets using `left_join()`.
- Saved cleaned data sets as CSV files for further analysis.

Data Aggregation and Analysis

With the data cleaned, trip-level information was grouped and summarized to uncover network trends. The analysis focused on:

1. **Route Popularity**: Identifying the most traveled routes.
2. **Hourly Usage**: Analyzing peak hours for station usage.
3. **Weekly Summaries**: Comparing total trips and average trip duration across years.

```
# Read the cleaned datasets
july_2014 <- read.csv("Cleaned_July_2014.csv")
july_2015 <- read.csv("Cleaned_July_2015.csv")

filtered_2014 <- july_2014 %>%
  filter(duration < 3600)

filtered_2015 <- july_2015 %>%
  filter(duration < 3600)

# 1. Aggregate by origin and destination stations to calculate route popularity
route_popularity_2014 <- july_2014 %>%
  group_by(start_station_name, end_station_name) %>%
  summarize(
    total_trips = n(),
    avg_duration = mean(duration, na.rm = TRUE)
  ) %>%
  arrange(desc(total_trips))
```

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

```
route_popularity_2015 <- july_2015 %>%
  group_by(start_station_name, end_station_name) %>%
  summarize(
    total_trips = n(),
    avg_duration = mean(duration, na.rm = TRUE)
  ) %>%
  arrange(desc(total_trips))
```

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

```
# Print top 5 routes for validation
print(head(route_popularity_2014, 5))
```

```
## # A tibble: 5 x 4
## # Groups:   start_station_name [5]
##   start_station_name                  end_station_name total_trips avg_duration
##   <chr>                               <chr>                 <int>        <dbl>
## 1 Harry Bridges Plaza (Ferry Building) Embarcadero at ~          82        1315.
```

```
## 2 Embarcadero at Sansome                Steuart at Mark~          49         413.
## 3 Market at Sansome                     2nd at South Pa~          46         372.
## 4 San Francisco Caltrain (Townsend at~ Embarcadero at ~          46         834.
## 5 2nd at South Park                     Market at Sanso~          44         398.
```

```r
# 2. Aggregate by origin station and hour of the day
hourly_usage_2014 <- july_2014 %>%
  mutate(hour = hour(start_date)) %>%
  group_by(start_station_name, hour) %>%
  summarize(
    total_trips = n()
  ) %>%
  arrange(desc(total_trips))
```

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

```r
hourly_usage_2015 <- july_2015 %>%
  mutate(hour = hour(start_date)) %>%
  group_by(start_station_name, hour) %>%
  summarize(
    total_trips = n()
  ) %>%
  arrange(desc(total_trips))
```

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

```r
# Print hourly usage sample
print(head(hourly_usage_2014, 5))
```

```
## # A tibble: 5 x 3
## # Groups:   start_station_name [5]
##   start_station_name                              hour total_trips
##   <chr>                                          <int>       <int>
## 1 San Francisco Caltrain (Townsend at 4th)           0         542
## 2 Harry Bridges Plaza (Ferry Building)               0         313
## 3 San Francisco Caltrain 2 (330 Townsend)            0         312
## 4 Market at Sansome                                  0         302
## 5 Temporary Transbay Terminal (Howard at Beale)      0         274
```

```r
# 3. Weekly aggregates: total trip volume and average trip duration
weekly_aggregates_2014 <- july_2014 %>%
  summarize(
    total_trips = n(),
    avg_duration = mean(duration, na.rm = TRUE)
  )

weekly_aggregates_2015 <- july_2015 %>%
  summarize(
    total_trips = n(),
```

```
    avg_duration = mean(duration, na.rm = TRUE)
  )

# Print weekly aggregates
print(weekly_aggregates_2014)
```

```
##   total_trips avg_duration
## 1        6875     1161.846
```

```
print(weekly_aggregates_2015)
```

```
##   total_trips avg_duration
## 1        7342     1202.155
```

```
# 4. Save the aggregated results
write.csv(route_popularity_2014, "Route_Popularity_July_2014.csv", row.names = FALSE)
write.csv(route_popularity_2015, "Route_Popularity_July_2015.csv", row.names = FALSE)
write.csv(hourly_usage_2014, "Hourly_Usage_July_2014.csv", row.names = FALSE)
write.csv(hourly_usage_2015, "Hourly_Usage_July_2015.csv", row.names = FALSE)
```

Code Summary

- Aggregated data by origin and destination stations to identify popular routes.
- Summarized hourly usage trends to analyze peak times.
- Compared total trips and average duration for weekly summaries.
- Saved aggregated results as CSV files.

Key Metrics Analysis

To understand connectivity and traffic patterns, this step focuses on analyzing station-level metrics, including **traffic balance** (out-degree minus in-degree), which helps identify overused or underutilized stations. Additionally, station trip totals highlight the most popular stations and their role in the network.

Goals to analyse:

- Measure **station connectivity** using in-degree (trips ending) and out-degree (trips starting).
- Identify **high-traffic stations** based on total trip counts.
- Analyze traffic balance to evaluate station efficiency.

```
station_connectivity_2014 <- july_2014 %>%
  group_by(station_name) %>%
  summarize(
    out_degree = n(),
    in_degree = sum(end_station_name == station_name),
    balance = n() - sum(end_station_name == station_name)  # Traffic balance
  )
```

```r
station_connectivity_2015 <- july_2015 %>%
  group_by(station_name) %>%
  summarize(
    out_degree = n(),
    in_degree = sum(end_station_name == station_name),
    balance = n() - sum(end_station_name == station_name)  # Traffic balance
  )

# Identify high-traffic stations
station_usage_2014 <- july_2014 %>%
  group_by(station_name) %>%
  summarize(total_trips = n()) %>%
  arrange(desc(total_trips))

station_usage_2015 <- july_2015 %>%
  group_by(station_name) %>%
  summarize(total_trips = n()) %>%
  arrange(desc(total_trips))

# Print top results for validation
print(head(station_connectivity_2014, 5))
```

```
## # A tibble: 5 x 4
##   station_name      out_degree in_degree balance
##   <chr>                  <int>     <int>   <int>
## 1 2nd at Folsom            153         0     153
## 2 2nd at South Park        180         3     177
## 3 2nd at Townsend          273         5     268
## 4 5th at Howard            111         3     108
## 5 Adobe on Almaden          11         0      11
```

```r
print(head(station_usage_2014, 5))
```

```
## # A tibble: 5 x 2
##   station_name                                    total_trips
##   <chr>                                                 <int>
## 1 San Francisco Caltrain (Townsend at 4th)                542
## 2 Harry Bridges Plaza (Ferry Building)                    313
## 3 San Francisco Caltrain 2 (330 Townsend)                 312
## 4 Market at Sansome                                       302
## 5 Temporary Transbay Terminal (Howard at Beale)           274
```

```r
# Save station connectivity and usage
write.csv(station_connectivity_2014, "Station_Connectivity_July_2014.csv", row.names = FALSE)
write.csv(station_connectivity_2015, "Station_Connectivity_July_2015.csv", row.names = FALSE)
write.csv(station_usage_2014, "Station_Usage_July_2014.csv", row.names = FALSE)
write.csv(station_usage_2015, "Station_Usage_July_2015.csv", row.names = FALSE)

# Save top stations for summary
top_stations_2014 <- head(station_usage_2014, 10)
top_stations_2015 <- head(station_usage_2015, 10)
write.csv(top_stations_2014, "Top_Stations_July_2014.csv", row.names = FALSE)
write.csv(top_stations_2015, "Top_Stations_July_2015.csv", row.names = FALSE)
```

Code Summary

- Calculated in-degree and out-degree to analyze traffic flow.
- Identified high-traffic stations by summarizing total trips.
- Saved results (connectivity, usage, top stations) for future exploration.

Comparison Across Years

Analyzing changes between 2014 and 2015 reveals how the network adapts to shifting commuter behaviors and infrastructure developments. This step highlights:

- Changes in trip volumes, route popularity, and station usage.
- Identification of stations and routes with significant growth or decline.

```
# Compare total trip volume and average trip duration
comparison_summary <- data.frame(
  Year = c(2014, 2015),
  Total_Trips = c(weekly_aggregates_2014$total_trips, weekly_aggregates_2015$total_trips),
  Avg_Duration = c(weekly_aggregates_2014$avg_duration, weekly_aggregates_2015$avg_duration)
)

# Compare top routes between 2014 and 2015
top_routes_2014 <- head(route_popularity_2014, 10)
top_routes_2015 <- head(route_popularity_2015, 10)

# Compare station usage between 2014 and 2015
station_comparison <- merge(
  station_usage_2014 %>% rename(total_trips_2014 = total_trips),
  station_usage_2015 %>% rename(total_trips_2015 = total_trips),
  by = "station_name", all = TRUE
) %>%
  mutate(
    trip_change = total_trips_2015 - total_trips_2014,
    percent_change = (trip_change / total_trips_2014) * 100
  )

# Compare route popularity between 2014 and 2015
route_comparison <- merge(
  route_popularity_2014 %>% rename(total_trips_2014 = total_trips),
  route_popularity_2015 %>% rename(total_trips_2015 = total_trips),
  by = c("start_station_name", "end_station_name"), all = TRUE
) %>%
  mutate(
    trip_change = total_trips_2015 - total_trips_2014,
    percent_change = (trip_change / total_trips_2014) * 100
  )

# Validate key outputs
print(comparison_summary)
```

```
##   Year Total_Trips Avg_Duration
## 1 2014        6875     1161.846
## 2 2015        7342     1202.155
```

```
print(head(top_routes_2014, 5))
```

```
## # A tibble: 5 x 4
## # Groups:   start_station_name [5]
##   start_station_name                  end_station_name total_trips avg_duration
##   <chr>                               <chr>                  <int>        <dbl>
## 1 Harry Bridges Plaza (Ferry Building) Embarcadero at ~          82        1315.
## 2 Embarcadero at Sansome               Steuart at Mark~          49         413.
## 3 Market at Sansome                    2nd at South Pa~          46         372.
## 4 San Francisco Caltrain (Townsend at~ Embarcadero at ~          46         834.
## 5 2nd at South Park                    Market at Sanso~          44         398.
```

```
print(head(station_comparison, 5))
```

```
##        station_name total_trips_2014 total_trips_2015 trip_change
## 1     2nd at Folsom              153              157           4
## 2 2nd at South Park              180              152         -28
## 3   2nd at Townsend              273              330          57
## 4     5th at Howard              111              130          19
## 5  Adobe on Almaden               11               13           2
##   percent_change
## 1       2.614379
## 2     -15.555556
## 3      20.879121
## 4      17.117117
## 5      18.181818
```

```
# Save results
write.csv(comparison_summary, "Yearly_Comparison_Summary.csv", row.names = FALSE)
write.csv(top_routes_2014, "Top_Routes_2014.csv", row.names = FALSE)
write.csv(top_routes_2015, "Top_Routes_2015.csv", row.names = FALSE)
write.csv(station_comparison, "Station_Comparison.csv", row.names = FALSE)
write.csv(route_comparison, "Route_Comparison.csv", row.names = FALSE)
```

Code Summary

- Compared trip volumes and average duration across years.
- Extracted top routes for both years to identify trends.
- Analyzed changes in station and route usage, highlighting areas of growth or decline.

Visualization

1. Station Traffic Heatmap

Heatmaps visualize trip volume at stations based on latitude and longitude, highlighting spatial trends and high-traffic areas.

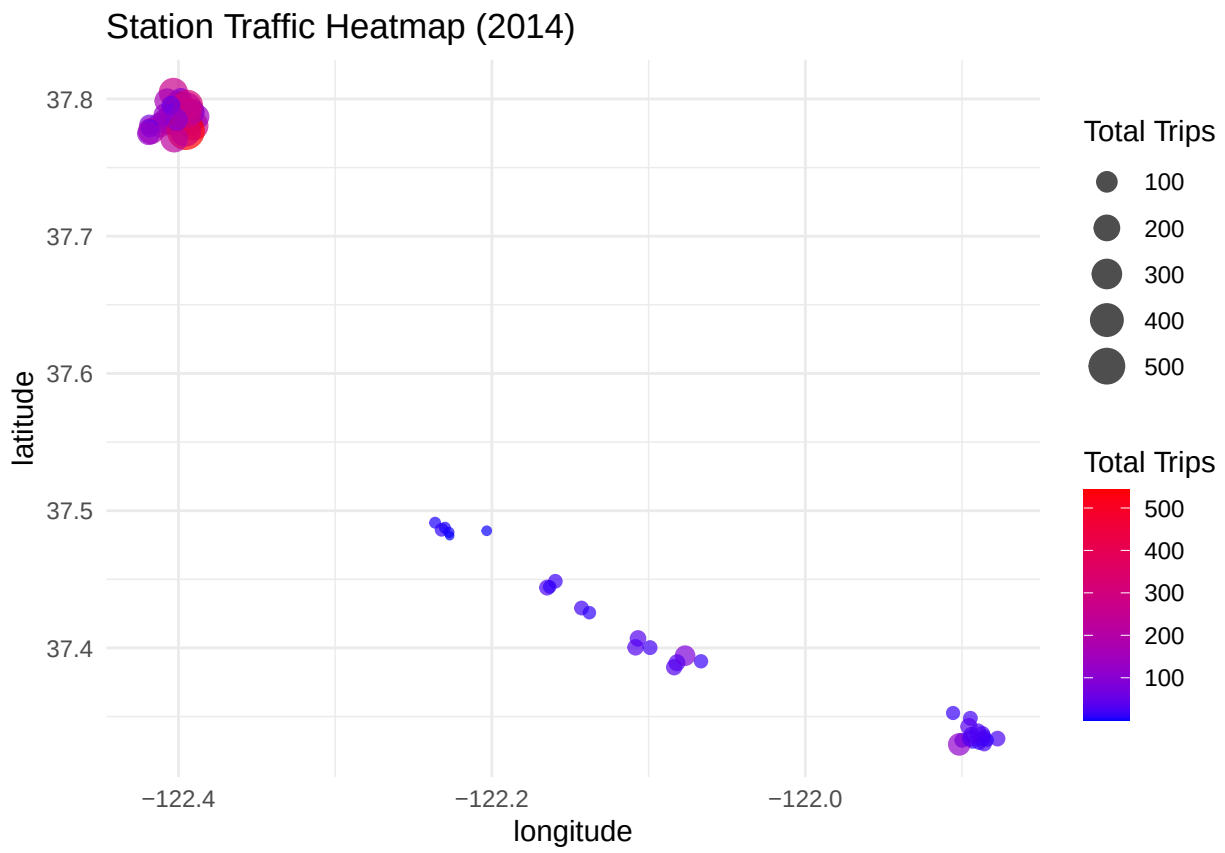
```
station_traffic_2014 <- july_2014 %>%
  group_by(station_name, latitude, longitude) %>%
  summarize(total_trips = n())
```

```
## `summarise()` has grouped output by 'station_name', 'latitude'. You can
## override using the `.groups` argument.
```

```
station_traffic_2015 <- july_2015 %>%
  group_by(station_name, latitude, longitude) %>%
  summarize(total_trips = n())
```

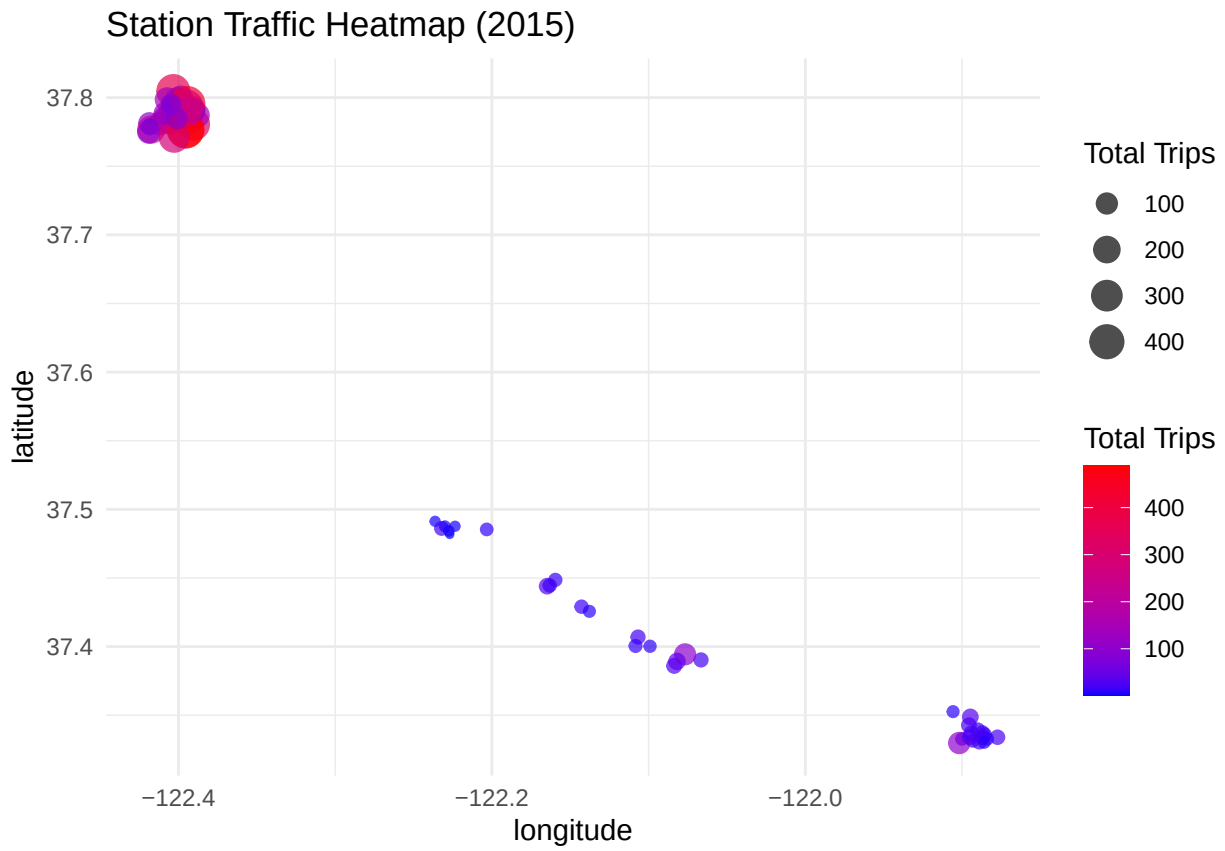
```
## `summarise()` has grouped output by 'station_name', 'latitude'. You can
## override using the `.groups` argument.
```

```
# Heatmap for 2014
ggplot(station_traffic_2014, aes(x = longitude, y = latitude)) +
  geom_point(aes(size = total_trips, color = total_trips), alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Station Traffic Heatmap (2014)", size = "Total Trips", color = "Total Trips") +
  theme_minimal()
```




```
# Heatmap for 2015
ggplot(station_traffic_2015, aes(x = longitude, y = latitude)) +
```

```
  geom_point(aes(size = total_trips, color = total_trips), alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Station Traffic Heatmap (2015)", size = "Total Trips", color = "Total Trips") +
  theme_minimal()
```



Code Summary

- Grouped trips by station latitude and longitude to calculate total trips.
- Visualized heatmaps using gradient scales to highlight traffic intensity.

2. Top Routes Bar Charts

Bar charts highlight the 10 most popular routes in 2014 and 2015, showing consistent commuter preferences.

```
# Select top 10 routes for 2014 and 2015
top_routes_2014 <- head(route_popularity_2014, 10)
top_routes_2015 <- head(route_popularity_2015, 10)

# Plot for 2014
ggplot(top_routes_2014, aes(x = reorder(paste(start_station_name, end_station_name, sep = " -> "), -tota
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
```

```
  labs(title = "Top 10 Routes (2014)", x = "Route", y = "Total Trips") +
  theme_minimal()
```


```
# Plot for 2015
ggplot(top_routes_2015, aes(x = reorder(paste(start_station_name, end_station_name, sep = " -> "), -tota
  geom_bar(stat = "identity", fill = "darkorange") +
  coord_flip() +
  labs(title = "Top 10 Routes (2015)", x = "Route", y = "Total Trips") +
  theme_minimal()
```

Code Summary

- Extracted top 10 routes for each year.
- Visualized routes using horizontal bar charts, with unique colors for each year.

3. Daily Usage Trends

Line charts compare daily trip totals for July 2014 and July 2015, identifying consistent patterns or significant shifts.

```
library(dplyr)
library(ggplot2)

# Ensure start_date is in the correct Date format
july_2014 <- july_2014 %>%
  mutate(start_date = as.Date(start_date, format = "%Y-%m-%d"))

july_2015 <- july_2015 %>%
  mutate(start_date = as.Date(start_date, format = "%Y-%m-%d"))

# Prepare daily usage data
daily_trends_2014 <- july_2014 %>%
  group_by(start_date) %>%
```

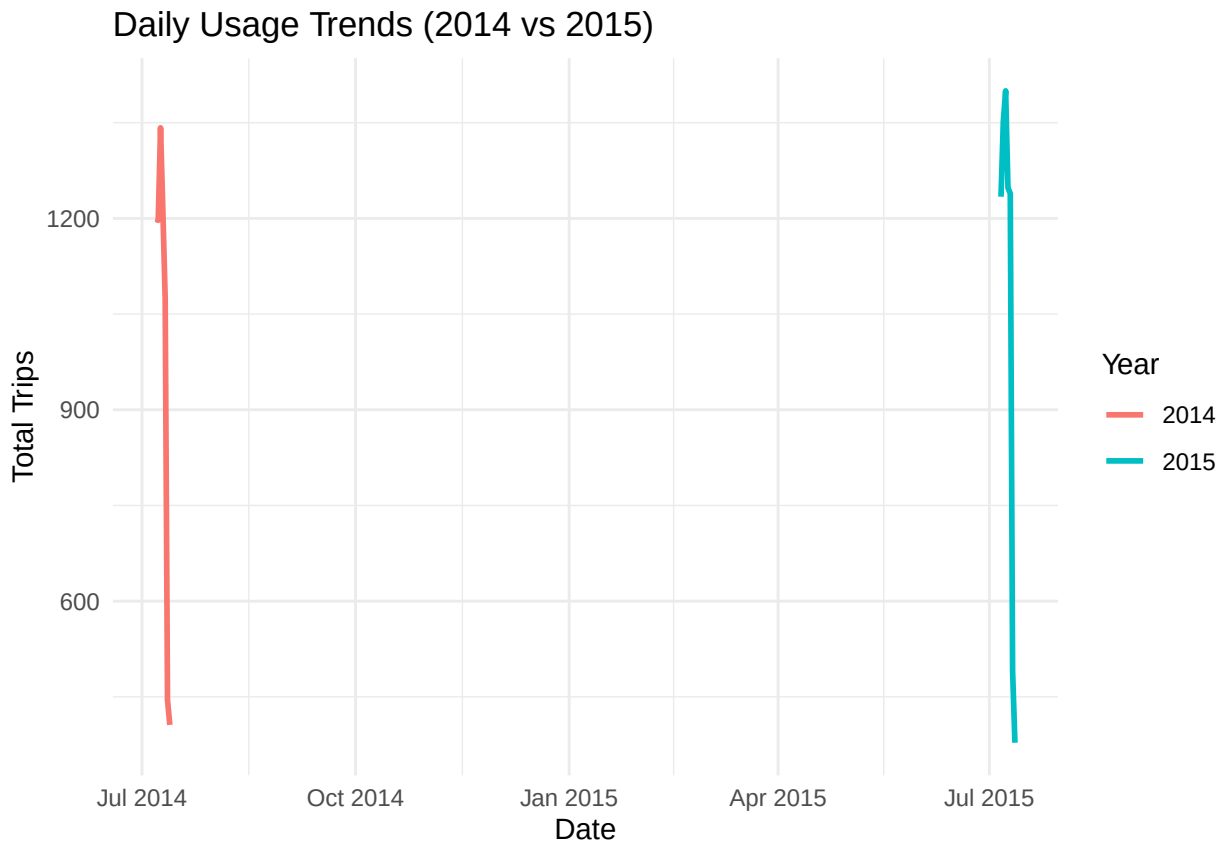
```r
  summarize(total_trips = n())

daily_trends_2015 <- july_2015 %>%
  group_by(start_date) %>%
  summarize(total_trips = n())

# Combine data for comparison
daily_trends <- bind_rows(
  daily_trends_2014 %>% mutate(year = "2014"),
  daily_trends_2015 %>% mutate(year = "2015")
)

# Plot daily trends
ggplot(daily_trends, aes(x = start_date, y = total_trips, color = year)) +
  geom_line(size = 1) +
  labs(title = "Daily Usage Trends (2014 vs 2015)", x = "Date", y = "Total Trips", color = "Year") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



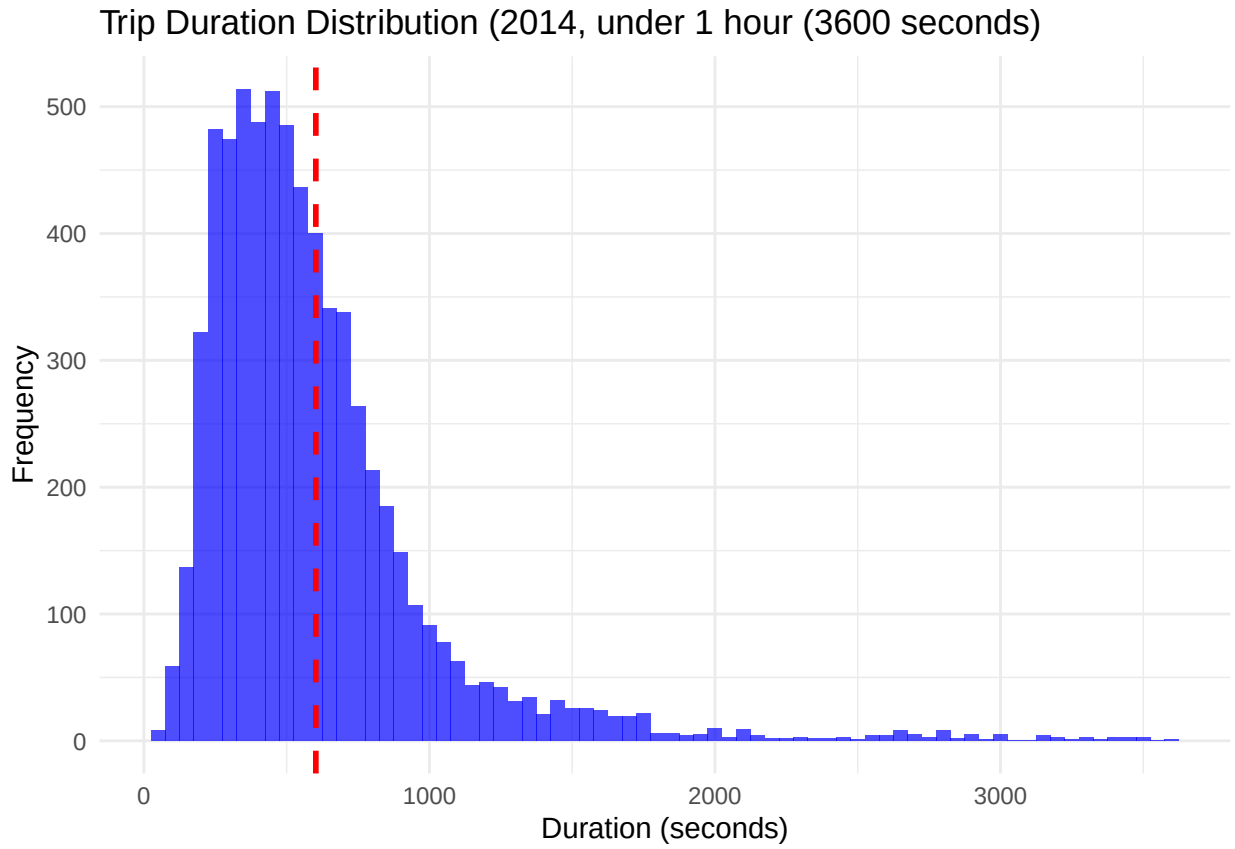
Code Summary

- Grouped trips by day to calculate daily totals.
- Combined data for side-by-side comparison.
- Plotted trends using line charts, color-coded by year

4. Trip Duration Histogram

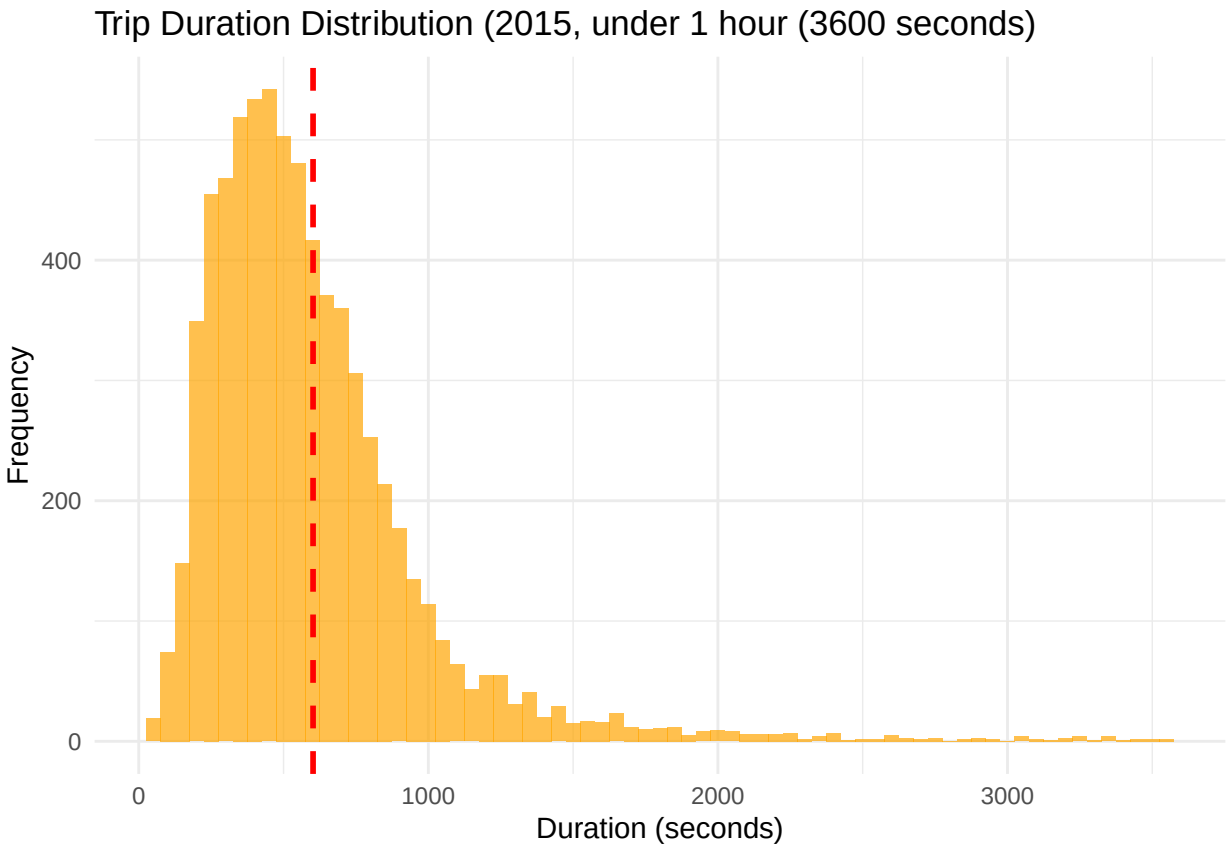
Histograms reveal the distribution of trip durations under 1 hour, highlighting common usage patterns.

```
# Histogram for 2014
ggplot(filtered_2014, aes(x = duration)) +
  geom_histogram(binwidth = 50, fill = "blue", alpha = 0.7) +
  geom_vline(aes(xintercept = mean(duration)), color = "red", linetype = "dashed", size = 1) +
  labs(
    title = "Trip Duration Distribution (2014, under 1 hour (3600 seconds)",
    x = "Duration (seconds)",
    y = "Frequency"
  ) +
  theme_minimal()
```




```
# Histogram for 2015
ggplot(filtered_2015, aes(x = duration)) +
  geom_histogram(binwidth = 50, fill = "orange", alpha = 0.7) +
  geom_vline(aes(xintercept = mean(duration)), color = "red", linetype = "dashed", size = 1) +
```

```
  labs(
    title = "Trip Duration Distribution (2015, under 1 hour (3600 seconds)",
    x = "Duration (seconds)",
    y = "Frequency"
  ) +
  theme_minimal()
```



Code Summary

- Filtered trips to exclude durations over 1 hour.
- Visualized distributions using histograms with mean markers for each year.

Insights :

- **Key Hubs**: Stations like Ferry Building and Embarcadero at Sansome are essential for connectivity and see the highest traffic.
- **Stable Routes**: Routes like Ferry Building to Embarcadero at Sansome remain popular across years.
- **Short Trips**: Most trips are under 1 hour, primarily for commuting, though outliers may indicate recreational usage.
- **Seasonal Trends**: While summer data was analyzed, higher traffic at tourist-heavy stations suggests significant seasonal effects.

- **Evolving Network**: Changes in route popularity between 2014 and 2015 highlight shifting commuter behavior.

Recommendations :

- **Expand Capacity**: Add docking stations at high-demand hubs to manage peak traffic.
- **Plan for Seasonality**: Collect data from other seasons to optimize service flexibility year-round.
- **Ensure Reliability**: Prioritize maintenance at critical stations to prevent disruptions.
- **Upgrade Emerging Routes**: Improve infrastructure on growing routes to support future demand.
- **Investigate Anomalies**: Address long-duration trips for data accuracy and consider tailored services for casual users.

Conclusion

This analysis provides a comprehensive view of the San Francisco Bike Share network, identifying key trends, critical stations, and actionable recommendations. Addressing these insights will enhance user satisfaction, increase commuter efficiency, and support long-term growth as a sustainable urban mobility solution. By prioritizing critical hubs, upgrading infrastructure, and planning for seasonality, the network can continue to evolve and thrive.