Exploratory Data Analysis of Uber Trips on Easter 2014 (NYC)

Aryaman Asthana (First-Year) | University of California, Berkeley

Project Description:

Using raw data to visualize the distribution of Uber trips in NYC on Easter 2014. Various aesthetic mappings, settings, and geometries, informed by Leland Wilkinson’s Grammar of Graphics framework, were used to capture the relationship among variables in the dataset.

Credits:

Project inspired by and modeled after Uber dataset analysis of Github user **harshr28**: https://github.com/harshr28/ubereda/tree/main

Dataset .csv file (**harshr28**): uber-raw-data-apr14.csv

apr <- read.csv(“https://raw.githubusercontent.com/harshr28/ubereda/main/uber-raw-data-apr14.csv”)

Project Layout:

Coded in R-Studio project directory* (stat20.datahub.berkeley.edu), and transferred to GitHub file Worktime: ~18 hours

Observational unit: one Uber trip in NYC on Easter 2014

Variables/vectors within original **apr** dataframe –

Date/Time : The date and time of the Uber pickup

Lat : the latitude of the pickup location

Lon : the longitude of the pickup location

Base : the TLC base code associated with the driver

*If code runs off in this file, please reference the markdown document.

Data

```r
library(tidyverse)
```

```
── Attaching core tidyverse packages ────────────────────────
tidyverse 2.0.0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.4     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2
── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>)
to force all conflicts to become errors
```

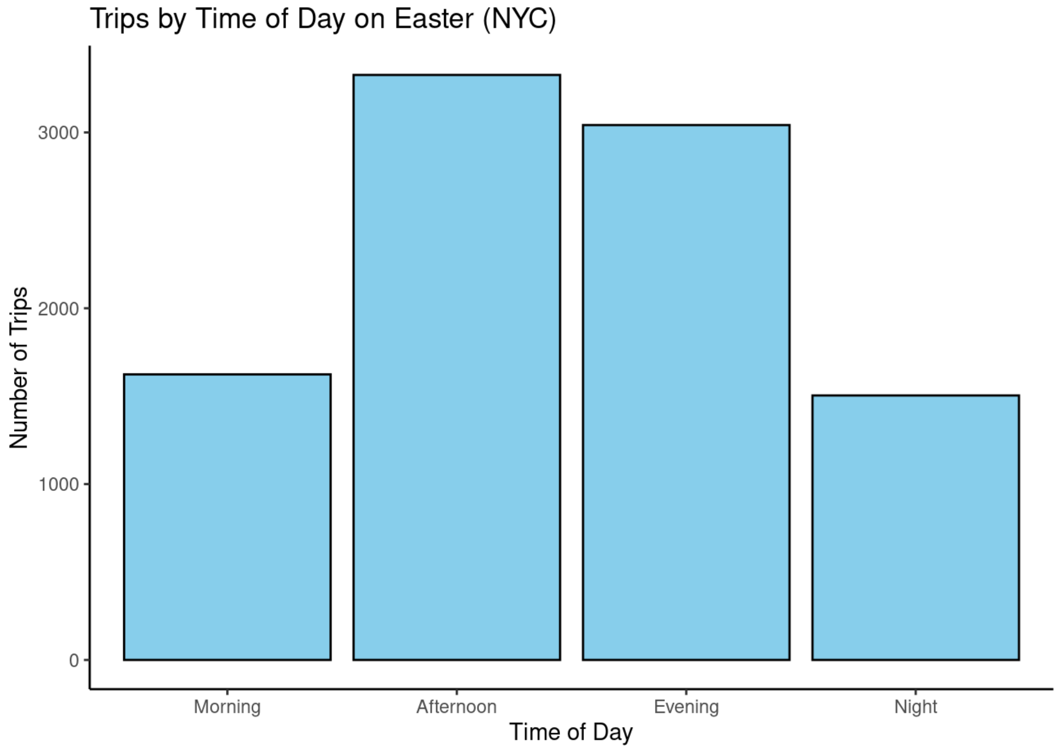
```r
library(ggplot2)
library(ggthemes)
library(dplyr)
library(readr)
library(patchwork)
apr <- read.csv("https://raw.githubusercontent.com/harshr28/ube
```

```r
new_df <- apr %>%
  filter(Date.Time > "4/20/2014 0:00:00",
         Date.Time < "4/20/2014 23:59:00")

splitted <- strsplit(as.character(new_df$Date.Time), " ")
part1 <- unlist(splitted)[2*(1:length(new_df$Date.Time))-1]
part2 <- unlist(splitted)[2*(1:length(new_df$Date.Time))]

Day <- c(part1)
Time <- c(part2)
new_df <- data.frame(Day,
                     Time,
                     new_df$Lat,
                     new_df$Lon,
                     new_df$Base)

new_df <- arrange(new_df, Time)
index_vector <- c(1:536,
                  6930:7362,
                  9223:9497,
                  537:1624,
                  1625:4951,
                  4952:6929,
                  7363:7993,
                  7994:9222)

first <- rbind(new_df[1:536, ], new_df[6930:7362, ])
second <- rbind(first, new_df[9223:9497, ])
third <- rbind(second, new_df[537:1624, ])
fourth <- rbind(third, new_df[1625:4951, ])
fifth <- rbind(fourth, new_df[4952:6929, ])
sixth <- rbind(fifth, new_df[7363:7993, ])
new_df <- rbind(sixth, new_df[7994:9222, ])

new_df <- new_df %>%
  mutate(Index = c(1:9497),
         time_category = fct_collapse(as.character(Index),
                                  "Morning" = 1:1624,
                                  "Afternoon" = 1625:4951,
                                  "Evening" = 4952:7993,
                                  "Night" = 7994:9497))

new_df$time_category <- factor(new_df$time_category,
                         levels = c("Morning",
                                    "Afternoon",
                                    "Evening",
                                    "Night"))

new_df <- new_df %>%
  mutate(long_west = ifelse(new_df.Lat > 40.6 &
                              new_df.Lat < 40.8 &
                              new_df.Lon < -73.8 &
                              new_df.Lon > -74.05,
                            TRUE,
                            FALSE))
```

Results:

The highest number of Uber trips on Easter 2014 in NYC were taken in the afternoon (noon - 4:59 pm).

```
tod_group <- new_df %>%
  ggplot(aes(x = time_category)) +
  geom_bar(fill = "skyblue",
           color = "black") +
  labs(x = "Time of Day",
       y = "Number of Trips",
       title = "Trips by Time of Day on Easter (NYC)") +
  theme_classic()
tod_group
```

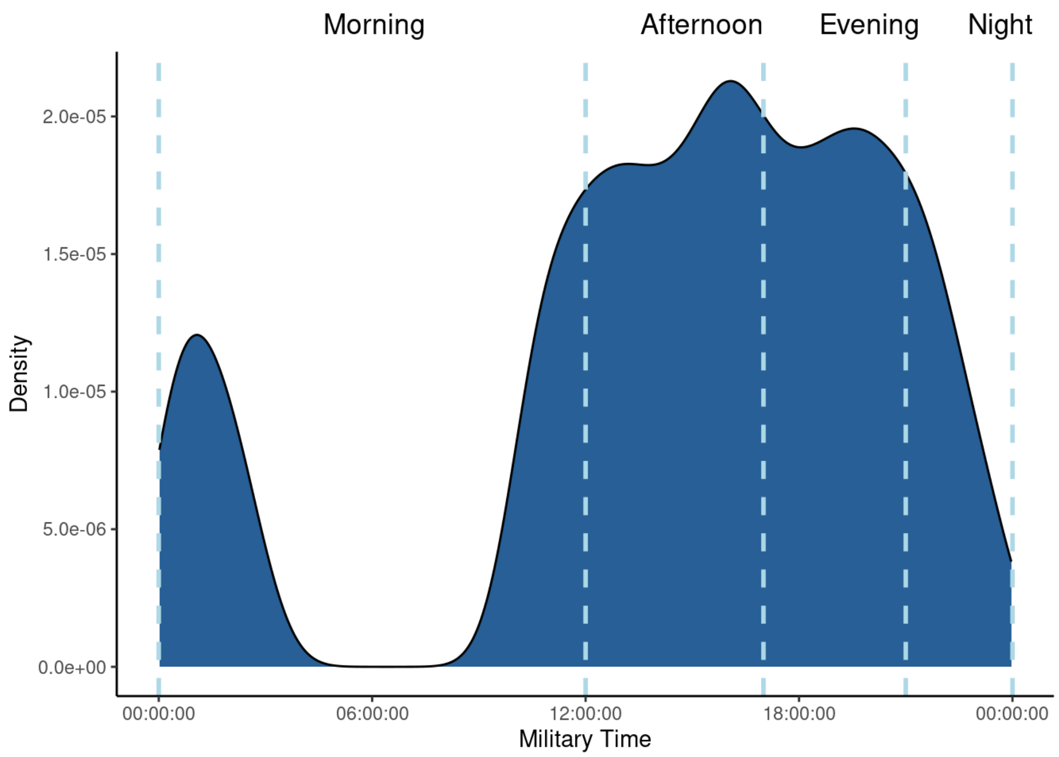


Elaborating on the chart above, this distribution portrays the density (defined as a histogram with an infinite number of bins) of Uber trips throughout Easter 2014 in NYC, grouped by a visual annotation of the time of day. This plot corroborates the description of the previous, while also showing that driver/rider activity peaking around 4:00 pm and little to no activity was observed at around 6:00 am.

```
annotate_1 <- as.POSIXct("00:00:00",
                         format = "%H:%M:%S")
annotate_2 <- as.POSIXct("12:00:00",
                         format = "%H:%M:%S")
annotate_3 <- as.POSIXct("17:00:00",
                         format = "%H:%M:%S")
annotate_4 <- as.POSIXct("21:00:00",
                         format = "%H:%M:%S")
annotate_5 <- as.POSIXct("24:00:00",
                         format = "%H:%M:%S")

time_density <- new_df %>%
  mutate(date_time_format = as.POSIXct(new_df$Time,
                                       format = "%H:%M:%S")) %>
  ggplot(aes(x = date_time_format)) +
  geom_density(fill = "dodgerblue4",
               alpha = 0.9) +
  scale_x_datetime(labels = scales::time_format("%H:%M:%S")) +
  labs(x = "Military Time",
       y = "Density",
       title = "                          Morning
  geom_vline(xintercept = as.numeric(annotate_1),
             linetype = "dashed",
             color = "lightblue",
             size = 1) +
  geom_vline(xintercept = as.numeric(annotate_2),
             linetype = "dashed",
             color = "lightblue",
             size = 1) +
  geom_vline(xintercept = as.numeric(annotate_3),
             linetype = "dashed",
             color = "lightblue",
             size = 1) +
  geom_vline(xintercept = as.numeric(annotate_4),
             linetype = "dashed",
             color = "lightblue",
             size = 1) +
  geom_vline(xintercept = as.numeric(annotate_5),
             linetype = "dashed",
             color = "lightblue",
             size = 1) +
  theme_classic()
```

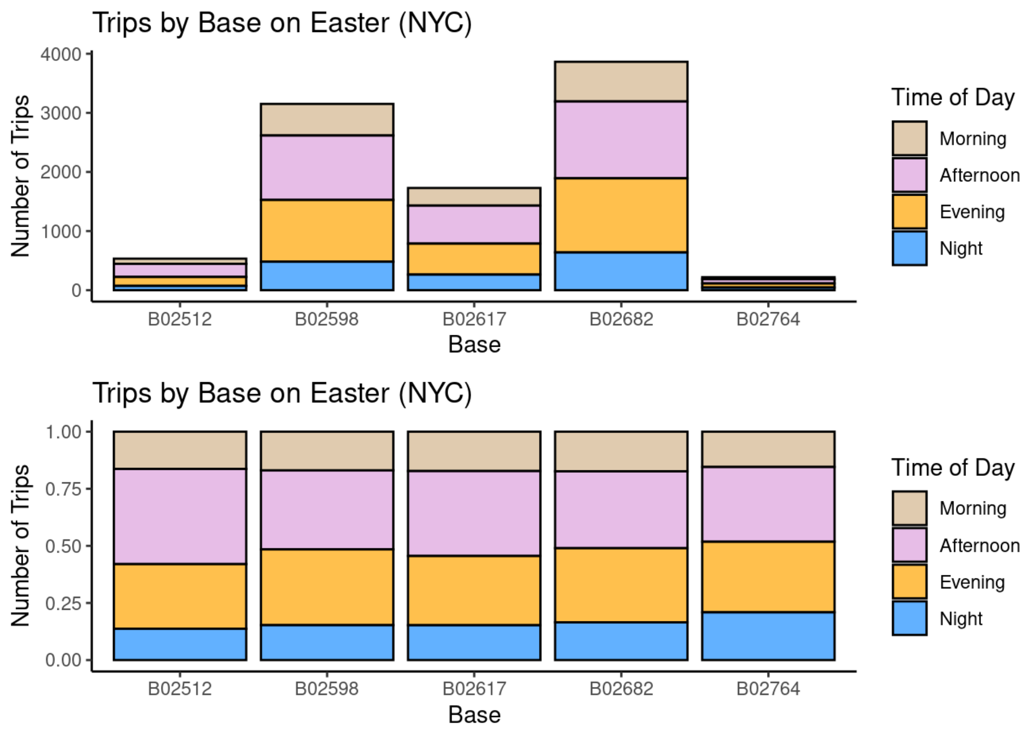
```
Warning: Using `size` aesthetic for lines was deprecated in
ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```

```
time_density
```



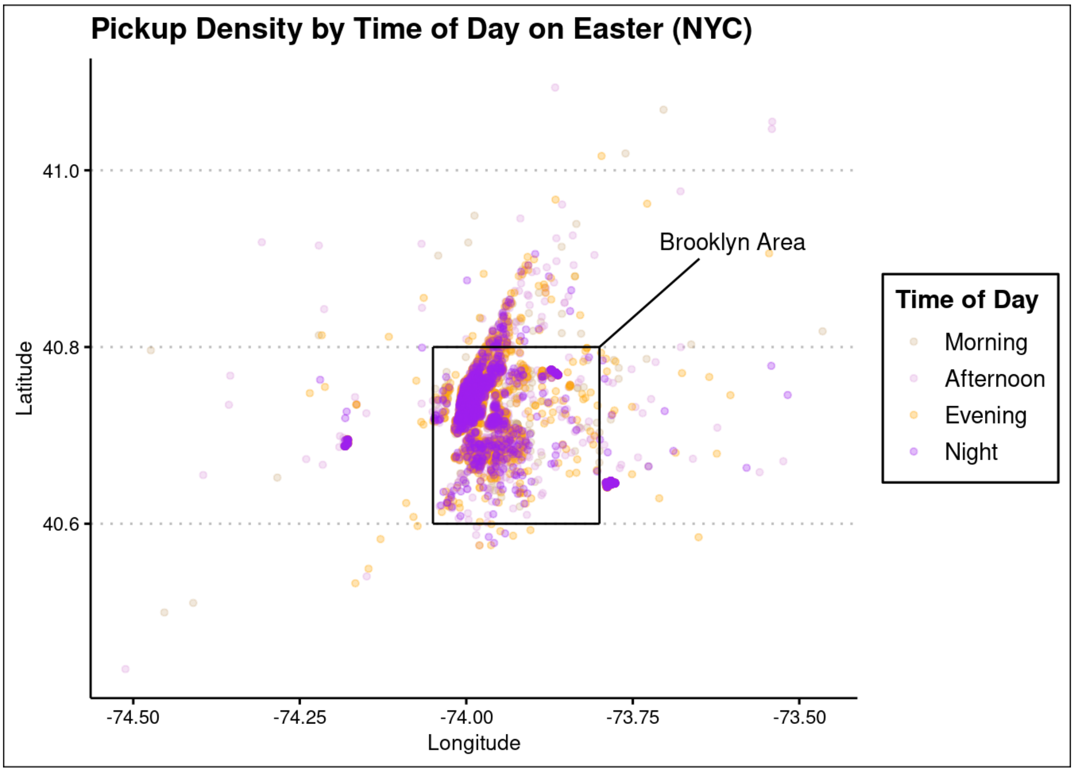
The first distribution represents the relationship between base and the number of trips for each time of day on Easter 2014 in NYC. Shown below is that the drivers with base code B02682 recorded the most number of trips, followed by those with base code B02598. It also shows the counts of the trips for each time of day per level of the base variable, but individual proportions may be of more relevance. In the second distribution, a stacked normalized bar chart is utilized to display the proportion and potential associations between time of day and the conditioned variable, base. There appears to be a weak association between what time of day Uber trips take place and the base code associated with the respective drivers; there is little to no variation of in the conditional proportions of the time of day as you move across the levels of the base variable.

```
bases_group <- new_df %>%
  ggplot(aes(x = new_df.Base,
             fill = time_category)) +
  geom_bar(color = "black",
           alpha = 0.7) +
  labs(x = "Base",
       y = "Number of Trips",
       title = "Trips by Base on Easter (NYC)") +
  scale_fill_manual(name = "Time of Day",
                    values = c("tan",
                               "plum",
                               "orange",
                               "dodgerblue")) +
  theme_classic()

bases_assoc <-  new_df %>%
  ggplot(aes(x = new_df.Base,
             fill = time_category)) +
  geom_bar(color = "black",
           position = "fill",
           alpha = 0.7) +
  labs(x = "Base",
       y = "Number of Trips",
       title = "Trips by Base on Easter (NYC)") +
  scale_fill_manual(name = "Time of Day",
                    values = c("tan",
                               "plum",
                               "orange",
                               "dodgerblue")) +
  theme_classic()
bases_group / bases_assoc
```



In the distribution below, the lattitude and longitude for each Uber trip for a given time of day on Easter 2014 in NYC were plotted. To gauge an understanding of the density of pickups in the Brooklyn area (where Uber drivers tend to be congested), a rectangular annotation was used to denote an estimate of the lattitude and longitude intervals of Upper Brooklyn.

```
max_lat <- max(new_df$new_df.Lat)
min_lat <- min(new_df$new_df.Lat)
max_lon <- max(new_df$new_df.Lon)
min_lon <- min(new_df$new_df.Lon)

location_group <- new_df %>%
  ggplot(aes(x = new_df.Lon,
             y = new_df.Lat,
             color = time_category)) +
  geom_point(alpha = .3,
             size = 1.2) +
  labs(x = "Longitude",
       y = "Latitude",
       title = "Pickup Density by Time of Day on Easter (NYC)")
  scale_color_manual(name = "Time of Day",
                     values = c("tan",
                                "plum",
                                "orange",
                                "purple")) +
  scale_x_continuous(limits = c(min_lon,
                                max_lon)) +
  scale_y_continuous(limits = c(min_lat,
                                max_lat)) +
  annotate("segment", x = -74.05,
           xend = -74.05,
           y = 40.8,
           yend = 40.6) +
  annotate("segment",
           x = -74.05,
           xend = -73.8,
           y = 40.6,
           yend = 40.6) +
  annotate("segment",
           x = -73.8,
           xend = -73.8,
           y = 40.6,
           yend = 40.8) +
  annotate("segment",
           x = -73.8,
           xend = -74.05,
           y = 40.8,
           yend = 40.8) +
  annotate("segment",
           x = -73.8,
           xend = -73.65,
           y = 40.8,
           yend = 40.9) +
  annotate("text",
           x = -73.6,
           y = 40.92,
           label = "Brooklyn Area") +
  theme_clean()
location_group
```

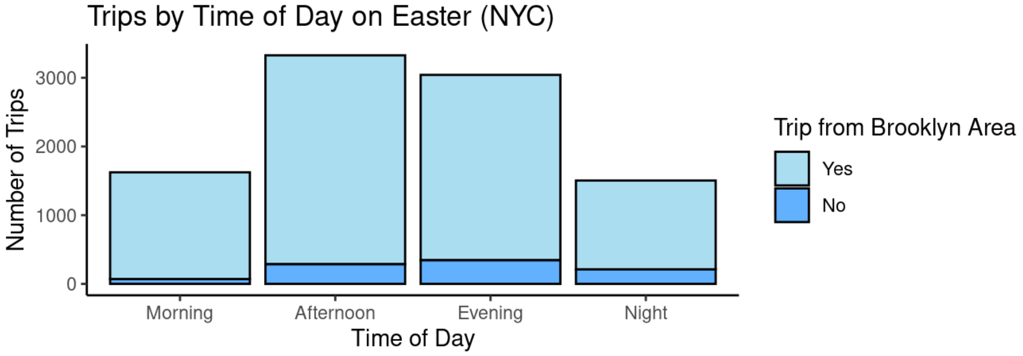


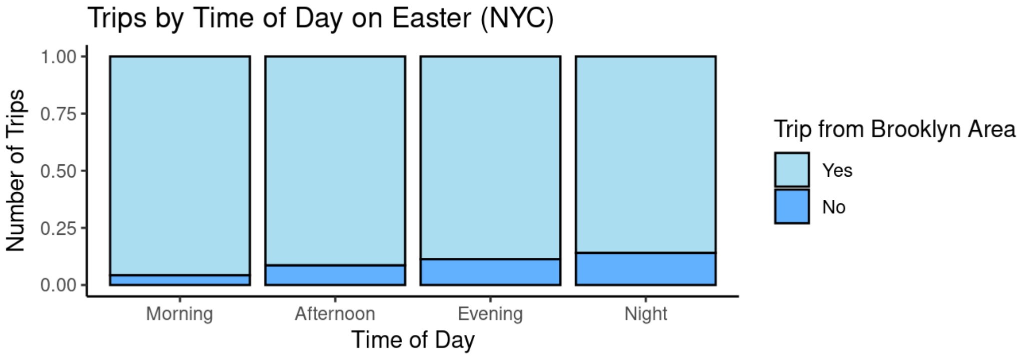
The distributions below mimics the same statistical principle as the third set of distributions above. It represents the relationship between the number of trips for each time of the day and whether or not the trip was taken from Upper Brooklyn on Easter 2014 in NYC. While the first distribution displays counts, we are more interested in the respective proportions. In the second distribution, it appears that throughout Easter the proportion of trips that were taken from Upper Brooklyn for a given time period increased. This can be explained by the fact that a greater area/scope of riders search for cabs at night due to safety concerns.

```
prop_values <- new_df %>%
  group_by(time_category) %>%
  summarize(prop = mean(long_west))

prop_group <- new_df %>%
  ggplot(aes(x = time_category,
             fill = factor(long_west,
                           levels = c(TRUE,
                                      FALSE)))) +
  geom_bar(color = "black",
           alpha = 0.7) +
  labs(x = "Time of Day",
       y = "Number of Trips",
       title = "Trips by Time of Day on Easter (NYC)") +
  scale_fill_manual(name = "Trip from Brooklyn Area",
                    labels = c("Yes",
                               "No"),
                    values = c("skyblue",
                               "dodgerblue")) +
  theme_classic()

prop_assoc <- new_df %>%
  ggplot(aes(x = time_category,
             fill = factor(long_west,
                           levels = c(TRUE, FALSE)))) +
  geom_bar(position = "fill",
           color = "black",
           alpha = 0.7) +
  labs(x = "Time of Day",
       y = "Number of Trips",
       title = "Trips by Time of Day on Easter (NYC)") +
  scale_fill_manual(name = "Trip from Brooklyn Area",
                    labels = c("Yes",
                               "No"),
                    values = c("skyblue",
                               "dodgerblue")) +
  theme_classic()
prop_values
```

```
# A tibble: 4 × 2
  time_category  prop
  <fct>         <dbl>
1 Morning       0.958
2 Afternoon     0.914
3 Evening       0.887
4 Night         0.860
```

```
prop_group / prop_assoc
```





How to Install and Run Project:

Specific instructions for installing packages and loading libraries can be found on the markdown document. Explanations and references for each relevant code section are also provided.

How to Use/Render this Project:

Should a user choose to, they may copy and paste code/code snippets into an R .qmd, markdown, or script file with the "html" format as seen in this document above. That file can then be rendered or repurposed to present a new analysis of the data. However, please cite this project according to the creative commons license stipulations below: