Weather and Uber & Lyft Ridership

John Cruz

2023-04-25

Introduction

The National Oceanic and Atmospheric Administration ('NOAA') defines the heat index as the apparent temperature of what the temperature feels like to the human body when relative humidity is combined with the air temperature. This has important considerations for the human body's comfort. When the body gets too hot, it begins to perspire or sweat to cool itself off.

As for the New York City subway system during the summer, it is notoriously known to have unbearable temperatures where the platform can be 104 degrees, compared to 86 degrees outside ('Curbed NY').

Given the health risks, and general discomfort during high heat index days, this project will look into alternative modes of transportation, particularly ridesharing companies such as Uber and Lyft.

Research question

Does high heat index days (>=90 degrees) increase the number of trips taken with Uber or Lyft compared to non-high heat index days?

Data Source

Weather (Oikolab)

Data was collected using Oikolab API historical data API service. It collects its data from the ECWMF and NOAA. Each case represents hourly weather measurements in from 2021-2022.

Uber & Lyft Trips (NYC Taxi and Limousine Commission)

Data was collected using the available 'parquet files'. The agency collects the data from Uber and Lyft. Each case represents a trip taken either via Uber or Lyft between 2021-2022.

Type of study

This is an observational study.

Variables

Dependent - total trips: numerical

Independent Variable(s) The independent variables are:

- type_of_day: categorical
- precipitation: numerical
- day_of_week: categorical
- month: categorical
- year: categorical

Note: Other potential factors that are important but not included: special events (i.e. sporting event), major delays with public transportation (MTA Subway) or alternative transportation such as Citi bikes.

Required Libraries

```
library(tidyverse)
library(arrow)
library(DBI)
library(lubridate)
library(weathermetrics)
library(infer)
library(psych)
```

Data Preparation

Load Historical Weather Data

Calculate Heat Index - Relative humidity is calculated using the temperature and dewpoint temperature. - Heat index is calculated using the temperature and relative humidity.

```
weather <- read_csv('data//oikolabs.csv') |>
  janitor::clean_names()

weather <-
  weather |>
  mutate(temp_deg_f = celsius.to.fahrenheit(temperature_deg_c),
         rel_humidity = dewpoint.to.humidity(t = temperature_deg_c,
                                             dp = dewpoint_temperature_deg_c,
                                             temperature.metric = "celsius"),
         heat_idx = heat.index(t = temp_deg_f,
                               rh = rel_humidity),
         total_precipitation_mm_of_water_equivalent = total_precipitation_mm_of_water_equivalent / 25.4)
  rename(total_precip = total_precipitation_mm_of_water_equivalent)
```

```r
weather <-
  weather |>
  mutate(day_of_week = wday(datetime_utc, label = TRUE, week_start = 1, abbr = FALSE),
         day_of_week = as.factor(day_of_week),
         datetime_ny = with_tz(datetime_utc, "America/New_York")) |>
  relocate(datetime_ny) |>
  select(datetime_ny, temp_deg_f, rel_humidity, heat_idx, total_precip) |>
  filter(between(datetime_ny, as.Date("2021-01-01"),
         as.Date("2022-12-31")))

knitr::kable(head(weather), caption = 'Weather Data')
```

Table 1: Weather Data

datetime_ny	temp_deg_f	rel_humidity	heat_idx	total_precip
2021-01-01 00:00:00	32.63	73.61101	33	0
2021-01-01 01:00:00	31.35	75.52820	31	0
2021-01-01 02:00:00	31.05	76.68467	31	0
2021-01-01 03:00:00	30.51	78.82627	31	0
2021-01-01 04:00:00	30.58	79.00774	31	0
2021-01-01 05:00:00	25.43	87.89063	25	0

Load Uber and Lyft Trips

The NYC Taxi and Limousine Commission provides a data dictionary 'here'. The rideshare app companies such as Uber is coded as (HV0003) and Lyft (HV0005).

- Trips were filtered because of huge outliers that were present such as:
  - Trip time had to be >0 seconds and <= 5 hours.
  - Trip miles had to be >= 0.
  - Driver pay > $0.01.
  - Base passenger fare > $0.01.
  - Pickup locations had to be within the NYC region and not unknown/outside of it.

Given the large amounts of data to be processed, some of the data cleaning and filtering was done through DuckDB. DuckDB contains columnar-vectorized query execution engine where it allows for memory resources not to be severely depleted while trying to aggregate through the data. For more information visit 'DuckDB'

```r
cnxn = dbConnect(duckdb::duckdb(), dbdir=":memory:")

dbExecute(cnxn, "CREATE VIEW tlc_trips AS
                      SELECT
                             CASE
                                 WHEN hvfhs_license_num == 'HV0003' THEN 'Uber'
                                 WHEN hvfhs_license_num == 'HV0005' THEN 'Lyft'
                                 ELSE 'Other'
                             END AS app,
                             pickup_datetime,
                             dropoff_datetime,
                             PULocationID,
```

```
                             trip_miles,
                             trip_time,
                             base_passenger_fare
                      FROM 'data\\*.parquet'
                      WHERE
                          trip_time >= 0 AND
                          trip_time < 18000 AND
                          trip_miles >= 0 AND
                          driver_pay > 0.01 AND
                          base_passenger_fare > 0.01 AND
                          PULocationID NOT IN (264, 265)"
          )
```

```
## [1] 0
```

```
query <- "WITH floor_date AS(
                SELECT
                  app,
                  time_bucket(interval '1 hour', pickup_datetime) AS pickup_datetime,
                  PULocationID,
                  trip_miles,
                  trip_time,
                  base_passenger_fare
                FROM tlc_trips
          )

          SELECT
            app,
            pickup_datetime,
            PULocationID,
            COUNT(*) as trips,
            SUM(trip_miles) AS trip_miles,
            SUM(trip_time) AS trip_time,
            SUM(base_passenger_fare) AS base_passenger_fare
          FROM floor_date
          GROUP BY app, pickup_datetime, PULocationID
         "

db_trips <- dbGetQuery(cnxn, query)

tlc_trips <-
  db_trips |>
  mutate(pickup_datetime = force_tz(pickup_datetime, tzone = 'America/New_York'))

tlc_trips <-
  tlc_trips |>
  group_by(pickup_datetime) |>
  summarise(total_trips = sum(trips),
            total_trip_dist = sum(trip_miles),
            total_trip_time = sum(trip_time),
            total_base_fare = sum(base_passenger_fare))

knitr::kable(head(tlc_trips), caption = 'Uber & Lyft Trips')
```

Table 2: Uber & Lyft Trips

pickup_datetime	total_trips	total_trip_dist	total_trip_time	total_base_fare
2021-01-01 00:00:00	30252	139167.71	26305805	512676.3
2021-01-01 01:00:00	35654	169601.16	31351009	700459.8
2021-01-01 02:00:00	33028	158558.99	28793246	639599.2
2021-01-01 03:00:00	26075	125819.15	22655554	452191.8
2021-01-01 04:00:00	16787	83757.50	14703813	314495.0
2021-01-01 05:00:00	12244	64789.97	10810829	270542.1

Merge Datasets

The data from both sources will be merged together based on the datetime columns. It is important to note that due to the large size of source files, a parquet file of the cleaned data will be exported as *trip_weather_data.parquet* and provided in the repo. For the original files, access them from the previously mentioned methods.

```
trip_weather_data <-
  tlc_trips |>
  left_join(weather, by = join_by(pickup_datetime == datetime_ny)) |>
  mutate(pickup_date = date(pickup_datetime)) |>
  select(!pickup_datetime) |>
  group_by(pickup_date) |>
  mutate(total_trips = sum(total_trips),
         trip_dist = sum(total_trip_dist),
         trip_time = sum(total_trip_time),
         base_fare = sum(total_base_fare),
         temp_deg_f = max(temp_deg_f),
         heat_idx = max(heat_idx),
         precip = sum(total_precip),
         .keep = "none") |>
  distinct() |>
  mutate(type_of_day = case_when(heat_idx >= 90 ~ 'hot',
                                 .default = 'normal'),
         day_of_week = factor(wday(pickup_date, label = TRUE, week_start = 1), ordered = FALSE),
         day_of_week = fct_relevel(day_of_week, "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"),
         month = month(pickup_date, label = TRUE),
         year = year(pickup_date)) |>
  drop_na()

trip_weather_data$day_of_week = relevel(trip_weather_data$day_of_week, ref='Mon')
trip_weather_data$type_of_day = relevel(factor(trip_weather_data$type_of_day, ordered = FALSE), ref='no
trip_weather_data$month = relevel(factor(trip_weather_data$month, ordered = FALSE), ref='Jan')
trip_weather_data$year = relevel(factor(trip_weather_data$year, ordered = FALSE), ref='2021')

write_parquet(trip_weather_data, "trip_weather_data.parquet")

knitr::kable(head(trip_weather_data))
```

total_trips	temp_deg	heat_idx	pickup_date	trip_dist	trip_time	base_fare	precip	type_of_day	day_of_week	month	year
403177	38.32	38	2021-01-01	1929395	354153513	7512445	0.3200787	normal	Fri	Jan	2021
329487	49.78	48	2021-01-02	1579836	313366813	5972419	0.1027559	normal	Sat	Jan	2021
297537	37.67	38	2021-01-03	1454448	267116540	5305180	0.2992126	normal	Sun	Jan	2021
317646	41.58	40	2021-01-04	1491053	304136151	5587092	0.0503937	normal	Mon	Jan	2021
333590	39.97	40	2021-01-05	1505182	320034529	5786943	0.0031496	normal	Tue	Jan	2021
352620	40.14	40	2021-01-06	1557586	339948316	6076668	0.0000000	normal	Wed	Jan	2021

Summary Statistics

Total Daily Trips

Here we can see the total number of daily trips taken via Uber and Lyft in NYC between 2021-2022. There is a seasonality trend occurring between the dates that we will look further into.

```
trip_weather_data |>
  ggplot(aes(x = pickup_date, y = total_trips)) +
  geom_line(stat = 'identity') +
  scale_y_continuous(labels = scales::comma) +
  theme_bw() +
  labs(title = "Total Daily Trips", caption = "Figure 1", x = '', y = '')
```

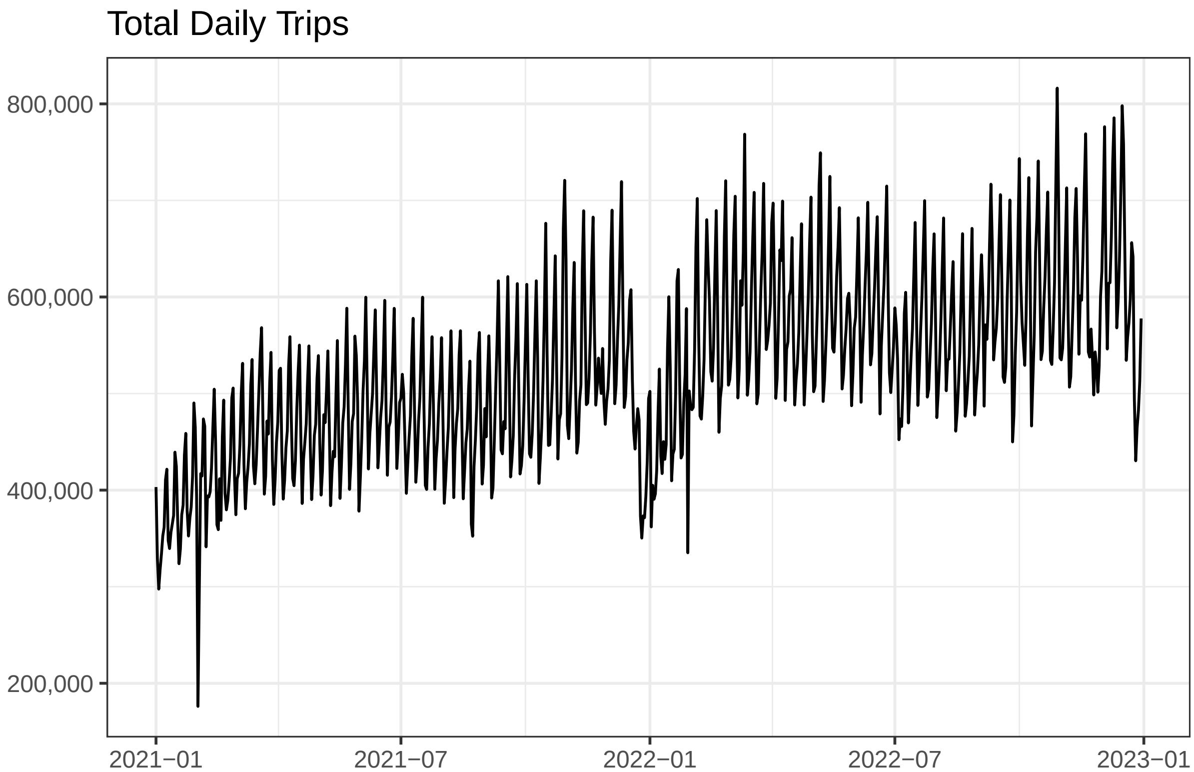

Figure 1

Day of the Week vs. Total Trips

Plotting each year's total trips and the day of the week, we do see a seasonal trend where on average Monday's have the lowest trip counts and it progressively increases until Sunday's drop.

```
ggplot(aes(x = day_of_week, y = total_trips), data = trip_weather_data) +
  geom_boxplot() +
  facet_grid(rows = vars(year(pickup_date))) +
  theme_bw() +
  labs(x = '', y = 'Total Trips', caption = 'Figure 2') +
  scale_y_continuous(labels = scales::comma)
```

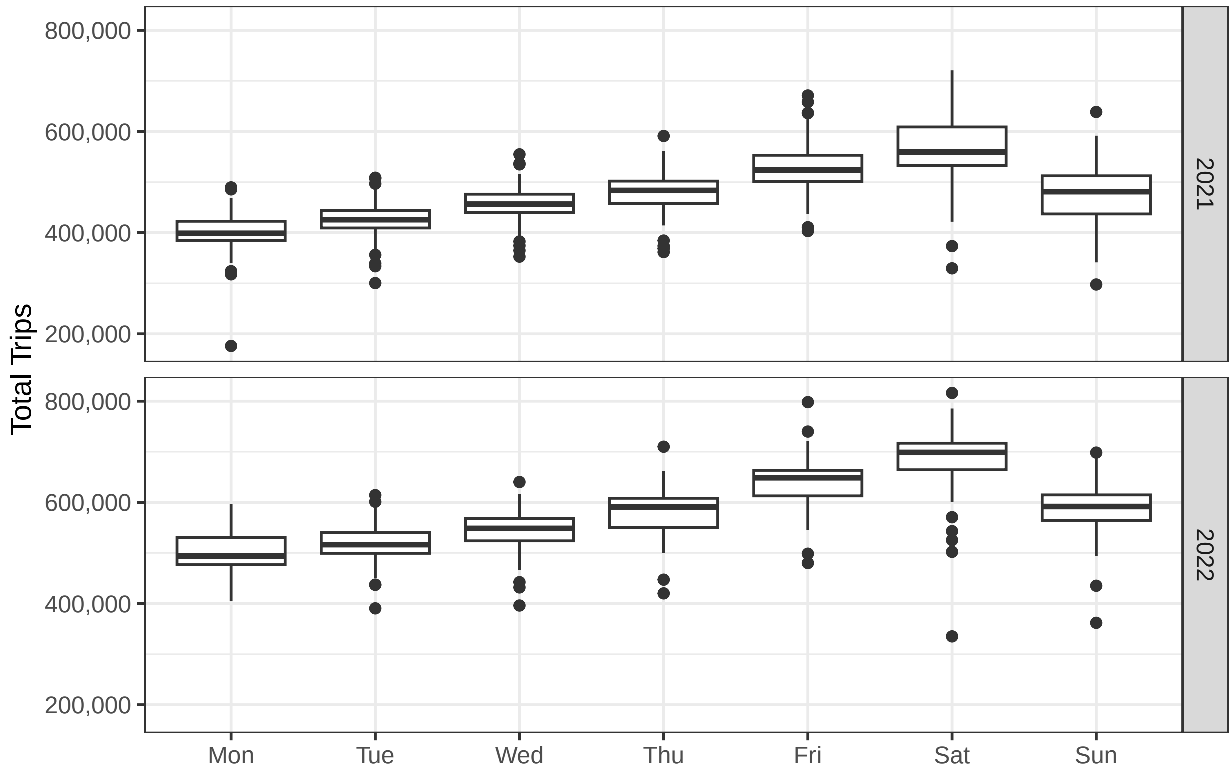

Figure 2

Daily Heat Index

Continuing looking into seasonal trends, we see both years have the same shape between the daily heat index and how it varies month to month. The high heat index months of interest is usually between late May until early August, which coincides with the summer months of NYC.

```
trip_weather_data |>
  ggplot(aes(x = month, y = heat_idx, color = type_of_day)) +
  geom_jitter(stat = 'identity') +
  facet_grid(rows = vars(year(pickup_date))) +
  theme_classic() +
  labs(x = '', y ='', title = "Daily Heat Index", caption = "Figure 3") +
  theme(legend.position = "top",
        legend.justification = c(0, 1)) +
  scale_color_manual(values = c('#4E79A7', '#F28E2B'), name = '')
```

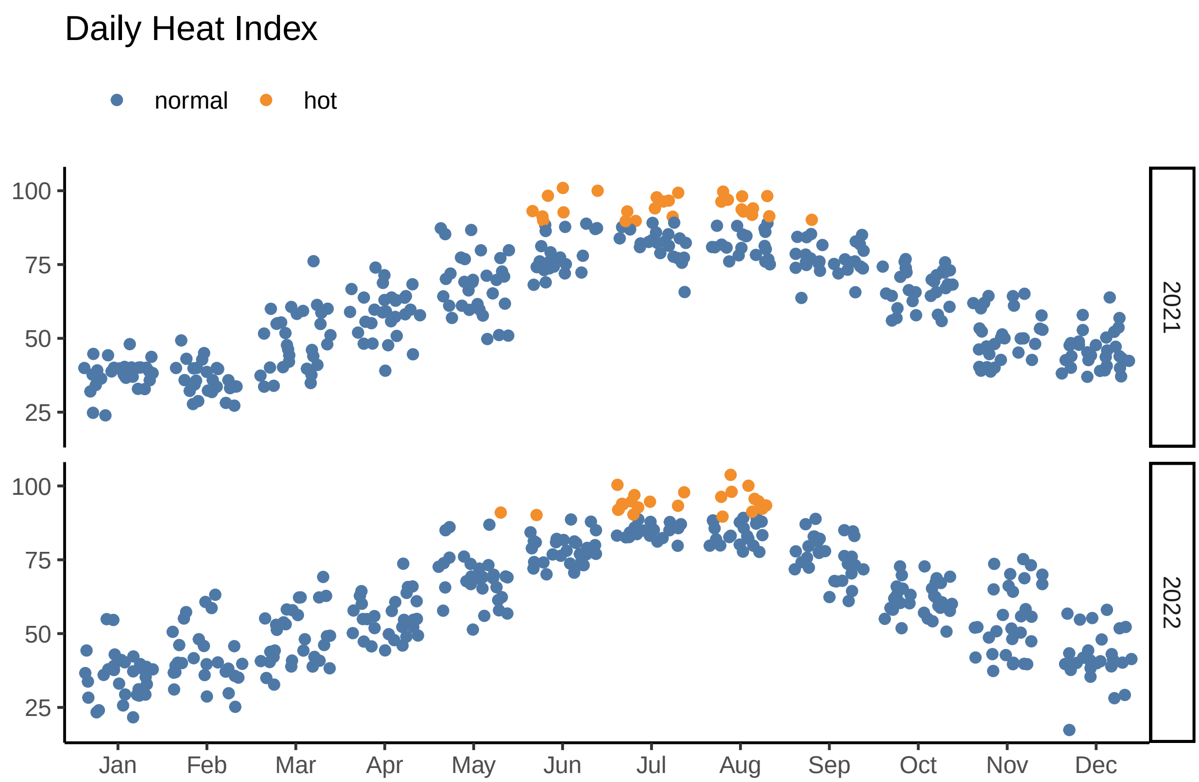

Figure 3

Heat Index vs. Daily Trips

Finally, let us look at the relationship of heat index to total number of trips. The scatter plot shows us that there is some bend initially as the heat index rises, but overall no strong trend. However, both years again follow the same seasonal patters.

```
trip_weather_data |>
  ggplot(aes(x = heat_idx, y = total_trips, color = year)) +
  geom_point(stat = 'identity') +
  labs(title = 'Heat Index vs. Total Trips', caption = "Figure 4", x = 'heat index', y = 'total trips')
  theme_classic() +
  theme(legend.position = "top",
        legend.justification = c(0, 1)) +
  scale_color_manual(values = c('#5F9ED1', '#C85200'), name = '') +
  scale_y_continuous(labels = scales::comma)
```

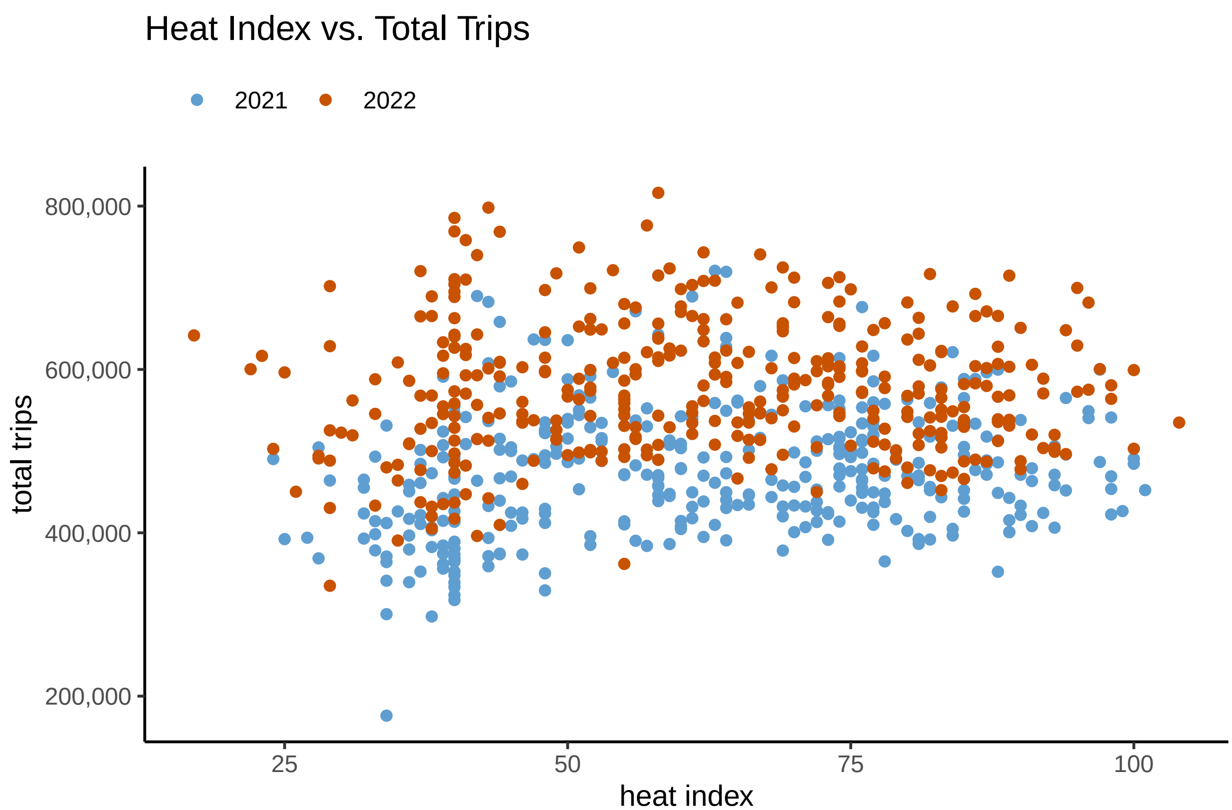

Figure 4

Observed Difference

Viewing the observed difference, we can see out of two years worth of data, there were 51 days where the heat index was >= 90 degrees. However, on average, the trips taken on hot days compared to normal days were slightly lower by about 6,000 trips.

```
comparison <-
  trip_weather_data |>
  group_by(type_of_day) |>
  summarise(avg_trips = mean(total_trips),
            n = n())

knitr::kable(comparison, caption = "Observed Differences")
```

Table 4: Observed Differences

type_of_day	avg_trips	n
normal	526154.8	678
hot	520064.5	51

Hypothesis Testing

To test the observed differences between both high heat index days and non-high heat index days, a hypothesis test will be performed. To account for the possible differences in the two year data, a bootstrap was performed to account for such variability.

```
set.seed(2000)
```

```
obs_diff <- trip_weather_data %>%
  specify(total_trips ~ type_of_day) %>%
  calculate(stat = "diff in means", order = c("hot", "normal"))
```

```
null_dist <- trip_weather_data %>%
  specify(total_trips ~ type_of_day) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("hot", "normal"))
```

We can see through the histogram distribution, that our data is normally distributed.

```
ggplot(data = null_dist, aes(x = stat)) +
  geom_histogram(bins = 30) +
  labs(caption = "Figure 5")
```

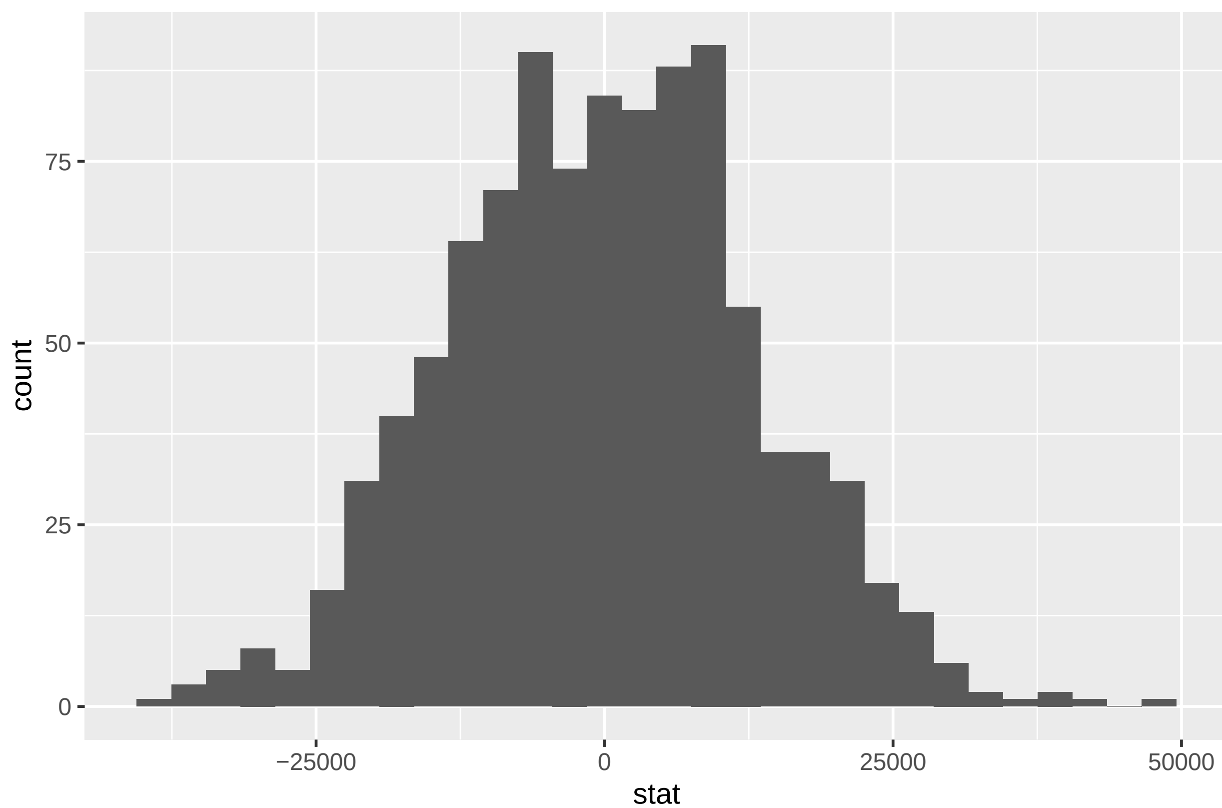

Figure 5

After confirming the normality of the distribution a two-tailed t-test will be performed.

```
t.test(total_trips ~ type_of_day, data = trip_weather_data)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_trips by type_of_day
## t = 0.56513, df = 63.91, p-value = 0.574
## alternative hypothesis: true difference in means between group normal and group hot is not equal to 0
## 95 percent confidence interval:
##  -15439.73  27620.44
## sample estimates:
## mean in group normal    mean in group hot
##             526154.8             520064.5
```

Using a 95% confidence t-test, it tells us that we are unable to reject the null hypothesis and claim that a high heat index differs from non-high heat days. We see in the confidence interval (-15439.73, 27620.44), it includes zero, where as the p-value is also 0.6675 which is >0.05 alpha threshold.

Alternative Heat Index Model

While we failed to reject the null hypothesis, using a t-test, there are other controlling variables to account for that may influence how a heat index may prove to be useful. A multiple linear regression model will be used to account for the seasonal variability in our previous figures. **day_of_week** will factor out each day of the week, having Monday as the reference variable, **month** will factor out each month of the year, having January as the reference variable and **year**, factoring out for the two year data collected, having 2021 as the reference variable. Lastly, total precipitation will be added as well.

Precipitation

Here we can see no linear trend of precipitation to total daily trips.

```
ggplot(aes(x = precip, y = total_trips), data = trip_weather_data) +
  geom_point() +
  labs(x = 'precipitation (inches)', y = 'total trips', caption = 'Figure 6') +
  scale_y_continuous(labels = scales::comma)
```

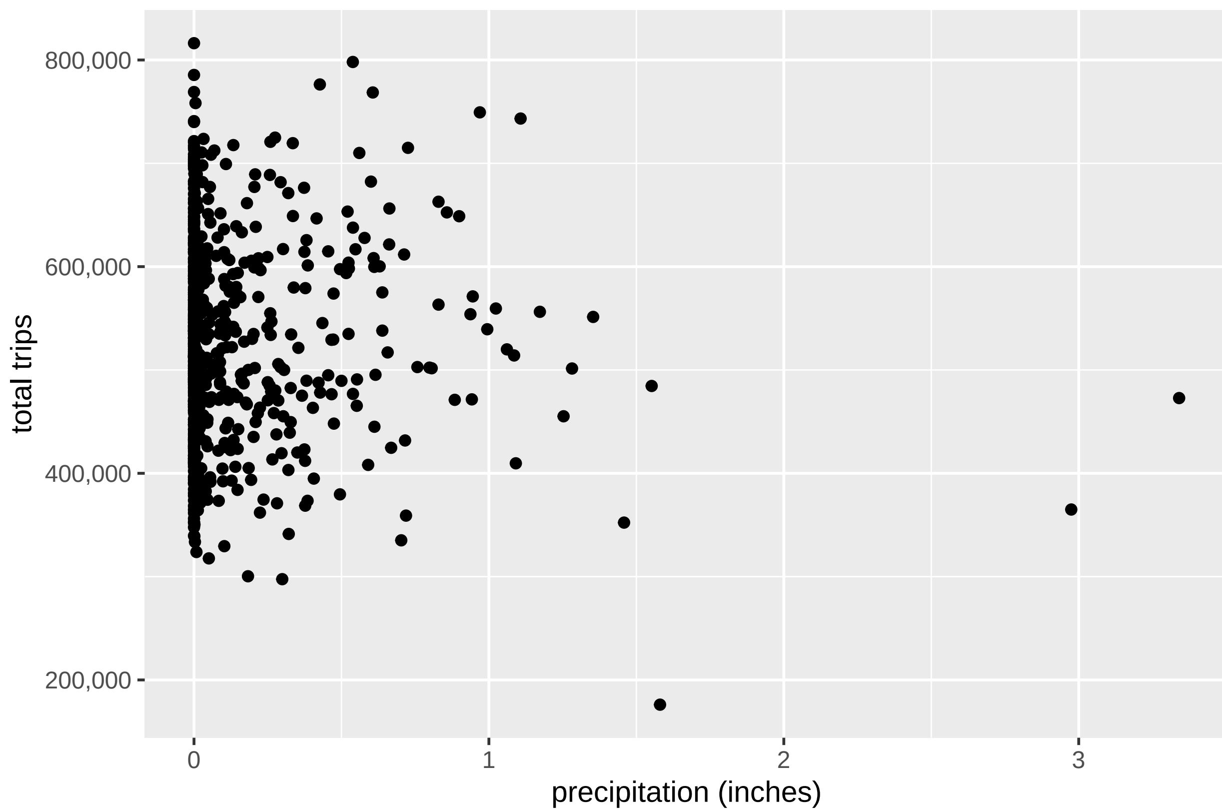

Figure 6

As for the correlation between heat index and precipitation we get 0.01965 which shows almost zero correlation with each other.

```
cor(trip_weather_data$heat_idx, trip_weather_data$precip)
```

```
## [1] 0.01965289
```

Model

Now we can build the linear regression as followed:

$$\hat{y} = \beta_0 + x\beta_1 + x\beta_2 + x\beta_3 + x\beta_4 + x\beta_5$$

where

$$\beta_1 = type\_of\_day, \beta_2 = precip, \beta_3 = day\_of\_week, \beta_4 = month, \beta_5 = year$$

Results

```
lm_mod <- lm(total_trips ~ type_of_day + precip + day_of_week + month + year, data = trip_weather_data)
summary(lm_mod)
```

```
##
## Call:
## lm(formula = total_trips ~ type_of_day + precip + day_of_week +
##     month + year, data = trip_weather_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -236799  -16242    1743   18865  130481
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      296687       6352  46.709  < 2e-16 ***
## type_of_dayhot    16807       6584   2.553 0.010900 *
## precip            -3264       5132  -0.636 0.524971
## day_of_weekTue    18886       5476   3.449 0.000596 ***
## day_of_weekWed    47471       5476   8.669  < 2e-16 ***
## day_of_weekThu    77662       5477  14.179  < 2e-16 ***
## day_of_weekFri   135131       5464  24.734  < 2e-16 ***
## day_of_weekSat   172810       5477  31.552  < 2e-16 ***
## day_of_weekSun    80637       5481  14.712  < 2e-16 ***
## monthFeb          65816       7278   9.043  < 2e-16 ***
## monthMar         103416       7090  14.585  < 2e-16 ***
## monthApr          99763       7147  13.959  < 2e-16 ***
## monthMay         104052       7090  14.676  < 2e-16 ***
## monthJun         117217       7200  16.280  < 2e-16 ***
## monthJul          85819       7431  11.549  < 2e-16 ***
## monthAug          82951       7401  11.208  < 2e-16 ***
## monthSep         116278       7154  16.253  < 2e-16 ***
## monthOct         147176       7094  20.745  < 2e-16 ***
## monthNov         141280       7149  19.762  < 2e-16 ***
## monthDec         141938       7118  19.939  < 2e-16 ***
## year2022         103358       2925  35.339  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39450 on 708 degrees of freedom
## Multiple R-squared:  0.8292, Adjusted R-squared:  0.8244
## F-statistic: 171.9 on 20 and 708 DF,  p-value: < 2.2e-16
```

The linear regression model

$$
\begin{aligned}
\hat{y} = 296687 \; & + 16807 \times type\_of\_dayhot \\
& - 3264 \times daily\_precip \\
& + 18886 \times day\_of\_weekTue \\
& + 47471 \times day\_of\_weekWed \\
& + 77662 \times day\_of\_weekThu \\
& + 135131 \times day\_of\_weekFri \\
& + 172810 \times day\_of\_weekSat \\
& + 80637 \times day\_of\_weekSun \\
& + 65816 \times monthFeb \\
& + 103416 \times monthMar \\
& + 99763 \times monthApr \\
& + 104052 \times monthMay \\
& + 117217 \times monthJun \\
& + 85819 \times monthJul \\
& + 82951 \times monthAug \\
& + 116278 \times monthSep \\
& + 147176 \times monthOct \\
& + 141280 \times monthNov \\
& + 141938 \times monthDec \\
& + 103358 \times year2022
\end{aligned}
$$

When controlling for the seasonality, the heat index (*type_of_dayhot*) became statistically significant. Holding all other variables constant, when the heat index is />=90 we can estimate an increase of 16807 trips than non high heat index days. As for the practical significance of the model, it is quite surprisingly a significant predictor as the adjusted R^2 of the model is 0.8244.

Checking Assumptions

Normality The Q-Q plot shows there is some "S" curvature within the band of residuals, but overall is straight.

```
ggplot(data = lm_mod, aes(sample = .resid)) +
  stat_qq() +
  labs(caption = "Figure 7")
```

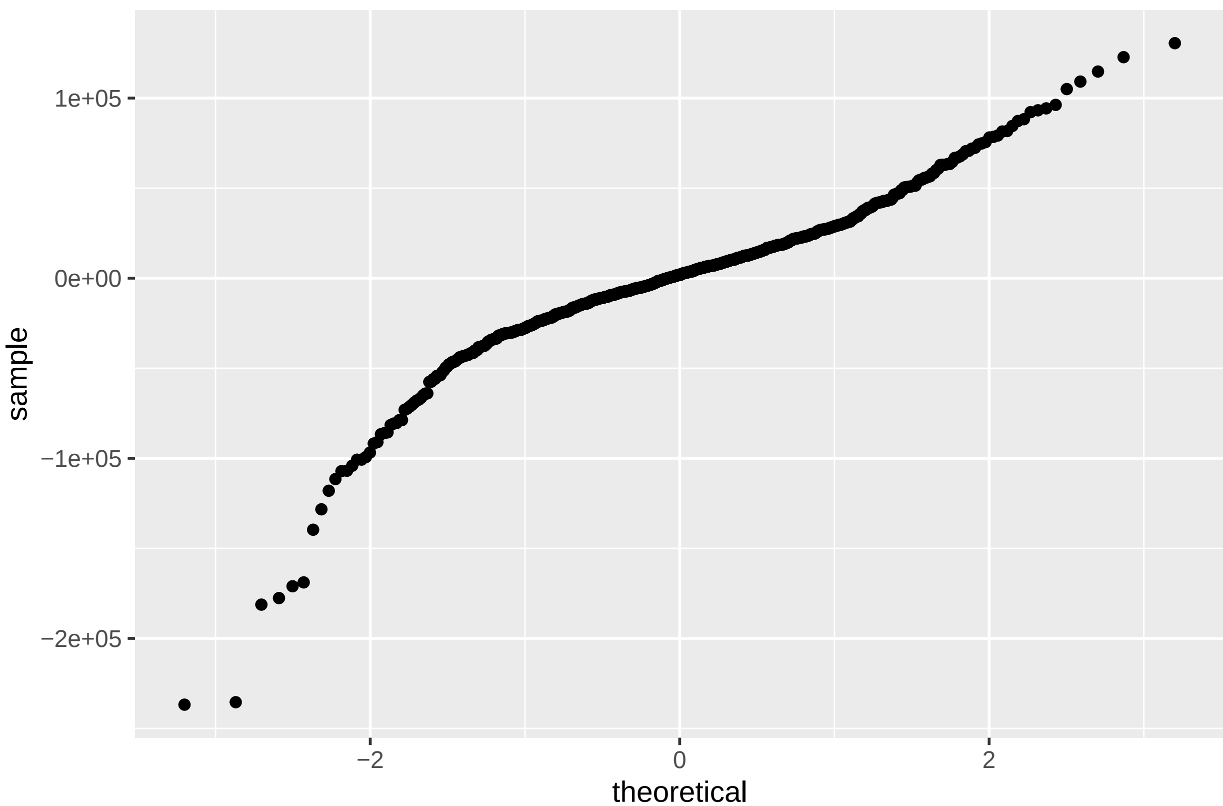

Figure 7

Constant variability The spread around zero does appear to have some heteroskedasticity as it is cone-shaped but overall nothing alarming.

```
ggplot(data = lm_mod, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals") +
  labs(caption = "Figure 8")
```

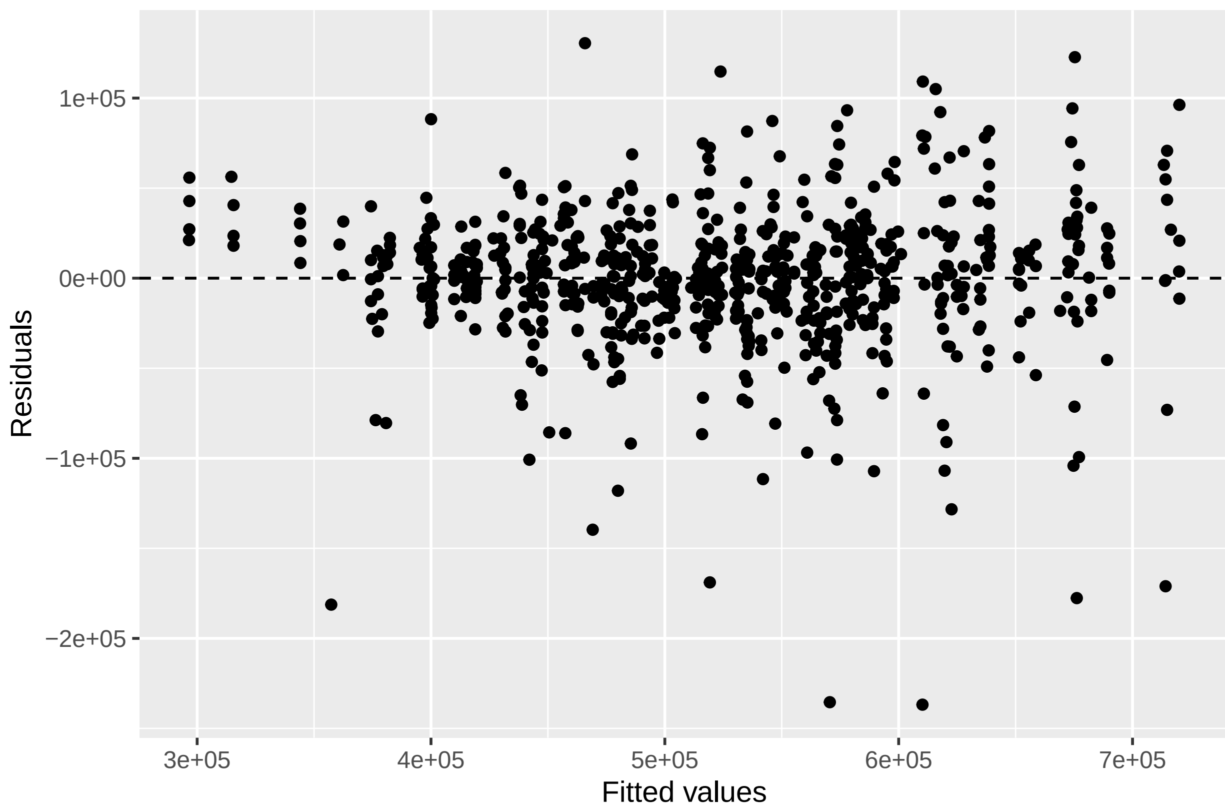

Figure 8

Linearity It passes the linearity test, even though there is some initial curve in Figure 4.

Conclusions

When using the heat index as a measure of trips taken, it alone does not seem to be an indicator of higher trips being taken via Uber and Lyft. However, when controlling for the seasonal effects, it does become a statistically significant indicator of trip counts.

There are some concerns about the high results of the adjusted R^2 and the yearly data used. Since the years were being held constant, this is partially because of the recovery of the industry since the pandemic. Trip counts are not fully where it was prior and more investigating has to be performed to understand riders behaviors commuting around the NYC region. It is also important to be aware that this analysis focuses on the unique environment of NYC but can the same be said for other hot climates such as Miami or Los Angeles?