Weather and Rideshare 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 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 August 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 in the month of August 2022.

Type of study

This is an observational study.

Variables

Dependent The response variable is total trips and is numerical

Independent Variable(s) The independent variables are:

- type_of_day: categorical
- precipitation: numerical
- day_of_week: 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(lubridate)
library(DBI)
```

Data Preparation

Load Historical Weather Data

Calculate Heat Index

The measurements for the United States is generally in Fahrenheit. The weather data will be converted from Celsius to Fahrenheit using the *weathermetrics* library.

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

Day of Week

Using the *lubridate* library, we will determine the day of the week and transform the data type with factor levels. The datetime_utc will also be updated to New York's local time to match the trip records.

```
weather <- read_parquet('weather.parquet') |>
  janitor::clean_names()

knitr::kable(head(weather))
```

datetime_ny	temp_deg_f	rel_humidity	heat_idx	total_precip
2022-07-31 20:00:00	79.34	69.89607	82	0.0007874
2022-07-31 21:00:00	78.69	73.22915	81	0.0003937

datetime_ny	temp_deg_f	rel_humidity	heat_idx	total_precip
2022-07-31 22:00:00	78.17	75.51439	80	0.0055118
2022-07-31 23:00:00	77.13	77.92819	78	0.0000000
2022-08-01 00:00:00	76.59	77.55404	78	0.0000000
2022-08-01 01:00:00	75.56	79.02121	77	0.0000000

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).

Note: The data has been cleaned and filtered using the R script *tlc_data_filter.R* that is within the same GitHub repo. Here are the changes:

- All the trips performed in August were found to be within two separate files for both August and September.
- 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.

```
tlc_trips <- read_parquet('tlc_trips.parquet')
```

```
tlc_trips_hourly <-
  tlc_trips |>
    group_by(app, pickup_floor) |>
    summarise(total_trips = n(),
              avg_trip_dist = mean(trip_miles),
              avg_trip_time = mean(trip_time),
              avg_base_fare = mean(base_passenger_fare))

knitr::kable(head(tlc_trips_hourly))
```

app	pickup_floor	total_trips	avg_trip_dist	avg_trip_time	avg_base_fare
Lyft	2022-08-01 00:00:00	5833	5.078416	1159.078	23.66116
Lyft	2022-08-01 01:00:00	7857	4.311040	1137.815	21.77827
Lyft	2022-08-01 02:00:00	7849	4.691822	1138.027	20.12130
Lyft	2022-08-01 03:00:00	6947	4.977280	1203.712	19.93163
Lyft	2022-08-01 04:00:00	6777	5.068789	1219.372	19.71429
Lyft	2022-08-01 05:00:00	6737	5.054369	1225.231	20.44661

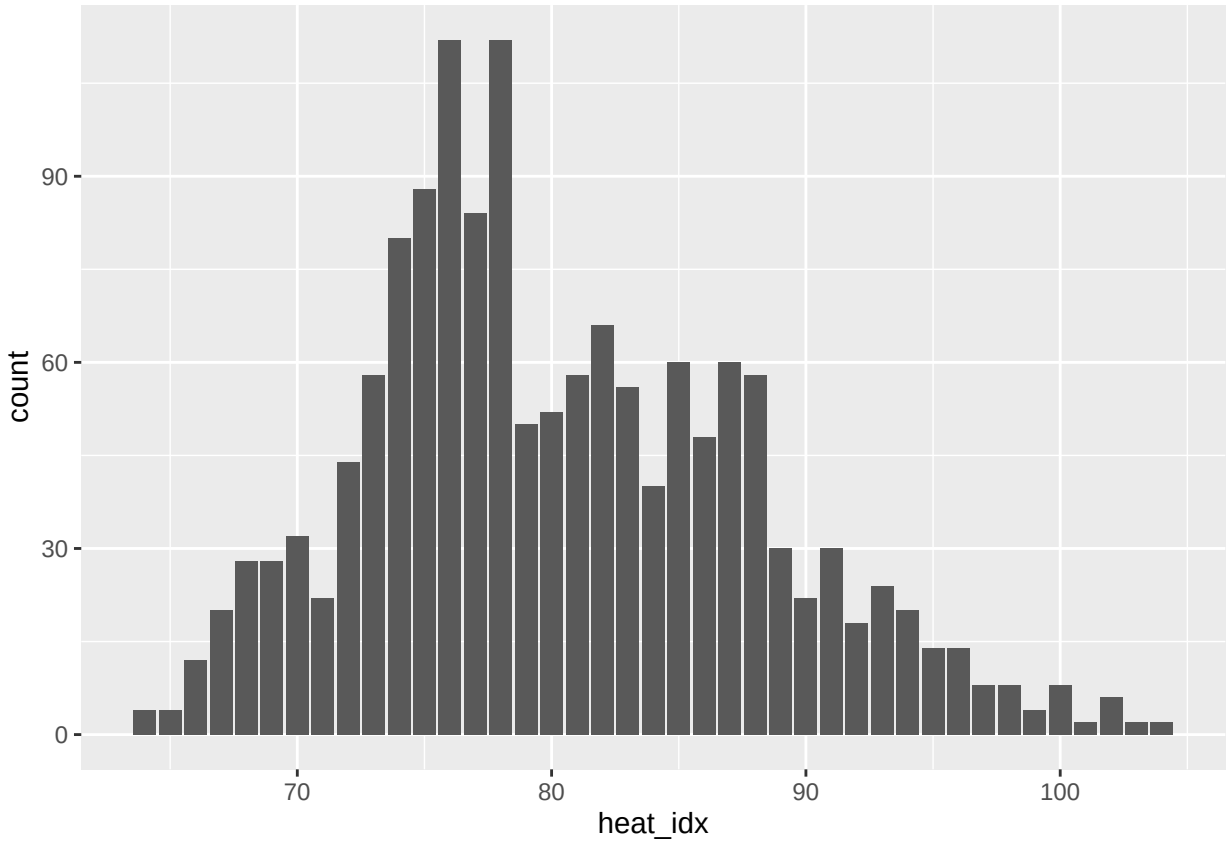
Merge Datasets

```
trip_weather_data <-
  tlc_trips_hourly |>
  left_join(weather, by = join_by(pickup_floor == datetime_ny)) |>
  mutate(type_of_day = ifelse(heat_idx >= 90, 'hot', 'not_hot'))

write_parquet(trip_weather_data, "trip_weather_data.parquet")
```

Relevant summary statistics

Count of hours for heat index throughout the month of August 2022.

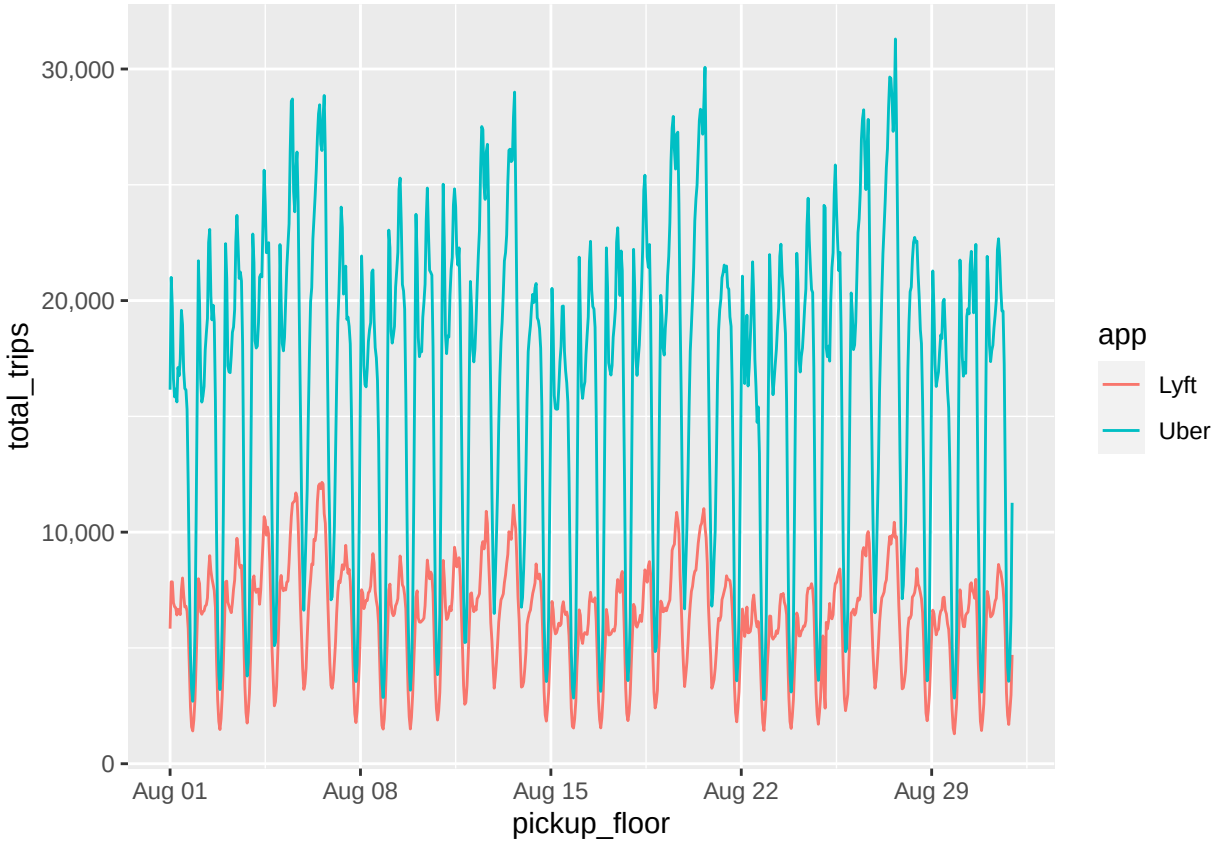
```
trip_weather_data |>
  mutate(hour = hour(pickup_floor)) |>
  group_by(heat_idx) |>
  summarise(count = n()) |>
  ggplot(aes(x = heat_idx, y = count)) +
  geom_bar(stat = 'identity')
```



Count of trips by Uber and Lyft in August.

```
trip_weather_data |>
  ggplot(aes(x = pickup_floor, y = total_trips, colour = app)) +
```

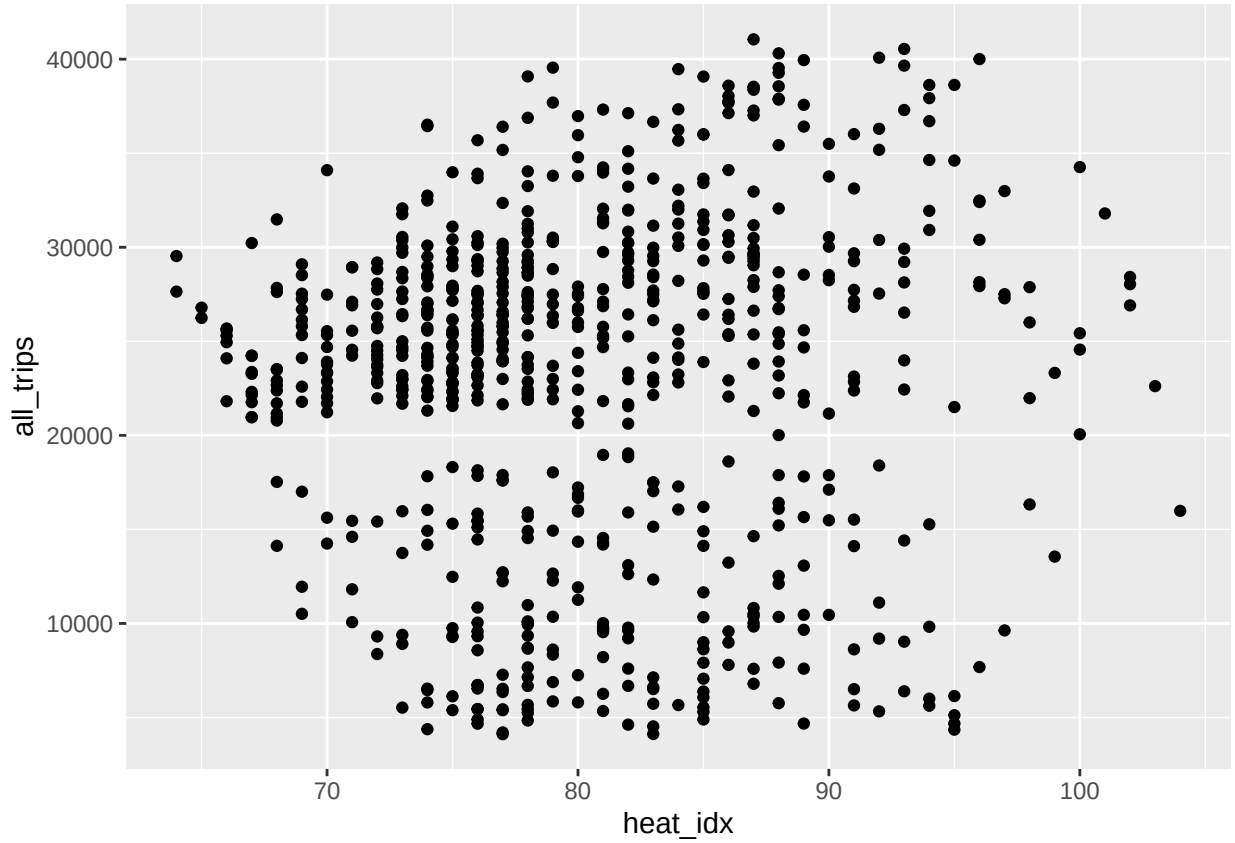
```
  geom_line(stat = 'identity') +
  scale_y_continuous(labels = scales::comma)
```



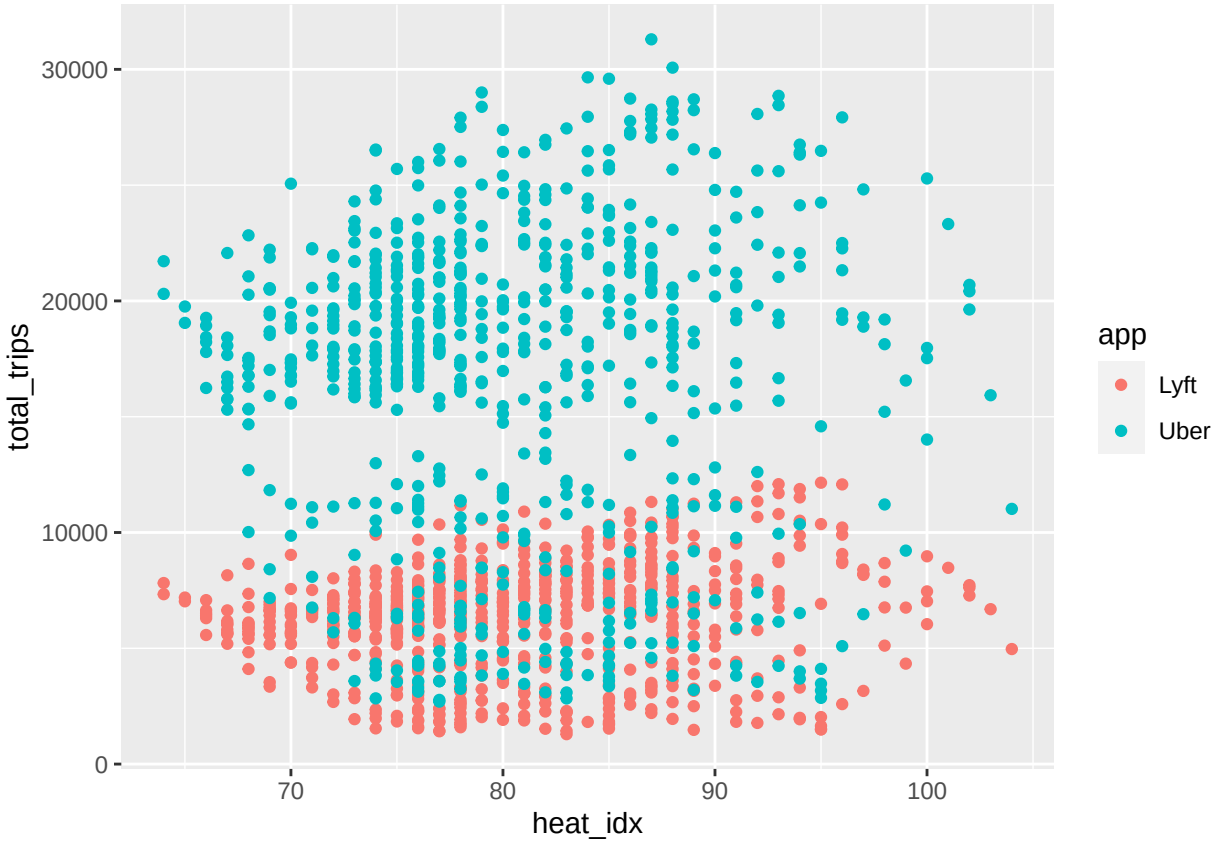

```
all_trips <-
  trip_weather_data |>
  group_by(pickup_floor, heat_idx) |>
  summarise(all_trips = sum(total_trips))
```

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

```
all_trips |>
  ggplot(aes(x = heat_idx, y = all_trips)) +
  geom_point(stat = 'identity')
```




```
trip_weather_data |>
  ggplot(aes(x = heat_idx, y = total_trips, color = app)) +
  geom_point(stat = 'identity')
```



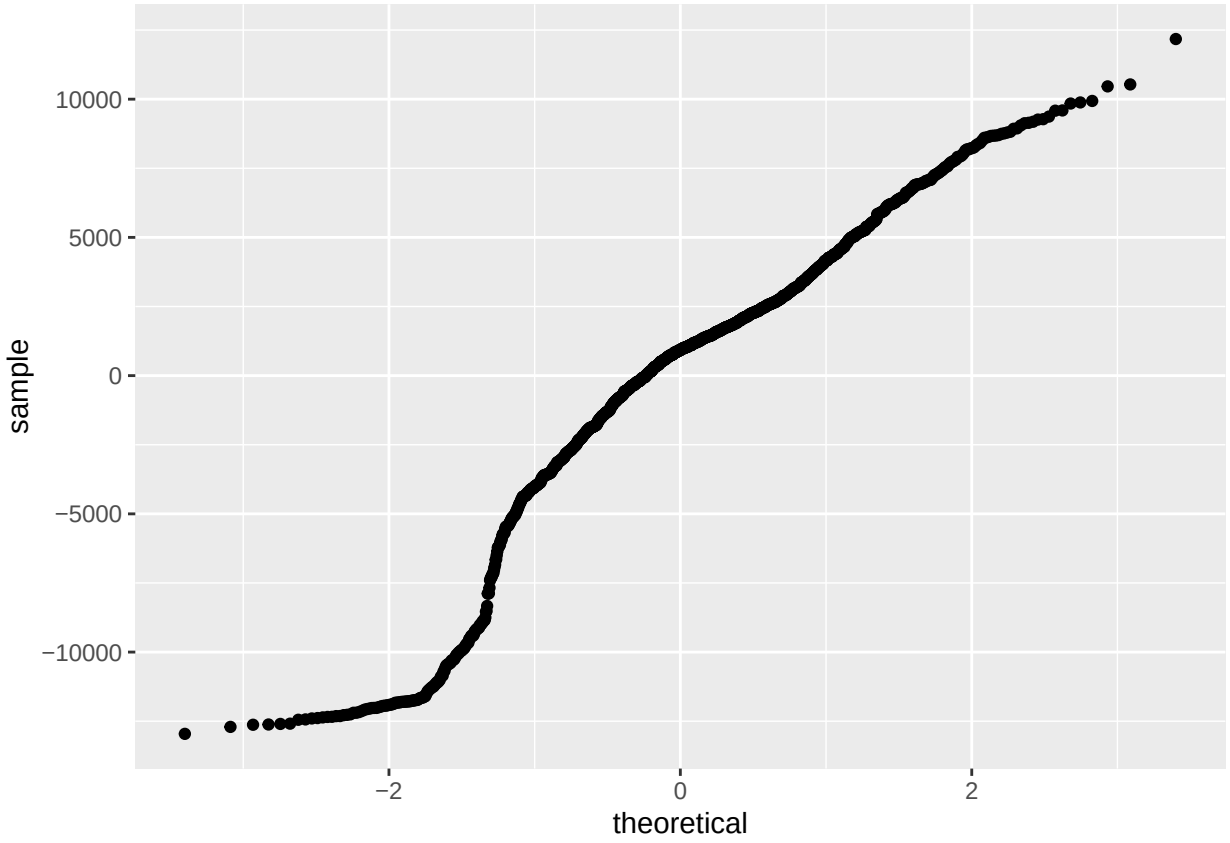

```
seasonal <-
  trip_weather_data |>
  mutate(day_of_week = factor(wday(pickup_floor, label = TRUE, week_start = 1), ordered = FALSE))

seasonal$day_of_week = relevel(seasonal$day_of_week, ref='Mon')
seasonal$type_of_day = relevel(factor(seasonal$type_of_day, ordered = FALSE), ref='not_hot')

lm_mod <- lm(total_trips ~ app + type_of_day + total_precip + day_of_week, data = seasonal)
summary(lm_mod)
```

```
##
## Call:
## lm(formula = total_trips ~ app + type_of_day + total_precip +
##     day_of_week, data = seasonal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12957.3  -2225.4    927.8   2733.5  12174.9
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4228.5      347.5  12.170  < 2e-16 ***
## appUber          10405.3      248.3  41.898  < 2e-16 ***
## type_of_dayhot     660.5      389.2   1.697 0.089847 .
## total_precip     30502.8    11501.3   2.652 0.008085 **
## day_of_weekTue     838.0      441.5   1.898 0.057906 .
```

```
## day_of_weekWed   1553.8      446.0   3.483 0.000509 ***
## day_of_weekThu   2401.5      471.0   5.099 3.86e-07 ***
## day_of_weekFri   4136.5      467.3   8.852  < 2e-16 ***
## day_of_weekSat   4487.2      469.6   9.555  < 2e-16 ***
## day_of_weekSun    702.7      467.0   1.505 0.132586
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4790 on 1478 degrees of freedom
## Multiple R-squared:  0.5647, Adjusted R-squared:  0.562
## F-statistic:   213 on 9 and 1478 DF,  p-value: < 2.2e-16
```

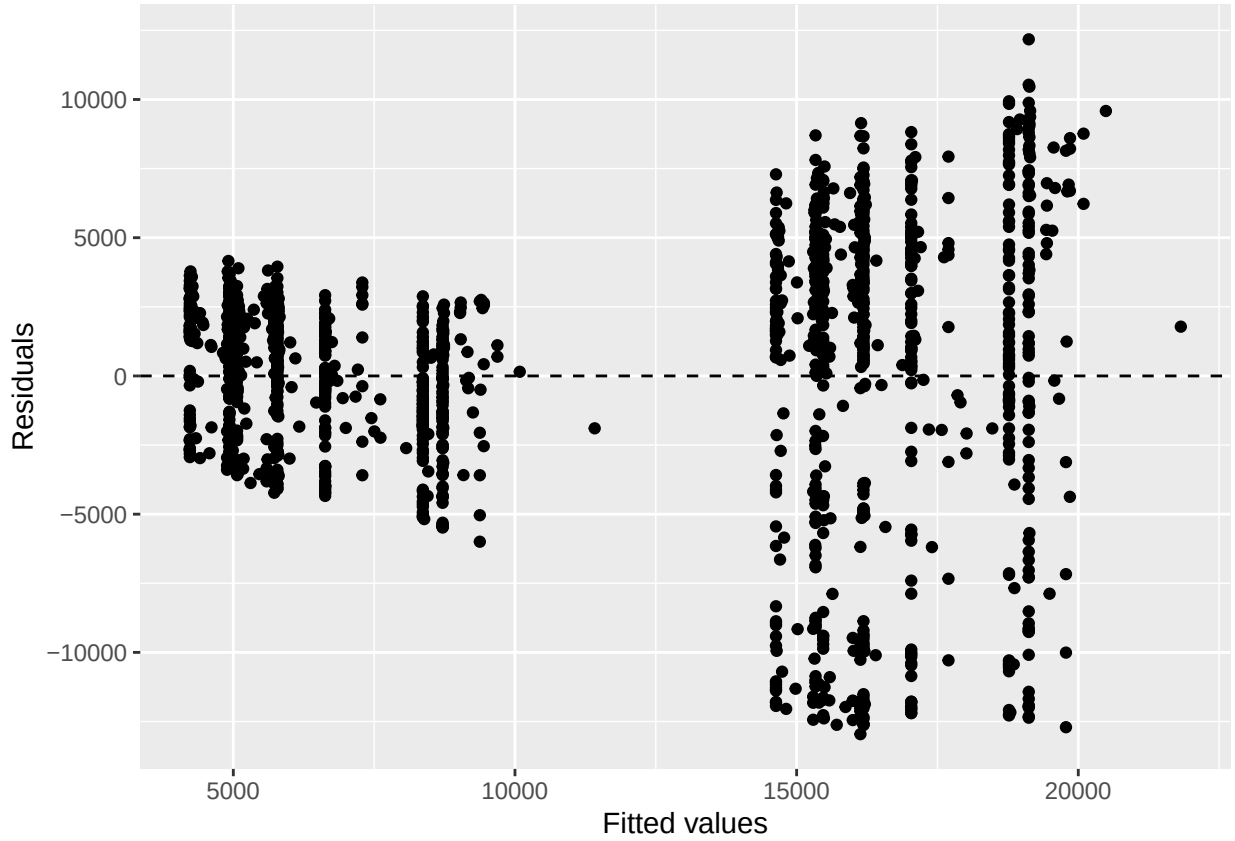
Normality The Q-Q plot shows there is a strong "S" curvature in the left side of the band of residuals, especially -1 standard deviation away.

```
ggplot(data = lm_mod, aes(sample = .resid)) +
  stat_qq()
```



Constant variability The spread around zero does appear to have some heteroskedasticity as it is cone-shaped

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



Linearity It generally passes the linearity test but again because of the cone shape of the residuals, some caution needs to be used.