

Manhattan-Taxis

Shreya Rao

Analyze analyze a random sample of 49999 New York journeys made in 2013 to help taxi drivers maximize their profits. Use regression trees and random forests to build a model that can predict the locations and times when the biggest fares can be earned.

```
# Loading the tidyverse
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.1.2      v dplyr  1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

# Reading in the taxi data
taxi <- read.table("C:/Users/Shreya/Documents/Projects/taxi.txt", sep = ",", header = T)

# Taking a look at the first few rows in taxi
head(taxi)
```

	medallion	pickup_datetime	pickup_longitude
## 1	4D24F4D8EF35878595044A52B098DFD2	2013-01-13T10:23:00Z	-73.94646
## 2	A49C37EB966E7B05E69523D1CB7BE303	2013-01-13T04:52:00Z	-73.99827
## 3	1E4B72A8E62388F53A9693C364AC05A	2013-01-13T10:47:00Z	-73.95346
## 4	F7E4E9439C46B8AD5B16AB9F1B3279D7	2013-01-13T11:14:00Z	-73.98137
## 5	A9DC75D59E0EA27E1ED328E8BE8CD828	2013-01-13T11:24:00Z	-73.96800
## 6	19BF1BB516C4E992EA3FBAEDA73D6262	2013-01-13T10:51:00Z	-73.98502

	pickup_latitude	trip_time_in_secs	fare_amount	tip_amount
## 1	40.77273	600	8.0	2.5
## 2	40.74041	840	18.0	0.0
## 3	40.77586	60	3.5	0.7
## 4	40.72473	720	11.5	2.3
## 5	40.76000	240	6.5	0.0
## 6	40.76341	540	8.5	1.7

Data Cleaning

```

# Renaming the location variables,
# dropping any journeys with zero fares and zero tips,
# and creating the total variable as the log sum of fare and tip
taxi <- taxi %>%
  rename(lat = pickup_latitude, long = pickup_longitude) %>%
  filter(fare_amount > 0 | tip_amount > 0) %>% mutate(total = log(fare_amount + tip_amount))

```

```

# Reducing the data to taxi trips starting in Manhattan
# Manhattan is bounded by the rectangle with
# latitude from 40.70 to 40.83 and
# longitude from -74.025 to -73.93
taxi <- taxi %>%
  filter(lat >= 40.70, lat <= 40.83, long >= -74.025, long <= -73.93)

```

```

# Loading in ggmap and viridis for nice colors
library(ggmap)

```

```

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

```

```

## Please cite ggmap if you use it! See citation("ggmap") for details.

```

```

library(viridis)

```

```

## Loading required package: viridisLite

```

```

register_google(key = "AIzaSyD1M0avrjJI0odaqQrRSs2qjBX6IlMksfI", write = TRUE)

```

```

## Replacing old key (AIzaSyD1M0avrjJI0odaqQrRSs2qjBX6IlMksfI) with new key in C:\Users\Shreya\.Renviron

```

```

# Retrieving a stored map object which originally was created by
manhattan <- get_map("manhattan", zoom = 12, color = "bw")

```

```

## Source : https://maps.googleapis.com/maps/api/staticmap?center=manhattan&zoom=12&size=640x640&scale=1

```

```

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=manhattan&key=xxx

```

```

# <- readRDS("C:/Users/Shreya/Documents/Projects/manhattan.rds")

```

```

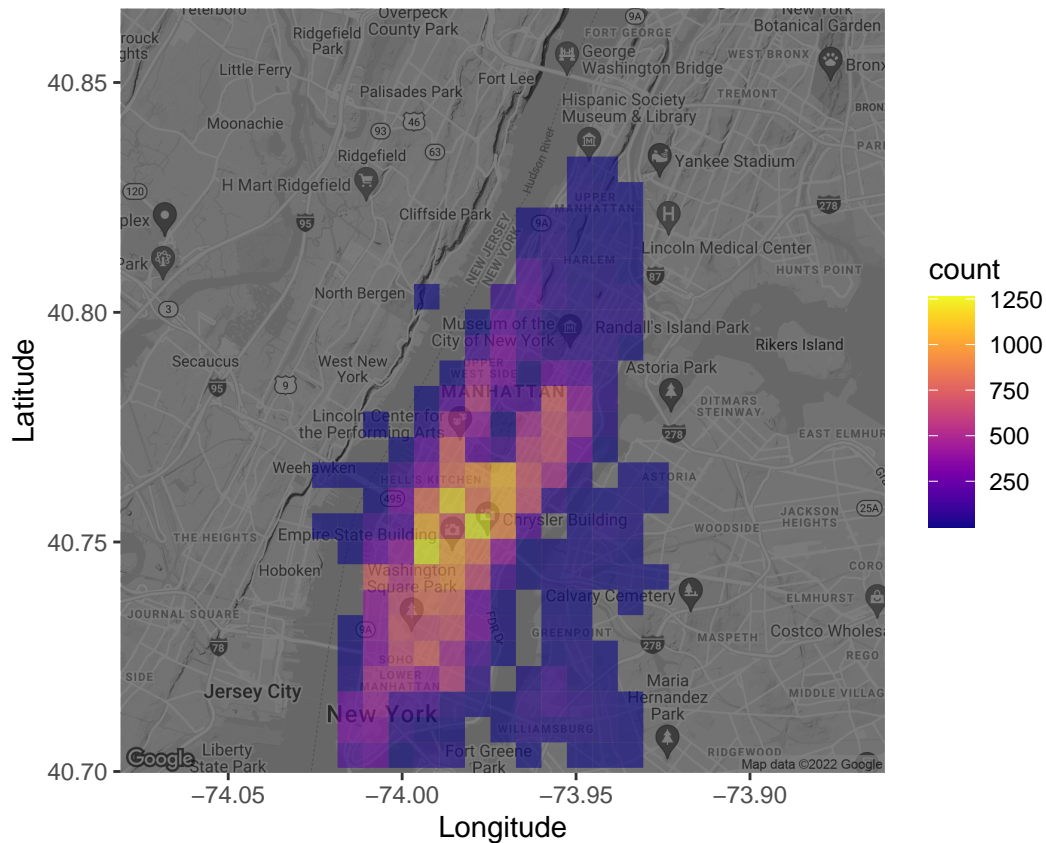
# Drawing a density map with the number of journey start locations
ggmap(manhattan, darken = 0.5) +
  scale_fill_viridis(option = 'plasma') +
  geom_bin2d(data = taxi, mapping = aes(x = long, y = lat), alpha = 0.6) +
  labs(x = "Longitude", y = "Latitude")

```

```

## Warning: Removed 4 rows containing missing values (geom_tile).

```



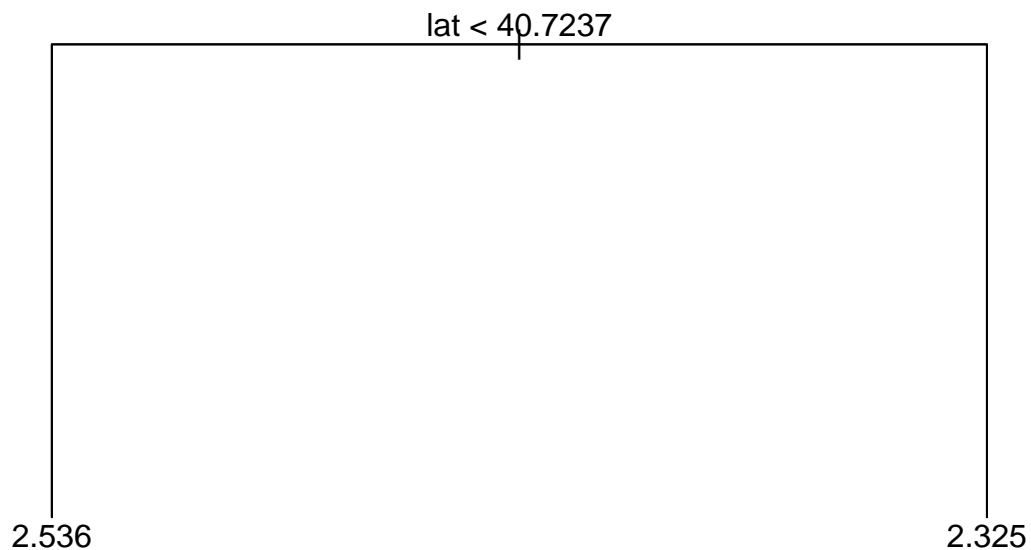
Predicting Taxi Fares Using a Tree

```
# Loading in the tree package
library(tree)

## Registered S3 method overwritten by 'tree':
##   method      from
##   print.tree cli

# Fitting a tree to lat and long
fitted_tree <- tree(total~lat+long, data=taxi)

# Draw a diagram of the tree structure
plot(fitted_tree)
text(fitted_tree)
```



It predicts that trips where $\text{lat} < 40.7237$ are more expensive, which makes sense as it is downtown Manhattan.

Adding some more predictors related to the time the taxi trip was made:

```
# Loading in the lubridate package
library(lubridate)
```

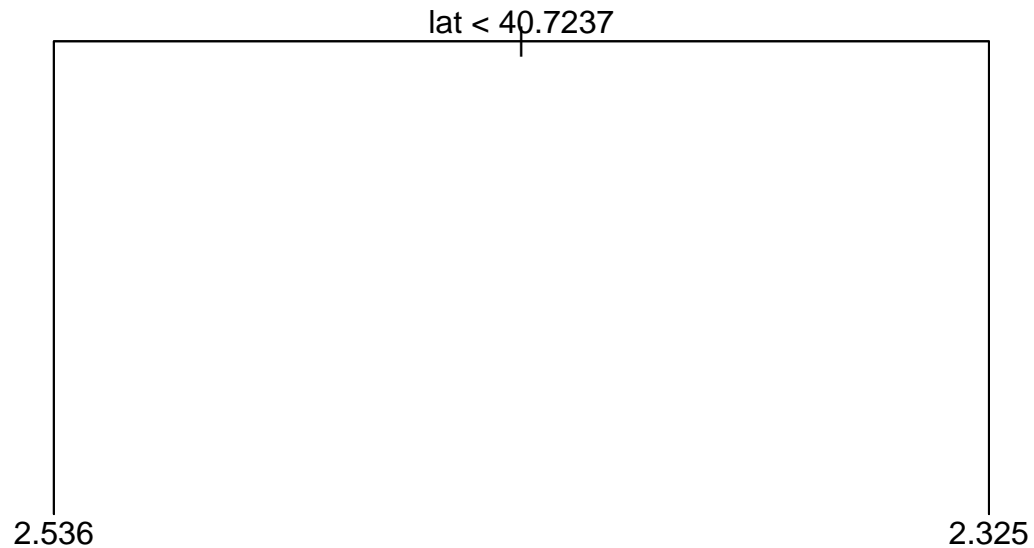
```
##
## Attaching package: 'lubridate'

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

```
# Generate the three new time variables
taxi <- taxi %>%
  mutate(hour = hour(pickup_datetime),
         wday = wday(pickup_datetime, label = TRUE),
         month = month(pickup_datetime, label = TRUE))
```

```
# Fitting a tree with total as the outcome and
# lat, long, hour, wday, and month as predictors
fitted_tree <- tree(total~lat+long+hour+wday+month, data=taxi)
```

```
# draw a diagram of the tree structure
plot(fitted_tree)
text(fitted_tree)
```



```
# Summarizing the performance of the tree
summary(fitted_tree)
```

```
##
## Regression tree:
## tree(formula = total ~ lat + long + hour + wday + month, data = taxi)
## Variables actually used in tree construction:
## [1] "lat"
## Number of terminal nodes: 2
## Residual mean deviance: 0.3041 = 13910 / 45760
## Distribution of residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -1.61900 -0.37880 -0.04244  0.00000  0.32660  2.69900
```

Random Forest

```
# Loading in the randomForest package
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##  
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':  
##  
## combine
```

```
## The following object is masked from 'package:ggplot2':  
##  
## margin
```

```
# Fitting a random forest  
fitted_forest <- randomForest(total~lat+long+hour+wday+month, data=taxi,  
                              ntree = 80, sampsize = 10000)  
  
# Printing the fitted_forest object  
print(fitted_forest)
```

```
##  
## Call:  
## randomForest(formula = total ~ lat + long + hour + wday + month, data = taxi, ntree = 80, samp  
##           Type of random forest: regression  
##           Number of trees: 80  
## No. of variables tried at each split: 1  
##  
##           Mean of squared residuals: 0.3029565  
##           % Var explained: 1.75
```

Plotting the Predicted Fare

```
# Extracting the prediction from fitted_forest  
taxi$pred_total <- fitted_forest$predicted  
  
# Plotting the predicted mean trip prices from according to the random forest  
ggmap(manhattan, darken = 0.5) +  
  scale_fill_viridis(option = 'plasma') +  
  stat_summary_2d(data = taxi, mapping = aes(x = long, y = lat, , z = pred_total), alpha = 0.6) +  
  labs(x = "Longitude", y = "Latitude", fill = "Predicted Fare")
```

```
## Warning: Removed 4 rows containing missing values (geom_tile).
```

