Taxi Fare

Kitchai Srichompu

2022-10-10

```
#install.packages("ggmap")
#install.packages("viridis")
#install.packages("tidyverse")
#install.packages("tree")
#install.packages("lubridate")
#install.packages("randomForest")
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
library(randomForest)
library(lubridate)
```

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

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

```
library(tree)
library(ggmap)
```

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

```
##
## Attaching package: 'ggplot2'
```

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

```
## 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
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
```

```
## v purrr   0.3.5
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x dplyr::combine()         masks randomForest::combine()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x ggplot2::margin()        masks randomForest::margin()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
#importing dataset
df <- read_csv("taxi.csv")
```

```
## Rows: 49999 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): medallion
## dbl  (5): pickup_longitude, pickup_latitude, trip_time_in_secs, fare_amount,...
## dttm (1): pickup_datetime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(df)
```

```
## # A tibble: 6 x 7
##   medallion          pickup_datetime     picku~1 picku~2 trip_~3 fare_~4 tip_a~5
##   <chr>              <dttm>                <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 4D24F4D8EF3587859~ 2013-01-13 10:23:00   -73.9    40.8     600     8       2.5
## 2 A49C37EB966E7B05E~ 2013-01-13 04:52:00   -74.0    40.7     840    18       0
## 3 1E4B72A8E623888F5~ 2013-01-13 10:47:00   -74.0    40.8      60     3.5     0.7
## 4 F7E4E9439C46B8AD5~ 2013-01-13 11:14:00   -74.0    40.7     720    11.5     2.3
## 5 A9DC75D59E0EA27E1~ 2013-01-13 11:24:00   -74.0    40.8     240     6.5     0
## 6 19BF1BB516C4E992E~ 2013-01-13 10:51:00   -74.0    40.8     540     8.5     1.7
## # ... with abbreviated variable names 1: pickup_longitude, 2: pickup_latitude,
## #   3: trip_time_in_secs, 4: fare_amount, 5: tip_amount
```

```
str(df)
```

```
## spec_tbl_df [49,999 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ medallion        : chr [1:49999] "4D24F4D8EF35878595044A52B098DFD2" "A49C37EB966E7B05E69523D1CB7B
##  $ pickup_datetime  : POSIXct[1:49999], format: "2013-01-13 10:23:00" "2013-01-13 04:52:00" ...
##  $ pickup_longitude : num [1:49999] -73.9 -74 -74 -74 -74 ...
##  $ pickup_latitude  : num [1:49999] 40.8 40.7 40.8 40.7 40.8 ...
##  $ trip_time_in_secs: num [1:49999] 600 840 60 720 240 540 0 120 720 180 ...
##  $ fare_amount      : num [1:49999] 8 18 3.5 11.5 6.5 8.5 2.5 4 14 4 ...
##  $ tip_amount       : num [1:49999] 2.5 0 0.7 2.3 0 1.7 0 0 2 3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   medallion = col_character(),
##   ..   pickup_datetime = col_datetime(format = ""),
##   ..   pickup_longitude = col_double(),
##   ..   pickup_latitude = col_double(),
##   ..   trip_time_in_secs = col_double(),
```

```
##   ..   fare_amount = col_double(),
##   ..   tip_amount = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(df)
```

```
##   medallion         pickup_datetime                  pickup_longitude
##  Length:49999       Min.   :2013-01-01 00:18:47.00   Min.   :-82.63
##  Class :character   1st Qu.:2013-03-29 14:55:30.50   1st Qu.:-73.99
##  Mode  :character   Median :2013-06-25 13:10:08.00   Median :-73.98
##                     Mean   :2013-06-29 16:41:39.99   Mean   :-72.56
##                     3rd Qu.:2013-10-01 08:52:43.50   3rd Qu.:-73.97
##                     Max.   :2013-12-31 23:33:00.00   Max.   : 40.81
##  pickup_latitude  trip_time_in_secs  fare_amount       tip_amount
##  Min.   :-74.01   Min.   :    0.0   Min.   :   0.00   Min.   : 0.000
##  1st Qu.: 40.73   1st Qu.:  365.0   1st Qu.:   6.50   1st Qu.: 0.000
##  Median : 40.75   Median :  600.0   Median :   9.50   Median : 1.000
##  Mean   : 39.83   Mean   :  757.3   Mean   :  12.44   Mean   : 1.377
##  3rd Qu.: 40.77   3rd Qu.:  960.0   3rd Qu.:  14.00   3rd Qu.: 2.000
##  Max.   : 41.59   Max.   :75240.0   Max.   :2069.50   Max.   :62.000
```

```
#cleaning data because tip and fare != 0
df <- df %>%
   filter(fare_amount > 0 | tip_amount > 0) #and create new column that containing amout paid
df <- df %>%  mutate(total = log(fare_amount + tip_amount) )
```

```
#plot which pick-up spot is popular for customer

register_google(key = "AIzaSyDoyuKwtfv3I9gKA2zS2mICD3M9vvTJy_w")
New_York <- get_map("new york", zoom = 12)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=new%20york&zoom=12&size=640x640&scale=
```

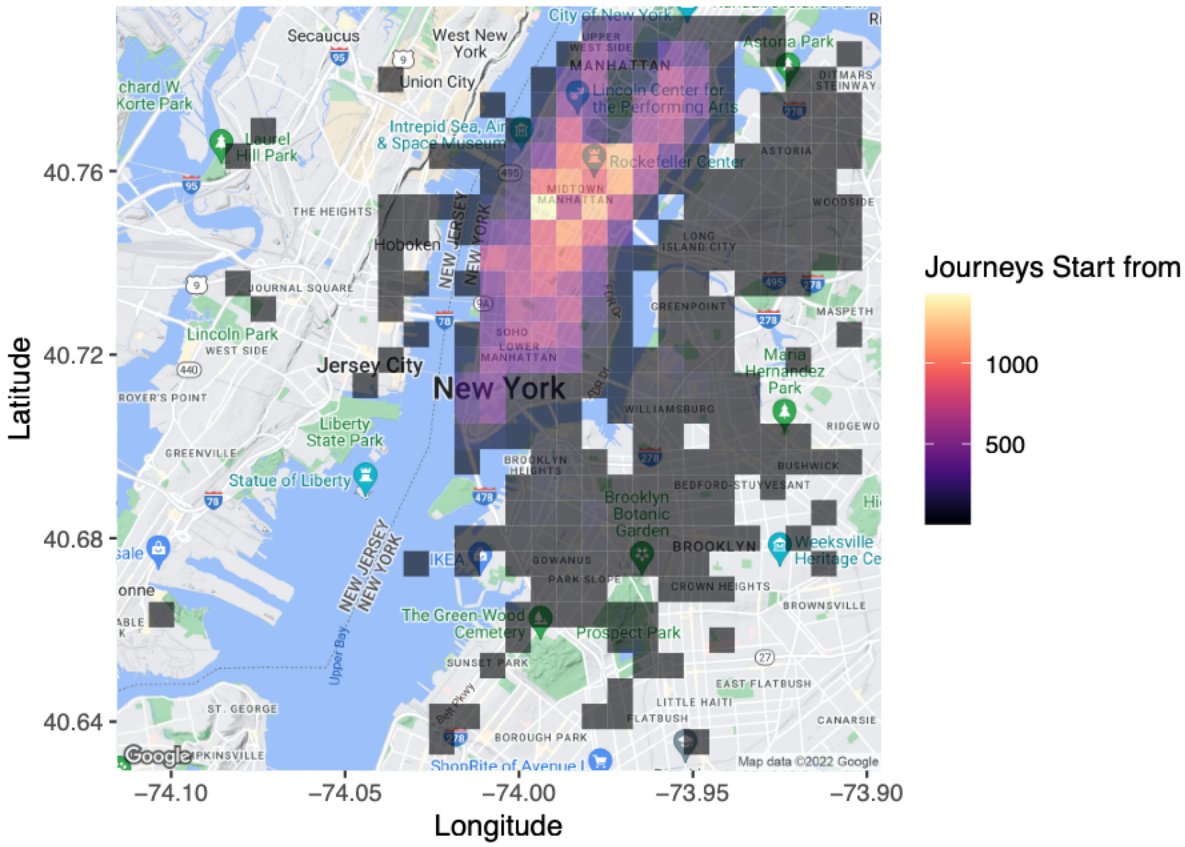
```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=new+york&key=xxx
```

```
ggmap(New_York) +
  scale_fill_viridis(option = 'magma') +
  geom_bin2d(data = df, aes(x=pickup_longitude, y=pickup_latitude), bin = 60, alpha = 0.6) +
  labs(x="Longitude",y="Latitude",fill="Journeys Start from")
```

```
## Warning: Ignoring unknown parameters: bin
```

```
## Warning: Removed 4363 rows containing non-finite values (stat_bin2d).
```

```
## Warning: Removed 6 rows containing missing values (geom_tile).
```

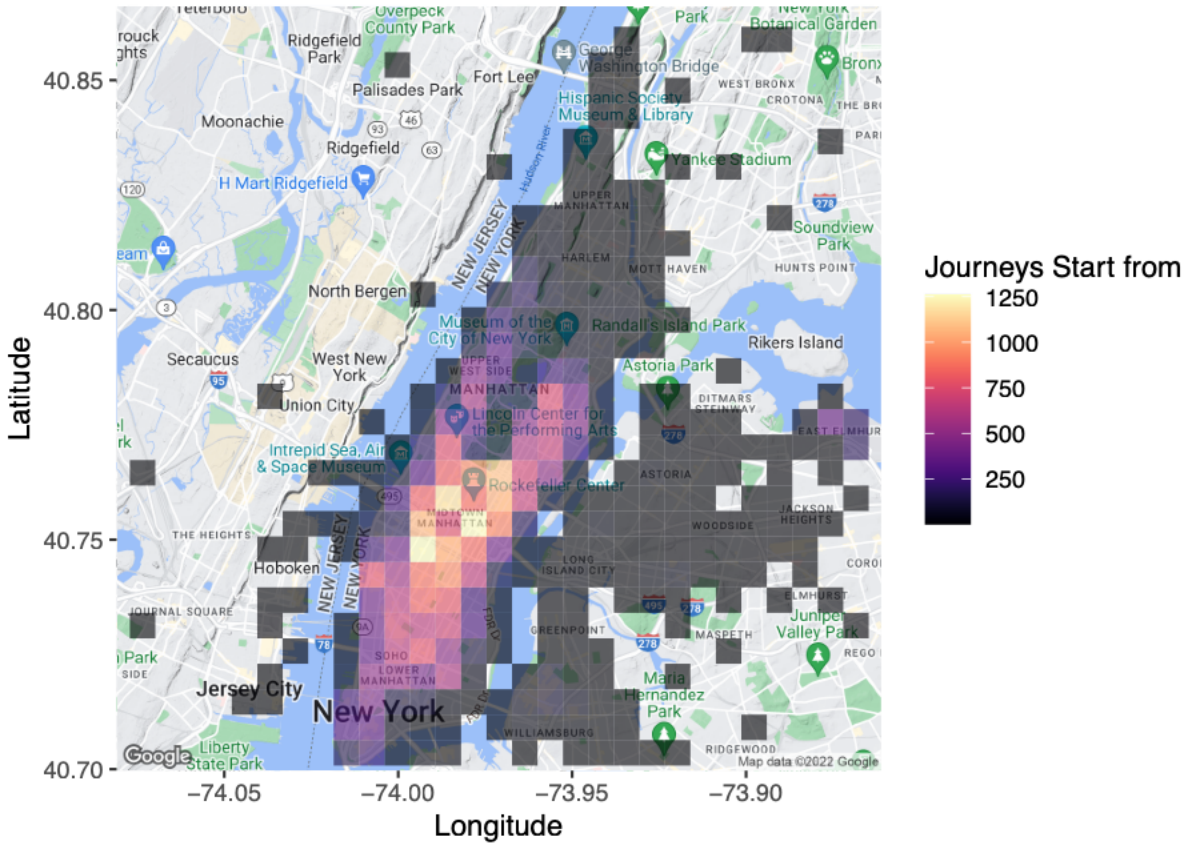



```
#mostly customer start their journey from manhattan
#lets get some closer

df_manhattan <- df %>% filter(between(pickup_latitude, 40.70, 40.83) & between(pickup_longitude, -74.025

Manhattan <- get_map("manhattan", zoom = 12)
```

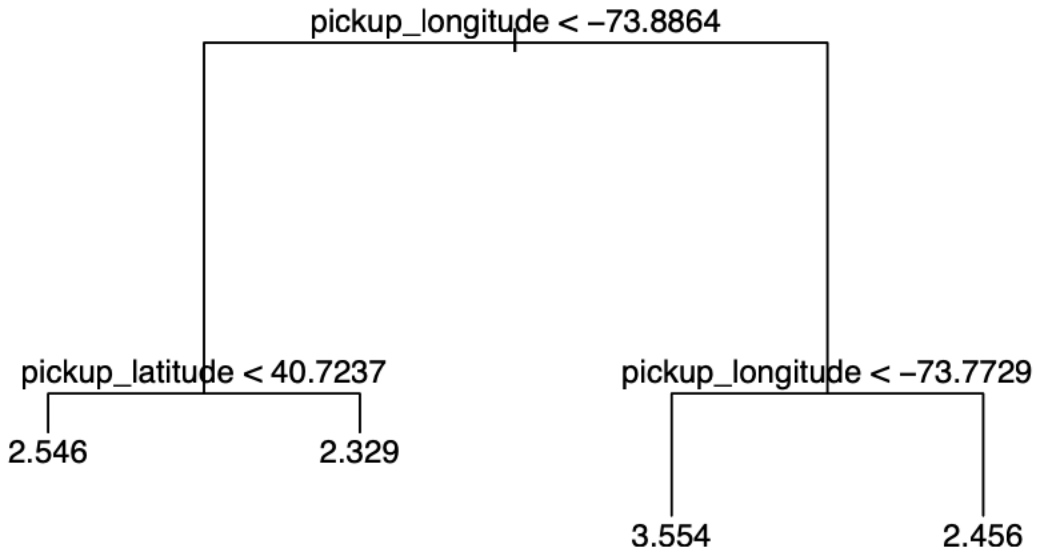
```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=manhattan&zoom=12&size=640x640&scale=1

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

```
ggmap(Manhattan) +
  scale_fill_viridis(option = 'magma') +
  geom_bin2d(data = df, aes(x=pickup_longitude, y=pickup_latitude), bin = 60, alpha = 0.6) +
  labs(x="Longitude",y="Latitude",fill="Journeys Start from")
```

```
## Warning: Ignoring unknown parameters: bin

## Warning: Removed 2573 rows containing non-finite values (stat_bin2d).

## Warning: Removed 7 rows containing missing values (geom_tile).
```



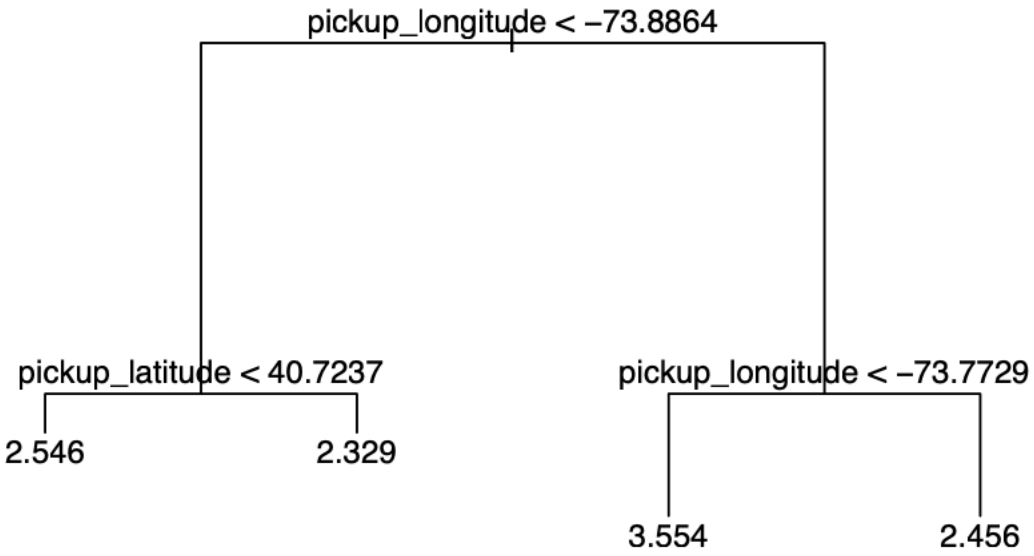

```
#Predicting taxi fares using a regression tree
df_tree <- tree(total ~ pickup_latitude + pickup_longitude, data = df)

plot(df_tree)
text(df_tree)
```




```
#where lat < 40.7237 is more expensive
```

```
#we need to do the predictor regarding time
df <- df %>%
```

```
    mutate(hour = hour(pickup_datetime),
           wday = wday(pickup_datetime, label = TRUE),
           month = month(pickup_datetime, label = TRUE))

dftree2 <- tree(total ~ pickup_latitude + pickup_longitude + hour + wday + month, data = df)

plot(dftree2)
text(dftree2)
```




```
summary(dftree2)
```

```
##
## Regression tree:
## tree(formula = total ~ pickup_latitude + pickup_longitude + hour +
##     wday + month, data = df)
## Variables actually used in tree construction:
## [1] "pickup_longitude" "pickup_latitude"
## Number of terminal nodes:  4
## Residual mean deviance:  0.3165 = 15820 / 49990
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.63800 -0.38270 -0.02603  0.00000  0.34550  5.17900
```

```
#the regression tree still the same
```

```
# Fitting a random forest
forest <- randomForest(total ~ pickup_latitude + pickup_longitude + hour + wday + month,
    data=df, ntree=80, sampsize=10000)

# Printing the fitted_forest object
forest
```

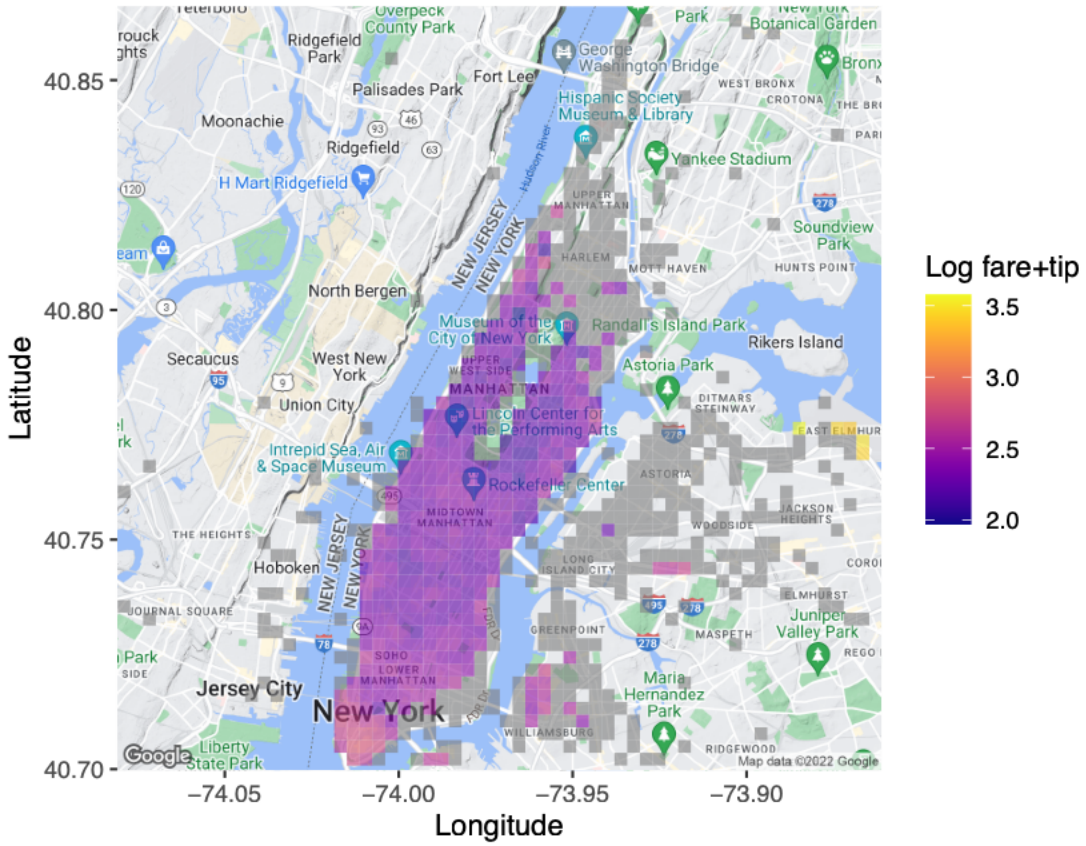
```
##
## Call:
##  randomForest(formula = total ~ pickup_latitude + pickup_longitude +      hour + wday + month, data =
##                Type of random forest: regression
##                      Number of trees: 80
## No. of variables tried at each split: 1
```

```
##
##           Mean of squared residuals: 0.3114711
##                     % Var explained: 17.9
```

```
#ploting base on price
#Function that returns the mean if there are 15 or more datapoints
mean_if_enough_data <- function(x) {
    ifelse( length(x) >= 15, mean(x), NA)
}

ggmap(Manhattan) +
    scale_fill_viridis(option = 'plasma') +
    stat_summary_2d(data=df, aes(x = pickup_longitude, y = pickup_latitude, z = total), fun = mean_if_e
        alpha = 0.6, bins = 60) +
    labs(x = 'Longitude', y = 'Latitude', fill = 'Log fare+tip')
```

```
## Warning: Removed 2573 rows containing non-finite values (stat_summary2d).
```

```
## Warning: Removed 6 rows containing missing values (geom_tile).
```




```
#tip spent most on downtoum
```