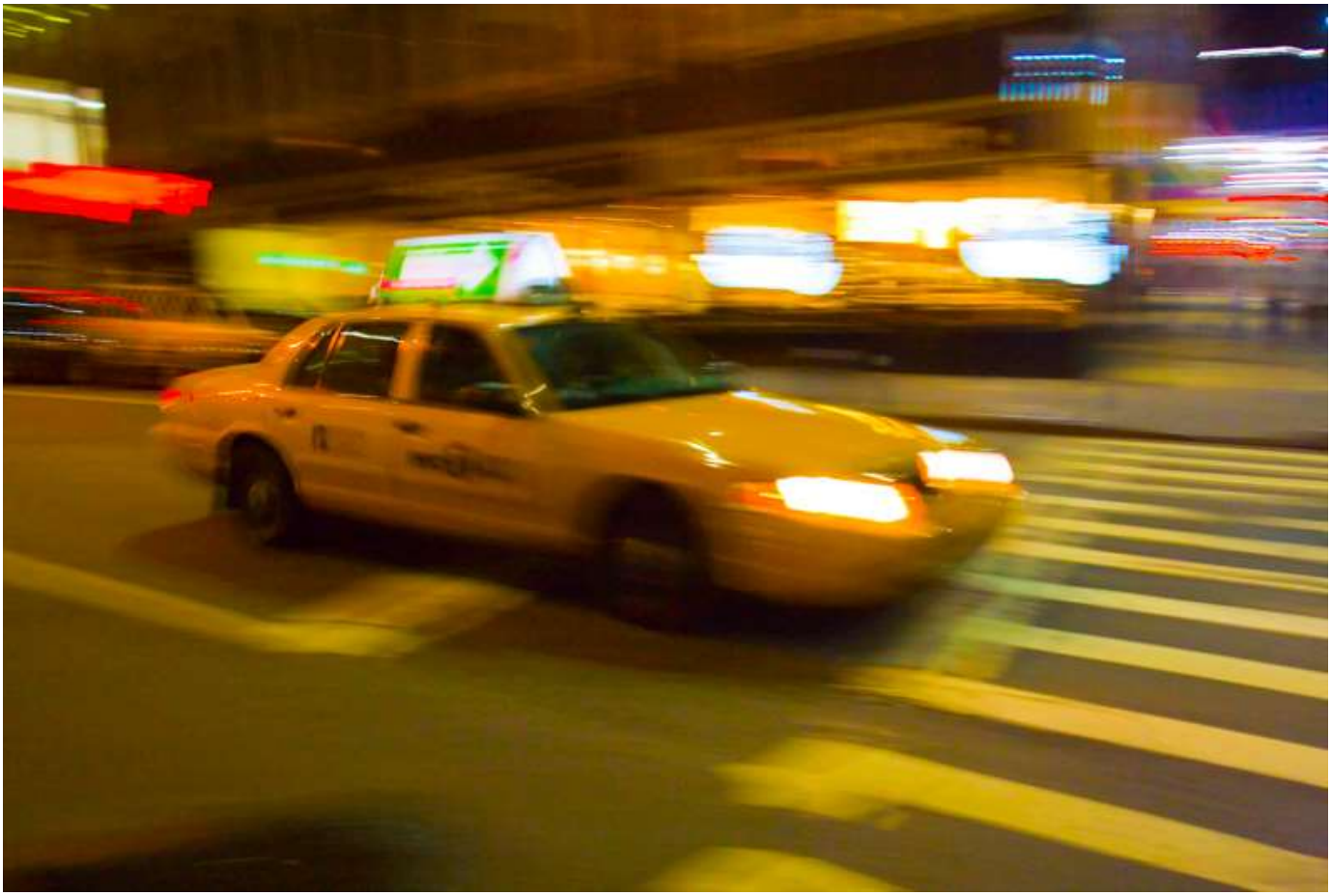
Nikhil Eatalpacka

1. New York taxi trips - 2013



In this project, we will analyze a random sample of 49999 New York journeys made in 2013. We will also use regression trees and random forests to build a model that can predict the locations and times when the biggest fares can be earned.

But this is the age of business intelligence and analytics! Even taxi drivers can stand to benefit from some careful investigation of the data, guiding them to maximize their profits

```
# Loading the tidyverse
library(tidyverse)

# Reading in the taxi data
taxi <- read_csv("datasets/taxi.csv")

head(taxi)
```

2. Cleaning the taxi data

The `taxi` dataset contains the times and price of a large number of taxi trips. Importantly we also get to know the location, the longitude and latitude, where the trip was started.

The data was altered to:
-Location variables were renamed
-Journeys with zero fares and tips were removed
-The log of the sum of fare and tip variables was calculated to deal with outliers, the log was stored in a new variable total

```
taxi <- taxi %>%
            rename(lat = pickup_latitude, long = pickup_longitude) %>%
            filter(fare_amount > 0, tip_amount > 0) %>%
            mutate(total = log(fare_amount + tip_amount))
head(taxi)
```

3. Zooming in on Manhattan

While the dataset contains taxi trips from all over New York City, the bulk of the trips are to and from Manhattan, so let's focus only on trips initiated there.

The dataset was filtered to include only trips in Manhattan. The latitude from Manhattan is between 40.70 to 40.83 and longitude is between -74.025 to -73.93.

```
taxi <- taxi  %>%
    filter(between(lat, 40.70, 40.83), between(long, -74.025, -73.93))
head(taxi)
```

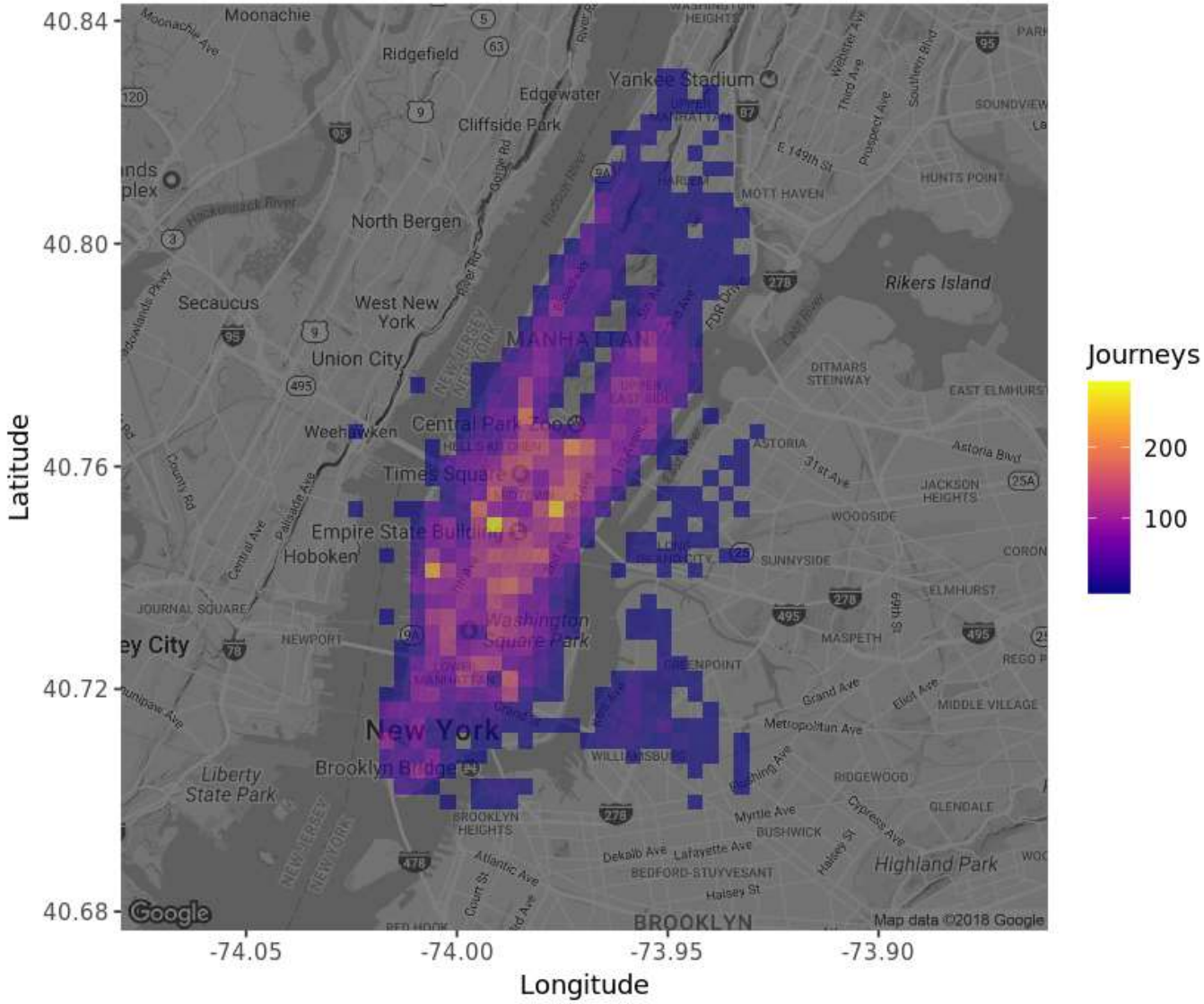
4. Where does the journey begin?

The `ggmap` package together with `ggplot2` to visualize where in Manhattan people tend to start their taxi journeys.

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

# manhattan <- get_map("manhattan", zoom = 12, color = "bw")
manhattan <- readRDS("datasets/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, aes(x=long, y=lat), bins=60, alpha=0.6) +
   labs(x='Longitude', y='Latitude', fill='Journeys')
```



png

5. Predicting taxi fares using a tree

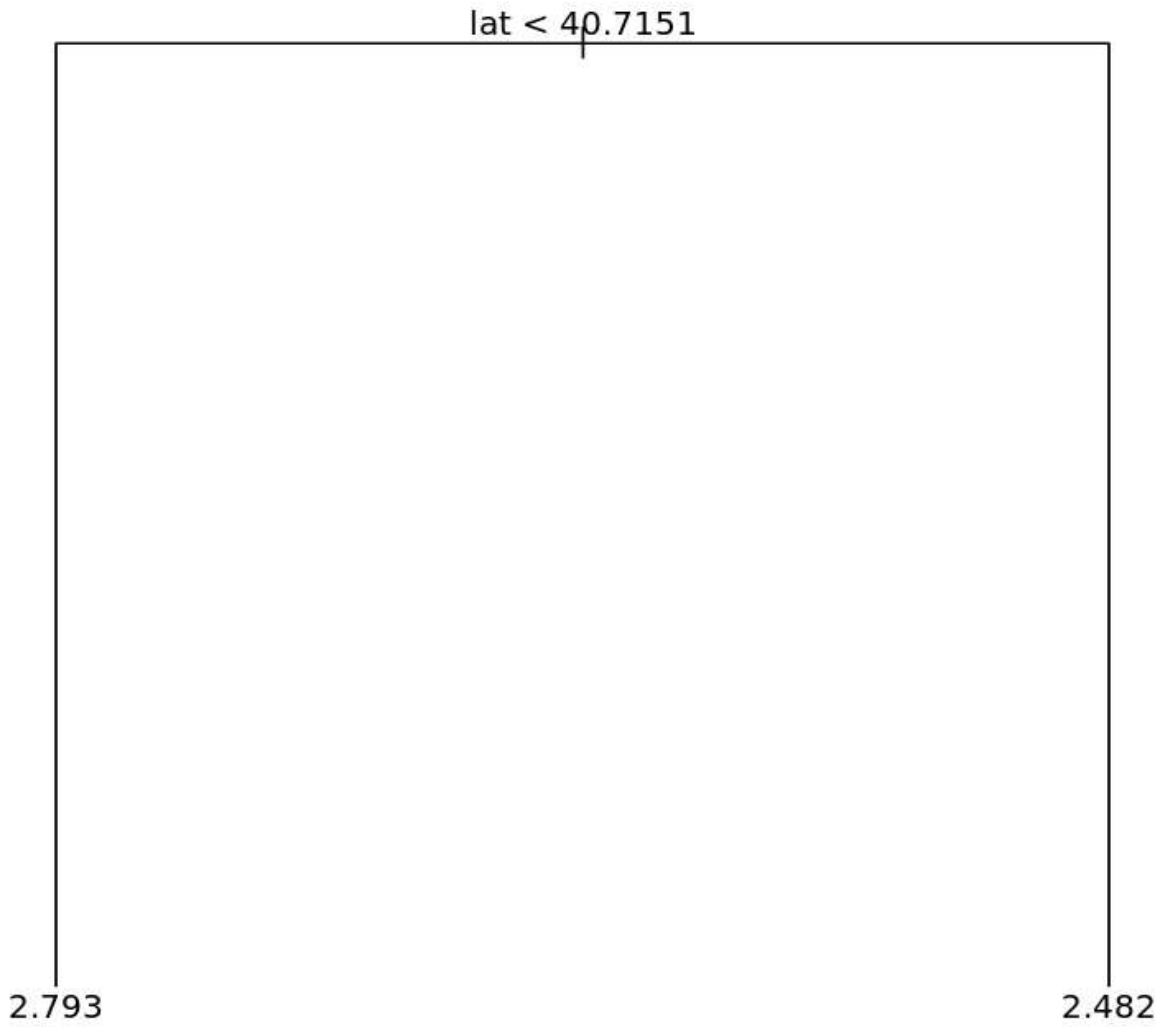
The map from the previous task showed that the journeys are highly concentrated in the business and tourist areas.

Regression tree was used to predict the `total` fare with `lat` and `long` being the predictors. The `tree` algorithm will try to find cutpoints in those predictors that results in the decision tree with the best predictive capability.

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

# 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)
```



png

6. Add More predictors.

The tree above looks a bit frugal, it only includes one split: It predicts that trips where `lat < 40.7237` are more expensive, which makes sense as it is downtown Manhattan. But that's it. It didn't even include `long` as `tree` deemed that it didn't improve the predictions. Taxi drivers will need more information than this and any driver

paying for your data-driven insights would be disappointed with that.

Some more predictors related to the *time* the taxi trip was made were added.

```
library(lubridate)

# 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))
head(taxi)
```

7. One more tree

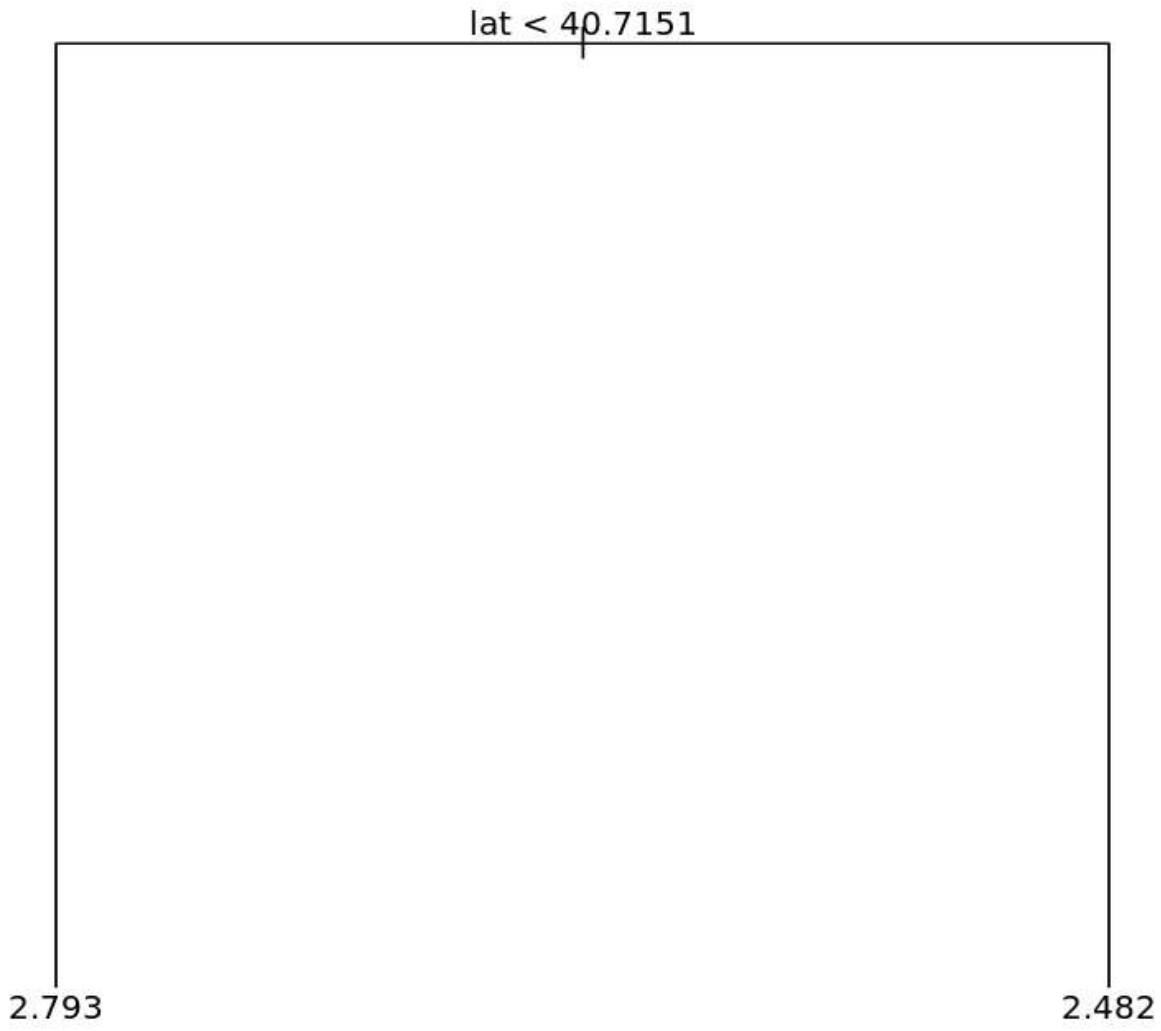
Fit a new regression tree where we include the new time variables.

```
# Fit 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)

# Summarize the performance of the tree
print(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:  0.2676 = 6435 / 24040
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.38300 -0.35370 -0.03954  0.00000  0.30620  2.54200
```



png

8. One tree is not enough

The regression tree has not changed after including the three time variables. This is likely because latitude is still the most promising first variable to split the data on, and after that split, the other variables are not informative enough to be included. A random forest model, where many different trees are fitted to subsets of the data, may well include the other variables in some of the trees that make it up.

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

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

fitted_forest
```

```
Call:
 randomForest(formula = total ~ lat + long + hour + wday + month,      data = taxi, ntree = 8
0, sampsize = 10000)
               Type of random forest: regression
                     Number of trees: 80
No. of variables tried at each split: 1

          Mean of squared residuals: 0.2641192
                    % Var explained: 2.79
```

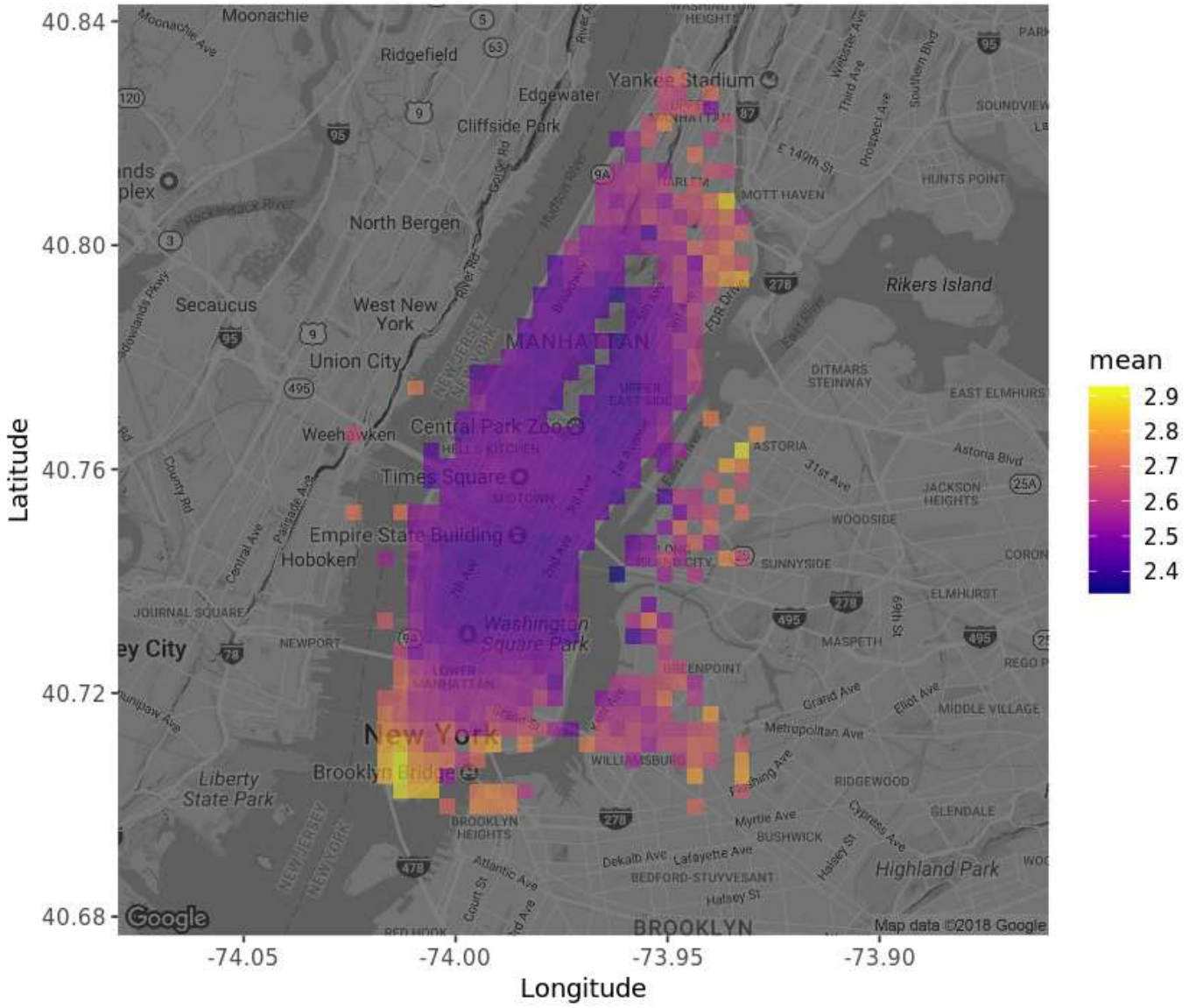
9. Plot the predicted fare

In the output of `fitted_forest` the `Mean of squared residuals`, that is, the average of the squared errors the model makes. Comparing these numbers, show that `fitted_forest` has a slightly lower error. Neither predictive model is *that* good, in statistical terms, they explain only about 3% of the variance.

Now, let's take a look at the predictions of `fitted_forest` projected back onto Manhattan.

```
# Extract the prediction from fitted_forest
taxi$pred_total <- fitted_forest$predicted

# Plot 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, aes(x=long, y=lat, z=pred_total), bins=60, alpha=0.6, fun=mea
n) +
   labs(x='Longitude', y='Latitude', fill='mean')
```



png

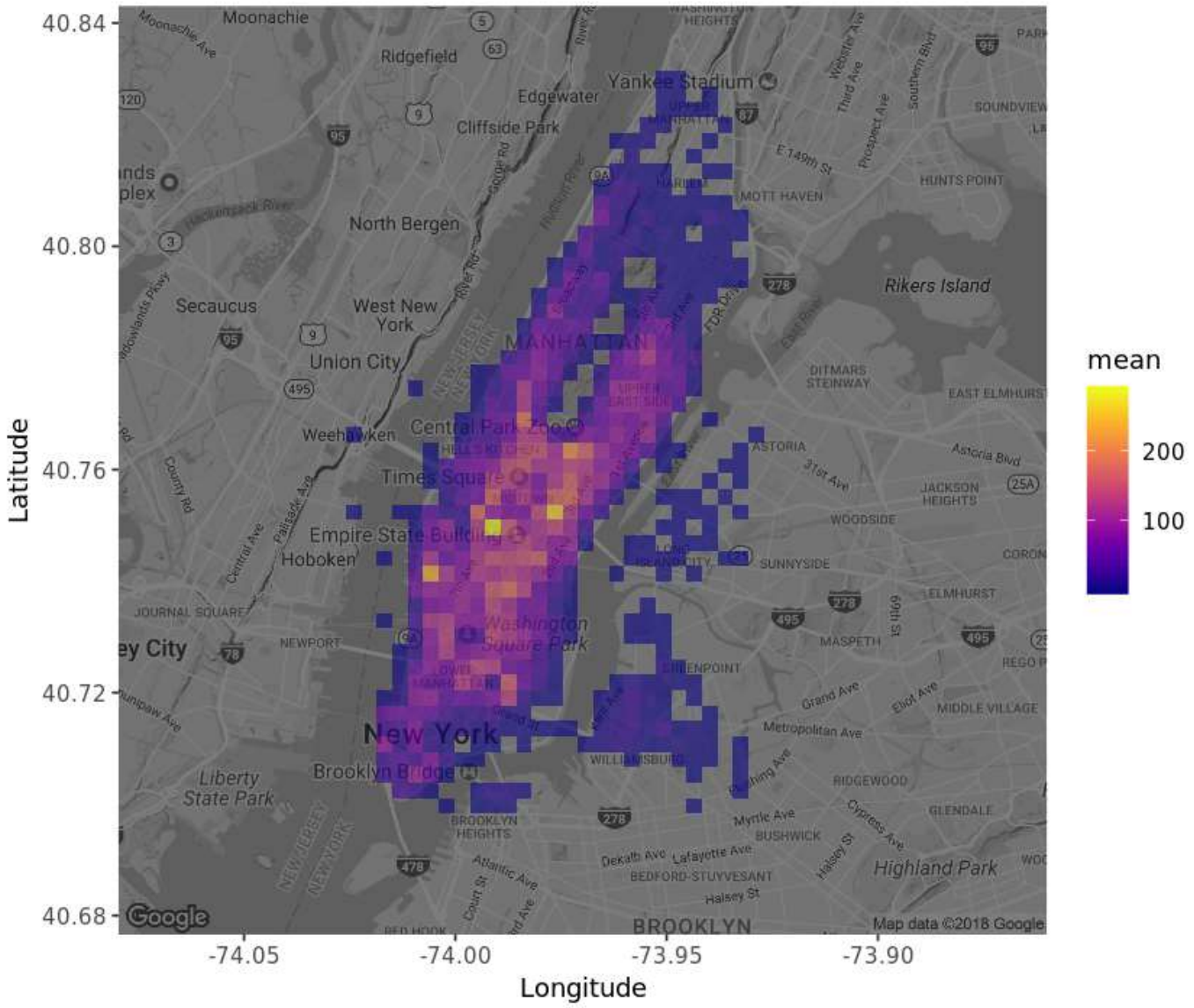
10. Plotting the actual fare

Looking at the map with the predicted fares we see that fares in downtown Manhattan are predicted to be high, while midtown is lower. This map shows the prediction as a function of `lat` and `long` , also plot the predictions over time, or a combination of time and space.

For now, let's compare the map with the predicted fares with a new map showing the mean fares according to the data.

Plot the mean trip prices from the data

```
ggmap(manhattan, darken = 0.5) +
   scale_fill_viridis(option = 'plasma') +
   geom_bin2d(data = taxi, aes(x=long, y=lat, z=total), bins=60, alpha=0.6, fun=mean_if_enough_data) +
   labs(x='Longitude', y='Latitude', fill='mean')
```



png