Airbnb Project

Matthew Coleman, Austin Mac, Jeff Pittman, and Nick Reyes

2/28/2020

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyv
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.2     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------- tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
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
```

Read in the CSV and check the dimensions of the data.

```
bnb <- read.csv('AB_NYC_2019.csv')

bnb <- as_tibble(bnb)

dims <- dim(bnb)

sprintf('Our dataset has %d observations and %d attributes', dims[1],dims[2])
```

```
## [1] "Our dataset has 48895 observations and 16 attributes"
```

Train-test split the data

```
train.ind <- sample(1:nrow(bnb),size = .8*nrow(bnb))


train <- bnb[train.ind,]
test <- bnb[-train.ind,]
```

The response variable for our analysis is going to be `price`, the price per night of the rental. The main predictor variables we are going to explore in this analyses are: `longitude`, `latitude`, `minimum_nights`,

number_of_reviews, reviews_per_month, neighbourhood, neighbourhood_group, room_type, and calculated_host_listings_count.

We can get the column names of the columns which contain NA's with the following code:

```
colnames(train)[apply(train, 2, anyNA)]
```

```
## [1] "reviews_per_month"
```

Need a better description on what reviews per month means

We can see that there are NA reviews in the reviews_per_month, the number of reviews per month. We can also see upon visual inspection last_review, the date of the last review the host received, also contains empty values. We will not be using last_review in our analysis, so we will not worry about imputing values here.

We believe the reason there are NA's in the reviews_per_month column is because the hosts have 0 reviews overall. We further explore this claim the below:

```
with(train, sum((is.na(reviews_per_month)) & (number_of_reviews!=0)) )
```

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

We can see there are no cases where the number_of_reviews and reviews_per_month. As a result, we will impute 0 where reviews_per_month is NA.

```
train[is.na(train$reviews_per_month),'reviews_per_month'] <- 0

sum(is.na(train$reviews_per_month))
```

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

Mention what the room types are in the paper when we describe the variables we are using in the report.

```
levels(train$room_type)
```

```
## [1] "Entire home/apt" "Private room"    "Shared room"
```

```
n_distinct(bnb$neighbourhood)
```

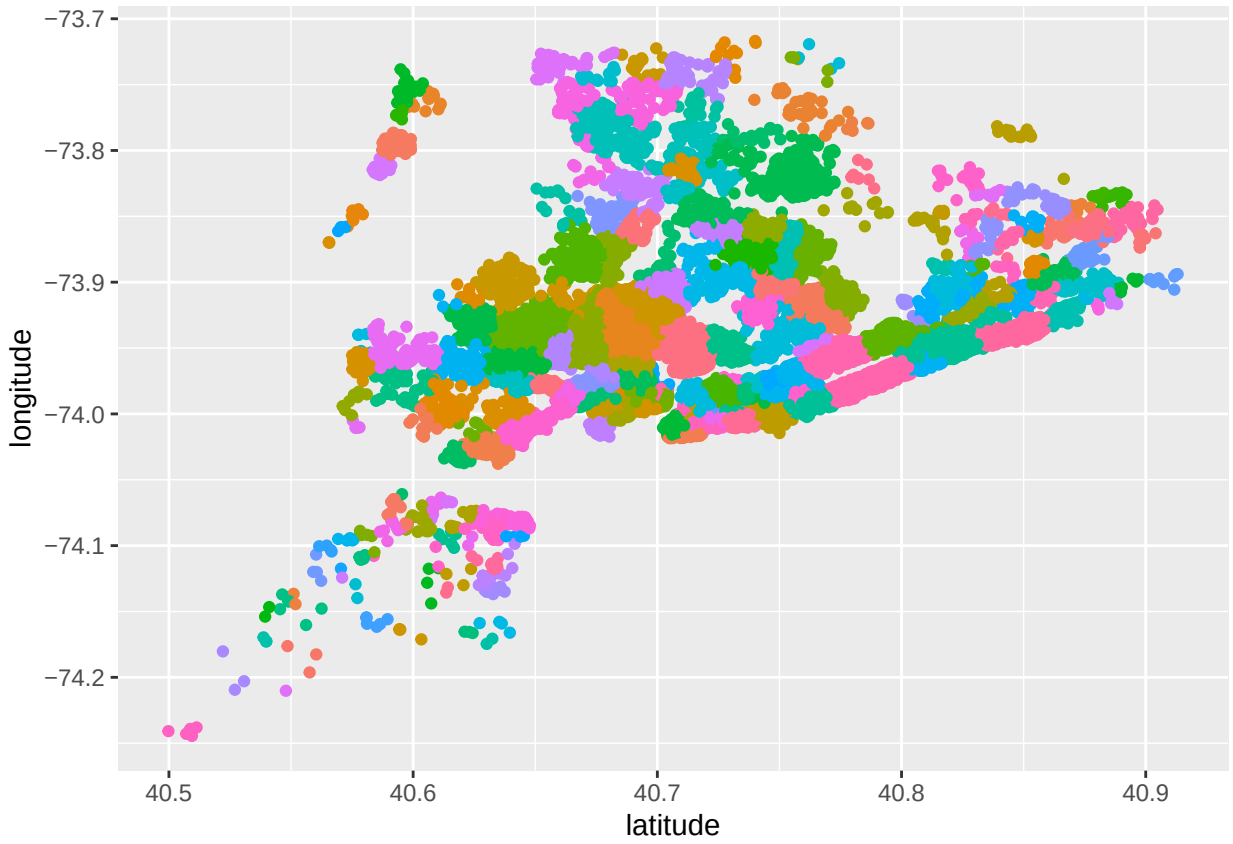
```
## [1] 221
```

```
n_distinct(train$neighbourhood_group)
```

```
## [1] 5
```

There are 221 neighborhoods covered in the overall data, but only 5 neighbourhood groups. We will further investigate whether we need to use the neighbourhood, or whether we would like to use the negihbourhood groups for simplicity of our model.

```
ggplot(data = train, aes(x = latitude, y = longitude, color = neighbourhood)) +
  geom_point() + theme(legend.position="none")
```



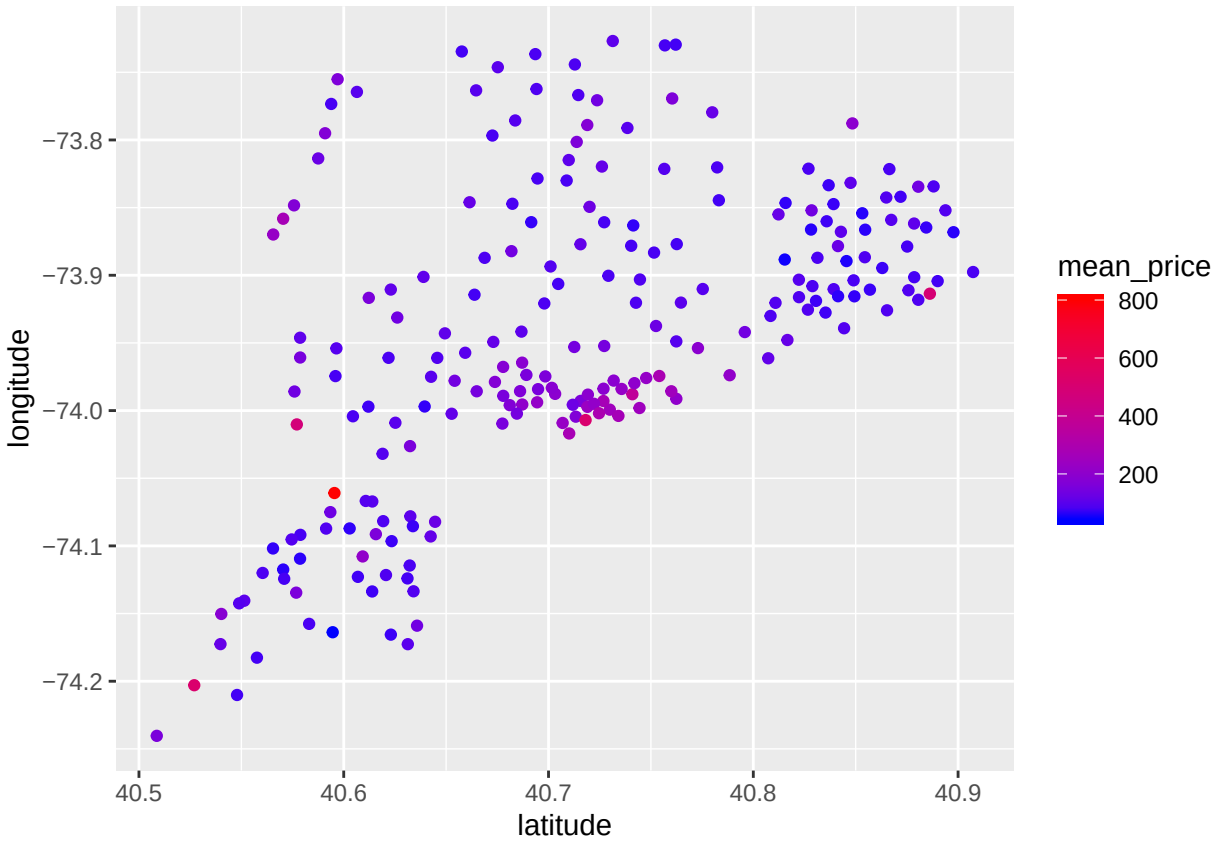

```
ggplot(data = train, aes(x = latitude, y = longitude, color = neighbourhood_group)) +
  geom_point() #+ theme(legend.title = 'Neighbourhood Group')
```

We will determine whether we should use the neigbourhood by seeing if there is a large disparity in mean price by calculating the mean price for the neighbourhood. If there seems to be large disparities within the neighbourhood group for mean pricing, we will attempt to use neighbourhood itself.

```
mean_price <- train %>% group_by(neighbourhood) %>% summarise(mean_price = mean(price),
                                                              latitude = median(latitude), longitude =
                                                              median(longitude))

#plot(mean_price$latitude, mean_price$longitude, col = mean_price$mean_price)

ggplot(data = mean_price, aes(x = latitude, y = longitude, color = mean_price)) +
  geom_point() + scale_color_gradient(low="blue", high="red")
```



It does not seem there are any large disparities in pricing, and all the neighbourhood groups seems to be similar to their nearby neighbours. To reduce the complexity of our model, we will use the neighbourhood group variables.

```
#rest <- randomForest(price ~ longitude + latitude + number_of_reviews + neighbourhood_group, data = tr
#summary(rest)
```