# Processo de Modelagem

## Resumo

Inicialmente, o grupo tentou obter insights por meio das informações presentes no dataset disponibilizado.

Foram feitas algumas análises utilizando plots, médias e exemplos de linhas com seus respectivos dados, o que nos respondeu algumas informações: trata-se de um conjunto de dados de 6 meses, a maioria das corridas só tem um passageiro, a maioria das corridas são extremamente curtas.

## Técnicas e Transformações

### Distâncias Euclidiana e Manhattan

Percebemos que seria possível gerar dados de distâncias dado à latitudes/longitudes de chegada e partida de uma corrida de táxi.

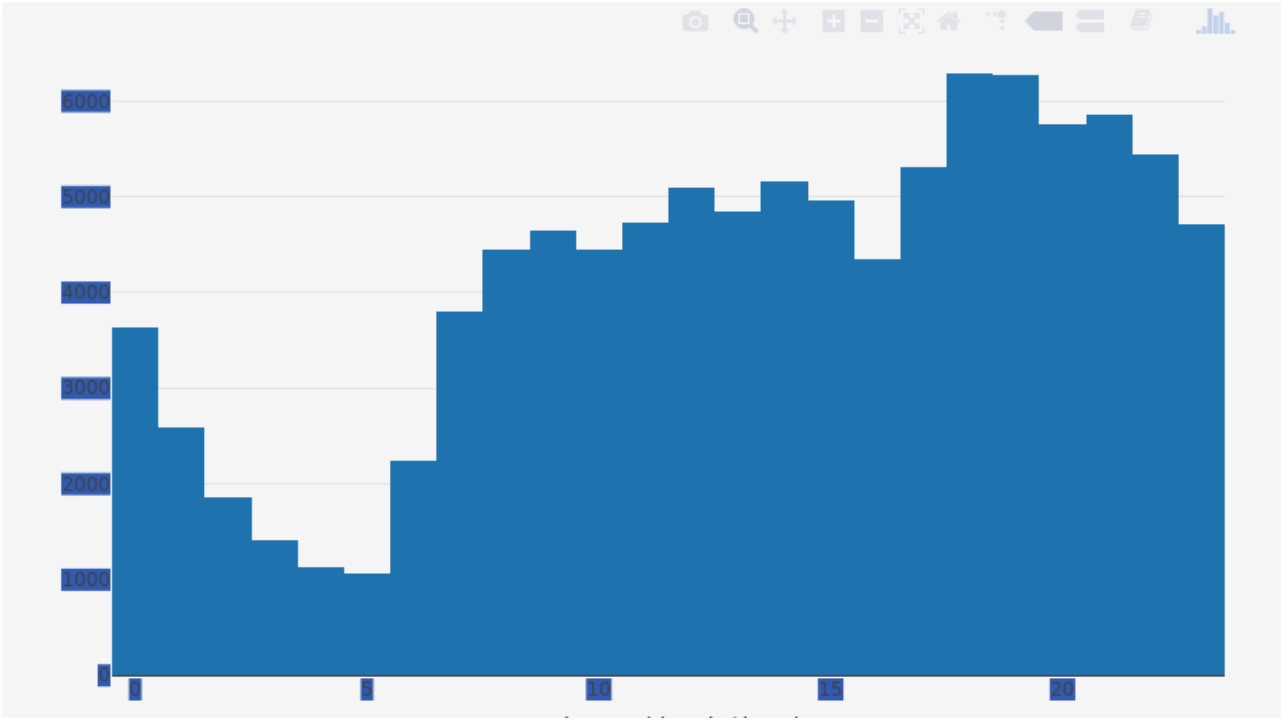
Com isso criamos duas funções com algoritmos que dado estes dados conseguiriam obter as distâncias Euclidiana e Manhattan.

### Dados Temporais

Percebemos também que haviam dados contendo o *timestamp* das corridas, o que graças a isso nos fizeram pensar em algumas perguntas como quantidade de corridas em determinados horários, dias e meses e fizemos algumas perguntas que podem ser vistas no *markdown* contendo análises temporais.

Além disso, agrupamos estas informações com latitude e longitude, assim conseguimos visualizar essas informações em três dimensões e descobrir os horários e regiões com mais pedidas de táxi.

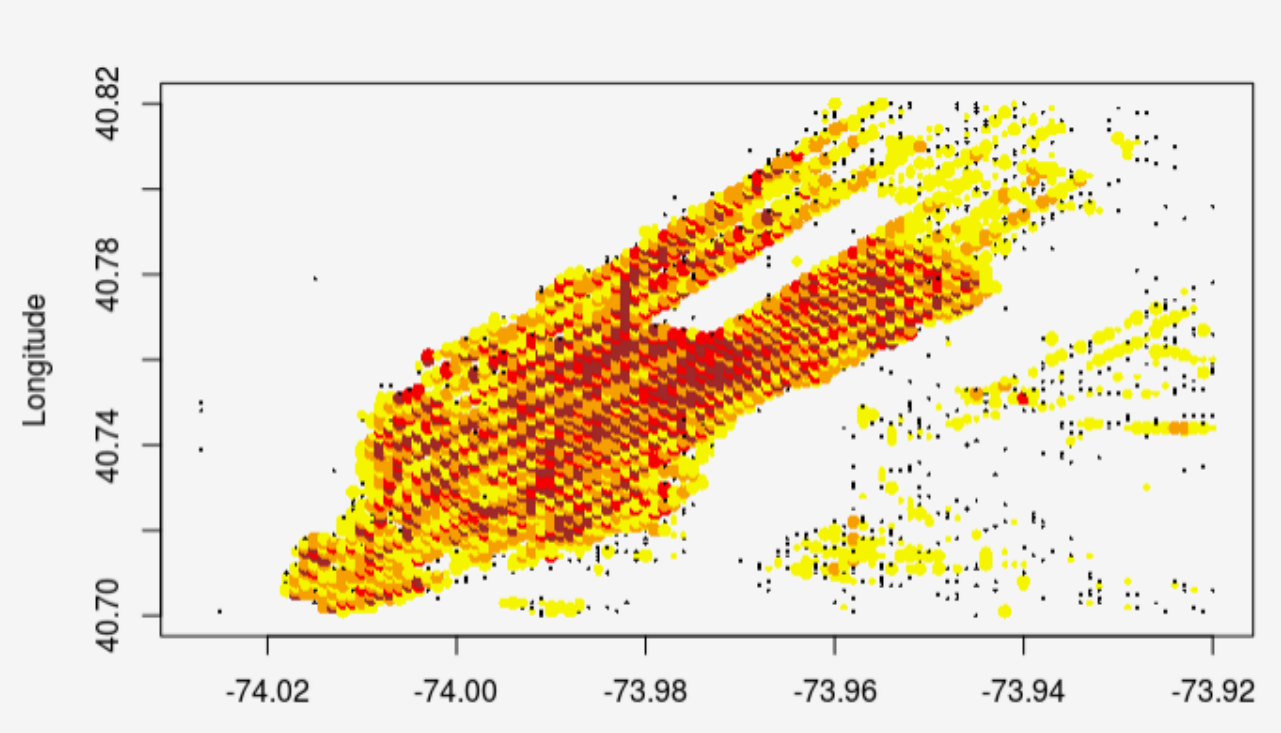
Como resultado, descobrimos que há uma grande concentração de pedidas de  táxi entre os períodos entre 17 e 20h. O grupo tem a hipótese de que este horário muitas pessoas estão saindo do trabalho.




**Regiões com mais pedidas e términos de corrida de táxi**

O grupo queria visualizar as regiões com mais pedidas e términos de corrida de táxi. Como a latitude e longitude estava muito precisa, arredondamos para três casas decimais para poder gerar os plots e mapas de calor.

Também limitamos o corte com maior concentração de dados, limitando somente à região de Manhattan.




**Quadrantes e pontos de interesse**

O grupo considerou interessante saber os principais pontos de interesse. Também dividimos a latitude e longitude em quadrantes arredondados para 3 dígitos. Com isso, poderíamos juntar as informações de corrida que estejam dentro de um quadrante com um quadrante de pontos de interesse.

Para isso, criamos algumas funções que conseguem determinar que dado uma corrida consegue obter os quadrantes que passou até o término da corrida.

**Regressão Linear Simples**

O grupo não explorou tanto a técnica de regressão, somente fazendo uma pergunta: “Existe alguma correlação entre o número de passageiros com a distância percorrida (Quanto maior a distância da corrida, maior a quantidade de passageiros?)

Com a regressão linear, respondemos essa pergunta: não há uma correlação forte entre os dois, logo

Porém, foi demonstrando um exemplo de alta correlação com dois dados obtidos no processo de enriquecimento: distância de Manhattan e distância Euclidiana.

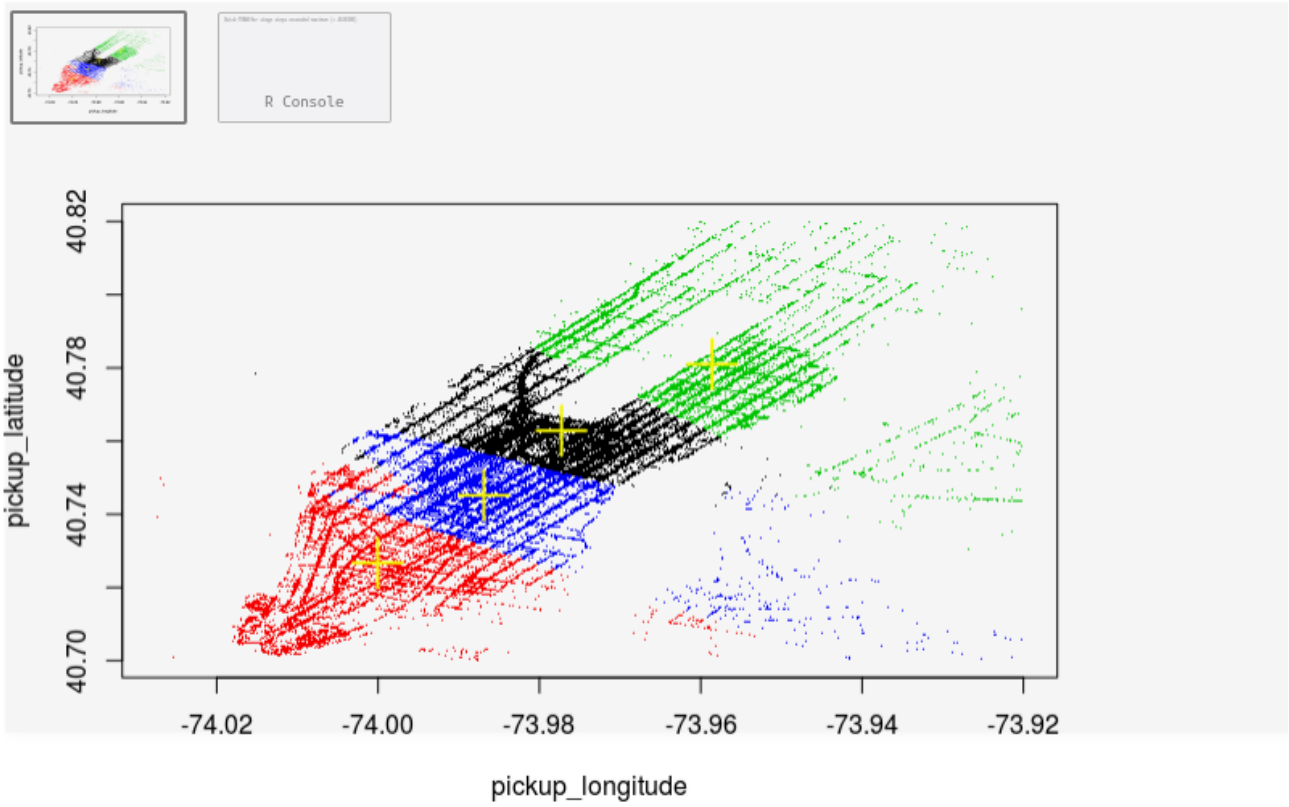
Para validação de conceito, os dois deveriam ter uma alta correlação, pois quanto maior a distância de Manhattan, maior deve ser a distância Euclidiana.

Foi criado também um modelo pra predizer a distância Euclidiana com 95% de assertividade.

**Clusterização por latitude/longitude de início de corrida**

O grupo realizou uma clusterização pela latitude/longitude, para agrupar os pontos de acordo com sua distância.

---




## Variáveis de Entrada

Para criação de um modelo, dividimos em quatro quadrantes: Norte, Sul, Leste e Oeste. A pergunta foi: “Dado uma corrida que iniciou em um local, em qual quadrante ela irá ficar?”

Então, como variáveis de entrada, foi necessário todo enriquecimento feito relacionado aos quadrantes.

## Variáveis de Saída

A saída do modelo deverá prever se a corrida dada a latitude da pedida de corrida ficará no quadrante leste ou não.

O modelo apresentou uma acurácia de aproximadamente 77%.

```{r criando_treinamento}
set.seed(100)
training_indexes <- sample(1:nrow(corridas), 0.7 * nrow(corridas))
training <-  corridas[training_indexes, ]
test <- corridas[-training_indexes, ]
```

```{r}
modelo <- lm( is_pickup_east ~ pickup_x ,data=training)
predicao <- predict(modelo, test)
```

```{r acuracia_modelo}
acuracia <- data.frame(cbind(actuals=test$is_pickup_east, predicteds=predicao))
acuracia_corelacao <- cor(acuracia)
acuracia_corelacao
```

```
             actuals predicteds
actuals    1.0000000  0.7789906
predicteds 0.7789906  1.0000000
```

## Treinamento

O treinamento utilizará 70% do dataset “corridas” e o resto dos dados serão para testes.