

Problema

Este mini projeto surgiu com o intuito de, dado um conjunto de dados de viagens de táxis numa cidade, ser possível calcular quais os pontos/áreas de interesse mais afluentes tirando conclusões sobre os mesmos. Parte-se do pressuposto que normalmente uma pessoa utiliza um táxi quando se quer dirigir para centros de cidades, pontos turísticos ou locais mais específicos para eventos.

Data Set

Chicago Taxi Rides: <https://www.kaggle.com/chicago/chicago-taxi-rides-2016>

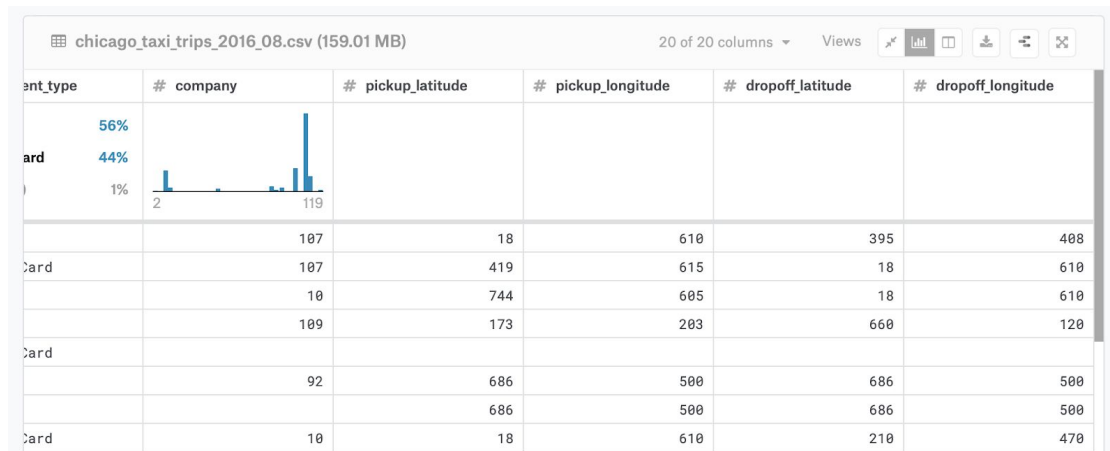
Data (2 GB)		
Data Sources	About this file	Columns
<ul style="list-style-type: none">chicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columnschicago_taxi_trips... 20 columns	January	<ul style="list-style-type: none">taxi_idtrip_start_timestamptrip_end_timestamptrip_secondstrip_milespickup_census_tractdropoff_census_tractpickup_community_areadropoff_community_area

O *data set* original encontra-se disponível em , tendo todos os seus dados contidos num único CSV com mais de 30 GB. O *data set* que está disponível no Kaggle encontra-se já tratado e normalizado ocupando apenas 2 GB, sendo este o principal motivo por esta opção.

O *data set* é composto por 12 ficheiros CSV com os registos dos táxis, sendo que cada ficheiro corresponde a um mês do ano.

Esses ficheiros encontram-se já "normalizados" onde muitos atributos (*pickupLatitude*, *pickupLongitude*, *dropoffLatitude*, *dropoffLongitude*, *company*, *commonArea*) estão definidos através de IDs.

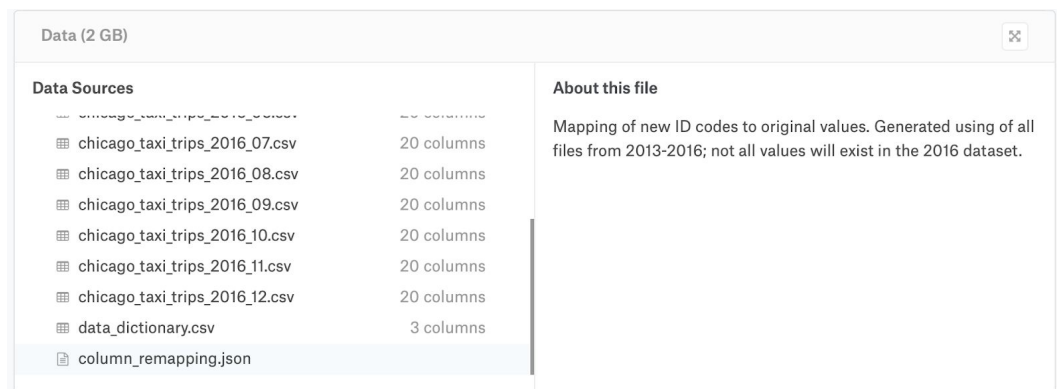
Considerações sobre Lab05



The screenshot shows a data viewer interface for a file named 'chicago_taxi_trips_2016_08.csv' (159.01 MB). It displays 20 columns. A histogram for the 'ent_type' column is shown, with bars for 'card' (56%), 'hard' (44%), and '1' (1%). Below the histogram, a table of data is visible with columns: ent_type, company, pickup_latitude, pickup_longitude, dropoff_latitude, and dropoff_longitude. The table contains several rows of data, including values like 107, 18, 610, 395, 408, 419, 615, 18, 610, 744, 605, 18, 610, 109, 173, 203, 660, 120, 92, 686, 500, 686, 500, 686, 500, 10, 18, 610, 210, 470.

A correspondência desses IDs com o conteúdo que realmente se pretende analisar encontra-se noutros ficheiros CSV que contém esse mapeamento, como é o caso das áreas de partida e chegada.

As latitudes e longitudes dos pontos de partida e destino encontram-se mapeadas num script JSON que faz parte do data set.



The screenshot shows a data viewer interface for a file named 'Data (2 GB)'. It displays a list of data sources on the left and a description of the 'column_remapping.json' file on the right. The data sources include 'chicago_taxi_trips_2016_07.csv', 'chicago_taxi_trips_2016_08.csv', 'chicago_taxi_trips_2016_09.csv', 'chicago_taxi_trips_2016_10.csv', 'chicago_taxi_trips_2016_11.csv', 'chicago_taxi_trips_2016_12.csv', 'data_dictionary.csv', and 'column_remapping.json'. The description of 'column_remapping.json' states: 'Mapping of new ID codes to original values. Generated using of all files from 2013-2016; not all values will exist in the 2016 dataset.'

Analisando o ficheiro JSON disponibilizado, conclui-se que este não se encontra bem formatado, o que tornou difícil a extração dos dados das latitudes e longitudes, como se pode verificar a seguir:

```
"pickup_latitude": [{"0": "41.941422478", "1": "41.920265121", "2": "41.898425258", "3": "42.005608023", "4": "41.884272021", "5": "42.001698194", "6": "41.921273105", "7": "41.946489764", "8": "41.921084583", "9": "41.848179659", "10": "41.697269192", "11": "41.994498038", "12": "41.86958386", "13": "41.905693424",
```

Posto isto, criou-se um *script* python (transformation.py) que transforma estes dados num ficheiro CSV bem formatado, para que seja utilizado como *input* de dados no processo ETL.

Download Data Set Transformado:

https://github.com/cmambuquerque/Assignment1_ED.git

Procedimento

Criação do Modelo Multidimensional

Criou-se um modelo multidimensional com todas as dimensões necessárias à análise do problema e, tendo este diagrama e o diagrama relacional, gerou-se o script DDL para criar a base de dados PostgreSQL.

Script DDL: ChicagoTaxisNew.sql

Data Integration Pentaho

Criaram-se transformações para extrair os dados do *data set*, tratá-los e carregá-los nas respectivas tabelas de dimensão na base de dados local criada.

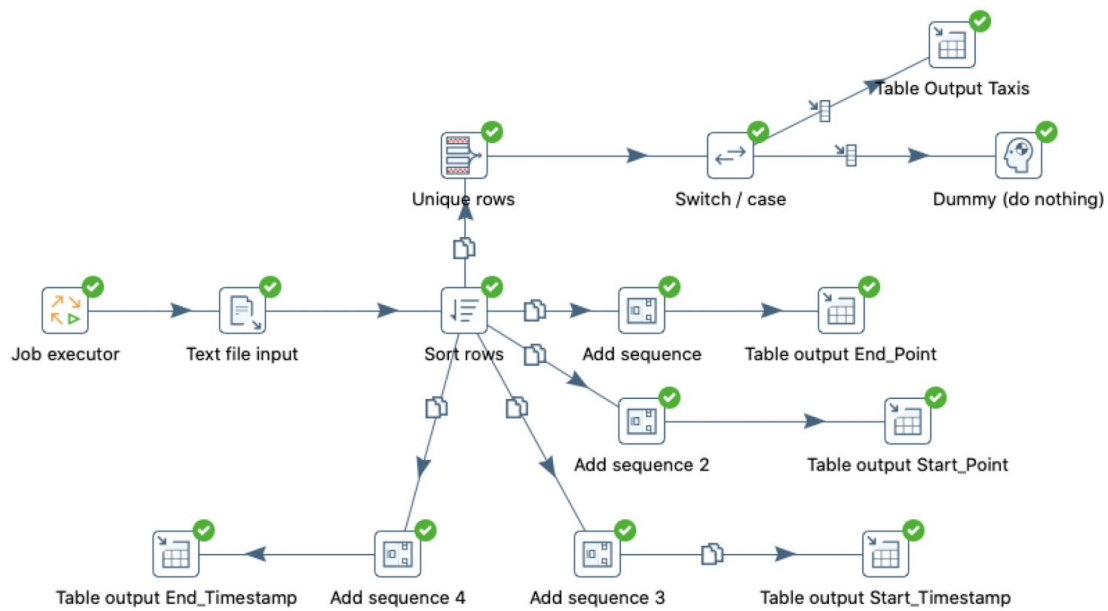
Usou-se o script python referido anteriormente para substituir diretamente os IDs de cada latitude e longitude pelos respectivos valores. Os ficheiros CSV editados encontram-se no diretório /Fixed with lat-long instead of id.

1. Criou-se um Job que faz truncate das tabelas da base de dados.
2. Criou-se uma transformação que faz load dos dados para a dimensão Community_Area (dimensão das áreas)

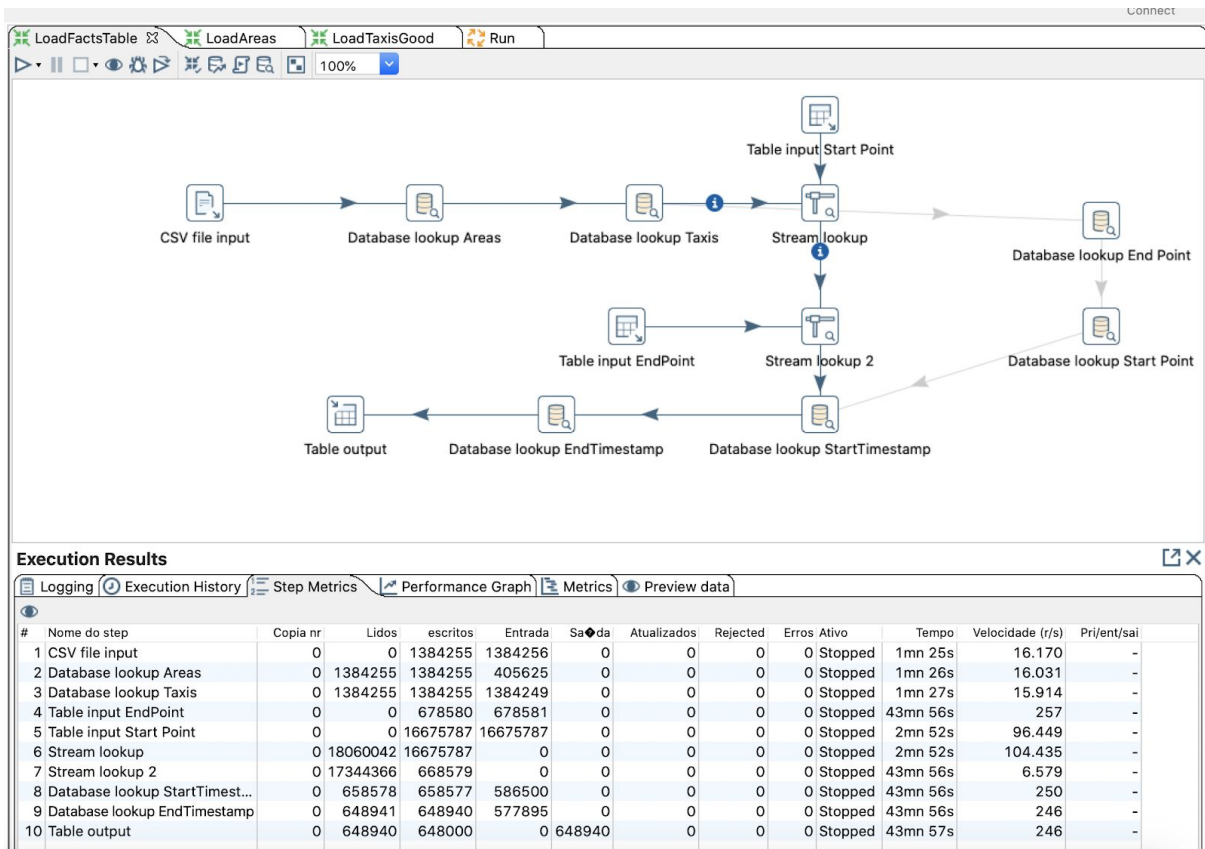


3. Criou-se uma transformação que dos 12 ficheiros CVS recebidos como input, carrega os seus dados nas tabelas de dimensão dos táxis, pontos de partida, pontos de chegada, timestamp da partida e timestamp da chegada. Como a leitura dos 12 ficheiros é um processo pesado e demorado, preferiu-se, com apenas uma leitura de todos os ficheiros, carregar logo todas as tabelas com os respetivos dados.

Considerações sobre Lab05



4. Criou-se uma transformação para carregar os dados para a Fact Table (f_trip), no entanto, não se conseguiu terminar o processo (demora mais de 40 minutos a fazer lookup a uma dimensão).



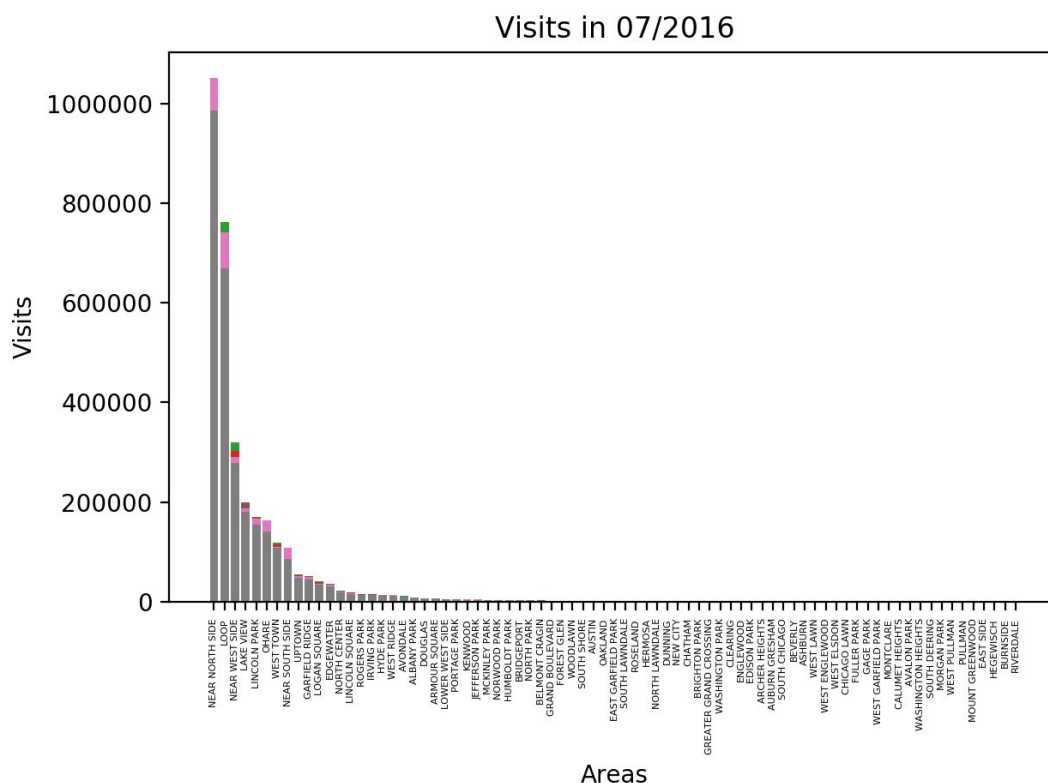
Como não se conseguiu identificar qual o erro, e como foram investidas imensas horas nesta abordagem (cerca de 15h no mínimo), decidiu-se entregar na mesma todo o trabalho desenvolvido.

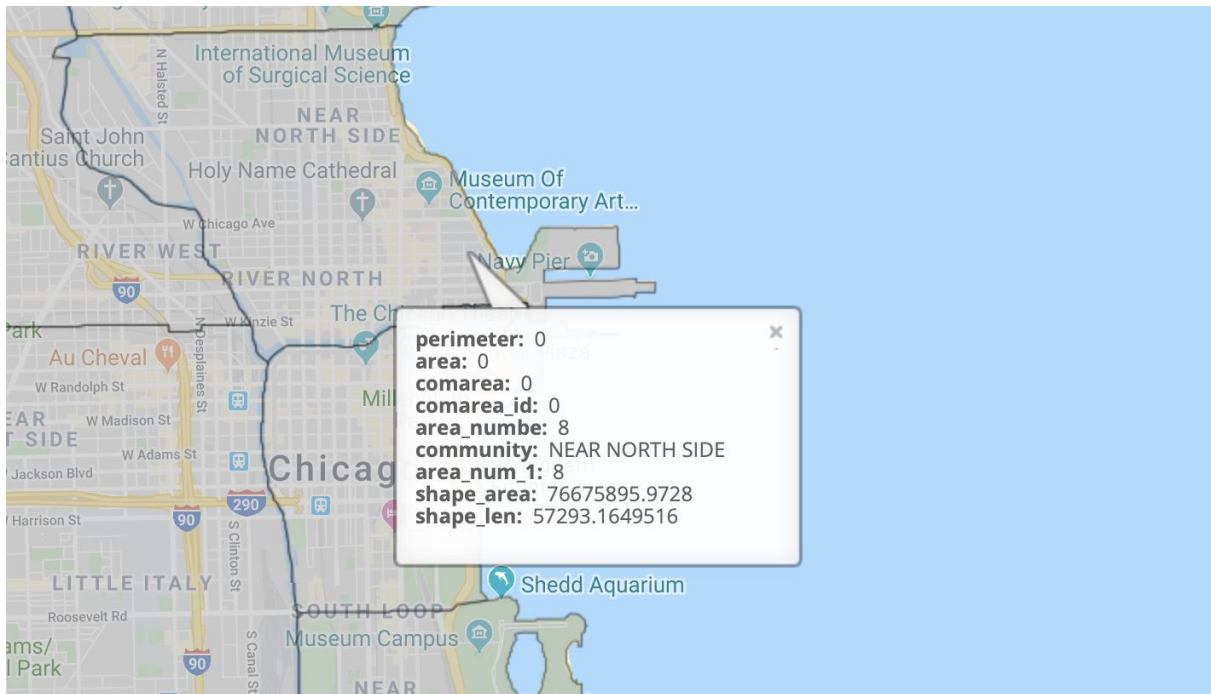
Como solução alternativa e face ao pouco tempo disponível, decidiu-se proceder à análise do *data set* recorrendo à linguagem Python 3 que será explicada no tópico seguinte.

Análise de Dados em Python 3

Usando na mesma os ficheiros CSV tratados anteriormente como forma de não se desperdiçar trabalho, e recorrendo ao ficheiro CSV que contém o mapeamento das áreas de Chicago assim como o seu ID, criou-se um *script* python 3 (*project.py*) que efetua as leituras dos ficheiros *CommAreas.CSV* (armazenando o seu conteúdo num dicionário) e dos 12 ficheiros que contém registos de cada mês do ano de 2016.

Por cada registo, é verificada qual a área de partida e de chegada, contando o número de ocorrências para cada área do *data set*. Esta contagem é depois transposta para um gráfico ordenado que permite visualizar, juntando quer as partidas quer chegadas, o número de visitas à área em questão. Recorreu-se à *library* matplotlib para gerar visualizações gráficas.





Através da leitura do mapa apresentado anteriormente, é visível graficamente que se trata de uma área bastante central na cidade Chicago. É um resultado bastante plausível dado que nessa mesma área encontram-se alguns pontos considerados turísticos, tais como o View Point Chicago 360°, Museu da Arte Contemporânea, a Baixa de Chicago, entre outros.

Conclusões

Apesar da iniciativa do uso do Pentaho Data Integration para implementar o processo ETL sob o *data set* escolhido não ter sido bem sucedida, procurou-se uma alternativa para obter resultados e conclusões face ao problema inicial. O tempo despendido para criar uma solução ao problema recorrendo ao Pentaho foi de facto elevado, cerca de 15 a 20h.

Em contrapartida, após ser reconhecido que talvez não seria possível encontrar resultados com essa implementação para o pouco tempo restante, a implementação de uma solução alternativa, usando uma linguagem de programação bastante familiar aos elementos do grupo, facilitou o processo de análise de dados.

Posto isto, em pouco tempo, obtiveram-se resultados que são considerados fiáveis e realistas.

Apesar das dificuldades encontradas no procedimento para carregar a Fact Table no Pentaho, aprendeu-se o essencial da ferramenta visto foram realizados todos os exercícios do guião 3 com sucesso que implicam o uso da mesma.