Data Wrangling

Nessa etapa foram realizados os procedimentos explicados abaixos:

1) Foi criada uma função auxiliar para identificar (mapear) quais eram as cidades das estações - create_station_map

2) Foi criada uma função principal (summarise_data)com as seguintes funções:

- Unir todos os dados em um único arquivo
- Mapear as estações finais e iniciais para as cidades (utilizando a função "create_station_map" para auxiliar)
- Formatar data e dividi-la em ano, mes, hora e dia da semana
- Selecionar infomações de interesse para a análise: 'duration', 'start_date', 'start_year', 'start_month', 'start_hour', 'weekday', 'start_city', 'end_city', 'subscription_type'

In [1]:

```
# Importa todas as bibliotecas necessárias
%matplotlib inline
import csv
from datetime import datetime
import numpy as np
import pandas as pd
from IPython.display import display
```

In [2]:

```
def create_station_mapping(station_data):
    """
    Cria um mapeamento (tambémm conhecido como de-para) entre a estação
    e a cidade
    """
    # TODO: Inicie esta variável de maneira correta.
    station_map = {}
    for data_file in station_data:
        with open(data_file, 'r') as f_in:
            # configura o objeto csv reader - note que está sendo usado o DictReader,
            # que usa a primeira linha do arquivo como cabeçalho e cria as chaves
            # do dicionário com estes valores.
            weather_reader = csv.DictReader(f_in)

            for row in weather_reader: #row irá percorrer todas as keys do disct weather_reader
                station_id = row['station_id']
                city = row['landmark']
                station_map[station_id] = city
    return station_map
```

In [3]:

```
def summarise_data(trip_in, station_data, trip_out):
    """
    Esta função recebe informações de viagem e estação e produz um novo
    arquivo de dados com um resumo condensado das principais informações de viagem.Os
    argumentos trip_in e station_data serão listas de arquivos de dados para
    as informações da viagem e da estação enquanto trip_out especifica o local
    para o qual os dados sumarizados serão escritos.
    """
    # gera o dicionário de mapeamento entre estações e cidades
    station_map = create_station_mapping(station_data)

    with open(trip_out, 'w') as f_out:
        # configura o objeto de escrita de csv
        out_colnames = ['duration', 'start_date', 'start_year',
                        'start_month', 'start_hour', 'weekday',
                        'start_city', 'end_city', 'subscription_type']
        trip_writer = csv.DictWriter(f_out, fieldnames = out_colnames)
        trip_writer.writeheader()

        for data_file in trip_in:
            with open(data_file, 'r') as f_in:
                # configura o leitor do csv
                trip_reader = csv.DictReader(f_in)

                # processa cada linha lendo uma a uma
                for row in trip_reader:
                    new_point = {}
```

```
            # converte a duração de segundos para minutos.
            ### TODO: Pergunta 3a: Adicione uma operação matemática       ###
            ### para converter a duração de segundos para minutos.  ###
            new_point['duration'] = float(row['Duration'])/float(60)

            # reformate strings com datas para múltiplas colunas
            ### TODO: Pergunta 3b: Preencha os __ abaixo para criar os       ###
            ### campos experados nas colunas (olhe pelo nome da coluna) ###
            trip_date = datetime.strptime(row['Start Date'], '%m/%d/%Y %H:%M')
            new_point['start_date']  = trip_date.strftime('%d/%m/%Y')
            #print '1) ', new_point['start_date']
            new_point['start_year']  = trip_date.year
            #print '2) ', new_point['start_year']
            new_point['start_month'] = trip_date.month
            #print '3) ', new_point['start_month']
            new_point['start_hour']  = trip_date.strftime('%H')
            #print '4) ', new_point['start_hour']
            new_point['weekday']     = trip_date.weekday()
            #print '5) ', new_point['weekday']

            # TODO: mapeia o terminal de inicio e fim com o a cidade de inicio e fim
            new_point['start_city'] =  station_map[row['Start Terminal']]
            #print '6) ', new_point['start_city']
            new_point['end_city'] =  station_map[row['End Terminal']]
            #print '7) ', new_point['end_city']
            # TODO: existem dois nomes diferentes para o mesmo campo. Trate cada um deles.
            #o arquivo csv 201402_trip_data tem 'Subscription_type' e o 201408 tem 'Subscriber type'
            if 'Subscription Type' in row:
                new_point['subscription_type'] = row['Subscription Type']
            else:
                new_point['subscription_type'] = row['Subscriber Type']

            # escreve a informação processada para o arquivo de saída.
            trip_writer.writerow(new_point)
```

Realizando o processamento dos dados

In [4]:

```
#Definindo os dados com informacao sobre as estacoes
station_data = ['201402_station_data.csv',
                '201408_station_data.csv',
                '201508_station_data.csv' ]
#Definindo
trip_in = ['201402_trip_data.csv',
           '201408_trip_data.csv',
           '201508_trip_data.csv' ]
trip_out = 'trip_data.csv'

# Esta função irá ler as informações das estações e das viagens
# e escreverá um arquivo processado com o nome trip_out
summarise_data(trip_in, station_data, trip_out)
```

Visualizando o arquivo 'summary_Bay_Area_Bike_Share.csv'

In [5]:

```
trip_data = pd.read_csv('trip_data.csv')

display(trip_data.head(20))

display(trip_data.tail(20))
```

	duration	start_date	start_year	start_month	start_hour	weekday	start_city	end_city	subscription_type
0	1.050000	29/08/2013	2013	8	14	3	San Francisco	San Francisco	Subscriber
1	1.166667	29/08/2013	2013	8	14	3	San Jose	San Jose	Subscriber
2	1.183333	29/08/2013	2013	8	10	3	Mountain View	Mountain View	Subscriber
3	1.283333	29/08/2013	2013	8	11	3	San Jose	San Jose	Subscriber
4	1.383333	29/08/2013	2013	8	12	3	San Francisco	San Francisco	Subscriber
5	1.716667	29/08/2013	2013	8	18	3	San Francisco	San Francisco	Subscriber
6	1.816667	29/08/2013	2013	8	13	3	San Jose	San Jose	Subscriber
7	1.850000	29/08/2013	2013	8	14	3	San Jose	San Jose	Subscriber

	duration	start_date	start_year	start_month	start_hour	weekday	start_city	end_city	subscription_type
8	1.883333	29/08/2013	2013	8	17	3	San Francisco	San Francisco	Subscriber
9	1.900000	29/08/2013	2013	8	11	3	San Jose	San Jose	Subscriber
10	2.083333	29/08/2013	2013	8	13	3	San Francisco	San Francisco	Subscriber
11	2.100000	29/08/2013	2013	8	13	3	San Jose	San Jose	Subscriber
12	2.150000	29/08/2013	2013	8	19	3	Mountain View	Mountain View	Subscriber
13	2.166667	29/08/2013	2013	8	13	3	San Francisco	San Francisco	Subscriber
14	2.233333	29/08/2013	2013	8	12	3	San Francisco	San Francisco	Subscriber
15	2.300000	29/08/2013	2013	8	16	3	San Francisco	San Francisco	Subscriber
16	2.350000	29/08/2013	2013	8	11	3	San Jose	San Jose	Subscriber
17	2.366667	29/08/2013	2013	8	12	3	San Francisco	San Francisco	Subscriber
18	2.366667	29/08/2013	2013	8	22	3	San Francisco	San Francisco	Subscriber
19	2.400000	29/08/2013	2013	8	22	3	San Francisco	San Francisco	Subscriber

	duration	start_date	start_year	start_month	start_hour	weekday	start_city	end_city	subscription_type
669939	25.600000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Customer
669940	25.750000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Customer
669941	21.500000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Subscriber
669942	10.500000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Subscriber
669943	5.550000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Subscriber
669944	115.616667	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Customer
669945	7.500000	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Subscriber
669946	2.683333	01/09/2014	2014	9	8	0	San Francisco	San Francisco	Subscriber
669947	289.933333	01/09/2014	2014	9	7	0	Mountain View	Mountain View	Customer
669948	288.283333	01/09/2014	2014	9	7	0	Mountain View	Mountain View	Customer
669949	2.816667	01/09/2014	2014	9	7	0	San Francisco	San Francisco	Subscriber
669950	94.450000	01/09/2014	2014	9	7	0	San Jose	San Jose	Customer
669951	7.350000	01/09/2014	2014	9	6	0	San Francisco	San Francisco	Subscriber
669952	6.633333	01/09/2014	2014	9	5	0	San Francisco	San Francisco	Subscriber
669953	4.000000	01/09/2014	2014	9	4	0	San Francisco	San Francisco	Subscriber
669954	10.316667	01/09/2014	2014	9	4	0	San Francisco	San Francisco	Subscriber
669955	111.866667	01/09/2014	2014	9	3	0	San Francisco	San Francisco	Customer
669956	8.966667	01/09/2014	2014	9	0	0	San Francisco	San Francisco	Customer
669957	9.466667	01/09/2014	2014	9	0	0	San Francisco	San Francisco	Customer
669958	9.483333	01/09/2014	2014	9	0	0	San Francisco	San Francisco	Customer