In [41]:

```
import pandas as pd
import os
```

In [42]:

```
def load_csv_as_df(file_name, sub_directories, column_numbers=None, column_names=None):
    '''
    Load any csv as a pandas dataframe. Provide the filename, the subdirectories, and col
umns to read(if desired).
    '''
    base_path = os.getcwd()
    full_path = base_path + sub_directories + file_name

    if column_numbers is not None:
        df = pd.read_csv(full_path, usecols=column_numbers)
    else:
        df = pd.read_csv(full_path)

    if column_names is not None:
        df.columns = column_names

    return df
```

In [79]:

```
def lookup(s):
    """
    This is an extremely fast approach to datetime parsing.
    For large data, the same dates are often repeated. Rather than
    re-parse these, we store all unique dates, parse them, and
    use a lookup to convert all dates.
    """
    dates = {date: pd.to_datetime(date) for date in s.unique()}
    return s.map(dates)


def label_trajectories(df, trajectory_number):
    df['time'] = lookup(df['time']) # add time for sorting
    updated_dfs = []
    taxi_ids = df['taxi_id'].unique()
    print('There are ', len(taxi_ids), ' unique taxi ids in this data')

    empty_route = -1
    completed_count = 0

    for taxi_id in taxi_ids:
        # get the df for that taxi
        taxi_df = df.loc[df['taxi_id'] == taxi_id]
        taxi_df.sort_values(by=['time'], inplace=True)
        passenger_got_in = False

        route_numbers = []
        route_starts = []
        route_ends = []
        relevant_starts = []
        relevant_ends = []

        airport_starts = []
        airport_ends = []
        bus_starts = []
        bus_ends = []

        for index, row in taxi_df.iterrows():
            passenger_in_taxi = row['occupancy_status']

            # Do we already have a passenger?
```

```
            if passenger_got_in:
                if passenger_in_taxi:
                    # trajectory still going
                    route_starts.append(False)
                    route_ends.append(False)
                    relevant_ends.append(False)
                    relevant_starts.append(False)
                    bus_starts.append(False)
                    airport_starts.append(False)
                    bus_ends.append(False)
                    airport_ends.append(False)
                    route_numbers.append(trajectory_number)
                    continue
                elif not passenger_in_taxi:
                    # trajectory ended
                    passenger_got_in = False
                    route_starts.append(False)
                    route_ends.append(True)
                    route_numbers.append(trajectory_number)
                    trajectory_number += 1

                    # Is this relevant?
                    end_lat = row['latitude']
                    end_long = row['longitude']

                    if near_airport(end_lat, end_long) or near_bus_station(end_lat, end_
long):
                        relevant_ends.append(True)

                        if near_airport(end_lat, end_long):
                            airport_ends.append(True)
                            bus_ends.append(False)
                        else:
                            airport_ends.append(False)
                            bus_ends.append(True)

                    else:
                        relevant_ends.append(False)
                        airport_ends.append(False)
                        bus_ends.append(False)

                    relevant_starts.append(False)
                    airport_starts.append(False)
                    bus_starts.append(False)

            elif passenger_in_taxi:
                # someone just got in
                passenger_got_in = True
                route_starts.append(True)
                route_ends.append(False)
                route_numbers.append(trajectory_number)
                # is this relevant?

                start_lat = row['latitude']
                start_long = row['longitude']

                if near_airport(start_lat, start_long) or near_bus_station(start_lat, st
art_long):
                    relevant_starts.append(True)

                    if near_airport(start_lat, start_long):
                        airport_starts.append(True)
                        bus_starts.append(False)
                    else:
                        bus_starts.append(True)
                        airport_starts.append(False)

                else:
                    relevant_starts.append(False)
                    airport_starts.append(False)
                    bus_starts.append(False)
```

```
                relevant_ends.append(False)
                airport_ends.append(False)
                bus_ends.append(False)

            else:
                # driving around without no passenger
                route_starts.append(False)
                route_ends.append(False)
                relevant_ends.append(False)
                relevant_starts.append(False)
                bus_starts.append(False)
                airport_starts.append(False)
                bus_ends.append(False)
                airport_ends.append(False)
                route_numbers.append(empty_route)

        taxi_df['route_number'] = route_numbers
        taxi_df['route_start'] = route_starts
        taxi_df['route_end'] = route_ends
        taxi_df['relevant_start'] = relevant_starts
        taxi_df['relevant_end'] = relevant_ends
        taxi_df['airport_start'] = airport_starts
        taxi_df['airport_end'] = airport_ends
        taxi_df['bus_start'] = bus_starts
        taxi_df['bus_end'] = bus_ends

        taxi_df = taxi_df[taxi_df.route_number != -1]
        updated_dfs.append(taxi_df)
        completed_count += 1

        if completed_count % 1000 == 0:
            print('Completed ', completed_count, ' taxi_ids out of ', len(taxi_ids))

    return pd.concat(updated_dfs), trajectory_number


def find_trajectories_at_airport_or_bus(df):
    relevant_starts_df = df[df['relevant_start'] == True]
    relevant_ends_df = df[df['relevant_end'] == True]

    relevant_start_numbers = relevant_starts_df.route_number.unique()
    relevant_end_numbers = relevant_ends_df.route_number.unique()

    intersection_numbers = list(set(relevant_start_numbers) & set(relevant_end_numbers))

    print('Found ', len(intersection_numbers), ' relevant routes!')

    return df[df['route_number'].isin(intersection_numbers)]


def near_airport(lat, long):
    if 22.605770 <= lat <= 22.667089 and 113.784647 <= long <= 113.837340:
        return True
    else:
        return False


def near_bus_station(lat, long):
    if 22.567210 <= lat <= 22.568807 and 114.089676 <= long <= 114.091320:
        return True
    else:
        return False
```

In [61]:

```
def load_data_and_find_relevant_routes(file_name, sub_directories, trajectory_number):
    col_numbers = [3, 4, 5, 6, 7, 8, 12]
    col_names = ['longitude', 'latitude', 'time', 'taxi_id', 'speed', 'direction', 'occu
pancy_status']

    df = load_csv_as_df(file_name, sub_directories, col_numbers, col_names)
```

```
    df, new_trajectory_number = label_trajectories(df, trajectory_number)

    relevant_df = find_trajectories_at_airport_or_bus(df)

    print('Found ', len(relevant_df), ' relevant routes in ', file_name)

    return relevant_df, new_trajectory_number


def load_all_data_from(folder_name, number_of_files):
    trajectory_number = 1
    base_file_name = 'part-m-'
    relevant_dfs = []

    for i in range(0, number_of_files):

        if i < 10:
            file_number = '0000' + str(i)
        else:
            file_number = '000' + str(i)

        file_name = base_file_name + file_number
        df, new_trajectory_number = load_data_and_find_relevant_routes(file_name, folder
_name, trajectory_number)

        relevant_dfs.append(df)
        trajectory_number = new_trajectory_number

        print('new_trajectory_number: ', new_trajectory_number)

    return relevant_dfs
```

In [89]:

```
%%time
col_numbers = [3, 4, 5, 6, 7, 8, 12]
col_names = ['longitude', 'latitude', 'time', 'taxi_id', 'speed', 'direction', 'occupanc
y_status']
df = load_csv_as_df('part-m-00037', '/2014-04-06/', col_numbers, col_names)
```

```
CPU times: user 1.45 s, sys: 162 ms, total: 1.61 s
Wall time: 1.68 s
```

In [90]:

```
%%time
df, trajectory_count = label_trajectories(df, 1)
```

```
There are  4215  unique taxi ids in this data
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:128: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:129: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:130: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:131: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:132: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:133: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:134: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:135: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:136: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
```

```
Completed  1000  taxi_ids out of  4215
Completed  2000  taxi_ids out of  4215
Completed  3000  taxi_ids out of  4215
Completed  4000  taxi_ids out of  4215
CPU times: user 31min 14s, sys: 13.2 s, total: 31min 28s
Wall time: 33min 14s
```

In [106]:

```
df.head()
```

Out[106]:

	longitude	latitude	time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	relevant_start	r

In [98]:

```
air_to_train_df = df[(df['airport_start'] == True) & (df['train_end'] == True)]
print(len(air_to_train_df))
```

```
0
```

In [99]:

```
train_to_air_df = df[(df['train_start'] == True) & (df['airport_end'] == True)]
print(len(train_to_air_df))
```

0

In [100]:

```
air_start = df[df['airport_start'] == True]
print(len(air_start))
```

0

In [101]:

```
air_end = df[df['airport_end'] == True]
print(len(air_end))
```

0

In [102]:

```
bus_start = df[df['bus_start'] == True]
print(len(bus_start))
```

0

In [103]:

```
bus_end = df[df['bus_end'] == True]
print(len(bus_end))
```

0

In [104]:

```
relevant_starts_df = df[df['relevant_start'] == True]
print(len(relevant_starts_df))
```

0

In [105]:

```
relevant_starts_df.head()
```

Out[105]:

	longitude	latitude	time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	relevant_start	r

In [55]:

```
print(near_airport(22.618299, 113.814003))
```

True

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [49]:

```
relevant_ends_df = df[df['relevant_end'] == True]
print(len(relevant_ends_df))
```

```
110
```

In [50]:

```
relevant_start_numbers = relevant_starts_df.route_number.unique()
relevant_end_numbers = relevant_ends_df.route_number.unique()

intersection_numbers = list(set(relevant_start_numbers) & set(relevant_end_numbers))

print('Found ', len(intersection_numbers), ' relevant routes!')
```

```
Found  20  relevant routes!
```

In [51]:

```
print(intersection_numbers)
```

```
[4160, 4737, 6720, 4987, 2340, 9313, 9222, 4391, 7752, 7014, 5224, 4273, 6897, 6898, 1877
, 4726, 9337, 4123, 1660, 8543]
```

In [52]:

```
success = df[df['route_number'].isin(intersection_numbers)]
print(len(success))
```

```
59
```

In [53]:

```
success.head(30)
```

Out[53]:

	longitude	latitude	time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	r
18378	113.805618	22.666817	2014-04-06 09:37:04	1299922	2	128	1	1660	True	False	
18381	113.817986	22.650917	2014-04-06 09:40:32	1299922	16	142	0	1660	False	True	
20883	113.818115	22.612150	2014-04-06 06:43:34	1298038	61	277	1	1877	True	False	
20882	113.811951	22.622900	2014-04-06 06:44:52	1298038	58	20	1	1877	False	False	
20881	113.809464	22.627518	2014-04-06 06:46:23	1298038	0	239	1	1877	False	False	
20884	113.822716	22.613716	2014-04-06 06:51:19	1298038	62	62	0	1877	False	True	
26883	113.811935	22.627434	2014-04-06 07:34:58	1298801	65	332	1	2340	True	False	
26884	113.812851	22.612650	2014-04-06 07:38:01	1298801	66	152	0	2340	False	True	
47093	113.809235	22.627300	2014-04-06	1319332	30	242	1	4123	True	False	

	longitude	latitude	06:05:50 time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	r
47109	113.812035	22.614117	2014-04-06 06:09:02	1319332	57	152	0	4123	False	True	
47490	113.824135	22.614500	2014-04-06 06:15:33	1319294	77	243	1	4160	True	False	
47489	113.812332	22.623449	2014-04-06 06:17:29	1319294	61	33	1	4160	False	False	
47488	113.810951	22.628151	2014-04-06 06:18:09	1319294	45	278	1	4160	False	False	
47487	113.809532	22.627550	2014-04-06 06:18:28	1319294	15	237	1	4160	False	False	
47491	113.824730	22.614683	2014-04-06 06:24:20	1319294	72	64	0	4160	False	True	
49063	113.824730	22.614901	2014-04-06 17:55:34	1319330	71	242	1	4273	True	False	
49062	113.813187	22.623949	2014-04-06 18:03:24	1319330	0	15	0	4273	False	True	
50454	113.808220	22.626917	2014-04-06 06:31:51	1319367	0	231	1	4391	True	False	
50455	113.808220	22.626917	2014-04-06 06:32:11	1319367	0	231	0	4391	False	True	
54118	113.822464	22.641951	2014-04-06 17:21:51	1319347	32	161	1	4726	True	False	
54127	113.825104	22.637667	2014-04-06 17:22:30	1319347	52	140	1	4726	False	False	
54131	113.829681	22.632999	2014-04-06 17:23:11	1319347	16	17	1	4726	False	False	
54105	113.812920	22.626034	2014-04-06 17:53:28	1319347	0	336	0	4726	False	True	
54132	113.829765	22.655367	2014-04-06 21:38:55	1319347	95	343	1	4737	True	False	
54128	113.828949	22.657883	2014-04-06 21:39:06	1319347	87	340	0	4737	False	True	
57588	113.814850	22.616484	2014-04-06 07:42:54	1319315	65	332	1	4987	True	False	
57587	113.814552	22.617050	2014-04-06 07:42:59	1319315	66	332	1	4987	False	False	
57586	113.812950	22.619967	2014-04-06 07:43:18	1319315	69	333	1	4987	False	False	
57584	113.806183	22.624666	2014-04-06 07:46:38	1319315	47	154	0	4987	False	True	
			2014-								

	longitude	latitude	time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	r
60751	113.833618	22.618668	04-06 16:21:26	1314795	0	60	1	5224	True	False	

In [ ]:

```
```

In [34]:

```
start_df = df[df['route_start'] == True]
print(len(start_df))
```

```
10096
```

In [35]:

```
# lat_start = start_df[start_df['latitude'] > 22.567210 &&]
lat_start = start_df[(start_df['latitude'] >= 22.567210) & (start_df['latitude'] <= 22.5
68807)]
print(len(lat_start))
```

```
104
```

In [36]:

```
lat_and_long_df = lat_start[(lat_start['longitude'] >= 114.089676) & (lat_start['longitu
de'] <= 114.091320)]
```

In [37]:

```
print(len(lat_and_long_df))
```

```
2
```

In [38]:

```
lat_and_long_df.head()
```

Out[38]:

	longitude	latitude	time	taxi_id	speed	direction	occupancy_status	route_number	route_start	route_end	r
62745	114.090103	22.567568	2014-04-06 22:30:44	1316435	63	254	1	5397	True	False	
84785	114.090416	22.567301	2014-04-06 02:27:04	1316471	74	253	1	7111	True	False	

In [24]:

```
%%time
all_relevant_df = load_all_data_from('/2014-04-06/', 2)
```

```
There are  4510  unique taxi ids in this data
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:91: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:92: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:93: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:94: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/ipykernel_l
auncher.py:95: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexin
g.html#indexing-view-versus-copy
```

```
Completed  1000  taxi_ids out of  4510
Completed  2000  taxi_ids out of  4510
Completed  3000  taxi_ids out of  4510
Completed  4000  taxi_ids out of  4510
Found  0  relevant routes!
Found  0  relevant routes in  part-m-00000
new_trajectory_number:  58455
There are  4221  unique taxi ids in this data
Completed  1000  taxi_ids out of  4221
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<timed exec> in <module>()

<ipython-input-23-a5c7c933b460> in load_all_data_from(folder_name, number_of_files)
     26
     27         file_name = base_file_name + file_number
---> 28         df, new_trajectory_number = load_data_and_find_relevant_routes(file_name,
folder_name, trajectory_number)
     29
     30         relevant_dfs.append(df)

<ipython-input-23-a5c7c933b460> in load_data_and_find_relevant_routes(file_name, sub_dire
ctories, trajectory_number)
      4
      5     df = load_csv_as_df(file_name, sub_directories, col_numbers, col_names)
----> 6     df, new_trajectory_number = label_trajectories(df, trajectory_number)
      7
      8     relevant_df = find_trajectories_at_airport_or_bus(df)

<ipython-input-22-19c4d44a9f71> in label_trajectories(df, trajectory_number)
     31         relevant_ends = []
     32
---> 33         for index, row in taxi_df.iterrows():
     34             passenger_in_taxi = row['occupancy_status']
     35

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/pandas/core
/frame.py in iterrows(self)
    746         klass = self._constructor_sliced
    747         for k, v in zip(self.index, self.values):
--> 748             s = klass(v, index=columns, name=k)
    749             yield k, s
    750
```

```
    750
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/pandas/core
/series.py in __init__(self, data, index, dtype, name, copy, fastpath)
    264                                         raise_cast_failure=True)
    265
--> 266                 data = SingleBlockManager(data, index, fastpath=True)
    267
    268         generic.NDFrame.__init__(self, data, fastpath=True)
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/pandas/core
/internals.py in __init__(self, block, axis, do_integrity_check, fastpath)
   4400         if not isinstance(block, Block):
   4401             block = make_block(block, placement=slice(0, len(axis)), ndim=1,
-> 4402                                fastpath=True)
   4403
   4404         self.blocks = [block]
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/pandas/core
/internals.py in make_block(values, placement, klass, ndim, dtype, fastpath)
   2955                         placement=placement, dtype=dtype)
   2956
-> 2957     return klass(values, ndim=ndim, fastpath=fastpath, placement=placement)
   2958
   2959 # TODO: flexible with index=None and/or items=None
```

```
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/pandas/core
/internals.py in __init__(self, values, ndim, fastpath, placement, **kwargs)
   2080
   2081         super(ObjectBlock, self).__init__(values, ndim=ndim, fastpath=fastpath,
-> 2082                                           placement=placement, **kwargs)
   2083
   2084     @property
```

```
KeyboardInterrupt:
```

In [ ]:

```
```