You are traveling to Dallas Texas for a business conference at the Hilton Anatole.   A rental car was not included in the company travel budget since the conference is in the hotel that you are staying at. You are unfamiliar with the area and like to explore some of the city without spending a lot of money on Uber fare. You want to find trending places in the area.

We will use Foursquare to see some of the top trending places that are in walking distance or a Uber ride away. The top places will be shown on a map using Folium to show the distance from the hotel.

In [201]:

```
import requests
import pandas as pd
import numpy as np
import random
import matplotlib.pylab as plt
from pandas.io.json import json_normalize
from urllib.request import urlopen
import ssl
import csv
from IPython.display import Image
from IPython.core.display import HTML
import matplotlib.cm as cm
import matplotlib.colors as colors

print('Done')
```

```
Done
```

In [208]:

```
!pip install geopy
from geopy.geocoders import Nominatim
print('Done')
```

```
Requirement already satisfied: geopy in /opt/conda/envs/Python36/lib/python3.6/site-packages (1.18.1)
Requirement already satisfied: geographiclib<2,>=1.49 in /opt/conda/envs/Python36/lib/python3.6/site-packages
(from geopy) (1.49)
Done
```

In [ ]:

Foursquare:

We start by locating the hotel's longitude and latitude

In [112]:
```
address = '2201 N Stemmons Fwy Dallas, TX 75207'
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
32.7997897 -96.82897066229901
```

Now we have our starting point. We explore the venues around our new found location.

In [297]:
```
url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, radius, LIMIT)
```

In [298]:
```
# send GET request and get trending venues
results = requests.get(url).json()
```

In [299]:
```
items = results['response']['groups'][0]['items']
```

Creating a dataframe

In [128]:

```
df = json_normalize(items)
df.head(2)
```

Out[128]:

	reasons.count	reasons.items	referralId	venue.beenHere.count	venue.beenHere.lastCheckinExpiredAt	venue.beenHe
0	0	[{'summary': 'This spot is popular', 'type': '...	e-0-503eae61e4b0052cdc5903dc-0	0	0	
1	0	[{'summary': 'This spot is popular', 'type': '...	e-0-4a6e9885f964a520edd41fe3-1	0	0	

2 rows × 39 columns

As you can see there is a lot of information in our current table that is not needed. We then clean the dataframe. We are trying to locate the venues around our hotel so we remove anything that will not help us.

In [167]:

```
df.rename(columns={'venue.name': 'venue', 'venue.location.lat':'lat', 'venue.location.lng':'lng', 'venue.loca
tion.postalCode':'postcode', 'venue.location.address':'address'}, inplace=True)
df.head(2)
```

Out[167]:

	reasons.count	reasons.items	referralId	venue.beenHere.count	venue.beenHere.lastCheckinExpiredAt	venue.beenHe
0	0	[{'summary': 'This spot is popular', 'type': '...	e-0-503eae61e4b0052cdc5903dc-0	0	0	
1	0	[{'summary': 'This spot is popular', 'type': '...	e-0-4a6e9885f964a520edd41fe3-1	0	0	

2 rows × 39 columns

Now the dataframe is cleaner and easier to read. We now can see a list of addresses, latitudes, longitudes and the name of the venues near our hotel

In [168]:

```
df2 = df.drop(columns=['reasons.count', 'venue.location.distance', 'venue.location.formattedAddress', 'reasons.items', 'referralId', 'venue.beenHere.count', 'venue.beenHere.marked', 'venue.beenHere.unconfirmedCount', 'venue.delivery.id', 'venue.delivery.provider.icon.name', 'venue.delivery.provider.icon.name', 'venue.location.state', 'venue.photos.count', 'venue.photos.groups', 'venue.stats.checkinsCount', 'venue.stats.tipCount', 'venue.location.city', 'venue.photos.groups', 'venue.venuePage.id', 'venue.location.state', 'reasons.items', 'venue.photos.count', 'reasons.count', 'venue.venuePage.id', 'venue.delivery.provider.icon.name', 'venue.delivery.provider.icon.prefix', 'venue.stats.tipCount', "venue.stats.visitsCount", 'venue.stats.usersCount', 'venue.location.labeledLatLngs', 'venue.verified', 'venue.location.crossStreet', 'venue.location.cc', 'venue.hereNow.summary', 'venue.hereNow.groups', 'venue.hereNow.count', 'venue.delivery.url', 'venue.delivery.provider.name', 'venue.delivery.provider.icon.sizes', 'venue.categories', 'venue.beenHere.lastCheckinExpiredAt', 'venue.location.country', 'venue.id'], axis=1)
df2.head(15)
```

Out[168]:

	address	lat	lng	postcode	venue
0	2201 N Stemmons Fwy	32.799945	-96.829562	75207	Sēr
1	2201 N Stemmons Fwy	32.799841	-96.829148	75207	Hilton
2	2026 Farrington St	32.796797	-96.828971	75207	Peticolas Brewing Company
3	NaN	32.797895	-96.828018	NaN	viva's
4	2201 N Stemmons Fwy	32.799401	-96.831787	75207	Verandah
5	NaN	32.799755	-96.831831	NaN	Sculpture Garden At Hilton Anatole
6	1950 Market Center Blvd	32.797356	-96.824487	75207	Ferris Wheeler's Backyard and BBQ
7	2201 N Stemmons Fwy	32.800250	-96.828774	75207	Counter Offer
8	2222 Vantage St	32.799170	-96.833974	75207	Pegasus City Brewery
9	Hilton Anatole	32.800518	-96.830171	75207	Executive Lounge
10	NaN	32.799754	-96.832647	75207	The Anatole Pool & Bar
11	2150 Market Center Blvd	32.799709	-96.824518	75207	Courtyard by Marriott - Dallas Market Center
12	2201 N Stemmons Fwy	32.799571	-96.829556	75207	FedEx Office Print & Ship Center
13	2201 N Stemmons Fwy	32.800143	-96.830280	75207	Gossip Bar
14	2015 Market Center Blvd	32.797454	-96.825984	75207	DoubleTree by Hilton

In [140]:

```
!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library
print("Done")
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    altair-4.1.0               |             py_1         614 KB  conda-forge
    branca-0.4.1               |             py_0          26 KB  conda-forge
    ca-certificates-2020.6.20  |       hecda079_0         145 KB  conda-forge
    certifi-2020.6.20          |   py36h9f0ad1d_0         151 KB  conda-forge
    python_abi-3.6             |          1_cp36m           4 KB  conda-forge
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    openssl-1.1.1g             |       h516909a_0         2.1 MB  conda-forge
    ------------------------------------------------------------
                                           Total:         3.1 MB

The following NEW packages will be INSTALLED:

    altair:          4.1.0-py_1        conda-forge
    branca:          0.4.1-py_0        conda-forge
    folium:          0.5.0-py_0        conda-forge
    python_abi:      3.6-1_cp36m       conda-forge
    vincent:         0.4.4-py_1        conda-forge

The following packages will be UPDATED:

    certifi:         2020.6.20-py36_0              --> 2020.6.20-py36h9f0ad1d_0 conda-forge
    openssl:         1.1.1g-h7b6447c_0             --> 1.1.1g-h516909a_0        conda-forge

The following packages will be DOWNGRADED:

    ca-certificates: 2020.6.24-0                   --> 2020.6.20-hecda079_0     conda-forge
```

```
Downloading and Extracting Packages
altair-4.1.0         | 614 KB    | ##################################### | 100% 
branca-0.4.1         | 26 KB     | ##################################### | 100% 
ca-certificates-2020 | 145 KB    | ##################################### | 100% 
certifi-2020.6.20    | 151 KB    | ##################################### | 100% 
python_abi-3.6       | 4 KB      | ##################################### | 100% 
vincent-0.4.4        | 28 KB     | ##################################### | 100% 
folium-0.5.0         | 45 KB     | ##################################### | 100% 
openssl-1.1.1g       | 2.1 MB    | ##################################### | 100% 
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Done
```

In [191]:

```
import matplotlib.cm as cm
import matplotlib.colors as colors

from sklearn.cluster import KMeans
```

In [224]:

```
len(df2['venue'].unique())
```

Out[224]:

```
24
```

In [231]:
```
df_onehot = pd.get_dummies(df2[['venue']], prefix="", prefix_sep="")


df_onehot['postalcode'] = df2['postcode']
df_onehot['venue'] = df2['venue']


fixed_columns = list(df_onehot.columns[-3:]) + list(df_onehot.columns[:-3])
df_onehot = df_onehot[fixed_columns]

print(df_onehot.shape)
df_onehot.head()
```

```
(24, 26)
```

Out[231]:

	viva's	postalcode	venue	Best Western Market Center	City View Terrace	Counter Offer	Courtyard by Marriott - Dallas Market Center	Days Inn	DoubleTree by Hilton	Executive Lounge	...	Media Grill + Bar	Pegasus City Brewery	Peticolas Brewing Company
0	0	75207	Sēr	0	0	0	0	0	0	0	...	0	0	0
1	0	75207	Hilton	0	0	0	0	0	0	0	...	0	0	0
2	0	75207	Peticolas Brewing Company	0	0	0	0	0	0	0	...	0	0	1
3	1	NaN	viva's	0	0	0	0	0	0	0	...	0	0	0
4	0	75207	Verandah	0	0	0	0	0	0	0	...	0	0	0

5 rows × 26 columns

In [232]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

Finding the top 10 venues in the area

In [274]:

```
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['venue']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
venues_sorted = pd.DataFrame(columns=columns)
venues_sorted['venue'] = df2['venue']

for ind in np.arange(df2.shape[0]):
    row_categories = df2.iloc[ind, :].iloc[3:]
    row_categories_sorted = row_categories.sort_values(ascending=False)
    venues_sorted.iloc[ind, 9:] = row_categories_sorted.index.values[0:num_top_venues]

venues_sorted.head()
```

Out[274]:

	venue	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Sēr	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	venue	postcode
1	Hilton	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	venue	postcode
2	Peticolas Brewing Company	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	venue	postcode
3	viva's	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	venue	postcode
4	Verandah	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	venue	postcode

Refine and cluster the venues using the k-means algorithm

In [245]:

```
kclusters = 2

df_clustering = df2.drop(['venue'], 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit

clustered_df = df2

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit

clustered_df
```

Out[245]:

	address	lat	lng	postcode	venue
0	2201 N Stemmons Fwy	32.799945	-96.829562	75207	Sēr
1	2201 N Stemmons Fwy	32.799841	-96.829148	75207	Hilton
2	2026 Farrington St	32.796797	-96.828971	75207	Peticolas Brewing Company
3	NaN	32.797895	-96.828018	NaN	viva's
4	2201 N Stemmons Fwy	32.799401	-96.831787	75207	Verandah
5	NaN	32.799755	-96.831831	NaN	Sculpture Garden At Hilton Anatole
6	1950 Market Center Blvd	32.797356	-96.824487	75207	Ferris Wheeler's Backyard and BBQ
7	2201 N Stemmons Fwy	32.800250	-96.828774	75207	Counter Offer
8	2222 Vantage St	32.799170	-96.833974	75207	Pegasus City Brewery
9	Hilton Anatole	32.800518	-96.830171	75207	Executive Lounge
10	NaN	32.799754	-96.832647	75207	The Anatole Pool & Bar
11	2150 Market Center Blvd	32.799709	-96.824518	75207	Courtyard by Marriott - Dallas Market Center
12	2201 N Stemmons Fwy	32.799571	-96.829556	75207	FedEx Office Print & Ship Center
13	2201 N Stemmons Fwy	32.800143	-96.830280	75207	Gossip Bar
14	2015 Market Center Blvd	32.797454	-96.825984	75207	DoubleTree by Hilton
15	NaN	32.803405	-96.831912	75207	The Renaissance Club Lounge
16	2101 N Stemmons Fwy	32.800493	-96.825241	75207	Sheraton Suites Market Center Dallas
17	2325 N Stemmons Fwy	32.801958	-96.833085	75207	Hilton Garden Inn
18	NaN	32.799920	-96.828155	75202	Terrace Bar & Grill
19	2050 N Stemmons Fwy, Suite 111	32.802523	-96.824949	75207	FedEx Office Ship Center
20	2201 N Stemmons Fwy	32.799874	-96.828973	75207	Media Grill + Bar
21	2023 Market Center Blvd	32.798250	-96.825683	75207	Best Western Market Center
22	2026 Market Center Blvd	32.798352	-96.824813	75207	Days Inn
23	2222 N Stemmons Fwy	32.803090	-96.832370	75207	City View Terrace

To better vizualize our list we can now put it all on a map

In [271]:

```
venues_map = folium.Map(location=[latitude, longitude], zoom_start=15)

folium.CircleMarker(
    [latitude, longitude],
    radius=11,
    color='red',
    popup='Hilton Anatole',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)

markers_colors = []
for lat, lng, post, vne, in zip(clustered_df['lat'], clustered_df['lng'], clustered_df['postcode'], clustered_df['venue']):
    label = folium.Popup('{} ({}): {} - Cluster {}'.format(lng, lat, vne, post), parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='purple',
        fill=True,
        fill_color='pink',
        fill_opacity=0.7).add_to(venues_map)

venues_map
```

```
Out[271]: Make this Notebook Trusted to load map: File -> Trust Notebook
```

Our final result shows the top 10 venues close to the Hilton Anatole hotel. This answers the question on what venues are in walking distance or a short Uber ride away.

The Hilton Anatole is a fairly large hotel. A few of those venues are in the hotel itself as shown on the map. Famished and don't feel like leaving the hotel? The Media Grill and bar has a varity of entries.The hotel is identified by the red circle marker. The popular venues are identified by the pink and purple circle markers. To see what each venue is you are able to to click on any marker to reveal the venue name and coordinates. This answers the question on what venues are in walking distance or a short Uber ride away.