

# Perfect Sports Arena: San Antonio, TX

## OVERVIEW/BACKGROUND

In the Nation Basketball Association (NBA), the arenas that each team plays in typically are located in the downtown of the city. As a fan of the NBA and more specifically the San Antonio Spurs, I enjoy going to games in person since I live in the same city. The AT&T Center is the name of the arena that the San Antonio Spurs play in. However, unlike other arenas for other teams in the league, the AT&T center is not even located in downtown San Antonio. It is located approximately 5 miles from downtown. The question then becomes: Would it make more sense to have a sports arena that is located downtown as opposed to its current location.

## HOW DATA CAN BE USED

For this report I will use San Antonio location data to visualize the neighborhood and venue densities of both the area surrounding the AT&T Center and downtown San Antonio. If downtown is more densely populated with neighborhoods and venues, then that could lead to an economic incentive to have an arena in that area.

## Methodologies

First, all of the data I will obtain to perform analysis on, will be received from the Foursquare API. With the folium library, I will visualize the venues around both the AT&T Center and the Riverwalk which are both located in San Antonio, TX. To cluster venues that are close together, I will use the K-Means clustering algorithm. Ultimately, I want to determine which location (the AT&T Center or the area around the Riverwalk) would be best start a new business or build a new basketball arena. The thought is that an area with many venues in it would be a great location for a business to start because people would naturally be in those areas already.

## Analysis

### Part 1: Find Venues within one mile of the AT&T Center and cluster them

First, lets import all the libraries we will need for this analysis

```
In [1]: # Import necessary libraries
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
import folium
from geopy.geocoders import Nominatim
from sklearn.cluster import KMeans
import matplotlib.cm as cm
```

```
import matplotlib.colors as colors
from pandas.io.json import json_normalize
```

Second, let's get location data for the AT&T Center so that we can create a map from that with venues within one mile of it.

```
In [2]: att_address = '1 AT&T Center Parkway, San Antonio, TX 78219'

att_geolocator = Nominatim(user_agent="att_explorer")
att_location = att_geolocator.geocode(att_address)
att_latitude = att_location.latitude
att_longitude = att_location.longitude

print('The geographical coordinates of the AT&T Center are {}, {}'.format(att_la
```

The geographical coordinates of the AT&T Center are 29.4270504, -98.43750706398404.

Account credentials for Fourquare API

```
In [3]: CLIENT_ID = 'W5JGSVZWGBU5NHJZ2PJEBZP5DEWSQQ0Y0MAH12S35WJLJ2C1' # your Foursquare
CLIENT_SECRET = 'HRWZ0NPUJ4WA0D00JJ2NRSGUX0QFVO4N4WO3MRQNSJ3L52K4' # your Foursq
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

print('Your credentials:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: ' + CLIENT_SECRET)
```

Your credentials:

CLIENT\_ID: W5JGSVZWGBU5NHJZ2PJEBZP5DEWSQQ0Y0MAH12S35WJLJ2C1

CLIENT\_SECRET: HRWZ0NPUJ4WA0D00JJ2NRSGUX0QFVO4N4WO3MRQNSJ3L52K4

Retrieve the url containing location data for the AT&T Center based on its latitude, longitude and distance from other venues

```
In [4]: LIMIT = 100

radius = 1609.34 # 1609.34 meters is approximately one mile

att_url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_sec
CLIENT_ID,
CLIENT_SECRET,
VERSION,
att_latitude,
att_longitude,
radius,
LIMIT)
att_url # display URL
```

```
Out[4]: 'https://api.foursquare.com/v2/venues/explore?&client_id=W5JGSVZWGBU5NHJZ2PJEBZP
5DEWSQQ0Y0MAH12S35WJLJ2C1&client_secret=HRWZ0NPUJ4WA0D00JJ2NRSGUX0QFVO4N4WO3MRQN
SJ3L52K4&v=20180605&ll=29.4270504,-98.43750706398404&radius=1609.34&limit=100'
```

Convert the url to json

```
In [5]: att_results = requests.get(att_url).json()
```

Create function to receive category type

```
In [6]: def get_category_type(row):
```

```

try:
    categories_list = row['categories']
except:
    categories_list = row['venue.categories']

if len(categories_list) == 0:
    return None
else:
    return categories_list[0]['name']

```

Create a dataframe based on venue name, category, latitude, and longitude that was recieved from the json-converted url

```

In [7]: att_venues = att_results['response']['groups'][0]['items']

att_nearby_venues = json_normalize(att_venues) # flatten JSON

# filter columns
att_filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',
att_nearby_venues = att_nearby_venues.loc[:, att_filtered_columns]

# filter the category for each row
att_nearby_venues['venue.categories'] = att_nearby_venues.apply(get_category_type, axis=1)

# clean columns
att_nearby_venues.columns = [col.split(".")[-1] for col in att_nearby_venues.columns]

att_nearby_venues

```

<ipython-input-7-6balccc0f865>:3: FutureWarning: pandas.io.json.json\_normalize is deprecated, use pandas.json\_normalize instead

```
att_nearby_venues = json_normalize(att_venues) # flatten JSON
```

Out[7]:

	name	categories	lat	lng
0	AT&T Center	Basketball Stadium	29.426889	-98.437409
1	Fan Shop	Souvenir Shop	29.426422	-98.437564
2	San Antonio Stock Show & Rodeo Hall Of Fame	Art Gallery	29.426773	-98.440274
3	AT&T Center - Plaza Level	Stadium	29.426890	-98.436948
4	Whataburger	Burger Joint	29.426497	-98.437688
5	Penske Truck Rental	Rental Service	29.425359	-98.425840
6	Freeman Coliseum	General Entertainment	29.426772	-98.439674
7	Starbucks	Coffee Shop	29.425538	-98.421664
8	SUBWAY	Sandwich Place	29.440743	-98.438017
9	Corner Store	Convenience Store	29.424536	-98.426425
10	Circle K	Convenience Store	29.440832	-98.437981
11	Spurs Larry O'Brien Trophies Display	Basketball Court	29.426806	-98.436888
12	Rampage Locker Room	Skating Rink	29.426912	-98.436579
13	Club Level Bar	Sports Bar	29.426931	-98.437125
14	Bud Light Lime Lounge	Bar	29.426911	-98.437061
15	Fan Favorites	Food Service	29.427013	-98.437400

	name	categories	lat	lng
16	At&t Center: Aramark Offices	Stadium	29.427704	-98.438266
17	Rocks & Brews (Rock & Brews)	Bar	29.427072	-98.437444
18	Rampage Hockey Section 104	Hockey Rink	29.426601	-98.437318
19	Pizza Hut Stand 110	Pizza Place	29.429638	-98.437505
20	Crossfit South Central Regionals	Track	29.426271	-98.440381
21	FedEx Ship Center	Shipping Store	29.437780	-98.427735
22	Crazy J's Sports Bar	Sports Bar	29.435315	-98.433153
23	South Salado Creek Greenway	Trail	29.424802	-98.428851
24	rio grande cafe	Café	29.424884	-98.446971
25	Ryder Truck Rental	Rental Car Location	29.436215	-98.434558
26	Travelodge by Wyndham	Hotel	29.425188	-98.422798
27	Illusions Rentals & Designs	Rental Service	29.438942	-98.436965
28	Quality Inn I-10 East Near At&T Center	Hotel	29.425182	-98.422788
29	Antioch Sports Complex	Gym / Fitness Center	29.426759	-98.452404
30	First Stop	Convenience Store	29.425099	-98.452433

Rename columns for better description

```
In [8]: att_nearby_venues.rename(columns={'name': 'Venue', 'categories': 'Category', 'lat': 'lat', 'lng': 'lng'})
```

Out[8]:

	Venue	Category	Latitude	Longitude
0	AT&T Center	Basketball Stadium	29.426889	-98.437409
1	Fan Shop	Souvenir Shop	29.426422	-98.437564
2	San Antonio Stock Show & Rodeo Hall Of Fame	Art Gallery	29.426773	-98.440274
3	AT&T Center - Plaza Level	Stadium	29.426890	-98.436948
4	Whataburger	Burger Joint	29.426497	-98.437688
5	Penske Truck Rental	Rental Service	29.425359	-98.425840
6	Freeman Coliseum	General Entertainment	29.426772	-98.439674
7	Starbucks	Coffee Shop	29.425538	-98.421664
8	SUBWAY	Sandwich Place	29.440743	-98.438017
9	Corner Store	Convenience Store	29.424536	-98.426425
10	Circle K	Convenience Store	29.440832	-98.437981
11	Spurs Larry O'Brien Trophies Display	Basketball Court	29.426806	-98.436888
12	Rampage Locker Room	Skating Rink	29.426912	-98.436579
13	Club Level Bar	Sports Bar	29.426931	-98.437125
14	Bud Light Lime Lounge	Bar	29.426911	-98.437061

	Venue	Category	Latitude	Longitude
15	Fan Favorites	Food Service	29.427013	-98.437400
16	At&t Center: Aramark Offices	Stadium	29.427704	-98.438266
17	Rocks & Brews (Rock & Brews)	Bar	29.427072	-98.437444
18	Rampage Hockey Section 104	Hockey Rink	29.426601	-98.437318
19	Pizza Hut Stand 110	Pizza Place	29.429638	-98.437505
20	Crossfit South Central Regionals	Track	29.426271	-98.440381
21	FedEx Ship Center	Shipping Store	29.437780	-98.427735
22	Crazy J's Sports Bar	Sports Bar	29.435315	-98.433153
23	South Salado Creek Greenway	Trail	29.424802	-98.428851
24	rio grande cafe	Café	29.424884	-98.446971
25	Ryder Truck Rental	Rental Car Location	29.436215	-98.434558
26	Travelodge by Wyndham	Hotel	29.425188	-98.422798
27	Illusions Rentals & Designs	Rental Service	29.438942	-98.436965
28	Quality Inn I-10 East Near At&T Center	Hotel	29.425182	-98.422788
29	Antioch Sports Complex	Gym / Fitness Center	29.426759	-98.452404
30	First Stop	Convenience Store	29.425099	-98.452433

Display the number of venues in or around the AT&T Center that are within one mile

```
In [9]: print('There are {} venues in or around the AT&T that are within one mile. This
```

There are 31 venues in or around the AT&T that are within one mile. This includes the AT&T center itself.

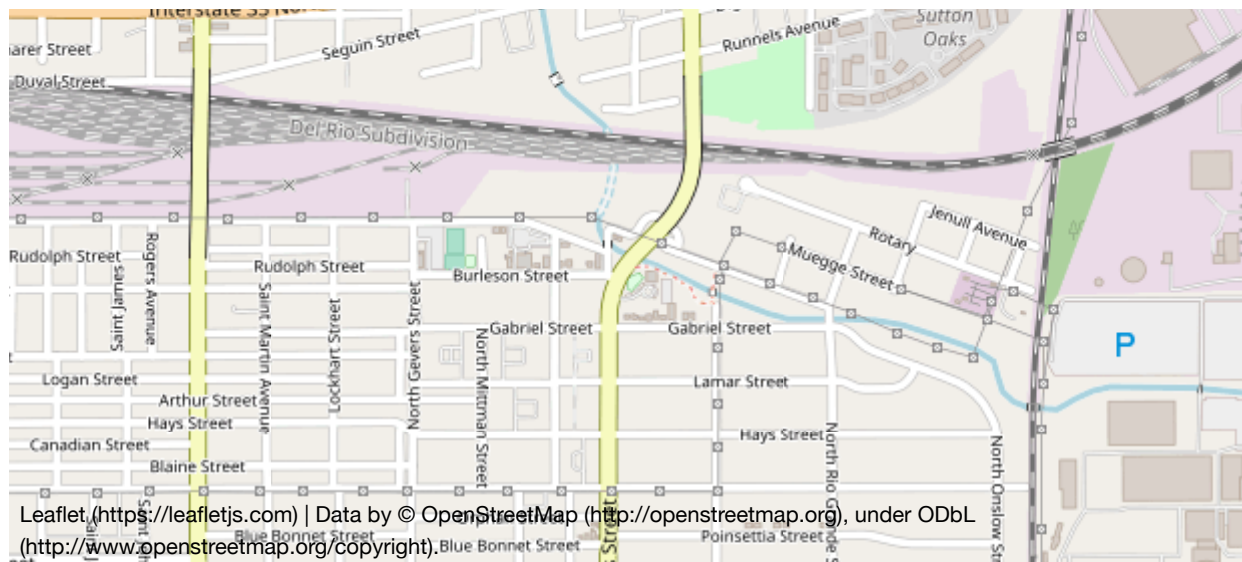
Create a map showing where the venues are located in respect to each other

```
In [10]: map_att = folium.Map(location=[att_latitude, att_longitude], zoom_start=15)

# add markers to map
for lat, lng, label in zip(att_nearby_venues['Latitude'], att_nearby_venues['Longitude']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_att)

map_att
```





Use K-Means clustering to group venues based on their distance from each other and create labels for each group

```
In [11]: att_kclusters = 4

att_cluster = att_nearby_venues.drop(['Venue', 'Category'], 1)
# run k-means clustering
att_kmeans = KMeans(n_clusters=att_kclusters, random_state=0).fit(att_cluster)

# check cluster labels generated for each row in the dataframe
att_kmeans.labels_[0:10]
```

```
Out[11]: array([0, 0, 0, 0, 0, 3, 0, 3, 1, 3], dtype=int32)
```

Add the cluster labels to the original dataframe

```
In [12]: att_nearby_venues['Cluster Label'] = att_kmeans.labels_
att_nearby_venues.head()
```

```
Out[12]:
```

	Venue	Category	Latitude	Longitude	Cluster Label
0	AT&T Center	Basketball Stadium	29.426889	-98.437409	0
1	Fan Shop	Souvenir Shop	29.426422	-98.437564	0
2	San Antonio Stock Show & Rodeo Hall Of Fame	Art Gallery	29.426773	-98.440274	0
3	AT&T Center - Plaza Level	Stadium	29.426890	-98.436948	0
4	Whataburger	Burger Joint	29.426497	-98.437688	0

Display the clusters on the map for a visualization of how the venues were distributed

```
In [13]: att_map_clusters = folium.Map(location=[att_latitude, att_longitude], zoom_start=13)

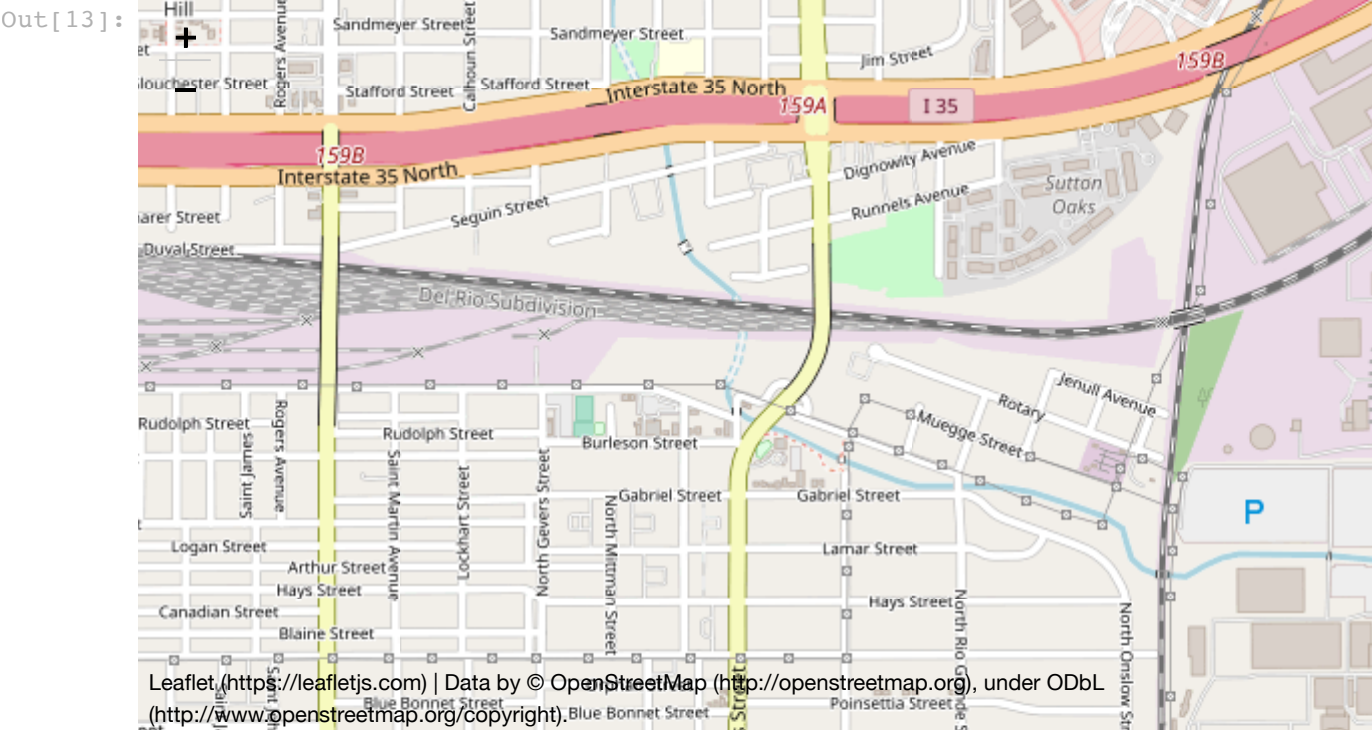
# set color scheme for the clusters
x = np.arange(att_kclusters)
ys = [i + x + (i*x)**2 for i in range(att_kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
```



```
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(att_nearby_venues['Latitude'], att_nearby_venue
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster],
        fill_opacity=0.7).add_to(att_map_clusters)

att_map_clusters
```



Display the venues that are associated with a bar

```
In [14]: att_nearby_venues.loc[att_nearby_venues['Category'].str.contains('Bar')]
```

Out[14]:

	Venue	Category	Latitude	Longitude	Cluster	Label
13	Club Level Bar	Sports Bar	29.426931	-98.437125	0	
14	Bud Light Lime Lounge	Bar	29.426911	-98.437061	0	
17	Rocks & Brews (Rock & Brews)	Bar	29.427072	-98.437444	0	
22	Crazy J's Sports Bar	Sports Bar	29.435315	-98.433153	1	

Find how many hotels are within one mile of the AT&T Center and how many of them have bars

```
In [15]: if len(att_nearby_venues.loc[att_nearby_venues['Category'] == 'Hotel']) == 1:
    print('There is {} hotel within one mile of the AT&T Center and it has 1 bar
    len(at
else:
```

```
print('There are {} hotels within one mile of the AT&T Center and {} of them
```

There are 2 hotels within one mile of the AT&T Center and 0 of them have bars.

## Part 2: Find Venues within one mile of the River Walk and cluster them.

The following steps will be similar to the steps taken for the previous section

Let's get location data for the River Walk so that we can create a map from that with venues within one mile of it.

```
In [16]: river_address = '849 E Commerce St, San Antonio, TX 78205'

river_geolocator = Nominatim(user_agent="river_explorer")
river_location = river_geolocator.geocode(river_address)
river_latitude = river_location.latitude
river_longitude = river_location.longitude

print('The geograpical coordinates of the Riverwalk are {}, {}'.format(river_la
```

The geograpical coordinates of the Riverwalk are 29.42463845, -98.48495091235024.

Retrieve the url containing location data for the Riverwalk based on its latitude, longitude and distance from other venues

```
In [17]: river_url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_s
CLIENT_ID,
CLIENT_SECRET,
VERSION,
river_latitude,
river_longitude,
radius,
LIMIT)
river_url # display URL
```

```
Out[17]: 'https://api.foursquare.com/v2/venues/explore?&client_id=W5JGSVZWGBU5NHJZ2PJEBZP
5DEWSQQ0Y0MAH12S35WJLJ2C1&client_secret=HRWZ0NPUJ4WA0D00JJ2NRSGUX0QFVO4N4WO3MRQN
SJ3L52K4&v=20180605&ll=29.42463845,-98.48495091235024&radius=1609.34&limit=100'
```

Convert the url to json

```
In [18]: river_results = requests.get(river_url).json()
```

Create a dataframe based on venue name, category, latitude, and longitude that was recieved from the json-converted url

```
In [19]: river_venues = river_results['response']['groups'][0]['items']

river_nearby_venues = json_normalize(river_venues) # flatten JSON

# filter columns
river_filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat']
river_nearby_venues = river_nearby_venues.loc[:, river_filtered_columns]

# filter the category for each row
river_nearby_venues['venue.categories'] = river_nearby_venues.apply(get_category
```



```
# clean columns
river_nearby_venues.columns = [col.split(".")[0] for col in river_nearby_venues
                                columns]
river_nearby_venues
```

<ipython-input-19-6e6d655eb3ca>:3: FutureWarning: pandas.io.json.json\_normalize is deprecated, use pandas.json\_normalize instead

```
river_nearby_venues = json_normalize(river_venues) # flatten JSON
```

Out[19]:

	name	categories	lat	lng
0	Fortress Alamo: The Key To Texas	History Museum	29.425306	-98.486392
1	Alamo Plaza	Plaza	29.425484	-98.486613
2	Fogo de Chao Brazilian Steakhouse	Brazilian Restaurant	29.423994	-98.484619
3	The Alamo	Historic Site	29.425779	-98.486113
4	San Antonio Marriott Riverwalk	Hotel	29.422346	-98.484504
...	...	...	...	...
95	Big Hops Growler Station "The Bridge"	Beer Garden	29.430622	-98.479910
96	El Colegio	Bar	29.425862	-98.490725
97	IHOP	Breakfast Spot	29.423614	-98.484687
98	Embassy Suites by Hilton	Hotel	29.426571	-98.492915
99	Pharm Table	Restaurant	29.429735	-98.488118

100 rows × 4 columns

Rename columns for better description

In [20]:

```
river_nearby_venues.rename(columns={'name': 'Venue', 'categories': 'Category', 'lat': 'Latitude', 'lng': 'Longitude'})
```

Out[20]:

	Venue	Category	Latitude	Longitude
0	Fortress Alamo: The Key To Texas	History Museum	29.425306	-98.486392
1	Alamo Plaza	Plaza	29.425484	-98.486613
2	Fogo de Chao Brazilian Steakhouse	Brazilian Restaurant	29.423994	-98.484619
3	The Alamo	Historic Site	29.425779	-98.486113
4	San Antonio Marriott Riverwalk	Hotel	29.422346	-98.484504
...	...	...	...	...
95	Big Hops Growler Station "The Bridge"	Beer Garden	29.430622	-98.479910
96	El Colegio	Bar	29.425862	-98.490725
97	IHOP	Breakfast Spot	29.423614	-98.484687
98	Embassy Suites by Hilton	Hotel	29.426571	-98.492915
99	Pharm Table	Restaurant	29.429735	-98.488118

100 rows × 4 columns

Display the number of venues on or around the Riverwalk that are within one mile

```
In [21]: print('There are {} venues on or around the Riverwalk that are within one mile.'
```

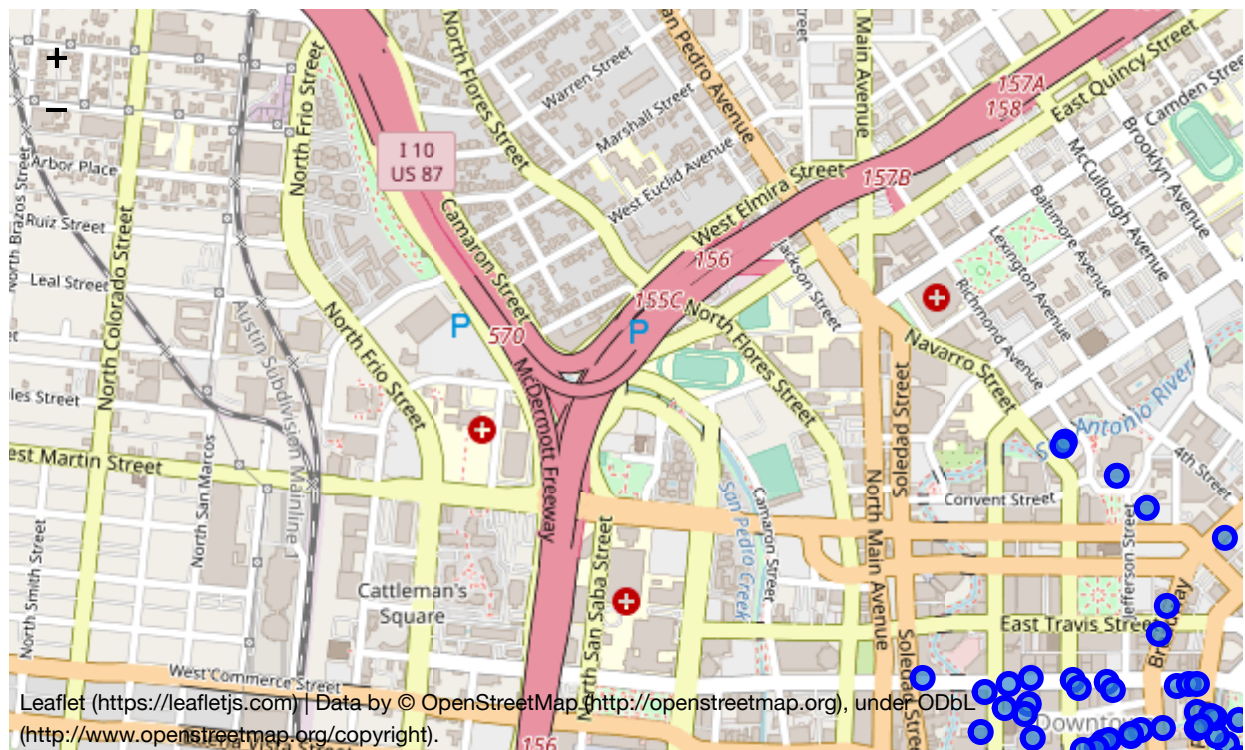
There are 100 venues on or around the Riverwalk that are within one mile.  
Create a map showing where the venues are located in respect to each other

```
In [22]: map_river = folium.Map(location=[river_latitude, river_longitude], zoom_start=15

# add markers to map
for lat, lng, label in zip(river_nearby_venues['Latitude'], river_nearby_venues[
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_river)

map_river
```

Out[22]:



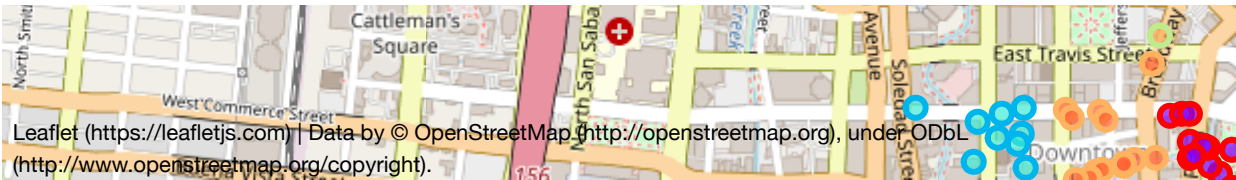
Use K-Means clustering to group venues based on their distance from each other and create labels for each group

```
In [23]: river_kclusters = 9

river_cluster = river_nearby_venues.drop(['Venue', 'Category'], 1)
# run k-means clustering
river_kmeans = KMeans(n_clusters=river_kclusters, random_state=0).fit(river_clus

# check cluster labels generated for each row in the dataframe
river_kmeans.labels_[0:10]
```





Display the venues that are associated with a bar

```
In [26]: river_nearby_venues.loc[river_nearby_venues['Category'].str.contains('Bar')]
```

Out[26]:

	Venue	Category	Latitude	Longitude	Cluster Label
8	Dave & Buster's	Sports Bar	29.424145	-98.485056	8
42	Howl at the Moon	Piano Bar	29.425366	-98.489024	7
50	Pat O'Brien's	Bar	29.424339	-98.487694	0
55	The Esquire Tavern	Cocktail Bar	29.424834	-98.491761	3
63	Menger Bar	Hotel Bar	29.425051	-98.486112	0
64	Moses Rose's Hideout	Bar	29.426414	-98.487432	0
67	Club Sirius	Bar	29.424563	-98.487807	0
70	The Bar At Marriott Rivercenter	Hotel Bar	29.423149	-98.483928	8
76	Revolucion Coffee + Juice	Juice Bar	29.426362	-98.489576	7
78	Texas T Pub	Dive Bar	29.427361	-98.487860	7
87	Bier Garten Riverwalk	Bar	29.425609	-98.487766	7
90	Soho Wine & Martini Bar	Wine Bar	29.424964	-98.490272	3
96	El Colegio	Bar	29.425862	-98.490725	3

Find how many hotels are within one mile of the Riverwalk and how many of them have bars

```
In [27]: if len(river_nearby_venues.loc[river_nearby_venues['Category'] == 'Hotel']) == 1:
          print('There is {} hotel within one mile of the Riverwalk and it has 1 bar.'
                len(river_nearby_venues.loc[river_nearby_venues['Category'] == 'Hotel'])
          else:
          print('There are {} hotels within one mile of the Riverwalk and {} of them have bars.'
```

There are 19 hotels within one mile of the Riverwalk and 2 of them have bars.

## Results

By counting the number of venues within one mile of both the AT&T Center and the Riverwalk, it is determined that the Riverwalk has 100 venues and the AT&T Center has 31 venues that fit that criteria. This means, the Riverwalk is within one mile of 69 more venues than the AT&T Center within the same distance. I also found that the Riverwalk has 17 more hotels within one mile of it then the AT&T Center does (19 to 2). For the AT&T Center, I determined that 4 venue clusters was optimal and for the Riverwalk, I determined that 9 venue clusters was optimal based on the visualization that the map provided.

## Conclusion

In conclusion, given the significant difference between the number of venues around the AT&T Center and the Riverwalk, if a new business is looking to come to San Antonio they should look downtown close to where the Riverwalk is. Many venues are in the area that would attract people to the location to begin with and they wouldn't have to go out of their way to visit there. A new arena in the downtown would also make sense given the many hotels and bars in the downtown area. People coming from out of town could see a basketball game with more ease, while also enjoying the many amenities that downtown brings.