University California in analysis

The Battle of Neighborhoods

Author: Hua Zhu

Apr, 2019

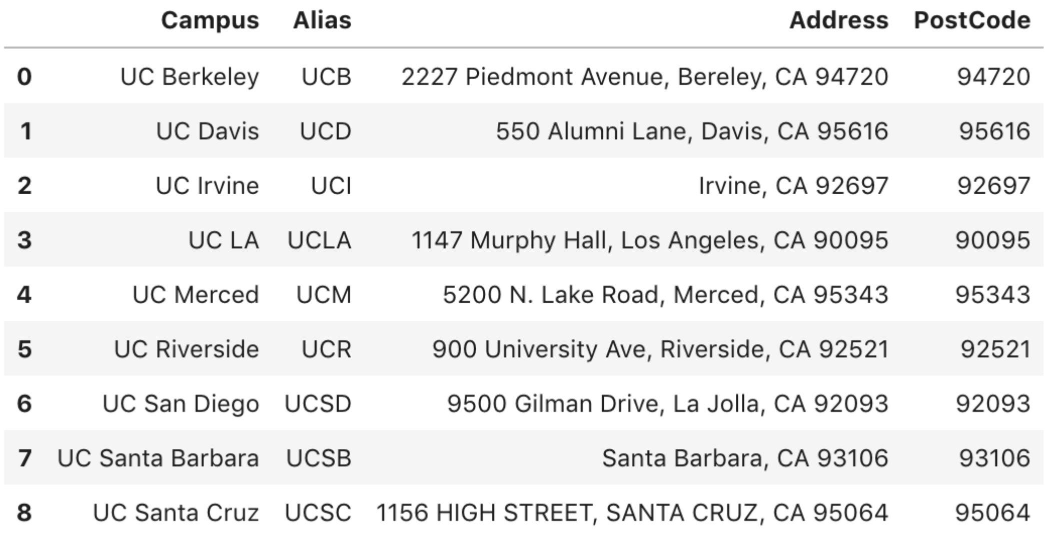
1. Business Problem

1.1 Background

Each year, many international students [in 2018, it is 28,556] apply their application into University California. There are 9 undergraduate campuses throughout California with 150 disciplines. More academic departments ranked in the top 10 nationally than any other public or private university. Other than regular information which can easier access from university website, for better decide which campus is most safe and convenient for their future study. Each student and their parents, they need to know:

- Activities around the campus [Venues of choice]
- Neighborhood crime-incident data

The campus who providing undergraduate education is listed in below table.



	Campus	Alias	Address	PostCode
0	UC Berkeley	UCB	2227 Piedmont Avenue, Bereley, CA 94720	94720
1	UC Davis	UCD	550 Alumni Lane, Davis, CA 95616	95616
2	UC Irvine	UCI	Irvine, CA 92697	92697
3	UC LA	UCLA	1147 Murphy Hall, Los Angeles, CA 90095	90095
4	UC Merced	UCM	5200 N. Lake Road, Merced, CA 95343	95343
5	UC Riverside	UCR	900 University Ave, Riverside, CA 92521	92521
6	UC San Diego	UCSD	9500 Gilman Drive, La Jolla, CA 92093	92093
7	UC Santa Barbara	UCSB	Santa Barbara, CA 93106	93106
8	UC Santa Cruz	UCSC	1156 HIGH STREET, SANTA CRUZ, CA 95064	95064

1.2 Target audience

This report try to analyze the 9 campus in University of California system for undergraduate education, provide intended audience for above information to

- Applicants and their parents/Relatives
- Every one who Interesting in UC system neighborhood

2. Data source and usage

2.1 Data source

- https://www.universityofcalifornia.edu/infocenter
- http://www.city-data.com/
- https://data.chhs.ca.gov/dataset/

2.2 Data usage

- https://www.universityofcalifornia.edu/infocenter and related campus website for basic information, gather and clean a csv file as start point
- Python reverse address to geolocation packages for, geolocation determination based on address. http://www.city-data.com/ , add latitude and longitude information into the original data
- Folium map rendering for OpenStreetMap data
- FourSquare API for venue related data
- https://data.chhs.ca.gov/dataset/ for California crime-incident related data

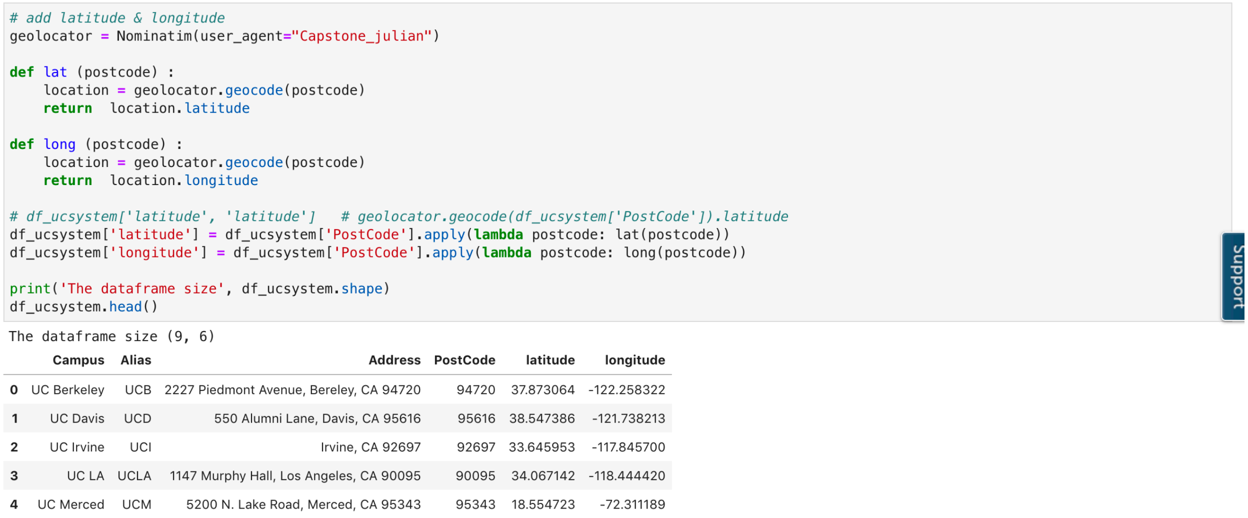
3. Methodology

3.1 Methodology in high level steps

- Environment setup aka import necessary libraries.
- Read previous csv file into data frame
- Leverage geocoder API to add latitude/longitude features
- Visualization via use Folium map rendering for OpenStreetMap data
- Use FourSquare API to gather venues data nearby
- Analyze top 10/top 5 venue category nearby and visualize it
- Get California state crime data from 2000~2013, get statistics mean to get safety index
- Visualize the safety index

3.2 Environment setup and map the 9 campus

Environment setup and csv code are skipped, you can reference the github notebook if necessary. One point here is use geocoder API to add latitude and longitude features into data frame. The code is listed below:

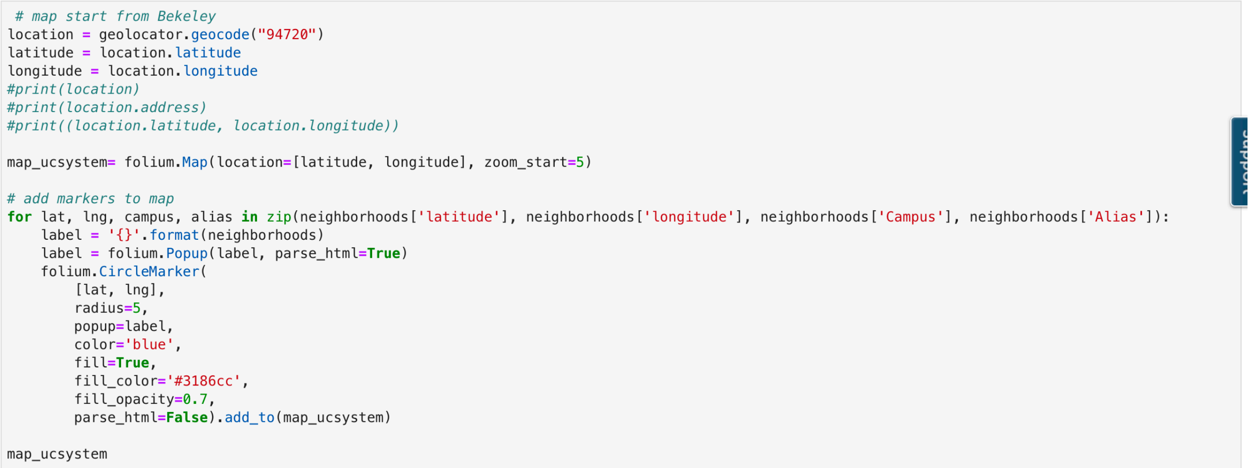


```
# add latitude & longitude
geolocator = Nominatim(user_agent="Capstone_julian")

def lat (postcode) :
    location = geolocator.geocode(postcode)
    return  location.latitude

def long (postcode) :
    location = geolocator.geocode(postcode)
    return  location.longitude

# df_ucsystem['latitude', 'latitude']   # geolocator.geocode(df_ucsystem['PostCode']).latitude
df_ucsystem['latitude'] = df_ucsystem['PostCode'].apply(lambda postcode: lat(postcode))
df_ucsystem['longitude'] = df_ucsystem['PostCode'].apply(lambda postcode: long(postcode))

print('The dataframe size', df_ucsystem.shape)
df_ucsystem.head()
```

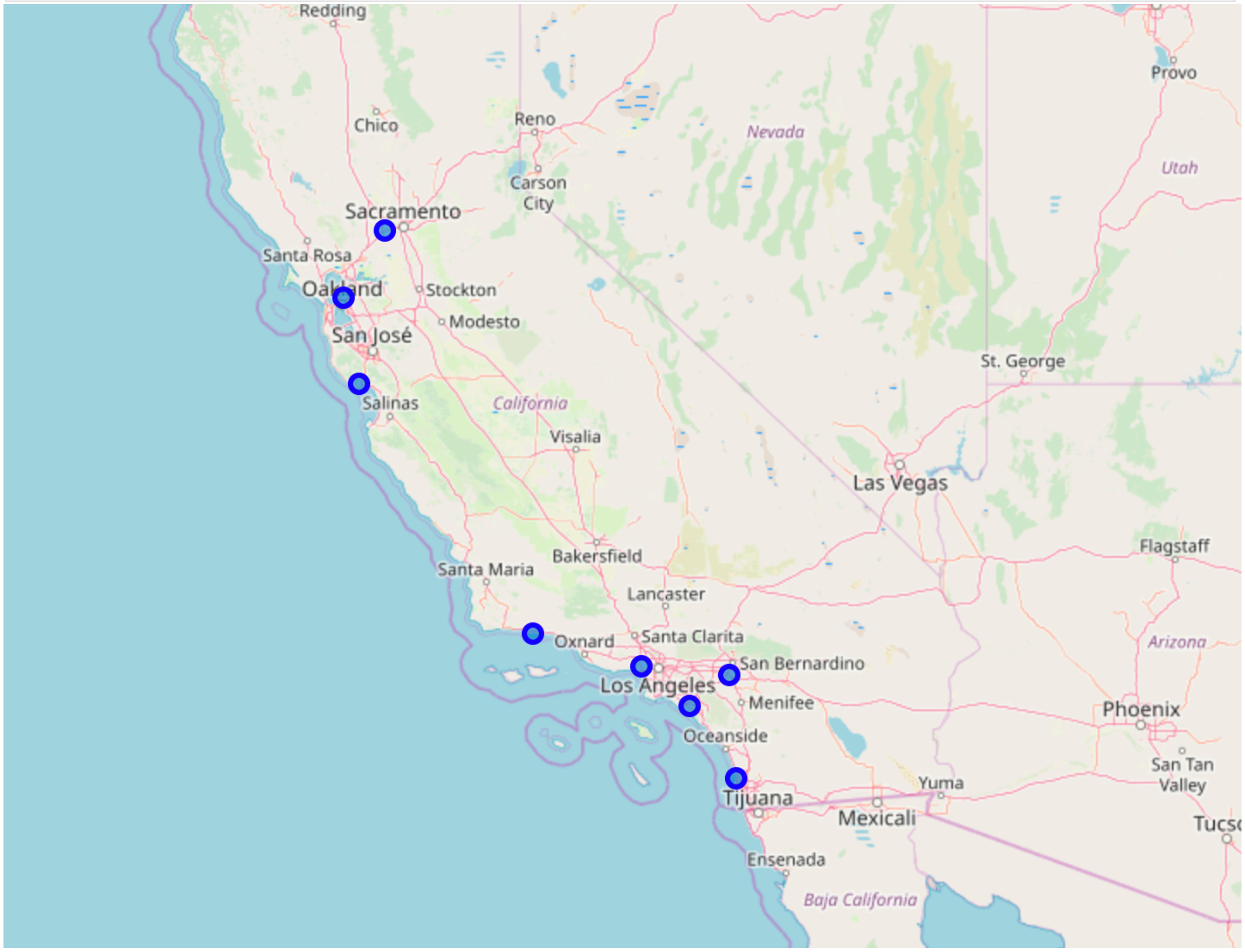
```
The dataframe size (9, 6)
```

	Campus	Alias	Address	PostCode	latitude	longitude
0	UC Berkeley	UCB	2227 Piedmont Avenue, Bereley, CA 94720	94720	37.873064	-122.258322
1	UC Davis	UCD	550 Alumni Lane, Davis, CA 95616	95616	38.547386	-121.738213
2	UC Irvine	UCI	Irvine, CA 92697	92697	33.645953	-117.845700
3	UC LA	UCLA	1147 Murphy Hall, Los Angeles, CA 90095	90095	34.067142	-118.444420
4	UC Merced	UCM	5200 N. Lake Road, Merced, CA 95343	95343	18.554723	-72.311189

After the data frame is prepared. Use Folium map rendering API to visualization. The code and map output are listed below.

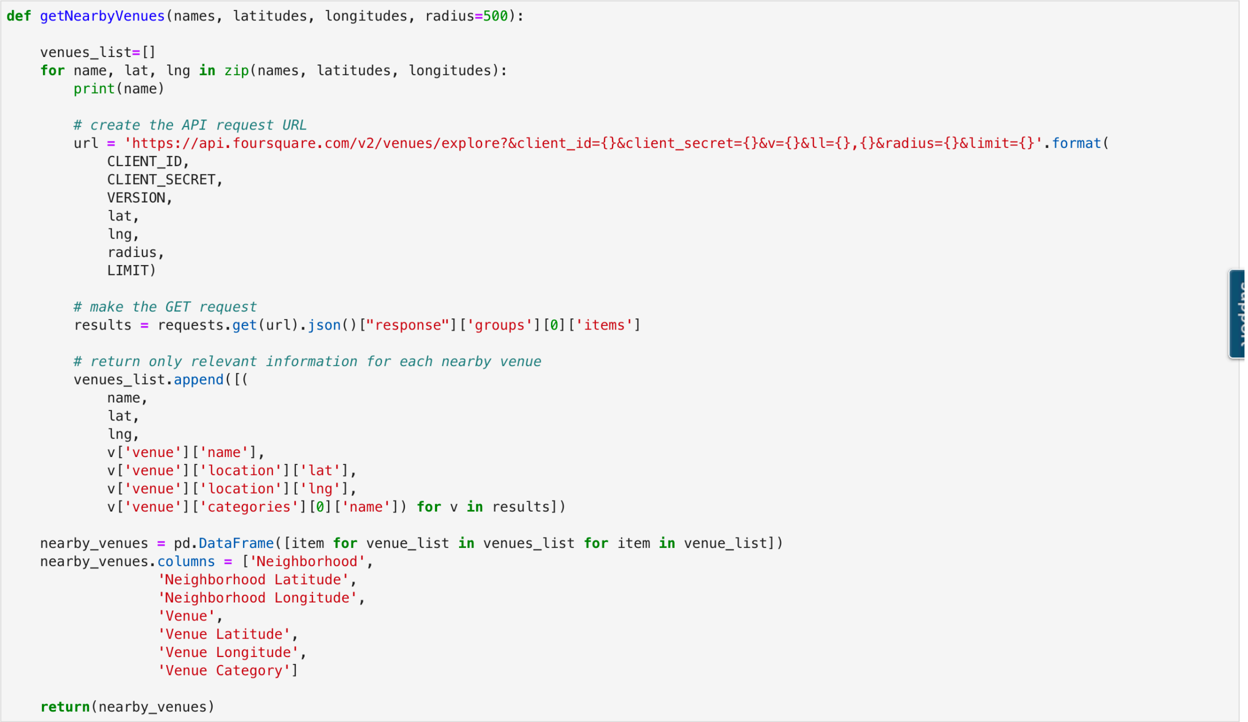


```
 # map start from Bekeley
location = geolocator.geocode("94720")
latitude = location.latitude
longitude = location.longitude
#print(location)
#print(location.address)
#print((location.latitude, location.longitude))

map_ucsystem= folium.Map(location=[latitude, longitude], zoom_start=5)

# add markers to map
for lat, lng, campus, alias in zip(neighborhoods['latitude'], neighborhoods['longitude'], neighborhoods['Campus'], neighborhoods['Alias']):
    label = '{}'.format(neighborhoods)
    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_ucsystem)

map_ucsystem
```



3.3 FourSquare API and top venues

I use below code to gather venues nearby, through FourSquare API discussed in previous course.



```
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

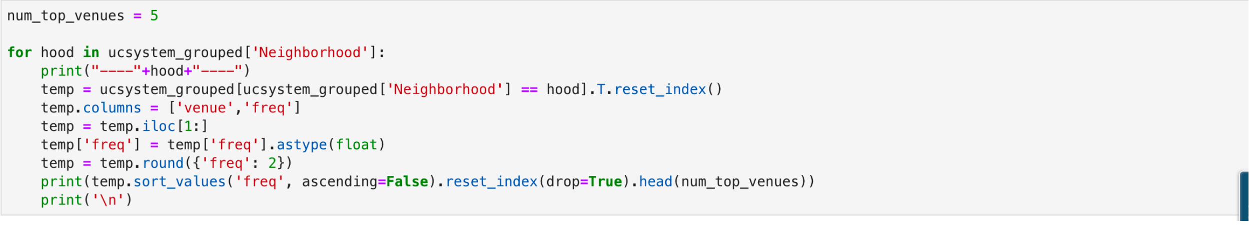
3.4 Top 5/10 venues

I use below code to analyze the data got from above step.

```
 # Now write the code to run the above function on each campus and create a new dataframe called
 ucsystem_venues

 ucsystem_venues = df_ucsystem
 ucsystem_venues = getNearbyVenues(names = ucsystem_venues['Campus'],
                        latitudes  = ucsystem_venues['latitude'],
                        longitudes = ucsystem_venues['longitude']
                        )
ucsystem_data = neighborhoods
# one hot encoding
ucsystem_onehot = pd.get_dummies(ucsystem_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
ucsystem_onehot['Neighborhood'] = ucsystem_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [ucsystem_onehot.columns[-1]] + list(ucsystem_onehot.columns[:-1])
ucsystem_onehot = ucsystem_onehot[fixed_columns]

print(ucsystem_onehot.shape)
```

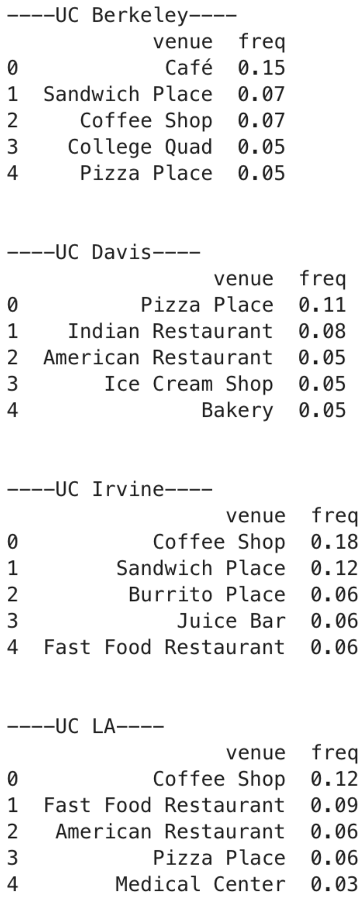
```
# Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of
each category

ucsystem_grouped = ucsystem_onehot.groupby('Neighborhood').mean().reset_index()
print(ucsystem_grouped.shape)
ucsystem_grouped
```

Let's print each neighborhood along with the top 5 most common venues.

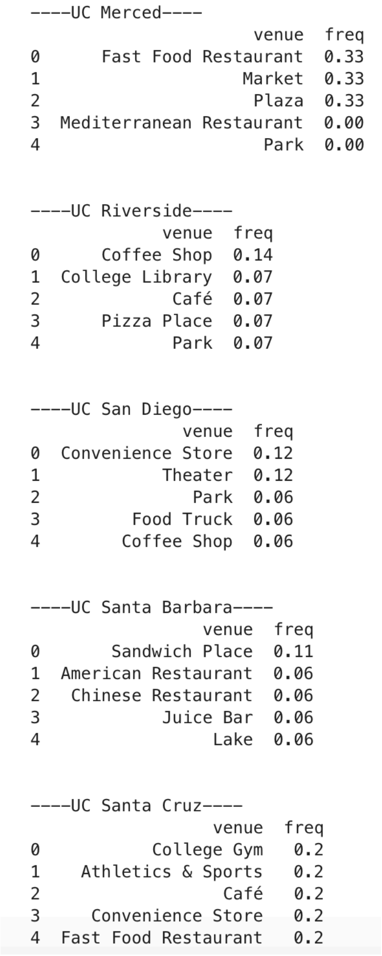


```
num_top_venues = 5

for hood in ucsystem_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = ucsystem_grouped[ucsystem_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

The output is listed below:

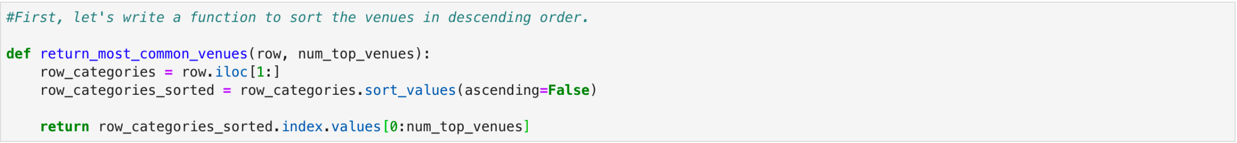


```
----UC Berkeley----
           venue  freq
0           Café  0.15
1 Sandwich Place  0.07
2    Coffee Shop  0.07
3   College Quad  0.05
4    Pizza Place  0.05


----UC Davis----
                venue  freq
0         Pizza Place  0.11
1   Indian Restaurant  0.08
2 American Restaurant  0.05
3      Ice Cream Shop  0.05
4              Bakery  0.05


----UC Irvine----
                 venue  freq
0          Coffee Shop  0.18
1       Sandwich Place  0.12
2        Burrito Place  0.06
3            Juice Bar  0.06
4 Fast Food Restaurant  0.06


----UC LA----
                 venue  freq
0          Coffee Shop  0.12
1 Fast Food Restaurant  0.09
2  American Restaurant  0.06
3          Pizza Place  0.06
4       Medical Center  0.03
```

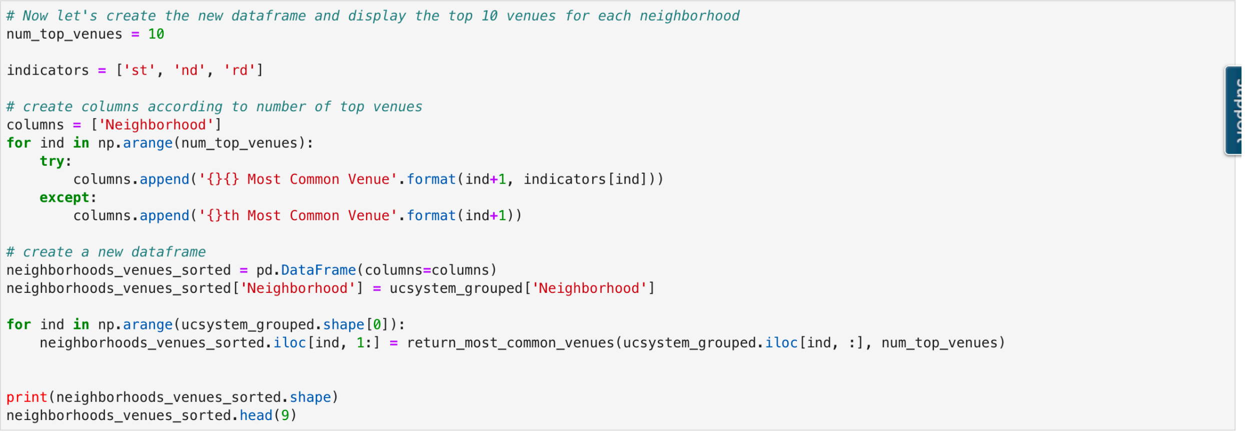


```
----UC Merced----
                      venue  freq
0      Fast Food Restaurant  0.33
1                    Market  0.33
2                     Plaza  0.33
3  Mediterranean Restaurant  0.00
4                      Park  0.00


----UC Riverside----
             venue  freq
0      Coffee Shop  0.14
1  College Library  0.07
2             Café  0.07
3      Pizza Place  0.07
4             Park  0.07


----UC San Diego----
               venue  freq
0  Convenience Store  0.12
1            Theater  0.12
2               Park  0.06
3         Food Truck  0.06
4        Coffee Shop  0.06


----UC Santa Barbara----
                 venue  freq
0       Sandwich Place  0.11
1  American Restaurant  0.06
2   Chinese Restaurant  0.06
3            Juice Bar  0.06
4                 Lake  0.06


----UC Santa Cruz----
                  venue  freq
0           College Gym   0.2
1      Athletics & Sports   0.2
2                  Café   0.2
3     Convenience Store   0.2
4  Fast Food Restaurant   0.2
```

The top 10 common venue was analyzed by below code:

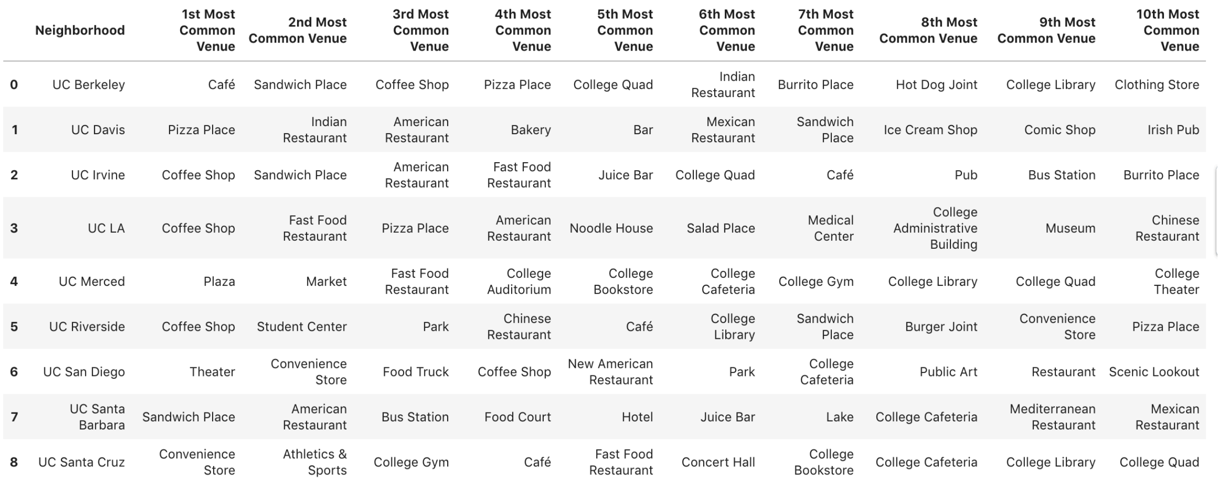


```
#First, let's write a function to sort the venues in descending order.

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]
```



```
# Now let's create the new dataframe and display the top 10 venues for each neighborhood
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighborhood']
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
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = ucsystem_grouped['Neighborhood']

for ind in np.arange(ucsystem_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(ucsystem_grouped.iloc[ind, :], num_top_venues)


print(neighborhoods_venues_sorted.shape)
neighborhoods_venues_sorted.head(9)
```

The top 10 common venue output list below:

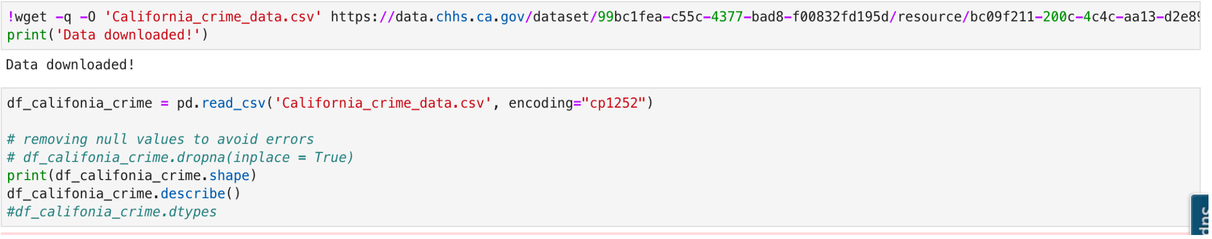


	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	UC Berkeley	Café	Sandwich Place	Coffee Shop	Pizza Place	College Quad	Indian Restaurant	Burrito Place	Hot Dog Joint	College Library	Clothing Store
1	UC Davis	Pizza Place	Indian Restaurant	American Restaurant	Bakery	Bar	Mexican Restaurant	Sandwich Place	Ice Cream Shop	Comic Shop	Irish Pub
2	UC Irvine	Coffee Shop	Sandwich Place	American Restaurant	Fast Food Restaurant	Juice Bar	College Quad	Café	Pub	Bus Station	Burrito Place
3	UC LA	Coffee Shop	Fast Food Restaurant	Pizza Place	American Restaurant	Noodle House	Salad Place	Medical Center	College Administrative Building	Museum	Chinese Restaurant
4	UC Merced	Plaza	Market	Fast Food Restaurant	College Auditorium	College Bookstore	College Cafeteria	College Gym	College Library	College Quad	College Theater
5	UC Riverside	Coffee Shop	Student Center	Park	Chinese Restaurant	Café	College Library	Sandwich Place	Burger Joint	Convenience Store	Pizza Place
6	UC San Diego	Theater	Convenience Store	Food Truck	Coffee Shop	New American Restaurant	Park	College Cafeteria	Public Art	Restaurant	Scenic Lookout
7	UC Santa Barbara	Sandwich Place	American Restaurant	Bus Station	Food Court	Hotel	Juice Bar	Lake	College Cafeteria	Mediterranean Restaurant	Mexican Restaurant
8	UC Santa Cruz	Convenience Store	Athletics & Sports	College Gym	Café	Fast Food Restaurant	Concert Hall	College Bookstore	College Cafeteria	College Library	College Quad

The top 5 common venue analysis is very similar.

3.5 Safety index

Data collection is from California states data organization, after get the dataset, read into data frame, group by campus and get group average by 14 year safety index. The key code is listed below:

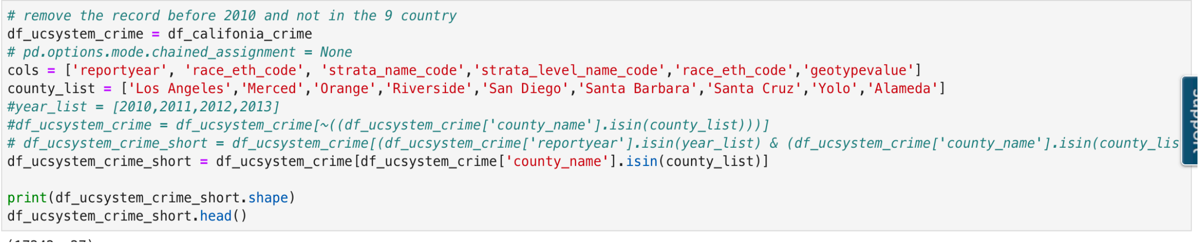


```
!wget -q -O 'California_crime_data.csv' https://data.chhs.ca.gov/dataset/99bc1fea-c55c-4377-bad8-f00832fd195d/resource/bc09f211-200c-4c4c-aa13-d2e8
print('Data downloaded!')
```

```
Data downloaded!
```

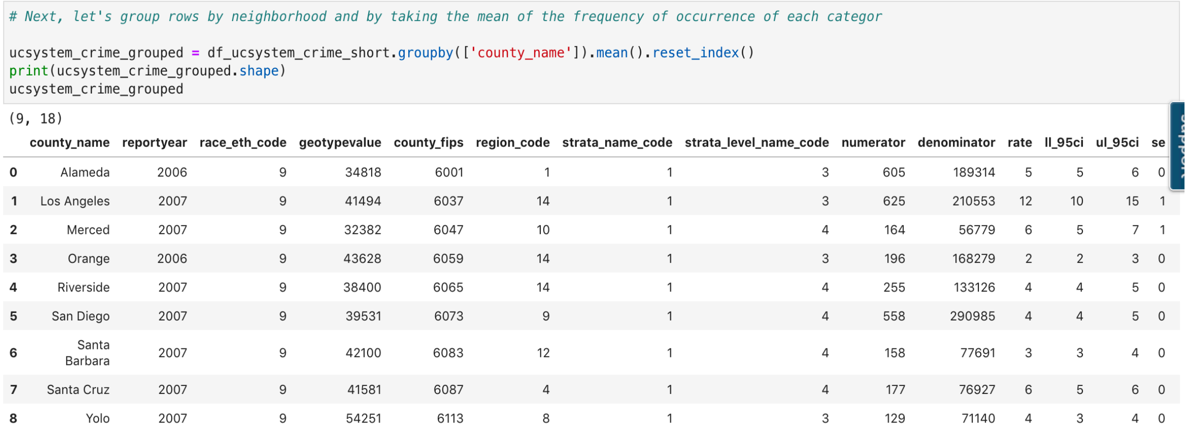
```
df_califonia_crime = pd.read_csv('California_crime_data.csv', encoding="cp1252")

# removing null values to avoid errors
# df_califonia_crime.dropna(inplace = True)
print(df_califonia_crime.shape)
df_califonia_crime.describe()
#df_califonia_crime.dtypes
```

Data cleaning, drop some dataset, filter other non-interested area.



```
# remove the record before 2010 and not in the 9 country
df_ucsystem_crime = df_califonia_crime
# pd.options.mode.chained_assignment = None
cols = ['reportyear', 'race_eth_code', 'strata_name_code','strata_level_name_code','race_eth_code','geotypevalue']
county_list = ['Los Angeles','Merced','Orange','Riverside','San Diego','Santa Barbara','Santa Cruz','Yolo','Alameda']
#year_list = [2010,2011,2012,2013]
#df_ucsystem_crime = df_ucsystem_crime[~((df_ucsystem_crime['county_name'].isin(county_list)))]
# df_ucsystem_crime_short = df_ucsystem_crime[(df_ucsystem_crime['reportyear'].isin(year_list) & (df_ucsystem_crime['county_name'].isin(county_lis
df_ucsystem_crime_short = df_ucsystem_crime[df_ucsystem_crime['county_name'].isin(county_list)]

print(df_ucsystem_crime_short.shape)
df_ucsystem_crime_short.head()
```

Data Grouping, statistics average

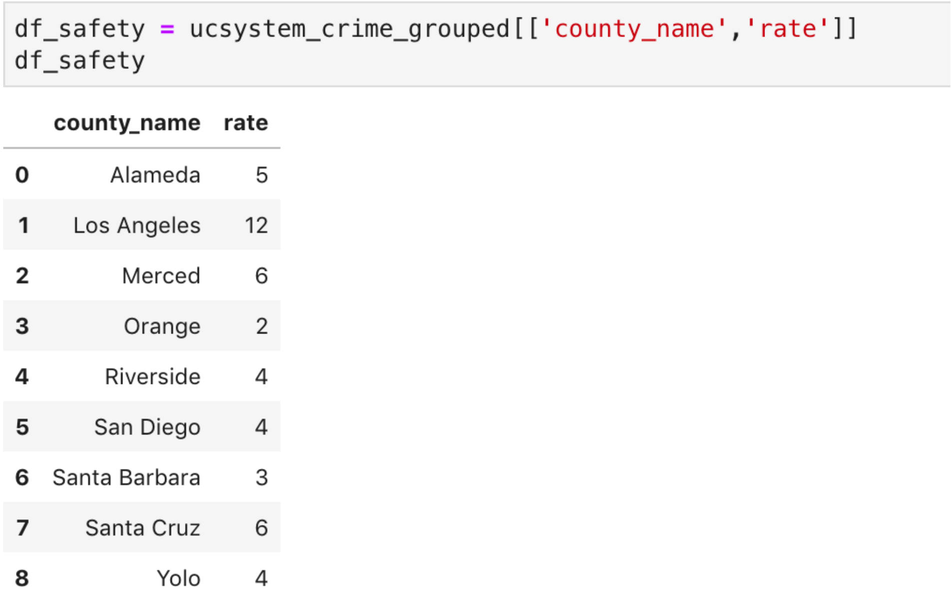


```
# Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each categor

ucsystem_crime_grouped = df_ucsystem_crime_short.groupby(['county_name']).mean().reset_index()
print(ucsystem_crime_grouped.shape)
ucsystem_crime_grouped
```

```
(9, 18)
```

	county_name	reportyear	race_eth_code	geotypevalue	county_fips	region_code	strata_name_code	strata_level_name_code	numerator	denominator	rate	ll_95ci	ul_95ci	se
0	Alameda	2006	9	34818	6001	1	1	3	605	189314	5	5	6	0
1	Los Angeles	2007	9	41494	6037	14	1	3	625	210553	12	10	15	1
2	Merced	2007	9	32382	6047	10	1	4	164	56779	6	5	7	1
3	Orange	2006	9	43628	6059	14	1	3	196	168279	2	2	3	0
4	Riverside	2007	9	38400	6065	14	1	4	255	133126	4	4	5	0
5	San Diego	2007	9	39531	6073	9	1	4	558	290985	4	4	5	0
6	Santa Barbara	2007	9	42100	6083	12	1	4	158	77691	3	3	4	0
7	Santa Cruz	2007	9	41581	6087	4	1	4	177	76927	6	5	6	0
8	Yolo	2007	9	54251	6113	8	1	3	129	71140	4	3	4	0

Final safety index for visualization.



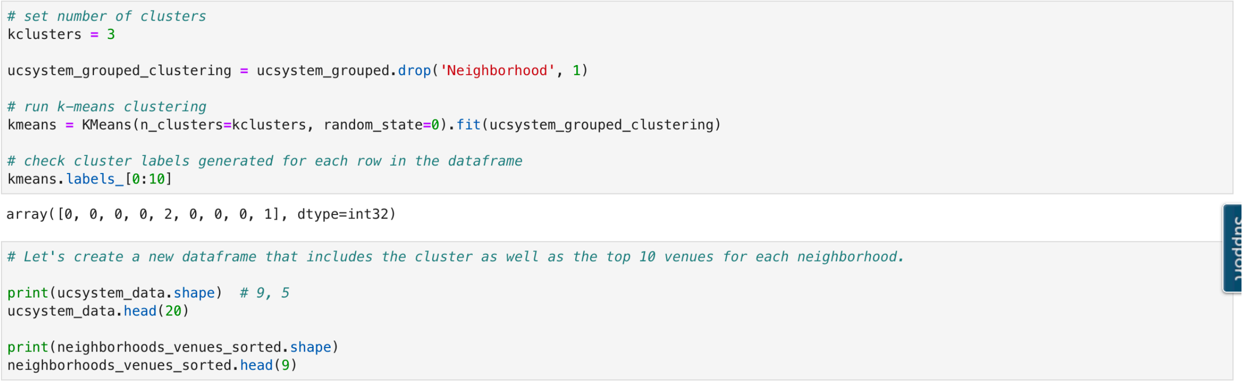
```
df_safety = ucsystem_crime_grouped[['county_name','rate']]
df_safety
```

	county_name	rate
0	Alameda	5
1	Los Angeles	12
2	Merced	6
3	Orange	2
4	Riverside	4
5	San Diego	4
6	Santa Barbara	3
7	Santa Cruz	6
8	Yolo	4

4. Result

4.1 Venues around campus

Use K-Mean to cluster method, code listed below:

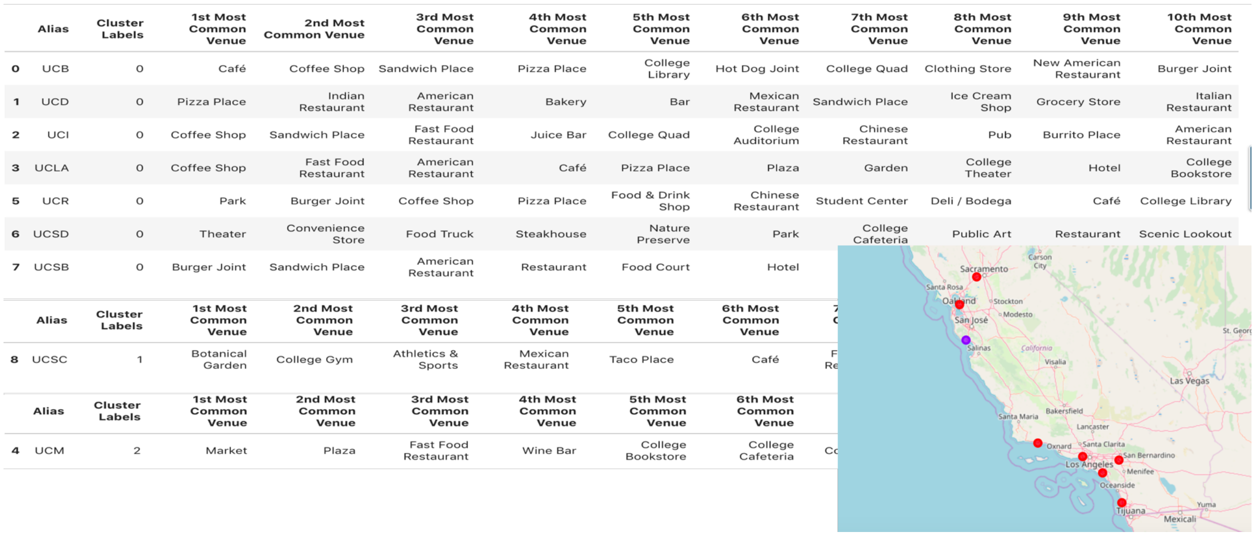


```
# set number of clusters
kclusters = 3

ucsystem_grouped_clustering = ucsystem_grouped.drop('Neighborhood', 1)

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

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

```
array([0, 0, 0, 0, 2, 0, 0, 0, 1], dtype=int32)
```

```
# Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

print(ucsystem_data.shape)  # 9, 5
ucsystem_data.head(20)

print(neighborhoods_venues_sorted.shape)
neighborhoods_venues_sorted.head(9)
```

- K-means 3 cluster layout



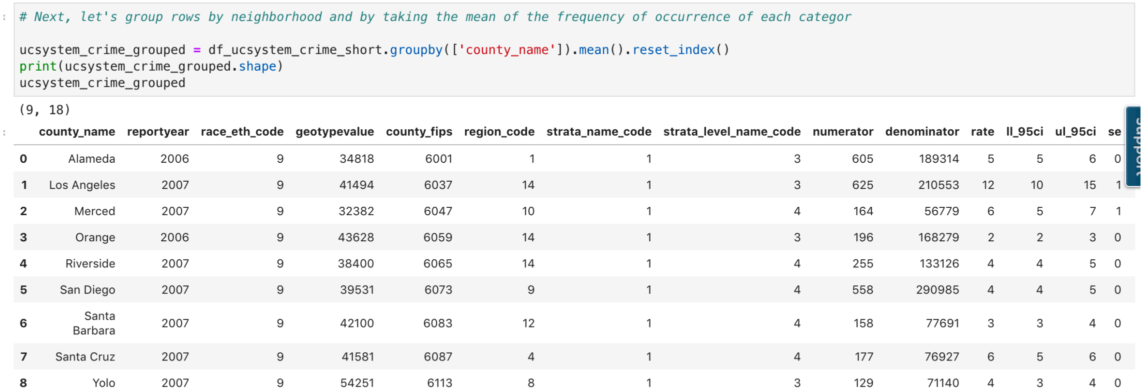
	Alias	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	UCB	0	Café	Coffee Shop	Sandwich Place	Pizza Place	College Library	Hot Dog Joint	College Quad	Clothing Store	New American Restaurant	Burger Joint
1	UCD	0	Pizza Place	Indian Restaurant	American Restaurant	Bakery	Bar	Mexican Restaurant	Sandwich Place	Ice Cream Shop	Grocery Store	Italian Restaurant
2	UCI	0	Coffee Shop	Sandwich Place	Fast Food Restaurant	Juice Bar	College Quad	College Auditorium	Chinese Restaurant	Pub	Burrito Place	American Restaurant
3	UCLA	0	Coffee Shop	Fast Food Restaurant	American Restaurant	Café	Pizza Place	Plaza	Garden	College Theater	Hotel	College Bookstore
5	UCR	0	Park	Burger Joint	Coffee Shop	Pizza Place	Food & Drink Shop	Chinese Restaurant	Student Center	Deli / Bodega	Café	College Library
6	UCSD	0	Theater	Convenience Store	Food Truck	Steakhouse	Nature Preserve	Park	College Cafeteria	Public Art	Restaurant	Scenic Lookout
7	UCSB	0	Burger Joint	Sandwich Place	American Restaurant	Restaurant	Food Court	Hotel				

	Alias	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
8	UCSC	1	Botanical Garden	College Gym	Athletics & Sports	Mexican Restaurant	Taco Place	Café

	Alias	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue
4	UCM	2	Market	Plaza	Fast Food Restaurant	Wine Bar	College Bookstore	College Cafeteria

4.2 Safety index around campus

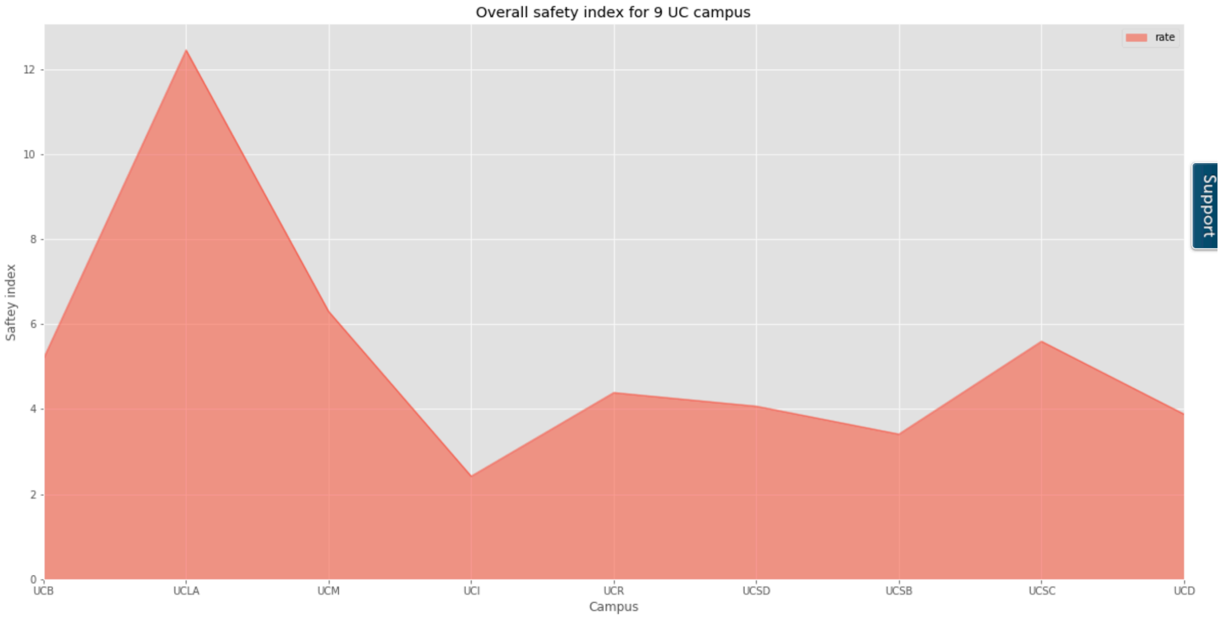
Safety index calculation and visualization code are listed below:



```
# Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each categor

ucsystem_crime_grouped = df_ucsystem_crime_short.groupby(['county_name']).mean().reset_index()
print(ucsystem_crime_grouped.shape)
ucsystem_crime_grouped
```

```
(9, 18)
```

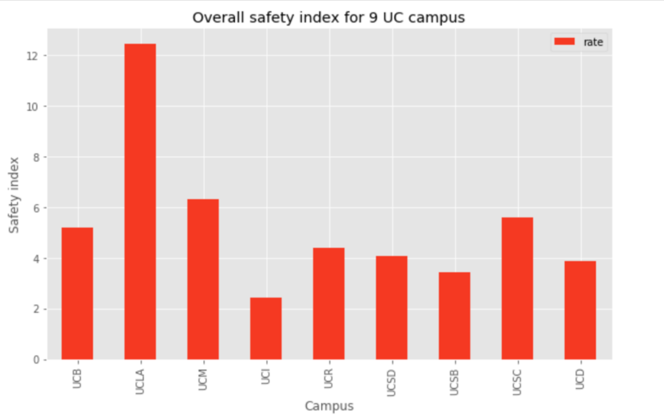
	county_name	reportyear	race_eth_code	geotypevalue	county_fips	region_code	strata_name_code	strata_level_name_code	numerator	denominator	rate	ll_95ci	ul_95ci	se
0	Alameda	2006	9	34818	6001	1	1	3	605	189314	5	5	6	0
1	Los Angeles	2007	9	41494	6037	14	1	3	625	210553	12	10	15	1
2	Merced	2007	9	32382	6047	10	1	4	164	56779	6	5	7	1
3	Orange	2006	9	43628	6059	14	1	3	196	168279	2	2	3	0
4	Riverside	2007	9	38400	6065	14	1	4	255	133126	4	4	5	0
5	San Diego	2007	9	39531	6073	9	1	4	558	290985	4	4	5	0
6	Santa Barbara	2007	9	42100	6083	12	1	4	158	77691	3	3	4	0
7	Santa Cruz	2007	9	41581	6087	4	1	4	177	76927	6	5	6	0
8	Yolo	2007	9	54251	6113	8	1	3	129	71140	4	3	4	0



Result for safety index around the 9 campuses

The campuses safety ranking are:

- UC Irving/Orange; [2]
- UCSB [3]
- UCR & UCSD, UCD [4]
- UCB [5]
- UCSC, UCM [6]
- UCLA [12]



5. Discussion

- Activities around campus
    - There are many people working/Living in campus, in return, different kind of food facility is top venues there.
    - UCSD is easy to access for convenient store and theater in top 2
    - UCSC Gym is the top venue
    - For Chinese international student, UCR and UCI is very friendly whilst UCSB is very closing

- Safety Index
    - The most safety campus is UCI, UC Irving/Orange; The second is UCSB [3] and UCR & UCSD, UCD [4] are very good as well
    - UCB[Berkeley] is not very good in frame in the past, but from 2000~2013 average, UCB is actually okay in the middle
    - There are some challenge in UCLA compare to others in safety condition

6. Conclusion

- Different kind of food facility are top venues around campuses

- The most safety campus is UCI, UC Irving/Orange

- Some information and conclusion in this report. But due to the nature of unsupervised learning, metrics can only guide, a best cluster determination algorithm is always arbitrary and should suit the project goal and intended audience

- There are other factors no consider this time, such as campus itself like application admit rate, yield rate and annual cost include fee and living cost. This could be put into future work steams

- Crime type could be further group by subtype, DBSCAN could be very good start point. It worth a trial in future as well.