

# Capstone Project — The Battle of Neighborhoods

## Suitable New Coffee Shop Locations in Riyadh



Riyadh at night

### 1. Introduction and Discussion of the Business Objective and Problem

The Middle East is seeing an enormous growth in its coffee industry. Here, particularly in Dubai and Saudi Arabia, specialty coffee shop chains are thriving. There is also a growing number of professional events, which the organizers describe as the largest coffee and chocolate exhibition in the Middle East.

The region has opportunity for investors and is providing a new market for global producers.

This project explores coffee shops and related opportunities in Riyadh, Saudi Arabia. It will help make data-driven decisions about the new locations that are best suited to their existing Riyadh coffee shops. This will be a big part of the decision-making process, the other being on-the-ground district qualitative research after analyzing and examining its data and paper.

### 2. Data Preparation and How It Solves The Problem

#### 2.1 Using Geopy to get the postal codes, names and coordinates of all districts in Riyadh.

There is no published list of districts in Riyadh. Using Geopy and pandas the district name, postal codes and coordinates are obtained. The first part of following method uses geocoding to find the latitude and longitude of a any postal code and uses reverse geocoding to get the name of the district/neighborhood.

#### 2.2 Using Foursquare API to get the top 10 districts having most coffee shops.

Foursquare data is very comprehensive and it powers location data for Apple, Uber etc. As a part of the assignment, the Foursquare API will be used to retrieve information about top 10 Districts having coffee shops of Riyadh.

### 3. Methodology and Exploratory Data Analysis

In order to obtain the complete list of postal codes of Riyadh, further analysis were required. After many trials the best way was to find a range of postal codes, loop them and let the above method do the job. Using <https://maps.address.gov.sa/> the range was considered to be between 12211 and 15000.

```
def get_riyadh_location_data(postal_code):
    location_info_list = []
    headers = {'User-agent': 'Googlebot'}

    KEY = ''
    url = 'https://maps.googleapis.com/maps/api/geocode/json?
    address=riyadh+' + postal_code + '&key=' + KEY
    response = requests.get(url, headers=headers)
    result = json.loads(response.text)

    if result['status'] == 'OK':
        if result['results'][0]['types'][0] == "postal_code":
            latitude = result['results'][0]['geometry']
            ['location']['lat']

            longitude = result['results'][0]['geometry']
            ['location']['lng']
            # Reverse geocoding
            url
            = 'https://maps.googleapis.com/maps/api/geocode/json?latlng=' +
            str(latitude) + ',' + str(longitude) + '&key=' + KEY
            response = requests.get(url, headers=headers)
            result = json.loads(response.text)
            neighborhood = ''
            sublocality_level_1 = ''
            for res in result['results']:
                for i in res['address_components']:
                    for i in add['types']:
                        if i == "neighborhood":
                            neighborhood = add['long_name']
                        if i == "sublocality_level_1":
                            sublocality_level_1 =

add['long_name']

break
```

Looping for each number,

```
riyadh_location_data = []

# for each postal code extract relevant data
for i in range(12211, 15000):
    try:
        postal_code = str(i)
    except:
        postal_code = None

    riyadh_location_data.append(get_riyadh_location_data(postal_code))
```

	PostalCode	District	Neighborhood	Latitude	Longitude
0	12211	As Sulimaniyah	Al Olaya	24.687631	46.686229
1	12212	As Sulimaniyah	Al Olaya	24.692344	46.684418
2	12213	As Sulimaniyah	Al Olaya	24.699387	46.680796
3	12214	As Sulimaniyah	Al Olaya	24.707189	46.675966
4	12215	Al Wurud	Al Wurud	24.717371	46.671138
5	12216	Al Wurud	Al Wurud	24.725982	46.666913
6	None	None	None	NaN	NaN
7	None	None	None	NaN	NaN
8	None	None	None	NaN	NaN
9	None	None	None	NaN	NaN

Dropping NaN rows,

```
riyadh_data_dropped = pd.DataFrame(riyadh_location_data,
columns=column_names).dropna()

df = riyadh_data_dropped
df.Neighborhood = np.where(df.Neighborhood == '', df.District,
df.Neighborhood)
df['Neighborhood'].replace('', np.nan, inplace=True)
df.dropna()
```

	PostalCode	Neighborhood	Latitude	Longitude
0	12211	Al Olaya	24.687631	46.686229
1	12212	Al Olaya	24.692344	46.684418
2	12213	Al Olaya	24.699387	46.680796
3	12214	Al Olaya	24.707189	46.675966
4	12215	Al Wurud	24.717371	46.671138
5	12216	Al Wurud	24.725982	46.666913
10	12221	Al Olaya	24.688704	46.695890
11	12222	Al Olaya	24.695643	46.688644
12	12223	As Sulimaniyah	24.699965	46.700721
20	12231	As Sulimaniyah	24.687393	46.705553

Finally a clean data frame is obtained including Postal Code, Neighborhood, Latitude and Longitude columns.

Using FourSquare API we will find all venues for each neighborhood.

```
def getNearbyVenues(names, latitudes, longitudes):

    CLIENT_ID = '' # Foursquare ID, note there is a daily call
    quota limit
    CLIENT_SECRET = '' # Foursquare Secret, note there is a daily
    call quota it it
    VERSION = '20180605' # Foursquare API version
    radius=500
    LIMIT=100
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):

        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']
```

## Finding Top 10 most common Neighborhood

```
riyadh_venues = getNearbyVenues(names=riyadh_data['Neighborhood'],
latitudes=riyadh_data['Latitude'],
longitudes=riyadh_data['Longitude'])

riyadh_venues.drop_duplicates(subset=['Venue', 'Venue Latitude',
'Venue Longitude', 'Venue Category'], keep='first', inplace =
True)
riyadh_venues.groupby('Neighborhood').count()
```

## Create a Data-Frame out of it to Concentrate Only on Coffee Shop

```
Riyadh_Venues_Coffee = riyadh_venues[riyadh_venues['Venue
Category']\str.contains('Coffee
Shop|Café')].reset_index(drop=True)
Riyadh_Venues_Coffee.index = np.arange(1,
len(Riyadh_Venues_Coffee)+1)
print ("Shape of The Data-Frame with Venue Category only Coffee
Shop: ", Riyadh_Venues_Coffee.shape)
Riyadh_Venues_Coffee.head(10)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
1	Al Olaya	24.687631	46.686229	Starbucks (ستاربكس)	24.689529	46.685666	Coffee Shop
2	Al Olaya	24.687631	46.686229	dr.CAFE COFFEE   د.كاف   V12	24.687085	46.683927	Coffee Shop
3	Al Olaya	24.687631	46.686229	CAF Cafe	24.688757	46.684281	Coffee Shop
4	Al Olaya	24.687631	46.686229	dr.CAFE COFFEE   د.كاف	24.690040	46.686827	Coffee Shop
5	Al Olaya	24.687631	46.686229	Starbucks	24.689646	46.682810	Coffee Shop
6	Al Olaya	24.687631	46.686229	Roma Café (روما كافيه)	24.686190	46.686661	Café
7	Al Olaya	24.687631	46.686229	Coffee Day (بري القهوه)	24.690730	46.684504	Coffee Shop
8	Al Olaya	24.687631	46.686229	Ko coffee (كو كافي)	24.685547	46.690131	Café
9	Al Olaya	24.687631	46.686229	Costa Coffee (كوستا كافيه)	24.690741	46.684379	Coffee Shop
10	Al Olaya	24.687631	46.686229	Coffeeshop Company	24.690817	46.684462	Coffee Shop

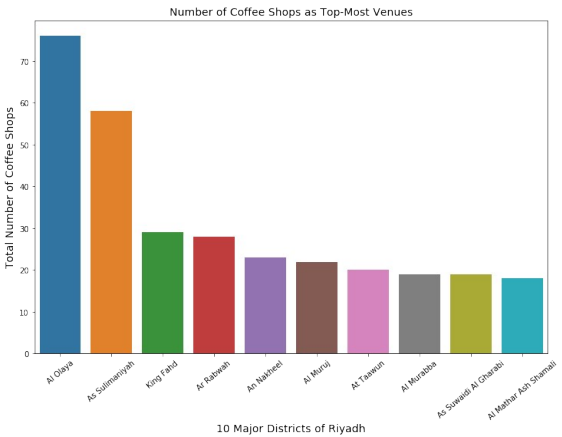
## Let's now find the venue numbers per neighborhood

```
Riyadh_Venues_Coffee_grouped =
Riyadh_Venues_Coffee.groupby('Neighborhood').count()
Riyadh_Venues_Coffee_grouped.head(10)
```

	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighborhood						
Ad Dar Al Baida	1	1	1	1	1	1
Ad Dhubbat	12	12	12	12	12	12
Ad Dirah	6	6	6	6	6	6
Ad Diriyah Al Jadidah	1	1	1	1	1	1
Ad Dubiyah	1	1	1	1	1	1
Ad Durahimiyah	2	2	2	2	2	2
Al Aarid	4	4	4	4	4	4
Al Andalus	1	1	1	1	1	1
Al Aqiq	3	3	3	3	3	3
Al Aziziyah	1	1	1	1	1	1

Time to find top 10 major nighborhood of Riyadh having most number of coffee shops.

```
Riyadh_10_Dist_Venues_coffee =
riyadh_venues.groupby(['Neighborhood'])['Venue
Category'].apply(lambda x: x[x.str.contains('Coffee
Shop|Café')].count())
Riyadh_10_Dist_Venues_coffee_df =
Riyadh_10_Dist_Venues_coffee.to_frame().reset_index()
Riyadh_10_Dist_Venues_coffee_df.columns = ['Neighborhood', 'Number
of Coffee Shops']
Riyadh_10_Dist_Venues_coffee_df.index = np.arange(1,
len(Riyadh_10_Dist_Venues_coffee_df)+1)
Riyadh_10_Dist_Venues_coffee_df_sorted =
(Riyadh_10_Dist_Venues_coffee_df[['Neighborhood', 'Number of
Coffee Shops']].sort_values('Number of Coffee Shops',
ascending=False).nlargest(10, 'Number of Coffee Shops'))
fig = plt.figure(figsize=(12,8))
bar = sns.barplot(x='Neighborhood', y='Number of Coffee Shops',
data=Riyadh_10_Dist_Venues_coffee_df_sorted)
bar.set_ylabel("Total Number of Coffee Shops", fontsize=14)
bar.set_xlabel("10 Major Districts of Riyadh", fontsize=14)
bar.set_xticklabels(bar.get_xticklabels(),rotation=40)
bar.set_title("Number of Coffee Shops as Top-Most Venues",
fontsize=14)
plt.show()
```



	Neighborhood	Number of Coffee Shops
57	Al Olaya	76
103	As Sulmaniyah	58
132	King Fahd	29
88	Ar Rabwah	28
79	An Nakheel	23
63	Al Muruj	22
110	At Taawun	20
61	Al Murabba	19
105	As Suwaidi Al Gharabi	19
47	Al Mathar Ash Shamali	18

Finally let's visualize the the major neighborhoods using folium library.

```
map_coffeeshops = folium.Map(location=[Riyadh_latitude,
Riyadh_longitude], zoom_start=11, tiles="openstreetmap",
attr="<a href=https://github.com/python-visualization/folium/>Folium</a>")

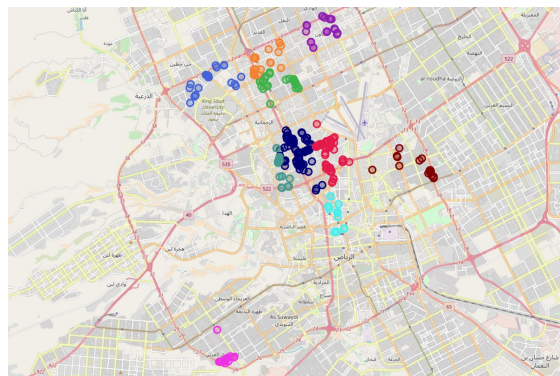
Districts =
Riyadh_10_Dist_Venues_coffee_df_sorted["Neighborhood"].values.tolist()

x = np.arange(len(Districts))

rainbow = ['#e6194b', '#3cb44b', '#800000', '#4363d8',
'#f58231', '#911eb4', '#46f0f0', '#f032e6', '#469990', '#000075']

for lat, lon, poi, distr in
zip(Riyadh_Venues_10_Dist_Coffee_Shops['Venue Latitude'],
Riyadh_Venues_10_Dist_Coffee_Shops['Venue Longitude'],
Riyadh_Venues_10_Dist_Coffee_Shops['Venue Category'],
Riyadh_Venues_10_Dist_Coffee_Shops['Neighborhood']):
    label = folium.Popup(str(poi) + ' ' + str(distr),
    parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=7,
        popup=label,
        color=rainbow[Districts.index(distr)-1],
        fill=True,
        fill_color=rainbow[Districts.index(distr)-1],
        fill_opacity=0.3).add_to(map_coffeeshops)

map_coffeeshops
```



Top 10 Coffee Shop Neighborhoods

## 4. Results

As expected, central districts have the highest number of coffee shops.

- Al Olaya is the neighborhood with highest number of Coffee Shops.
- As Sulimaniyah is the second followed by King Fahd.
- As Suwaidi Al Gharabi is supprisingly in top 10 being far away from the centre of the city.

## 5. Conclusions

There are many ways in which this research could have been carried out based on di erent methodologies and possibly di erent sources of data. Evaluation and analysis are not an objective, but rather a starting point to direct the next part of the process in identifying speci c locations for the shop.

The process could have been drawn out without leveraging data to make focused decisions, leading to the opening of new stores for this retailer in sub-standard areas. Data has helped to deliver a better strategy, and these data-driven decisions will ultimately lead to a better solution.

