Import necessary Libraries

In [1]:

```
#!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into lat
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

#!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library

print('Folium installed')
print('Libraries imported.')
```

```
Folium installed
Libraries imported.
```

Getting the data

All datasets were already downloaded and stored in the IBM Db2 database. We need to first get all the data for further analysis.

Connect to the database

In [2]:

```
%load_ext sql
%sql ibm_db_sa://
```

Out[2]:

```
'Connected: lsp45830@BLUDB'
```

Retreive data from the database

In [3]:

```
res = %sql SELECT * FROM CENSUS_DATA;

#retrieve the query results into a pandas dataframe
df_census = res.DataFrame()
df_census.head()
```

```
 * ibm_db_sa://lsp45830:***@dashdb-txn-sbox-yp-dal09-03.services.dal.bluemi
x.net:50000/BLUDB
Done.
```

Out[3]:

	Community_Area_Number	community_area_name	percent_of_housing_crowded	percent_hou
0	1.0	Rogers Park	7.7	
1	2.0	West Ridge	7.8	
2	3.0	Uptown	3.8	
3	4.0	Lincoln Square	3.4	
4	5.0	North Center	0.3	

In [4]:

```
# sql query for Public Schools
res = %sql SELECT * FROM CHICAGO_PUBLIC_SCHOOLS;

#retrieve the query results into a pandas dataframe
df_schools = res.DataFrame()
df_schools.head()
```

```
 * ibm_db_sa://lsp45830:***@dashdb-txn-sbox-yp-dal09-03.services.dal.bluemi
x.net:50000/BLUDB
Done.
```

Out[4]:

	School_ID	Name_of_School	Elementary, Middle, or High School	Street_Address	City	State	ZIP_Code	Phone
0	610038	Abraham Lincoln Elementary School	ES	615 W Kemper Pl	Chicago	IL	60614	(773)
1	610281	Adam Clayton Powell Paideia Community Academy ...	ES	7511 S South Shore Dr	Chicago	IL	60649	(773)
2	610185	Adlai E Stevenson Elementary School	ES	8010 S Kostner Ave	Chicago	IL	60652	(773)
3	609993	Agustin Lara Elementary Academy	ES	4619 S Wolcott Ave	Chicago	IL	60609	(773)
4	610513	Air Force Academy High School	HS	3630 S Wells St	Chicago	IL	60609	(773)

5 rows × 79 columns

In [5]:

```
# sql query for Crime Data
res = %sql SELECT * FROM CHICAGO_CRIME_DATA;

#retrieve the query results into a pandas dataframe
df_crime = res.DataFrame()
df_crime.head()
```

```
 * ibm_db_sa://lsp45830:***@dashdb-txn-sbox-yp-dal09-03.services.dal.bluemi
x.net:50000/BLUDB
Done.
```

Out[5]:

	id	Case_Number	Date	Block	iucr	Primary_Type	Description	Location
0	3512276	HK587712	08/28/2004 05:50:56 PM	047XX S KEDZIE AVE	0890	THEFT	FROM BUILDING	SI
1	3406613	HK456306	06/26/2004 12:40:00 PM	009XX N CENTRAL PARK AVE	0820	THEFT	$500 AND UNDER	
2	8002131	HT233595	04/04/2011 05:45:00 AM	043XX S WABASH AVE	0820	THEFT	$500 AND UNDER	HOME/F
3	7903289	HT133522	12/30/2010 04:30:00 PM	083XX S KINGSTON AVE	0840	THEFT	FINANCIAL ID THEFT: OVER $300	
4	10402076	HZ138551	02/02/2016 07:30:00 PM	033XX W 66TH ST	0820	THEFT	$500 AND UNDER	

5 rows × 22 columns

Let's visualize crime in some communities

In [6]:

```
df_community = df_schools[['Community_Area_Number', 'Community_Area_Name']].c
df_community.dtypes
```

Out[6]:

```
Community_Area_Number     int64
Community_Area_Name      object
dtype: object
```

In [7]:

```
# convert some object columns to float
df_crime.dropna(inplace=True)
df_crime = df_crime.apply(pd.to_numeric, errors='ignore')
df_crime.dtypes
```

Out[7]:

```
id                        int64
Case_Number              object
Date                     object
Block                    object
iucr                     object
Primary_Type             object
Description              object
Location_Description     object
Arrest                   object
Domestic                 object
Beat                      int64
District                float64
Ward                    float64
Community_Area          float64
FBI_Code                 object
X_Coordinate            float64
Y_Coordinate            float64
Year                      int64
Updated_On               object
Latitude                float64
Longitude               float64
Location                 object
dtype: object
```

In [8]:

```
# Group by Community_Area and count the number of crime
df_grouped = df_crime
df_grouped = df_grouped.groupby('Community_Area', as_index=False)['Community_

# get Community_Area_Name
df_grouped.rename(columns={'Community_Area':'Community_Area_Number'}, inplace
df_grouped = pd.merge(df_grouped,
                      df_community,
                      how='left',
                      on = ['Community_Area_Number']).drop(['Community_Area_N

# sort dataframe on 'Count' column (descending)
df_top15 = df_grouped.set_index(['Community_Area_Name'])
df_top15.sort_values(by='Count', ascending=True, inplace=True)

# get top 15 areas
df_top15 = df_top15['Count'].tail(15)
df_top15
```

Out[8]:

```
Community_Area_Name
WEST GARFIELD PARK        10
BRIGHTON PARK             10
LAKE VIEW                 11
ROSELAND                  11
GREATER GRAND CROSSING    11
WEST ENGLEWOOD            12
CHICAGO LAWN              12
WEST TOWN                 13
AUBURN GRESHAM            14
NEAR NORTH SIDE           15
NEAR WEST SIDE            16
NORTH LAWNDALE            16
ENGLEWOOD                 21
HUMBOLDT PARK             21
AUSTIN                    42
Name: Count, dtype: int64
```

In [9]:

```
# use the inline backend to generate the plots within the browser
%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.style.use('ggplot') # optional: for ggplot-like style

# check for latest version of Matplotlib
print ('Matplotlib version: ', mpl.__version__) # >= 2.0.0
```

```
Matplotlib version:  2.2.3
```

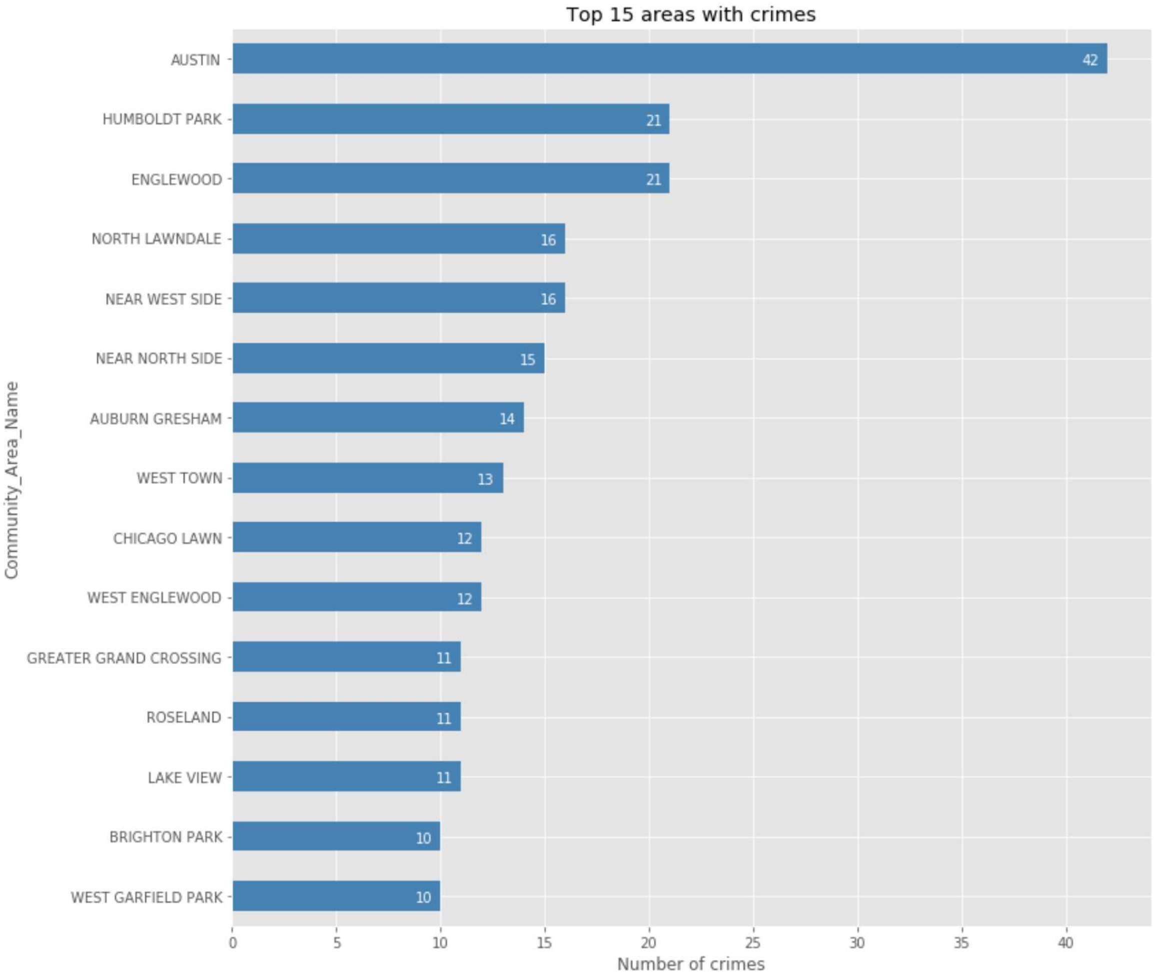
In [10]:

```
df_top15.plot(kind='barh', figsize=(12, 12), color='steelblue')
plt.xlabel('Number of crimes')
plt.title('Top 15 areas with crimes')

# annotate value labels to each country
for index, value in enumerate(df_top15):
    label = str(value) # format int with commas

    # place text at the end of bar (subtracting 47000 from x, and 0.1 from y
    plt.annotate(label, xy=(value - 1.2, index - 0.10), color='white')

plt.show()
```



In [11]:

```
df_top15 = df_grouped.copy()
df_top15.sort_values(by='Count', ascending=True, inplace=True)

# get top 15 areas
df_top15 = df_top15.head(15)
print(df_top15)

lats = []
longs = []
geolocator = Nominatim()
for area in df_top15.loc[:,'Community_Area_Name']:
    location = geolocator.geocode(area)
    lats.append(location.latitude)
    longs.append(location.longitude)

df_top15['Latitude'] = lats
df_top15['Longitude'] = longs
df_top15.drop(['Count'], axis=1, inplace=True)
df_top15.reset_index(inplace=True)
print(df_top15)
```

```
    Count Community_Area_Name
30      1     NEAR SOUTH SIDE
67      1         MORGAN PARK
41      1           BURNSIDE
53      1         BRIDGEPORT
49      1          HEGEWISCH
10      1         FOREST GLEN
35      2          HYDE PARK
15      2          MONTCLARE
46      2          EAST SIDE
48      2          RIVERDALE
69      2          EDGEWATER
50      2      GARFIELD RIDGE
52      2       MCKINLEY PARK
57      2           CLEARING
32      2         FULLER PARK

/home/biosys/anaconda3/lib/python3.6/site-packages/geopy/geocoders/osm.p
y:143: UserWarning: Using Nominatim with the default "geopy/1.17.0" `user
 agent` is strongly discouraged, as it violates Nominatim's ToS https://o
```

In [12]:

```
# create map of top 15 crime areas in Chicago using latitude and longitude va
latitude = df_top15.loc[0, 'Latitude']
longitude = df_top15.loc[0, 'Longitude']
print(type(latitude))
map_crime = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lng, label in zip(df_top15['Latitude'], df_top15['Longitude'], df_to
    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_crime)

'''
for lat, lng, label in zip(healthy_schools['Latitude'], healthy_schools['Long
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_crime)    '''
map_crime
```

```
<class 'numpy.float64'>
```

Out[12]:

+
−
47A
47B
Lincoln Park
48B
IL 64
49A
49A
IL 50
ark
51B
51I
51H
Chicago
23A
23B
23B
25
26A
27C
28A
52A
University Village/
Little Italy
South Loop
Pilsen
53

Let's find top 10 community by number of puclic schoools

In [14]:

```
# Top 10 community by number of puclic schoools
res = %sql SELECT "Community_Area_Name",\
        COUNT("Elementary, Middle, or High School") AS "Number of Schools" FR
        GROUP BY "Community_Area_Name"\
        ORDER BY "Number of Schools" DESC\
        LIMIT 10;
community_top10 = res.DataFrame()
community_top10 = community_top10.apply(pd.to_numeric, errors='ignore')
community_top10.set_index(['Community_Area_Name'], inplace=True)
community_top10
```

```
 * ibm_db_sa://lsp45830:***@dashdb-txn-sbox-yp-dal09-03.services.dal.bluemi
x.net:50000/BLUDB
Done.
```

Out[14]:

	Number of Schools
Community_Area_Name	
AUSTIN	23.0
SOUTH LAWNDALE	22.0
WEST TOWN	20.0
ENGLEWOOD	17.0
NEAR WEST SIDE	16.0
NORTH LAWNDALE	16.0
EAST GARFIELD PARK	13.0
ROSELAND	13.0
NEW CITY	13.0
HUMBOLDT PARK	13.0

In [15]:

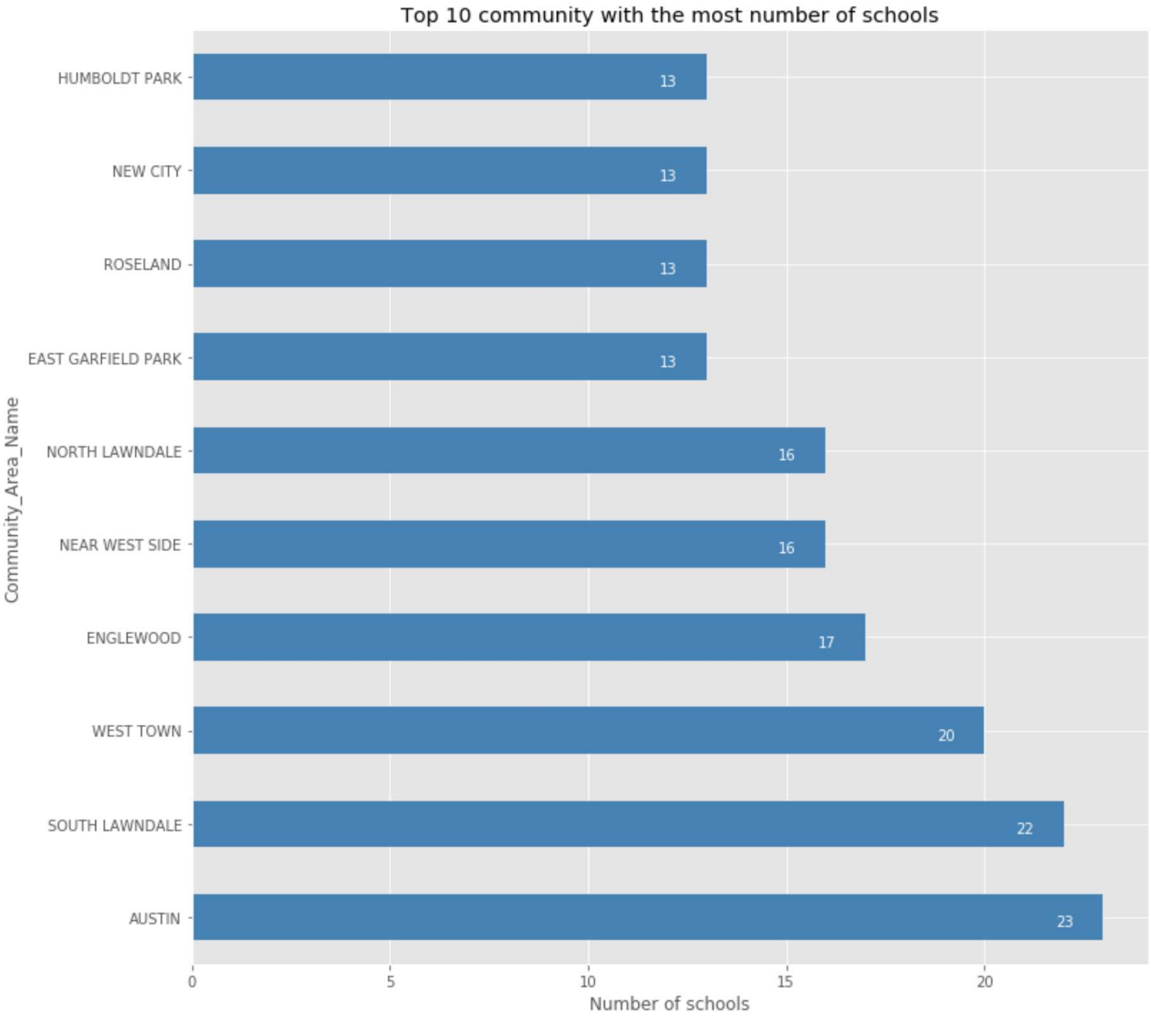
```
df_top10 = community_top10['Number of Schools']

df_top10.plot(kind='barh', figsize=(12, 12), color='steelblue')

# annotate value labels to each country
for index, value in enumerate(df_top10):
    label = format(int(value), ',') # format int with commas

    # place text at the end of bar (subtracting 47000 from x, and 0.1 from y
    plt.annotate(label, xy=(value - 1.2, index - 0.10), color='white')

plt.xlabel('Number of schools')
plt.title('Top 10 community with the most number of schools')
plt.show()
```



It seems to be that SOUTH LAWNDALE is a community with a good number of schools and the lowest number of crime as well. We could analysis it's venues furthure thanks to Foursquare

Let's explore SOUTH LAWNDALE

Define Foursquare Credentials and Version

In [16]:

```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180604'
LIMIT = 100
```

Use geopy library to get the latitude and longitude values of SOUTH LAWNDALE

In [17]:

```
address = 'SOUTH LAWNDALE'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of SOUTH LAWNDALE are {}, {}.'.format(latit
```

```
The geograpical coordinate of SOUTH LAWNDALE are 41.8436437, -87.7125544.

/home/biosys/anaconda3/lib/python3.6/site-packages/geopy/geocoders/osm.py:1
43: UserWarning: Using Nominatim with the default "geopy/1.17.0" `user_agen
t` is strongly discouraged, as it violates Nominatim's ToS https://operatio
ns.osmfoundation.org/policies/nominatim/ (https://operations.osmfoundation.
org/policies/nominatim/) and may possibly cause 403 and 429 HTTP errors. Pl
ease specify a custom `user_agent` with `Nominatim(user_agent="my-applicati
on")` or by overriding the default `user_agent`: `geopy.geocoders.options.d
efault_user_agent = "my-application"`. In geopy 2.0 this will become an exc
eption.
  UserWarning
```

In [18]:

```
def getNearbyVenues(names, latitudes, longitudes, radius=1000):

    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={}&cli
            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 iten
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [19]:

```
south_lawndale_venues = getNearbyVenues(names=['SOUTH LAWNDALE'],
                                   latitudes=[41.8436437],
                                   longitudes=[-87.7125544]
                                  )
south_lawndale_venues
```

```
SOUTH LAWNDALE
```

Out[19]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	V Cate
0	SOUTH LAWNDALE	41.843644	-87.712554	La Catedral Cafe & Restaurant	41.846218	-87.708894	Resta
1	SOUTH LAWNDALE	41.843644	-87.712554	Restaurante Nuevo Leon	41.844396	-87.717063	Me Resta
2	SOUTH LAWNDALE	41.843644	-87.712554	Azucar	41.842580	-87.715514	De
3	SOUTH LAWNDALE	41.843644	-87.712554	Moreno's Liquors	41.844425	-87.718220	Liquor
4	SOUTH LAWNDALE	41.843644	-87.712554	El Milagro Taqueria	41.844717	-87.702247	Me Resta
5	SOUTH LAWNDALE	41.843644	-87.712554	Mi Tierra	41.845352	-87.705347	Me Resta
6	SOUTH LAWNDALE	41.843644	-87.712554	Los Candiles Restaurant	41.843627	-87.714736	Me Resta
7	SOUTH LAWNDALE	41.843644	-87.712554	La Chaparrita Taqueria	41.846354	-87.701661	Taco I
8	SOUTH LAWNDALE	41.843644	-87.712554	Taqueria Atotonilco	41.844303	-87.722848	Me Resta
9	SOUTH LAWNDALE	41.843644	-87.712554	Cermak Produce	41.845487	-87.707904	Gr
10	SOUTH LAWNDALE	41.843644	-87.712554	Little Village Arch	41.844484	-87.703793	Scul Ga
11	SOUTH LAWNDALE	41.843644	-87.712554	Dunkin' Donuts	41.840580	-87.704668	Donut
12	SOUTH LAWNDALE	41.843644	-87.712554	Birrieria Patiños Ocotlan	41.844101	-87.720338	Me Resta
13	SOUTH LAWNDALE	41.843644	-87.712554	Tortillería El Milagro	41.844726	-87.702255	Taco I
14	SOUTH LAWNDALE	41.843644	-87.712554	T-Mobile	41.844626	-87.709617	M Phone
15	SOUTH LAWNDALE	41.843644	-87.712554	El Pollo Feliz	41.842945	-87.702733	Me Resta
16	SOUTH LAWNDALE	41.843644	-87.712554	Carniceria Aguascalientes	41.844682	-87.704267	Me Resta
17	SOUTH LAWNDALE	41.843644	-87.712554	"Las Quecas" Quesadillas de México	41.846215	-87.709119	Me Resta

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	V Cate
18	SOUTH LAWNDALE	41.843644	-87.712554	AT&T	41.844120	-87.718564	M Phone
19	SOUTH LAWNDALE	41.843644	-87.712554	Taquería Los Comales	41.844459	-87.704090	Me Resta
20	SOUTH LAWNDALE	41.843644	-87.712554	Taqueria Los Barrilitos	41.846263	-87.713156	Me Resta
21	SOUTH LAWNDALE	41.843644	-87.712554	Subway	41.840620	-87.704688	Sanc I
22	SOUTH LAWNDALE	41.843644	-87.712554	Discount Mall	41.843578	-87.702629	Clo
23	SOUTH LAWNDALE	41.843644	-87.712554	ALDI	41.838257	-87.705649	Superm
24	SOUTH LAWNDALE	41.843644	-87.712554	Redbox	41.848924	-87.719919	Video
25	SOUTH LAWNDALE	41.843644	-87.712554	Walgreens	41.844353	-87.701925	Phar
26	SOUTH LAWNDALE	41.843644	-87.712554	La Chiquita	41.842973	-87.724289	M
27	SOUTH LAWNDALE	41.843644	-87.712554	Burger King	41.842357	-87.705449	Fast Resta
28	SOUTH LAWNDALE	41.843644	-87.712554	Chase Bank	41.844489	-87.721931	
29	SOUTH LAWNDALE	41.843644	-87.712554	Fifth Third Bank & ATM	41.844056	-87.724345	
30	SOUTH LAWNDALE	41.843644	-87.712554	Volkan	41.845968	-87.704967	Nigh
31	SOUTH LAWNDALE	41.843644	-87.712554	Dona Torta	41.844301	-87.708836	Me Resta
32	SOUTH LAWNDALE	41.843644	-87.712554	Las 3 Campanitas	41.844353	-87.720112	De
33	SOUTH LAWNDALE	41.843644	-87.712554	Slice Factory	41.844400	-87.711370	Pizza I
34	SOUTH LAWNDALE	41.843644	-87.712554	El Tecolote Ostioneria Y Restaurante	41.844658	-87.710972	Me Resta
35	SOUTH LAWNDALE	41.843644	-87.712554	La Chiquita Supermarket	41.844528	-87.714739	Superm
36	SOUTH LAWNDALE	41.843644	-87.712554	Jacaranda's	41.844432	-87.715203	
37	SOUTH LAWNDALE	41.843644	-87.712554	McDonald's	41.844080	-87.704772	Fast Resta
38	SOUTH LAWNDALE	41.843644	-87.712554	Tropical Optical	41.844369	-87.716134	Optical
39	SOUTH LAWNDALE	41.843644	-87.712554	La Espiga de Oro	41.846272	-87.713782	Ba
40	SOUTH LAWNDALE	41.843644	-87.712554	Little Village Plaza	41.844141	-87.704491	I

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	V Cate
41	SOUTH LAWNDALE	41.843644	-87.712554	K-OZ	41.846146	-87.704701	Nigh
42	SOUTH LAWNDALE	41.843644	-87.712554	Falcon Repair and Car Wash	41.847298	-87.705338	Resta
43	SOUTH LAWNDALE	41.843644	-87.712554	Los Olivos	41.837233	-87.709976	Me Resta
44	SOUTH LAWNDALE	41.843644	-87.712554	Wic Food Center	41.848160	-87.705400	
45	SOUTH LAWNDALE	41.843644	-87.712554	La Baguette Bakery	41.844304	-87.703188	Ba
46	SOUTH LAWNDALE	41.843644	-87.712554	Shedd Park	41.849904	-87.716968	
47	SOUTH LAWNDALE	41.843644	-87.712554	Dennis Carry Outs	41.848747	-87.719703	Fast Resta
48	SOUTH LAWNDALE	41.843644	-87.712554	Our Lady Of Tepeyac Social Center	41.847687	-87.702679	(Fi C
49	SOUTH LAWNDALE	41.843644	-87.712554	Fresh Moves - Mobile Grocery.	41.851357	-87.717933	Food
50	SOUTH LAWNDALE	41.843644	-87.712554	Homan Park	41.852166	-87.710305	
51	SOUTH LAWNDALE	41.843644	-87.712554	Kenny Jenkins	41.834750	-87.712194	La

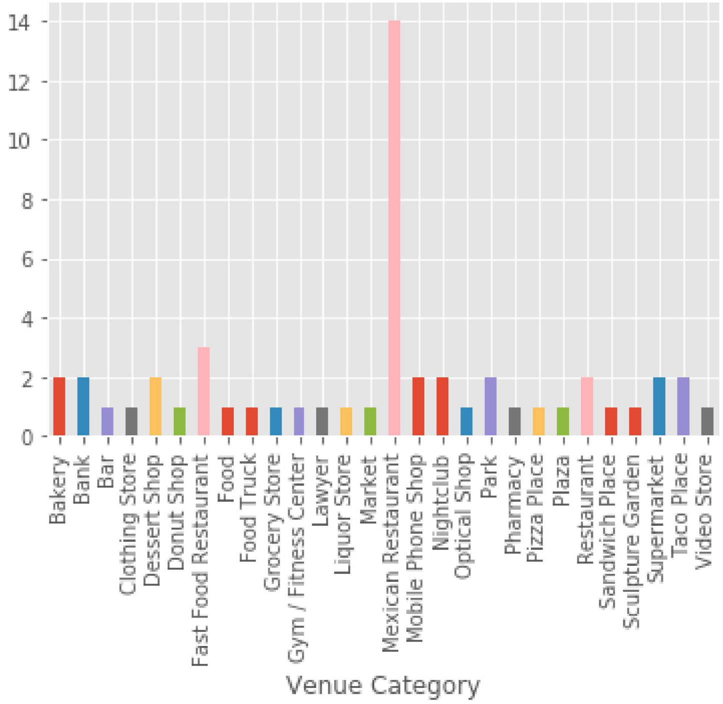
In [23]:

```
df = south_lawndale_venues.groupby(['Venue Category'])['Neighborhood'].count(

df.plot.bar()
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcca6e73c18>
```


Bakery
Bank
Bar
Clothing Store
Dessert Shop
Donut Shop
Fast Food Restaurant
Food
Food Truck
Grocery Store
Gym / Fitness Center
Lawyer
Liquor Store
Market
Mexican Restaurant
Mobile Phone Shop
Nightclub
Optical Shop
Park
Pharmacy
Pizza Place
Plaza
Restaurant
Sandwich Place
Sculpture Garden
Supermarket
Taco Place
Video Store
Venue Category