In [8]:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('future50.csv')
```

In [9]:

```
df.head()
```

Out[9]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchising
0	1	Evergreens	Seattle, Wash.	24	130.5%	26	116.7%	1150	No
1	2	Clean Juice	Charlotte, N.C.	44	121.9%	105	94.4%	560	Yes
2	3	Slapfish	Huntington Beach, Calif.	21	81.0%	21	90.9%	1370	Yes
3	4	Clean Eatz	Wilmington, N.C.	25	79.7%	46	58.6%	685	Yes
4	5	Pokeworks	Irvine, Calif.	49	77.1%	50	56.3%	1210	Yes

```
In [10]: df
```

Out[10]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franch
0	1	Evergreens	Seattle, Wash.	24	130.5%	26	116.7%	1150	
1	2	Clean Juice	Charlotte, N.C.	44	121.9%	105	94.4%	560	
2	3	Slapfish	Huntington Beach, Calif.	21	81.0%	21	90.9%	1370	
3	4	Clean Eatz	Wilmington, N.C.	25	79.7%	46	58.6%	685	
4	5	Pokeworks	Irvine, Calif.	49	77.1%	50	56.3%	1210	
5	6	Playa Bowls	Belmar, N.J.	39	62.9%	76	28.8%	580	
6	7	The Simple Greek	Blue Bell, Pa.	24	52.5%	36	33.3%	775	
7	8	Melt Shop	New York, N.Y.	20	39.6%	19	35.7%	1260	
8	9	Creamistry	Yorba Linda, Calif.	24	36.8%	60	27.7%	465	
9	10	Joella's Hot Chicken	Louisville, Ky.	29	35.5%	17	30.8%	1930	
10	11	Eggs Up Grill	Spartanburg, S.C.	30	35.4%	41	36.7%	860	
11	12	Dog Haus	Pasadena, Calif.	39	34.5%	50	42.9%	1200	
12	13	Teriyaki Madness	Denver, Colo.	41	34.1%	63	65.8%	890	
13	14	Bluestone Lane	New York, N.Y.	48	33.0%	48	37.1%	1175	
14	15	Original ChopShop	Plano, Texas	21	32.5%	12	20.0%	1930	
15	16	Rapid Fired Pizza	Kettering, Ohio	24	32.2%	35	29.6%	780	
16	17	Ike's Love & Sandwiches	San Francisco, Calif.	44	30.8%	71	29.1%	700	
17	18	Vitality Bowls	San Ramon, Calif.	37	30.1%	77	24.2%	535	
18	19	Hawkers Asian Street Fare	Orlando, Fla.	22	30.0%	7	40.0%	3800	
19	20	Maple Street Biscuit Co.	Orange Park, Fla.	39	28.9%	33	10.0%	1260	
20	21	Bulla Gastrobar	Doral, Fla.	32	28.2%	8	14.3%	4300	
21	22	Duck Donuts	Mechanicsburg, Pa.	44	28.0%	90	16.9%	530	
22	23	The Little Beet	New York, N.Y.	23	26.5%	12	33.3%	2230	
23	24	Joe & The Juice	New York, N.Y.	47	25.9%	69	25.5%	760	

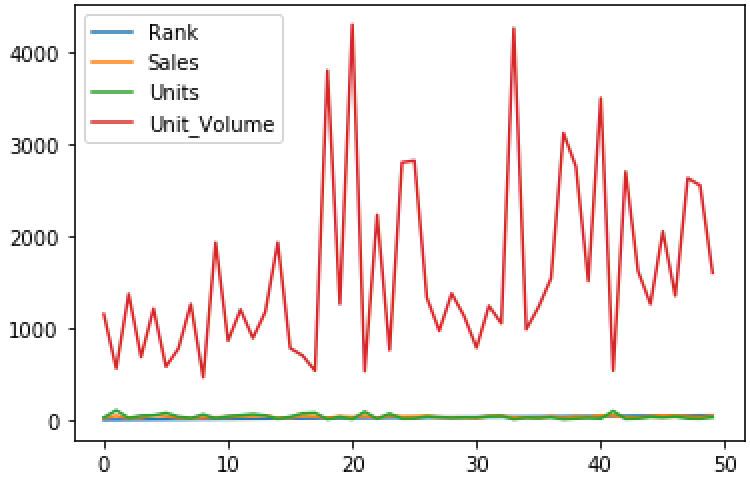
	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franch
24	25	By Chloe	New York, N.Y.	37	25.6%	14	7.7%	2800	
25	26	Sugarfire Smokehouse	Olivette, Mo.	39	25.4%	15	15.4%	2820	
26	27	Bibibop Asian Grill	Columbus, Ohio	44	24.6%	37	23.3%	1330	
27	28	Bubbakoo's Burritos	Wall Township, N.J.	28	24.3%	31	14.8%	970	
28	29	Dos Toros Taqueria	New York, N.Y.	28	24.0%	22	10.0%	1375	
29	30	The Pizza Press	Anaheim, Calif.	27	23.7%	27	22.7%	1130	
30	31	The Lost Cajun	Frisco, Texas	20	23.0%	26	4.0%	785	
31	32	Just Salad	New York, N.Y.	42	22.7%	38	26.7%	1240	
32	33	Jeni's Splendid Ice Creams	Columbus, Ohio	42	22.4%	44	22.2%	1050	
33	34	Boqueria	New York, N.Y.	27	22.0%	7	16.7%	4260	
34	35	Quickway Japanese Hibachi	Fairfax, Va.	22	21.4%	25	19.0%	985	
35	36	Surcheros	Douglas, Ga.	22	21.3%	20	17.6%	1230	
36	37	Famous Toastery	Charlotte, N.C.	40	21.2%	27	8.0%	1540	
37	38	Culinary Dropout	Scottsdale, Ariz.	20	20.8%	7	16.7%	3120	
38	39	Condado Tacos	Columbus, Ohio	38	19.9%	15	15.4%	2755	
39	40	The Flying Biscuit Cafe	Atlanta, Ga.	29	19.5%	21	16.7%	1510	
40	41	Blue Sushi Sake Grill	Omaha, Neb.	49	19.5%	14	16.7%	3500	
41	42	The Human Bean	Medford, Ore.	47	19.0%	97	19.8%	535	
42	43	Spitz Mediterranean Street Food	Los Angeles, Calif.	28	18.8%	11	10.0%	2700	
43	44	Tacos 4 Life	Conway, Ark.	25	18.4%	16	6.7%	1620	
44	45	Pita Mediterranean Street Food	Fairburn, Ga.	38	17.8%	32	10.3%	1260	
45	46	LA Crawfish	McAllen, Texas	48	17.6%	25	13.6%	2050	
46	47	&pizza	Washington, D.C.	45	17.1%	35	9.4%	1350	
47	48	Super Duper Burgers	San Francisco, Calif.	39	16.9%	16	14.3%	2630	

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Francl
48	49	StoneFire Grill	Agoura Hills, Calif.	31	16.0%	13	8.3%	2550	
49	50	Gus's World Famous Fried Chicken	Memphis, Tenn.	44	14.4%	28	7.7%	1600	

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 9 columns):
Rank           50 non-null int64
Restaurant     50 non-null object
Location       50 non-null object
Sales          50 non-null int64
YOY_Sales      50 non-null object
Units          50 non-null int64
YOY_Units      50 non-null object
Unit_Volume    50 non-null int64
Franchising    50 non-null object
dtypes: int64(4), object(5)
memory usage: 3.6+ KB
```

In [11]:
```
df.plot()
plt.show()
```



In [15]:

```
df.plot(kind='scatter',x='Restaurant',y='Sales')
plt.show()
```

```
ERROR:root:Internal Python error in the inspect module.
Below is the traceback from this internal error.


Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py", line 2910, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "<ipython-input-15-3203944125c6>", line 1, in <module>
    df.plot(kind='scatter',x='Restaurant',y='Sales')
  File "C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py", l
ine 2677, in __call__
    sort_columns=sort_columns, **kwds)
  File "C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py", l
ine 1902, in plot_frame
    **kwds)
  File "C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py", l
ine 1687, in _plot
    kind=kind, **kwds)
  File "C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py", l
ine 837, in __init__
    super(ScatterPlot, self).__init__(data, x, y, s=s, **kwargs)
  File "C:\ProgramData\Anaconda3\lib\site-packages\pandas\plotting\_core.py", l
ine 812, in __init__
    raise ValueError(self._kind + ' requires x column to be numeric')
ValueError: scatter requires x column to be numeric

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py", line 1828, in showtraceback
    stb = value._render_traceback_()
AttributeError: 'ValueError' object has no attribute '_render_traceback_'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\ultratb.py", li
ne 1090, in get_records
    return _fixed_getinnerframes(etb, number_of_lines_of_context, tb_offset)
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\ultratb.py", li
ne 311, in wrapped
    return f(*args, **kwargs)
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\ultratb.py", li
ne 345, in _fixed_getinnerframes
    records = fix_frame_records_filenames(inspect.getinnerframes(etb, context))
  File "C:\ProgramData\Anaconda3\lib\inspect.py", line 1483, in getinnerframes
    frameinfo = (tb.tb_frame,) + getframeinfo(tb, context)
  File "C:\ProgramData\Anaconda3\lib\inspect.py", line 1441, in getframeinfo
    filename = getsourcefile(frame) or getfile(frame)
  File "C:\ProgramData\Anaconda3\lib\inspect.py", line 696, in getsourcefile
    if getattr(getmodule(object, filename), '__loader__', None) is not None:
  File "C:\ProgramData\Anaconda3\lib\inspect.py", line 733, in getmodule
```

```
    if ismodule(module) and hasattr(module, '__file__'):
  File "C:\ProgramData\Anaconda3\lib\site-packages\py\_vendored_packages\apipk
g.py", line 195, in __getattribute__
    return getattr(getmod(), name)
  File "C:\ProgramData\Anaconda3\lib\site-packages\py\_vendored_packages\apipk
g.py", line 179, in getmod
    x = importobj(modpath, None)
  File "C:\ProgramData\Anaconda3\lib\site-packages\py\_vendored_packages\apipk
g.py", line 69, in importobj
    module = __import__(modpath, None, None, ['__doc__'])
  File "C:\ProgramData\Anaconda3\lib\site-packages\pytest.py", line 13, in <mod
ule>
    from _pytest.fixtures import fixture, yield_fixture
  File "C:\ProgramData\Anaconda3\lib\site-packages\_pytest\fixtures.py", line 8
42, in <module>
    class FixtureFunctionMarker(object):
  File "C:\ProgramData\Anaconda3\lib\site-packages\_pytest\fixtures.py", line 8
44, in FixtureFunctionMarker
    params = attr.ib(convert=attr.converters.optional(tuple))
TypeError: attrib() got an unexpected keyword argument 'convert'


---------------------------------------------------------------------------
```

In [16]:

```
jsn = pd.read_json('dat.json')
```

In [21]:
```
print(jsn)
```

```
     petalLength  petalWidth  sepalLength  sepalWidth    species
0            1.4         0.2          5.1         3.5     setosa
1            1.4         0.2          4.9         3.0     setosa
2            1.3         0.2          4.7         3.2     setosa
3            1.5         0.2          4.6         3.1     setosa
4            1.4         0.2          5.0         3.6     setosa
5            1.7         0.4          5.4         3.9     setosa
6            1.4         0.3          4.6         3.4     setosa
7            1.5         0.2          5.0         3.4     setosa
8            1.4         0.2          4.4         2.9     setosa
9            1.5         0.1          4.9         3.1     setosa
10           1.5         0.2          5.4         3.7     setosa
11           1.6         0.2          4.8         3.4     setosa
12           1.4         0.1          4.8         3.0     setosa
13           1.1         0.1          4.3         3.0     setosa
14           1.2         0.2          5.8         4.0     setosa
15           1.5         0.4          5.7         4.4     setosa
16           1.3         0.4          5.4         3.9     setosa
17           1.4         0.3          5.1         3.5     setosa
18           1.7         0.3          5.7         3.8     setosa
19           1.5         0.3          5.1         3.8     setosa
20           1.7         0.2          5.4         3.4     setosa
21           1.5         0.4          5.1         3.7     setosa
22           1.0         0.2          4.6         3.6     setosa
23           1.7         0.5          5.1         3.3     setosa
24           1.9         0.2          4.8         3.4     setosa
25           1.6         0.2          5.0         3.0     setosa
26           1.6         0.4          5.0         3.4     setosa
27           1.5         0.2          5.2         3.5     setosa
28           1.4         0.2          5.2         3.4     setosa
29           1.6         0.2          4.7         3.2     setosa
..           ...         ...          ...         ...        ...
120          5.7         2.3          6.9         3.2  virginica
121          4.9         2.0          5.6         2.8  virginica
122          6.7         2.0          7.7         2.8  virginica
123          4.9         1.8          6.3         2.7  virginica
124          5.7         2.1          6.7         3.3  virginica
125          6.0         1.8          7.2         3.2  virginica
126          4.8         1.8          6.2         2.8  virginica
127          4.9         1.8          6.1         3.0  virginica
128          5.6         2.1          6.4         2.8  virginica
129          5.8         1.6          7.2         3.0  virginica
130          6.1         1.9          7.4         2.8  virginica
131          6.4         2.0          7.9         3.8  virginica
132          5.6         2.2          6.4         2.8  virginica
133          5.1         1.5          6.3         2.8  virginica
134          5.6         1.4          6.1         2.6  virginica
135          6.1         2.3          7.7         3.0  virginica
136          5.6         2.4          6.3         3.4  virginica
137          5.5         1.8          6.4         3.1  virginica
138          4.8         1.8          6.0         3.0  virginica
139          5.4         2.1          6.9         3.1  virginica
140          5.6         2.4          6.7         3.1  virginica
141          5.1         2.3          6.9         3.1  virginica
```

```
142            5.1          1.9           5.8          2.7  virginica
143            5.9          2.3           6.8          3.2  virginica
144            5.7          2.5           6.7          3.3  virginica
145            5.2          2.3           6.7          3.0  virginica
146            5.0          1.9           6.3          2.5  virginica
147            5.2          2.0           6.5          3.0  virginica
148            5.4          2.3           6.2          3.4  virginica
149            5.1          1.8           5.9          3.0  virginica

[150 rows x 5 columns]
```

In [22]:

```
jsn
```

Out[22]:

	petalLength	petalWidth	sepalLength	sepalWidth	species
0	1.4	0.2	5.1	3.5	setosa
1	1.4	0.2	4.9	3.0	setosa
2	1.3	0.2	4.7	3.2	setosa
3	1.5	0.2	4.6	3.1	setosa
4	1.4	0.2	5.0	3.6	setosa
5	1.7	0.4	5.4	3.9	setosa
6	1.4	0.3	4.6	3.4	setosa
7	1.5	0.2	5.0	3.4	setosa
8	1.4	0.2	4.4	2.9	setosa
9	1.5	0.1	4.9	3.1	setosa
10	1.5	0.2	5.4	3.7	setosa
11	1.6	0.2	4.8	3.4	setosa
12	1.4	0.1	4.8	3.0	setosa
13	1.1	0.1	4.3	3.0	setosa
14	1.2	0.2	5.8	4.0	setosa
15	1.5	0.4	5.7	4.4	setosa
16	1.3	0.4	5.4	3.9	setosa
17	1.4	0.3	5.1	3.5	setosa
18	1.7	0.3	5.7	3.8	setosa
19	1.5	0.3	5.1	3.8	setosa
20	1.7	0.2	5.4	3.4	setosa
21	1.5	0.4	5.1	3.7	setosa
22	1.0	0.2	4.6	3.6	setosa
23	1.7	0.5	5.1	3.3	setosa
24	1.9	0.2	4.8	3.4	setosa
25	1.6	0.2	5.0	3.0	setosa
26	1.6	0.4	5.0	3.4	setosa
27	1.5	0.2	5.2	3.5	setosa
28	1.4	0.2	5.2	3.4	setosa
29	1.6	0.2	4.7	3.2	setosa
...	...	...	...	...	...
120	5.7	2.3	6.9	3.2	virginica
121	4.9	2.0	5.6	2.8	virginica
122	6.7	2.0	7.7	2.8	virginica

	petalLength	petalWidth	sepalLength	sepalWidth	species
123	4.9	1.8	6.3	2.7	virginica
124	5.7	2.1	6.7	3.3	virginica
125	6.0	1.8	7.2	3.2	virginica
126	4.8	1.8	6.2	2.8	virginica
127	4.9	1.8	6.1	3.0	virginica
128	5.6	2.1	6.4	2.8	virginica
129	5.8	1.6	7.2	3.0	virginica
130	6.1	1.9	7.4	2.8	virginica
131	6.4	2.0	7.9	3.8	virginica
132	5.6	2.2	6.4	2.8	virginica
133	5.1	1.5	6.3	2.8	virginica
134	5.6	1.4	6.1	2.6	virginica
135	6.1	2.3	7.7	3.0	virginica
136	5.6	2.4	6.3	3.4	virginica
137	5.5	1.8	6.4	3.1	virginica
138	4.8	1.8	6.0	3.0	virginica
139	5.4	2.1	6.9	3.1	virginica
140	5.6	2.4	6.7	3.1	virginica
141	5.1	2.3	6.9	3.1	virginica
142	5.1	1.9	5.8	2.7	virginica
143	5.9	2.3	6.8	3.2	virginica
144	5.7	2.5	6.7	3.3	virginica
145	5.2	2.3	6.7	3.0	virginica
146	5.0	1.9	6.3	2.5	virginica
147	5.2	2.0	6.5	3.0	virginica
148	5.4	2.3	6.2	3.4	virginica
149	5.1	1.8	5.9	3.0	virginica

150 rows × 5 columns

```
In [23]: jsn.size
```

```
Out[23]: 750
```

```
In [24]: jsn.shape
```

```
Out[24]: (150, 5)
```

```
In [26]: df.size
```

```
Out[26]: 450
```

```
In [27]: jsn.ndim
```

```
Out[27]: 2
```

```
In [28]: df.shape
```

```
Out[28]: (50, 9)
```

```
In [42]: import matplotlib.image as mpimg
         img = mpimg.imread(r'S:\pml lab\image_dataset\aaa.jpg')
         plt.imshow(img)
```

```
Out[42]: <matplotlib.image.AxesImage at 0x2b2e9c0fbe0>
```

```
In [40]:
```

```
In [ ]:
```

```
In [ ]:
```