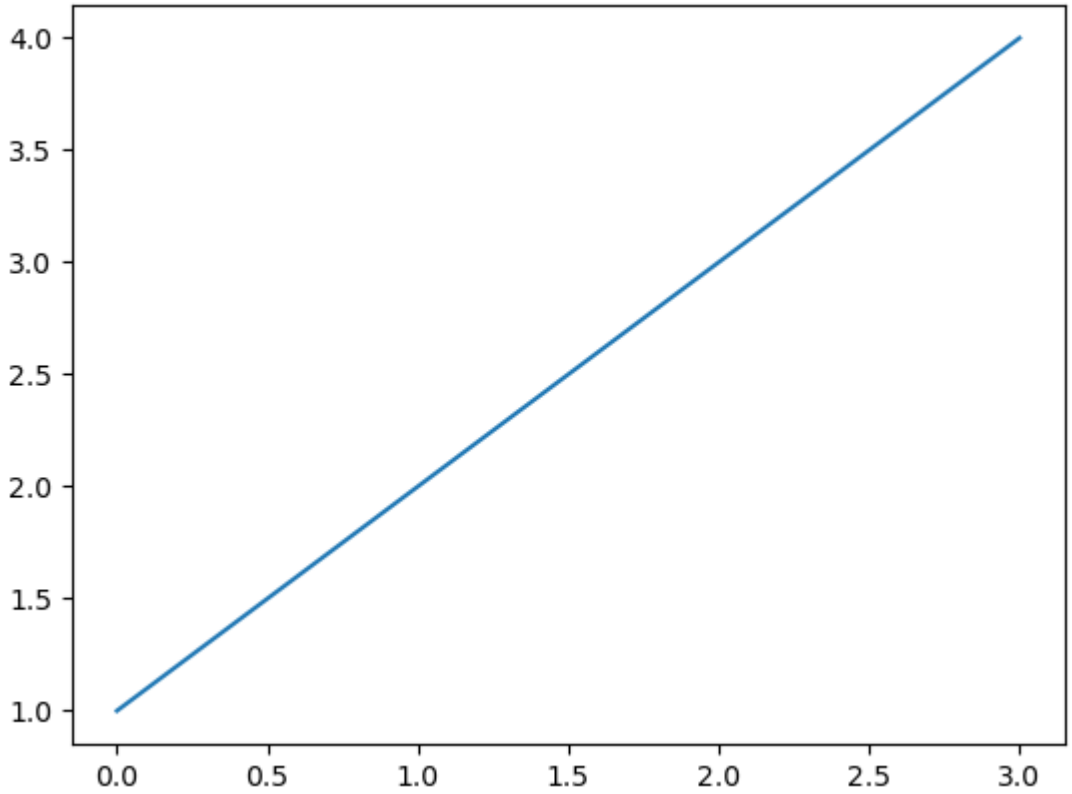
Matplotlib.pyplot를 활용한 데이터 분석

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html
(https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html)

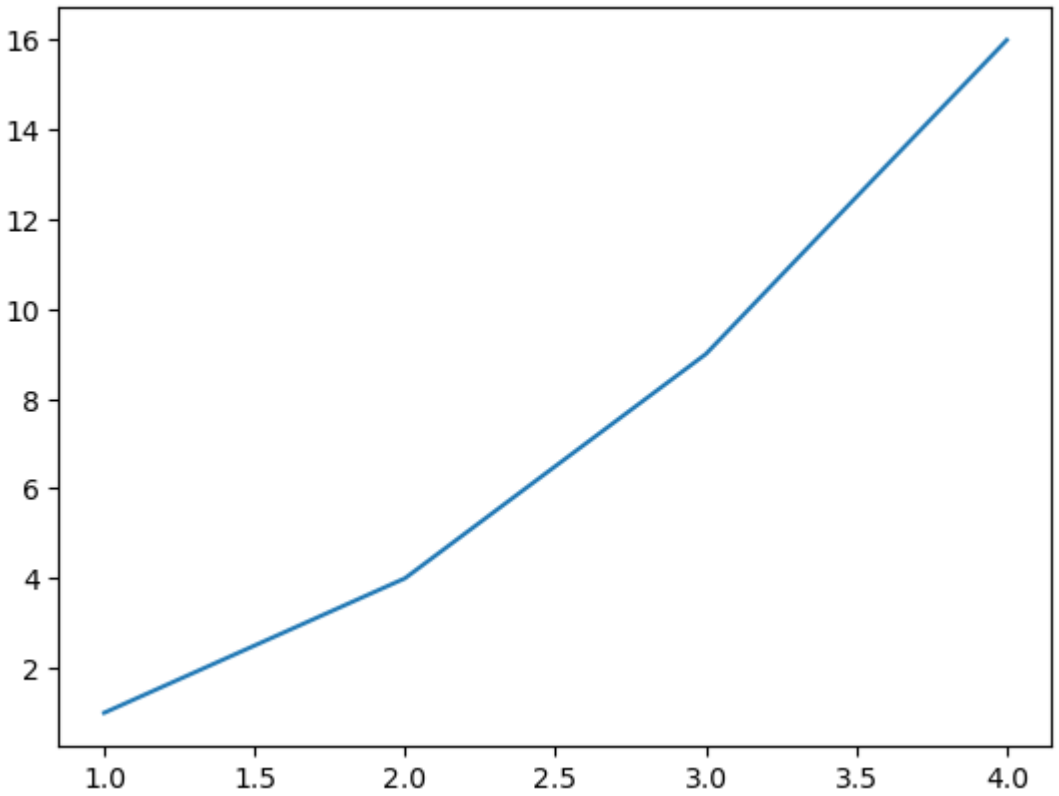
In [1]:

```
1 import matplotlib.pyplot as plt
2
3 #plot() 함수는 리스트의 값들이 y 값들이라고 가정하고, x 값 [0, 1, 2, 3]을 자동으로 생성
4 plt.plot([1, 2, 3, 4])
5 plt.show()
```

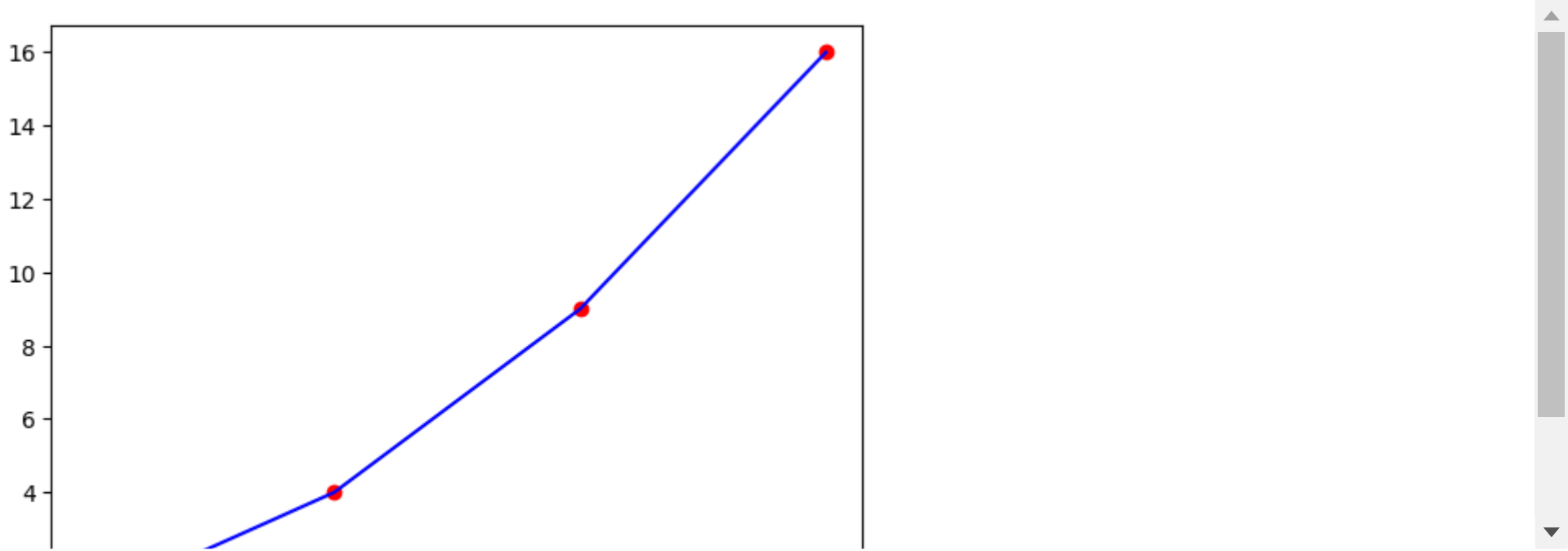


In [2]:

```
1 # x축과 y축을 기반으로 그래프
2 plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
3 plt.show()
```



In [3]:
```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro') #red 색상으로'o' 원형
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'b-') #blue 색상으로'-' 원형
#plt.axis([0, 10, 0, 20])
plt.show()
```



In [4]:
```
import numpy as np

# 0에서 5까지 0.2 간격으로 균일하게 샘플된 값
t = np.arange(0., 5., 0.2)
```

In [5]:
```
t
```

Out[5]:
```
array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4,
       2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. , 4.2, 4.4, 4.6, 4.8])
```

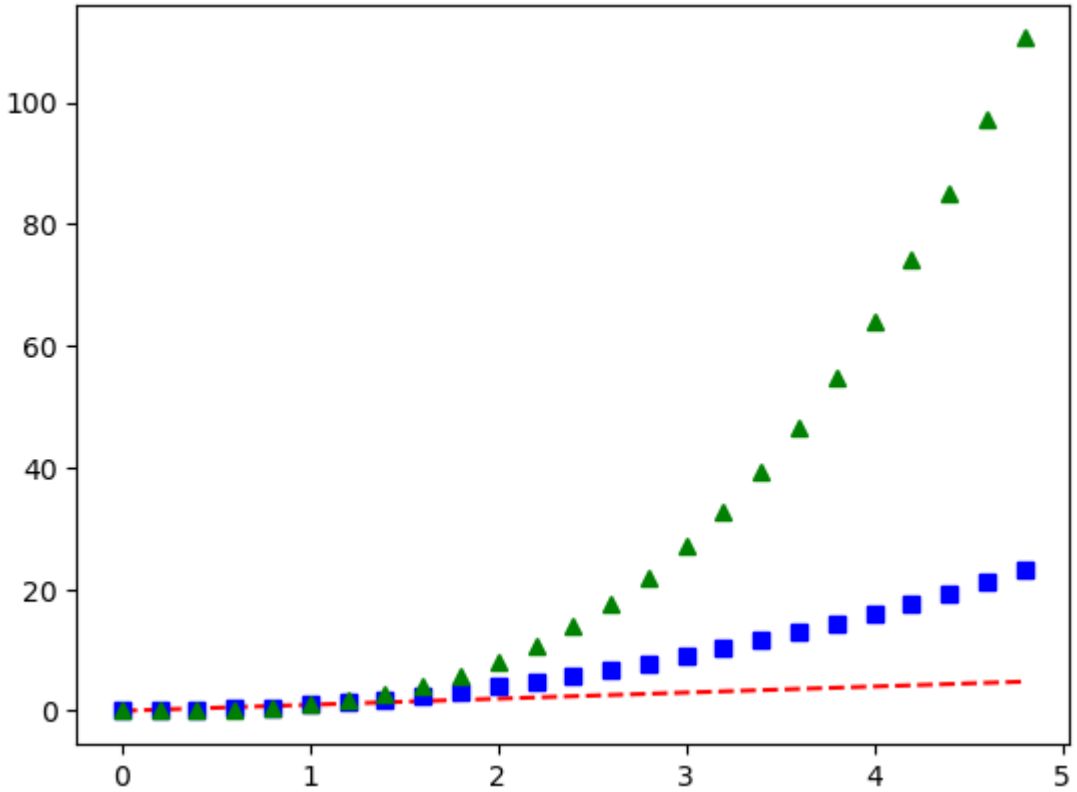
In [6]:
```
t**2
```

Out[6]:
```
array([ 0.  ,  0.04,  0.16,  0.36,  0.64,  1.  ,  1.44,  1.96,  2.56,
        3.24,  4.  ,  4.84,  5.76,  6.76,  7.84,  9.  , 10.24, 11.56,
       12.96, 14.44, 16.  , 17.64, 19.36, 21.16, 23.04])
```

In [7]:
```
type(t)
```

Out[7]:
```
numpy.ndarray
```

In [8]:
```
# red 대쉬, blue 사각형, green 삼각형
# plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')

plt.plot(t, t, 'r--')  # red 대쉬
plt.plot(t, t**2, 'bs')  #blue 사각형
plt.plot(t, t**3, 'g^')  #green 삼각형
plt.show()
```



In [9]:
```
# 딕셔너리
data = {'apple': 10, 'orange': 15, 'lemon': 5, 'lime': 20}
```

In [10]:
```
data.keys()
```

Out[10]:
```
dict_keys(['apple', 'orange', 'lemon', 'lime'])
```

In [11]:
```
1 data.values()
```

Out[11]:
```
dict_values([10, 15, 5, 20])
```

In [12]:
```
1 type(data.keys())
```

Out[12]:
```
dict_keys
```

In [13]:
```
1 names = list(data.keys())  #리스트로 변환
2 values = list(data.values())  #리스트로 변환
```

In [14]:
```
1 names
```

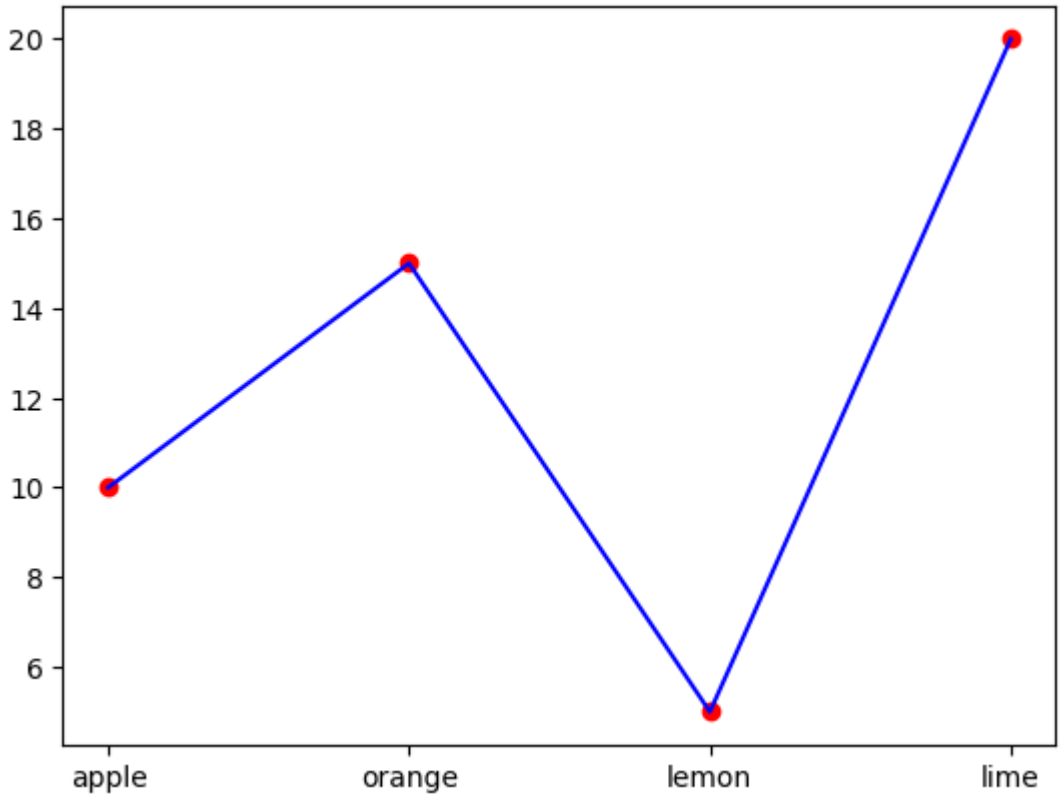
Out[14]:
```
['apple', 'orange', 'lemon', 'lime']
```

In [15]:
```
1 values
```

Out[15]:
```
[10, 15, 5, 20]
```

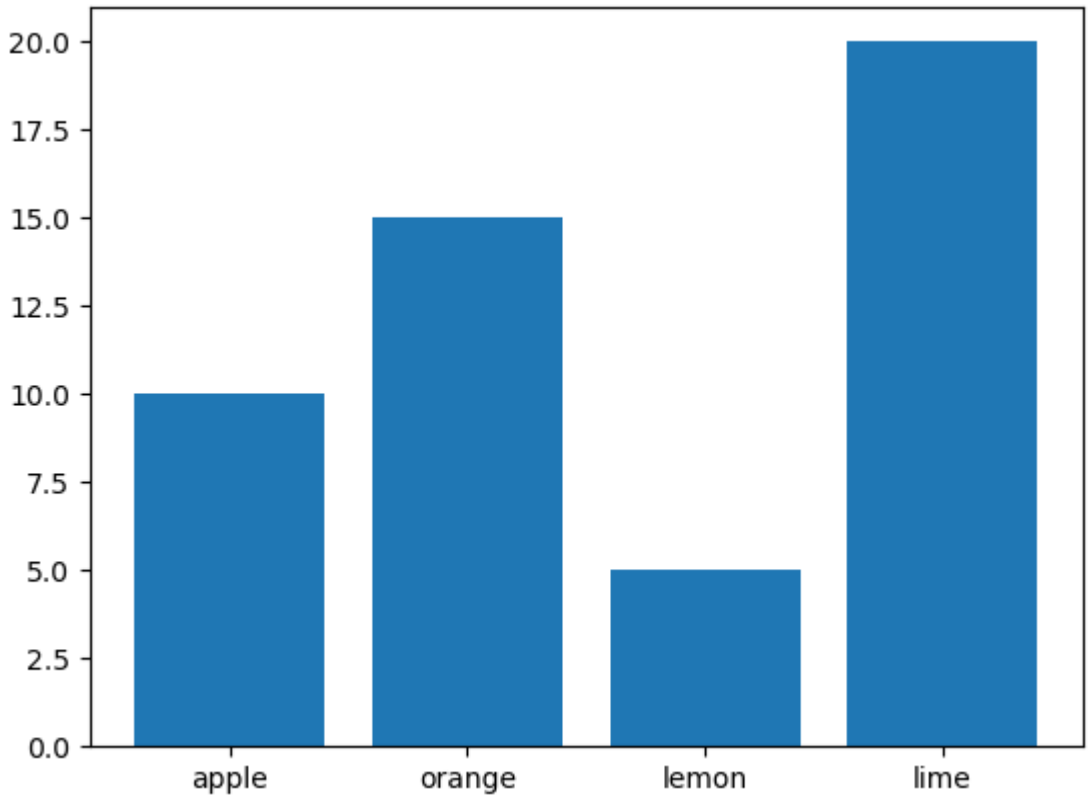
In [16]:

```
#plt.plot 그래프 그리기

plt.plot(names, values, 'ro')
plt.plot(names, values, 'b-')
plt.show()
```



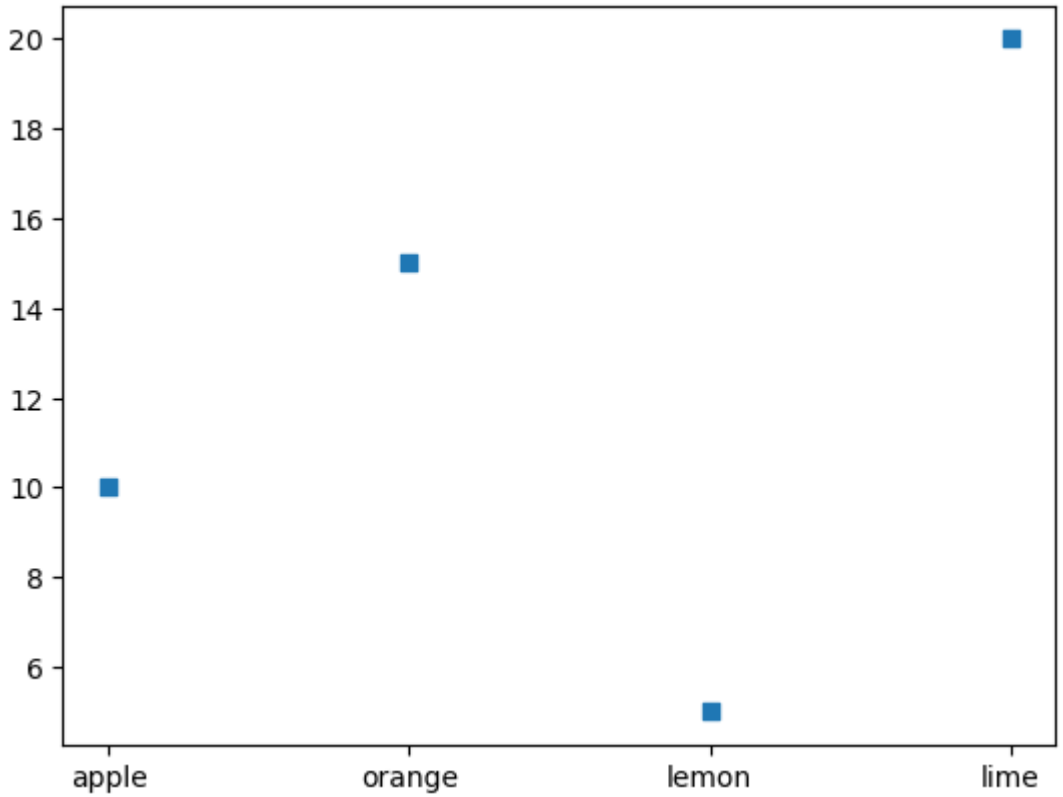
In [17]:

```
#plt.bar 그래프 그리기

plt.bar(names, values)
plt.show()
```



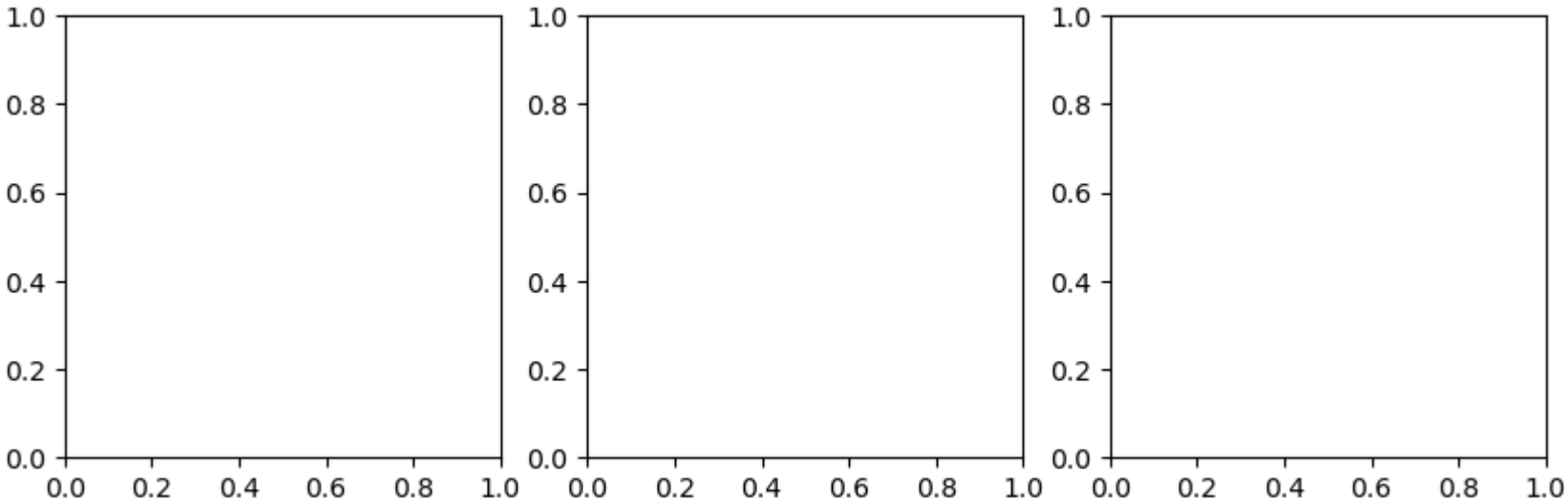
In [18]:

```
#plt.scatter 그래프 그리기

#plt.scatter(names, values)
plt.scatter(names, values, marker='s')
plt.show()
```

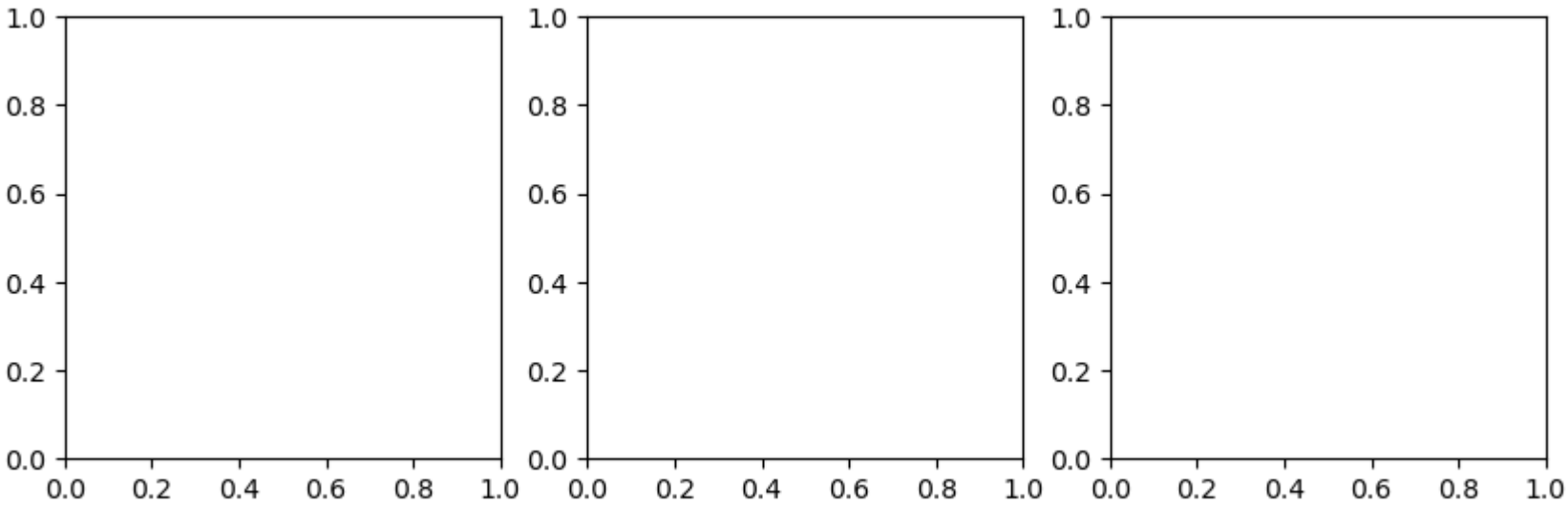


In [19]:

```
plt.figure(figsize=(10,3))  #figure() 호출할 때 마다 그래프를 그릴 수 있는 사각형 프레임이 생성


#figure 메소드로 figure을 작성하고, subplot 메소드로 axes를 생성

plt.subplot(1,3,1)  #figure()로 인해 생성된 사각형 프레임 내에서 1행 3열 중에서 첫번째 프레임을 의미
plt.subplot(1,3,2)  #figure()로 인해 생성된 사각형 프레임 내에서 1행 3열 중에서 두번째 프레임을 의미
plt.subplot(1,3,3)  #figure()로 인해 생성된 사각형 프레임 내에서 1행 3열 중에서 세번째 프레임을 의미
plt.show()
```

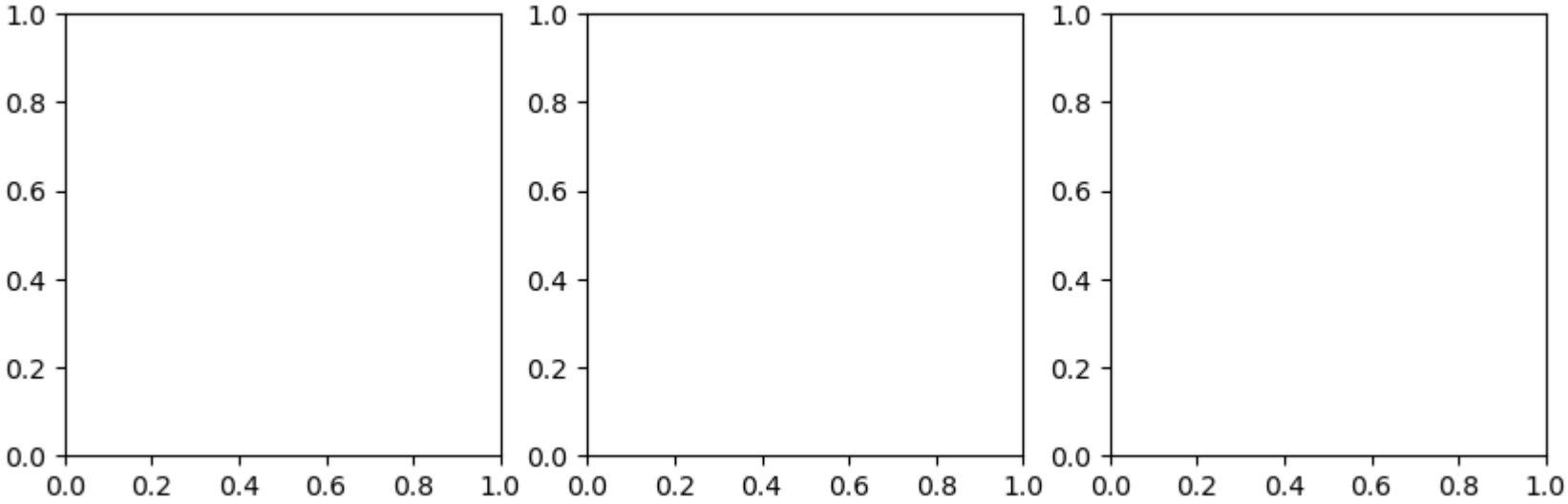


In [20]:
```
#위의 코드를 1줄로 작성

fig, axs = plt.subplots(1, 3, figsize=(10, 3), sharey=False)
```



In [21]:
```
fig #fig는 프레임을 생성
```

Out[21]:

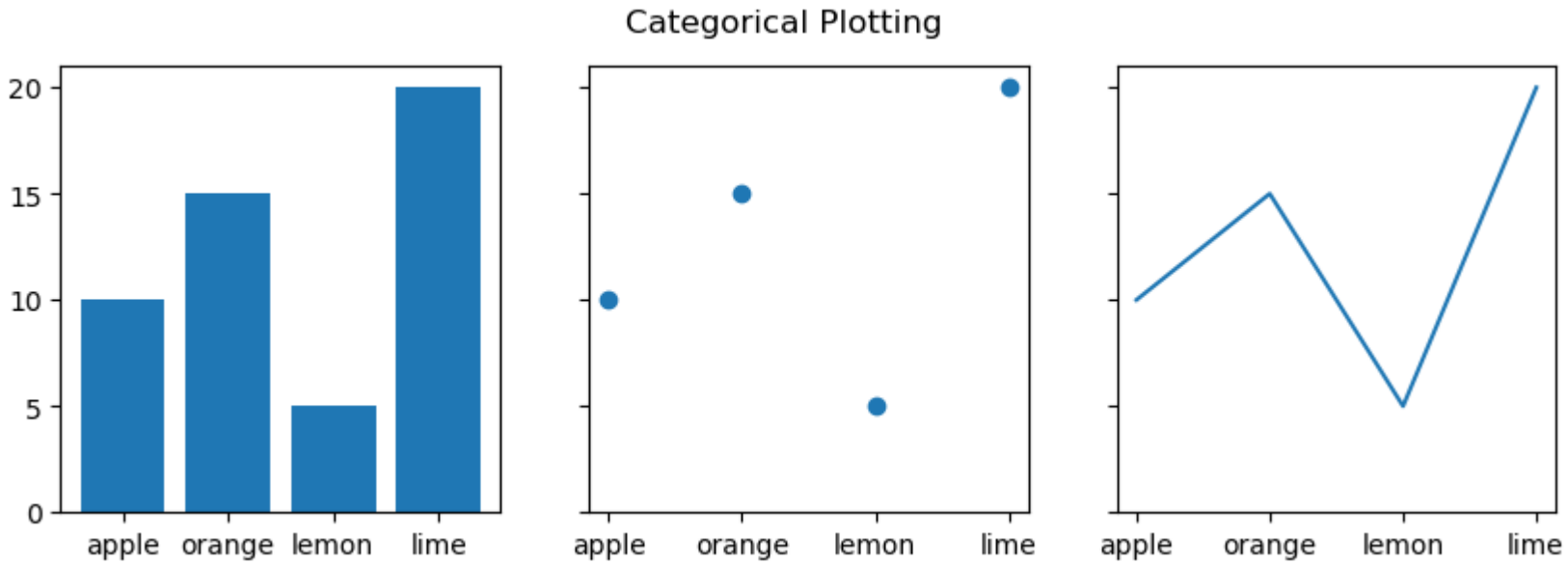


In [22]:
```
axs  #axs는 3개의 축을 생성
```

Out[22]:
```
array([<Axes: >, <Axes: >, <Axes: >], dtype=object)
```

In [23]:

```
fig, axs = plt.subplots(1, 3, figsize=(10, 3), sharey=True)
#fig, axs = plt.subplots(1, 3, figsize=(10, 3), sharey=False)

fig.suptitle('Categorical Plotting')

axs[0].bar(names, values)      #1번째 축
axs[1].scatter(names, values)  #2번째 축
axs[2].plot(names, values)     #3번째 축

plt.show()
```



시각화를 위해 numpy를 이용하여 데이터셋 생성

In [24]:

```
vegetables = ["cucumber", "tomato", "lettuce", "asparagus", "potato", "wheat", "barley"]
#vegetables = [ "오이", "토마토", "상추", "아스파라거스", "감자", "밀", "보리"]

print(vegetables)
print(type(vegetables))
print(len(vegetables),"종류의 채소 데이터")
```

```
['cucumber', 'tomato', 'lettuce', 'asparagus', 'potato', 'wheat', 'barley']
<class 'list'>
7 종류의 채소 데이터
```

In [25]:
```
farmers = ["Farmer Joe", "Upland Bros", "Smith Gardening",
           "Agrifun", "Organiculture", "BioGoods Ltd", "Cornylee Corp."]

print(farmers)
print(type(farmers))
print(len(farmers),"명의 농부 데이터")
```

```
['Farmer Joe', 'Upland Bros', 'Smith Gardening', 'Agrifun', 'Organiculture', 'BioGoods Ltd', 'Cornylee Corp.']
<class 'list'>
7 명의 농부 데이터
```

In [26]:
```
harvest = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                    [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                    [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                    [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                    [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                    [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                    [0.1, 2.0, 0.0, 1.4, 0.0, 1.9, 6.3]])

print(harvest)
print(type(harvest))
print(len(harvest),"라인", len(harvest[0]),"컬럼의 데이터")
```

```
[[0.8 2.4 2.5 3.9 0.  4.  0. ]
 [2.4 0.  4.  1.  2.7 0.  0. ]
 [1.1 2.4 0.8 4.3 1.9 4.4 0. ]
 [0.6 0.  0.3 0.  3.1 0.  0. ]
 [0.7 1.7 0.6 2.6 2.2 6.2 0. ]
 [1.3 1.2 0.  0.  0.  3.2 5.1]
 [0.1 2.  0.  1.4 0.  1.9 6.3]]
<class 'numpy.ndarray'>
7 라인 7 컬럼의 데이터
```

시각화

In [27]:
```
np.arange(7)
```

Out[27]:
```
array([0, 1, 2, 3, 4, 5, 6])
```

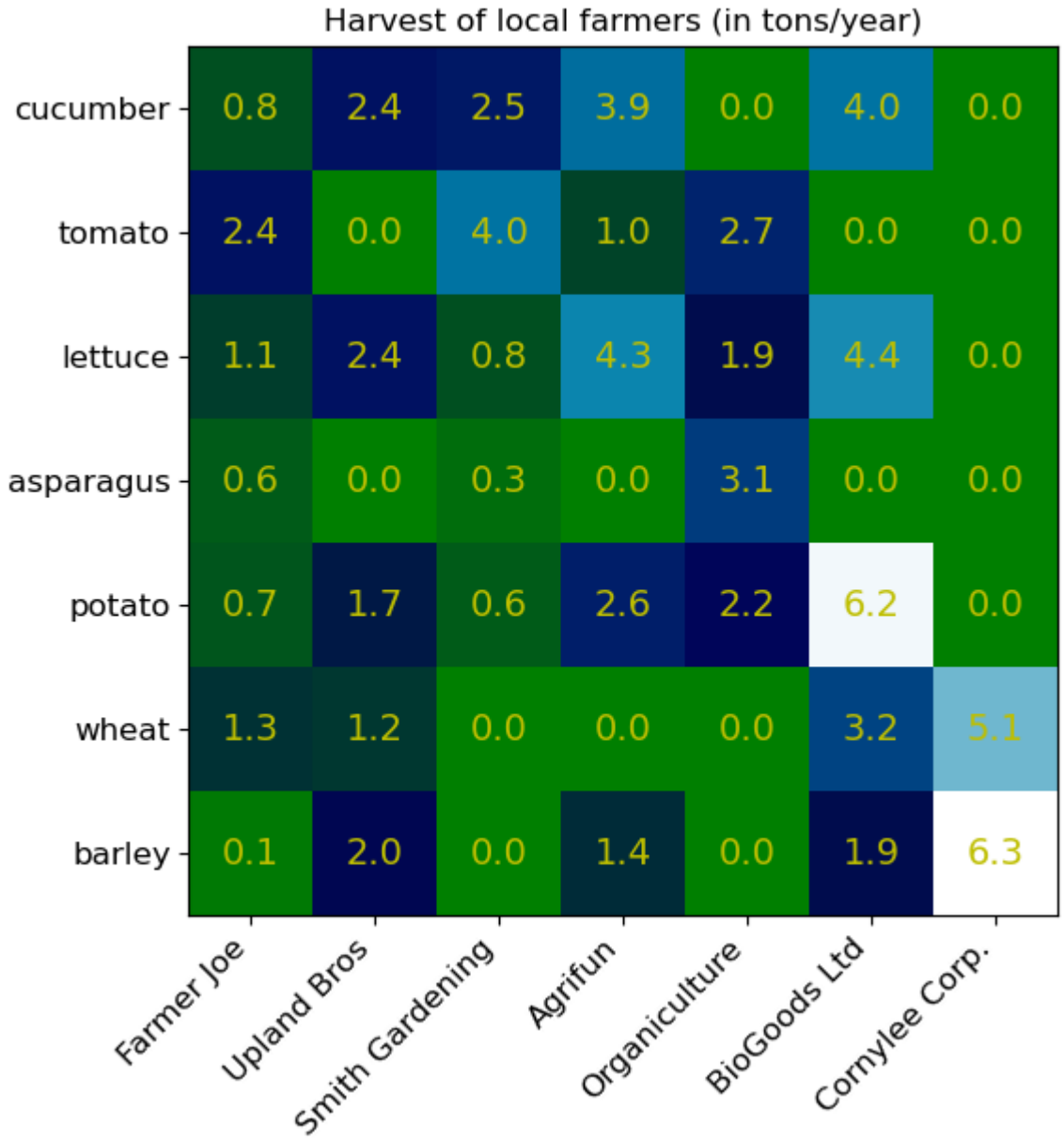
In [28]:

```
#setp 1 ----------------------------------------------------------------
fig, ax = plt.subplots(figsize=(10, 6))
print(fig)  #프레임 생성
print(ax)   #1개의 축 생성

#setp 2 ----------------------------------------------------------------
ax.set_title("Harvest of local farmers (in tons/year)")
ax.imshow(harvest, cmap='ocean')  #numpy.ndarray 타입 사용 가능
#ax.imshow(harvest)  #numpy.ndarray 타입 사용 가능

#setp 3 ----------------------------------------------------------------

#xaxis, yaxis의 눈금 위치를 설정
print('● len(farmers)은', len(farmers), '이고, 값은  ', end='')
print(np.arange(len(farmers)))

print('● len(vegetables)은', len(vegetables),  '이고, 값은  ',end='')
print(np.arange(len(vegetables)))

ax.set_xticks(np.arange(len(farmers)))      #x축 [0 1 2 3 4 5 6]
ax.set_yticks(np.arange(len(vegetables)))   #y축 [0 1 2 3 4 5 6]

#setp 4 ----------------------------------------------------------------

#각각의 x, y축에 레이블링
ax.set_xticklabels(farmers)     #x축 [0 1 2 3 4 5 6]부분에 레이블링이 됨
ax.set_yticklabels(vegetables)  #y축 [0 1 2 3 4 5 6]부분에 레이블링이 됨

#setp 5 ----------------------------------------------------------------

#x축 레이블을 회전하고 정렬
#ha = 'right'는 수평 레이블 텍스트의 오른쪽 끝을 눈금에 맞춥니다.
#ha = 'left'는 수평 레이블 텍스트의 왼쪽 끝을 눈금에 맞춥니다.
#ha = 'center'는 수평 레이블 텍스트의 중심을 눈금에 맞춥니다.

plt.setp(ax.get_xticklabels(), rotation=45, ha="right", fontsize=12)
#plt.setp(ax.get_xticklabels(), rotation=45, ha="left", fontsize=12)
#plt.setp(ax.get_xticklabels(), rotation=45, ha="center", fontsize=12)

#y축 레이블을 정렬
plt.setp(ax.get_yticklabels(), ha="right", fontsize=12)

#setp 6 ----------------------------------------------------------------
#ha 수평 축
#va 수직 축

```

```
for i in range(len(vegetables)):  #column
    for j in range(len(farmers)):   #row
        text = ax.text(j, i, harvest[i, j], ha="center", va="center", color="y", fontsize=14)
        print(text, end='')
        print('-->', harvest[i, j])
    print('-'*50)

#---------------------------------------------------------------------------
plt.show()
```

```
Figure(1000x600)
Axes(0.125,0.11;0.775x0.77)
● len(farmers)은 7 이고, 값은  [0 1 2 3 4 5 6]
● len(vegetables)은 7 이고, 값은  [0 1 2 3 4 5 6]
Text(0, 0, '0.8')--> 0.8
Text(1, 0, '2.4')--> 2.4
Text(2, 0, '2.5')--> 2.5
Text(3, 0, '3.9')--> 3.9
Text(4, 0, '0.0')--> 0.0
Text(5, 0, '4.0')--> 4.0
Text(6, 0, '0.0')--> 0.0
--------------------------------------------------
Text(0, 1, '2.4')--> 2.4
Text(1, 1, '0.0')--> 0.0
Text(2, 1, '4.0')--> 4.0
Text(3, 1, '1.0')--> 1.0
Text(4, 1, '2.7')--> 2.7
Text(5, 1, '0.0')--> 0.0
Text(6, 1, '0.0')--> 0.0
--------------------------------------------------
Text(0, 2, '1.1')--> 1.1
Text(1, 2, '2.4')--> 2.4
Text(2, 2, '0.8')--> 0.8
Text(3, 2, '4.3')--> 4.3
Text(4, 2, '1.9')--> 1.9
Text(5, 2, '4.4')--> 4.4
Text(6, 2, '0.0')--> 0.0
--------------------------------------------------
Text(0, 3, '0.6')--> 0.6
Text(1, 3, '0.0')--> 0.0
Text(2, 3, '0.3')--> 0.3
Text(3, 3, '0.0')--> 0.0
Text(4, 3, '3.1')--> 3.1
Text(5, 3, '0.0')--> 0.0
Text(6, 3, '0.0')--> 0.0
--------------------------------------------------
Text(0, 4, '0.7')--> 0.7
Text(1, 4, '1.7')--> 1.7
Text(2, 4, '0.6')--> 0.6
Text(3, 4, '2.6')--> 2.6
Text(4, 4, '2.2')--> 2.2
Text(5, 4, '6.2')--> 6.2
Text(6, 4, '0.0')--> 0.0
--------------------------------------------------
Text(0, 5, '1.3')--> 1.3
Text(1, 5, '1.2')--> 1.2
```

```
Text(2, 5, '0.0')--> 0.0
Text(3, 5, '0.0')--> 0.0
Text(4, 5, '0.0')--> 0.0
Text(5, 5, '3.2')--> 3.2
Text(6, 5, '5.1')--> 5.1
------------------------------------------------
Text(0, 6, '0.1')--> 0.1
Text(1, 6, '2.0')--> 2.0
Text(2, 6, '0.0')--> 0.0
Text(3, 6, '1.4')--> 1.4
Text(4, 6, '0.0')--> 0.0
Text(5, 6, '1.9')--> 1.9
Text(6, 6, '6.3')--> 6.3
------------------------------------------------
```


Harvest of local farmers (in tons/year)
cucumber
tomato
lettuce
asparagus
potato
wheat
barley
0.8 2.4 2.5 3.9 0.0 4.0 0.0
2.4 0.0 4.0 1.0 2.7 0.0 0.0
1.1 2.4 0.8 4.3 1.9 4.4 0.0
0.6 0.0 0.3 0.0 3.1 0.0 0.0
0.7 1.7 0.6 2.6 2.2 6.2 0.0
1.3 1.2 0.0 0.0 0.0 3.2 5.1
0.1 2.0 0.0 1.4 0.0 1.9 6.3
Farmer Joe
Upland Bros
Smith Gardening
Agrifun
Organiculture
BioGoods Ltd
Cornylee Corp.

참고

In [29]:

```
#setp 1 ----------------------------------------------------------------
fig, ax = plt.subplots(figsize=(10, 6))
print(fig)  #프레임 생성
print(ax)   #1개의 축 생성

#setp 2 ----------------------------------------------------------------
ax.set_title("Harvest of local farmers (in tons/year)")
#ax.imshow(harvest, cmap='ocean')  #numpy.ndarray 타입 사용 가능
ax.imshow(harvest)  #numpy.ndarray 타입 사용 가능

#setp 3 ----------------------------------------------------------------

#xaxis, yaxis의 눈금 위치를 설정
print('● len(farmers)은', len(farmers), '이고, 값은  ', end='')
print(np.arange(len(farmers)))

print('● len(vegetables)은', len(vegetables),  '이고, 값은  ',end='')
print(np.arange(len(vegetables)))

ax.set_xticks(np.arange(len(farmers)+5))      #x축 [0 1 2 3 4 5 6]
ax.set_yticks(np.arange(len(vegetables)+3))   #y축 [0 1 2 3 4 5 6]

plt.show()
```

```
Figure(1000x600)
Axes(0.125,0.11;0.775x0.77)
● len(farmers)은 7 이고, 값은  [0 1 2 3 4 5 6]
● len(vegetables)은 7 이고, 값은  [0 1 2 3 4 5 6]
```

한글을 포함한 그래프 그리기

In [30]:

```
#한글 사용을 위해서 matplotlib.font_manager 사용

import matplotlib
import matplotlib.font_manager as fm


font_location="c:/windows/fonts/HMFMMUEX.TTC"  #한글 폰트를 셋팅함

font_name=fm.FontProperties(fname=font_location).get_name()

matplotlib.rc('font',family=font_name) #한글 깨짐을 방지하기 위해서 반드시 해야할 부분
```

In [31]:

```
#vegetables = ["cucumber", "tomato", "lettuce", "asparagus", "potato", "wheat", "barley"]
vegetables = [ "오이", "토마토", "상추", "아스파라거스", "감자", "밀", "보리"]

farmers = ["Farmer Joe", "Upland Bros.", "Smith Gardening", "Agrifun", "Organiculture", "BioGoods Ltd.", "Cornylee Corp."]

harvest = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                    [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                    [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                    [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                    [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                    [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                    [6.3, 2.0, 0.0, 1.4, 0.0, 1.9, 0.1]])
```

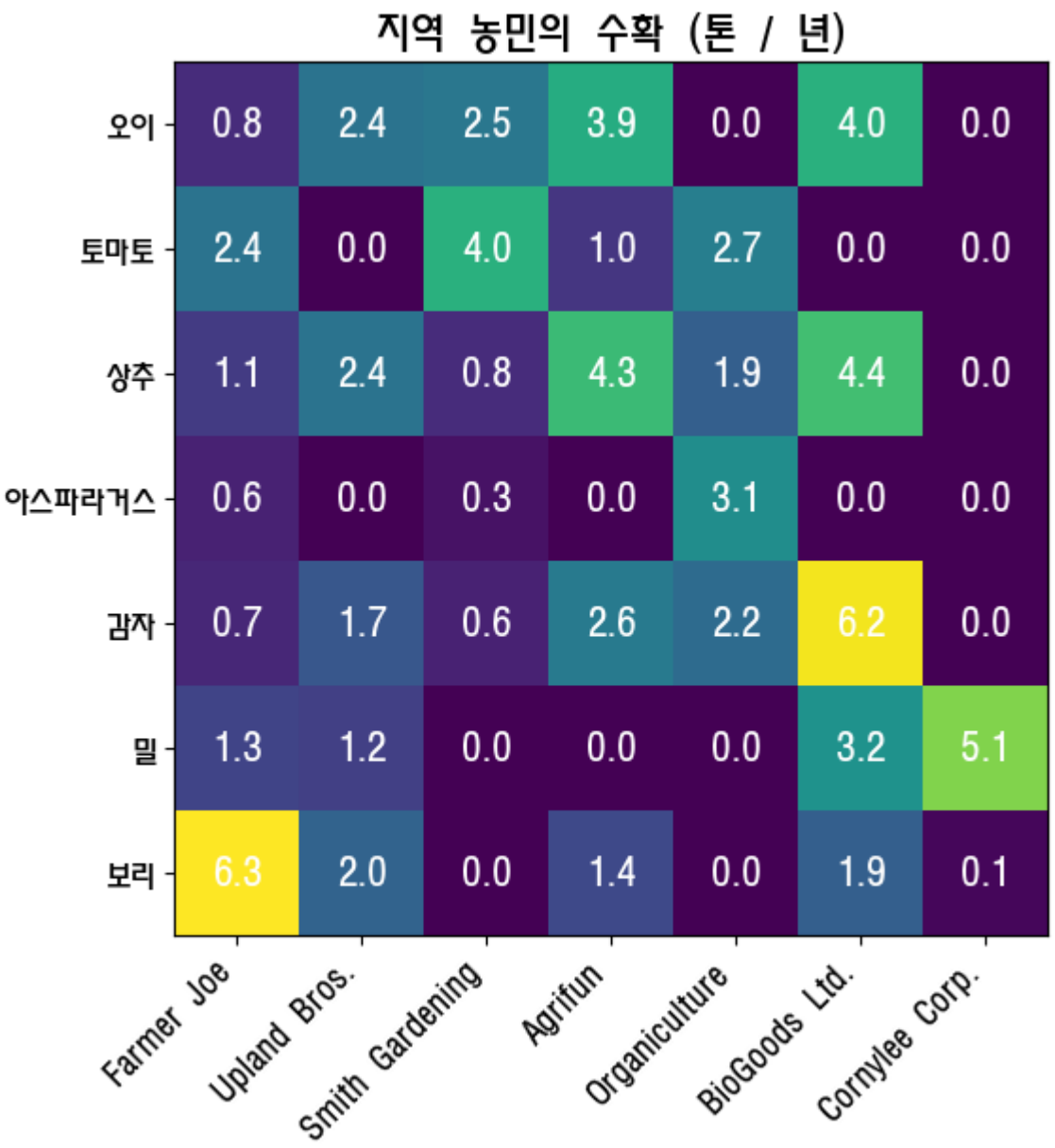
In [32]:

```
#setp 1 ----------------------------------------------------------------
fig, ax = plt.subplots(figsize=(10, 6))
print(fig)  #프레임 생성
print(ax)   #1개의 축 생성

#setp 2 ----------------------------------------------------------------
ax.set_title("지역 농민의 수확 (톤 / 년)", fontsize=16)
#ax.imshow(harvest, cmap='ocean')  #numpy.ndarray 타입 사용 가능
ax.imshow(harvest)  #numpy.ndarray 타입 사용 가능

#setp 3 ----------------------------------------------------------------

#xaxis, yaxis의 눈금 위치를 설정
print('● len(farmers)은', len(farmers), '이고, 값은  ', end='')
print(np.arange(len(farmers)))

print('● len(vegetables)은', len(vegetables),  '이고, 값은  ',end='')
print(np.arange(len(vegetables)))

ax.set_xticks(np.arange(len(farmers)))      #x축 [0 1 2 3 4 5 6]
ax.set_yticks(np.arange(len(vegetables)))   #y축 [0 1 2 3 4 5 6]


#setp 4 ----------------------------------------------------------------

#각각의 x, y축에 레이블링
ax.set_xticklabels(farmers)     #x축 [0 1 2 3 4 5 6]부분에 레이블링이 됨
ax.set_yticklabels(vegetables)  #y축 [0 1 2 3 4 5 6]부분에 레이블링이 됨

#setp 5 ----------------------------------------------------------------

#x축 레이블을 회전하고 정렬
#ha = 'right'는 수평 레이블 텍스트의 오른쪽 끝을 눈금에 맞춥니다.
#ha = 'left'는 수평 레이블 텍스트의 왼쪽 끝을 눈금에 맞춥니다.
#ha = 'center'는 수평 레이블 텍스트의 중심을 눈금에 맞춥니다.

plt.setp(ax.get_xticklabels(), rotation=45, ha="right", fontsize=12)
#plt.setp(ax.get_xticklabels(), rotation=45, ha="left", fontsize=12)
#plt.setp(ax.get_xticklabels(), rotation=45, ha="center", fontsize=12)

#y축 레이블을 정렬
plt.setp(ax.get_yticklabels(), ha="right", fontsize=12)

#setp 6 ----------------------------------------------------------------
#ha 수평 축
#va 수직 축
```

```
47
48 for i in range(len(vegetables)):  #column
49     for j in range(len(farmers)):   #row
50         text = ax.text(j, i, harvest[i, j], ha="center", va="center", color="w", fontsize=14)
51         print(text, end='')
52         print('-->', harvest[i, j])
53     print('-'*50)
54
55 #--------------------------------------------------------------------------
56 plt.show()
```

```
Figure(1000x600)
Axes(0.125,0.11;0.775x0.77)
● len(farmers)은 7 이고, 값은  [0 1 2 3 4 5 6]
● len(vegetables)은 7 이고, 값은  [0 1 2 3 4 5 6]
Text(0, 0, '0.8')--> 0.8
Text(1, 0, '2.4')--> 2.4
Text(2, 0, '2.5')--> 2.5
Text(3, 0, '3.9')--> 3.9
Text(4, 0, '0.0')--> 0.0
Text(5, 0, '4.0')--> 4.0
Text(6, 0, '0.0')--> 0.0
--------------------------------------------------
Text(0, 1, '2.4')--> 2.4
Text(1, 1, '0.0')--> 0.0
Text(2, 1, '4.0')--> 4.0
Text(3, 1, '1.0')--> 1.0
Text(4, 1, '2.7')--> 2.7
Text(5, 1, '0.0')--> 0.0
Text(6, 1, '0.0')--> 0.0
--------------------------------------------------
Text(0, 2, '1.1')--> 1.1
Text(1, 2, '2.4')--> 2.4
Text(2, 2, '0.8')--> 0.8
Text(3, 2, '4.3')--> 4.3
Text(4, 2, '1.9')--> 1.9
Text(5, 2, '4.4')--> 4.4
Text(6, 2, '0.0')--> 0.0
--------------------------------------------------
Text(0, 3, '0.6')--> 0.6
Text(1, 3, '0.0')--> 0.0
Text(2, 3, '0.3')--> 0.3
Text(3, 3, '0.0')--> 0.0
Text(4, 3, '3.1')--> 3.1
Text(5, 3, '0.0')--> 0.0
Text(6, 3, '0.0')--> 0.0
--------------------------------------------------
Text(0, 4, '0.7')--> 0.7
Text(1, 4, '1.7')--> 1.7
Text(2, 4, '0.6')--> 0.6
Text(3, 4, '2.6')--> 2.6
Text(4, 4, '2.2')--> 2.2
Text(5, 4, '6.2')--> 6.2
Text(6, 4, '0.0')--> 0.0
--------------------------------------------------
Text(0, 5, '1.3')--> 1.3
Text(1, 5, '1.2')--> 1.2
```

```
Text(2, 5, '0.0')--> 0.0
Text(3, 5, '0.0')--> 0.0
Text(4, 5, '0.0')--> 0.0
Text(5, 5, '3.2')--> 3.2
Text(6, 5, '5.1')--> 5.1
------------------------------------------------
Text(0, 6, '6.3')--> 6.3
Text(1, 6, '2.0')--> 2.0
Text(2, 6, '0.0')--> 0.0
Text(3, 6, '1.4')--> 1.4
Text(4, 6, '0.0')--> 0.0
Text(5, 6, '1.9')--> 1.9
Text(6, 6, '0.1')--> 0.1
------------------------------------------------
```


지역 농민의 수확 (톤 / 년)
오이
토마토
상추
아스파라거스
감자
밀
보리
0.8 2.4 2.5 3.9 0.0 4.0 0.0
2.4 0.0 4.0 1.0 2.7 0.0 0.0
1.1 2.4 0.8 4.3 1.9 4.4 0.0
0.6 0.0 0.3 0.0 3.1 0.0 0.0
0.7 1.7 0.6 2.6 2.2 6.2 0.0
1.3 1.2 0.0 0.0 0.0 3.2 5.1
6.3 2.0 0.0 1.4 0.0 1.9 0.1
Farmer Joe
Upland Bros.
Smith Gardening
Agrifun
Organiculture
BioGoods Ltd.
Cornylee Corp.

In [ ]:
```
1
```