In [5]:

```
import pandas as pd
df=pd.DataFrame(['A+','A','A-','B+','B','B-','C+','C','C-',
                 'D+','D'],
                index=['excellent','excellent','excellent',
                       'good','good','good','ok','ok','ok',
                       'poor','poor'])
df.rename(columns={0:'Grades'},inplace=True)
df
```

Out[5]:

	Grades
excellent	A+
excellent	A
excellent	A-
good	B+
good	B
good	B-
ok	C+
ok	C
ok	C-
poor	D+
poor	D

In [7]:

```
df['Grades'].astype('category').head()
```

Out[7]:

```
excellent    A+
excellent     A
excellent    A-
good         B+
good          B
Name: Grades, dtype: category
Categories (11, object): [A, A+, A-, B, ..., C+, C-, D, D+]
```

In [23]:

```
#grades=df['Grades'].astype('category',
#                          categories= ['D','D+','C-','C+',
 #                                      'B-','B','B+',
 #                                      'A-','A','A+'],
 #                         ordered=True)
from pandas.api.types import CategoricalDtype
grades=df['Grades'].astype(CategoricalDtype(categories= ['D','D+','C-','C','C+',
                                     'B-','B','B+',
                                     'A-','A','A+'],ordered=True))
grades.head()
```

Out[23]:

```
excellent    A+
excellent     A
excellent    A-
good         B+
good          B
Name: Grades, dtype: category
Categories (11, object): [D < D+ < C- < C ... B+ < A- < A < A+]
```

In [24]:

```
grades >'C'
```

Out[24]:

```
excellent     True
excellent     True
excellent     True
good          True
good          True
good          True
ok            True
ok           False
ok           False
poor         False
poor         False
Name: Grades, dtype: bool
```

In [25]:

```
grades > 'B+'
```

Out[25]:

```
excellent     True
excellent     True
excellent     True
good         False
good         False
good         False
ok           False
ok           False
ok           False
poor         False
poor         False
Name: Grades, dtype: bool
```

In [26]:

```
s=pd.Series([168,180,174,190,170,185,179,181,175,169,
             182,177,180,171])
s
```

Out[26]:

```
0     168
1     180
2     174
3     190
4     170
5     185
6     179
7     181
8     175
9     169
10    182
11    177
12    180
13    171
dtype: int64
```

In [28]:

```
pd.cut(s,3)#구간이 3개
```

Out[28]:

```
0     (167.978, 175.333]
1     (175.333, 182.667]
2     (167.978, 175.333]
3       (182.667, 190.0]
4     (167.978, 175.333]
5       (182.667, 190.0]
6     (175.333, 182.667]
7     (175.333, 182.667]
8     (167.978, 175.333]
9     (167.978, 175.333]
10    (175.333, 182.667]
11    (175.333, 182.667]
12    (175.333, 182.667]
13    (167.978, 175.333]
dtype: category
Categories (3, interval[float64]): [(167.978, 175.333] < (175.333, 182.667] < (18
2.667, 190.0]]
```

In [30]:

```
pd.cut(s,3,labels=['Small','Medium','Large'])
```

Out[30]:

```
0      Small
1     Medium
2      Small
3      Large
4      Small
5      Large
6     Medium
7     Medium
8      Small
9      Small
10    Medium
11    Medium
12    Medium
13     Small
dtype: category
Categories (3, object): [Small < Medium < Large]
```

In [31]:

```
## Basic Plotting with matplotlib
```

In [32]:

```
%matplotlib notebook
# %matplotlib inline
import matplotlib as mpl
mpl.get_backend()
```

Out[32]:

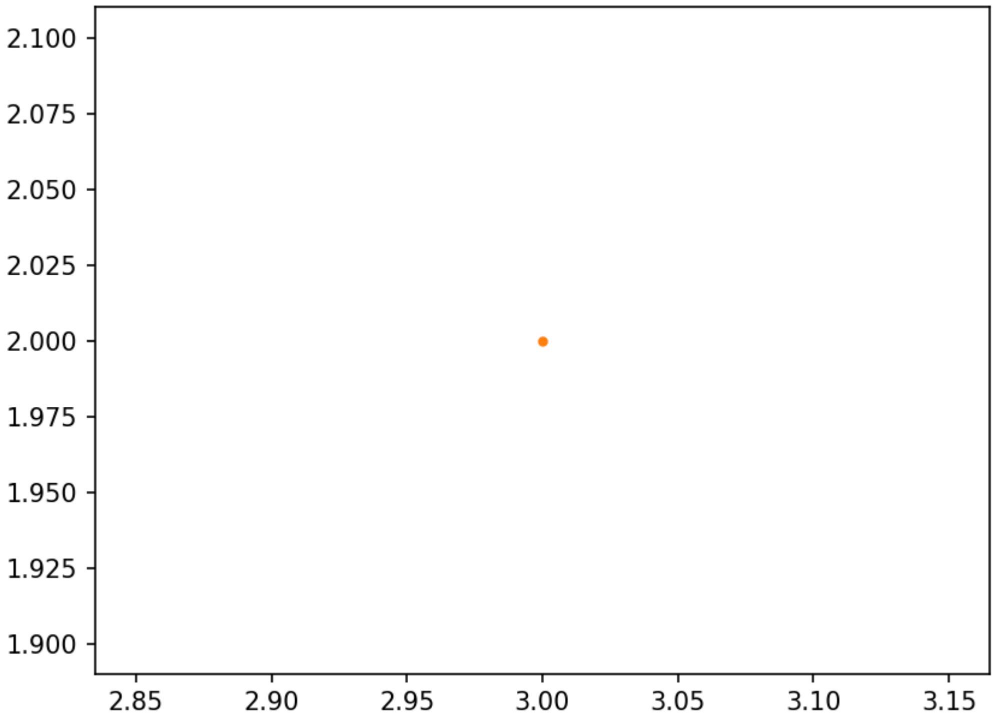
```
'nbAgg'
```

In [33]:

```
import matplotlib.pyplot as plt
```

In [34]:

```
plt.plot(3,2)
```



Out[34]:

```
[<matplotlib.lines.Line2D at 0x1ade3934988>]
```

In [35]:

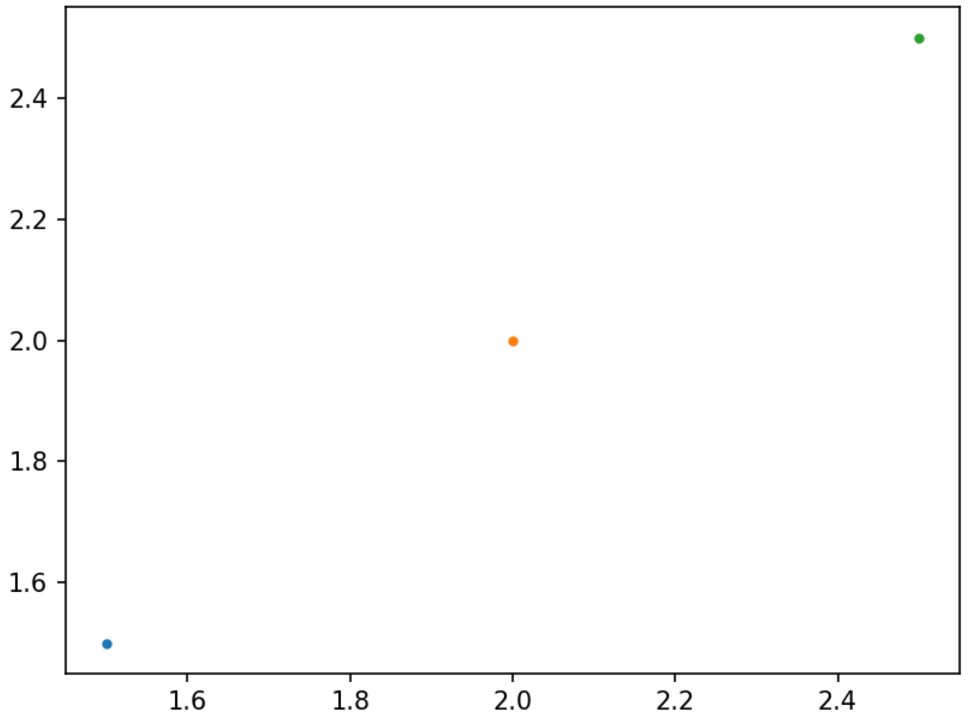
```
plt.plot(3,2,'.')
```

Out[35]:

```
[<matplotlib.lines.Line2D at 0x1ade3f2fe48>]
```

In [37]:

```
plt.figure()
plt.plot(1.5,1.5,'.')
plt.plot(2,2,'.')
plt.plot(2.5,2.5,'.')
```

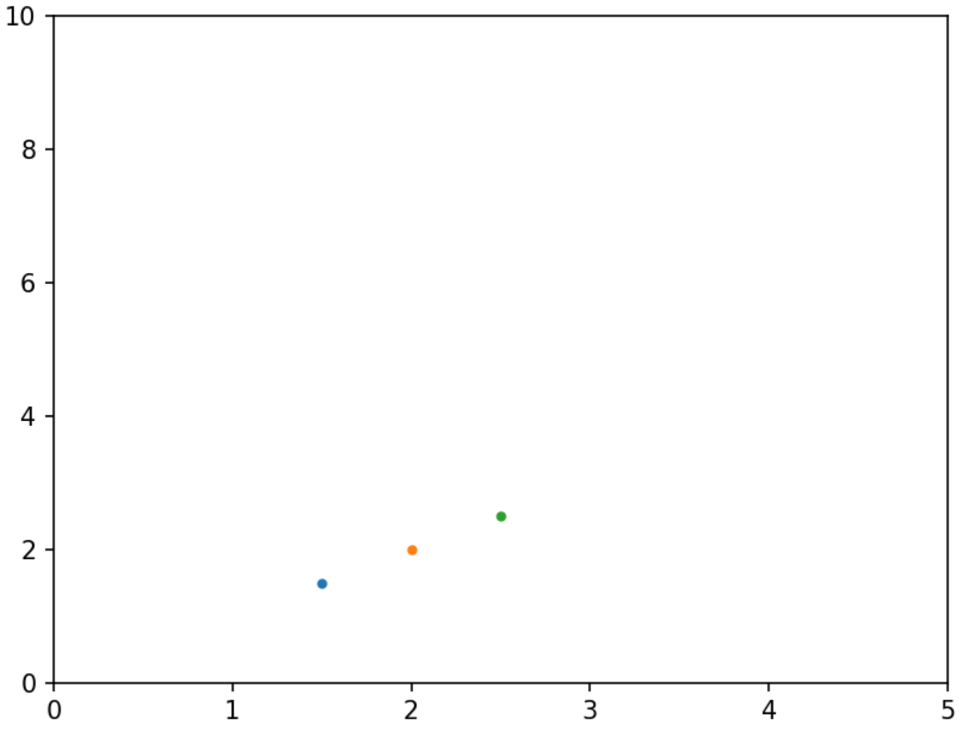


Out[37]:

```
[<matplotlib.lines.Line2D at 0x1ade3f51b88>]
```

In [40]:

```
plt.figure()
plt.plot(1.5,1.5,'.')
plt.plot(2,2,'.')
plt.plot(2.5,2.5,'.')
ax=plt.gca()
ax.axis([0,5,0,10])
```



Out[40]:

```
[0, 5, 0, 10]
```

In [41]:

```
import numpy as np
```

In [42]:

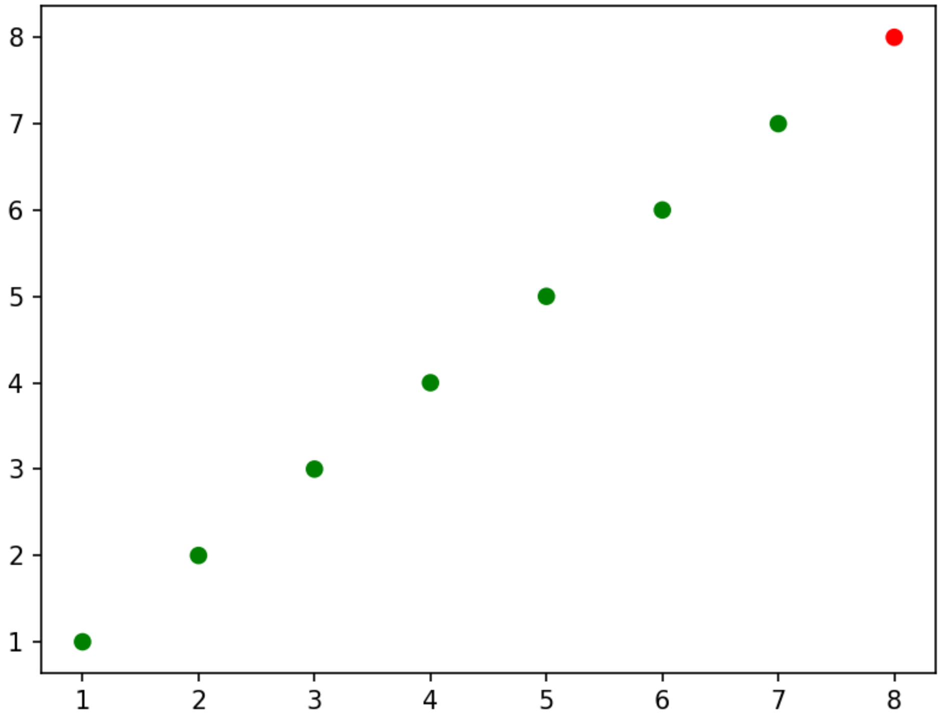
```
x=np.array([1,2,3,4,5,6,7,8])
```

In [43]:

```
y=x
```

In [44]:

```
colors=['green']*(len(x)-1)
colors.append('red')
plt.figure()
plt.scatter(x,y,c=colors)
```

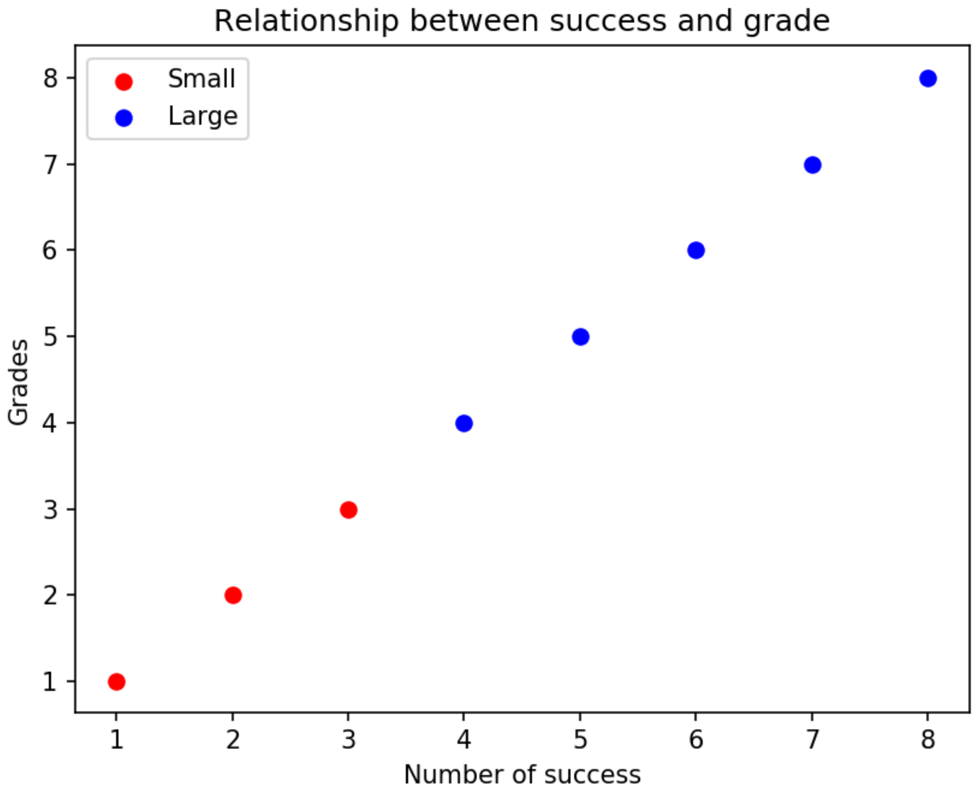


Out[44]:

```
<matplotlib.collections.PathCollection at 0x1ade5ca5608>
```

In [45]:

```
plt.figure()
plt.scatter(x[:3],y[:3],c='red',label='Small')
plt.scatter(x[3:],y[3:],c='blue',label='Large')
```



Out[45]:

```
<matplotlib.collections.PathCollection at 0x1ade6265cc8>
```

In [47]:

```
plt.xlabel('Number of success')
plt.ylabel('Grades')
plt.title('Relationship between success and grade')
plt.legend()# 좌측 상단 범례를 보여줌
```

Out[47]:

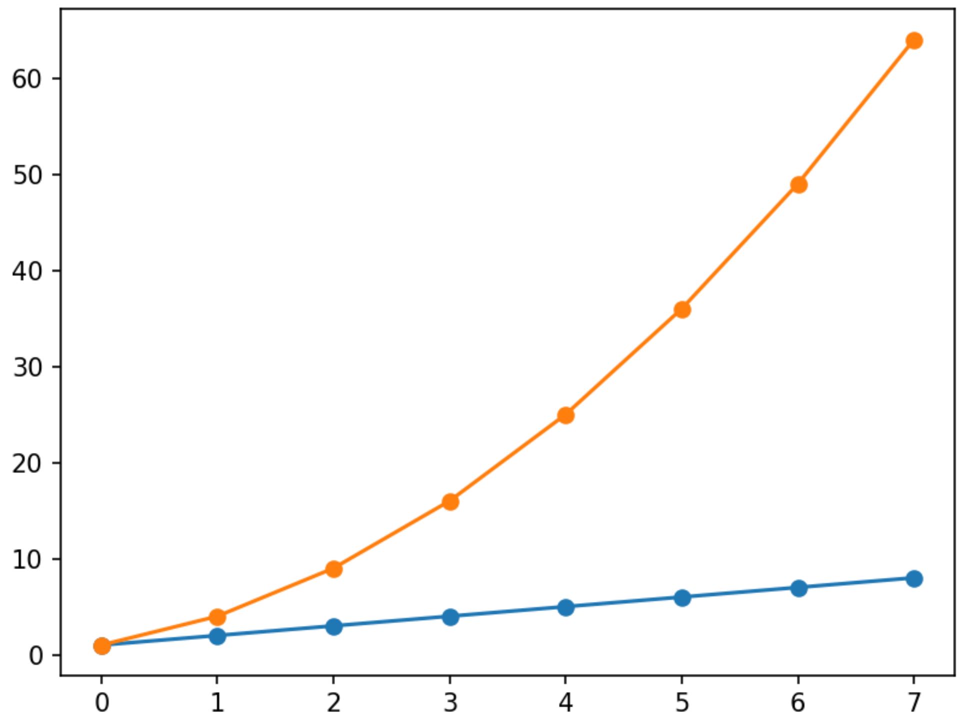
```
<matplotlib.legend.Legend at 0x1ade6820c48>
```

In [48]:

```
## Line Plots
```

In [49]:

```
linear_data=np.array([1,2,3,4,5,6,7,8])
exponential_data=linear_data**2
plt.figure()
plt.plot(linear_data,'-o', exponential_data,'-o')
```

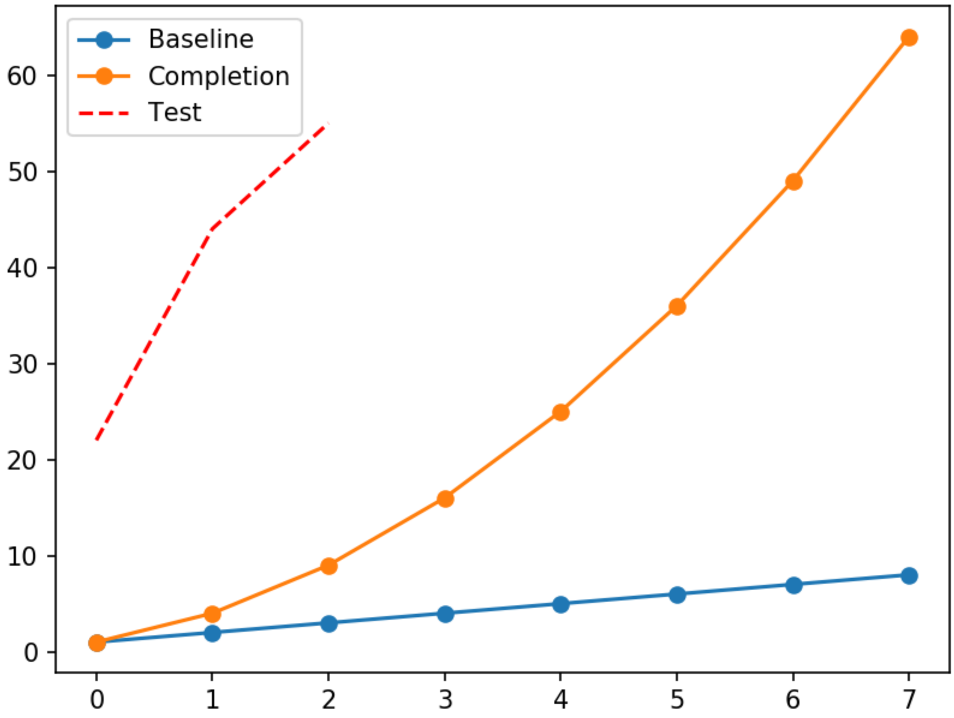


Out[49]:

```
[<matplotlib.lines.Line2D at 0x1ade6836b48>,
 <matplotlib.lines.Line2D at 0x1ade6856f08>]
```

In [51]:

```
plt.figure()
plt.plot(linear_data,'-o', exponential_data,'-o')
#-o는 라인 스타일
plt.plot([22,44,55],'--r')
plt.legend(['Baseline','Completion','Test'])
```

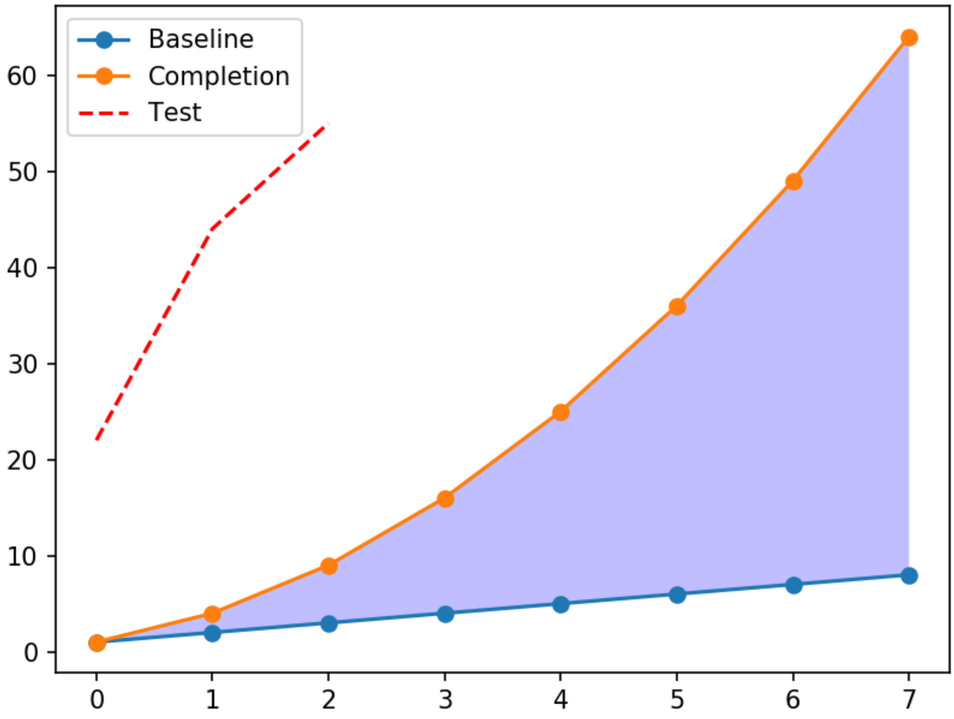


Out[51]:

```
<matplotlib.legend.Legend at 0x1ade8410448>
```

In [54]:

```
plt.figure()
plt.plot(linear_data,'-o', exponential_data,'-o')
#-o는 라인 스타일
plt.plot([22,44,55],'--r')
plt.legend(['Baseline','Completion','Test'])
plt.fill_between(range(len(linear_data)),
                    linear_data,exponential_data,
                    facecolor='blue', alpha=0.25)
```



Out[54]:

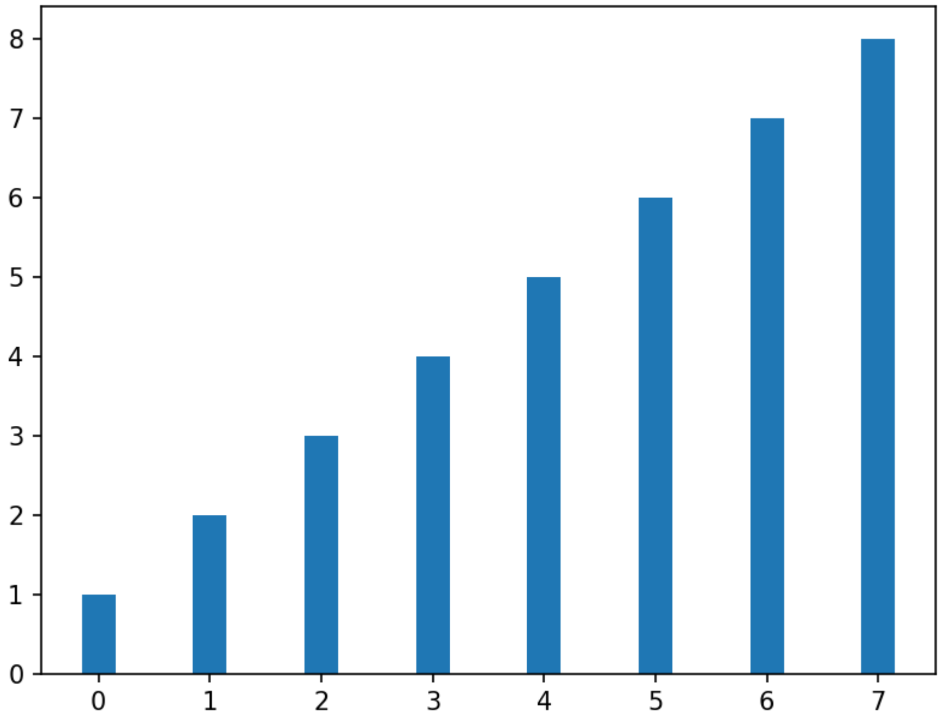
```
<matplotlib.collections.PolyCollection at 0x1ade8fe19c8>
```

In [56]:

```
##Bar Charts
```

In [58]:

```
plt.figure()
xvals=range(len(linear_data))
plt.bar(xvals,linear_data,width=0.3)
```



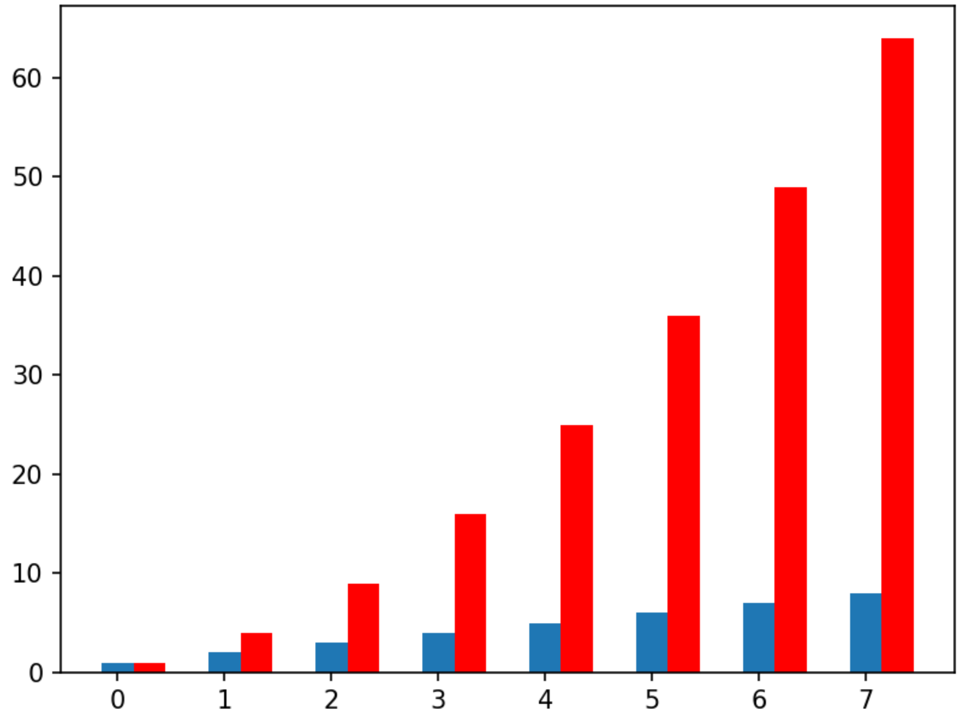
Out[58]:

```
<BarContainer object of 8 artists>
```

In [61]:

```
plt.figure()
xvals=range(len(linear_data))
plt.bar(xvals,linear_data,width=0.3)
new_xvals=[]
for item in xvals:
    new_xvals.append(item+0.3)

plt.bar(new_xvals,exponential_data,width=0.3,color='red')
```

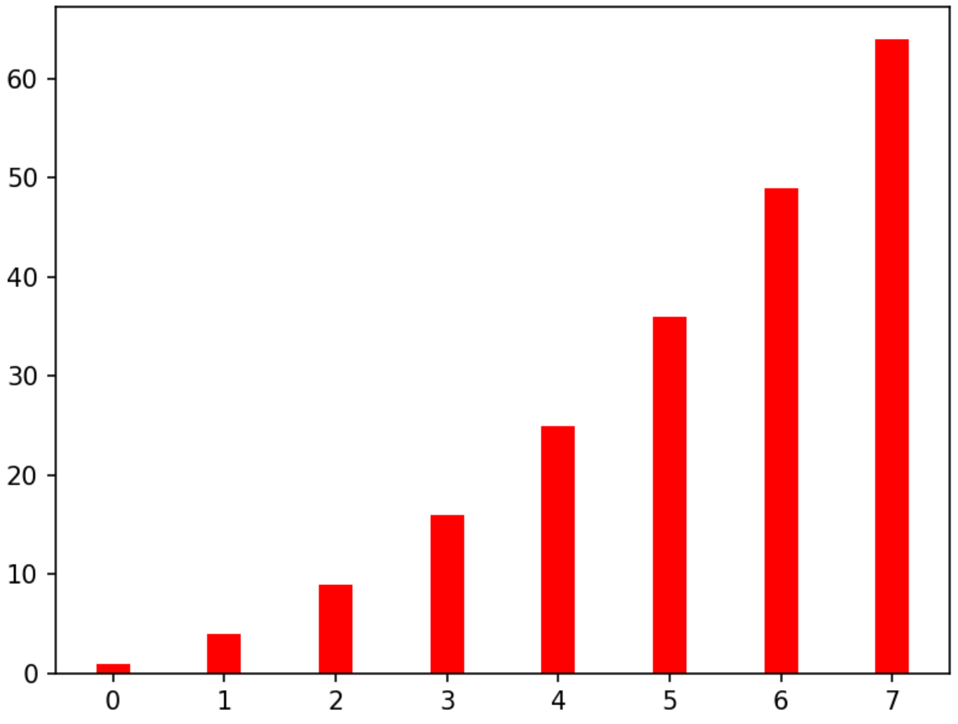


Out[61]:

```
<BarContainer object of 8 artists>
```

In [63]:

```
plt.figure()
xvals=range(len(linear_data))
plt.bar(xvals,linear_data,width=0.3)
plt.bar(xvals,exponential_data,width=0.3,color='red')
```

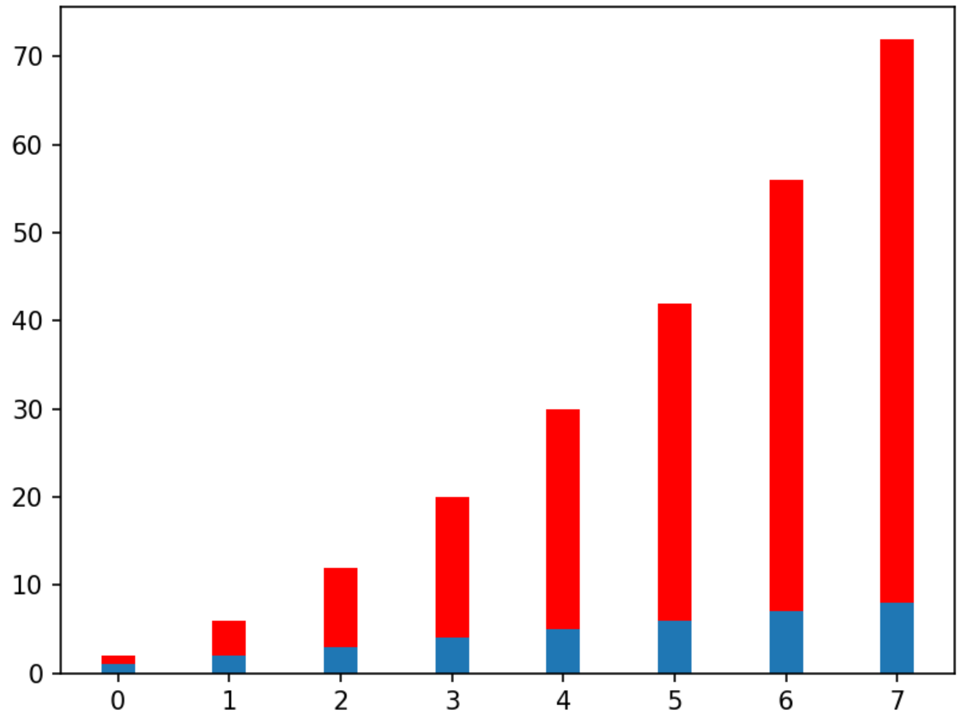


Out[63]:

```
<BarContainer object of 8 artists>
```

In [64]:

```
plt.figure()
xvals=range(len(linear_data))
plt.bar(xvals,linear_data,width=0.3)
plt.bar(xvals,exponential_data,width=0.3,bottom=linear_data,color='red')
```



Out[64]:

```
<BarContainer object of 8 artists>
```

In [ ]: