In [1]:

```
import matplotlib
matplotlib.__version__
```

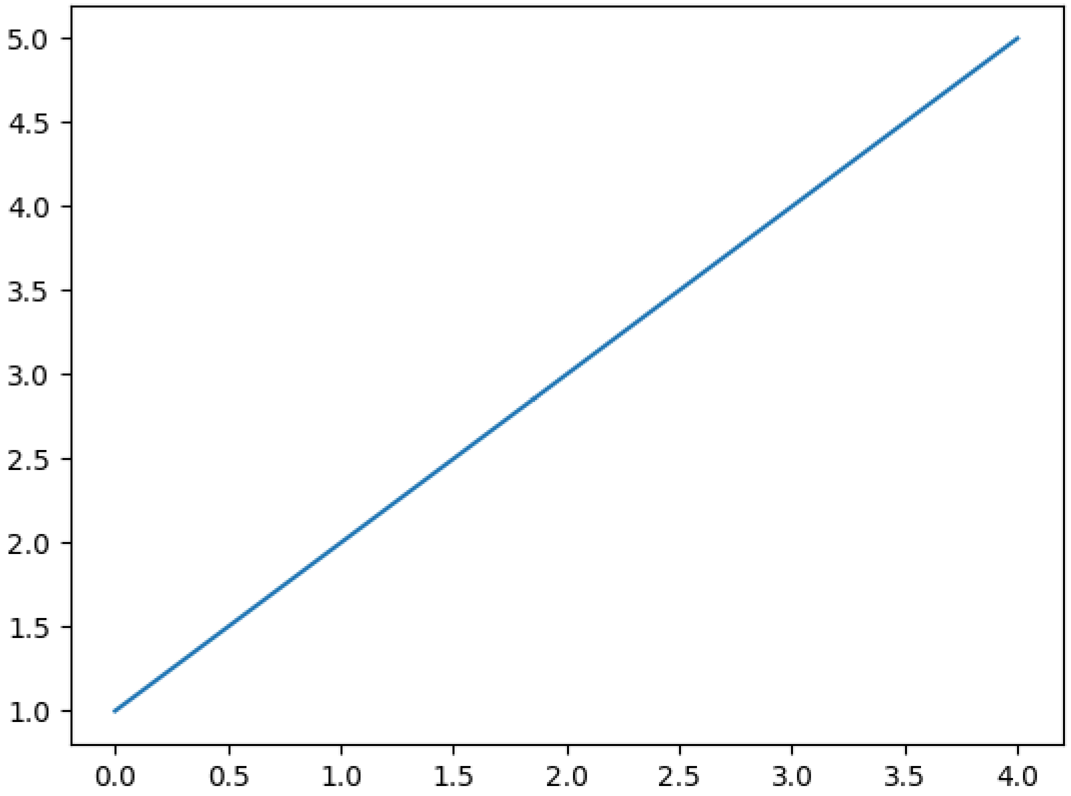
Out[1]:

```
'3.7.1'
```

In [5]:

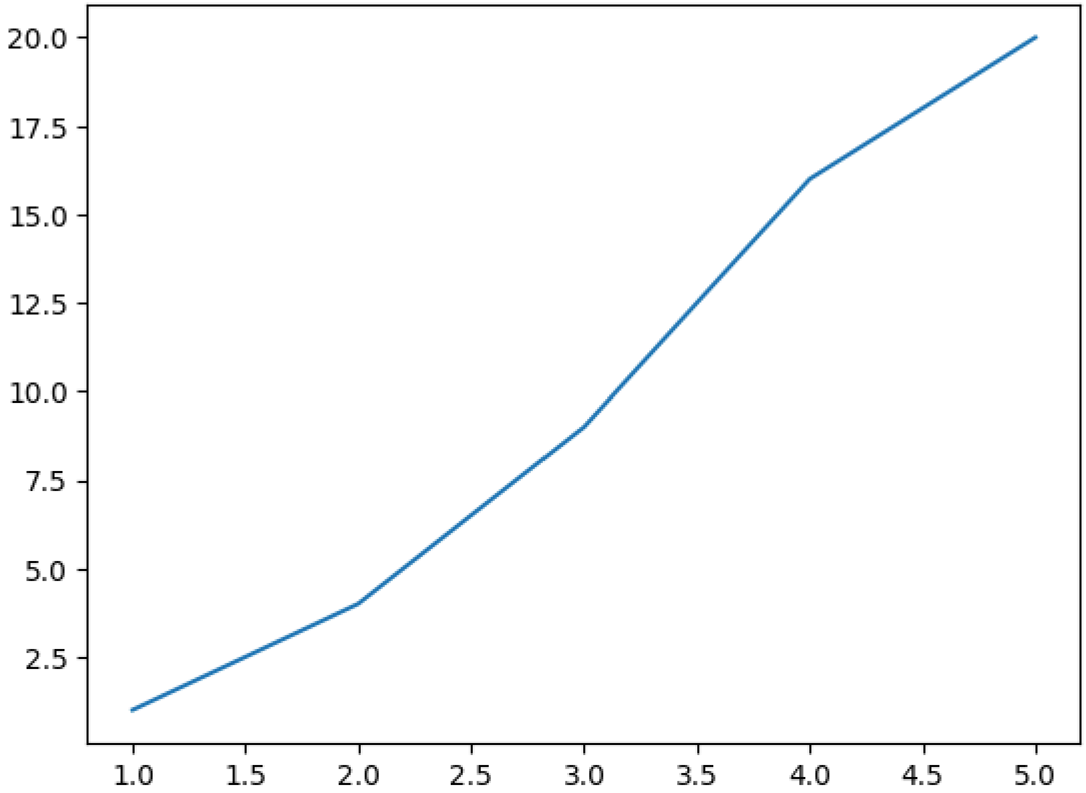
```
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4, 5])
plt.show()
```



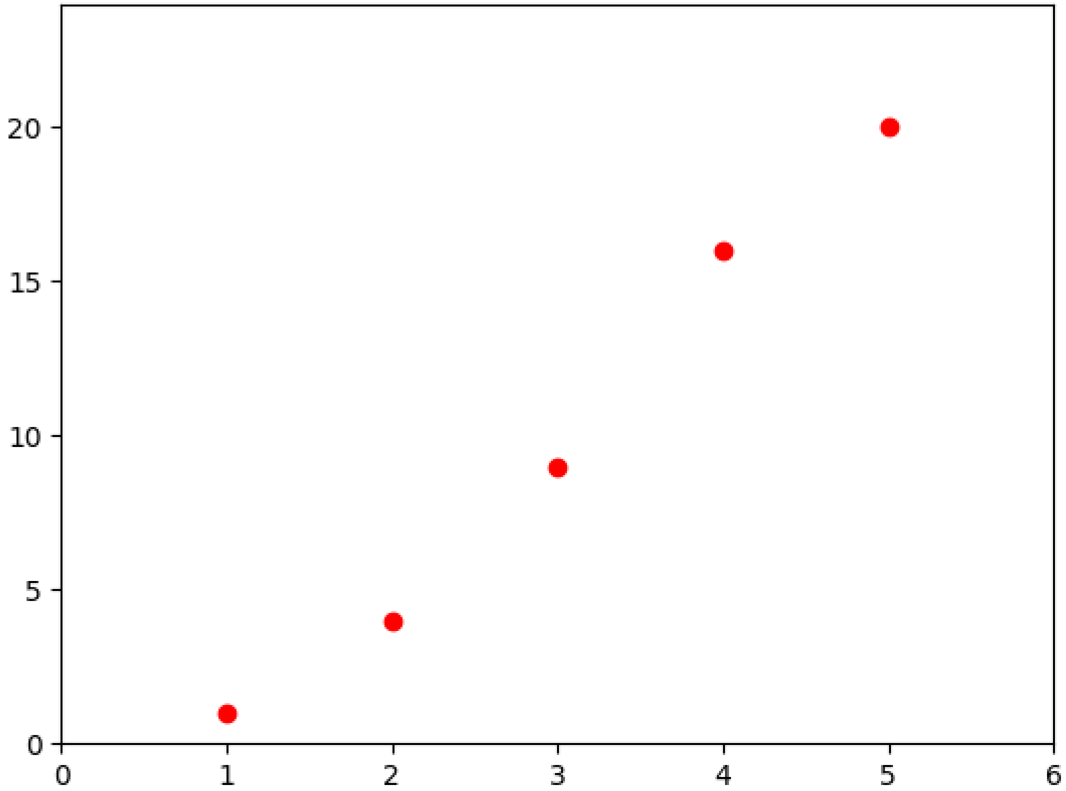
In [7]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
plt.show()
```



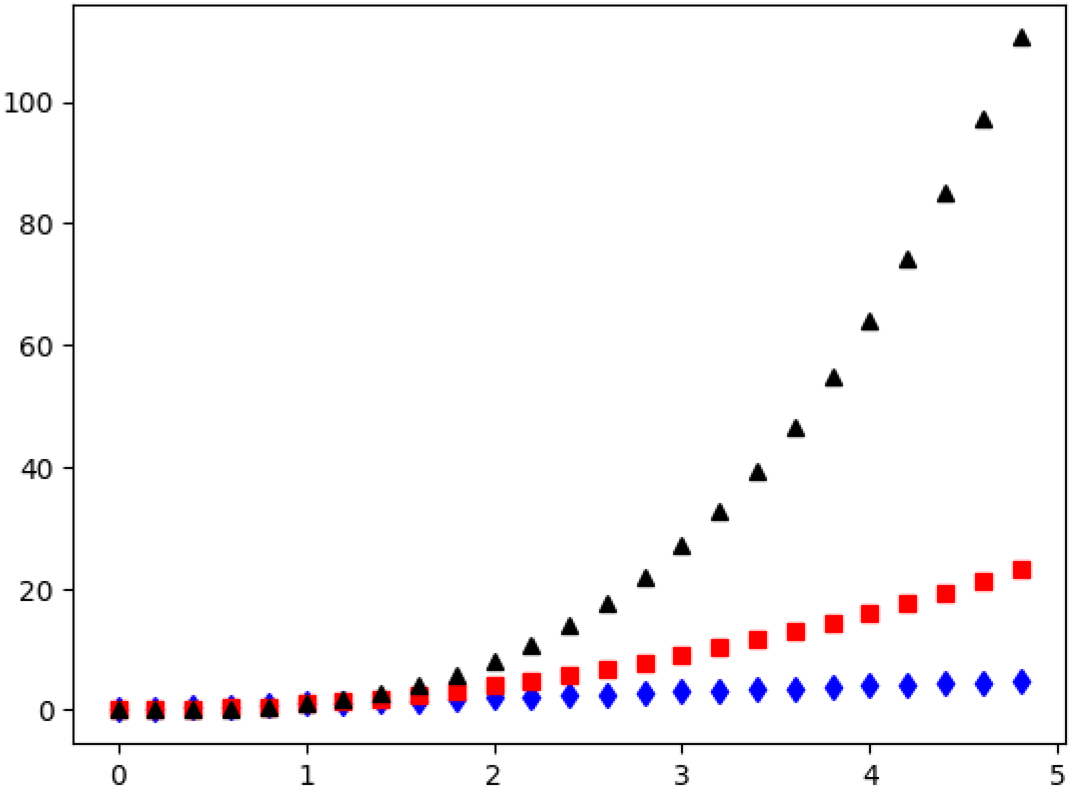
In [8]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'ro')
plt.axis([0, 6, 0, 24])
plt.show()
```



In [ ]:

```
import numpy as np
t = np.arange(0., 5., 0.2)
plt.plot(t, t, 'bd', t, t**2, 'rs', t, t**3, 'k^')
plt.show()
```



색깔 포맷

포맷	예시
RGB 또는 RGBA (빨강, 초록, 파랑, 알파) 튜플. 값 범위는 [0, 1]	(0.2, 0.3, 0.4) (0.2, 0.3, 0.4, 0.3)
대소문자 구분 없는 hex RGB 또는 RGBA 문자열	'#0f0f0f' '#0f0f0f80'
중복된 문자로 구성된 대소문자 구분 없는 hex RGB 또는 RGBA 축약형 문자열	'#abc' -> '#aabbcc' '#fb1' -> '#ffbb11'
[0, 1] 범위의 float 값을 문자열로 표현 (그레이스케일 값)	'0' -> 검정 '1' -> 흰색 '0.8' -> 밝은 회색
기본 색상에 대한 한 글자 축약 표기	'b' -> 파랑, 'g' -> 초록, 'r' -> 빨강 'c' -> 청록, 'm' -> 자홍, 'y' -> 노랑 'k' -> 검정, 'w' -> 흰색
'T10' 범주형 팔레트에서 대소문자 구분 없는 Tableau 색상	'tab:blue', 'tab:orange', 'tab:green' 'tab:red', 'tab:purple', 'tab:brown' 'tab:pink', 'tab:gray', 'tab:olive', 'tab:cyan'

선 스타일

스타일 이름	문자	설명	예시
solid	-	실선 스타일	————————
dashed	--	점선 스타일	— — — — — — —
dashdot	-.	점-선 혼합 스타일	— · — · — · — · —
dotted	:	점선 스타일	....

마커 타입

Unfilled markers - 색 변환 불가

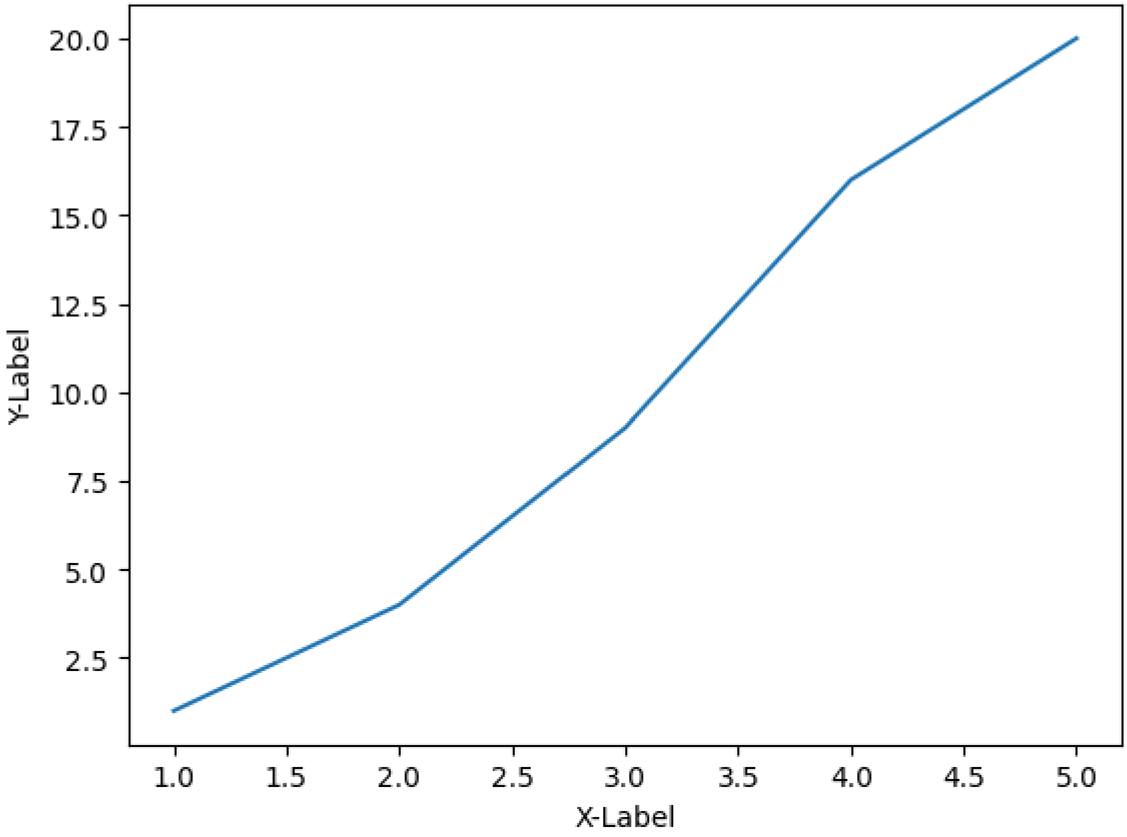
마커	심볼	설명
','	·	픽셀 (pixel)
'1'	⊥	아래 삼각형 (tri_down)
'2'	⊤	위 삼각형 (tri_up)
'3'	├	왼쪽 삼각형 (tri_left)
'4'	┤	오른쪽 삼각형 (tri_right)
'+'	+	플러스 (plus)
'x'	×	엑스 (x)
' | '	|	수직선 (vline)
'_'	—	수평선 (hline)
0	—	왼쪽 눈금 (tick left)
1	—	오른쪽 눈금 (tick right)
2	|	위쪽 눈금 (tick up)
3	|	아래쪽 눈금 (tick down)
4	◀	왼쪽 꺾쇠 (caret left)
5	▶	오른쪽 꺾쇠 (caret right)
6	▲	위쪽 꺾쇠 (caret up)
7	▼	아래쪽 꺾쇠 (caret down)
8	◁	왼쪽 중심 꺾쇠 (caret left)
9	▷	오른쪽 중심 꺾쇠 (caret right)
10	△	위쪽 중심 꺾쇠 (caret up)
11	▽	아래쪽 중심 꺾쇠 (기준선 중심) (caret down centered at base)

Filled markers

마커	심볼	설명
'.'	●	점 (point)
'o'	●	원 (circle)
'v'	▼	아래 삼각형 (triangle_down)
'^'	▲	위 삼각형 (triangle_up)
'<'	◀	왼쪽 삼각형 (triangle_left)
'>'	▶	오른쪽 삼각형 (triangle_right)
'8'	◉	팔각형 (octagon)
's'	■	정사각형 (square)
'p'	⬟	오각형 (pentagon)
'*'	★	별 (star)
'h'	⬢	육각형1 (hexagon1)
'H'	⬣	육각형2 (hexagon2)
'D'	◆	마름모 (diamond)
'd'	◇	얇은 마름모 (thin_diamond)
'P'	✚	플러스 (채워진) (plus(filled))
'X'	✖	엑스 (채워진) (x(filled))

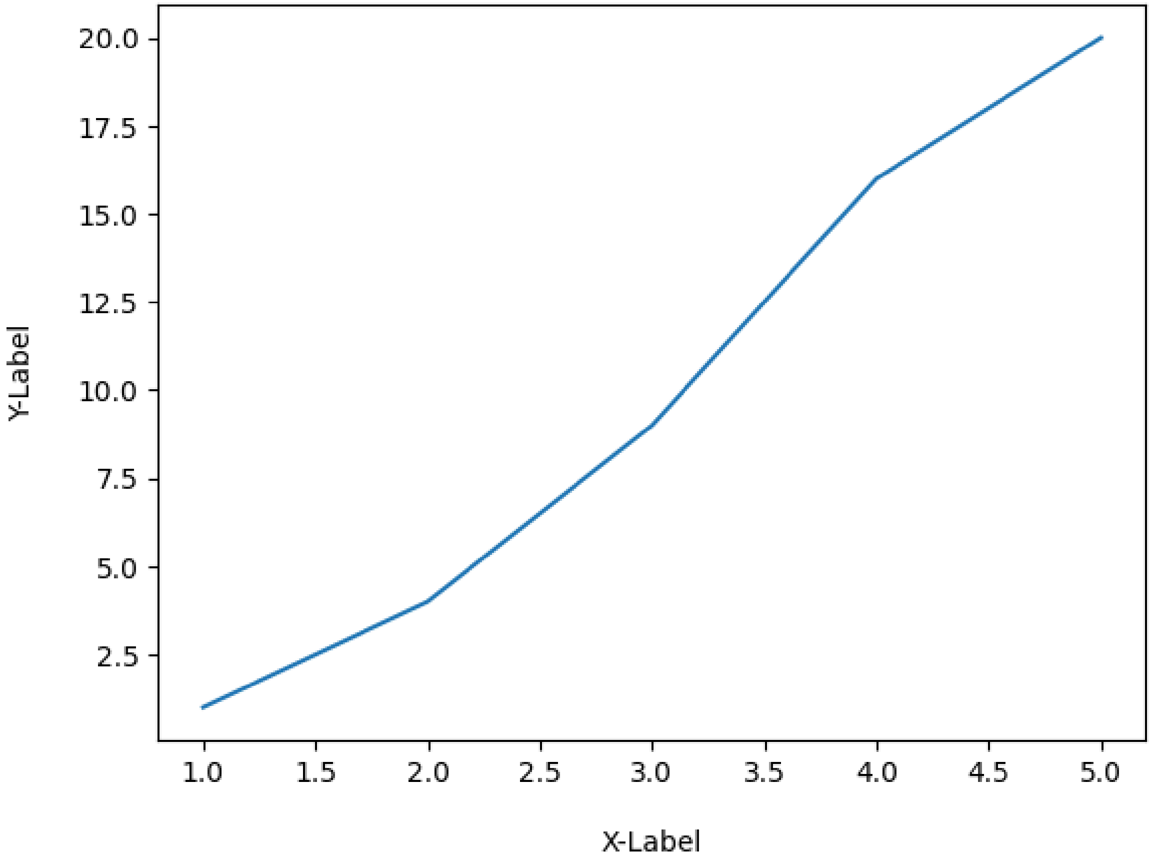
In [11]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.show()
```

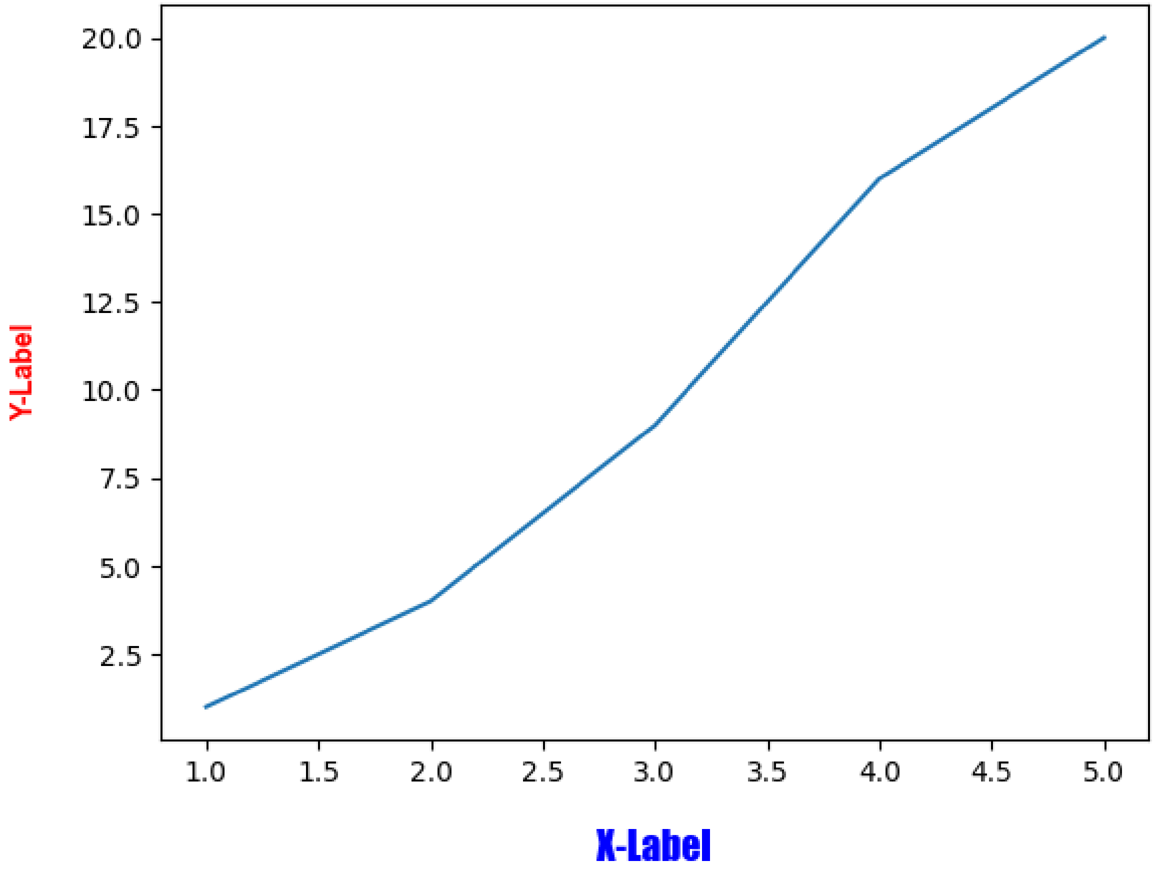


In [12]:

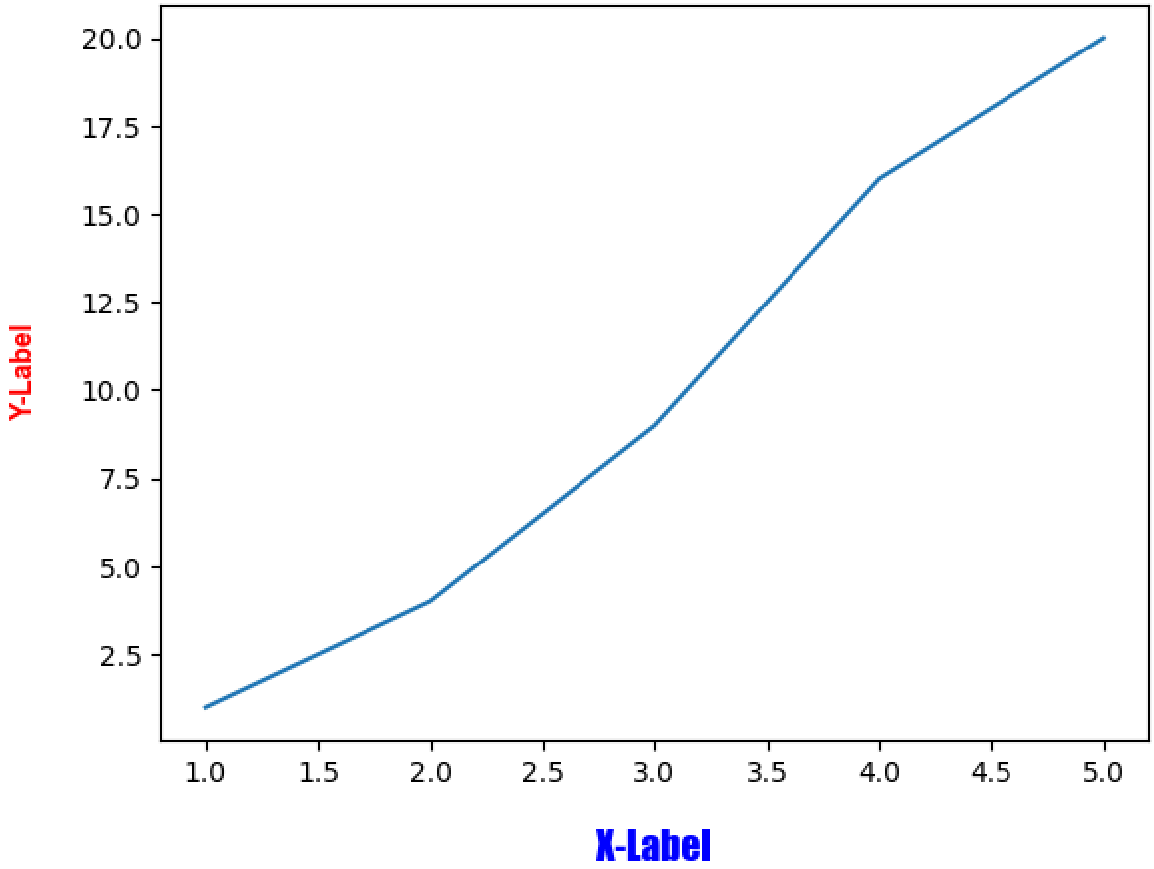
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
plt.xlabel('X-Label', labelpad=15)
plt.ylabel('Y-Label', labelpad=15)
plt.show()
```



In [13]:
```
# Setting Labels Fonts
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
xlabel_font = {'fontname': 'fantasy', 'color': 'b', 'fontweight': 'heavy', 'font
ylabel_font = {'fontname': 'Arial', 'color': 'r', 'fontweight': 'bold', 'fontsiz
plt.xlabel('X-Label', labelpad=15, fontdict = xlabel_font)
plt.ylabel('Y-Label', labelpad=15, fontdict = ylabel_font)
plt.show()
```

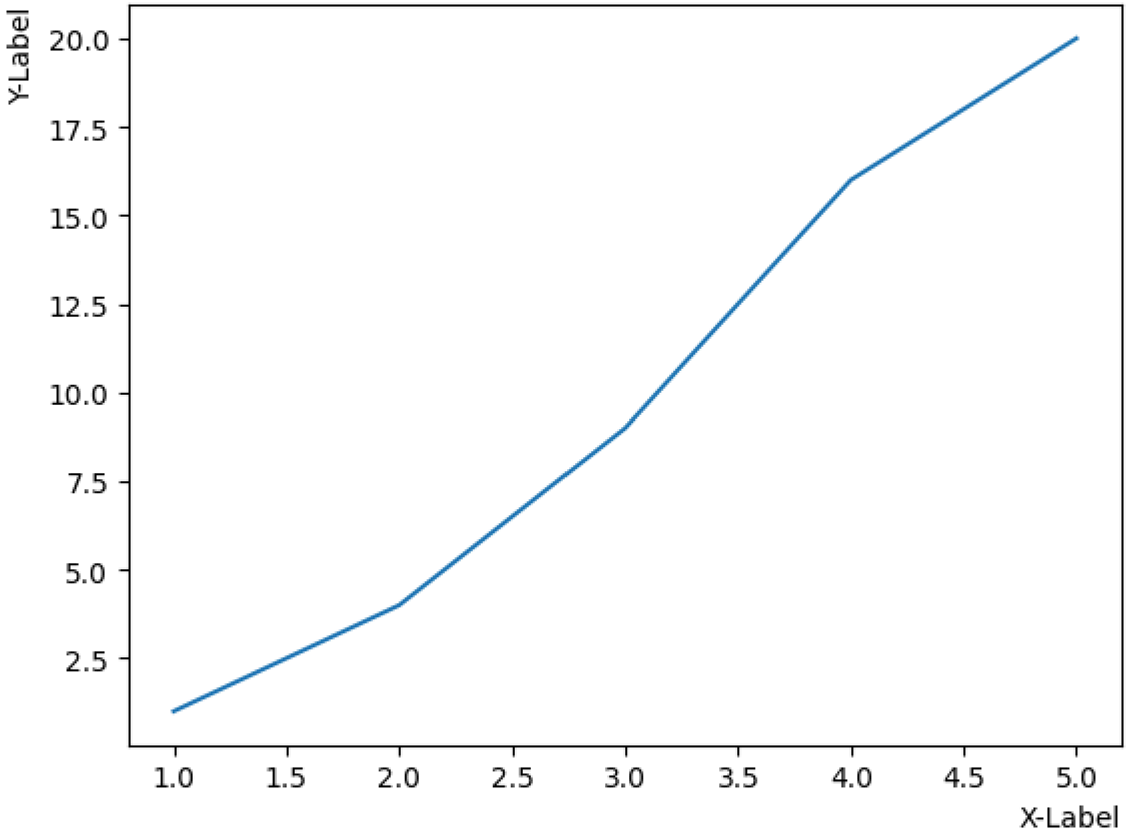


In [14]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
xlabel_font = {'name': 'fantasy', 'color': 'b', 'weight': 'heavy', 'size': 14}
ylabel_font = {'name': 'Arial', 'color': 'r', 'weight': 'bold', 'size': 10}
plt.xlabel('X-Label', labelpad=15, **xlabel_font)
plt.ylabel('Y-Label', labelpad=15, **ylabel_font)
plt.show()
```

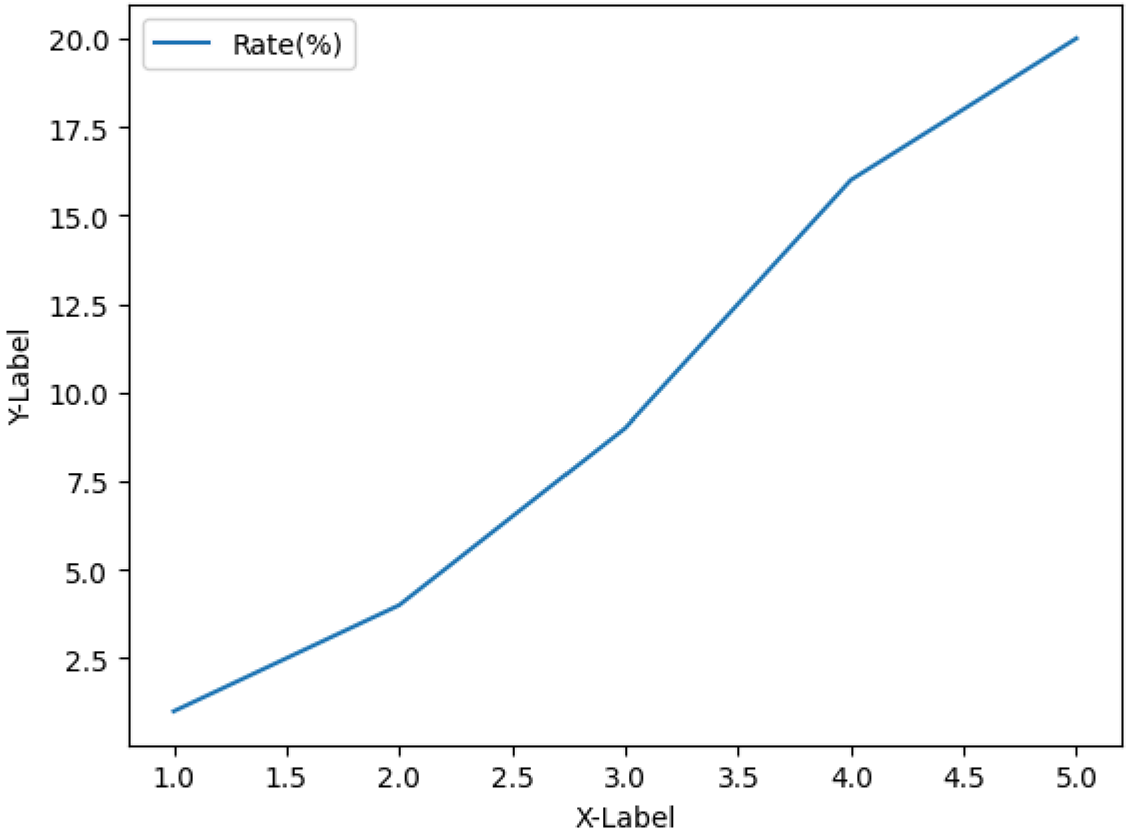


In [15]:

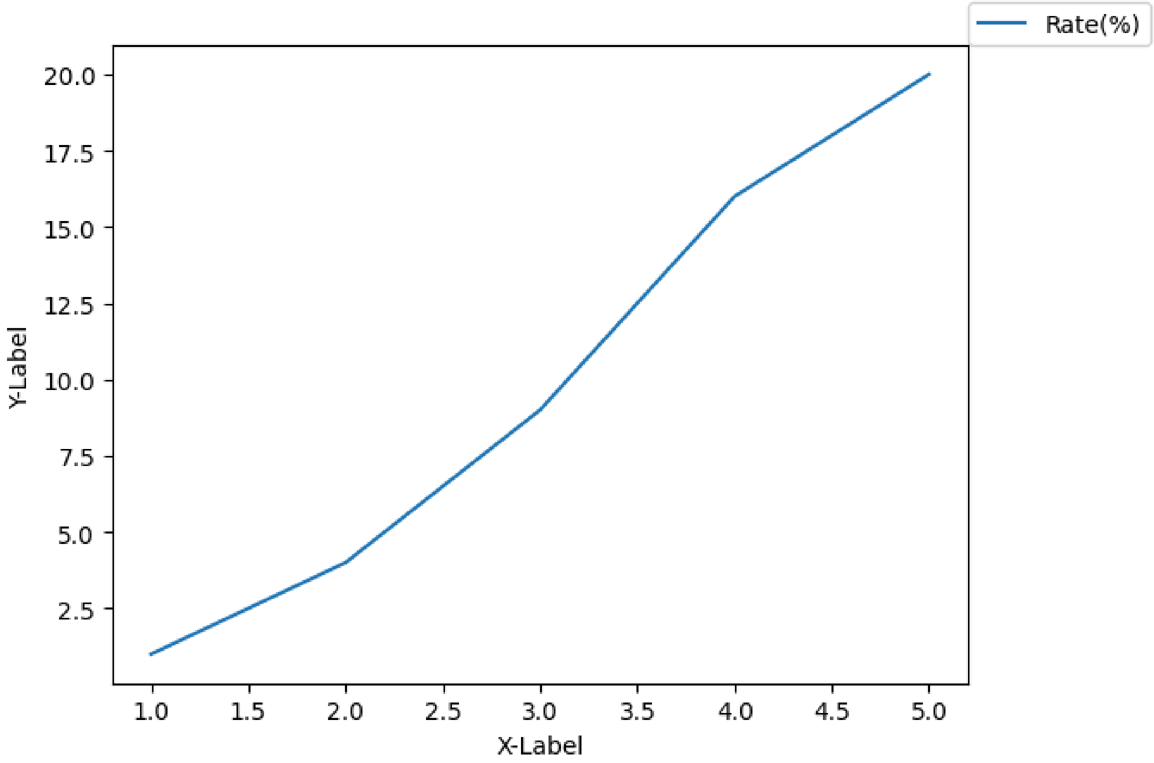
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
plt.xlabel('X-Label', loc = 'right')
plt.ylabel('Y-Label', loc = 'top')
plt.show()
```



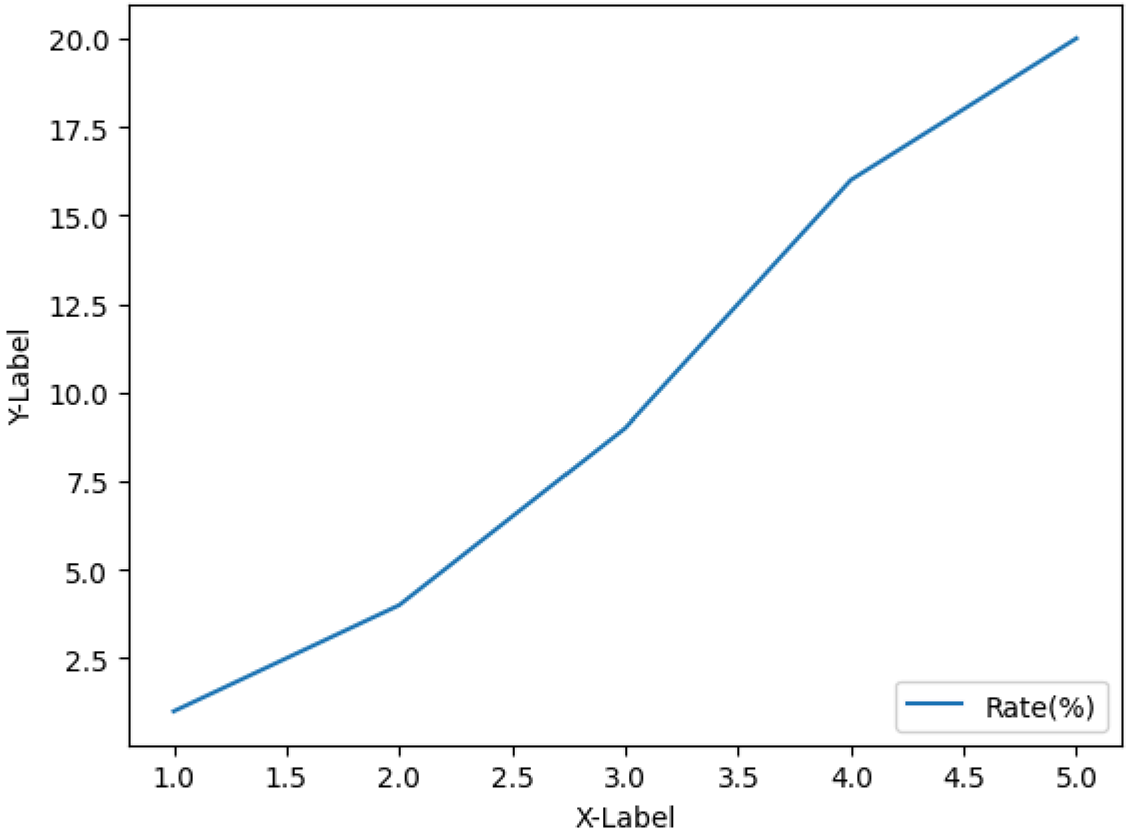
In [18]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], label='Rate(%)')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.legend()
plt.show()
```



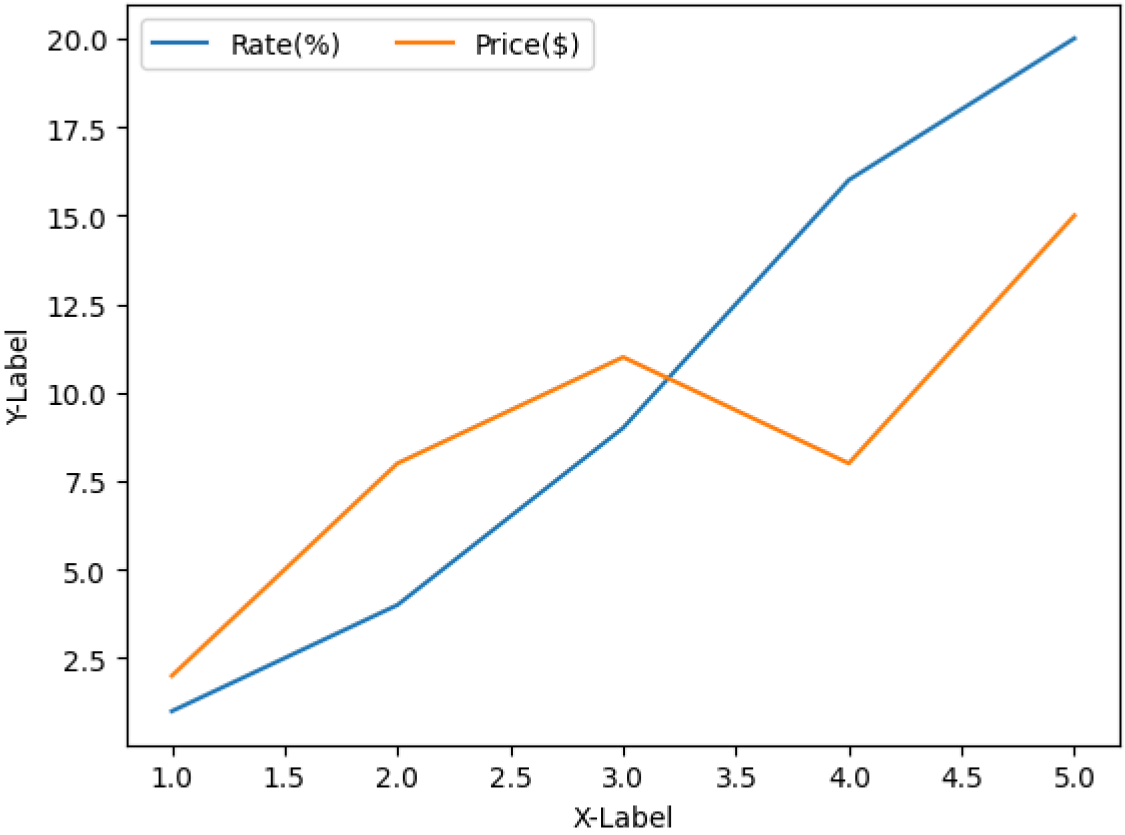
In [19]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], label='Rate(%)')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
# plt.legend(loc =(0.0, 0.0))
# plt.legend(loc =(0.5, 0.5))
plt.legend(loc =(1.0, 1.0))
plt.show()
```



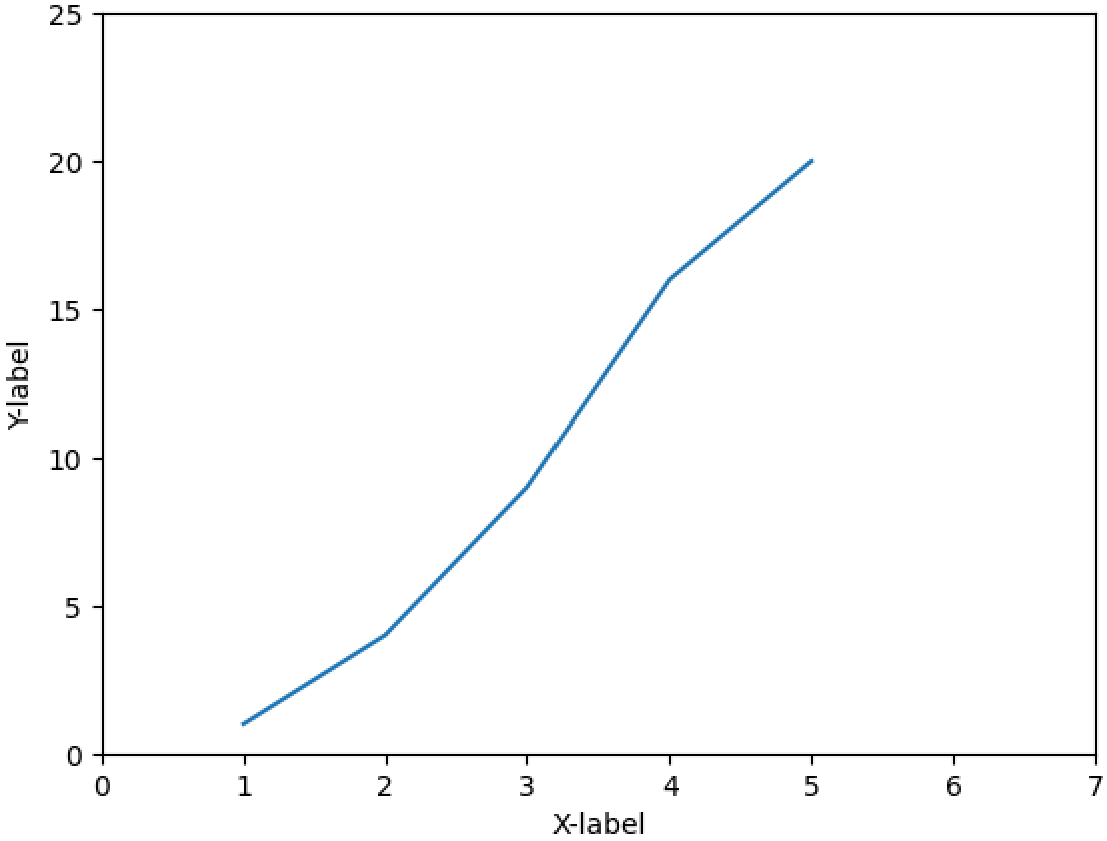
In [22]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], label='Rate(%)')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.legend(loc = 'lower right')
plt.show()
```



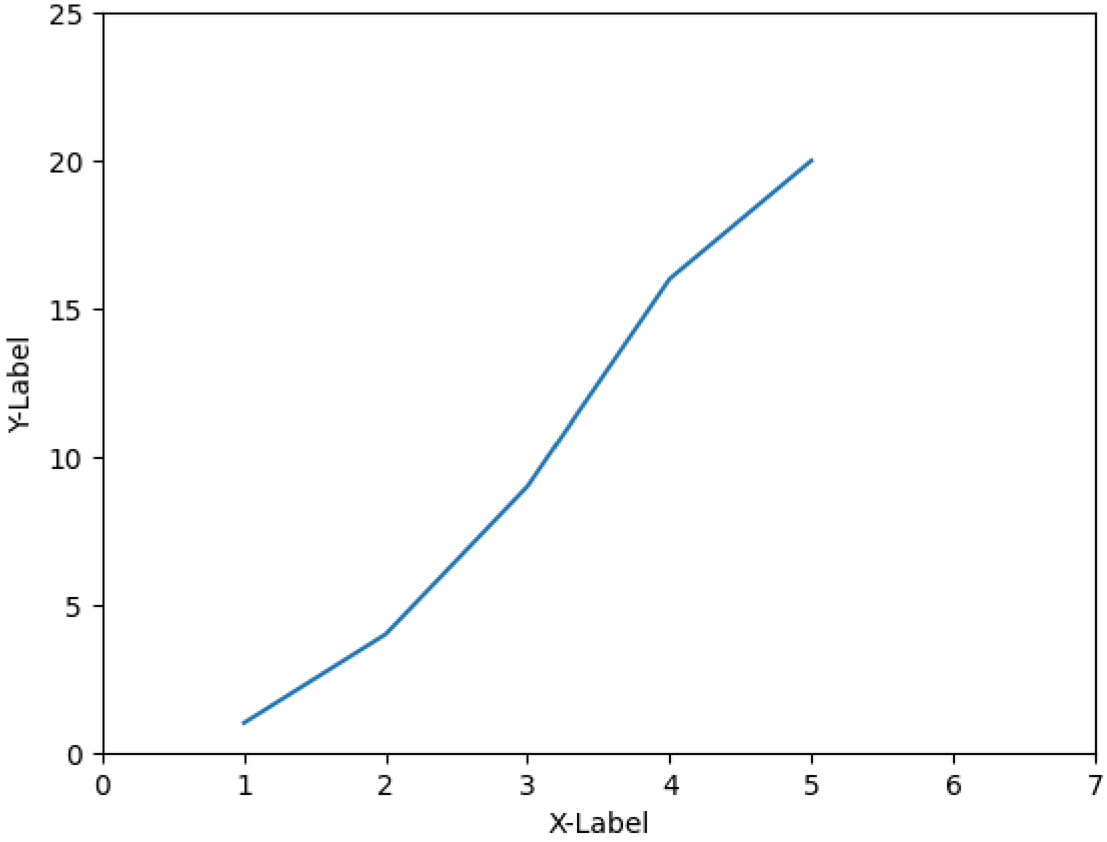
In [23]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], label='Rate(%)')
plt.plot([1, 2, 3, 4, 5], [2, 8, 11, 8, 15], label='Price($)')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.legend(ncol = 2)
plt.show()
```



In [24]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
plt.xlabel('X-label')
plt.ylabel('Y-label')
plt.xlim([0, 7]) # X축의 범위 : [x_min, x_max]
plt.ylim([0, 25]) # Y축의 범위 : [y_min, y_max]
plt.show()
```

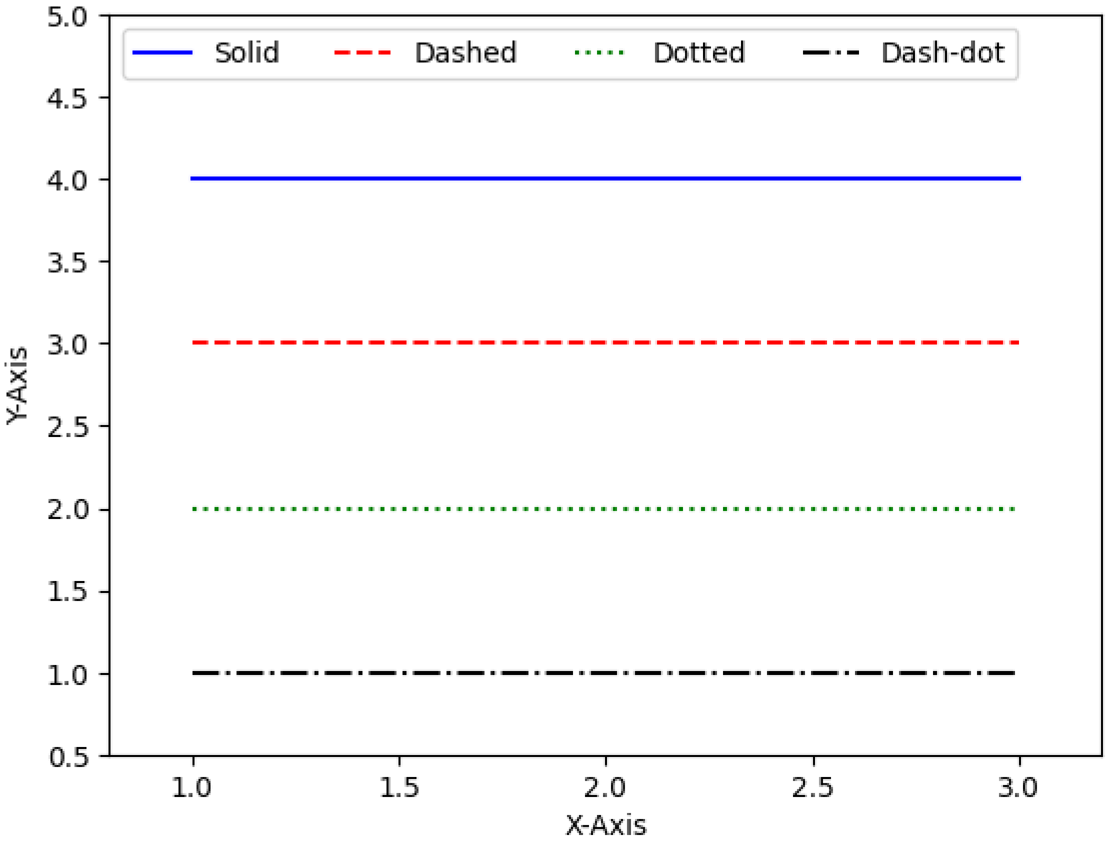


```
In [25]: plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20])
         plt.xlabel('X-Label')
         plt.ylabel('Y-Label')
         plt.axis([0, 7, 0, 25]) # [x_min, x_max, y_min, y_max]
         plt.show()
```



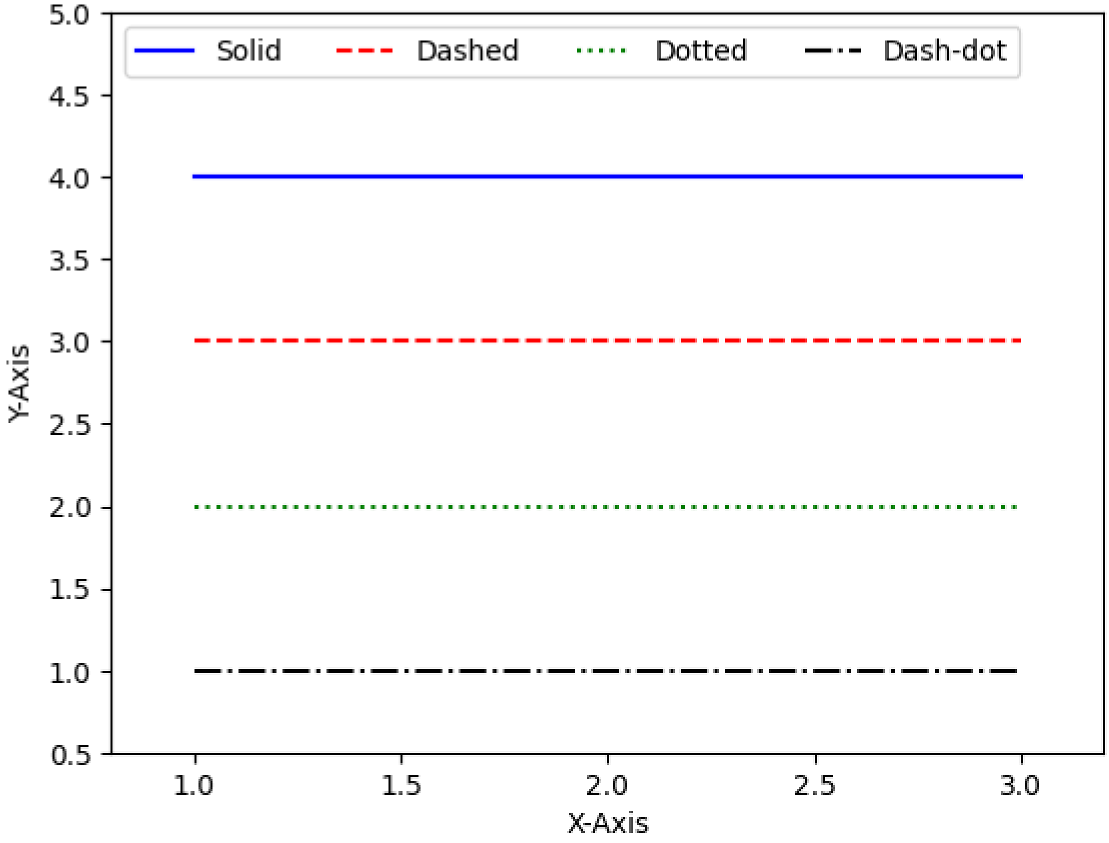
In [26]:

```
plt.plot([1, 3], [4, 4], '-', color='b', label='Solid')
plt.plot([1, 3], [3, 3], '--', color='r', label='Dashed')
plt.plot([1, 3], [2, 2], ':', color='g', label='Dotted')
plt.plot([1, 3], [1, 1], '-.', color='k', label='Dash-dot')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.axis([0.8, 3.2, 0.5, 5.0])
plt.legend(loc='upper left', ncol=4)
plt.show()
```



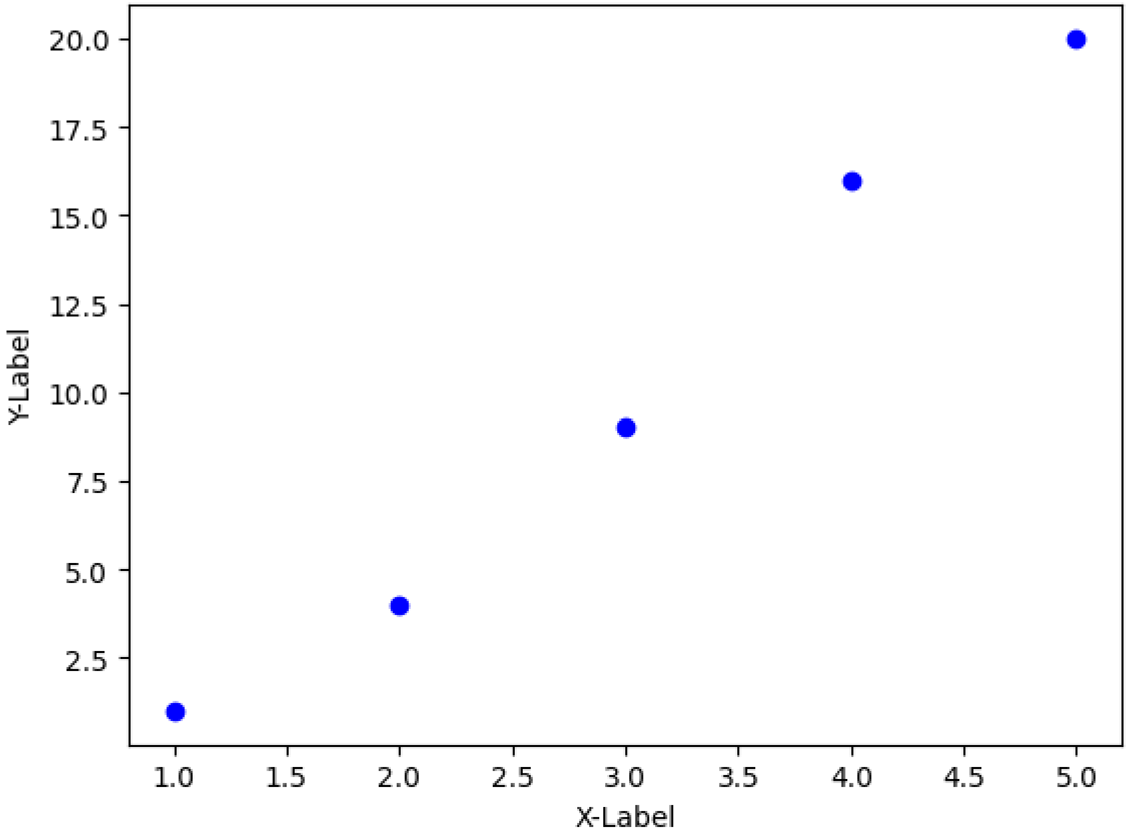
In [27]:

```
plt.plot([1, 3], [4, 4], linestyle='solid', color='b', label='Solid')
plt.plot([1, 3], [3, 3], linestyle='dashed', color='r', label='Dashed')
plt.plot([1, 3], [2, 2], linestyle='dotted', color='g', label='Dotted')
plt.plot([1, 3], [1, 1], linestyle='dashdot', color='k', label='Dash-dot')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.axis([0.8, 3.2, 0.5, 5.0])
plt.legend(loc='upper left', ncol=4)
plt.show()
```



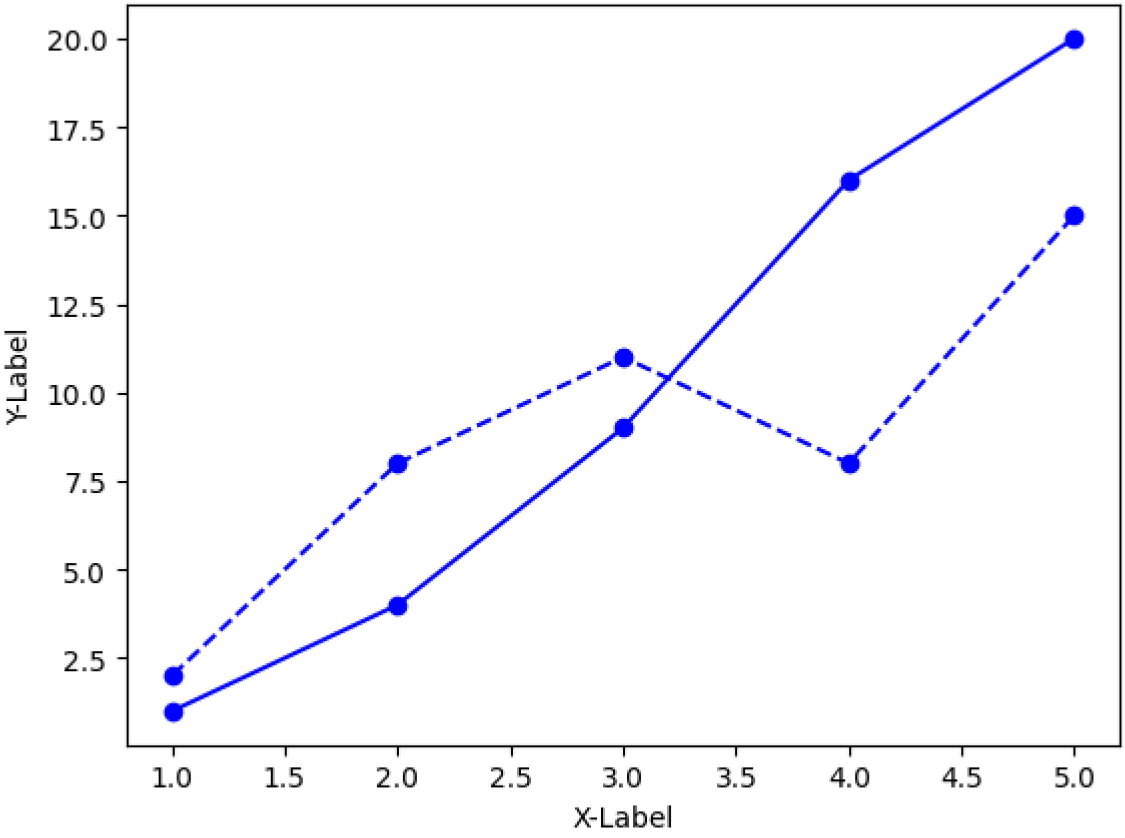
In [28]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'bo')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.show()
```



In [30]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'bo-')
plt.plot([1, 2, 3, 4, 5], [2, 8, 11, 8, 15], 'bo--')
plt.xlabel('X-Label')
plt.ylabel('Y-Label')
plt.show()
```

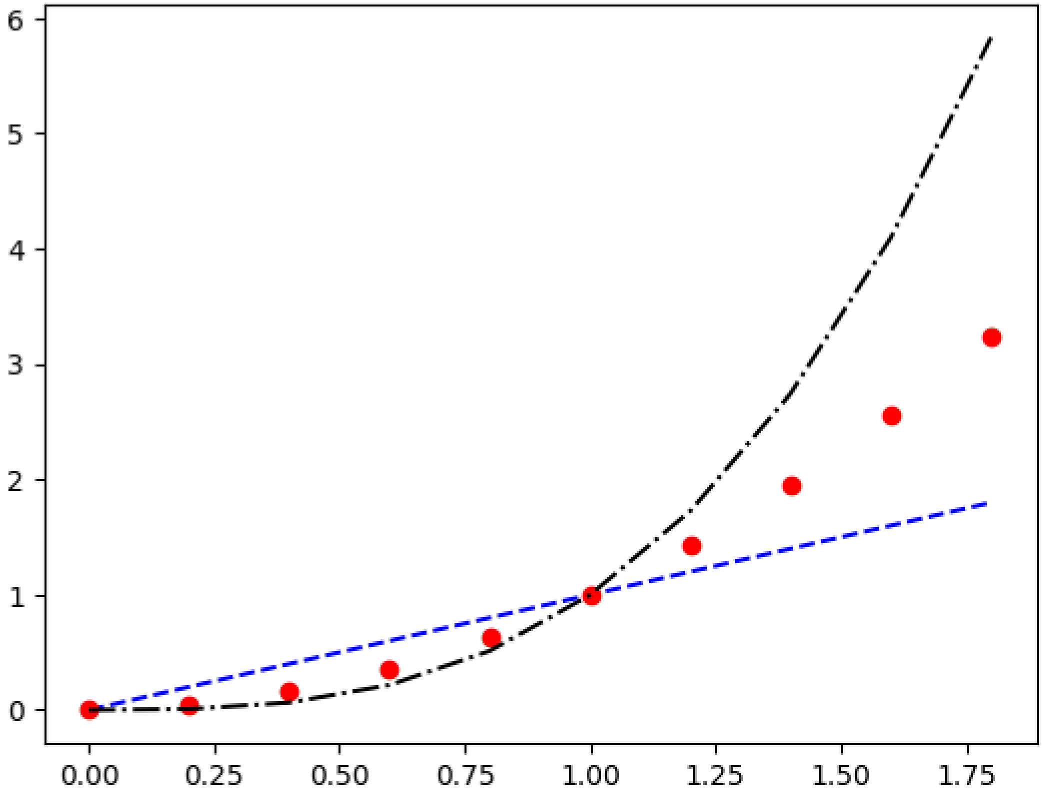


In [ ]:

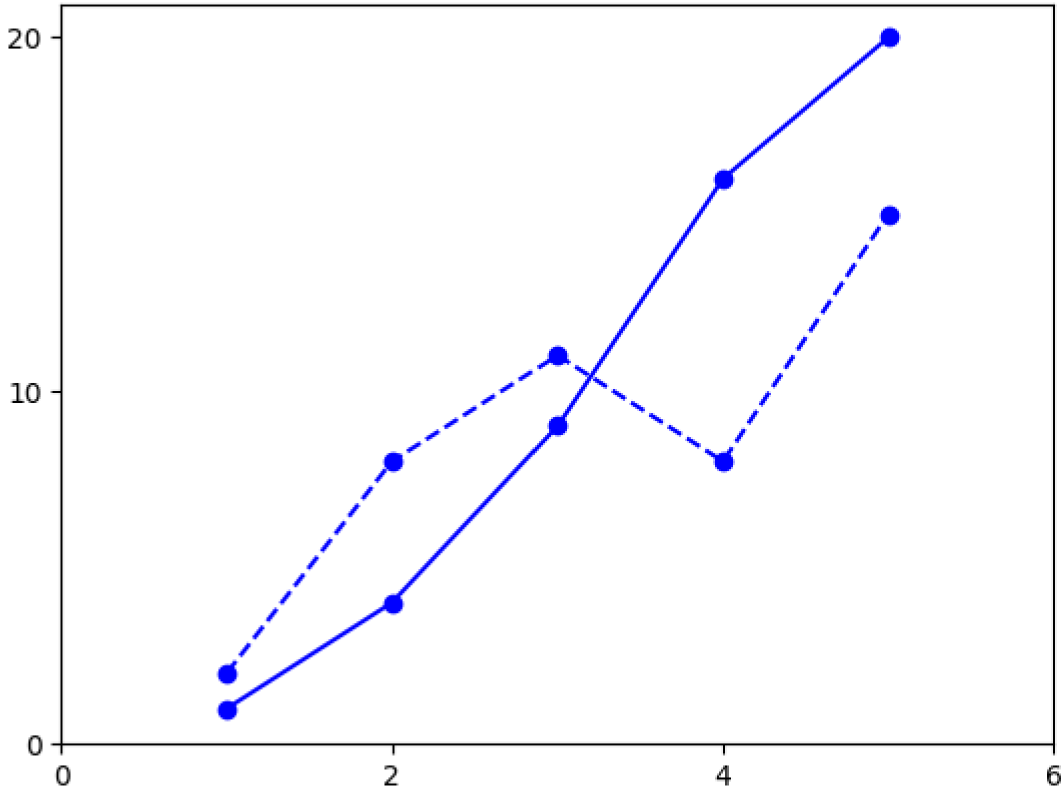
```
plt.plot([3.0, 3.0, 3.0], marker="h")
plt.plot([2.5, 2.5, 2.5], marker="+")
plt.plot([2.0, 2.0, 2.0], marker="x")
plt.plot([1.5, 1.5, 1.5], marker=11)
plt.plot([1.0, 1.0, 1.0], marker='$Y$')
plt.show()
```

In [32]:

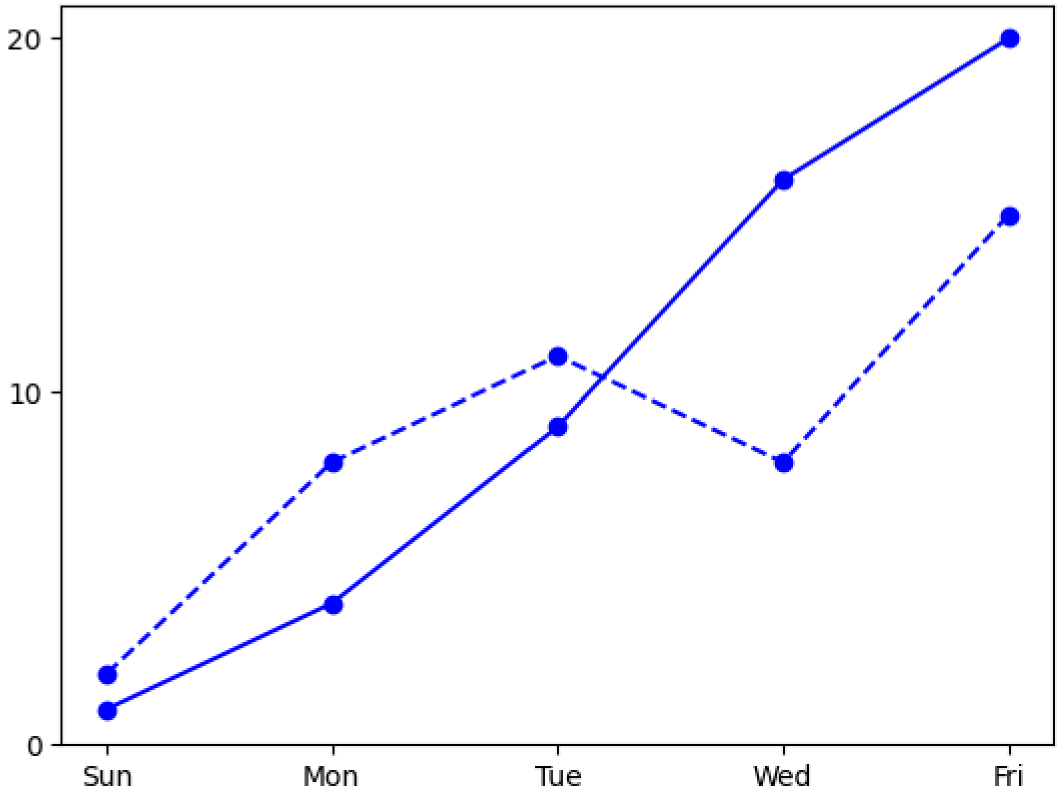
```
import numpy as np
x = np.arange(0, 2, 0.2)
plt.plot(x, x, 'b--', x, x**2, 'ro', x, x**3, 'k-.')
plt.show()
```



In [33]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'bo-')
plt.plot([1, 2, 3, 4, 5], [2, 8, 11, 8, 15], 'bo--')
plt.xticks([0, 2, 4, 6])
plt.yticks([0,10,20])
plt.show()
```

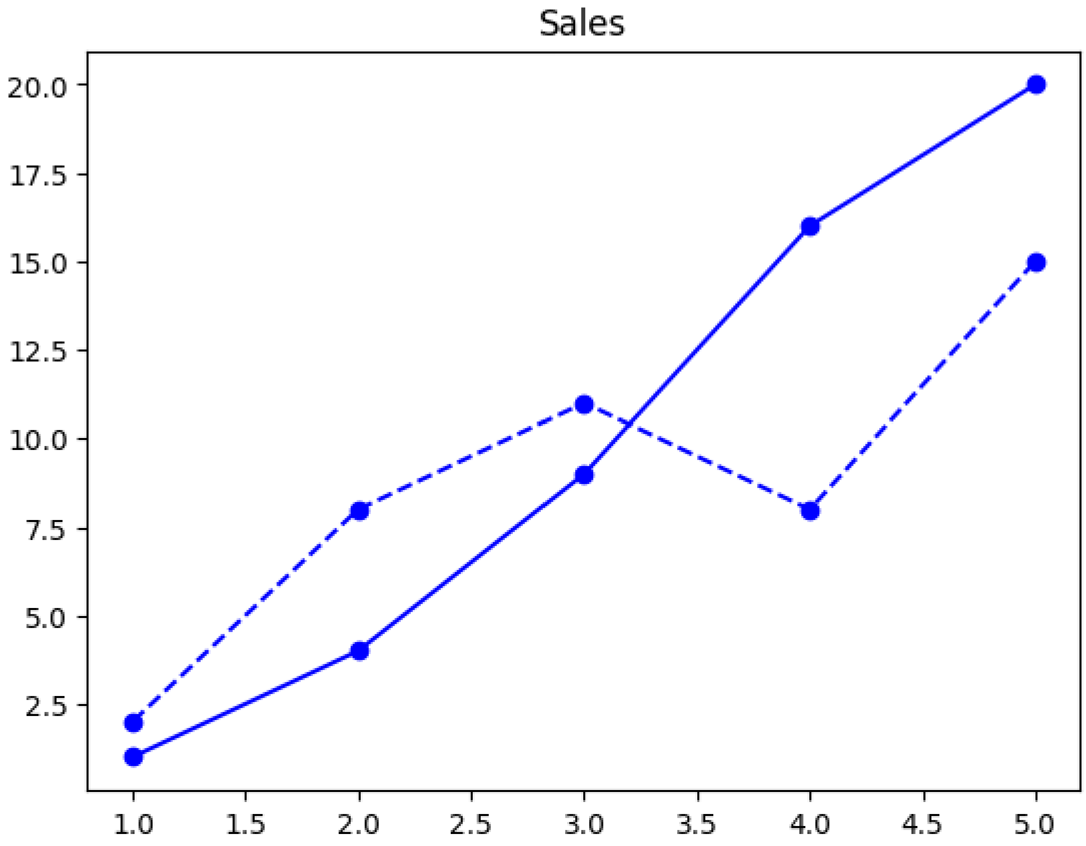


In [34]:
```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'bo-')
plt.plot([1, 2, 3, 4, 5], [2, 8, 11, 8, 15], 'bo--')
plt.xticks([1, 2, 3, 4, 5], labels=['Sun', 'Mon', 'Tue', 'Wed', 'Fri'])
plt.yticks([0,10,20])
plt.show()
```



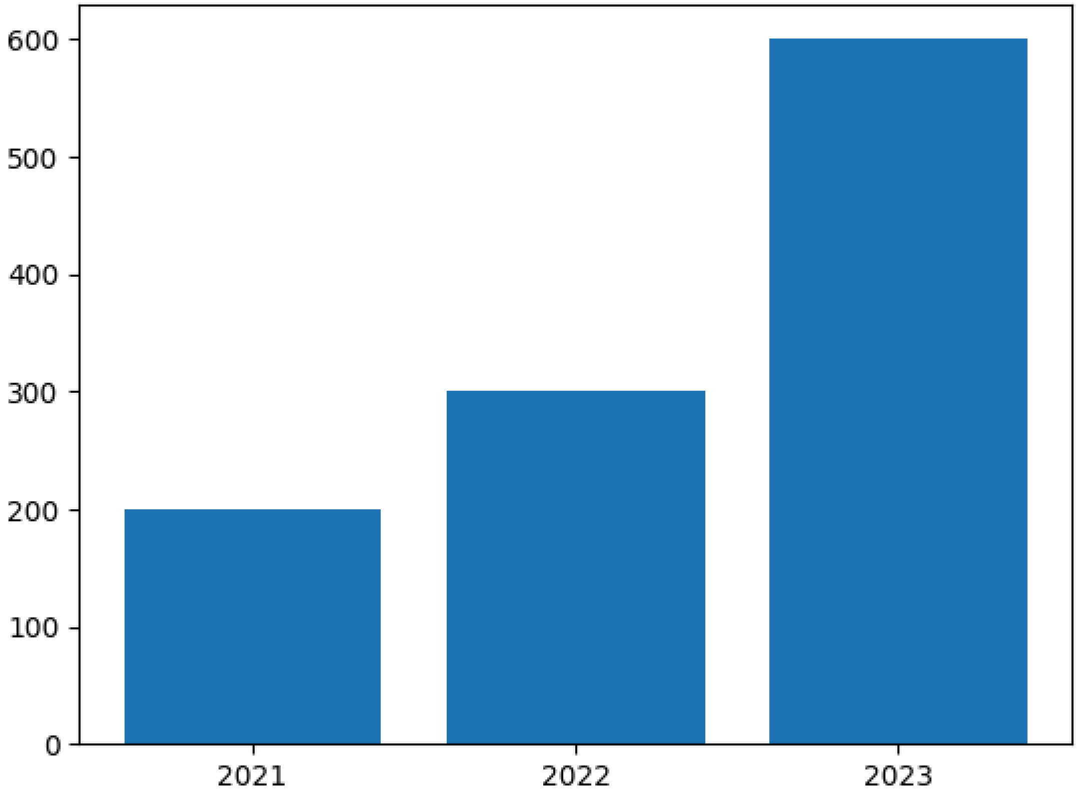
In [35]:

```
plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 20], 'bo-')
plt.plot([1, 2, 3, 4, 5], [2, 8, 11, 8, 15], 'bo--')
plt.title('Sales')
plt.show()
```



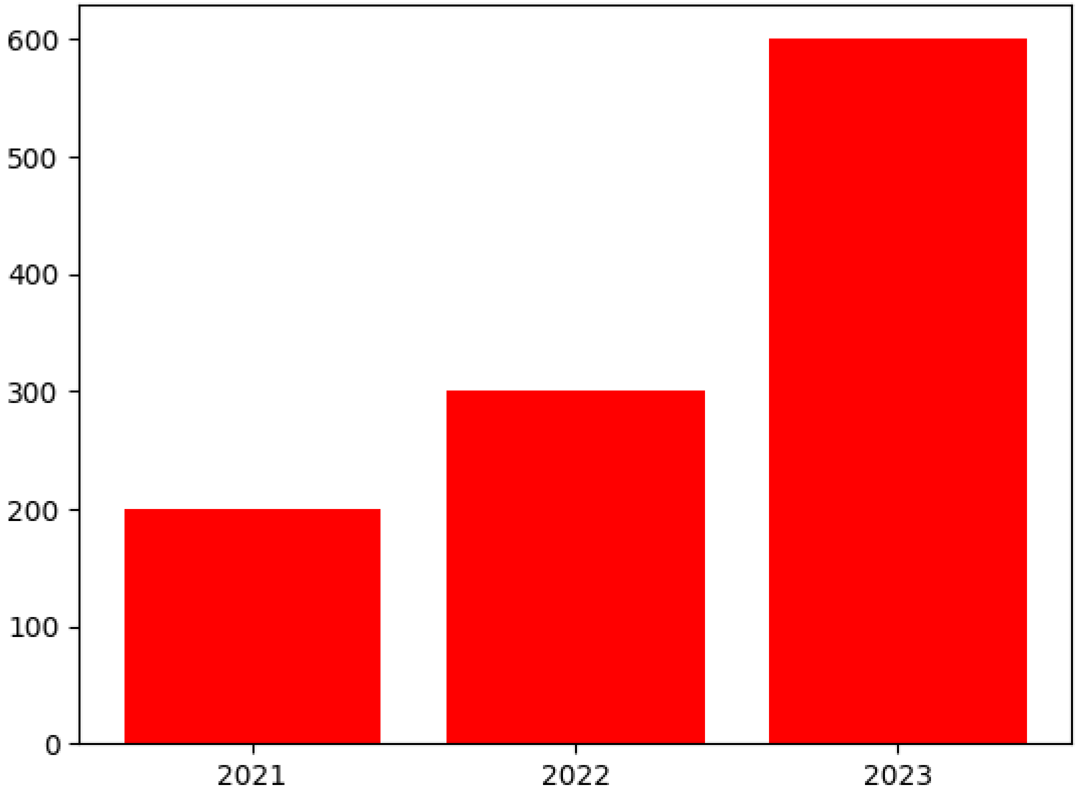
In [36]:

```
x =[1, 2, 3]
values = [200, 300, 600]
years = ['2021', '2022', '2023']
plt.bar(x, values)
plt.xticks(x, years)
plt.show()
```



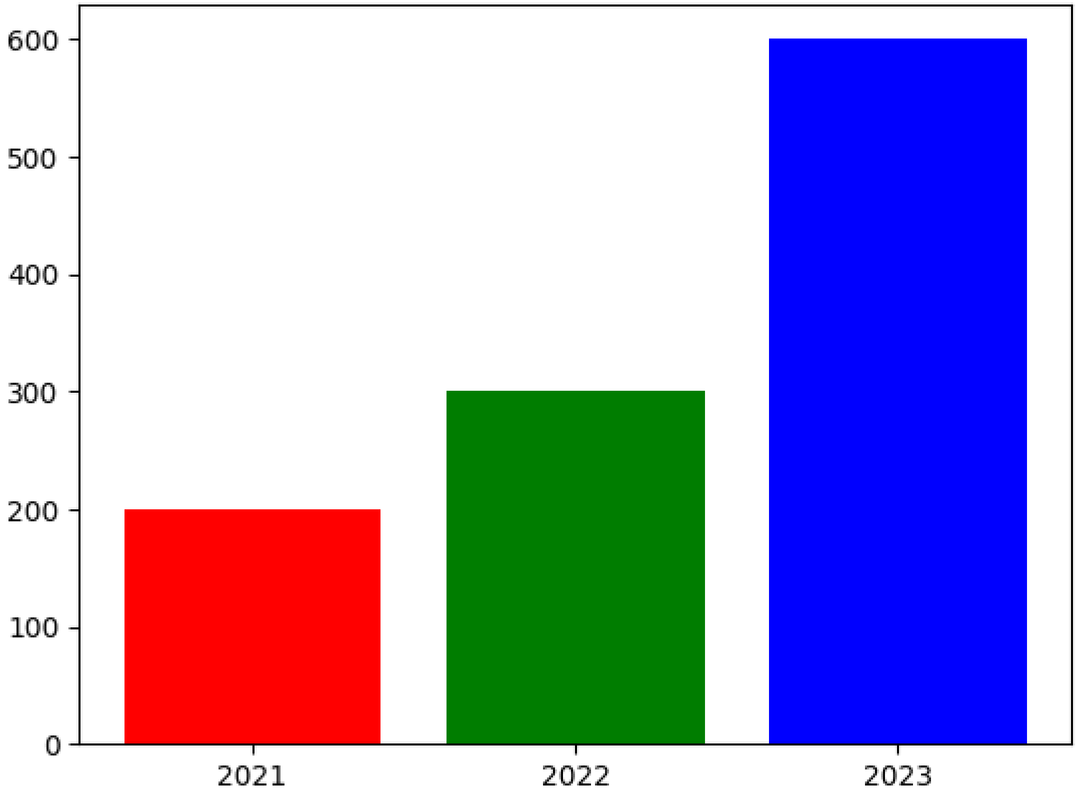
In [37]:

```
x = [1, 2, 3]
values = [200, 300, 600]
years = ['2021', '2022', '2023']
plt.bar(x, values, color='r')
plt.xticks(x, years)
plt.show()
```



In [38]:

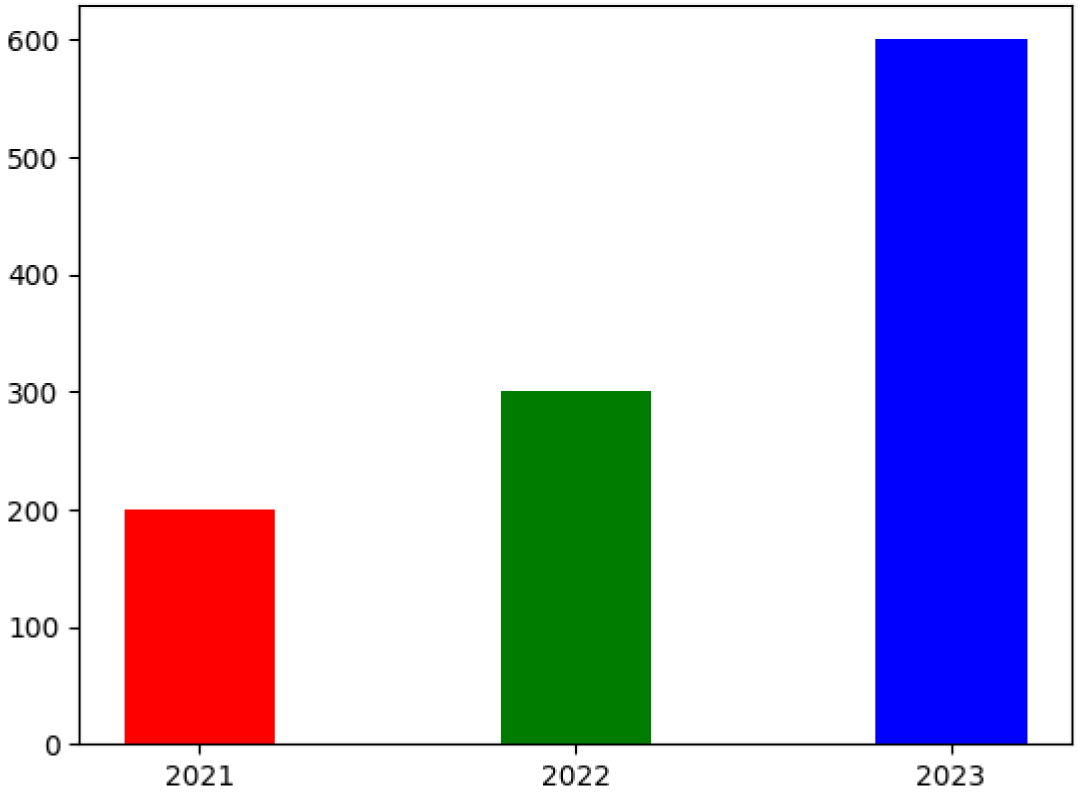
```
x = [1, 2, 3]
values = [200, 300, 600]
years = ['2021', '2022', '2023']
plt.bar(x, values, color = ['r', 'g', 'b'])
plt.xticks(x, years)
plt.show()
```



In [39]:

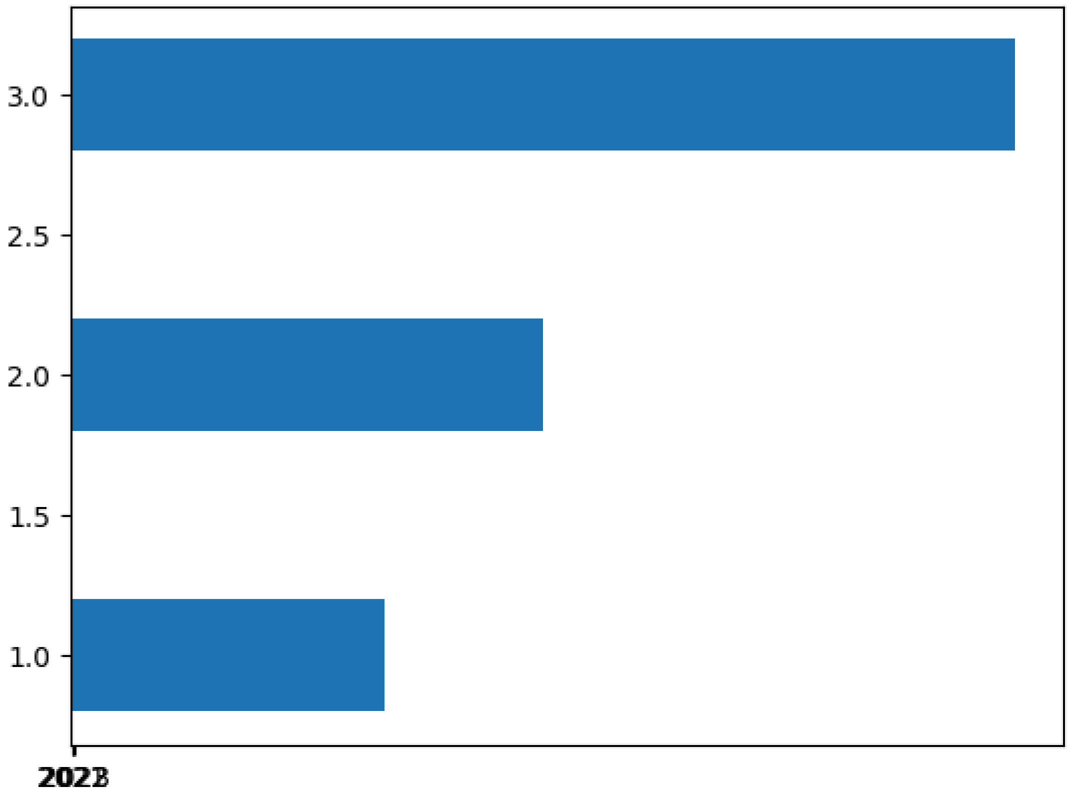
```
x = [1, 2, 3]
values = [200, 300, 600]
```

```
years = ['2021', '2022', '2023']
plt.bar(x, values, color = ['r', 'g', 'b'], width = .4)
plt.xticks(x, years)
plt.show()
```



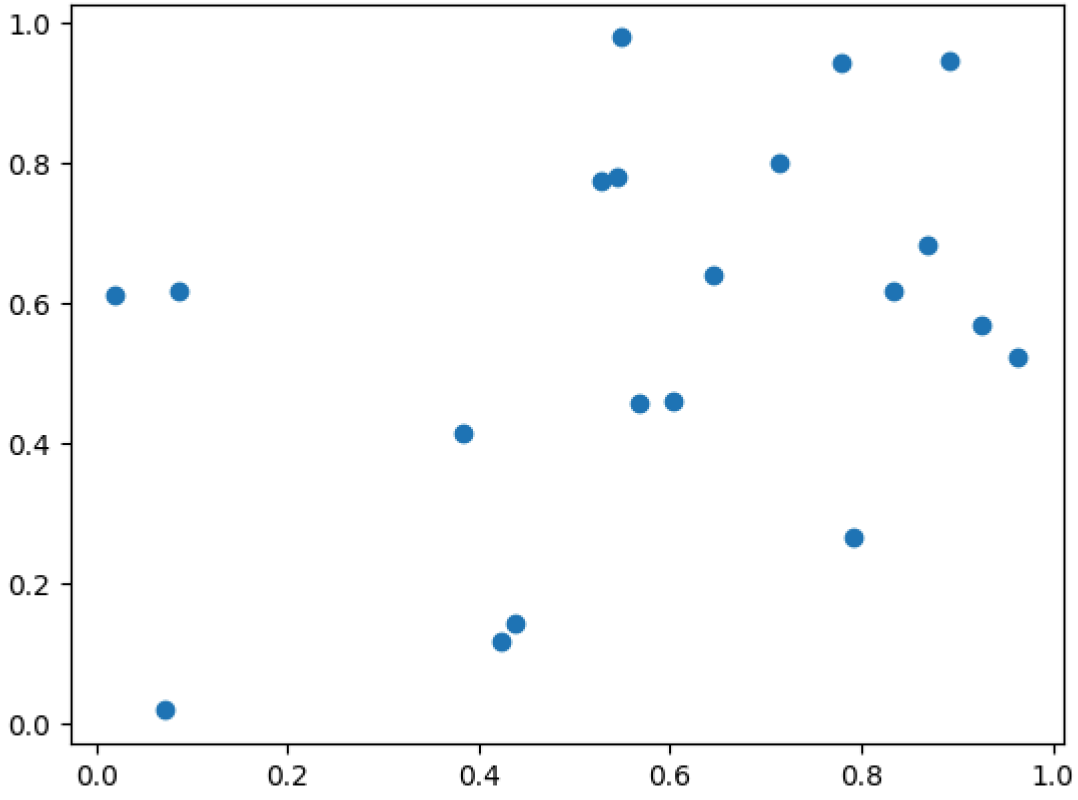
In [40]:

```
x = [1, 2, 3]
values = [200, 300, 600]
years = ['2021', '2022', '2023']
plt.barh(x, values, height=.4)
plt.xticks(x, years)
plt.show()
```



In [41]:

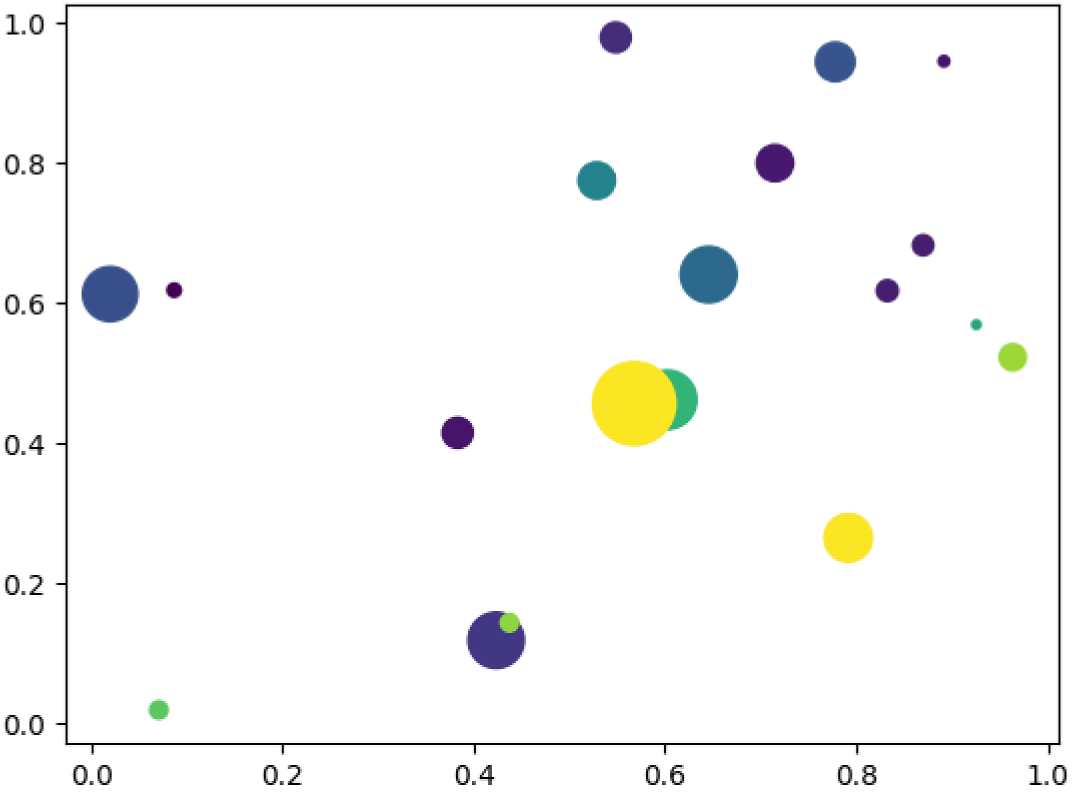
```
np.random.seed(0)
n = 20
x = np.random.rand(n)
y = np.random.rand(n)
plt.scatter(x, y)
plt.show()
```



In [42]:

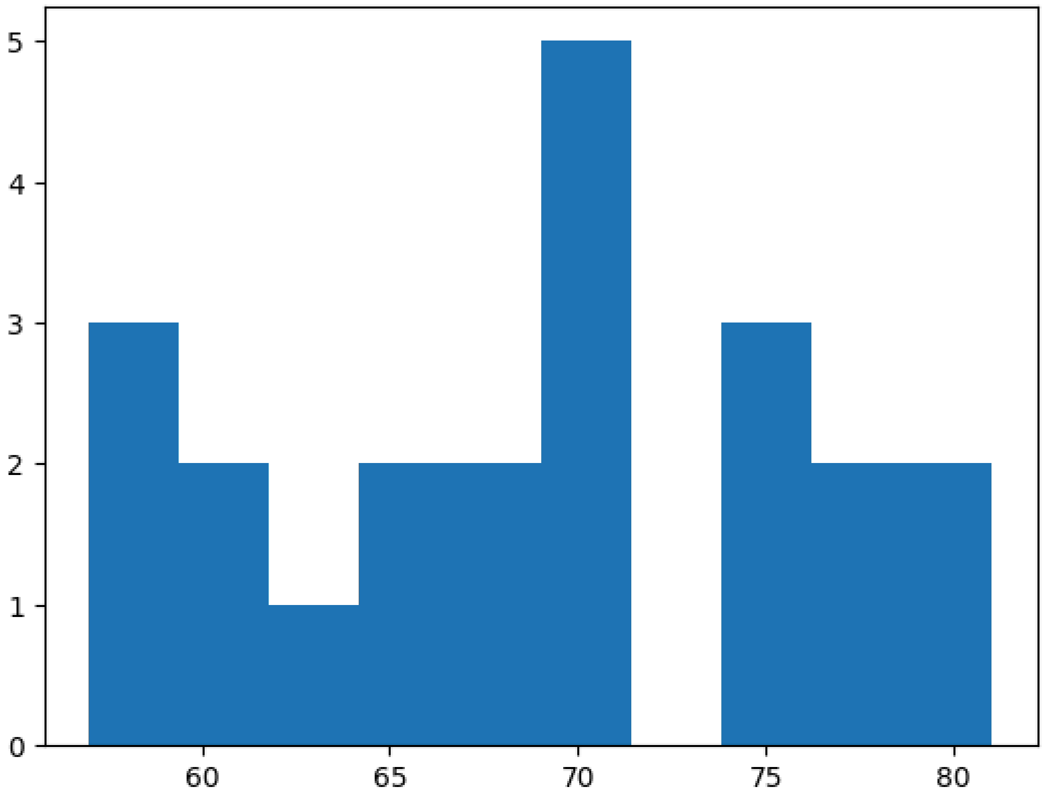
```
np.random.seed(0)
n = 20
```

```
x = np.random.rand(n)
y = np.random.rand(n)
area = (30 * np.random.rand(n))**2
colors = np.random.rand(n)
plt.scatter(x, y, s=area, c=colors)
plt.show()
```



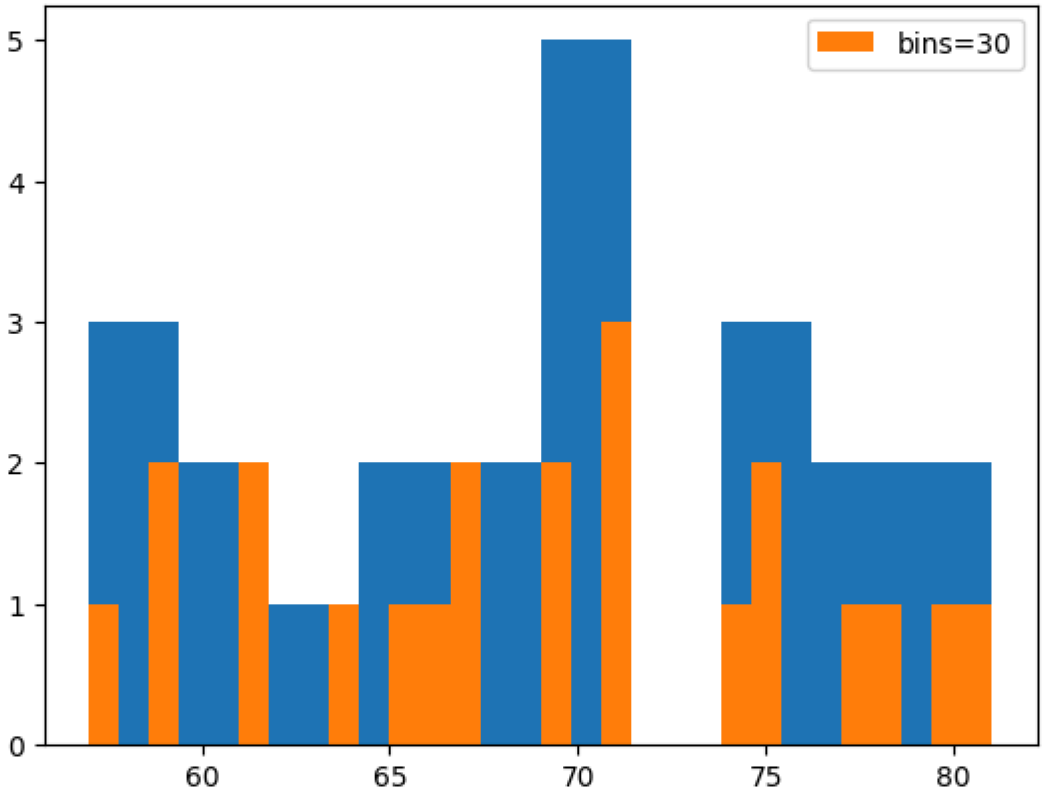
In [ ]:

```
weight =  [65, 80, 61, 59, 77, 74, 61, 75, 67, 66, 59, 69, 81, 57, 64, 78, 71, 7
plt.hist(weight)
plt.show()
```



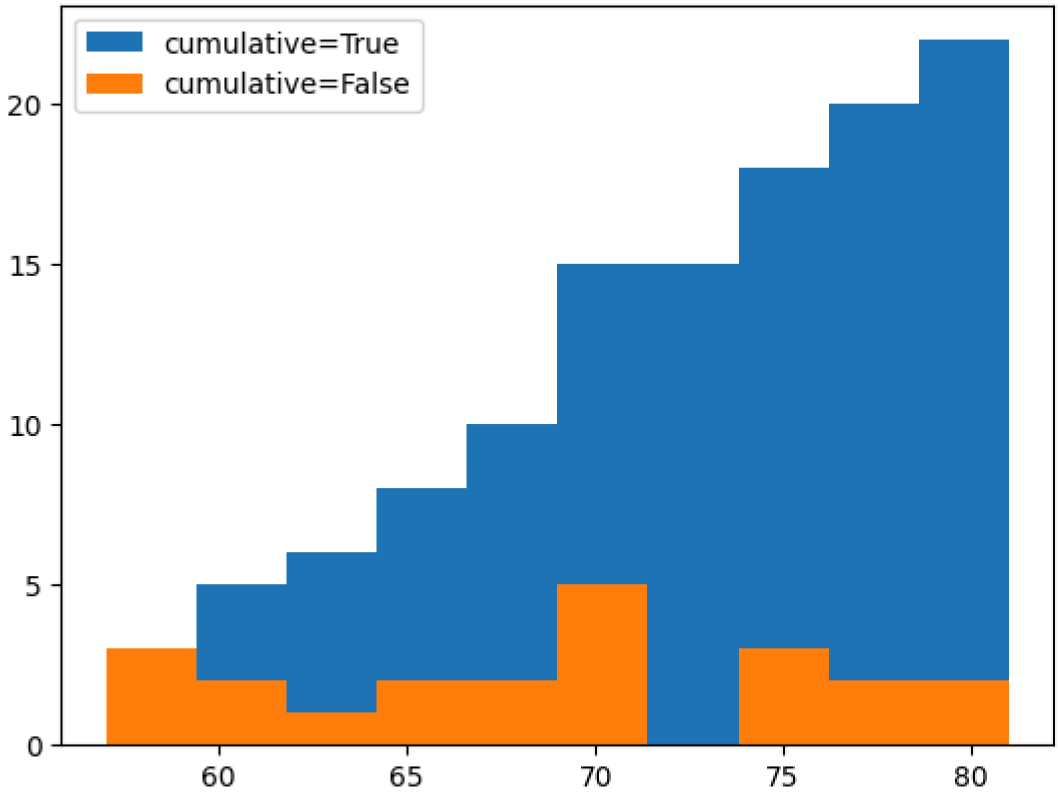
In [44]:

```
weight =  [65, 80, 61, 59, 77, 74, 61, 75, 67, 66, 59, 69, 81, 57, 64, 78, 71, 7
plt.hist(weight)
plt.hist(weight, bins=30, label='bins=30')
plt.legend()
plt.show()
```



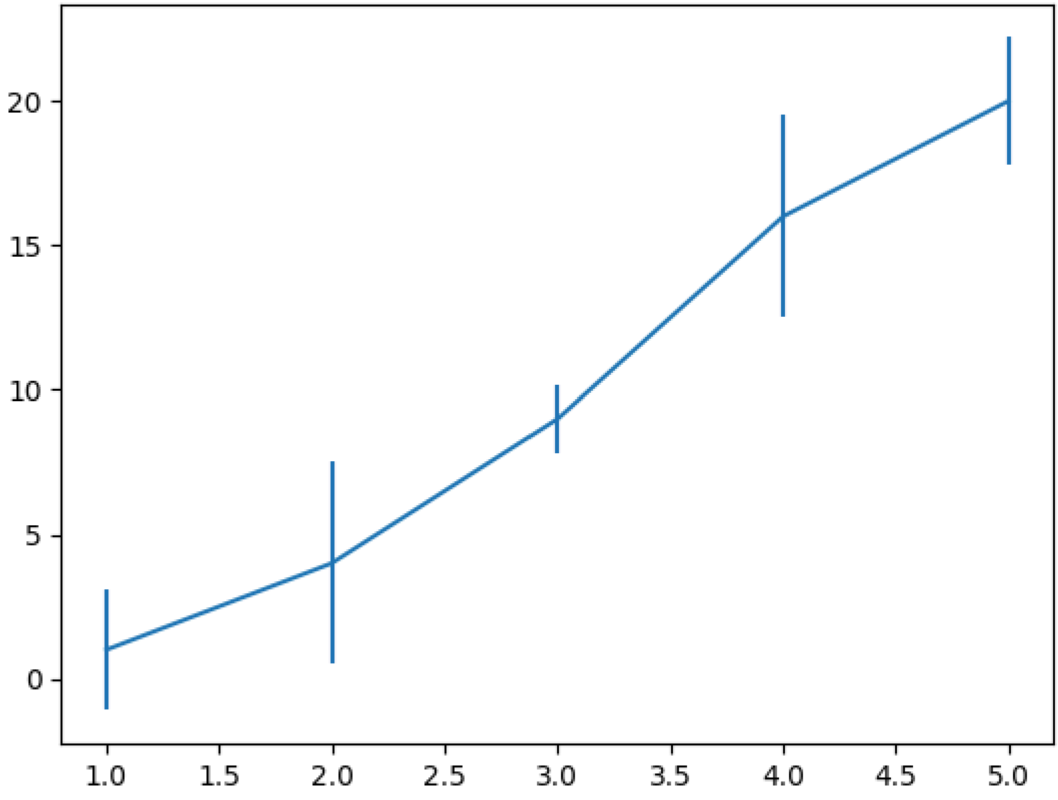
In [45]:

```
weight = [65, 80, 61, 59, 77, 74, 61, 75, 67, 66, 59, 69, 81, 57,
64, 78, 71, 71, 71, 75, 69, 67]
plt.hist(weight, cumulative=True, label='cumulative=True')
plt.hist(weight, cumulative=False, label='cumulative=False')
plt.legend(loc='upper left')
plt.show()
```

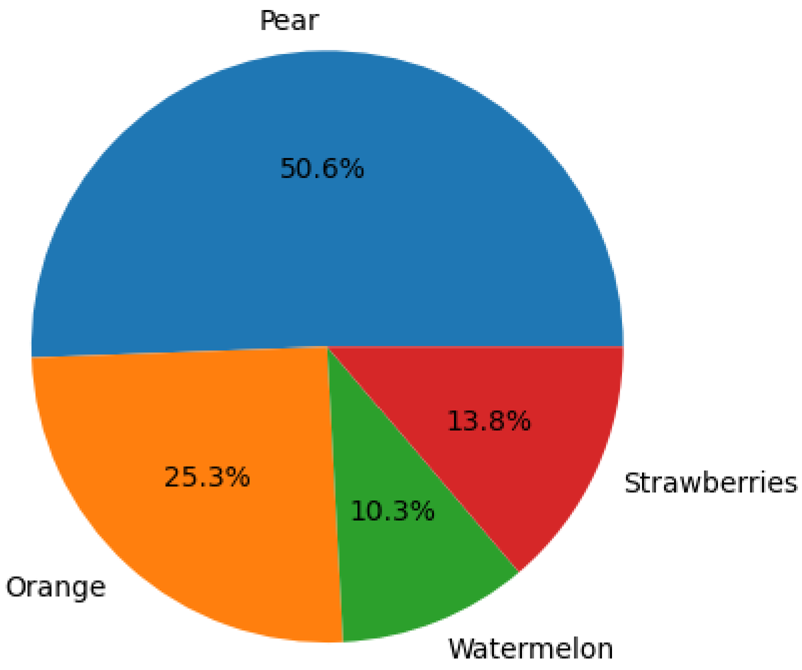


In [46]:

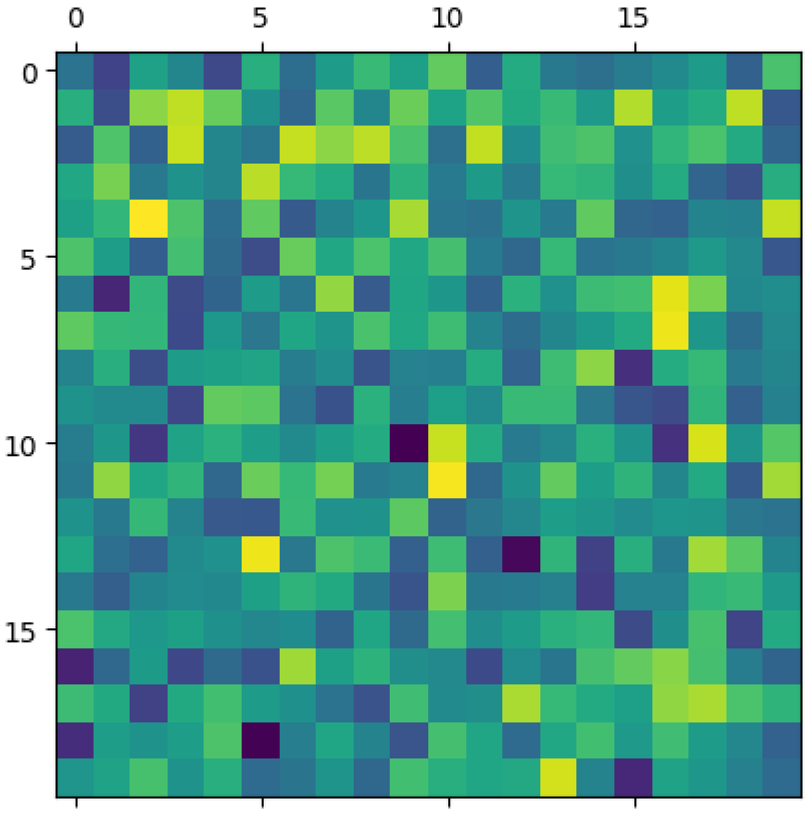
```
x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 20]
yerr = [2.1, 3.5, 1.2, 3.5, 2.2]
plt.errorbar(x, y, yerr=yerr)
plt.show()
```



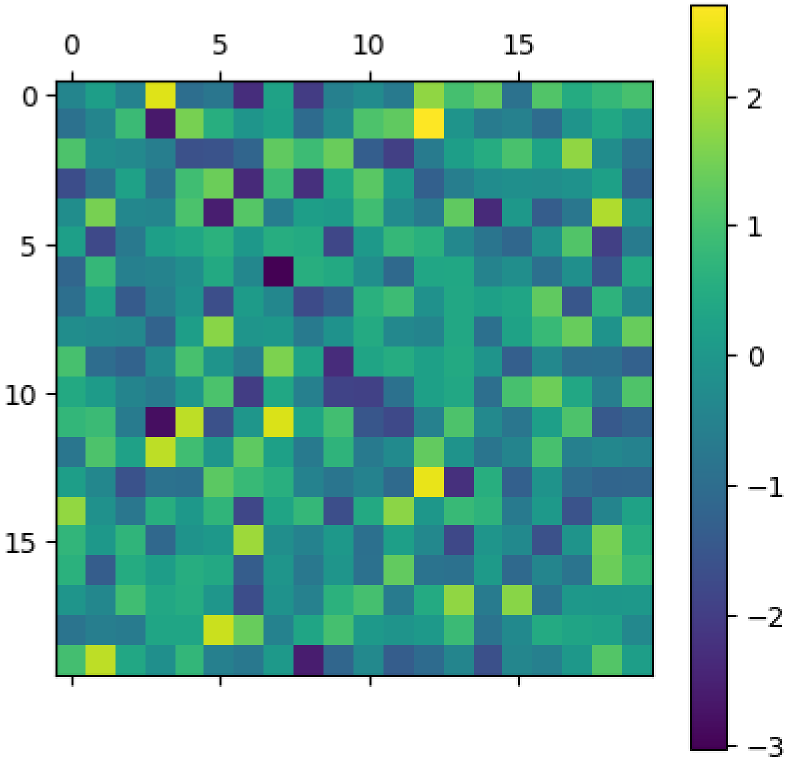
In [ ]:
```
ratio = [44, 22, 9, 12]
labels = ['Pear', 'Orange', 'Watermelon', 'Strawberries']
plt.pie(ratio, labels=labels, autopct='%.1f%%')
plt.show()
```



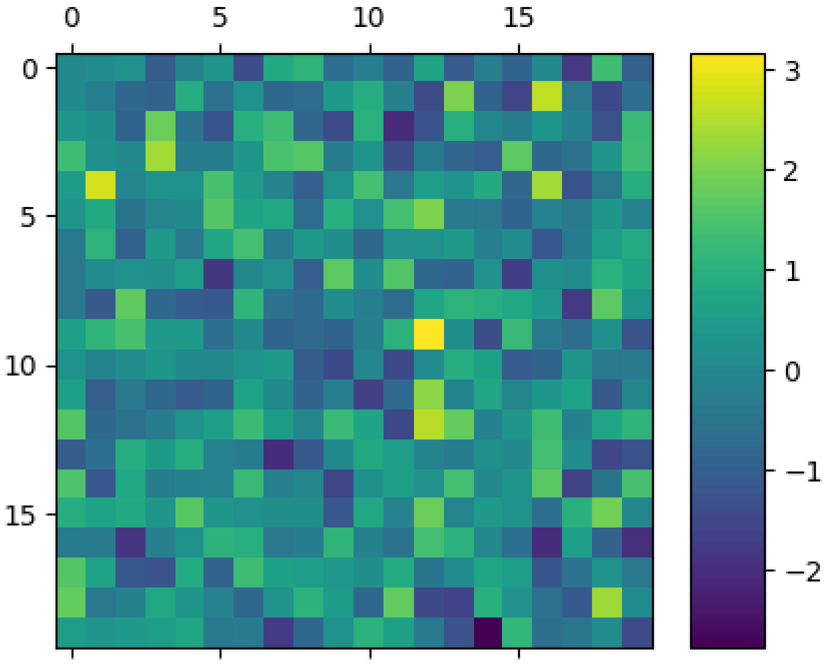
In [51]:
```
arr = np.random.standard_normal((20, 20))
plt.matshow(arr)
plt.show()
```



In [52]:
```
arr = np.random.standard_normal((20, 20))
plt.matshow(arr)
plt.colorbar()
plt.show()
```

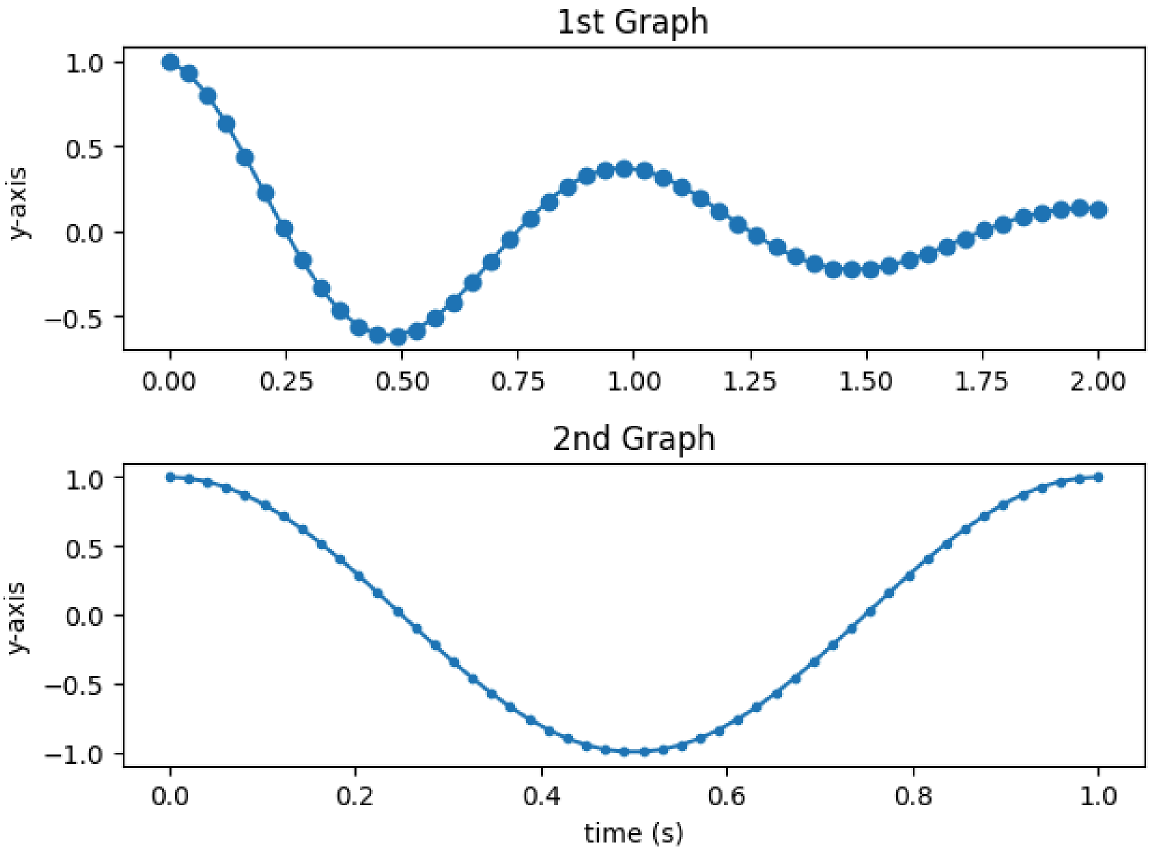


In [53]:
```
arr = np.random.standard_normal((20, 20))
plt.matshow(arr)
plt.colorbar(shrink=0.8, aspect=8)
plt.show()
```



In [54]:
```
x1 = np.linspace(0.0, 2.0)
x2 = np.linspace(0.0, 1.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)
```

```
plt.subplot(2, 1, 1) # nrows=2, ncols=1, index=1
plt.plot(x1, y1, 'o-')
plt.title('1st Graph')
plt.ylabel('y-axis')
plt.subplot(2, 1, 2) # nrows=2, ncols=1, index=2
plt.plot(x2, y2, '.-')
plt.title('2nd Graph')
plt.xlabel('time (s)')
plt.ylabel('y-axis')
plt.tight_layout()
plt.show()
```

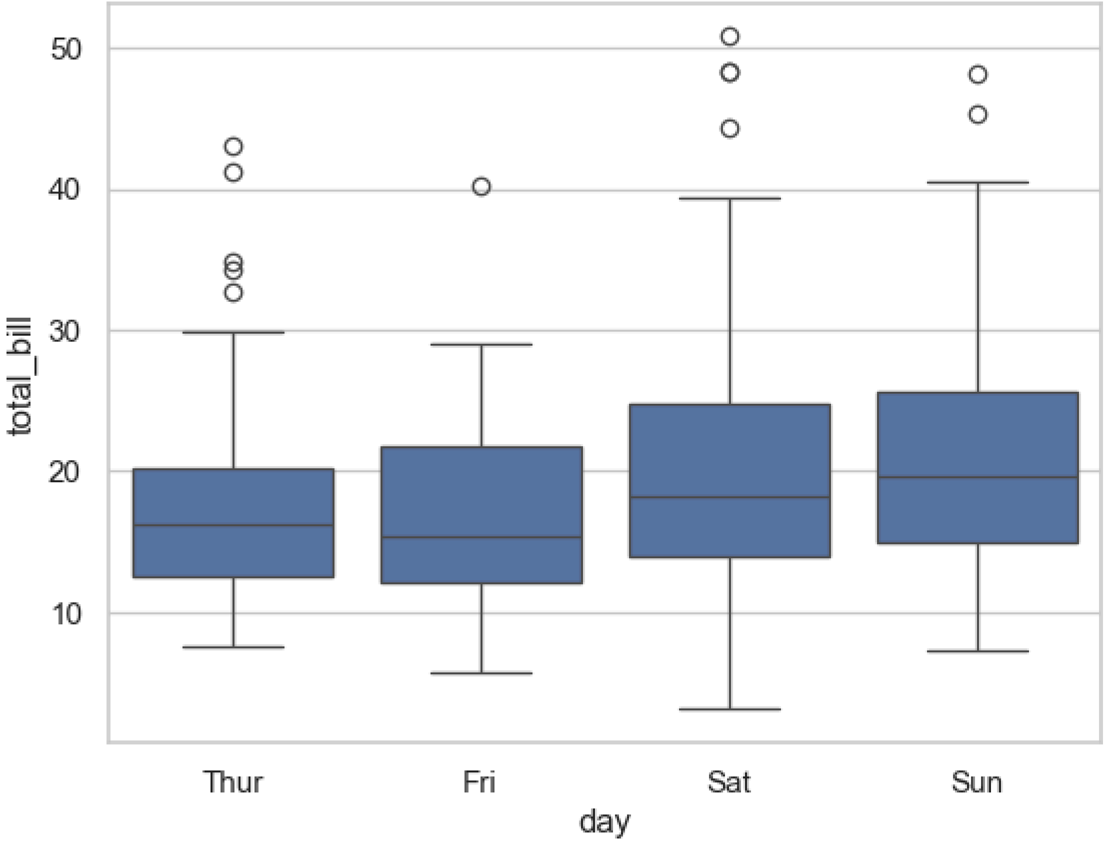


In [57]:
```
import seaborn as sns
```

In [59]:
```
sns.set(style="whitegrid")
```

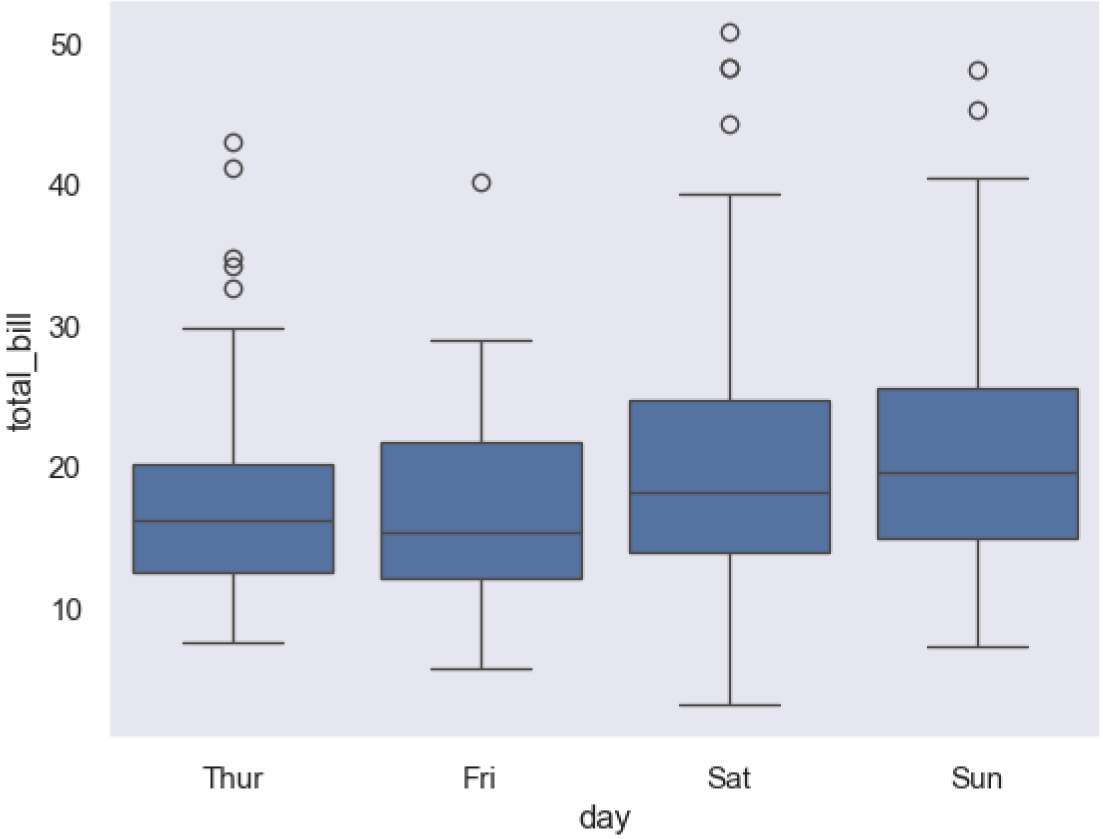
In [ ]:
```
df = sns.load_dataset("tips")
sns.boxplot(x="day", y="total_bill", data=df)
```

Out[ ]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



```
In [61]: sns.set(style="dark")
         sns.boxplot(x="day", y="total_bill", data=df)
```

```
Out[61]: <Axes: xlabel='day', ylabel='total_bill'>
```



In [62]:

```
flights = sns.load_dataset("flights")
flights
```

Out[62]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
...	...	...	...
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

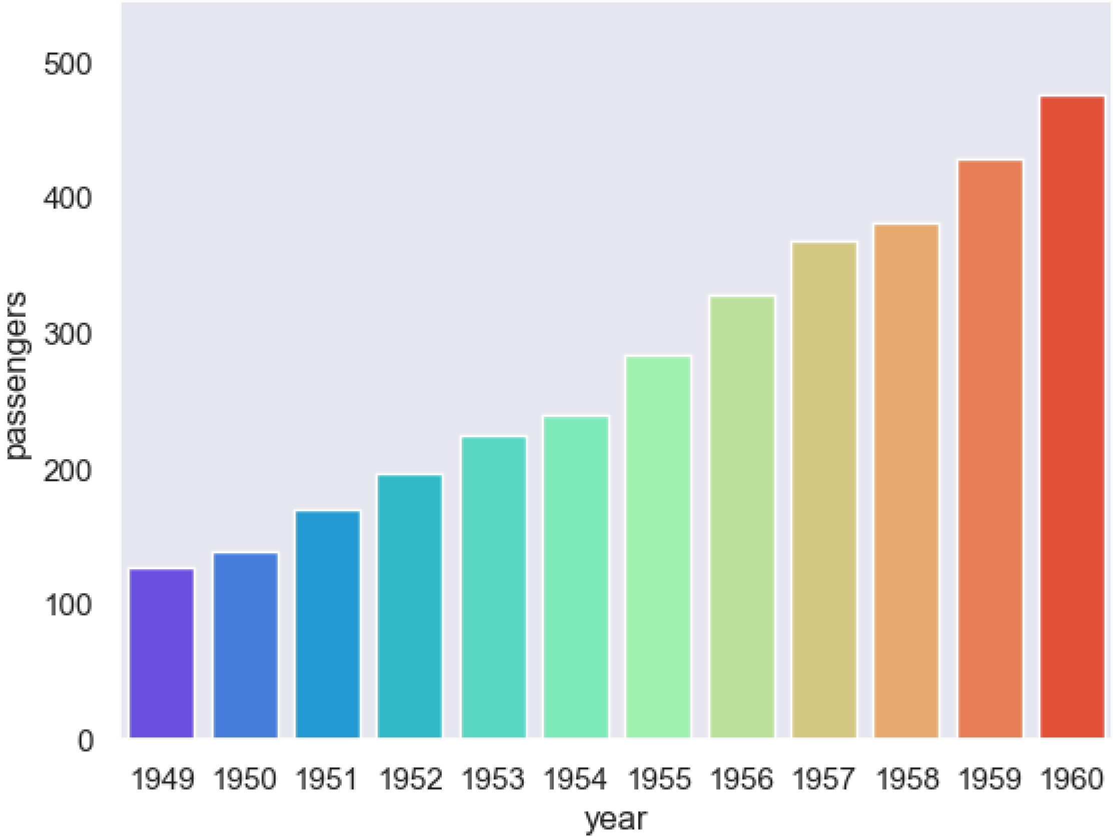
144 rows × 3 columns

In [81]:

```
import warnings
warnings.filterwarnings("ignore")
sns.barplot(data=flights, x="year", y="passengers", palette="rainbow", errwidth=
```

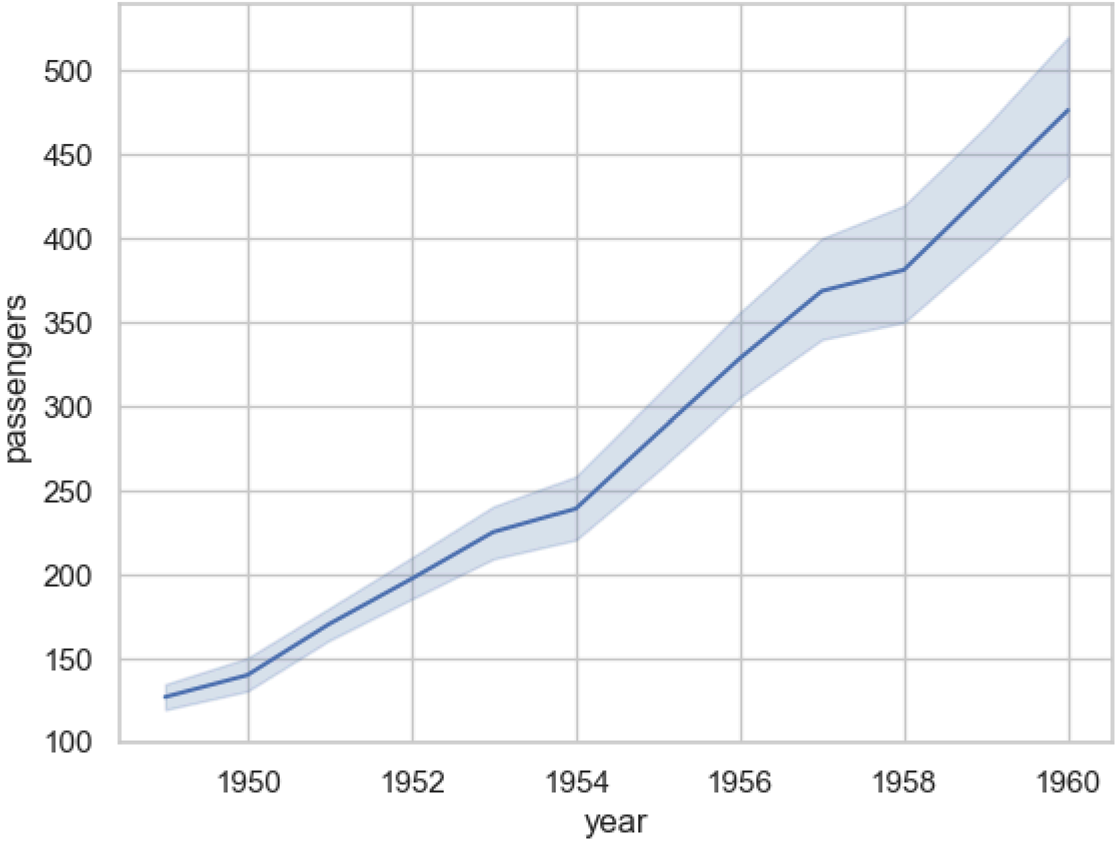
Out[81]:

```
<Axes: xlabel='year', ylabel='passengers'>
```



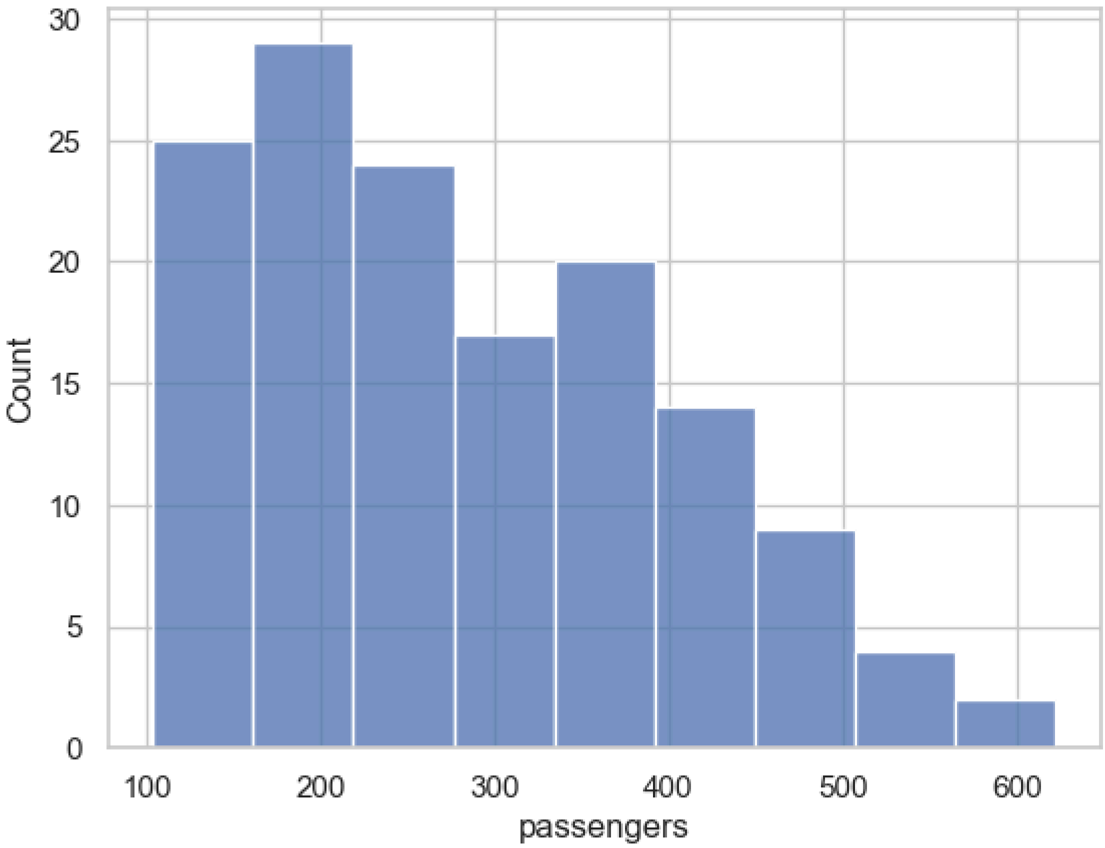
```
In [85]: #그리드 표시
         sns.set_theme(style="whitegrid")
         sns.lineplot(data=flights, x="year", y="passengers")
```

```
Out[85]: <Axes: xlabel='year', ylabel='passengers'>
```



```
In [86]: sns.histplot(flights["passengers"])
```

```
Out[86]: <Axes: xlabel='passengers', ylabel='Count'>
```



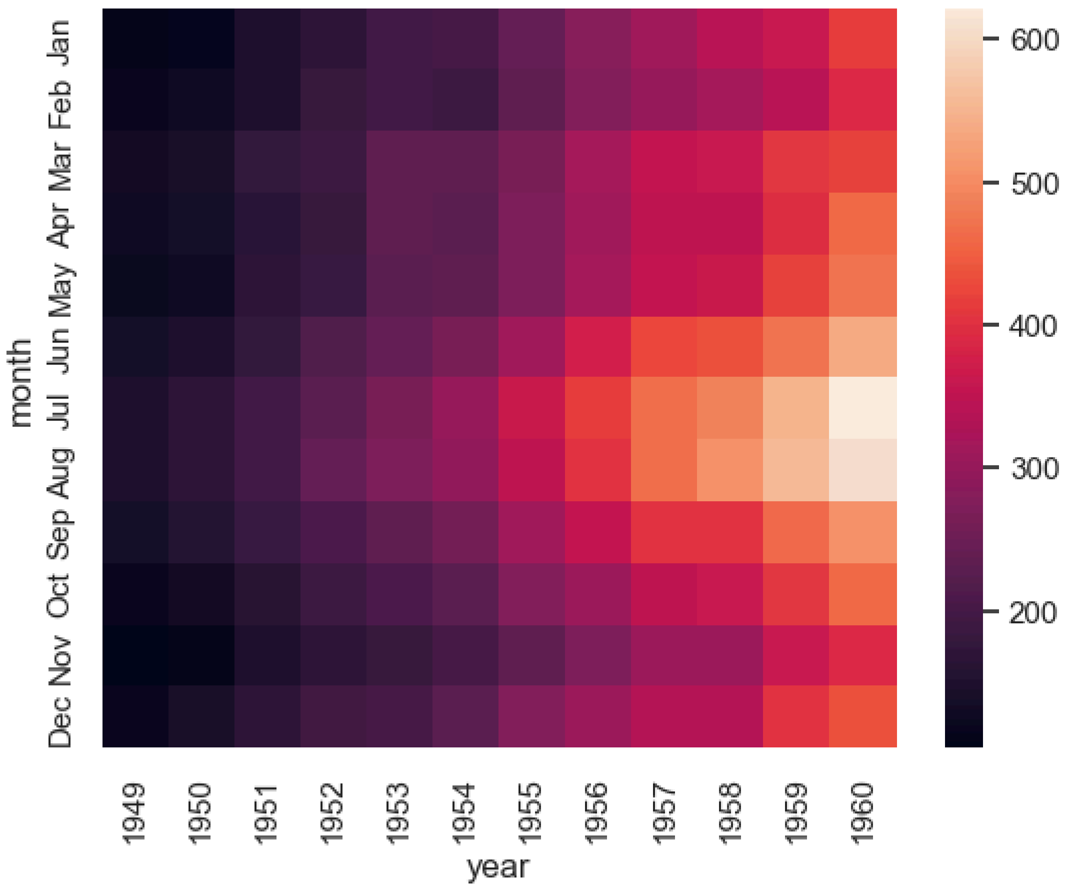
In [87]:
```
df = flights.pivot(index='month', columns='year', values='passengers')
df
```

Out[87]:

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535
Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

In [88]:
```
sns.heatmap(df)
```

Out[88]:
```
<Axes: xlabel='year', ylabel='month'>
```


month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
year
600
500
400
300
200