필요한 모듈 불러오기

In [1]:
```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
import seaborn as sns

# 주피터 노트북을 사용하는 경우에는 다음처럼 매직(magic) 명령으로 노트북 내부에 그림을 표시하도록 지정
%matplotlib inline
```

kind

문자열	의미
line	line plot (default)
bar	vertical bar plot
barh	horizontal bar plot
hist	histogram
box	boxplot
kde	Kernel Density Estimation plot
pie	pie plot
scatter	scatter plot

style

색깔

문자열	약자
blue	b
green	g
red	r
cyan	c
magenta	m
yellow	y
black	k
white	w

마커

마커	의미
.	point marker
,	pixel marker
o	circle marker
v	triangle_down marker
^	triangle_up marker
<	triangle_left marker
>	triangle_right marker
1	tri_down marker
2	tri_up marker
3	tri_left marker
4	tri_right marker
s	square marker
p	pentagon marker
*	star marker
h	hexagon1 marker
H	hexagon2 marker
+	plus marker
x	x marker
D	diamond marker
d	thin_diamond marker

선스타일

선 스타일 문자열	의미
-	실선
--	대시선
-.	점선
:	대시-점선

기타 스타일

스타일	약자	의미
color	c	선 색깔
linewidth	lw	선 굵기
linestyle	ls	선 스타일
marker		마커 종류
markersize	ms	마커 크기
markeredgecolor	mec	마커 선 색깔
markeredgewidth	mew	마커 선 굵기
markerfacecolor	mfc	마커 내부 색깔

색상: https://matplotlib.org/examples/color/named_colors.html (https://matplotlib.org/examples/color/named_colors.html)

기타 스타일: https://matplotlib.org/api/_as_gen/matplotlib.lines.Line2D.html#matplotlib.lines.Line2D (https://matplotlib.org/api/_as_gen/matplotlib.lines.Line2D.html#matplotlib.lines.Line2D)

1. line 그리기

(1) plt.plot(data,'rs--')

- 색깔(color), 마커(marker), 선 종류(line style)의 순서로 지정한다. 만약 이 중 일부가 생략되면 디폴트값이 적용

- list, series 둘다 가능

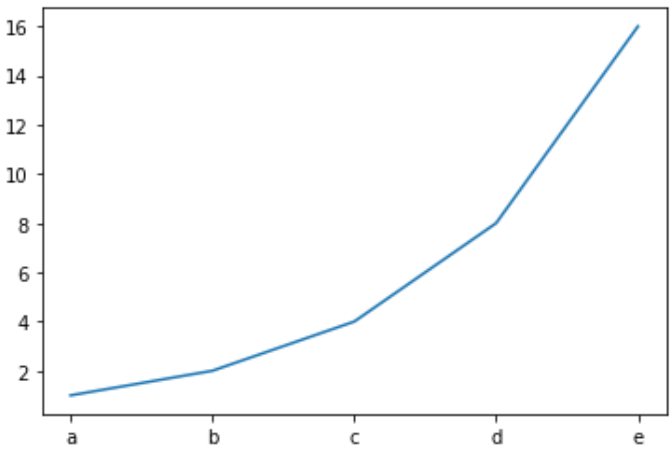
- 기본적으로 해당 자료구조의 index가 x축 value가 y축으로 들어가게됨

In [12]:
```
sr = pd.Series([1, 2, 4, 8, 16], index=['a', 'b', 'c', 'd', 'e'])
sr
```

Out[12]:
```
a     1
b     2
c     4
d     8
e    16
dtype: int64
```

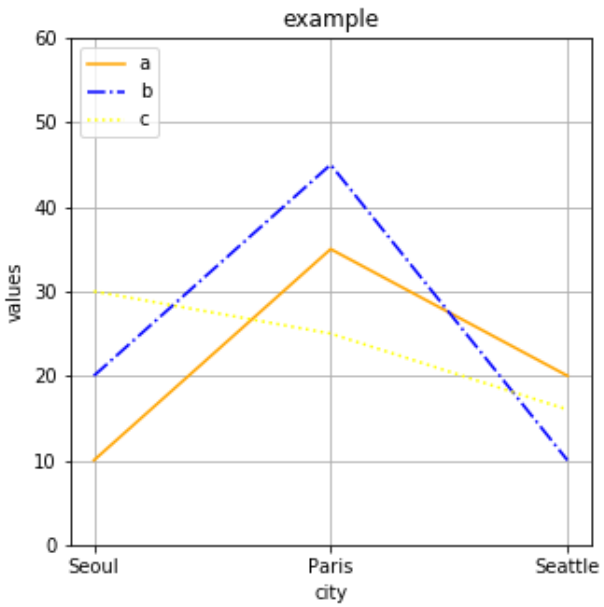
In [13]:
```
# line을 그리는 함수
plt.plot(sr)

# jupyter notebook에서는 셀마다 실행하기 때문에 필요 없지만 다른 interpreter방식에서는 적어줘야
```

Out[13]:
```
[<matplotlib.lines.Line2D at 0x1a21870210>]
```

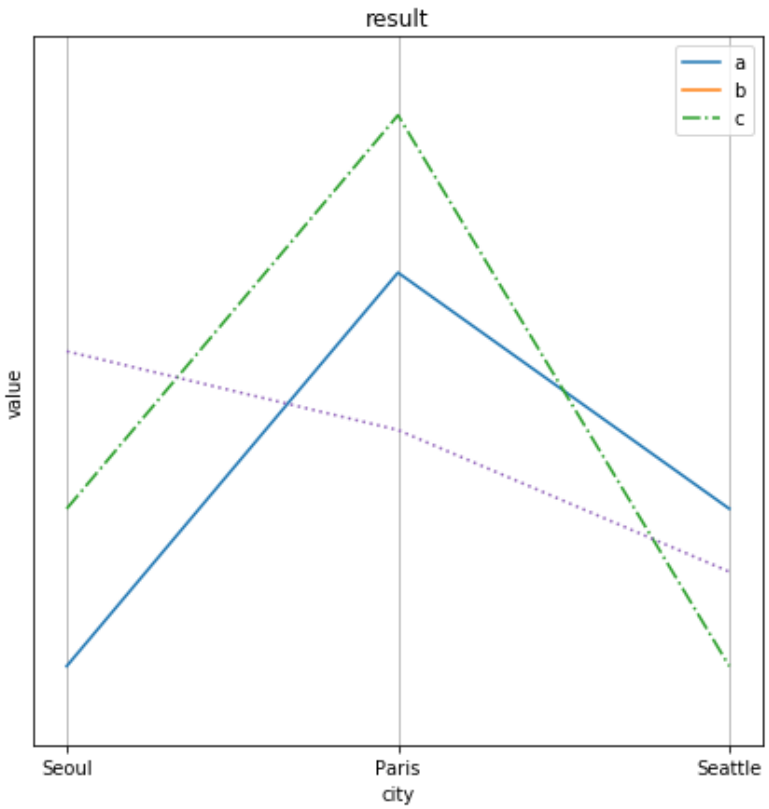


실습

In [14]:
```
# 이미지 크기 변경
plt.figure(figsize = (5, 5))

# plot 그리기
plt.plot(["Seoul", "Paris", "Seattle"], [10, 35, 20], '-', color = 'orange')
plt.plot(["Seoul", "Paris", "Seattle"], [20, 45, 10], '-.', color = 'blue')
plt.plot(["Seoul", "Paris", "Seattle"], [30, 25, 16], ':', color = 'yellow')

# x축 이름 지정
plt.xlabel('city')

# y축 이름 지정
plt.ylabel('values')

# 제목 지정
plt.title('example')

# 범례 추가
# upper, center, lower / left, center, right
plt.legend(['a', 'b', 'c'], loc = 'upper left')

# 축값 변경
plt.ylim(0, 60)

# grid 표시
plt.grid(True)
```



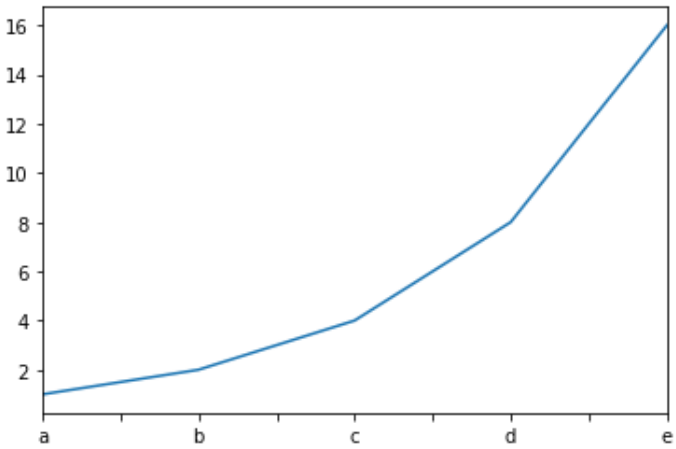
In [8]:
```
# 이미지 크기 변경
plt.figure(figsize = (7, 7))

# plot 그리기
plt.plot(["Seoul", "Paris", "Seattle"], [10, 35, 20], '-', 'r1--')
plt.plot(["Seoul", "Paris", "Seattle"], [20, 45, 10], '-.', 'g2-')
plt.plot(["Seoul", "Paris", "Seattle"], [30, 25, 16], ':', 'b3:')

# x축 이름 지정
plt.xlabel('city')

# y축 이름 지정
plt.ylabel('value')

# 제목 지정
plt.title('result')

# 범례 추가
# upper, center, lower / left, center, right
plt.legend(['a', 'b', 'c'], loc = 'upper right')

# 축값 변경
plt.ylim(5, 50)

# grid 표시
plt.grid(True)
```



(2) data.plot()

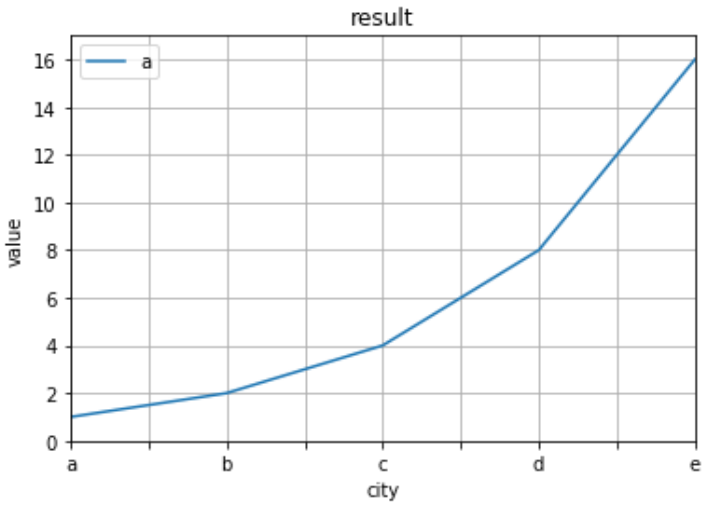
axes = data.plot()으로 지정후 시각화 변경

In [15]:
```
axes = sr.plot()
```



In [19]:
```
# x축 이름 지정
axes.set_xlabel('city')

# # y축 이름 지정
axes.set_ylabel('value')

# # 제목 지정
axes.set_title('result')

# 범례 추가
# upper, center, lower / left, center, right
axes.legend('a', loc = 'upper left')

# 축값 변경
axes.set_ylim(0, 17)

# grid 표시
axes.grid(True)

axes.figure
```

Out[19]:



2. bar 그리기

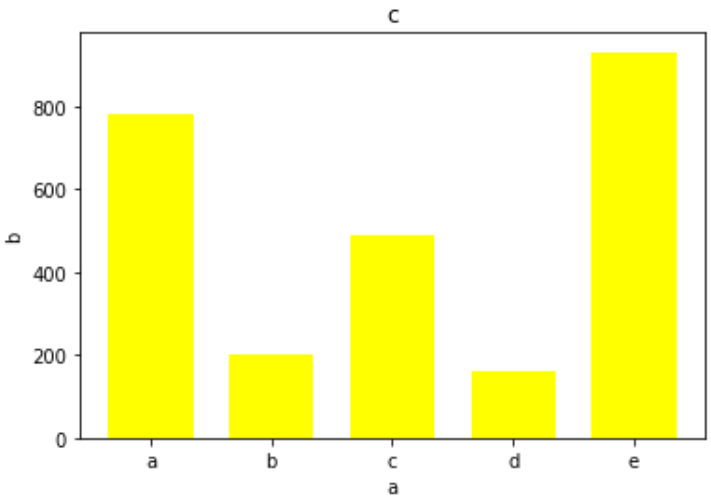
plt.bar(x = 'x축 표시', height = 'y축 표시', width = '너비', color = '색깔')

In [20]:
```
import random

sr2 = pd.Series(random.sample(range(100, 1001), 5), index = list('abcde'))
sr2
```

Out[20]:
```
a    780
b    200
c    490
d    162
e    932
dtype: int64
```

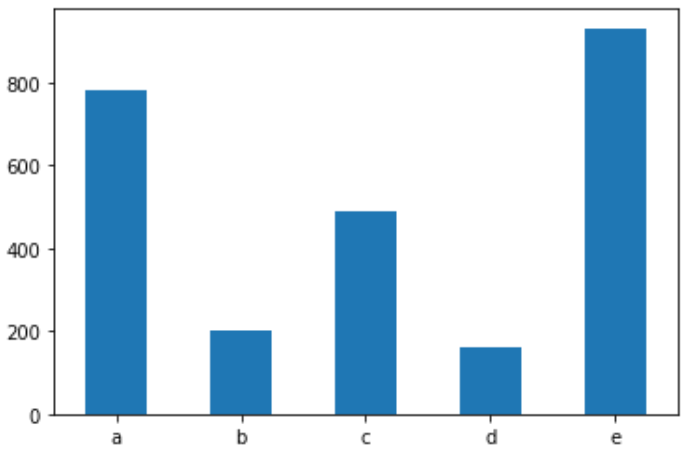
In [23]:
```
# bar 그리기
plt.bar(x = sr2.index, height = sr2.values, width = 0.7, color = 'yellow')

# x축 지정
plt.xlabel('a')

# y축 지정
plt.ylabel('b')

# 제목
plt.title('c')
```

Out[23]:
```
Text(0.5, 1.0, 'c')
```



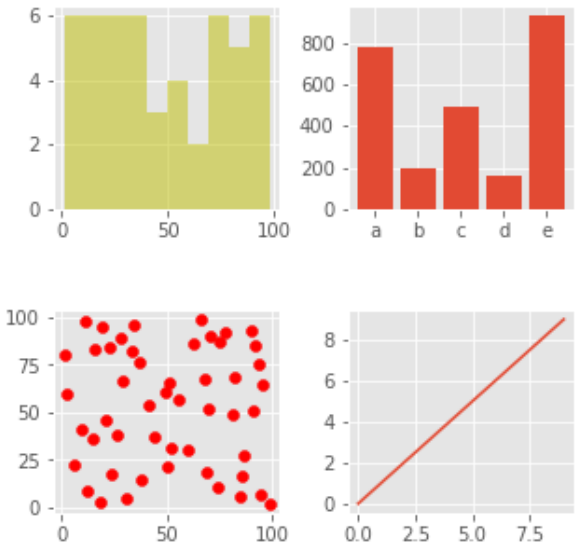
In [25]:
```
sr2.plot(kind = 'bar', rot = 0)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a20fa3b10>
```



여러개의 그래프를 하나로 그리기

In [37]:
```
# figsize 지정
fig = plt.figure(figsize = (5, 5))

# ax1
ax1 = fig.add_subplot(2, 2, 1)

# ax2
ax2 = fig.add_subplot(2, 2, 2)

# ax3
ax3 = fig.add_subplot(2, 2, 3)

# ax4
ax4 = fig.add_subplot(2, 2, 4)

# histogram 그리기
ax1.hist(random.sample(range(1, 100), 50), bins = 10, color = 'y', alpha = 0.5)

# bar 그리기
ax2.bar(x = sr2.index, height = sr2.values)

# scatter 그리기
ax3.scatter(random.sample(range(1, 100), 50),
            random.sample(range(1, 100), 50), color = 'r')

# ax4 위치에 y=x 그래프를 그려보기
ax4.plot([x for x in range(10)])

# hspace: height, wspace: width
plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
```



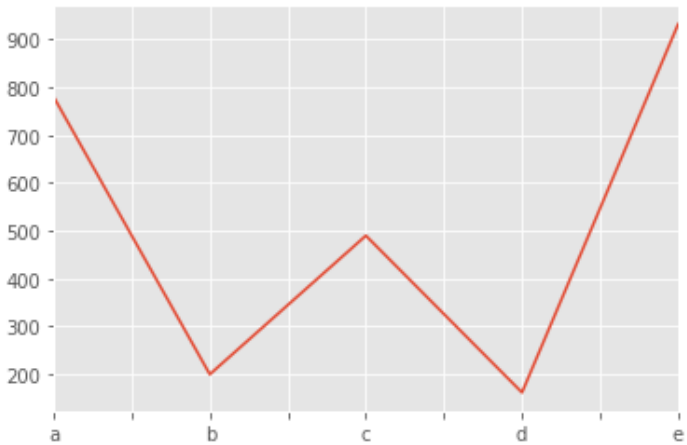
In [34]:
```
# style 변경
# 사용가능한 style 출력
plt.style.available
```

Out[34]:
```
['seaborn-dark',
 'seaborn-darkgrid',
 'seaborn-ticks',
 'fivethirtyeight',
 'seaborn-whitegrid',
 'classic',
 '_classic_test',
 'fast',
 'seaborn-talk',
 'seaborn-dark-palette',
 'seaborn-bright',
 'seaborn-pastel',
 'grayscale',
 'seaborn-notebook',
 'ggplot',
 'seaborn-colorblind',
 'seaborn-muted',
 'seaborn',
 'Solarize_Light2',
 'seaborn-paper',
 'bmh',
 'tableau-colorblind10',
 'seaborn-white',
 'dark_background',
 'seaborn-poster',
 'seaborn-deep']
```

In [35]:
```
# ggplot 사용
plt.style.use('ggplot')
```

In [36]:
```
# plot그려보기
sr2.plot()
```

Out[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a229d4890>
```



1. iris에서 sepal_length(꽃받침의 길이) 1행부터 20행까지의 데이터로 막대그래프 그리기

단 sepal_length의 길이가 5가 넘으면 파란색 아니면 빨간색

실습

In [2]:
```
# 데이터 불러오기
iris = sns.load_dataset('iris')
titanic = sns.load_dataset('titanic')
```

In [3]:
```
iris
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [6]:
```
import matplotlib
# 한글 깨짐 방지
matplotlib.rcParams['font.family'].insert(0, 'AppleGothic')
```

In [7]:
```
matplotlib.rcParams['font.family']
```

Out[7]:
```
['AppleGothic', 'sans-serif']
```

In [9]:
```
# 의도
my_color = []
for i in iris['sepal_length'][:20]:
    if i >= 5:
        my_color.append('blue')
    else:
        my_color.append('red')

my_color
```

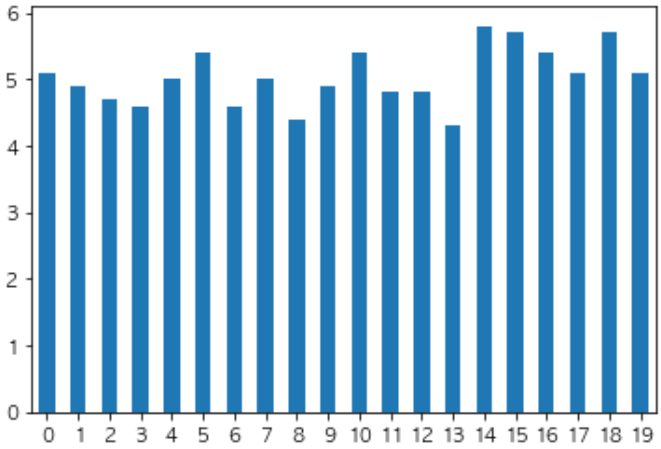
Out[9]:
```
['blue',
 'red',
 'red',
 'red',
 'blue',
 'blue',
 'red',
 'blue',
 'red',
 'red',
 'blue',
 'red',
 'red',
 'red',
 'blue',
 'blue',
 'blue',
 'blue',
 'blue',
 'blue']
```

In [12]:
```
['blue' if x > 5 else 'red' for x in iris['sepal_length'][:20]]
```

Out[12]:
```
['blue',
 'red',
 'red',
 'red',
 'red',
 'blue',
 'red',
 'red',
 'red',
 'red',
 'blue',
 'red',
 'red',
 'red',
 'blue',
 'blue',
 'blue',
 'blue',
 'blue',
 'blue']
```

In [8]:
```
iris['sepal_length'][:20].plot(kind = 'bar', rot = 0, color = )
```

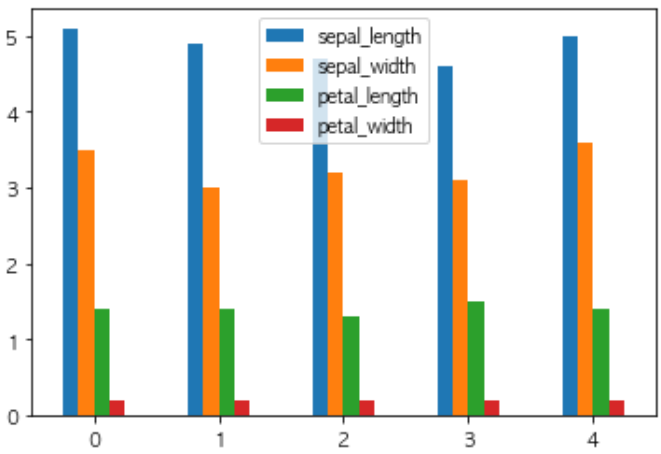
Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1891f490>
```



2. iris에서 1행부터 5행까지의 데이터를 시각화해보기

In [13]:
```
iris[:5].plot(kind = 'bar', rot = 0)
```

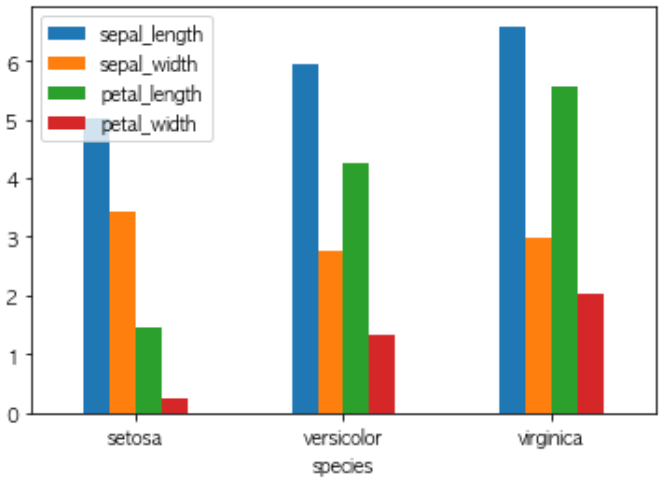
Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a190f6310>
```



3. species별 평균 막대그래프 시각화

In [15]:
```
iris.groupby('species').mean().plot(kind = 'bar', rot = 0)
```

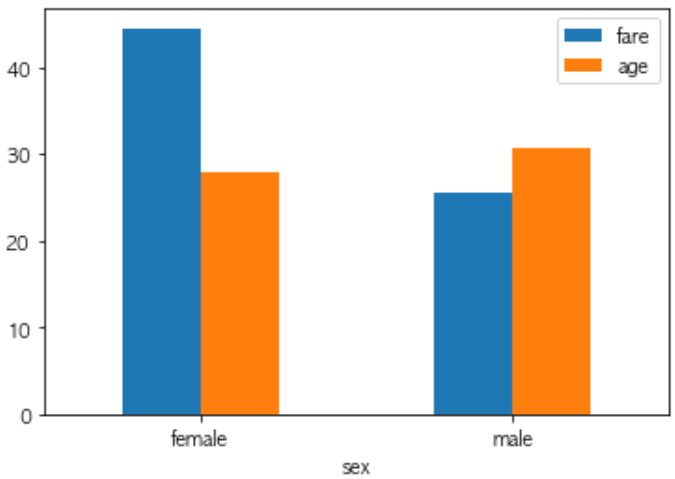
Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1946fd50>
```



4. 성별에따른 fare, age 평균 막대그래프 시각화

In [18]:
```
titanic.groupby('sex')[['fare', 'age']].mean().plot(kind = 'bar', rot = 0)
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a196a9f90>
```



5. titanic에서

- fare와 age column에서 NaN값을 지우고

- 나이를 기준으로 오름차순 정렬 이후

- 나이에 따른 요금 선그래프 시각화

In [22]:
```
titanic.dropna(subset = ['fare', 'age'], inplace = True)
```

In [25]:
```
temp = titanic.sort_values(by = 'age', ascending = True)
temp
```

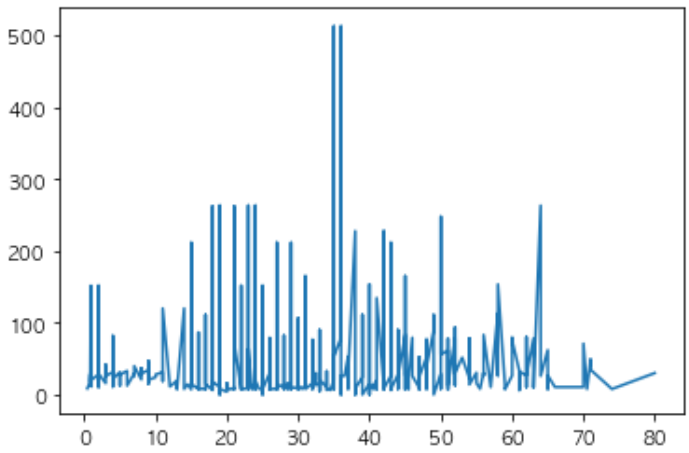
Out[25]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
803	1	3	male	0.42	0	1	8.5167	C	Third	child	False	NaN
755	1	2	male	0.67	1	1	14.5000	S	Second	child	False	NaN
644	1	3	female	0.75	2	1	19.2583	C	Third	child	False	NaN
469	1	3	female	0.75	2	1	19.2583	C	Third	child	False	NaN
78	1	2	male	0.83	0	2	29.0000	S	Second	child	False	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
116	0	3	male	70.50	0	0	7.7500	Q	Third	man	True	NaN
493	0	1	male	71.00	0	0	49.5042	C	First	man	True	NaN
96	0	1	male	71.00	0	0	34.6542	C	First	man	True	A
851	0	3	male	74.00	0	0	7.7750	S	Third	man	True	NaN
630	1	1	male	80.00	0	0	30.0000	S	First	man	True	A

714 rows × 15 columns

In [26]:
```
plt.plot(temp.age, temp.fare)
```

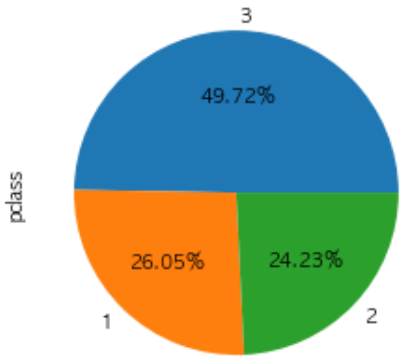
Out[26]:
```
[<matplotlib.lines.Line2D at 0x1a19906150>]
```



선실별 승객 수 비율 pie chart 시각화

In [28]:
```
temp = titanic.pclass.value_counts()
temp.plot(kind = 'pie', autopct = '%.2f%%')
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1998c690>
```



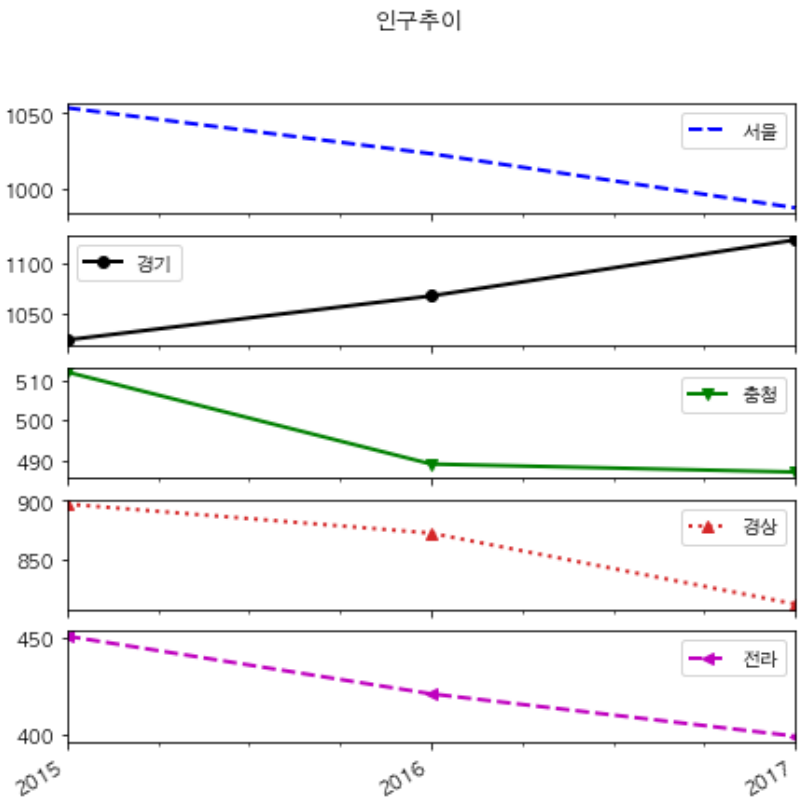
In [29]:
```
인구통계 = {'서울': [1053.5, 1023, 987],
            '경기':[1023, 1067, 1123],
            '충청':[512,489,487],
            '경상':[897, 872, 811],
            '전라':[451, 421, 399]
            }

df = pd.DataFrame(인구통계)
df.index = [2015, 2016, 2017]
df
```

Out[29]:

	서울	경기	충청	경상	전라
2015	1053.5	1023	512	897	451
2016	1023.0	1067	489	872	421
2017	987.0	1123	487	811	399

In [30]:
```
# 기본 그리기
# https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.plot.
df.plot(kind = 'line',
        title = '인구추이',
        figsize = (7, 7),
        style = ['b,--', 'ko-', 'vg-', '^:', 'm<--'],
        lw = 2,   # line width: 선 두께
        subplots = True,
        xticks = df.index
       )
```

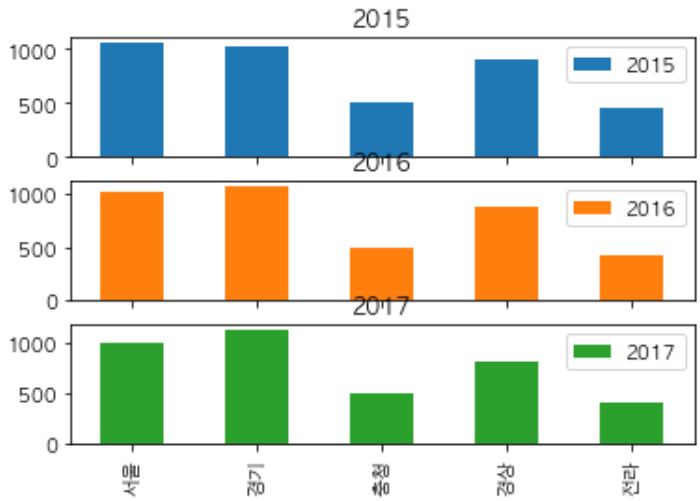
Out[30]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x1a1b0ed650>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1b5e2590>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1a698e10>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1c94a810>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1c90fe90>],
      dtype=object)
```



In [31]:
```
import matplotlib
# 한글 깨짐 방지
matplotlib.rcParams['font.family'].insert(0, 'Malgun Gothic')
# Mac OS인 경우에는 AppleGothic 추가
```

In [32]:
```
# 실습. 아래 그림처럼 연도별 지역별 인구수 그래프 그리기
df.T.plot(kind = 'bar',
          subplots = True)
```

Out[32]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x1a17c45950>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1c7db950>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1a1c7cda50>],
      dtype=object)
```



2. 왕좌의 게임 데이터 분석 및 시각화 실습

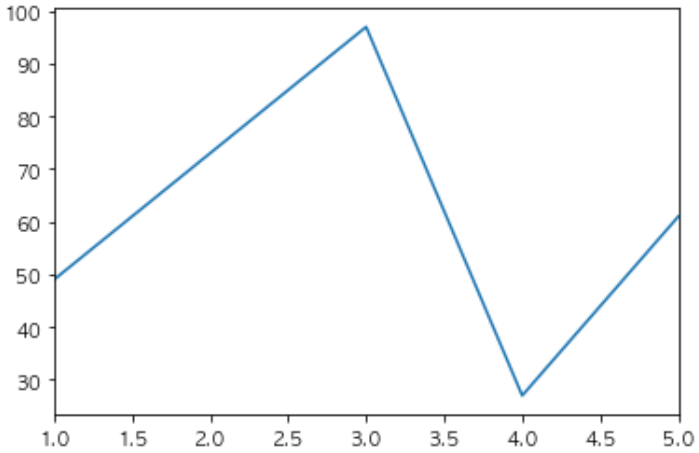
1. 데이터 불러오기

In [33]:
```
battles = pd.read_csv('data/data-society-game-of-thrones/battles.csv')
death = pd.read_csv('data/data-society-game-of-thrones/character-deaths.csv')
```

2. 책의 챕터(Book of Death)별로 사망자 수 추이 시각화

In [35]:
```
death['Book of Death'].value_counts().sort_index().plot()
# plt.xlim(0, 6)
# plt.grid(True)
```

Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1c8b3290>
```



3. 대규모 전쟁의 공격진영과 수비 진영의 숫자 시각화

대규모 전쟁 = 공격과 수비 모두 합쳐서 10,000명 이상 참가한 전쟁

In [41]:
```
# 1. 대규모 전쟁만 선택
big_battles = battles[battles['attacker_size'] + battles['defender_size'] >= 1(
big_battles
```

Out[41]:

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	attacker_3	a
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN	
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN	
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	NaN	
5	Battle of the Camps	298	6	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	NaN	
14	Battle of Oxcross	299	15	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	NaN	
15	Siege of Storm's	299	16	Stannis Baratheon	Renly Baratheon	Baratheon	NaN	NaN	

In [42]:
```
# 2. 필요한 컬럼만 선택
big_battles = big_battles[['name', 'attacker_size', 'defender_size']]
big_battles
```

Out[42]:

	name	attacker_size	defender_size
0	Battle of the Golden Tooth	15000.0	4000.0
2	Battle of Riverrun	15000.0	10000.0
3	Battle of the Green Fork	18000.0	20000.0
5	Battle of the Camps	6000.0	12625.0
14	Battle of Oxcross	6000.0	10000.0
15	Siege of Storm's End	5000.0	20000.0
16	Battle of the Fords	20000.0	10000.0
19	Battle of the Blackwater	21000.0	7250.0
27	Battle of Castle Black	100000.0	1240.0
37	Siege of Winterfell	5000.0	8000.0

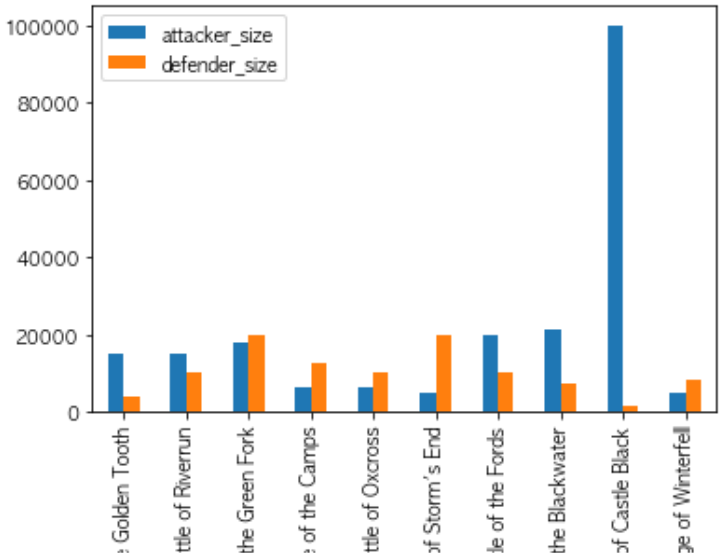
In [43]:
```
big_battles.set_index('name', inplace = True)
big_battles
```

Out[43]:

name	attacker_size	defender_size
Battle of the Golden Tooth	15000.0	4000.0
Battle of Riverrun	15000.0	10000.0
Battle of the Green Fork	18000.0	20000.0
Battle of the Camps	6000.0	12625.0
Battle of Oxcross	6000.0	10000.0
Siege of Storm's End	5000.0	20000.0
Battle of the Fords	20000.0	10000.0
Battle of the Blackwater	21000.0	7250.0
Battle of Castle Black	100000.0	1240.0
Siege of Winterfell	5000.0	8000.0

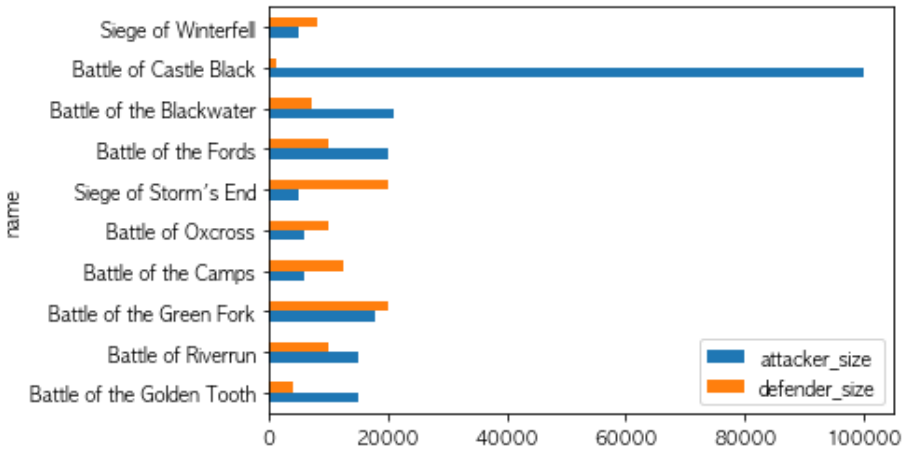
In [44]:
```
# big_battles 시각화
big_battles.plot(kind = 'bar')
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1ca1a9d0>
```



In [45]:
```
big_battles.plot(kind = 'barh')
```

Out[45]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1cb58490>
```



4. 년도별로 사망자의 숫자와 전쟁이 벌어진 횟수 시각화

In [46]:
```
death
```

Out[46]:

	Name	Allegiances	Death Year	Book of Death	Death Chapter	Book Intro Chapter	Gender	Nobility	GoT	CoK	SoS	FfC	D
0	Addam Marbrand	Lannister	NaN	NaN	NaN	56.0	1	1	1	1	1	1	
1	Aegon Frey (Jinglebell)	None	299.0	3.0	51.0	49.0	1	1	0	0	1	0	
2	Aegon Targaryen	House Targaryen	NaN	NaN	NaN	5.0	1	1	0	0	0	0	
3	Adrack Humble	House Greyjoy	300.0	5.0	20.0	20.0	1	1	0	0	0	0	
4	Aemon Costayne	Lannister	NaN	NaN	NaN	NaN	1	1	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
912	Zollo	None	NaN	NaN	NaN	21.0	1	0	0	0	1	0	
913	Yurkhaz zo Yunzak	None	300.0	5.0	59.0	47.0	1	0	0	0	0	0	
914	Yezzan Zo Qaggaz	None	300.0	5.0	57.0	25.0	1	1	0	0	0	0	
915	Torwynd the Tame	Wildling	300.0	5.0	73.0	73.0	1	0	0	0	1	0	
916	Talbert Serry	Tyrell	300.0	4.0	29.0	29.0	1	1	0	0	0	1	

917 rows × 13 columns

In [48]:
```
temp = death.groupby('Death Year')['Name'].count()
temp
```

Out[48]:
```
Death Year
297.0      3
298.0     46
299.0    156
300.0    100
Name: Name, dtype: int64
```

In [49]:
```
battles
```

Out[49]:

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	attacker_3
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	NaN
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	NaN
4	Battle of the Whispering Wood	298	5	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	NaN
5	Battle of the Camps	298	6	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	NaN

In [51]:
```
temp2 = battles.groupby('year')['name'].count()
temp2
```

Out[51]:
```
year
298      7
299     20
300     11
Name: name, dtype: int64
```

In [53]:
```
temp3 = pd.concat([temp, temp2], axis = 1)
temp3
```

Out[53]:

	Name	name
297.0	3	NaN
298.0	46	7.0
299.0	156	20.0
300.0	100	11.0

In [54]:
```
temp3.columns = ['Death', 'battles']
temp3.plot()
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1ccc3750>
```