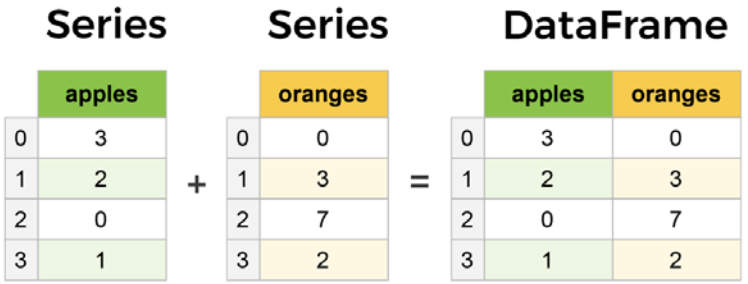
07. 데이터 분석을 위한 주요 라이브러리

- **pandas**
  - **Series**
  - **DataFrame**




```
import pandas as pd

data = {
    'apples': [3, 2, 0, 1],
    'oranges': [0, 3, 7, 2]
}

purchases = pd.DataFrame(data)
purchases
purchases = pd.DataFrame(data, index=['June', 'Robert', 'Lily', 'David'])
purchases
col1=pd.Series([3, 2, 0, 1], name='apples')
col2=pd.Series([3, 2, 0, 1], name='oranges', index=['June', 'Robert', 'Lily', 'David'])
col1
col1.name
col1.values
col1.index
purchases2 = pd.DataFrame(col1)
purchases2
purchases3 = pd.DataFrame(col2)
purchases3
col2=pd.Series([3, 2, 0, 1], name='oranges')
purchases4= pd.concat([col1, col2], axis=1)
purchases4
purchases4.index=['June', 'Robert', 'Lily', 'David']
purchases4
#purchases4.name
purchases4.index
purchases4.columns
```

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

- Series 자료형

```
import pandas as pd
pd.__version__
'1.0.3'
data1 = [10, 20, 30, 40, 50]
data1
[10, 20, 30, 40, 50]
data2 = ['1반', '2반', '3반', '4반', '5반']
data2
['1반', '2반', '3반', '4반', '5반']
sr1 = pd.Series(data1)
sr1
0   10
1   20
2   30
3   40
4   50
dtype: int64
sr2 = pd.Series(data2)
sr2
0   1반
1   2반
2   3반
3   4반
4   5반
dtype: object
```

```
sr3 = pd.Series([101, 102, 103, 104, 105])
sr3
0   101
1   102
2   103
3   104
4   105
dtype: int64
sr4 = pd.Series(['월', '화', '수', '목', '금'])
sr4
0   월
1   화
2   수
3   목
4   금
dtype: object
```

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

- Series 자료형

```
sr5 = pd.Series(data1, index = [1000, 1001, 1002, 1003, 1004])
sr5
1000   10
1001   20
1002   30
1003   40
1004   50
dtype: int64
sr6 = pd.Series(data1, index = data2)
sr6
1반   10
2반   20
3반   30
4반   40
5반   50
dtype: int64
sr7 = pd.Series(data2, index = data1)
sr7
10   1반
20   2반
30   3반
40   4반
50   5반
dtype: object
```

```
sr8 = pd.Series(data2, index = sr4)
sr8
월   1반
화   2반
수   3반
목   4반
금   5반
dtype: object
sr8[2]
'3반'
sr8['수']
'3반'
sr8[-1]
'5반'
sr8[0:4]
월   1반
화   2반
수   3반
목   4반
dtype: object
sr8.index
Index(['월', '화', '수', '목', '금'], dtype = 'object')
sr8.values
array(['1반', '2반', '3반', '4반', '5반'], dtype = object)
```

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

- DataFrame 자료형

```
data_dic = {
            'year': [2018, 2019, 2020],
            'sales': [350, 480, 1099] }
data_dic
{'year': [2018, 2019, 2020], 'sales': [350, 380, 1099]}
df1 = pd.DataFrame(data_dic)
df1
```

	year	sales
0	2018	350
1	2019	380
2	2020	1099

```
data2 = ['1반', '2반', '3반', '4반', '5반']
df2 = pd.DataFrame([[89.2, 92.5, 90.8], [92.8, 89.9, 95.2]],
index = ['중간고사', '기말고사'], columns = data2[0:3])
df2
```

	1반	2반	3반
중간고사	89.2	92.5	90.8
기말고사	92.8	89.9	95.2

```
data_df = [['20201101', 'Hong', '90', '95'], ['20201102',
'Kim', '93', '94'], ['20201103', 'Lee', '87', '97']]
df3 = pd.DataFrame(data_df)
df3
```

	0	1	2	3
0	20201101	Hong	90	95
1	20201102	Kim	93	94
2	20201103	Lee	87	97

```
df3.columns = ['학번', '이름', '중간고사', '기말고사']
df3
```

	학번	이름	중간고사	기말고사
0	20201101	Hong	90	95
1	20201102	Kim	93	94
2	20201103	Lee	87	97

```
df3.head(2)
```

	학번	이름	중간고사	기말고사
0	20201101	Hong	90	95
1	20201102	Kim	93	94

```
df3.tail(2)
```

	학번	이름	중간고사	기말고사
1	20201102	Kim	93	94
2	20201103	Lee	87	97

```
df3['이름']
0    Hong
1    Kim
2    Lee
Name: 이름, dtype: object
```

07. 데이터 분석을 위한 주요 라이브러리

■ pandas indexing 살펴보기

```
import pandas as pd
df = pd.DataFrame([[60, 61, 62], [70, 71, 72], [80, 81, 82], [90, 91, 92]],
index = ['1반', '2반', '3반','4반'], columns = ['퀴즈1', '퀴즈2', '퀴즈3'])
#df : 열 선택
df.퀴즈1
df['퀴즈1']
df['퀴즈1'][2]
#df.loc : 행 선택, 행열선택
df.loc['2반']
df.loc['2반', '퀴즈1']
df.loc['2반':'4반', '퀴즈1']            # type(df.loc[ ' 2반 ' : ' 4반 ' ,  ' 퀴즈1 ' ])
df.loc['2반':'4반', '퀴즈1':'퀴즈3']    # type(df.loc[ ' 2반 ' : ' 4반 ' ,  ' 퀴즈1 ' : ' 퀴즈2 ' ])
#df.iloc : 행 선택, 행열선택
df.iloc[2]
df.iloc[2, 1]
df.iloc[2:4, 0]
df.iloc[2:4, 0:2]
df.iloc[2:4, 0:1]
```

Index	퀴즈1	퀴즈2	퀴즈3
1반	60	61	62
2반	70	71	72
3반	80	81	82
4반	90	91	92

[참고]
1. https://bearwoong.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-DataFrame-%EC%9D%B8%EB%8D%B1%EC%8B%B1-%ED%95%98%EB%8A%94-%EB%B0%A9%EB%B2%95df-dfloc-dfiloc

2. https://nittaku.tistory.com/111

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

Index	중간고사	기말고사	성적
1반	89.1	90.1	B
2반	89.2	90.2	A
3반	89.3	90.3	A
4반	89.4	90.4	C
5반	89.5	90.5	B

```
#
import pandas as pd
data2 = ['1반', '2반', '3반', '4반', '5반']

df = pd.DataFrame([[89.1, 90.1, 'B'], [89.2, 90.2, 'A'], [89.3,
90.3, 'A'], [89.4, 90.4, 'C'], [89.5, 90.5, 'B']],
      index = data2, columns = ['중간고사', '기말고사', '성적'])
```

```
# df

df['기말고사']
df.기말고사
df[['중간고사','기말고사']]
df['2반':'4반']   # // df['2반','4반']  error
df['중간고사'][3] // df.iloc[0, 0]
df['중간고사']['1반':'2반']
df['중간고사'][0:2]
df[0:2]['중간고사']

# loc
df.loc['5반']    # df.loc['중간고사']  error
df.loc['1반':'2반', '중간고사']
df.loc[ : , '기말고사']

# iloc
df.iloc[0:2]['중간고사']
df.iloc[4]
```

읽어보기 => https://bearwoong.tistory.com/65

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

Index	중간고사	기말고사	성적
1반	89.1	90.1	B
2반	89.2	90.2	A
3반	89.3	90.3	A
4반	89.4	90.4	C
5반	89.5	90.5	B

```
df[df['성적'] == 'B']
df[df.성적 == 'B']

df.loc[df.성적 == 'B']
df.loc[df.성적 == 'B']
```

Index	중간고사	기말고사	성적
1반	89.1	90.1	B
2반	89.2	90.2	A
3반	89.3	90.3	A
4반	89.4	90.4	C
5반	89.5	90.5	B

```
df[df.성적.isin(['B', 'C'])]
df.loc[df.성적.isin(['B', 'C'])]
df[(df.성적 == 'A') & (df.중간고사 >= 90)]
df.loc[(df.성적 == 'A') & (df.중간고사 >= 90)]
```

읽어보기 => https://bearwoong.tistory.com/65

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

Index	중간고사	기말고사	성적
1반	89.1	90.1	B
2반	89.2	90.2	A
3반	89.3	90.3	A
4반	89.4	90.4	C
5반	89.5	90.5	B

dataframe-loc-iloc.py

```
## summary function and maps
df.describe()
df.중간고사.describe()
df.head(1)
df.중간고사.unique()
df.중간고사.mean()
df.중간고사.value_counts()
df_mean = df.중간고사.mean()
df.중간고사.map(lambda p: p - df_mean)

## grouping and sorting
df.groupby('중간고사').중간고사.count()
df.groupby('중간고사').중간고사.min()
df.groupby(['중간고사']).중간고사.agg([len, min, max])
df.sort_values(by='중간고사')
df.sort_values(by='중간고사', ascending=False)
df.sort_index(ascending=False)
```

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

Index	중간고사	기말고사	성적
1반	89.1	90.1	B
2반	89.2	90.2	A
3반	89.3	90.3	A
4반	89.4	90.4	C
5반	89.5	90.5	B

```
# data types and missing values
df.dtypes
df.중간고사.dtypes
df.loc['6반']=[10, 10, np.nan]
df[pd.isnull(df.성적)]

# renaming and combining
df.rename(columns={'성적': '등급'})
df.rename_axis("반이름", axis='rows')

df1 = pd.DataFrame([[89.2, 92.5, 'B'],
                    [90.8,92.8, 'A'],
                    [89.9, 95.2, 'A'],
                    [89.9, 85.2, 'C'],
                    [89.9, 90.2, 'B']],
    columns = ['중간고사', '기말고사', '성적'],
    index = ['1반', '2반', '3반', '4반', '5반'])

df0=pd.concat([df, df1])
```

읽어보기 => https://bearwoong.tistory.com/65

07. 데이터 분석을 위한 주요 라이브러리

■ pandas

- DataFrame 자료형 : score.csv 파일은 아래와 같이 만들어 읽어들이기

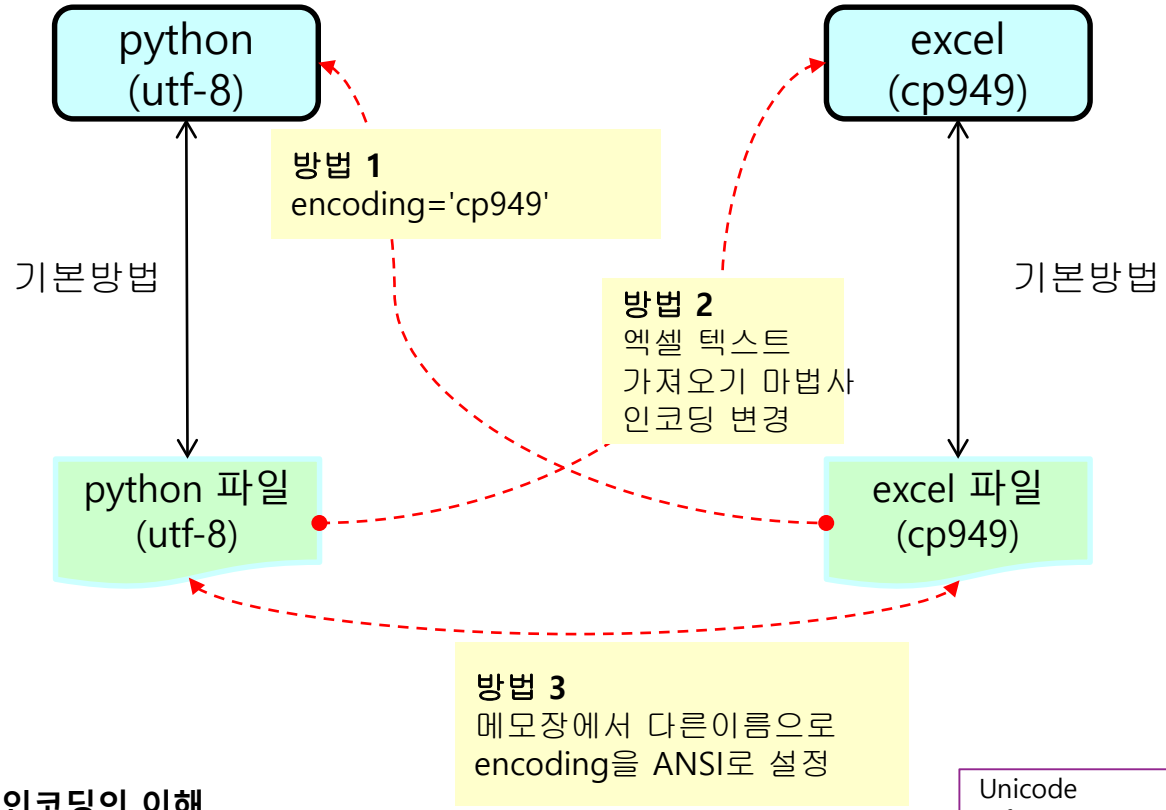
```
  학번      이름      중간고사    기말고사
0 20201101 Hong          90          95
1 20201102 Kim           93          94
2 20201103 Lee           87          97
```

```
# df4 = pd.read_csv('C:/Users/kmj/My_Python/score.csv') – 오류나는지 확인해보기
df4 = pd.read_csv('C:/Users/kmj/My_Python/score.csv', encoding='utf-8', index_col=0, engine='python')
df4
```

```
    학번      이름      중간고사    기말고사
0 20201101 Hong          90          95
1 20201102 Kim           93          94
2 20201103 Lee           87          97
```

```
df3.to_csv('C:/Users/kmj/My_Python/score2.csv', header = 'False')
```

한글코드(python, excel)



[참고] 한글인코딩의 이해

1. https://ifyourfriendishacker.tistory.com/5
2. https://smorning.tistory.com/269

Unicode	: 2바이트
utf-8	: 가변길이 Unicode 1~4바이트
euc-kr	:
cp949	: 완성형, euc-kr 확장 및 하위호환

07. 데이터 분석을 위한 주요 라이브러리

- DataFrame 자료 보기 명령어

구분	pandas *DataFrame (df)*	pandas *Series (s)*
행 개수 세기 (row count)	len(df) df.shape[0] len(df.index)	len(s) s.size len(s.index)
열 개수 세기 (column count)	df.shape[1] len(df.columns)	N/A
Null 값이 아닌 행 개수 세기 (Non-null row count)	df.count()	s.count()
그룹 별 행 개수 세기 (Row count per group)	df.groupby(...).size()	s.groupby(...).size()
그룹 별 Null 값이 아닌 행 개수 세기 (Non-null row count per group)	df.groupby(...).count()	s.groupby(...).count()

https://www.w3resource.com/pandas/dataframe/dataframe-shape.php

https://www.w3resource.com/pandas/dataframe/dataframe-count.php

https://www.w3resource.com/pandas/dataframe/dataframe-groupby.php

https://www.w3resource.com/pandas/dataframe/dataframe-count.php

데이터프레임의 행과 열의 개수 세기 => https://rfriend.tistory.com/450

07. 데이터 분석을 위한 주요 라이브러리

■ matplotlib

- 라인플롯 차트 그리기

1. 데이터 준비
>>> x = [2016, 2017, 2018, 2019, 2020]
>>> y = [350, 410, 520, 695, 543]

2. x축과 y축 데이터를 지정하여 라인플롯 생성
>>> plt.**plot**(x, y)
[<matplotlib.lines.Line2D object at 0x0000015DB82D58C8>]

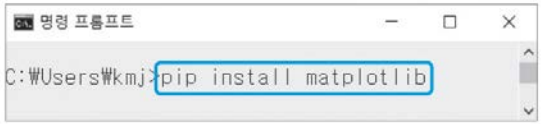
3. 차트 제목 설정
>>> plt.**title**('Annual sales')
Text(0.5, 1.0, 'Annual sales')

4. x축 레이블 설정
>>> plt.**xlabel**('years')
Text(0.5, 0, 'years')

5. y축 레이블 설정
>>> plt.**ylabel**('sales')
Text(0, 0.5, 'sales')

6. 라인플롯 표시
>>> plt.**show()**

- 임포트



>>> import matplotlib
matplotlib 버전 확인 >>> matplotlib.__version__
'3.2.1'
pyplot 모듈 임포트하기 >>> import matplotlib.pyplot as plt

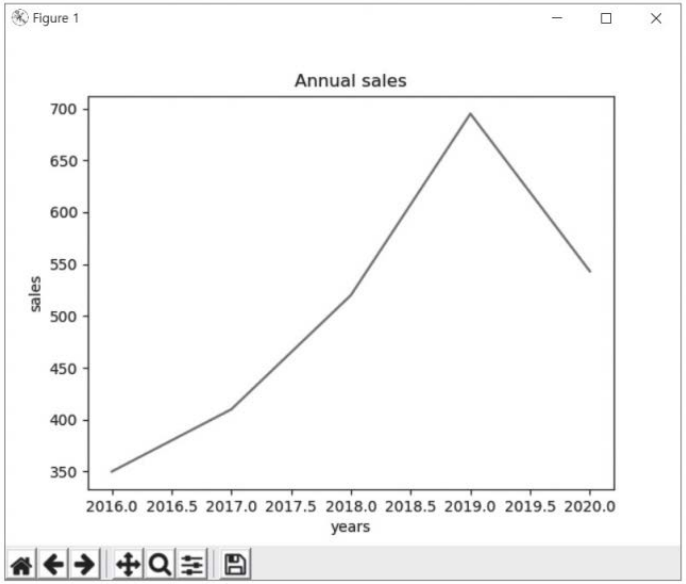


그림 4-7 라인플롯 차트

07. 데이터 분석을 위한 주요 라이브러리

■ matplotlib

- 바차트 차트 그리기

1. 데이터 준비
```
>>> y1 = [350, 410, 520, 695]
>>> y2 = [200, 250, 385, 350]
>>> x = range(len(y1))
```

2. x축과 y축 데이터를 지정하여 라인플롯 생성
```
>>> plt.bar(x, y1, width = 0.7, color = "blue")
<BarContainer object of 4 artists>
>>> plt.bar(x, y2, width = 0.7, color = "red",
bottom = y1)
<BarContainer object of 4 artists>
```

3. 차트 제목 설정
```
>>> plt.title('Quarterly sales')
Text(0.5, 1.0, 'Quarterly sales')
```

4. x축 레이블 설정
```
>>> plt.xlabel('Quarters')
Text(0.5, 0, 'Quarters')
```

5. y축 레이블 설정
```
>>> plt.ylabel('sales')
Text(0, 0.5, 'sales')
```

6. 눈금 이름 리스트 생성
```
>>> xLabel = ['first', 'second', 'third', 'fourth']
```

7. 바 차트의 x축 눈금 이름 설정
```
>>> plt.xticks(x, xLabel, fontsize = 10)
([<matplotlib.axis.XTick object at
0x0000015DB5722B48>, <matplotlib.axis.XTick
object at 0x0000015DB5722B08>, <matplotlib.
axis.XTick object at 0x0000015DB82E2688>,
<matplotlib.axis.XTick object at
0x0000015DB60C5188>], [Text(0, 0, 'first'),
Text(0, 0, 'second'), Text(0, 0, 'third'), Text(0,
0, 'fourth')])
```

8. 범례 설정
```
>>> plt.legend(['chairs', 'desks'])
<matplotlib.legend.Legend object at
0x0000020F2BBA0908>
```

9. 바 차트 표시
```
>>> plt.show()
```

07. 데이터 분석을 위한 주요 라이브러리

■ matplotlib

- 바차트 차트 그리기

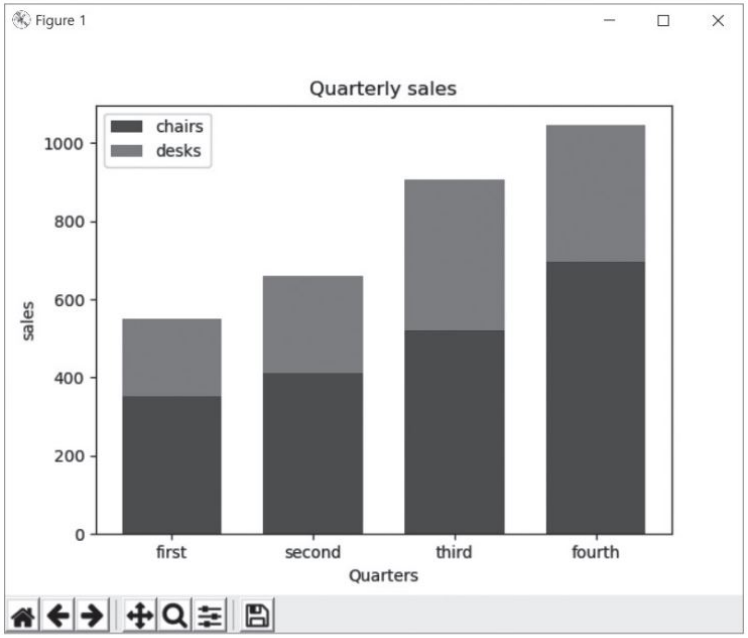


그림 4-8 바 차트