Py 데이터 파일

Ⅰ. Py 데이터 파일 만들기

1. 명령어를 이용하는 방법

- 변수명은 한글이나 영문 모두 사용이 가능한데 프로그래밍 편의상 변수명을 영어로 입력
- 예를 들어, 이름(name), 경제원론(prin), 미시경제(micro), 거시경제(macro)를 보여주는 데이터는 프로그램 작성 시 다음(b3-ch1-3.Py)과 같이 명령어를 입력

b3-ch1-3.Py

```
import pandas as pd
# 데이터 프레임 만들기
df1 = pd.DataFrame({'name' :['김기훈','박수동','원선희','위계영','최동팔','최종열','최종수','김기팔','이상수','강창수'],
'prin' :[98,100,50,50,80,90,30,80,65,95],
'micro' :[82,92,45,100,95,60,30,25,70,90],
'macro' :[95,80,75,100,95,60,30,25,70,90]})
df1
print("Data Frame of df1 is :",  f'\n{df1}\n')
```

b3-ch1-3.Py 실행결과

```
runfile('C:/BOOK/PyBasics/PyEda/code/b3-ch1-3.py', wdir='C:/BOOK/PyBasics/PyEda/code')
Data Frame of df1 is :
name prin micro macro
0 김기훈 98 82 95
1 박수동 100 92 80
2 원선희 50 45 75
3 위계영 50 100 100
4 최동팔 80 95 95
5 최종열 90 60 60
6 최종수 30 30 30
7 김기팔 80 25 25
8 이상수 65 70 70
9 강창수 95 90 9
```

2. 파생변수 만들기

- 기존의 변수를 조합하거나 함수를 이용해 새 변수를 만들어 분석할 수도 있는데 이를 파생변수(derived variable)라고 함
- 예를 들어, 복수 변수의 평균을 계산하거나 조건문을 이용하여 파생변수를 만들 시 다음(b3-ch1-3.Py)과 같이 명령어를 입력

b3-ch1-3.Py(계속)

```
# 파생변수 만들기
df1['mean'] = (df1['prin'] + df1['micro'] + df1['macro'])/3
print("Data Frame of df1 is ;",f'\n{df1}\n')

# 조건문 이용하여 파생변수 만들기
df1['P/F'] = np.where(df1['mean']>=60, 'pass', 'fail')
print("Data Frame of df1 is ;",f'\n{df1}\n')

df1['grade'] = np.where(df1['mean']>=90, 'A',
            np.where(df1['mean']>=80, 'B',
            np.where(df1['mean']>=70, 'C',
            np.where(df1['mean']>=60, 'D', 'F'))))
print("Data Frame of df1 is ;",f'\n{df1}\n')
```

b3-ch1-3.Py(계속) 실행결과

```
Data Frame of df1 is ;
  name  prin  micro  macro       mean
0 김기훈    98     82     95  91.666667
1 박수동   100     92     80  90.666667
2 원선희    50     45     75  56.666667
3 위계영    50    100    100  83.333333
4 최동팔    80     95     95  90.000000
5 최종열    90     60     60  70.000000
6 최종수    30     30     30  30.000000
7 김기팔    80     25     25  43.333333
8 이상수    65     70     70  68.333333
9 강창수    95     90     90  91.666667
```

b3-ch1-3.Py(계속) 실행결과

Data Frame of df1 is ;

	name	prin	micro	macro	mean	P/F
0	김기훈	98	82	95	91.666667	pass
1	박수동	100	92	80	90.666667	pass
2	원선희	50	45	75	56.666667	fail
3	위계영	50	100	100	83.333333	pass
4	최동팔	80	95	95	90.000000	pass
5	최종열	90	60	60	70.000000	pass
6	최종수	30	30	30	30.000000	fail
7	김기팔	80	25	25	43.333333	fail
8	이상수	65	70	70	68.333333	pass
9	강창수	95	90	90	91.666667	pass

b3-ch1-3.Py(계속) 실행결과

Data Frame of df1 is ;

	name	prin	micro	macro	mean	P/F	grade
0	김기훈	98	82	95	91.666667	pass	A
1	박수동	100	92	80	90.666667	pass	A
2	원선희	50	45	75	56.666667	fail	F
3	위계영	50	100	100	83.333333	pass	B
4	최동팔	80	95	95	90.000000	pass	A
5	최종열	90	60	60	70.000000	pass	C
6	최종수	30	30	30	30.000000	fail	F
7	김기팔	80	25	25	43.333333	fail	F
8	이상수	65	70	70	68.333333	pass	D
9	강창수	95	90	90	91.666667	pass	A

1. Py에서 외부 파일 불러오기

- **pandas 패키지는 텍스트, CSV, Excel, Stata 등 많은 프로그램에서 사용하는 데이터를 가져오고 내보낼 수 있는 기능을 제공하고 있음**

파일 형태	Method(Import)	Method(Export)
Text 파일 (TXT)	read_table	to_table
CSV 파일 (CSV)	read_csv	to_csv
MS Excel 파일 (XLS 및 XLSX)	read_excel	to_excel,
Stata 파일 (DTA)	read_stata	to_stata,

b3-ch1-3-import.Py

```
import pandas as pd
df_txt = pd.read_table("http://kanggc.iptime.org/book/data/sample1-1.txt")
df_txt
df_txt.info()
df_csv = pd.read_csv("http://kanggc.iptime.org/book/data/csv_sample1.csv")
df_csv
df_csv.head()
df_dta = pd.read_stata("http://kanggc.iptime.org/book/data/chap11-3-2.dta")
df_dta
df_xls = pd.read_excel("http://kanggc.iptime.org/book/data/score.xlsx")
df_xls
```

b3-ch1-3-import.Py 실행결과

```
 #  Column      Non-Null Count  Dtype
 0  year        17 non-null     int64
 1  gdp         17 non-null     float64
 2  consumption 17 non-null     float64
 year     gdp  consumption
0 2000  635184.6    413461.2
1 2001  688164.9    460668.2
2 2002  761938.9    515616.0
3 2003  810915.3    535967.4
4    2004  876033.1    562020.2
  var1  var2
0 75.0  82.0
1 71.0  73.0
2 52.0  59.0
3 46.0  48.0
4 70.0  69.0
5    83.0  93.0
 이름 경제원론 미시경제 거시경제
0 김기훈   98   82   95
1 박수동  100   90   80
2 원선희   50   45   75
3 위계영   50  100  100
4 최동팔   80   95   95
5 최종열   90   60   60
6 최종수   30   30   30
7 김기팔   80   25   25
8 이상수   65   70   70
9 강창수   95   90   90
```

2. Py에서 Excel 파일 만들기

- Py에서 작업한 데이터를 Excel 데이터 파일로 저장하면 Excel에서 그 파일을 불러올 수 있음
- 예를 들어 Py에서 학생 10명의 경제원론, 미시경제, 거시경제 성적을 입력하여 데이터를 만든 후 합계(sum), 평균(mean), 합격/불합격(PF) 판단, 등급(grade) 부여하고, Excel 데이터 파일로 저장
- 이렇게 만들어진 Excel 데이터 파일은 Excel을 실행하여 불러와서 사용

b3-ch1-3-export.Py

```
import pandas as pd
import numpy as np
# 데이터 프레임 만들기
df1 = pd.DataFrame({'name'  :['김기훈','박수동','원선희','위계영','최동팔','최종열','최종수','김기팔','이상수','강창수'],
            'prin' :[98,100,50,50,80,90,30,80,65,95],
            'micro' :[82,92,45,100,95,60,30,25,70,90],
            'macro' :[95,80,75,100,95,60,30,25,70,90]})
df1
print("Data Frame of df1 is :",  f'\n{df1}\n')
# 파생변수 만들기
df1['mean'] = (df1['prin'] + df1['micro'] + df1['macro'])/3
print("Data Frame of df1 is ;",f'\n{df1}\n')
# 조건문 이용하여 파생변수 만들기
df1['P/F'] = np.where(df1['mean']>=60, 'pass', 'fail')
print("Data Frame of df1 is ;",f'\n{df1}\n')
df1['grade'] = np.where(df1['mean']>=90, 'A',
          np.where(df1['mean']>=80, 'B',
          np.where(df1['mean']>=70, 'C',
          np.where(df1['mean']>=60, 'D', 'F'))))
print("Data Frame of df1 is ;",f'\n{df1}\n')
# to_xlsx 명령어로 지정된 디렉토리에 지정된 이름의 엑셀파일로 저장
df1.to_excel(r'D:/Backup/BOOK/PyBasics/PyEda/code/df1.xlsx', index=False)
```

	A	B	C	D	E	F	G	H
1	name	prin	micro	macro	sum	mean	PF	grade
2	김기훈	98	82	95	275	91.66667	pass	A
3	박수동	100	92	80	272	90.66667	pass	A
4	원선희	50	45	75	170	56.66667	fail	F
5	위계영	50	100	100	250	83.33333	pass	B
6	최동팔	80	95	95	270	90	pass	A
7	최종열	90	60	60	210	70	pass	C
8	최종수	30	30	30	90	30	fail	F
9	김기팔	80	25	25	130	43.33333	fail	F
10	이상수	65	70	70	205	68.33333	pass	D
11	강창수	95	90	90	275	91.66667	pass	A
12								

1. 변수 변환 및 변수명 변경

- 변수 변환이란 기존의 변수를 이용하여 새로운 변수를 만드는 것임

• 예 : gdp 및 consumption에 자연로그를 취하여 lgdp 및 lconsumption 변수를 생성

lgdp = np.log(gdp)

lconsumption=np.log(consumption).

- 변수명 변경은 하나의 변수명을 변경하거나 전체 변수명을 변경

• 예를 들어 df 데이터에 year, gdp, consumption 등 3개 변수가 있을 경우

• df_new = df.rename(columns = {'consumption' : 'cons'}) # consumption을 cons로 변경

• df_all = df.rename(columns = {'gdp':'y', 'consumption' : 'c'}) # gdp 및 consumption을 y 및 c로 변경

b1-ch2-5.Py

```
import pandas as pd
import numpy as np
df = pd.read_excel("http://kanggc.iptime.org/book/data/sample1-n.xlsx")
df.head()
print("Data Frame of df is : ",f'\n{df.head()}\n')
# yearly time series starting from 2000 to 2016:
# df.index = pd.date_range(start='2000', end='2017', freq='Y').year
# df.head()
# print("Data Frame of df is : ",f'\n{df.head()}\n')
year = df['year']
year
gdp = df['gdp']
gdp
consume = df['consumption']
consume
# 자연로그 변수 만들기
lgdp = np.log(gdp) # numpy packaage의 log 활용
lgdp.head()
print("Data of log(gdp) is : ",f'\n{lgdp.head()}\n')
lconsume = np.log(consume) # numpy packaage의 log 활용
lconsume.head()
print("Data of log(consume) is : ",f'\n{lconsume.head()}\n')
# 단일 변수명 바꾸기
df_new = df.rename(columns = {'consumption' : 'cons'} ) # consumption을 cons로 수정
df_new
# 복수 변수명 바꾸기
df_all = df.rename(columns = {'gdp':'y', 'consumption' : 'c'} ) # gdp 및 consumption을 y 및 c로 수정
df_all
```

2. 부분자료 추출

- 데이터의 일부분을 추출할 수도 있는데 예를 들어, (b1-ch2-7.py)를 실행하면 2000년대(2000년-2009년)와 2010년대(2010년-2016년) 데이터 추출

b1-ch2-7.Py

```
import pandas as pd
df = pd.read_excel("http://kanggc.iptime.org/book/data/sample1-n.xlsx")
df

# 두 개의 sub data를 생성
df1 = df.iloc[0:10,0:3]
df1
df2 = df.iloc[10:17,0:3]
df2

print("Data Frame of df is : ",f'\n{df}\n')
print("Data Frame of df1 is : ",f'\n{df1}\n')
print("Data Frame of df2 is : ",f'\n{df2}\n')
```

b1-ch2-7.Py실행결과

Data Frame of df is :

	year	gdp	consumption
0	2000	635184.6	413461.2
1	2001	688164.9	460668.2
2	2002	761938.9	515616.0
3	2003	810915.3	535967.4
4	2004	876033.1	562020.2
5	2005	919797.3	602345.4
6	2006	966054.6	643408.0
7	2007	1043257.8	691740.4
8	2008	1104492.2	740804.6
9	2009	1151707.8	769588.6
10	2010	1265308.0	819821.2
11	2011	1332681.0	873522.7
12	2012	1377456.7	911938.2
13	2013	1429445.4	942267.2
14	2014	1486079.3	972925.0
15	2015	1564123.9	1006005.6
16	2016	1637420.8	1047482.4

b1-ch2-7.Py실행결과

Data Frame of df1 is :

	year	gdp	consumption
0	2000	635184.6	413461.2
1	2001	688164.9	460668.2
2	2002	761938.9	515616.0
3	2003	810915.3	535967.4
4	2004	876033.1	562020.2
5	2005	919797.3	602345.4
6	2006	966054.6	643408.0
7	2007	1043257.8	691740.4
8	2008	1104492.2	740804.6
9	2009	1151707.8	769588.6

Data Frame of df2 is :

	year	gdp	consumption
10	2010	1265308.0	819821.2
11	2011	1332681.0	873522.7
12	2012	1377456.7	911938.2
13	2013	1429445.4	942267.2
14	2014	1486079.3	972925.0
15	2015	1564123.9	1006005.6
16	2016	1637420.8	1047482.4