상명대학교
SANGMYUNG UNIVERSITY

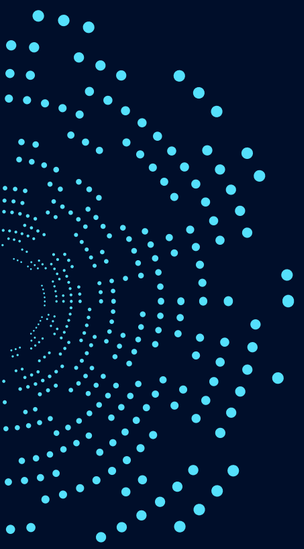
파이썬 빅데이터시각화

전준헌 · 이우진 교수

파이썬빅데이터시각화

상명대학교 SANGMYUNG UNIVERSITY

7주차. Numpy 활용 및 Pandas 활용



파이썬빅데이터시각화

상명대학교 SANGMYUNG UNIVERSITY

1 학습목표

1. numpy 개념을 이해하고 활용할 수 있다
2. Pandas에서 제공하는 series 자료구조를 이해하고,
   데이터를 사용해 series를 생성할 수 있다
3. Series를 사용하여 데이터를 조작할 수 있다
4. Series를 활용한 데이터 분석을 통해 원하는 결과를 이끌어낼 수 있다

2 학습내용

1. Numpy (5)
2. Pandas (1)
3. Pandas (2)

1강. Numpy (5)

1. Numpy (5)

상명대학교 SANGMYUNG UNIVERSITY

1 실습

- 다음을 수행하시오
  - [7, 5, 9, 10]과 [4, 10, 6, 2]인 2개의 배열을 생성
    - 1) 각 배열의 값을 3배로 만든 다음 더한 결과를 출력하시오

1. Numpy (5)

상명대학교 SANGMYUNG UNIVERSITY

1 실습

- 다음을 수행하시오
  - [[7, 5, 9, 10]과 [4, 10, 6, 2]]과 [[1, 3, 4, 9], [14, 7, 6, 4]]인 2개의 배열을 생성
    - 2) 앞의 배열에서 뒤의 배열을 뺀 뒤 각 값이 양수인지 아닌지를 True, False로 출력하시오

1. Numpy (5)

1 실습

- 1~12의 정수로 이루어진 (3,4) A와
  linspace로 1~12의 실수 3개로 이루어진 (3,1) B를 생성하시오.
- C=A+B와 C'=B+A를 계산하고 C와 C'가 동일한 지 비교하시오.

- 3x4x4의 임의의 실수 배열을 생성한 후 최대값 과 최소값 찾아보고,
  최대값과 최소값의 위치를 출력 하시오.
  - 단, 임의의 실수는 0이상 1미만.

1. Numpy (5)

1 실습

- 다음을 계산하시오 (-1이상 20미만의 임의의 정수 3X3X4 행렬 생성)
  - 전체 행렬의 합, 평균과 중간값
  - 2번째 행렬의 합, 최소값, 최대값
  - 1,4번째 열을 제외한 부분 행렬의 최대, 최소, 평균, 중간값

1. Numpy (5)

1 실습

- 다음을 수행하시오
  - 1~10의 정수로 1D array를 생성한 후, np.random.shuffle()을 이용하여 행렬을 뒤섞은 다음, 행렬을 정렬하시오
  - 1~30의 정수로 행렬을 생성하여, np.random.shuffle()을 이용하여 행렬을 섞은 다음, (3,10) 행렬도 만든 후, axis 값을 바꾸어 가면서 정렬의 결과를 확인하시오
  - 위 3X10 행렬을 중복을 제거하고 1차원으로 정렬하시오

2강. Pandas (1)

2. Pandas (1)

1 Series의 이해와 생성

Pandas 소개

- 파이썬에서 사용하는 데이터 분석 라이브러리로 '판다스'라고 읽음
- 다차원으로 구조화된 데이터를 뜻하는 계량 경제학 용어인 Panel data와 파이썬 데이터 분석인 Python data analysis에서 따온 이름
- 안정적으로 대용량의 데이터를 처리하는데 편리한 도구
- NumPy의 고성능 배열 계산 기능과 스프레드시트, SQL과 같은 관계형 데이터베이스의 데이터 조작 기능을 조합한 것
- Series와 dataframe 자료구조를 제공함
  - Series: list와 dictionary의 장점을 섞어 놓은 듯한 자료구조
  - DataFrame: 행과 열로 이루어진 2차원 형태의 자료구조
- Pandas의 기능을 이용해 데이터의 재배치와 집계, 부분집합 구하기 등을 보다 쉽게 할 수 있음

2. Pandas (1)

2 Series 개요

- Series: 1차원 배열 + index
  - Index: values를 선택할 때 주소 역할을 하는 배열(값이 모두 달라야 함)
  - Values: 데이터 부분에 해당하는 배열
- 1차원 배열 vs. Series: list vs. dictionary와 비슷
  - 1차원 배열/list: index 번호(자동)로 값 접근
  - Series/Dictionary: index명 / key(지정)으로 값 접근

2. Pandas (1)

3 Series 생성

- Values만 입력하는 방법
  - 생성 방법: s = Series(list/array)
  - Index 값은 0, 1, 2, …, 로 자동 생성
  - 예시

```
import pandas as pd

score = [84, 21, 87, 100, 59, 46]

s = pd.Series(score)

print(s)
```

```
0     84
1     21
2     87
3    100
4     59
5     46
dtype: int64
```

2. Pandas (1)

3 Series 생성

- Index + Values 입력하는 방법
  - 생성 방법: s = Series(list/array, index=list/array)
  - Index는 주어진 list나 array로 지정
  - 예시

```
import pandas as pd

names = ['철수', '영이', '길동', '미영', '순이', '철이']

score = [84, 21, 87, 100, 59, 46]

s = pd.Series(score, index=names)

print(s)
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
dtype: int64
```

2. Pandas (1)

3 Series 생성

- Dictionary를 이용하는 방법
  - 생성 방법: s = Series(dictionary)
  - 예시

```
import pandas as pd

dic = {'철수':84, '영이':21, '길동':87, '미영':100, '순이':59, '철이':46}

s = pd.Series(dic)

print(s)
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
dtype: int64
```

2. Pandas (1)

4 Series의 산술 연산

- 덧셈
  - Array간 덧셈: score1 + score2 → 순서대로 하나씩 더함
  - Series간 덧셈: s0 + s1 → 순서와 상관없이 같은 index명을 갖는 값끼리 더함
    - Values만 연산에 관여함
    - Index가 같은 값끼리 연산 수행 → 데이터 관리에 유리
  - 뺄셈, 곱셈 등도 덧셈과 같은 방식으로 처리

2. Pandas (1)

4 Series의 산술 연산

■예시

```
import numpy as np

import pandas as pd

names1 = np.array(['철수', '영이', '길동', '미영', '순이', '철이'])

score1 = np.array([84, 21, 87, 100, 59, 46])

names2 = np.array(['길동', '철수', '영이', '철이', '순이', '미영'])

score2 = np.array([99, 87, 87, 84, 77, 15])

s1 = pd.Series(score1, index=names1)

s2 = pd.Series(score2, index=names2)
```

```
s1

철수      84
영이      21
길동      87
미영     100
순이      59
철이      46
dtype: int64

s2

길동      99
철수      87
영이      87
철이      84
순이      77
미영      15
dtype: int64
```

2. Pandas (1)

4 Series의 산술 연산

■예시

```
s1
```

```
철수    84
영이    21
길동    87
미영   100
순이    59
철이    46
dtype: int64
```

```
s2
```

```
길동    99
철수    87
영이    87
철이    84
순이    77
미영    15
dtype: int64
```

```
s1 + 10
```

```
철수    94
영이    31
길동    97
미영   110
순이    69
철이    56
dtype: int64
```

```
s1 + s2
```

```
길동    186
미영    115
순이    136
영이    108
철수    171
철이    130
dtype: int64
```

```
s1 - s2
```

```
길동   -12
미영    85
순이   -18
영이   -66
철수    -3
철이   -38
dtype: int64
```

```
(s1 + s2) / 2
```

```
길동    93.0
미영    57.5
순이    68.0
영이    54.0
철수    85.5
철이    65.0
dtype: float64
```

2. Pandas (1)

5 Series에서 부분 정보 선택하기

- Index번호를 사용한 부분 정보 선택
- 예시

```
s1
```

```
철수    84
영이    21
길동    87
미영   100
순이    59
철이    46
dtype: int64
```

```
s1[2]
```

```
87
```

```
s1[2:]
```

```
길동    87
미영   100
순이    59
철이    46
dtype: int64
```

```
s1[:3]
```

```
철수    84
영이    21
길동    87
dtype: int64
```

```
s1[::2]
```

```
철수    84
길동    87
순이    59
dtype: int64
```

```
s1[1:3]
```

```
영이    21
길동    87
dtype: int64
```

2. Pandas (1)

5 Series에서 부분 정보 선택하기

- Index번호를 사용한 부분 정보 선택
- 예시

```
s1

철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
dtype: int64
```

```
s1['영이']

21
```

```
s1['영이':'순이']

영이     21
길동     87
미영    100
순이     59
dtype: int64
```

```
s1['미영':]

미영    100
순이     59
철이     46
dtype: int64
```

```
s1[:'길동']

철수     84
영이     21
길동     87
dtype: int64
```

```
s1[:'길동':2]

철수     84
길동     87
dtype: int64
```

```
s1['철이':'길동':-1]

철이     46
순이     59
미영    100
길동     87
dtype: int64
```

2. Pandas (1)

6 Series에 값 추가하기

- Index명을 사용하여 값 추가
- 예시

```
s3 = s1
```

```
s3
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
dtype: int64
```

```
s3['슬기'] = 98
```

```
s3
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
슬기     98
dtype: int64
```

2. Pandas (1)

7 Series에 값 수정하기

- Index번호와 index명을 사용하여 값 수정
- 예시

```
s3
```

```
철수    84
영이    21
길동    87
미영   100
순이    59
철이    46
슬기    98
dtype: int64
```

```
s3[2] = 88
```

```
s3
```

```
철수    84
영이    21
길동    88
미영   100
순이    59
철이    46
슬기    98
dtype: int64
```

```
s3['길동'] = 87
```

```
s3
```

```
철수    84
영이    21
길동    87
미영   100
순이    59
철이    46
슬기    98
dtype: int64
```

2. Pandas (1)

8 Series에서 값 삭제하기

- Index명을 사용하여 값 삭제
- 예시

```
s3
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
슬기     98
dtype: int64
```

```
del s3['철이']
```

```
s3
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
슬기     98
dtype: int64
```

2. Pandas (1)

9 논리 연산과 filtering

■예시

```
s1
```

```
철수     84
영이     21
길동     87
미영    100
순이     59
철이     46
dtype: int64
```

```
s2
```

```
길동     99
철수     87
영이     87
철이     84
순이     77
미영     15
dtype: int64
```

```
x = s1 > 85
```

```
x
```

```
철수    False
영이    False
길동     True
미영     True
순이    False
철이    False
dtype: bool
```

```
s1[x]
```

```
길동     87
미영    100
dtype: int64
```

```
s2[x]
```

```
길동     99
미영     15
dtype: int64
```

3강. Pandas (2)

3.Pandas (2)

1 실습

- 학생들의 이름, 국어성적, 영어성적, 수학성적을 다음과 같이 입력 받아 각 성적에 대한 Series를 만드시오

```
학생들의 이름 입력(,로 구분): 영희,철수,미나,순이
학생들의 국어성적 입력(,로 구분): 20,30,70,50
학생들의 영어성적 입력(,로 구분): 40,25,80,30
학생들의 수학성적 입력(,로 구분): 15,25,70,75
국어성적
영희    20
철수    30
미나    70
순이    50
dtype: int64
영어성적
영희    40
철수    25
미나    80
순이    30
dtype: int64
수학성적
영희    15
철수    25
미나    70
순이    75
dtype: int64
```

3.Pandas (2)

상명대학교 SANGMYUNG UNIVERSITY

1 실습

- 문제. 학생들의 국어, 영어, 수학 성적 Series에 대하여 다음 작업을 수행하시오
  - 각 학생들의 국어, 영어, 수학 성적의 합계 구하기
  - 국어 성적이 70점 이상인 학생들의 영어 성적 구하기
  - 수학 성적이 70점 이상인 학생들의 영어 성적 구하기
  - 국어 성적과 영어 성적의 차이 구하기
  - 수학 성적과 영어 성적의 차이 구하기
  - 국어 성적이 30점 이하인 학생들의 수학 성적 구하기
  - 영어 성적이 30점 이하인 학생들의 수학 성적 구하기