Pandas(판다스)

1. 설치

pip install pandas

1. 시리즈(series)

시리즈 클래스는 1차원 배열의 값(values)에 각 값에 대응되는 인덱스(index)를 부여

>>> import pandas as pd

>>> sr = pd.Series([1000,2000,3000,4000], index=['피자', '치킨', '콜라','맥주'])

>>> print(sr)

피자 1000

치킨 2000

콜라 3000

맥주 4000

dtype: int64

>>> sr.values

array([1000, 2000, 3000, 4000], dtype=int64)

>>> sr.index

Index(['피자', '치킨', '콜라', '맥주'], dtype='object')

>>> sr = pd.Series({"United Kingdom":"London", "India":"New Delhi", "United States":"Washington", "Belgium":"Brussels"})

>>> sr

United Kingdom London

India New Delhi

United States Washington

Belgium Brussels

dtype: object

1. 데이터프레임(dataFrame)

* 데이터프레임 이해

>>> values = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

>>> index = ['one', 'two', 'three']

>>> columns = ['A', 'B', 'C']

>>> df = pd.DataFrame(values, index=index, columns=columns)

>>> df

A B C

one 1 2 3

two 4 5 6

three 7 8 9

>>> print(df.index)

Index(['one', 'two', 'three'], dtype='object')

>>> print(df.columns)

Index(['A', 'B', 'C'], dtype='object')

>>> print(df.values)

[[1 2 3]

[4 5 6]

[7 8 9]]

* 데이터프레임 생성

데이터프레임은 2차원 리스트를 매개변수로 전달.

2차원이므로 행방향 인덱스(index)와 열방향 인덱스(column)가 존재. 행과 열을 가지는 자료구조.

시리즈가 인덱스(index)와 값(values)으로 구성된다면, 데이터프레임은 열(columns)까지 추가되어 열(columns), 인덱스(index), 값(values)으로 구성

>>> data = [

['1000', 'Steve', 90.72],

['1001', 'James', 78.09],

['1002', 'Doyeon', 98.43],

['1003', 'Jane', 64.19],

['1004', 'Pilwoong', 81.30],

['1005', 'Tony', 99.14],

]

>>> df = pd.DataFrame(data)

>>> df

0 1 2

0 1000 Steve 90.72

1 1001 James 78.09

2 1002 Doyeon 98.43

3 1003 Jane 64.19

4 1004 Pilwoong 81.30

5 1005 Tony 99.14

>>> df = pd.DataFrame(data, columns=['학번', '이름', '점수'])

>>> df

학번 이름 점수

0 1000 Steve 90.72

1 1001 James 78.09

2 1002 Doyeon 98.43

3 1003 Jane 64.19

4 1004 Pilwoong 81.30

5 1005 Tony 99.14

>>> df = pd.DataFrame(data=[4,5,6,7], index=range(0,4), columns=['A'])

>>> df

A

0 4

1 5

2 6

3 7

* 딕셔너리로 생성

>>> data = { '학번' : ['1000', '1001', '1002', '1003', '1004', '1005'],

'이름' : [ 'Steve', 'James', 'Doyeon', 'Jane', 'Pilwoong', 'Tony'],

'점수': [90.72, 78.09, 98.43, 64.19, 81.30, 99.14]}

>>> df = pd.DataFrame(data)

>>> df

학번 이름 점수

0 1000 Steve 90.72

1 1001 James 78.09

2 1002 Doyeon 98.43

3 1003 Jane 64.19

4 1004 Pilwoong 81.30

5 1005 Tony 99.14

* 조회

데이터프레임에서 원하는 구간 조회

df.head(n) - 앞 부분을 n개만 보기

df.tail(n) - 뒷 부분을 n개만 보기

df['열이름'] - 해당되는 열을 확인

>>> print(df.head(3))

학번 이름 점수

0 1000 Steve 90.72

1 1001 James 78.09

2 1002 Doyeon 98.43

>>> print(df.tail(3))

학번 이름 점수

3 1003 Jane 64.19

4 1004 Pilwoong 81.30

5 1005 Tony 99.14

>>> print(df['학번'])

0 1000

1 1001

2 1002

3 1003

4 1004

5 1005

Name: 학번, dtype: object

* 외부 데이터 읽기

Pandas는 CSV, 텍스트, Excel, SQL, HTML, JSON 등 다양한 데이터 파일을 읽고 데이터 프레임을 생성할 수 있다.

<sample1.csv>

col0,col1,col2,col3

row1,aa,ee,ii

row2,bb,ff,jj

row3,cc,gg,kk

row4,dd,hh,ll

>>> df=pd.read\_csv('sample1.csv')

>>> df

col0 col1 col2 col3

0 row1 aa ee ii

1 row2 bb ff jj

2 row3 cc gg kk

3 row4 dd hh ll

>>> df.index

RangeIndex(start=0, stop=4, step=1)

>>> df.columns

Index(['col0', 'col1', 'col2', 'col3'], dtype='object')

>>> df.values

array([['row1', 'aa', 'ee', 'ii'],

['row2', 'bb', 'ff', 'jj'],

['row3', 'cc', 'gg', 'kk'],

['row4', 'dd', 'hh', 'll']], dtype=object)

* 외부파일에 쓰기

df.to\_csv('C:\Temp\Output.csv')

1. Matplotlib

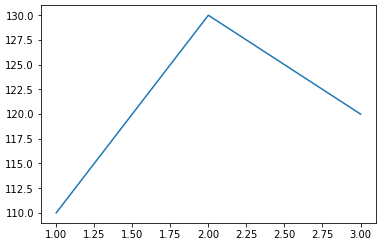
참고 사이트

<https://matplotlib.org/gallery.html>

from matplotlib import pyplot as plt

plt.plot([1,2,3], [110,130,120])

plt.show()



from matplotlib import pyplot as plt

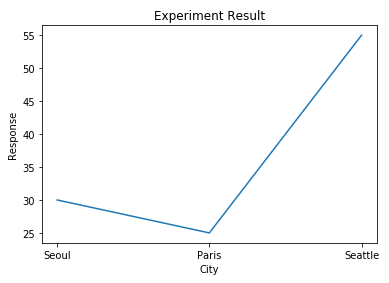
plt.plot(["Seoul","Paris","Seattle"], [30,25,55])

plt.xlabel('City')

plt.ylabel('Response')

plt.title('Experiment Result')

plt.show()



from matplotlib import pyplot as plt

plt.plot([1,2,3], [1,4,9])

plt.plot([2,3,4],[5,6,7])

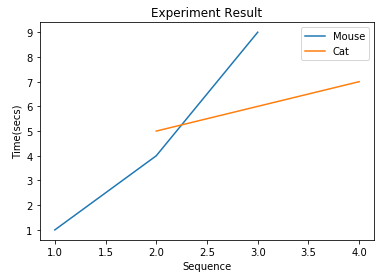
plt.xlabel('Sequence')

plt.ylabel('Time(secs)')

plt.title('Experiment Result')

plt.legend(['Mouse', 'Cat'])

plt.show()



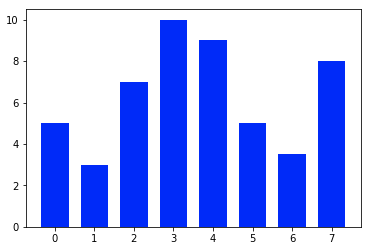
from matplotlib import pyplot as plt

y = [5, 3, 7, 10, 9, 5, 3.5, 8]

x = range(len(y))

plt.bar(x, y, width=0.7, color="blue")

plt.show()



#머신러닝 scikit 설치

>pip3 install -U scikit-learn scipy scikit-image

#에디터 jupyter 설치

>pip3 install jupyter

#에디터 jupyter 실행

>jupyter notebook

1. struct 모듈

바이너리 데이터를 int, float, char 등 다양한 타입으로 변환하여 받은 바이너리

데이터를 처리할 수 있다.

\*struct.unpack('format', 바이너리값)

format 문자

'>': 빅 엔디안

'I' : unsigned int(4)

'B': unsigend char(1)

2. 람다

1)기본사용

>>>f1 = lambda x: x\*\*2

>>>print(f1(2))

4

>>>print(f1(3))

9

>>>f2 = lambda x, y: x+y

>>>print(f2(1,2))

3

2)map

>>> a = [1,2,3,4]

>>> b = [17,12,11,10]

>>> list(map(lambda x, y:x+y, a,b))

[18, 14, 14, 14]

3)filter

>>> foo = [2, 18, 9, 22, 17, 24, 8, 12, 27]

>>> list( filter(lambda x: x % 3 == 0, foo) )

[18, 9, 24, 12, 27]

3. 정규식

1) ^[a-z0-9\_-]{3,16}$

^:문자열의 시작

$: 문자열의 끝

[a-z]: 소문자 a~z

[0-9]: 숫자 0~9

[\_-]: \_, -

{3, 16}: 문자의 길이가 3 ~ 16

2) ^#?([a-f0-9]{6}|[a-f0-9]{3})$

?: 앞의 문자가 와도 되고 안와도 되고, 이 코드에서는 앞에 #이 있으므로

#으로 시작해도 되고 아니어도 되고를 의미함

[a-f0-9]{6}: a~f, 0~9 요소들을 섞어 6글자

| : or, 또는

[a-f0-9]{3}: a~f, 0~9 요소들을 섞어 3글자

3) ^[a-z0-9-]+$

+: 한개 이상

소문자, 숫자, 하이픈(-)이 한개 또는 한개이상

4) ^([a-z0-9\_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$

이메일 정규화

\d: any number