part 9

numpy - 라이브러리 .import해서 불러와야함.

중요한 함수들은 그 함수들이 모여있는 라이브러리를 불러와야함.

함수가 들어있는 어떤 라이브러리를 . 패키지.

numpy 라는 패키지 라이브러리 속에 또 작은게 들어있을수 있음, 그 밑에 패키지 이름 A, B 해서 만들고 또 그안에 만들수도.

(점을 함수라고 생각)

numpy 안에 내부적으로 함수가 존재할수 있음. A안에 패키지가 또있다면. 패키지. 내부적으로 안에 있는걸 부를때는 " numby.A.D.어떤함수". numpy가 젤 위에 있는 상위개념이고 그 안에 또 패키지... 점이 그런 용도로 쓰임.

1) import numpy (np안에 있는 건 쓸수있음)
   numpy-A-D-f를 쓰고 싶으면 import np를 했으니까
   numpy.A.D.f라고 쓰면됨> (너무 복잡)

>불러올 때 젤 큰 낍데기를 불러오는 방법.

2) from Numpy Import A (> numpy 에 있는 A를 불러오자) (from을 쓸수도)
  A.D.f (그러면 A를 그때부터 쓸수있음_)
  from Numpy import A.D (이렇게도 가능)

>from 으로 import할 수도 있음

사진

In [2]

np라는 젤 큰거를 불러옴.

import해서 들어오고 numpy를 줄여서 쓰고 싶을 때 as.

np 라고 줄여서 쓰고 싶을 때 as

mat를 불러움,

matplotlib가 젤 위에 있고(껍데기) 그안에 포함되있는게 - pyplot가 속해있음. 포함되어 있는 subset을 plt로 받아온다.

or

쓸 수 있는 다른 방법중 틀린 것-시험

from mat import pyplot as plt

-그 다음부턴 plt쓸수 있음

(점은 포함관계로 되있다 라이브러리에)

numpy는 왜 필요? list 하고 아주 비슷한데 쓰는 이유는 수학적으로 계산도 할 수 있고 해서 처리할 모든 data는 다 np 처리를 해야

[3]

np.empty

empty는 함수 (괄호로 input을 받음)

np라는 젤 큰 라이브러리 안에 empty 라는 함수.-(그 아래에 subset 은 포함되지 않고)

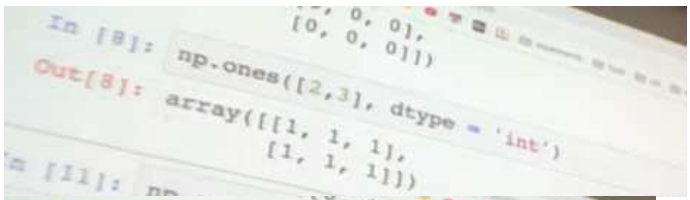
list가 입력으로 들어감.

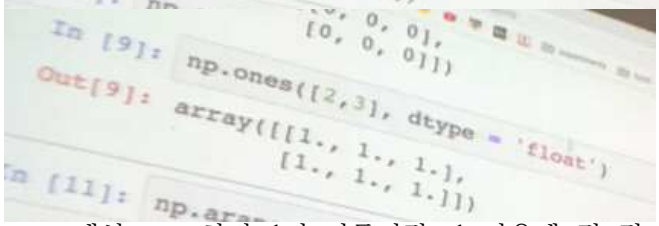
2 by 3 로 리스트로 .

가로 세로 행렬 : 직사각형의 숫자의 array

2행 3열 - 뚱뚱 옆으로
내부적으로 들어가는 것은 datatype을 .dtype을 int로 해서 하나 만들어라
empty 라는 함수는 빈. (비엇다 하지만 빈 건 아님)
out[2]
numpy속에 들어있는 함수를 이용했기 때문에 숫자가 만들어지는데 계싼가능한 숫자
2by 3의 array로 되있음. - 3개짜리 list가 하나가 있고 그게 두줄이 있음.
int : dtype을 int로 해서. 안에 든 숫자는 랜덤한 숫자. 소수점 숫자 아님.

[4]
np라이브러리 속에 있는 zeros라는 함수를 이용. 거기 입력에 list로 2곱하기 3을 넣음.
이 함수는 2by 3의 행렬을 만드는데 0으로 채워진.
** [0,0,0]
>>그냥 0이 들어가있는 백타
이걸 2by 3로 만들고 싶을때는
** [[0,0,0] , [0,0,0]]
2행이 있고 3열이있는 리스트.
>>쓸수가 없음 계산안됨.
계산이 되는 array로 만드는 방법이
np.array ([[0,0,0] , [0,0,0]])
리스트에서 array로 바꿔줌
[4]와 같음. zeros함수와 같음.
np.array - list를 array 로 convert해줘라 >위에거랑 똑같은 결과





zeros대신 ones하면 1이 만들어짐. 1 다음에 점 점이 붙은건 int가 아니라 float. (dtype = int 안했을 때)
ones라는 함수가 디폴트로 datatype을 float로.
dtype= 'int' 적어주면 점이 사라짐

float64 로해도 변화는 없음. 소수점 몇째짜리 까지 할까에 대한 정의. precision을 얼마나 밑에까지 할까에 대한 정의. - 오차를 허용하고 싶지 않을 때
64- 정교하고 싶을 때 - 숫자를 길게 하면 메모리를 많이 차지함. 1.00000...
정확도와 데이터 양 반비례.

```
In [5]: np.arange(5)
Out[5]: array([0, 1, 2, 3, 4])

In [6]: np.arange(0,10)
Out[6]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [7]: np.arange(0,10,2)
Out[7]: array([0, 2, 4, 6, 8])

In [8]: np.arange(0,10,2, dtype='float')
Out[8]: array([0., 2., 4., 6., 8.])
```

np에 있는 arange라는 함수는 계산이 될 수 있는 array를 만들어줌. 숫자가 다섯 개. 5라고 쓰는순간 index 5개가 만들어짐.(0부터)
arange (0,10) : 0부터 10까지 하면 10은 포함안되고 10 밑으로까지.
(0,10,2) : 2만큼 증가increment하면서 10미만까지 increment- 증가분
data type설정도 할수있음- float, float32, float64도 쓸수있음. - 얼마나 정확성을 높이느냐와 관련.

```
In [9]: np.linspace(0,10,6)
Out[9]: array([ 0.,  2.,  4.,  6.,  8., 10.])

In [10]: np.linspace(0,10,7)
Out[10]: array([ 0.        ,  1.66666667,  3.33333333,  5.        ,  6.66666667,
         8.33333333, 10.        ])

In [15]: X= np.array([4,5,6])
         X
Out[15]: array([4, 5, 6])

In [14]: X= np.array([[1,2,3],[4,5,6]])
         X
Out[14]: array([[1, 2, 3],
                [4, 5, 6]])

In [17]: X= np.array([[1,2],[3,4],[5,6]])
         X
Out[17]: array([[1, 2],
                [3, 4],
                [5, 6]])

In [18]: X= np.array([[[1,2],[3,4],[5,6]],[[1,2],[3,4],[5,6]]])
         X
Out[18]: array([[[1, 2],
                 [3, 4],
                 [5, 6]],

                [[1, 2],
                 [3, 4],
                 [5, 6]]])

In [20]: X. ndim
Out[20]: 3
```

X. ndim - 3차원.
[9]

linspace - linear의 준말
linspace(0,10,6) : 0부터 10까지 0, 10포함(arange와 달리). 그걸 총 6개로 똑같이 나누어 준다.
linear space - 차이가 다 똑같음. 첫째에서 둘째로 가는
[11]
np. array : list에서 array 로 바꾸는거
백터를 만들고 싶을 때,
2by3를 만들고 싶을 때.(행렬을 만들고 싶을 때)
3by 2 만들고 싶을 때.
*백터면 1차원. 직사각형 2차원. 직육면체로 되면 3차원. 2차원 행렬 두 개 있으면 3차원
대괄호 두 개- 2차원. 대괄호 하나 더 쓰고 컴마하고 같은거 쓰고 대괄호닫기. >3차원 -대괄호 3개.
차원

```
In [21]: X.shape
Out[21]: (2, 3, 2)

In [22]: X.dtype
Out[22]: dtype('int32')

In [23]: X.astype(np.float64)
Out[23]: array([[[1., 2.],
                 [3., 4.],
                 [5., 6.]],

                [[1., 2.],
                 [3., 4.],
                 [5., 6.]]])

In [25]: np.zeros_like(X)
Out[25]: array([[[0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0]]])

In [24]: X*0
Out[24]: array([[[0, 0],
                 [0, 0],
                 [0, 0]],

                [[0, 0],
                 [0, 0],
                 [0, 0]]])
```

x. shape
(2,3,2)
2곱하기 3곱하기 2 의 차원이다. (3개니까 3차원인데
-3개니까 3차원인데 3곱하기 2가 2차원짜리 직사각형. 이게 두 개가 있다. 젤 큰 괄호속에있는게 두 개. (가 첫 번째 나오는 2) 그다음괄호속에 있는 3개의 리스트 안에 2개의 숫자.
X. dtype
: x속에 나오는 숫자가 어떤 type인가 >int /32 precision과 관련
타입을 바꾸고 싶을 때 astype 이라는 function .
-np.float64로 바꿔줘라
np. zeros_like / x를 만들었는데
모든 형태를 유지한채 숫자를 0으로 바꿔주라. : zeros like

X*0 해도/ 이제 계산이 가능하니 똑같이 만들어짐.

#data = np. random ~~
np 라는 큰 라이브러리 속에 random 이라는 sub패키지 속에 있는 normal이라는 함수를 쓴다.
from np import random. normal 도 가능
normal함수 - normal distribution을 만들어주는. 정규분포. 종모양 데이터를 만들어주는. 종 모양을 만들기위해 필요한게 두가지 정보 - mean값(평균) 과 얼마나 뚱뚱한지./ 0 - mean 1- 뚱뚱 100개의 데이터 / random한 데이터. 값(나올때마다 다ㄹㄹ) / 총 백개의 데이터가 나옴.

data. ndim 은 1차원. data의 디멘션은 일차원- 대괄호 하나
data. shape- 총 100개의 숫자가 나옴.
100개의 데이터를 가지고 정규분포를 가지는지 플래팅을 할 수 있음.
mat라는 라이브러리 속에있는 plt라는 서브패키지를 import해올 것. plt - subpackage 속세 속에 hist라는 함수를 씀.- 백터 array 계싼할수있는 data를 입력으로 하고
히스토그램: 늘 옵션에 bins 바구니를 총몇개로 할건가 정함. 바구니를 10개로 만들어 0기준 한쪽으론 -. 한쪽으론 + 값. 값들이 점점 쌓이는.
x축에다가 데이터값을 공으로 생각하고 던지면 바구니 속에 쑥 들어감. 위에 공들이 쌓임.
그렇게 그래프 그리는게 히스토그램.
그림 - 바구니 총 10개. range 속에 들어가는 값들 . y값에 해당되는 건 무조건 0을 포함한 정수값 .값들을 다 합하면 100개- 시험

<11/5>
part 10.

Phasor
1초동안 얼마나 숫자가 빽빽하게 값들을 담을건가 - sampling rate
SR을 만으로 한다고 규정하면 1초동안 총 만개의 숫자를 담을거다.
1초에 얼마다 할 때 hz라는 단위.

list상에서는 어떠한 계산도 안되기 때문에 numpy 에 담아서 계산가능.
np.array [ ]
다차원의 array를 만들 수 있다. 1차원이면 백터 2차원이면 직사각형 3차원이면 volume
numpy - list에 숫자 담는것보다
np. array [리스트] 하면 np형태의 데이터로 바뀐다. >계산 가능
Sound
다양한 pure tone의 합으로 복잡한 사인.
sinusoidal-사인 코사인처럼 곡선으로 생긴. phasor- sinusoidal function을 만들어내는 것 / 사인, 코사인 펑션이 phasor이 됨. 물결모양 만들어 내는게 phasor.

싸인하고 코사인에 들어가는 입력 - sin(입력)입력값은 degrees 가 들어가면 안되고 radians 가 들어가야함. / 파이는 숫자값 3.1415...무리수. / 2곱하기 파이면 6. 몇
0은 0도, 2파이는 360도와 corresponding. / 파이는 180도. / sine, cos functiin에 들어가는 입력값은 degrees 가 들어가면 안되고 radians 가 들어가야함. / 720 두바퀴 도는거니 똑같음. 360까지 알면됨.
0부터 100파이 까지 사인 코사인 그래프를 그리면 총 몇 번의 반복? - 50번. 2파이가 반복되니까 .
>사인, 코사인 함수는 phasor.
2.
theta - 각도 - radians
theta 가 3/2 파이면 0.

파이는 무리수
sin(π/4)
오일러 공식
e도 무리수 2.71...
세타 는 입력값.
sin - 입력변하면 출력을 해주는 함수
sin cos(입력만 바꿔주면 출력하는함수)처럼 e새타i 저거도 함수 - e, I 는 fix 되있음. radian값이 변함으로써 어떤 값이 뱉어지는 함수.
세타만 변하면 어떤 값이 나옴 -e, i는 숫자. 세타에 파이(radian- 숫자) 넣으면 다 숫자 니까 숫자값이 나옴.
젤 크게 포괄할 수 있는 수의 카테고리 - 복소수.
e 저거는 새로운 phasor.
오일러 펑션의 값들은 어떻게 그릴까.
sin cos 는 숫자값이 실수라 그림그리기 가능, 오일러 펑션의 아웃풋들은 어떻게 표현할까.
-복소수를 플랫 하는 방법.
복소수 평면 (complex plane)
(a,b) : (1.0) , (0,1) , (-1,0) , (0,-1)
세타가 0도 파이/4, 파이,2 ...
함수 에 입력값을 (세타. 각도.radian) 만 넣으면 복소평면에서 점을 줌. 아무 radian 값을 넣었다 치면 /
각도값을 그려서 만들고 몇 콤마 몇 하면 저절로 0.3 + 0.8 i 이렇게 나옴
모든 데이터는 백타화 되어야 한다. 백타의 정의는 숫자열 . (,) 몇콤마 몇 하는게 다 백타. 백타값으로 표현이 됨.
sin - t가 바뀜에 따라 오르락 내리락.
오일러 - t세타가 바뀜에 따라 원을 따라 .. 0 ~파이 ~ 2파이 ~ . 계속 도는.
projection : x축 a에 project를 하면 위에서 보는. > ____ 가로로 왔다갔다 함.
y축에 project를 해서 b 보면 세로로 왔다갔다.
a가 실수 -실수의 관점에서만 보겠다 하면 위에서 보면 됨. 1에서 시작해서 왔다갔다
허수만 보겠다 하면 0에서부터 위아래로 왔다갔다 > projection의 두 방향.

실수만 볼 때, - cos 허수만 볼 때, - sin
>cos그래프는 1부터 시작 sin 은 0부터 시작
실수 - 1부터 시작해서 좌우로.(cos - 1부터 내려갔다 올라갔다) 허수 - 0부터 시작해서 위 > 아래 0부터 올라갔다가 내려가는. = sin 과 같음
오일러 phasor는 sin cos 성질을 동시에 갖는(결합물). 원하는 거에 따라 두 방향으로 projection해서 sin, cos 따로 볼 수 있다.

인풋은 공통적으로 세타- 레디언. 각도값.
prat 에 pure tone해봤는데 소리 났음. phasor첨에 define할대 들어가는 입력 중에 또 하나는
pure tone에 넣는 입력값중에 하나가 frequency (1초에 몇 번 왔다갔다 하는가)

사인 세타를 쓰고 각도값이 변한다고 할 때 시간의 개념이 안들어감 -각도값이기 때문에. (몇바퀴 도는거지 몇초에 몇바퀴 도는가는 안들어가 있음)
시간의 개념이 들어가야 1초에 몇 번 왔다갔느냐해서 소리의 높이가 결정됨.
각도개념, 초 개념을 같이 넣어줘야 진정한 소리가 나옴. 소리라는 실체는 반드시 시간의 개념이 들어있어야.

sampling rate : 1초에 총 만개의 숫자로 표현. / 음의 음질상 얼마나 고해상도로 하는가.
#

<11/7>
크게 matplotlib 가 있고
import matplotlib. pyplot
해도 똑같은거.
젤큰 라이브러리 밑에 sub라이브러리.
#parameter setting
time 은

0부터 2 파이까지 만들어짐.

part 11
phasor - cos sin phasor & 오일러 phasor (cos, sin component 동시에 가지는)
numpy 라이브러리를 import.
첫 번째 줄 . 플래팅 할 때 쓰는 라이브러릴. 큰 라이브러리 이름은 matplotlib . 큰 라이브러리 밑에 sub라이브러리. - 이것과 똑같은 역할) import matplotlib. pyplot
as plt 해서 plt 쓰면 이 속에 있는 걸 부를수있음.
from을 쓰며 그 속에 pyplot을 부름.
Phasor에서 변수들을 parameter setting -변수에다 값을 담아 놓음.
>amp 나 sr dur을 바꿀 때 이것만 고치면 결과값이 쉽게 바뀜 .이런게 parameter
time을 왜 만들까 - phasor 함수 사인 코사인 오일러 펑션 등은 입력값이 각도값임 (

degree 안돼 radian만) . time이 필요한이유는 각도값 만으로는 실체의 소리를 만들수없음.
각도값만 만들어서 사인 코사인으로 플래팅을 했을 때 어떤게 플랫 되는지 해볼거임.
theta = np.arange(0,2*np.pi, 0.1)
0, 0.1,0.3 해서 2파이까지 만들엇 j sin 에다가 넣으면 사인곡선이 나옴.
각도의 값들 백터를 어떻게 만들수있을까
첫 번째 값은 0 마지막 값은 2*np.pi
6.28까지 increment 디폴트는 1.
theta 는 radian으로 정의한 것. 그 값들이 사인에 하나하나 들어가서 7개의 백타값이 나옴.
@
22: 2줄 , 2열 1 2
3 4
0,0 1 , 2
y축은 그것의 사인함수의 결과.
문제는 너무 덜 빽빽함. 좀 더 빽빽하게 하려면. aragne 쓸 때 increment 디폴트 사용해서 1만큼 띄웠는데 좀 더 작게 하면 빽빽하게 됨. 0.1로 (100개)
@
x축이 뭔지 이름을 달아줄 때 - set xlabel 이라는 함수.
x축상에서 equidistance로 쪼개짐 - y축은 아님. y축에서도 eqi하는거는 linear하는 경우.
line 직선이면. y=2x처럼.(x의 변화가 equi 하면 대응하는 y도 equ9 - linear)
그림은 non-linear (x와 y의 관계가 line이 아니다. y=ax+b 제외한 모든 함수 xy의 관계는 non-linear)
곡선이 나타난다는 말 자체가 x의 equi한 성격이 y에는 반영되지 않는다.
'-'하면 라인으로도 나타남.
@
theta 0부터 2파이 까지. 끝나는 점이 10파이라면 총 5번 돔. > 바꾸고 플래팅 해보면 5번 돔
@
time tick 의 개수를 index로 먼저만듬. 1초라면 time tick의 개수는 sr과 일치.
@ theta = t * 2*np.pi * freq
time을 가지고 theta를 연결시키는 것.
fre 가 없다고 생각하면, t 이 0~1 일 때, 끝이 1인 것. -1초동안에 한바퀴도는 것을 만들어라
100바퀴 돌려고 만들려면 바퀴에 해당되는 fre를 곱해주는.
@ s = amp*np.sin(theta)
아깐 arbitary 하게 만드는 theta를 넣었지만 이젠 time이 연동된 theta를 sin에 넣어서 s singal을 만듬 > theta, t, s 세 개가 다 있음.
세 개를 어떻게 플래팅 할까
원하면 s, theta / time, s를 플래팅 할 수 잇음.
theta 가 어떻게 변하는지는 - 한바퀴가 이파이 2바퀴가 4파이. / 그게 몇초동안에 흐름이 있는가가 관심. 실제 플랫할대는 x축에 time을 씀.
100까지 하면 듬성

sr은 1초에 몇 개가 들어나는지 - 1초라는 말이 들어가면 time과 관련있는거.

2차원 - 머 콤마 머 해서 2차원의 숫자가 들어가는 숫자가 두 개있는 백타.

part 12

pure tone sine wave 만들어내는데 time tick 을 먼저 만들고 .
소리라고 하는 순간 시간의 개념이 반드시 들어가야.
ipd.Audio(c.real, rate=sr)I
pd ipython이라는 라이브러리 - display 라는 서브라이브러리. ipd라고 명령을 함.
complex number 의 백터중에 real 값. real값의 백터들. sr : 만 > 그거에 해당되는 소리 나옴.
parameter
amp sr (얼마나 정보를 촘촘하게 할건가/점, 숫자들이 1초동안 얼마나 나오는가) freq (얼마나 사인웨이브가 1초동안 왔다갔다 할까/shape 반복이 얼마나 왔다갔다 하는가)
sr 과 fre는 반드시 구분해야.
어떻게 amp를 장착시킬까 - 증폭. / complex wave 소용돌이 치듯이 생김. 그게 얼마나 커지는가. # generate signal by cos 에 적어줌.

그림
sr=10hz. 주어진 숫자가 10개. 이걸로 100번 왔다갔다 하는 걸 표현 할수 잇을까
아무리 동그라미를 아껴도 5번의 왔다갔다까지박에 안됨.
주어진 숫자에 개수의 표현할 수 있는 주파수는 반밖에 안됨 멕시멈이.
1초에 웨이브를 한 두 개 만드는건 쉬움.
숫자가 주어져 있으면 무한대로 표현할 수 없음.
sr의 반에 해당되는것만 표현가능. Fre = 5hz 가 Maximum.
CD가 담는 음질은 sr = 44100Hz 1초에 주어진 숫자가 44100. 사람이 들을 수 있는 가청 주파수가 2만. sr가 저정도면 경제적으로 충분히 다 표현가능- CD 음질이 저기에 고정.
NF 는 22050 hz. - 피치로 쳤을 때 아주 높은 소리. /아주 높은 소리까지 표현 가능/ 사람의 가청 주파수가 2만 / 사람이 그 위로는 들을 수 없음. /
+유선전화도 소리가 디지털로 선을 따라감. 숫자값들이 움직임. 얼마나 빽빽하게 전송을 해야하는지가 중요. sr을 얼마까지 해야하는가가 중요. / 유선전화의 sr은 8000hz. fre 4000까지 표현가능.- 낮음. - 들을 수 있는 최대. /사람이 헷갈릴대 많음 엄만지 딸인지 말소리는 다 들리지만. / spectrogram - first , second, third format 4000안에 거의 다있음. 사람의 말소리는 4000안에서 구분이 다되는데 누구인지 는 그위로 더 있음.
휴대폰은 16000hz. NF 8000 까지 . >누군지 별로 안헷갈.
sr의 half 에 해당되는 NF까지가 숫자상으로 표현할 수 있는 fre의 Max. - CD음질의 sr를 가지고 박쥐 소리를 녹음하면 저기 들어가지 않음. 고주파가 담기지 않음.
초음파: fre에 해당하는 얘기 - fre가 2만보다 훨씬 높은 소리.

F0 = 100; Fend = int(sr/2); s = np.zeros(len(t));
#초기의 s값 정의. 0으로 시작. 거기다 첫 번째 sin wave 더하고 돌고 두 번째 사인웨이브 더하고 더하고... / time 백터의 개수만큼 0을 만듬.
마지막에 오천hz까지 더해지면 s가 다 더해진 상태로 남음.
>젤 낮은 주파수 Fo. 주파수 올릴수 있는 끝은 sr/2. sr을 정함에따라 얼만큼 하는지 정해짐.
젤 첫 번째 Fo을 100으로 했다면 젤 마지막은 Fend. sr의 반= NF . 소수점이 나오면 지저분하니 int로 함(반올림 처리)
*time은 만들어졌다고 assume.
time을 만들고 -> theta만들고 -> sin에 다가 입력. / 이걸 반복하는데 fre 만 바꾸면서 반복하는게 이 코드의 핵심.
for freq in range(F0, Fend+1, F0):
# fo부터 fend 까지 늘려감. increment는 100만큼(fo만큼) +1은 젤 마지막것도 포함시키기 위해. 100부터 5000까지가면서 몇 번의 루프가 돔. 50번의 루프.
첫 번째 fre가 100값이 fre로 들어오고 두 번째는 200, 300..
    theta = t * 2*np.pi * freq
  # time x 2파이 x fre / sin의 입력이 되는 theta를 만듬.
theta만들 때 한번은 100hz로 만들고 두 번째는 200hz로 (더 빨리) , 300hz 그 phase를 만듬. 이 theta값을 사인에다 더함.
  (tmp = amp * np.sin(theta)) # 사인에다 입력을 함.
    s = s + amp * np.sin(theta) # signal 변수에다가 더하고 더하고 하는걸 쓴 것. 젤 첨 s가 있다면 더하고 그걸 다시 s에 넣어라. 한바퀴 돌고 다시 old s에서 계속 업데이트.
#젤 첨 fre가 fo일 때 들어올 때 s값이 정의 되지 않아 에러가 뜸. 처음 루프에서 젤 첨 s가 뭔가를 정의해줘야함 위에서.
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(t[0:1000], s[0:1000]); # 다 더해진 s를 플랫. 1부터 천번째까지만. time을 x축. 더해진 s값을 y축.
ax.set_xlabel('time (msec)')
ipd.Audio(s, rate=sr)
소리가 엥. 여러 사인웨이브를 harmonics로 해서 합해서 올린 것. prat으로 만든것과 같음 더 정교.

pulse train : 사인웨이브 부드러웠던 모양이 없어짐. 점점 없어짐. sr이 훨씬 많으면 더 밑으로 가 0이될 것. 선하나가 남고 000..선하나 이렇게 될 것.
pulse train :
그래프는 다 더했을 때 나오는 모양.
실제 소리와 가깝게 만들 것 담 시간에는.

**frequency, sine wave, sr의 개념을 다시 보기 + fre 랑 sine wave의 관계.

part 13
웨이브를 차곡차곡 더하면 pulse train처럼 나옴. > wave form
fre를 440hz로.
generate time > phase> signal by cosine-phasor
시간. 0.5초까지 만들었는데 0.01초까지 display.
parameter setting을 바꿔서 440hz(라 소리)를 880hz로 바꾸면 아까 소리와 높이가 같음. - 1760hz로 하면 . 다 '라'가 됨. -220,110 도 라. 배수로 하면 다 같은 음으로. 옥타브를 뜀.
s = amp*np.cos(theta)
sin 대신 cos 을 쓰면 시작점이 달라짐. - 1부터 . 1초간 몇 번 왔다각쓰다 하는지는 같음 fre는 고정이기 때문에. / 두 소리는 둘 다 '라'. 다르게 들리지 않음. / sin cos는 shape은 같지만 sin는 살짝 이동한거 -π/2, 90도 차이가 있음. / cos에서 90도 옆으로 이동하면 sin.
π/8만 이동하면 소리가 달라질까 ? 노. 얼만큼 이동하든 소리는 다 같음.
phase 에 대한 거는 우리는 인식을 못한다. frequency 변화는 느끼지만 phase shift는 아님.

complex phasor 실행하고 나오는 값이 complex 값이 나옴.
complex number 자체는 플래팅이 안됨. a ,b를 각각해서 2차원으로 플래팅하는 방법.
c에서 real 값만 x축에 imag파트를 y축에 해서 2차원상에서 찍음.
ipd.Audio(c.real, rate=sr)
>>라 에 해당하는 소리가 나옴.
>real값만 듣는 것. cos으로 받는 값이랑 c.real 이랑 같고 sin 펑션을 썻다면 c. imag와 같음. - 소리는 같음

Pulse train
F0를 하나 정하고. 오천까지(NF)
pulse train 그림은 wave form spectrum 이 아님.
spectrum - 한 타임 포인트에서의 어떤 주파수 성분이 많은지 보여주는거. 낮은 주파수부터 있는 그림. 거기서 curving을 함.
decreasing 하게 만들어야.
pulse train 만든걸 prat.

<11/19>

데이터 기계(인공지능,함수) 데이터
함수
데이터 부분은 벡터-숫자열로 되있음. 숫자열이 앞뒤 같을 필요는 없음.
음성 text
>음성이 들어가서 text 형태로 나오는 >>음성인식

text 음성 - 음성합성
일본어text 한국어text - 기계 번역
-알파고 같은 경우 바둑상태가 입력으로 들어가서 몇 번째 수를 뽑아야 하는가가 나옴.

중간에 있는게 인공지능, 함수/어떤 형태를 지닐까 -행렬의 형태
입력벡터를 출력벡터로 만들어주는 함수의 역할. -인공지능
음성 텍스트 사람이름,, 뭐든 될수있음 벡터는
모든 데이터가 벡터로 되야하는건 행렬 곱을 하기 위해서.
형렬의 형태를 거치면 다른형태가 될수있음- 이미지가 들어가면 누구의 얼굴
5 3 0 1 -1 0 2 -3 6
0 1 3
3 -5 7
2 3 4
4 x 1(세로가 몇 개있냐와 관련) 4x 3 가로가 몇 개 있냐와 관련.
젤 왼쪽 컬럼이 왼쪽에 들어가 곱하고 더해서 오른쪽벡터가 됨.
중간게 행렬함수. 왼쪽이 입력(음성) 이라 생각하고 함수를 곱하기해서 통과하고 , 행렬과 행렬의 곱. 입력이 함수를 통해서 출력이 됨.
기계가 많은 데이터로부터 학습을 해서 얻어내는 것이라고 생각 - 인공지능.
인공지능- 행렬의 곱. 입력벡터를

Linear algebra 선형대수 - 행렬
모든 인공지능이 선형대수.

<11/21>
part 15
A x = b
1 3
5 1 6 2 0 15 29 3
6 -1 3 9 1 33 19 1
33 3 -1
기계 입력 출력
3 by 2. 2 by 3__ 3 by 3
입력 부분은 반드시 백터화해서 들어감.
A부분이 잘 훈련되서 갖고있으면 잘 써먹을 수 있음.
중앙에 곱해질 수 있는건 왼쪽에 colum 개수와 row개수가 같아야함 2 by 뭐가 되야함.
인접한 두 수가 같아야 곱해짐.
6 x 1 3 x 3 해서 더함 , 6 5 + 3 1 , 6 6 3 -1 , 2,9곱한걸 또 적고.
결과값은 3 by 3. 바깥에 있는게 결과값과 같아짐.
백터도 행렬.
1 3 6 15

5 1 3 33
6 -1 33
3 by 2 2 by 1 = 3 by 1
[6
 3]을 A앞에 두면 곱해지지 않음 2 x 1, 3 x 2 / 1x2 [6 3] 으로 만들면됨.
A부분도 2 x 3 으로 만들면. [1 5 6
3 1 -1]
두 개 곱하면 [15 33 33] 아까 했던거랑 같음.
xA-1=b-1 (tranpose) 반드시 위치가 어디에 있을 필요 x
입력백터가 양쪽으로 차원만 맞으면 곱해질 수 있음. 곱해진다는건 함수 matrix를 통과해서 출력백터가 나온다. 출력, 입력 백타도 길게 있는 cv 가 될수 있고 rv가 될수도 있음.

A =
[1 2
 -1 0
 3 5]
3 by 2 matrix
column 은 3쪽. 갖고있는 성분이 3개. 점을 찍으려면 3차원에서 찍을 수 있음.
3- column vector 들이 represent 되는 세상이 whole space. col-ws : 3차원
column space 는 원점과 두점을 연결하면 plane 위에 있는 삼각형이 됨. 무한대로 밀어보면 전체를 카바하게 됨 - spanning 시키면 2차원의 plane이 나옴. : column vecotr을 표현하고 원저
cs 와 ws는 같다 ? x - 2차원,< 3차원
ws차원의 나머지 한 차원은 어디 갔느냐 ? 이걸 null space 라고 함. / left null
spanning 시키면 2차원의 plane이 나옴. = column space - cv를 표현하고 원점과 연결시키면
spanning 된 2차원이 둥글게 있다고 생각. plane. orthoganal - 수직 직각
이 plane와 orthoganal 되는 거는 한선이 있음. 원점을 관통하는 한 선. - 이 선은 일차원. - 이게 null space.
cv가 만들어 내는 space 가 있고 그거에 수직인게 null space. (3- 2 = 1)
>plane& 선이 3차원 부분을 커버.
spanning :
LC으로 표현가능한 모든걸 spanning이라고함 기하적으로 . cv 백터 두 개를 LC 하면 B가 나오는데 B가 될 수 있는 것은 뭐가 될까. , all possible Bs 가 column space가 됨. -spanning 하는 plane을 넘어서지 않음. plane을 다 이룸.
b를 곱하면 선상에서 왔다갔다 그것의 합은 평행사변형해서 만든 점. 모든 가능한 점을 찍으면 삼각형을 span한것과 동일. cs 의 정의 이해하기, 차원. ws를 채우지 못한다면 그 나머지

는 ns

row space

rowise-ws 는 2차원/(1,2) (-1,0) , (3,5) . row vector들이 만들어내는 space는 , 이것들이 spanning해내는 공간은 2차원을 넘어가지 않음. 삼각형을 확장시키면 2차원은 확보 > 2차원. cs 나 rs의 차원은 I한게 몇 개있는지 중요./ cv 에는 Independent 한 백터 두 개 = cv의 spanning된 space와 같음.

column I 2 개 -> cs와 같음. = rank / 이 matrix 의 rank 는 뭐냐

그림)

rv 가 3개. / 일직선이 아닌 세 개의 백터 , spanning해서 만들어내는게 row space.

I하다는 것은 그 space에 있는 것들의 LC로 만들어지면 안됨.

>a,b에다 뭘 곱하고 더하면 C가 됨 - dependent 한 것. >

rowwise상에서 I한개수를 찾으면 2개.

column 으로 하든 row로 하든 I 한 백터들의 숫자는 늘 같음 - rank

3 by 2 matrix에서 3은 columize- ws , 2는 rowize - ws. rank는 2.

2 2 만큼이 I.

(2차원 2차원)

2 = 2 = I 한것의 개수

1(3차원에서 평면빼면 선 남음) 0(2차원을 다 차지) > null space 는 다를수있음.

(3-1, 2-2)

Null space

기하적으로 3차원이 있을 때 plane이 있다고 생각. - 두 cv가 만들어내는 plane이 원점을 지나고. - column space

그것에 수직인 . 일차원에 해당되는 부분. 이 ns. ws - 평면

수학적 정의

> xA = 0 [0 0 0]

A라는 matrix 가 있는데 x에 뭐를 곱해도 0이 되는 것. 그 모든 x가 선상에 있음.- 이 백타들은 3차원 점 하나가 3차원.

이 x 에는 matrix가 곱해져서 늘 0 (0도 3차원 [0

0

0]

x의 차원은? A는 3 x 2

x에는 1by 3가 와야됨. (3차원) 결과값은 [0 0 0]

<11/26> part 16

실용적인 측면에서 왜 ns 가 필요한가?

A x = b

[1 5 3

2 6 4]

x에는 3x1 (3차원) 2x1 (2차원) < 백터 입력, 백터 출력 >

A 2x3 .3차원이 ws가 되는건 row (row 백터 두개가 스페닝해서 만들어내는거) rowwise
row - ws : 3차원, 3차원에서 표현할 수 있는 두 개의 row 백터.
3차원에서 표현할 수 있는 두 개의 row vetor - LC 되는 상황은 아님. independent
- 이 백터 두 개가 스패닝 해서 만들어내는 공간 vs 은 2차원. / 3차원은 될 수 없음 갖고 있는게 두 개 밖에 없어서 . / 원점을 지나고 2차원.
row-vs가 2차원일 때 ws에 차지하지 않는 공간 - ns . (orthoganal) . 평면과 직각이 되는 선 하나. > 이게 의미하는 바가 뭘까
ns를 정의할 때 null space : Ax = 0 / x가 뭐가되든 [0 이 되는 x를 찾아라
0]
그림에 선상에 있는 모든 점 vector들이 만족시키는 해당되는 x.
ns 가 어디에 쓰일까 > 어떤 입력이 들어오든 output 에 영향을 미치지 않는. 효과가 영이다. / 어떤 걸 새로 습득 배울 때 점점 숙달되는 과정. 그 과정에서 공통적으로 나타나는게 처음에는 stiff >바이올린 소리 내기 위해 쓸데없는 몸짓을 함. 배워나가는 과정에서 skillful 하면할수록 쓸데없는 걸 많이 함. 필요없는. / 어떤 입력이 들어갔을 때 출력에 별로 영향을 미치지 않는 공간들: ns. (살아가는 과정이 ns를 늘려가는 과정일 수 있음. - 배워가는 과정) / 실제로 그 task를 위해서는 영향을 미치지 않지만 , output 에는 영향을 미치지 않음./ 2차원에 있는 무수한 공간중에서 어떤 값이 들어가면 출력이 다르게 나옴.
선 방향 만큼 간다고 생각하면 어떤 점을 선택 한다 하더라도 결과값에 영향을 미치지 않는게 ns.
ns 가 아닌 것 이 들어가면 출력을 바뀌게 함./ ns 는 어떤 값이 들어가면 / 이 방향으로 변형을 시키면 다르게 값이 나올텐데 plane과 수직인 걸로 변형을 시키면 그 값은 출력값에 영향을 미치지 않음. 어떤 방향으로의 변화는 출력에 미치는게 있고 안 미치는게 있음. (ns와 그렇지 않은 space의 차이)
A 부분은 인공지능. 기계- 데이터 통해 학습가능/ matrix 만 학습되있다면 ns와 그렇지 않은 공간으로 쪼갤 수 잇음. / 이 기계의 성격을 이 방향으로 움직이면 영향을 안미쳐 - ns. 하나는 출력이 바뀌는& 영향을 안미치는 공간. / 입력이 조금만 바뀌어도 출력에 영향 & 출력
ns 의 수학적, 기하적, 실용적 응용적 해석.
인공지능
-그림이나 이미지가 입력으로 들어가면 뭐다 하고 인식을 함 . x에 강아지 사진을 대면 b에 강아지가 나옴. 강아지 사진은 무한대의 종류 입력값이 바뀌어도 강아지가 나옴. (입력이 변해도 출력이 변하지 않는 공간을 따라 움직임. ns의 공간을 따른다.) ns와 평행하게 가는게 강아지 여러 종류의 변화. 평행하지 않게 가면 사진 이미지가 변해서 강아지라고 할 수 없는 다른 category 가 나오는 상황.
-인공지능과 선형대수의 관계를 ns의 차원에서 말한 것.
x라는 입력에 변화를 줬을 때 b에 영향을 미치지 않는. ns공간을 평행하게 따라가면 b에 영향이 없음.

Eigenvectors and Eigenvalues

given A 가 있을 때 Ax = b /x가 v (입력값)
Av = [ ]
basis grid를 만들고 주어진 matrix(A) 의 cv가 어떻게 되느냐 . - 어떻게 생겼냐에 따라 입

력이 어떻게 바뀌는지 예측 할 수 있다.
a1(2,1) a2(1,2) 이 cv
입력 (v) 이 들어가면 Av이렇게 바뀜.
입력이 들어가면 어떻게 바뀌는지 예츨이 됨.
안바뀌게 하고 싶다 하면 basis(1,0 / 0,1)-아무런 영향도 안미치는. -> 입력이 어떻게 되도 그대로 있음.
matrix를 바꿨을 때 입력 바뀌면 바뀜
원래 grid 가 정사각형이 었는데 a1, a2 로확장. grid 자체를 찌부러트리는.
확대를 통해 변함.
v를 정사각형 모서리에 갔다놓으면 Av는 평행사변형의 꼭대기로 감(원래는 정사각형의 꼭지점(1,0 / 0,1) 에 있었는데) . - 이런 느낌의 이동.
v(입력) Av (출력)
->transformation matrix 와 기하적인 visualization
Eigen analysis
eigenvector는 막 움직였다가 모든 가능한 입력(v)중 원점과 Av가 평행하는 순간 생김.
원점과 입력 출력이 평행하는 순간에 (보라색) - 원점과 일직선상 / 행렬을 바꾸면 라인이 또 바뀌어 /
vector는 행렬의 한 형태, 숫자열. + 백터는 방향이다. (ns의 방향으로 가면됨. 방향이 중요하지 반드시 그 선을 따라갈 필요 x) / 보라색 선상이 Eigen vactor. 평행한 선이 E . 그 주어진 행렬에 ei가 뭔가가 질문(given 행렬이 있을 때 E는 뭔가) >>Eigenvactor는 보라색 방향 전체. 어떤 값만 딱 존재한다 아님. 어떤 값 하나만 얘기해도 정답이긴함- 다 줄그으면 되니깐. 여러 가능한 Ei vetor의 집합 - ei vector space.
2 by 2행렬에서 Ei vecotr는 2개가 있다 , 3 by 3는 3개.
2개의 Ei .각각의 Evec에 대해 Ei value가 있음.
Ei value - v가 Av 까지 감. 얼마큼 확대를 시키는 느낌. 그 비율이 항상 똑같음 . / 어떤 값이 있을대 2,35 배로 확장시키는 것. 반대방향) 길이가 1이었다면 확장되는게 1.49 만큼이다.
각 ei 가 있다면 ei value 도 따로.
ns 왜 필요? 살면서 ns는 늘 쓴다. 사람은 진화를 해오면서 ns를 더 확고를 해왔음.
출력의 부분이 우리가 할려는 일 -task(b) 장애물이 없으면 x는 하나만 있으면 딤. ns가 필요 없음 b하나만 넣으면 x가 나와. 하지만, 늘 장애물이 존재함-피해서 돌아가야함 task를 하는데는 지장이 없는> ns를 확장시켜야 하는 이유. b라는 task를 하는 방해를 피해갈 수 있는데 하는 공간을 ns/ 기계에서 이미지를 인식을 하는데 인식 결과가 다른 입력에도 하나가 나올 수. '입력이 많이 변해도' 하는 부분이 ns를 많이 이용하고 있는것.

Ei - a1 , a2 의 cv를 또 다른 두 개의 백터로 한 것. 2 by 2(2개의 cv) 를 또 다른 두 개의 eigenvactor로 갖고 있게 됨. 2 by 2를 다시 2 by 2로 만든 것.
왜 하냐 ?
- Ei : 고유 백터 . 훨씬 고유한 것. unique하다 / 어떤걸 분석할 때 이 사람의능력은 이것과 이것이라고 쪼개고 싶을 때 가 만음. 이것과 이것은 서로 영향을 안미치는 고유의 능력으로 쪼개고 싶은 것. > 그런걸 ei vactor가 해준다(고유. 주어진 행렬은 섞여 있는 것, Eigne

analysis를 하면 훨씬 unique 하고 고유한걸로 바꾸어준다)

상관관계 correlation

모든 사람들의 영어 , 수학 , 국어 과학 점수를 받아. (한 점 찍으면 한사람...) 점은 백터 몇콤마 몇. / 4차원으로도 할 수 있음. (4차원의 백터) 한점은 4개의 값.

영어 , 수학 / 영어, 국어 - 상관관계 같이 가는 느낌.

상관관계를 수치로 표현하면, -1<_r<_ 1 . 0일 때 상관관계 젤 낮음. / 음의 상관관계의 예 - 커피(x)와 수면량(y)

정확히 1,-1이 나오는 순간은 한 선에 있으면 완전한 선상에 있으면.

얼마나 선상에 가까운지가 r값의 절대값

동그랗게 되면 원 어디로 가는지 알 수없음 r=0

여기서 선형대수가 어떻게 쓰일까 ?

국어 영어의 값이 85개. 국어, 영어 백타 가 각각 85차원. 85차원에서 한차원은 한사람을 나타냄. 한점을 찍을 수 잇음(숫자가 85개 들어있음) . / 아무리 차원이 높아도 점 세 개. (원점, 국어 백터, 영어 백터) 삼각형. 이 때 이루는 각도값. 수학이 생겼을 때 이루는 각도값. 각도값에 cos를 붙이면 r(correlation) 이 됨. cos theta = r , cos 90 = 0 (t서로서로 관계없다 = 서로서로 orthoganal하다 . 만나는 지점이 없다.)

국어 영어 correlation 이 안보였다 하면, 각도가 더 멀어져있음. 젤 멀어지면 90도 까지.

r 값이 0.

cos 0 = 1 (국어 영어가 붙어있는. 두방향이 완전히 같아지면 일직선상에 correlate)

각도값을 어떻게 구하지 ?

inner(dot) product : 두 백터 a, b 가 있을대 차원 상관없이 그것의 inner product는 - 각각 안쪽으로 곱해서 더함. / 하나를 수직으로 내리고 두 길이를 곱하면 됨. - 기하학적 계산법 / a 길이는 원점에서 한 점까지의 길이- 차원 상관없음.

- I P 왜 필요 ? 어떤 signal 이 있을때 어떤 fre가 많은지 알려주는게 - spectorgran - 이걸 만들어주기 위한거 .(어떤 성분이 얼만큼 있는지) spec- 소리 백타. 에 다가 사인 웨이브를 여러개 만들어서 inner prodcut를 하면 어떤 값이 나오는데 서로 유사한 성분이 많으면 , correlation 이 만으면 ip 가 높이 나옴 , 적으면 낮게 나옴.

<11/28>

A

[1 2

5 6]

A라는 행렬 2 by 2 가 있을 때 두 개의 열 백터를 만들고 그것에 곱해지는 값 x가 어디로 이동 Ax(결과값) 이동되는게 원점으로부터 직선상에 있게 되는 x를 모두 구하라 - Eigne vector / 어떤 부분에서는 원점과 일직선이 되는게 생김.

eigenvactor : Av =

given A, matirx 에 대해 어떤 백터를 곱했을 때 Av부분은 transformed 된 결과. 가 tranform 되지 않았던 그것의 상수배 밖에 안된다. > 만약 eigenvector라인이 원래 v - Av

가 있었다면 원점 v , Av 가 일직선상. / 어떤 백터가 있고 곱하기 7하면 그것의 확장.
원래 입력에 상수배 한ㄴ것은 transform 하고 나서이다. = 한라인에 있다
transformed 된게 trans되지 않은 입력 백터에 상수배밖에 안됨.
귄점, v, Av는 일직선.
v 가 얼만큼 expand 되느냐 ratio가 eigen value
이것을 만족하는 v가 Evec . 만족하는 라운다가 eigen value.
inner product
몇차원이 되든 원점을 중심으로 어떤 두 백터가 있으면- 언제나 삼각형. 이차원의 평면을 이룸.
a, b 1 by 3 matrix 는 곱하기 가 안돼 b를 3 x 1 로 만들기 - transpose
값이 하나 나오도록 하는 방법 (inner product) 차원을 줄여서 하나만 만드는. !! a.b . 값은 무조건 1 by 1 . scalar. / 값이 3 x 3의 matrix 나오는 방법 (outer product) 더 크기 만드는.
IP 의 기하학적인거 !!
project : 빛을 비춘다고 생각하면 . b 에 다가 a가 project 된 길이 곱하기 원래 b의 길이 >> IP. (31)
각도를 알고 싶다면 ?
a, b 값이 있을 때 다 구할 수 있음.
a의 길이는 3차원 상에서 원점으로 부터의 거리는 루트 ~.
cos 세타는 correlation r 과 아주 유사하다.
세타가 90도 면 cos t = 0 (correlation 이 없다)
IP) projection을 시키고 나서의 길이가 없음 0. 0 곱하기 길이해봤자 0. 무조건 IP는 0 될 수밖에.
**a, b 사람의점수.
4차원상의 a, b백터. cos theta 값을 cos similarity 라고 함. - a 와 b 가 얼마나 비슷한가 수치적으로 이야기 해줄수 있는 방법 > 시험
두 개의 백터 주고 CSi구해라.
**wave
어떤 성분이 얼만큼 들어있나 뽑아내는 방법이 IP를 쓰는 것.
wave라는게 다 어떤 수치, 점으로 되어있음. 숫자값 value가 100개가 있다.
-똑같은 숫자값으로 밑에 웨이브 만들 수 잇음. (백터의 사이즈가 똑같도록) 100차원의 숫자. (??숫자가 100갠데 왜 100차원)
동그라미 친거 두 개를 IP 할수 잇음. > 무슨 값이 나옴.
밑에 웨이브랑 위에 거 파란색 동그라미 하면 또 어떤 IP값이 나옴.
*차곡차곡 하나하나씩 1
사인 웨이브를 100~ 만 hz까지 같은 간격으로 해서 IP. ->값들이 나옴.
100 , 200 hz 사인웨이브 만드는게 phasor. 얼만큼 크게 만드냐면 젤 위에거랑 백터값이 똑같도록.
값이 100, 10 , 1000 나왔다 하면 100hz성분은 어느정도 있구나 , 200hz성분은 거의 없네, .. 2000 hz성분이 크게 있네.
spectrogram 어디가 진하고 약하고 - 10부분이 약하고 1000부분이 진한. (흐린부분 진한부

분)
-시간마다 계속감. 한번 하면 한 time slice에서 하는거. 한번더 하고 또 하고 해서 spectrogram을 만듬. / spectrogram에선 100hz 가 젤 밑에 ..

**똑같은거 a , b IP / C는 2배 빠름.
a,b 곱하는거랑 , a c 곱하는거랑 어떤 차이?
a x b : a가 올라갈 때 b 도 올라가고 a떨어질 때 b 도. 같이 감. 완전 correlate. a.b 의 값은 큼.
a c : a가 올라가는데 어떨땐 c에서 떨어짐. 불일치 때문에 a.c의 값은 a.b보다 작음.
>어떤 복잡한 웨이브가 있다 치고 어떤 것을 IP를 하면 그거랑 똑같은게 있으면 그것에 대해 반응을 크게 하고 큰 IP값을 뱉고. 똑같은 성분이 없으면 아님. /완전 같으면 값이 높고 좀 다르면 값이 낮게 나옴.
**두바퀴 도는 사인 웨이브.
**
sin, cos 웨이브. 90 도만큼 왼쪽 간게 밑에거 . 차이가 90도. 이걸 IP하면 / fre는 완전 같은데 90이동하니 완전 0이되버림. / 조금 옆으로 햇다해서 빨간색 아님 하면 안되는데... fre 같음을 이용한 아이디어는 좋았지만 phase shift 해서 맛이 가는 상황.
IP가 0이 되려면 90도가 되야. 9차원 상에서 a, b 백터는 90도가 되어야 두 개의 IP가 0이됨. 웨이브 상(time , radian축) 에서 90도 와 9차원상 백터 공간에서 90도가 어떻게 같아졌을까?
-공간이 다른데 .. / phase difference 가 0 이라면 백터공간에서도 딱 맞아서 a b 는 합쳐짐. (둘의 각도가 필연적으로 같음)
IP를 써서 한 웨이브 내에 어떤 주파수 성분이 많은지 보는게 목적, phase가 왔다갔다 너무 민감하게 작용하는건 별로 안좋음. phase 에 민감하게 작용하지 않도록 / b를 만들어내는 phasor를 sin cos phasor말고 complex phasor을 씀. / sn cos 은 phase shift에 대한 민감도가 안좋아서. complex phasor를 써서 IP. / 원래 wave 는 real 공간에서 존재하는데 IP할거를 complex 로 함. complex value 도 어차피 한값이니까 9개를 만들어서 IP. 하면 phase에 대한 민감도 부분을 해결할 수 있음.

<12/3> part 18
a.b = c (scalar) - 서로서로 얼마나 비슷한가 봄
1xn nx1 1x1
a , b 길이가 똑같다면(고정되있다면) dot product의 크기를 결정하는건 angle/theta . - 0에 가까울수록 커짐. 90에 가까워질수록 최소 .
웨이브에 이용. -시간으로 볼 때 백터. 웨이브가 어떤 백타가 됨. (30개의 숫자)
어떤 웨이브속에 어떤 사인 성분이 많은가 아는게 중요. / spectro analysis - 어떤 fre 성분이 많은가. 어떤 한 시점에서의 analysis - spectrum.
아무리 복잡한 signal도 사인웨이브 성분들의 합이다 - Furior ?
slow한것부터 빨라지는 성분 각각이 얼마나 많은지가 중요.(어떤건 10 들어있고 어떤건 0.0001..)
어느정도 있는지 알아서 플래팅.

이걸 할 때 IP의 기법을 씀 /웨이브 자체가 백터/ 사이즈 똑같이 만들어서 IP>해당되는 값이 나옴. 값이 5나오면 5만큼 이 성분이 여기 들어있구나 / 0.1 나오면 다른거에 비해 적게 들어있네. / 여러 종류의 사인 웨이브를 fre별로 만들어서 원래 복잡한 complex wave. signal에 IP하면된다. > 플래팅을 하면 spectrogram 이된다.
fre관점에선 똑같은데 조금 90도정도 이동하면 0 이 됨.
Target wave 에 대해 여러 웨이브를 만들어서 probe. Target 이 조금 shift함에 따라 IP 되는 결과값이 변함, phasor 민감도가 너무 큼.
>민감하지 않은 complex phasor을 씀. -허수를 포함하는 complex number .백터이긴 하지만. 곱하기는 할 수 있지만 IP해서 나오는 값이 한 값이 나오긴 하지만 complex 값. /IP 할 땐 두 개의 demension이 같아야함 / 개수도 같아야함/ - 사인웨이브에선 real nmber 실수값이 나와 플래팅 가능, complex number - 플래팅이 불가능. >어떻게 플래팅 하냐 - 절대값을 씌우면 실수값이 나옴.
웨이브(실수)에 complex fre IP하면 값이 complex number.
complex 숫자의 절대값 하는 방법
샘플이 30 개면 30개로 만들어야.

```
nSamp = len(s) # 백터의 사이즈.  # target wave의 샘플의 개수가 얼만지
```
이걸 한 이유는 complex wave을 여러개 만들어서 여기다 각각 곱해서 성분이 얼마나 강한지 IP. 곱하기 위해서 길이가 얼마나 되는지 알아야 phasor을 그 길이와 똑같이 만듬.
>그래서 샘플 개수가 중요
밑엔 안함
```
dur = nSamp / sr
t = np.linspace(1/sr, dur, nSamp
```

```
nFFT = nSamp # 샘플 개수를 nFFT라고 이름을 바꿈 둘은 같은거.
# for loop .
amp = [];
for n in range(0,nFFT): #샘플 개수 만큼 for loop을 돔.
    omega = 2*np.pi*n/nFFT # angular velocity
    z = np.exp(omega*1j)**(np.arange(0,nSamp)) # complex wave를 만드는 줄.
```
complex phasor을 이용하는 줄. 노트)lj 는 I. **는 지수. 뒷부분은 앞의 지수. range로. 샘플이 백개라 하면 백개. 지수니까 앞에 다 곱하기 됨.
샘플개수 100개라 가정.
오메가는 2파이 곱하기 n은 0부터 1,2,,...샘플의 개수까지 바뀜. 나누기 샘플.
첫 번째 루프에서는 오메가는 0. 두 번째 루프에서 1이들어와
1부터 왔을 때 다 곱하면 0부터 2파이 까지 되는 세타가 됨. e의 세타 I- 한바퀴도는게 됨.
샘플의 개수는 총 100개를 해서 한바퀴돈 것. > 여기까지가 두 번째 루프.
세 번째 루프) 2 들어오면 4 파이로 바뀜. 2바퀴 도는 것. > 100개의 샘플로 2바퀴 돔. 50개씩 두 번. / 사인웨이브로 생각한다면 오른쪽.
이런식으로 nsamp바퀴까지 감. 100바퀴까지.
이걸 타겟에다가 다 곱해. 백타니까 오리지날 시그널에 덮어씌움. 오리지널 시그널 s가 있고

z 여러개를 만듬.- complex 웽브를 루프의 개수만큼 만듬.
Z0~ Zsamp-1까지 다 IP를하면 값이 각각 나옴. for loop 할때마다 z는 다른값.
amp.append(np.abs(np.dot(s,z))) > dot product . abs 가 절대값. absolute. dot product 에서 complex 값이 나오더라도 abs취하면 크기값을 알 수 있음.
(-3 + 0i라면 원점으로부터의 길이는 3.)
amp. append 할때마다 하나씩 append를 함 amp라는 variable에다가 첫 번째 루프에다가 만들고 계산하고 넣고... 루프 다 끝나면 루프의 개수만큼 amp는 크기를 가짐.
for loop 끝나고 나서 amp의 길이는?
amp에 허수가 들어있다. - 틀린말. abs를 했기 때문에 .
amp를 플랫팅해야함 이제.
y에 해당되는 값들을 알아냄. x에 해당되는 좌표는 뭔가?
fig = plt.figure()
ax = fig.add_subplot(111)
freq = np.arange(1,nFFT+1)*sr/nFFT;
1부터 100((1,nFFT+1) 까지 백터를 만들고 sr (만) 을 곱해주고 또 샘플의 개수로 나눠줌. > 100부터 만까지. > 이걸 x축으로 함.
ax.plot(freq, amp)
ax.set_xlabel('frequency (Hz)')
ax.set_ylabel('amplitude') # 각 fre성분에 대한 에너지 값.

x축이 0부터 만까지. 한 점. 한 바. 들이 하나의 IP한 값.
총 바의 개수는 샘플의 개수와 같음.
이 그래프 의미 ? 왼쪽 오른쪽이 대칭.
*sr의 half만 의미가 있다 - 허용가능한 fre의 범위.
그래프도 중간까지 의미가 있음. 이 half를 부를 때 spectrum이라 함. - 각 fre성분들이 probing을 하면서 가져오는 에너지값. IP의 abs값.
2000hz가 있다 하면 이천 hz의 성분이 얼만큼 있냐 - 요만큼. / 500hz 성분은 엄청 큼.

첨에 gradually decreasing하게 만들고 500에 산을 만들었음. 1500에도 산, 2500에 산. > 그대로 그래프에 나타남. / 만약 위에서 2500에 산을 안만들었으면 없을 것.
gra decre 하게 안했으면 평평하게 나옴.
어떤 고정 time에서 어떤 성분이 많은가.
바뀌는 시간별로 변하는게 아님. - spectrogram.

<12 / 5> 교수님 gitub 프린트
90도가 된다면 cos(t) = 0
a b 값이 signal일때도 마찬가지로 적용. shift 할 때 dot product해보면 0이 나옴. - 백타들의 세타값과 같다.
cos (theta) = r
a - 영어 점수 b -국어 점수 . 10명이있으면 10차원의 백타.
시험 ! 데이터를 그래프에 찍었더니 일직선 선상 > 영어, 국어 백터가 두 개의 각도가 0. r

=1. / r = -1 이면 영어 국어가 반대쪽으로. - 각도가 180도. cos(파이) = -1
10차원의 한 점은 10개의 숫자를 갖고 있음. 영어 백터, 국어 백터가 있다. 한축이 sub.
1) 한점이 sub-일직선에 가까울수록 r 이 1에 가깝다.
영어, 국어 백터의 각도값 의 cos을 취하니 r값과 같아짐.

180도 shift되면 어떤게 올라갈대 내려가고 정반대의 상황. - 서로의 차이는 180도의 phase 차이가 있다. 180도는 cos similarity 가 -1. 2번 그래프의 180도와도 같음.
cos 값이 엄청 커도 r 값은 -1. 크기하곤 상관없음. 2) 길이가 증폭과 관련.

spectrogram
spectrum 그림 - x축은 sr끝까지를 range. 양쪽이 대칭. 반은 필요없는. 5000까지가 표현할 수 있는 주파수. peak들이 formant다. 한순간.
spectrogram - 한 슬라이스 한 time poing에서 자른게 spectrum. / y축 밑쪽이 저주파.
fourier 분석 하면 한 장짜리 스펙트럼 나올텐데 어떻게 spectrogram 만들지?

```
max_freq = None  # cutoff freq
win_size = 0.008 # sec  # 길게 웨이브 있다 생각하면 앞부분 조금만 잘라서 spectrum 만들고 조금 이동해서 또 만들고.... 한 장한장씩 만드는거. 얼만큼의 벡터를 가지고 dot procut 해야겠다 정해야됨. 얼만큼의 사이즈. 1000분의 8초 하고 .
win_step = 0.001  # sec # 천분의 일초 하고 이동... 그게 합쳐지면 spectrogram
# 한번만 하면 spectrum
win_type = 'hanning' # options: 'rect', 'hamming', 'hanning', 'kaiser', 'blackman'
nfft = 1024

# Emphasize signal
s = preemphasis(s)
# Frame signal
frames = frame_signal(s, sr, win_size, win_step)
# Apply window function
frames *= get_window(win_size, sr, win_type)
print('frames:', frames.shape)
```

spectrogram 에 첫 번째 해당되는 부분은 진하고 두 번째 세 번째는 훨 연함.
그냥하면 너무 연하니 한번더 처리.
complex number 로 dot product나오는데 abs취하면 실수값 그리고 플래팅 된 것. 값들이 진한거는 1보다 큰값이 나오고 연한부분은 1보다 작은값이 나옴. - 이걸 제곱을 치면 진한 부분은 더 커지고 100제곱하면 만. 1보다 작은값은 훨씬 작아지는. > power spectrum
magspec**2 #원래값을 제곱함 .양의 값은 엄청 커지고 1보다 작으면 0에 가까워짐.
>log를 취하기 위해서 이런 작업을 함.
베이스가 10이되는 로그를 치면 0에 가까워지는 값들은 0.00001에 로그하면 -4, 5 정도로 뚝 나옴. 10만 있으면 4. 이렇게 나와 /다룰 수 있는 범위내로 바꿔줌 - 로그 값이 골고루 나올

수 있다. 산맥이 급하게 낮아지는 걸 visualize를 고주파에서도 뚜렷하게 보이는 작업.
원래값의 power을 하고 log.