1806554 Ganesh Bhandarkar Assignment 5

Numpy

In [35]:
```
import numpy as np
```

In [36]:
```
a = np.array([1,2,3,4,5])
print(a)
```

```
[1 2 3 4 5]
```

In [37]:
```
a = np.zeros(5)
print(a)
```

```
[0. 0. 0. 0. 0.]
```

In [38]:
```
a = np.ones(5)
print(a)
```

```
[1. 1. 1. 1. 1.]
```

In [39]:
```
print(dir(np),end="")
```

```
['ALLOW_THREADS', 'AxisError', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSource', 'ERR_C
ALL', 'ERR_DEFAULT', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE', 'ERR_WARN', 'FLO
ATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_OVERFLOW', 'FPE_UNDERFLO
W', 'False_', 'Inf', 'Infinity', 'MAXDIMS', 'MAY_SHARE_BOUNDS', 'MAY_SHARE_EXACT', 'Mach
Ar', 'ModuleDeprecationWarning', 'NAN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'RAIS
E', 'RankWarning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDER
FLOW', 'ScalarType', 'Tester', 'TooHardError', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_
PYVALS_NAME', 'VisibleDeprecationWarning', 'WRAP', '_NoValue', '_UFUNC_API', '__NUMPY_SE
TUP__', '__all__', '__builtins__', '__cached__', '__config__', '__dir__', '__doc__', '__
file__', '__getattr__', '__git_revision__', '__loader__', '__mkl_version__', '__name__',
'__package__', '__path__', '__spec__', '__version__', '_add_newdoc_ufunc', '_distributor
_init', '_globals', '_mat', '_pytesttester', 'abs', 'absolute', 'add', 'add_docstring',
'add_newdoc', 'add_newdoc_ufunc', 'alen', 'all', 'allclose', 'alltrue', 'amax', 'amin',
'angle', 'any', 'append', 'apply_along_axis', 'apply_over_axes', 'arange', 'arccos', 'ar
ccosh', 'arcsin', 'arcsinh', 'arctan', 'arctan2', 'arctanh', 'argmax', 'argmin', 'argpar
tition', 'argsort', 'argwhere', 'around', 'array', 'array2string', 'array_equal', 'array
_equiv', 'array_repr', 'array_split', 'array_str', 'asanyarray', 'asarray', 'asarray_chk
finite', 'ascontiguousarray', 'asfarray', 'asfortranarray', 'asmatrix', 'asscalar', 'atl
least_1d', 'atleast_2d', 'atleast_3d', 'average', 'bartlett', 'base_repr', 'binary_repr',
'bincount', 'bitwise_and', 'bitwise_not', 'bitwise_or', 'bitwise_xor', 'blackman', 'bloc
k', 'bmat', 'bool', 'bool8', 'bool_', 'broadcast', 'broadcast_arrays', 'broadcast_to',
'busday_count', 'busday_offset', 'busdaycalendar', 'byte', 'byte_bounds', 'bytes0', 'byt
es_', 'c_', 'can_cast', 'cast', 'cbrt', 'cdouble', 'ceil', 'cfloat', 'char', 'characte
r', 'chararray', 'choose', 'clip', 'clongdouble', 'clongfloat', 'column_stack', 'common_
type', 'compare_chararrays', 'compat', 'complex', 'complex128', 'complex64', 'complex_',
'complexfloating', 'compress', 'concatenate', 'conj', 'conjugate', 'convolve', 'copy',
'copysign', 'copyto', 'core', 'corrcoef', 'correlate', 'cos', 'cosh', 'count_nonzero',
```

```
'cov', 'cross', 'csingle', 'ctypeslib', 'cumprod', 'cumproduct', 'cumsum', 'datetime64',
'datetime_as_string', 'datetime_data', 'deg2rad', 'degrees', 'delete', 'deprecate', 'dep
recate_with_doc', 'diag', 'diag_indices', 'diag_indices_from', 'diagflat', 'diagonal',
'diff', 'digitize', 'disp', 'divide', 'divmod', 'dot', 'double', 'dsplit', 'dstack', 'dt
ype', 'e', 'ediff1d', 'einsum', 'einsum_path', 'emath', 'empty', 'empty_like', 'equal',
'errstate', 'euler_gamma', 'exp', 'exp2', 'expand_dims', 'expm1', 'extract', 'eye', 'fab
s', 'fastCopyAndTranspose', 'fft', 'fill_diagonal', 'find_common_type', 'finfo', 'fix',
'flatiter', 'flatnonzero', 'flexible', 'flip', 'fliplr', 'flipud', 'float', 'float16',
'float32', 'float64', 'float_', 'float_power', 'floating', 'floor', 'floor_divide', 'fma
x', 'fmin', 'fmod', 'format_float_positional', 'format_float_scientific', 'format_parse
r', 'frexp', 'frombuffer', 'fromfile', 'fromfunction', 'fromiter', 'frompyfunc', 'fromre
gex', 'fromstring', 'full', 'full_like', 'fv', 'gcd', 'generic', 'genfromtxt', 'geomspac
e', 'get_array_wrap', 'get_include', 'get_printoptions', 'getbufsize', 'geterr', 'geterr
call', 'geterrobj', 'gradient', 'greater', 'greater_equal', 'half', 'hamming', 'hannin
g', 'heaviside', 'histogram', 'histogram2d', 'histogram_bin_edges', 'histogramdd', 'hspl
it', 'hstack', 'hypot', 'i0', 'identity', 'iinfo', 'imag', 'in1d', 'index_exp', 'indice
s', 'inexact', 'inf', 'info', 'infty', 'inner', 'insert', 'int', 'int0', 'int16', 'int3
2', 'int64', 'int8', 'int_', 'intc', 'integer', 'interp', 'intersect1d', 'intp', 'inver
t', 'ipmt', 'irr', 'is_busday', 'isclose', 'iscomplex', 'iscomplexobj', 'isfinite', 'isf
ortran', 'isin', 'isinf', 'isnan', 'isnat', 'isneginf', 'isposinf', 'isreal', 'isrealob
j', 'isscalar', 'issctype', 'issubclass_', 'issubdtype', 'issubsctype', 'iterable', 'ix
_', 'kaiser', 'kron', 'lcm', 'ldexp', 'left_shift', 'less', 'less_equal', 'lexsort', 'li
b', 'linalg', 'linspace', 'little_endian', 'load', 'loads', 'loadtxt', 'log', 'log10',
'log1p', 'log2', 'logaddexp', 'logaddexp2', 'logical_and', 'logical_not', 'logical_or',
'logical_xor', 'logspace', 'long', 'longcomplex', 'longdouble', 'longfloat', 'longlong',
'lookfor', 'ma', 'mafromtxt', 'mask_indices', 'mat', 'math', 'matmul', 'matrix', 'matrix
lib', 'max', 'maximum', 'maximum_sctype', 'may_share_memory', 'mean', 'median', 'memma
p', 'meshgrid', 'mgrid', 'min', 'min_scalar_type', 'minimum', 'mintypecode', 'mirr', 'mk
l', 'mod', 'modf', 'moveaxis', 'msort', 'multiply', 'nan', 'nan_to_num', 'nanargmax', 'n
anargmin', 'nancumprod', 'nancumsum', 'nanmax', 'nanmean', 'nanmedian', 'nanmin', 'nanpe
rcentile', 'nanprod', 'nanquantile', 'nanstd', 'nansum', 'nanvar', 'nbytes', 'ndarray',
'ndenumerate', 'ndfromtxt', 'ndim', 'ndindex', 'nditer', 'negative', 'nested_iters', 'ne
waxis', 'nextafter', 'nonzero', 'not_equal', 'nper', 'npv', 'numarray', 'number', 'obj2s
ctype', 'object', 'object0', 'object_', 'ogrid', 'oldnumeric', 'ones', 'ones_like', 'o
s', 'outer', 'packbits', 'pad', 'partition', 'percentile', 'pi', 'piecewise', 'place',
'pmt', 'poly', 'poly1d', 'polyadd', 'polyder', 'polydiv', 'polyfit', 'polyint', 'polymu
l', 'polynomial', 'polysub', 'polyval', 'positive', 'power', 'ppmt', 'printoptions', 'pr
od', 'product', 'promote_types', 'ptp', 'put', 'put_along_axis', 'putmask', 'pv', 'quant
ile', 'r_', 'rad2deg', 'radians', 'random', 'rate', 'ravel', 'ravel_multi_index', 'rea
l', 'real_if_close', 'rec', 'recarray', 'recfromcsv', 'recfromtxt', 'reciprocal', 'recor
d', 'remainder', 'repeat', 'require', 'reshape', 'resize', 'result_type', 'right_shift',
'rint', 'roll', 'rollaxis', 'roots', 'rot90', 'round', 'round_', 'row_stack', 's_', 'saf
e_eval', 'save', 'savetxt', 'savez', 'savez_compressed', 'sctype2char', 'sctypeDict', 's
ctypeNA', 'sctypes', 'searchsorted', 'select', 'set_numeric_ops', 'set_printoptions', 's
et_string_function', 'setbufsize', 'setdiff1d', 'seterr', 'seterrcall', 'seterrobj', 'se
txor1d', 'shape', 'shares_memory', 'short', 'show_config', 'sign', 'signbit', 'signedint
eger', 'sin', 'sinc', 'single', 'singlecomplex', 'sinh', 'size', 'sometrue', 'sort', 'so
rt_complex', 'source', 'spacing', 'split', 'sqrt', 'square', 'squeeze', 'stack', 'std',
'str', 'str0', 'str_', 'string_', 'subtract', 'sum', 'swapaxes', 'sys', 'take', 'take_al
ong_axis', 'tan', 'tanh', 'tensordot', 'test', 'testing', 'tile', 'timedelta64', 'trac
e', 'tracemalloc_domain', 'transpose', 'trapz', 'tri', 'tril', 'tril_indices', 'tril_ind
ices_from', 'trim_zeros', 'triu', 'triu_indices', 'triu_indices_from', 'true_divide', 't
runc', 'typeDict', 'typeNA', 'typecodes', 'typename', 'ubyte', 'ufunc', 'uint', 'uint0',
'uint16', 'uint32', 'uint64', 'uint8', 'uintc', 'uintp', 'ulonglong', 'unicode', 'unicod
e_', 'union1d', 'unique', 'unpackbits', 'unravel_index', 'unsignedinteger', 'unwrap', 'u
se_hugepage', 'ushort', 'vander', 'var', 'vdot', 'vectorize', 'version', 'void', 'void
0', 'vsplit', 'vstack', 'warnings', 'where', 'who', 'zeros', 'zeros_like']
```

In [40]:

```
x = np.unique(a,return_counts = True)

x
```

Out[40]:

```
(array([1.]), array([5], dtype=int64))
```

In [41]:
```
s = np.sum(a)
s
```

Out[41]: 5.0

In [42]:
```
p = np.prod(a)
p
```

Out[42]: 1.0

In [44]:
```
d = np.divide([2,5],[1,2])
d
```

Out[44]: array([2. , 2.5])

In [46]:
```
ss = np.array_split(a,2)
ss
```

Out[46]: [array([1., 1., 1.]), array([1., 1.])]

In [47]:
```
sss = np.cumsum(a)
sss
```

Out[47]: array([1., 2., 3., 4., 5.])

In [48]:
```
diff = np.diff(a)
diff
```

Out[48]: array([0., 0., 0., 0.])

Pandas

In [13]:
```
import pandas as pd
```

In [15]:
```
print(dir(pd),end="")
```

```
['BooleanDtype', 'Categorical', 'CategoricalDtype', 'CategoricalIndex', 'DataFrame', 'Da
teOffset', 'DatetimeIndex', 'DatetimeTZDtype', 'ExcelFile', 'ExcelWriter', 'Float64Inde
x', 'Grouper', 'HDFStore', 'Index', 'IndexSlice', 'Int16Dtype', 'Int32Dtype', 'Int64Dtyp
e', 'Int64Index', 'Int8Dtype', 'Interval', 'IntervalDtype', 'IntervalIndex', 'MultiInde
```

```
x', 'NA', 'NaT', 'NamedAgg', 'Period', 'PeriodDtype', 'PeriodIndex', 'RangeIndex', 'Seri
es', 'SparseDtype', 'StringDtype', 'Timedelta', 'TimedeltaIndex', 'Timestamp', 'UInt16Dt
ype', 'UInt32Dtype', 'UInt64Dtype', 'UInt64Index', 'UInt8Dtype', '__builtins__', '__cach
ed__', '__doc__', '__docformat__', '__file__', '__getattr__', '__git_version__', '__load
er__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_config', '_ha
shtable', '_is_numpy_dev', '_lib', '_libs', '_np_version_under1p16', '_np_version_under1
p17', '_np_version_under1p18', '_testing', '_tslib', '_typing', '_version', 'api', 'arra
y', 'arrays', 'bdate_range', 'compat', 'concat', 'core', 'crosstab', 'cut', 'date_rang
e', 'describe_option', 'errors', 'eval', 'factorize', 'get_dummies', 'get_option', 'infe
r_freq', 'interval_range', 'io', 'isna', 'isnull', 'json_normalize', 'lreshape', 'melt',
'merge', 'merge_asof', 'merge_ordered', 'notna', 'notnull', 'offsets', 'option_context',
'options', 'pandas', 'period_range', 'pivot', 'pivot_table', 'plotting', 'qcut', 'read_c
lipboard', 'read_csv', 'read_excel', 'read_feather', 'read_fwf', 'read_gbq', 'read_hdf',
'read_html', 'read_json', 'read_orc', 'read_parquet', 'read_pickle', 'read_sas', 'read_s
pss', 'read_sql', 'read_sql_query', 'read_sql_table', 'read_stata', 'read_table', 'reset
_option', 'set_eng_float_format', 'set_option', 'show_versions', 'test', 'testing', 'tim
edelta_range', 'to_datetime', 'to_numeric', 'to_pickle', 'to_timedelta', 'tseries', 'uni
que', 'util', 'value_counts', 'wide_to_long']
```

In [16]:
```
d = pd.read_csv("info_large.csv")
```

In [18]:
```
d.head()
```

Out[18]:

	Ganesh	Bhandarkar	1806554
0	Moti	Doggy	13
1	Ram	Turtle	11

In [19]:
```
d.tail()
```

Out[19]:

	Ganesh	Bhandarkar	1806554
0	Moti	Doggy	13
1	Ram	Turtle	11

In [20]:
```
len(d.value_counts())
```

Out[20]: 2

In [21]:
```
d.nunique()
```

Out[21]:
```
Ganesh        2
Bhandarkar    2
1806554       2
dtype: int64
```

In [22]:
```
d.describe()
```

Out[22]:

	1806554

	1806554
count	2.000000
mean	12.000000
std	1.414214
min	11.000000
25%	11.500000
50%	12.000000
75%	12.500000
max	13.000000

In [23]:
```
d.isna().any()
```

Out[23]:
```
Ganesh        False
Bhandarkar    False
1806554       False
dtype: bool
```

In [24]:
```
d.isna().sum()
```

Out[24]:
```
Ganesh        0
Bhandarkar    0
1806554       0
dtype: int64
```

In [25]:
```
d.mean()
```

Out[25]:
```
1806554    12.0
dtype: float64
```

Scipy

In [5]:
```
from scipy import linalg,ndimage
import cv2
import matplotlib.pyplot as plt
```

In [32]:
```
mat = np.array([[2,1],[4,3]])
sc.linalg.det(mat)
```

Out[32]:
```
2.0
```

In [33]:
```
linalg.inv(mat)
```

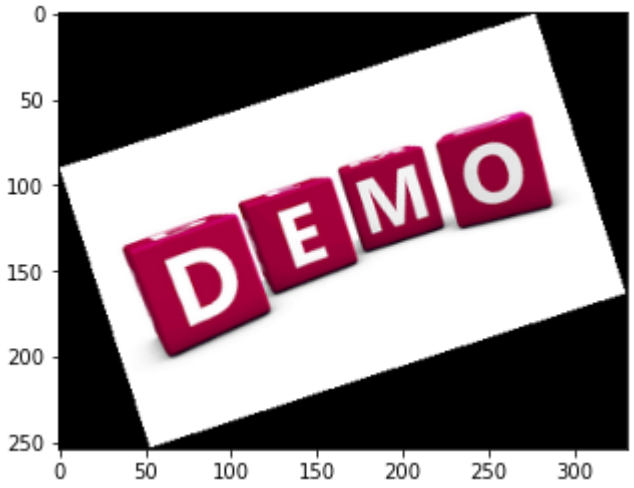
Out[33]:
```
array([[ 1.5, -0.5],
       [-2. ,  1. ]])
```

In [34]:
```
linalg.svd(mat)
```

Out[34]:
```
(array([[-0.40455358, -0.9145143 ],
        [-0.9145143 ,  0.40455358]]),
 array([5.4649857 , 0.36596619]),
 array([[-0.81741556, -0.57604844],
        [-0.57604844,  0.81741556]]))
```

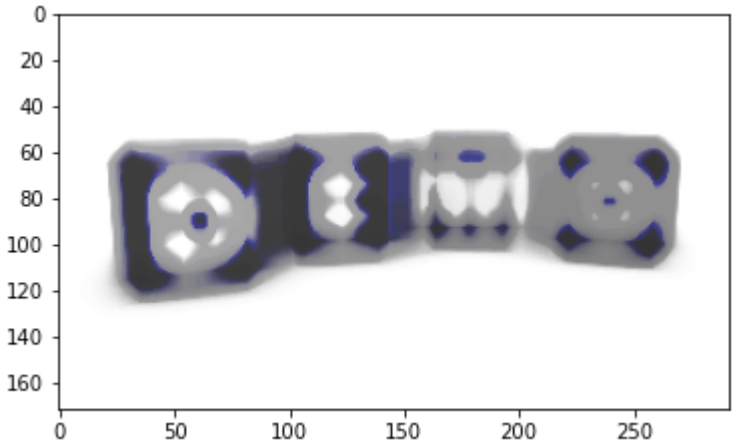
In [12]:
```
image = cv2.imread('demo.png')
a = ndimage.rotate(image,18)
plt.imshow(a)
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x16159d818b0>
```



In [13]:
```
b = ndimage.median_filter(image,20)
plt.imshow(b)
```

Out[13]:
```
<matplotlib.image.AxesImage at 0x16159dd8460>
```



In [21]:

```
c = ndimage.binary_opening(image)
d = ndimage.binary_erosion(image)
e = ndimage.binary_dilation(image)
f = ndimage.binary_closing(image)
print(c,d,e,f,end="")
```

```
[[[False False False]
  [False  True False]
  [False  True False]
  ...
  [False  True False]
  [False  True False]
  [False False False]]

 [[False  True False]
  [ True  True  True]
  [ True  True  True]
  ...
  [ True  True  True]
  [ True  True  True]
  [False  True False]]

 [[False  True False]
  [ True  True  True]
  [ True  True  True]
  ...
  [ True  True  True]
  [ True  True  True]
  [False  True False]]

 ...

 [[False  True False]
  [ True  True  True]
  [ True  True  True]
  ...
  [ True  True  True]
  [ True  True  True]
  [False  True False]]

 [[False  True False]
  [ True  True  True]
  [ True  True  True]
  ...
  [ True  True  True]
  [ True  True  True]
  [False  True False]]

 [[False False False]
  [False  True False]
  [False  True False]
  ...
  [False  True False]
  [False  True False]
  [False False False]]] [[[False False False]
  [False False False]
  [False False False]
  ...
  [False False False]
  [False False False]
```

```
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

...

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

[[False False False]
 [False False False]
 [False False False]
 ...
 [False False False]
 [False False False]
 [False False False]]] [[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]
```

```
...

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]

[[ True  True  True]
 [ True  True  True]
 [ True  True  True]
 ...
 [ True  True  True]
 [ True  True  True]
 [ True  True  True]]] [[[False False False]
 [False False False]
 [False False False]
 ...
 [False False False]
 [False False False]
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

...

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
 [False False False]]

[[False False False]
 [False  True False]
 [False  True False]
 ...
 [False  True False]
 [False  True False]
```

```
  [False False False]]

 [[False False False]
  [False False False]
  [False False False]
  ...
  [False False False]
  [False False False]
  [False False False]]]
```

Time

In [28]:
```
import time,calendar
```

In [25]:
```
localtime = time.localtime(time.time())
print(localtime)
```

```
time.struct_time(tm_year=2021, tm_mon=2, tm_mday=15, tm_hour=10, tm_min=33, tm_sec=5, tm
_wday=0, tm_yday=46, tm_isdst=0)
```

In [32]:
```
cal = calendar.month(2021, 2)
print(cal)
```

```
   February 2021
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

In [34]:
```
localtime = time.asctime( time.localtime(time.time()))
localtime
```

Out[34]:
```
'Mon Feb 15 10:35:55 2021'
```

In [35]:
```
srctime = time.strptime("30 Nov 00", "%d %b %y")
srctime
```

Out[35]:
```
time.struct_time(tm_year=2000, tm_mon=11, tm_mday=30, tm_hour=0, tm_min=0, tm_sec=0, tm_
wday=3, tm_yday=335, tm_isdst=-1)
```

MoviePy

In [83]:
```
!pip install moviepy
#from moviepy.editor import *
```

```
Requirement already satisfied: moviepy in c:\users\kiit\anaconda3\lib\site-packages (1.
0.3)
Requirement already satisfied: imageio-ffmpeg>=0.2.0; python_version >= "3.4" in c:\user
s\kiit\anaconda3\lib\site-packages (from moviepy) (0.4.3)
Requirement already satisfied: proglog<=1.0.0 in c:\users\kiit\anaconda3\lib\site-packag
es (from moviepy) (0.1.9)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in c:\users\kiit\anaconda3\lib\site-pac
kages (from moviepy) (4.50.2)
Requirement already satisfied: numpy; python_version >= "2.7" in c:\users\kiit\anaconda3
\lib\site-packages (from moviepy) (1.19.2)
Requirement already satisfied: decorator<5.0,>=4.0.2 in c:\users\kiit\anaconda3\lib\site
-packages (from moviepy) (4.4.2)
Requirement already satisfied: imageio<3.0,>=2.5; python_version >= "3.4" in c:\users\ki
it\anaconda3\lib\site-packages (from moviepy) (2.9.0)
Requirement already satisfied: requests<3.0,>=2.8.1 in c:\users\kiit\anaconda3\lib\site-
packages (from moviepy) (2.24.0)
Requirement already satisfied: pillow in c:\users\kiit\anaconda3\lib\site-packages (from
imageio<3.0,>=2.5; python_version >= "3.4"->moviepy) (8.0.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\kiit
\anaconda3\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\kiit\anaconda3\lib\site-pac
kages (from requests<3.0,>=2.8.1->moviepy) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kiit\anaconda3\lib\site-pa
ckages (from requests<3.0,>=2.8.1->moviepy) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\users\kiit\anaconda3\lib\site-packages
(from requests<3.0,>=2.8.1->moviepy) (2.10)
```

In [93]:

```
from moviepy.editor import *
# Load myHolidays.mp4 and select the subclip 00:00:50 -
00:00:60
clip = VideoFileClip("myHoliday.mp4").subclip(50,60)

# Reduce the audio volume (volume x 0.8)
clip = clip.volumex(0.8)

# Generate a text clip. You can customize the font, color, etc.
txt_clip = TextClip("My Holidays
2020",fontsize=70,color='white')

# Say that you want it to appear 10s at the center of the
screen
txt_clip = txt_clip.set_pos('center').set_duration(15)

# Overlay the text clip on the first video clip
video = CompositeVideoClip([clip, txt_clip])
```

```
# Write the result to a file (many options available !)
video.write_videofile("myHoliday.mp4")
```

Requests

In [1]:
```
import requests
```

In [2]:
```
response = requests.get('https://api.github.com')
```

In [3]:
```
response
```

Out[3]:
```
<Response [200]>
```

In [4]:
```
if response.status_code == 200:
    print('Success!')
elif response.status_code == 404:
    print('Not Found.')
```

```
Success!
```

In [5]:
```
response.json()
```

Out[5]:

```
{'current_user_url': 'https://api.github.com/user',
 'current_user_authorizations_html_url': 'https://github.com/settings/connections/applic
ations{/client_id}',
 'authorizations_url': 'https://api.github.com/authorizations',
 'code_search_url': 'https://api.github.com/search/code?q={query}{&page,per_page,sort,or
der}',
 'commit_search_url': 'https://api.github.com/search/commits?q={query}{&page,per_page,so
rt,order}',
 'emails_url': 'https://api.github.com/user/emails',
 'emojis_url': 'https://api.github.com/emojis',
 'events_url': 'https://api.github.com/events',
 'feeds_url': 'https://api.github.com/feeds',
 'followers_url': 'https://api.github.com/user/followers',
 'following_url': 'https://api.github.com/user/following{/target}',
 'gists_url': 'https://api.github.com/gists{/gist_id}',
 'hub_url': 'https://api.github.com/hub',
 'issue_search_url': 'https://api.github.com/search/issues?q={query}{&page,per_page,sor
t,order}',
 'issues_url': 'https://api.github.com/issues',
 'keys_url': 'https://api.github.com/user/keys',
 'label_search_url': 'https://api.github.com/search/labels?q={query}&repository_id={repo
sitory_id}{&page,per_page}',
 'notifications_url': 'https://api.github.com/notifications',
 'organization_url': 'https://api.github.com/orgs/{org}',
 'organization_repositories_url': 'https://api.github.com/orgs/{org}/repos{?type,page,pe
r_page,sort}',
 'organization_teams_url': 'https://api.github.com/orgs/{org}/teams',
```

```
 'public_gists_url': 'https://api.github.com/gists/public',
 'rate_limit_url': 'https://api.github.com/rate_limit',
 'repository_url': 'https://api.github.com/repos/{owner}/{repo}',
 'repository_search_url': 'https://api.github.com/search/repositories?q={query}{&page,per_page,sort,order}',
 'current_user_repositories_url': 'https://api.github.com/user/repos{?type,page,per_page,sort}',
 'starred_url': 'https://api.github.com/user/starred{/owner}{/repo}',
 'starred_gists_url': 'https://api.github.com/gists/starred',
 'user_url': 'https://api.github.com/users/{user}',
 'user_organizations_url': 'https://api.github.com/user/orgs',
 'user_repositories_url': 'https://api.github.com/users/{user}/repos{?type,page,per_page,sort}',
 'user_search_url': 'https://api.github.com/search/users?q={query}{&page,per_page,sort,order}'}
```

Cpython

In [72]:
```
!pip install Cython
```

```
Requirement already satisfied: Cython in c:\users\kiit\anaconda3\lib\site-packages (0.29.21)
```

In [96]:
```
%load_ext cython
```

In [97]:
```
%%cython


cdef int a = 0
for i in range(10):
    a += i
print(a)
```

```
45
```

Bokeh

In [9]:
```
import bokeh
import matplotlib.pyplot as plt
```

In [15]:
```
import numpy as np


from bokeh.layouts import gridplot
from bokeh.plotting import figure, output_file, show
```

```
# prepare some data
N = 100
x = np.linspace(0, 4*np.pi, N)
y0 = np.sin(x)
y1 = np.cos(x)
y2 = np.sin(x) + np.cos(x)

# output to static HTML file
output_file("linked_panning.html")

# create a new plot
s1 = figure(width=250, plot_height=250, title=None)
s1.circle(x, y0, size=10, color="navy", alpha=0.5)

# NEW: create a new plot and share both ranges
s2 = figure(width=250, height=250, x_range=s1.x_range,
y_range=s1.y_range, title=None)
s2.triangle(x, y1, size=10, color="firebrick", alpha=0.5)

# NEW: create a new plot and share only one range
s3 = figure(width=250, height=250, x_range=s1.x_range,
title=None)
s3.square(x, y2, size=10, color="olive", alpha=0.5)

# NEW: put the subplots in a gridplot
p = gridplot([[s1, s2, s3]], toolbar_location=None)

# show the results
show(p)
```

In [16]:

```
import numpy as np

from bokeh.plotting import figure, output_file, show
```

```
# prepare some data
N = 4000
x = np.random.random(size=N) * 100
y = np.random.random(size=N) * 100
radii = np.random.random(size=N) * 1.5
colors = [
    "#%02x%02x%02x" % (int(r), int(g), 150) for r, g in
zip(50+2*x, 30+2*y)
]

# output to static HTML file (with CDN resources)
output_file("color_scatter.html", title="color_scatter.py
example", mode="cdn")

TOOLS =
"crosshair,pan,wheel_zoom,box_zoom,reset,box_select,lasso_select"



# create a new plot with the tools above, and explicit ranges
p = figure(tools=TOOLS, x_range=(0, 100), y_range=(0, 100))

# add a circle renderer with vectorized colors and sizes
p.circle(x, y, radius=radii, fill_color=colors, fill_alpha=0.6,
line_color=None)

# show the results
show(p)
```

TextBlob

In [18]:

```
! pip install textblob
```

```
Collecting textblob
  Downloading textblob-0.15.3-py2.py3-none-any.whl (636 kB)
```

```
Requirement already satisfied: nltk>=3.1 in c:\users\kiit\anaconda3\lib\site-packages (f
rom textblob) (3.5)
Requirement already satisfied: tqdm in c:\users\kiit\anaconda3\lib\site-packages (from n
ltk>=3.1->textblob) (4.50.2)
Requirement already satisfied: click in c:\users\kiit\anaconda3\lib\site-packages (from
nltk>=3.1->textblob) (7.1.2)
Requirement already satisfied: regex in c:\users\kiit\anaconda3\lib\site-packages (from
nltk>=3.1->textblob) (2020.10.15)
Requirement already satisfied: joblib in c:\users\kiit\anaconda3\lib\site-packages (from
nltk>=3.1->textblob) (0.17.0)
Installing collected packages: textblob
Successfully installed textblob-0.15.3
```

In [24]:
```
from textblob import TextBlob


wiki = TextBlob("Python is a high-level, general-purpose
programming language.")
```

In [26]:
```
wiki.tags
```

Out[26]:
```
[('Python', 'NNP'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('high-level', 'JJ'),
 ('general-purpose', 'JJ'),
 ('programming', 'NN'),
 ('language', 'NN')]
```

In [27]:
```
wiki.sentiment
```

Out[27]:
```
Sentiment(polarity=0.0, subjectivity=0.0)
```

In [28]:
```
wiki.words
```

Out[28]:
```
WordList(['Python', 'is', 'a', 'high-level', 'general-purpose', 'programming', 'languag
e'])
```

In [29]:
```
wiki.sentences
```

Out[29]:
```
[Sentence("Python is a high-level, general-purpose programming language.")]
```

PyAudioAnalysis

In [57]:
```
!pip install plotly
```

```
Collecting plotly
  Downloading plotly-4.14.3-py2.py3-none-any.whl (13.2 MB)
Requirement already satisfied: six in c:\users\kiit\anaconda3\lib\site-packages (from pl
otly) (1.15.0)
```

```
Collecting retrying>=1.3.3
  Downloading retrying-1.3.3.tar.gz (10 kB)
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py): started
  Building wheel for retrying (setup.py): finished with status 'done'
  Created wheel for retrying: filename=retrying-1.3.3-py3-none-any.whl size=11434 sha256
=06af50b83549c20d4d818b4ad03467510e336a98c7e81869ad87cc3711828278
  Stored in directory: c:\users\kiit\appdata\local\pip\cache\wheels\c4\a7\48\0a434133f6d
56e878ca511c0e6c38326907c0792f67b476e56
Successfully built retrying
Installing collected packages: retrying, plotly
Successfully installed plotly-4.14.3 retrying-1.3.3
```

In [82]:

```
from pyAudioAnalysis import audioTrainTest as aT
aT.extract_features_and_train(["classifierData/music","classifier
 1.0, 1.0, aT.shortTermWindow, aT.shortTermStep, "svm",
"svmSMtemp", False)
aT.file_classification("python scripts/cool.wav",
"svmSMtemp","svm")
```

```
trainSVM_feature ERROR: No data found in any input folder!
fileClassification: input model_name not found!
```

Out[82]: (-1, -1, -1)

gTT

In [31]:

```
! pip install gTTS
```

```
Collecting gTTS
  Downloading gTTS-2.2.2-py3-none-any.whl (25 kB)
Requirement already satisfied: click in c:\users\kiit\anaconda3\lib\site-packages (from
gTTS) (7.1.2)
Requirement already satisfied: requests in c:\users\kiit\anaconda3\lib\site-packages (fr
om gTTS) (2.24.0)
Requirement already satisfied: six in c:\users\kiit\anaconda3\lib\site-packages (from gT
TS) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\kiit\anaconda3\lib\site-packages
(from requests->gTTS) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\kiit
\anaconda3\lib\site-packages (from requests->gTTS) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\kiit\anaconda3\lib\site-pac
kages (from requests->gTTS) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kiit\anaconda3\lib\site-pa
ckages (from requests->gTTS) (2020.6.20)
Installing collected packages: gTTS
Successfully installed gTTS-2.2.2
```

In [ ]: