List =[1,2,’c’]

append() : append a new list as a list at the end

extend() : append the list and assimilate both

index() : gives the index of an element

remove() : removes an element

pop() : removes last element(LIFO stack)

pop(0): removes first element(FIFO que)

count(a) : counts a's in the list

sort(reverse=True): sorts my ass

a=a+[1] : extends list

a=a\*2 : doubles elements

list[first:last:step] for slicing

[i for i in list if i==0] returns a list where element is 0. efficient

l1=l2 does not copy list. creates a reference

l1=list(l2), l1=l2[:],l1=copy.copy(1)

l1=copy.deepcopy(l2) copies lists inside a list(prevents reference)

bisect.insort(l1,3) : inserts element in a sorted list

Tuple=(1,2,’c’)

(1,2) or 1,2

value cant be changed

single unit =(1,) or 1,

count(a) : count a's

faster than lists

dict([('a',1)('b',2)]) is same as {'a':1,'b':2}

can be used to unpack values into variables.

len(t1) gives length

unlike list, tuple can be copied using '=', no references created

a list inside a tuple is mutable

Dictionary={‘a’:1,’b’:2}

d={'a':1,'b':2}

d.keys() gives key values

d.values() gives values

d.items() gives one key,value pair

d.has\_key('key') checks for that key or [key in d]

d.get('key') gives value

d.pop('key') gives and removes from dict

d.popitem() give dict pair.

copy() is used to copy dicts

clear() truncates dictionary

s.setdefault('key','value') , returns value of key if it exists, else returns 'value'

d.fromkeys(['k1','k2']) creates sub dicts from these keys. If these keys dont exist, them these keys correspond to empty values.

d1.update(d2) updates d1 with the values of d2

set(1,2,3)

a|b union of sets

a&b or a.intersection(b) intersection

a.issubset(b) or a<=b

a.issuperset(b) or a=>b

a^b , join half of this and half of that.

Series

pd.Series(l) list to pandas series

dictionary can be converted to pandas series and vice versa using dict() and pd.Series(s)

index can be added seperately in a series as pd.Series([1,2,3],index=3,4,5)

to access p[['row1','row2','row2']] where row is the index.

Just like R, conditions can be used p[p>3], the p>3 condition can be stored in a variable.

Items in a Series can be iterated through.

operations can be performed on the entire Series. (notnull(),isnull())

DataFrame

dictionary can be converted into a dataframe.

to select columns in a dataframe from a dict, we select the keys as colums as pd.DataFrame(d,columns=[c1,c1])

pd.read\_clipboard() reads data from clipboard.

From a numpy array=array(range(25)) which is changed into a 5x5 matrix as array.reshape(5,5), conversion can be done to a DF, while setting columns and indexes as

pd.DataFrame(v,columns=['a','b','c','d','e'],index=[1,2,3,4,5])

check for not null/null d1[pd.notnull(d1['time'])] other method is isnull()

Sort d1.sort('time')

D1.join(d2) to join

D1.column=[‘new col name’]

D1.drop=[‘drop col’]

Change column name : d3.rename(columns={'date':'time'},inplace=True)

Code to convert string to variable:

#a=’b’

#exec(a+'=c')

#print b