**Pandas Notes**

**October 26th**

Pandas has 2 important data structures: DataFrames and Series

import pandas as pd

from pandas import Series

Series: 1D array of a particular NumPy type and an associated array of data labels called index.

Ser = Series([1, 2, 3])

Ser.values -> list -> [1, 2, 3]

Ser.index ->RangeIndex(start, stop, step)

Ser.values.dtype

Ser.index.dtype

(In this case both are int)

ser = Series([1, 5, -2, 16], index=['a','b','x','d'])

ser['x'] -> returns -2

you can change the series by assigning this to something

ser[['x','a','b']] # like NumPy syntax to access rows

returns:

x -2

b 5

a 1

Arithmetic operations & filtering similar to Numpy array.

Series is like an ordered dictionary.

‘x’ in ser -> returns a Boolean after checking if ‘x’ is an index

You can create a series out of a python dictionary.

make a new series out of a subset of useries plus another univ name only.

univs = ['Purdue','Rutgers','MIT','USC']

useries2 = Series(udict, index=univs)

**In the above, indexes common to the argument index (univs) and the udict index are kept with their udict values But for any index in univs that is not in udict, the value is NaN.**

Dictionaries could also have list values and you will get the corresponding series.

Checking for null/not null values

useries2.isnull() # NaN is equivalent to null

Purdue True

Rutgers False

MIT False

USC False

Similarly, you can do useries2.notnull() to get the opposite result [False then all Trues]

You can give a name to the series and its index

useries.name = "student population"

useries.index.name = "university"

# if an index appears in one and not the other, result here is NaN

useries + useries2

You could change the index by doing useries.index = […..]

Nan = np.nan

Dat:

0 1.0

1 NaN

2 2.6

3 NaN

4 6.0

Dat.dropna()

0 1.0

2 2.6

4 6.0

(Is not in place)

# alternatively, you can filter with notnull, this is also not in place

dat[dat.notnull()]

dat1 = dat.copy() //gets a copy not view

To do above in place

dat1 = dat1[dat1.notnull()]

dat1.dropna(inplace=True)

useries = useries.reset\_index() # index becomes a column

useries is now a dataframe.

Changing name of columns in a dataframe:

useries.columns = ['Univ','Student Population']