Week2

We talked about puty?/ list/ dictionary/ numpy array and now about series

It is between lists and dictionaries

It is like dictionary where you have a key to access the data but in this case it is an index

We use pandas as it is much faster and more effiecient to deal with data in pandas series compared to python lists

To check if you have a nan you use np.isnan(variable)

If we use pd.Series() with a dictionary then the index will be the key of the dictionary rather than numerical integers

For a panda series to get the index use the function seriesName.index

Keep checking the notebook it has very useful info!

The difference between data series and data frame in pandas is that the data frame is multi columns

A numpy array can also be multi dimensional btw

The way we work with dataframes is we read the data frame then we reduce it to the rows and columns we want to process first

The problem with pandas is that it will manipulate the original dataset

So better to create a copy and then work with it

The function pd.read\_csv is the function we need to read the csv file

This function has multiple parameters to indicate what we want to use

df = pd.read\_csv('olympics.csv', index\_col = 0, skiprows=1)

some of the parameters with pd.read\_csv function

df.columns give the columns name

to rename a column in a data frame

df.rename(columns ={}, inplace = true)

\_\_

Querying a dataframe in Pandas is done by what is called Boolen masking

df['Gold'] > 0 will apply the boolen mask and return a series

if I wanna return a dataframe with it then we use the4 function where

only\_gold = df.where(df['Gold'] > 0)

df[] is the normal way with index inside

however you can write a condition inside

df[df[‘Gold’] > 0]

if you have more than one operation inside the index then each operation have to closed by ()

e.g. df[(df['Gold.1'] > 0) & (df['Gold'] == 0)]