**Some ways of running a Python code**

1. Using the command line (python filename.py)
2. Using an appropriate python IDE
3. Jupyter notebook / Google colab

**%matplotlib inline is used for** 🡪 %matplotlib inline sets the backend of matplotlib to the 'inline' backend. With this, the outputs of plotting commands are displayed inline within frontends like the Jupyter notebook and Google colab. We can have those GUI s directly below the code segment that produces it.

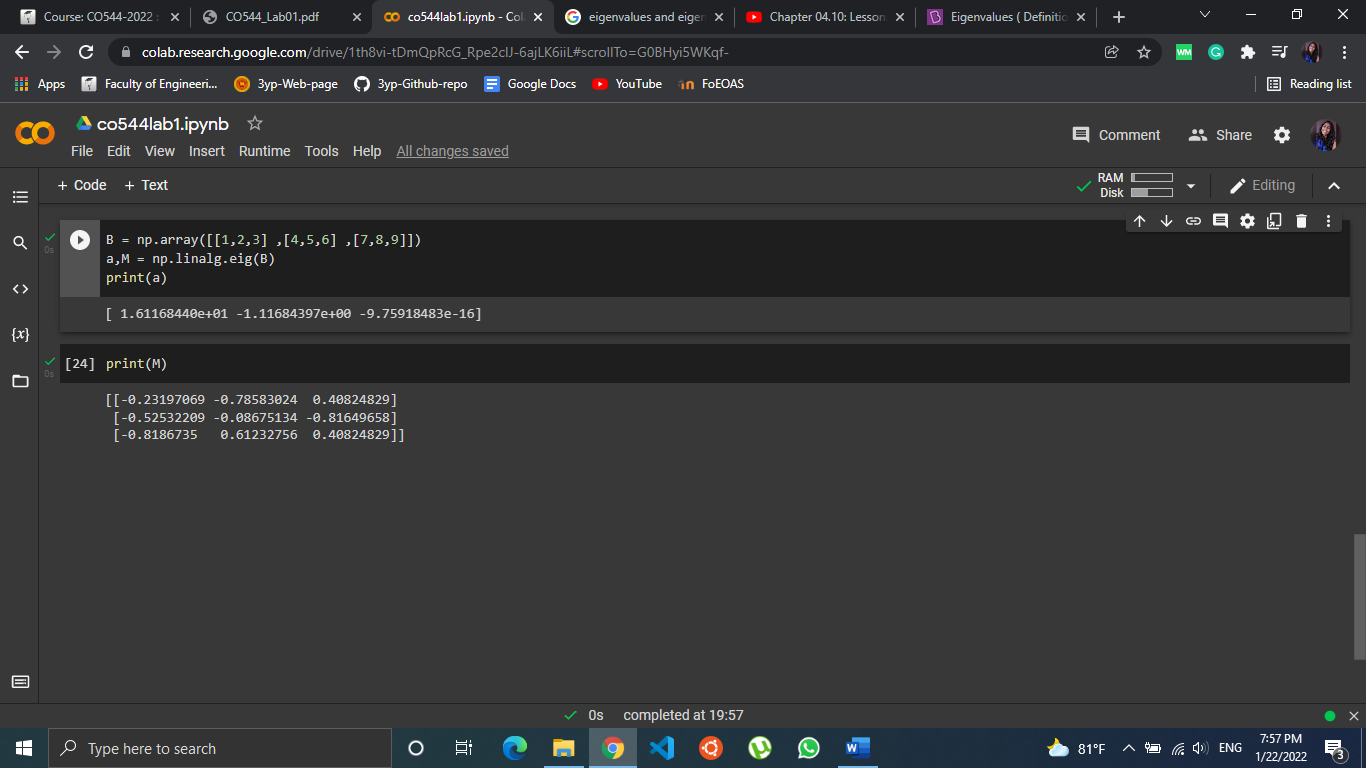
**Eigen Values and Eigen Vector**

If [A] is an nxn matrix (square matrix), Then, [X] ≠ **0** is an Eigen vector of [A] if,

[A][X] = λ[X] and λ is the Eigen value where λ is a scaler. An nxn matrix has n number of eigen values (λ1,λ2,λ3,…λn) and corresponding to each eigen value, we have an eigen vector.

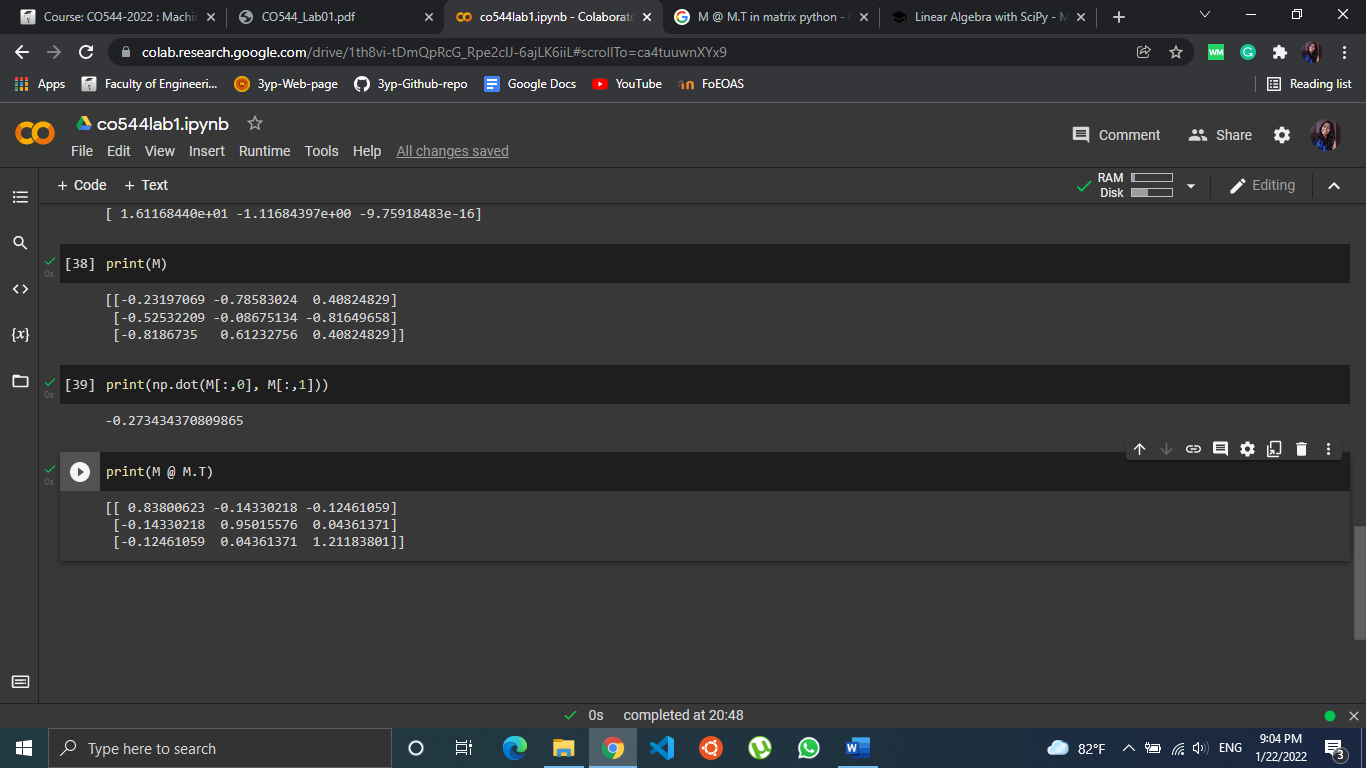
In the lab, we took the matrix B as, B = np.array([[1, 2, 3],[4, 5, 6],[7, 8, 9]])

By executing🡪 a, M = np.linalg.eig(B) we can directly find **a** which represents the vector containing the eigenvalues and **M** represents the array containing the corresponding eigenvectors.



**Singular Value Decomposition (SVD)**

We use the @ operator to do matrix multiplication with numpy arrays.

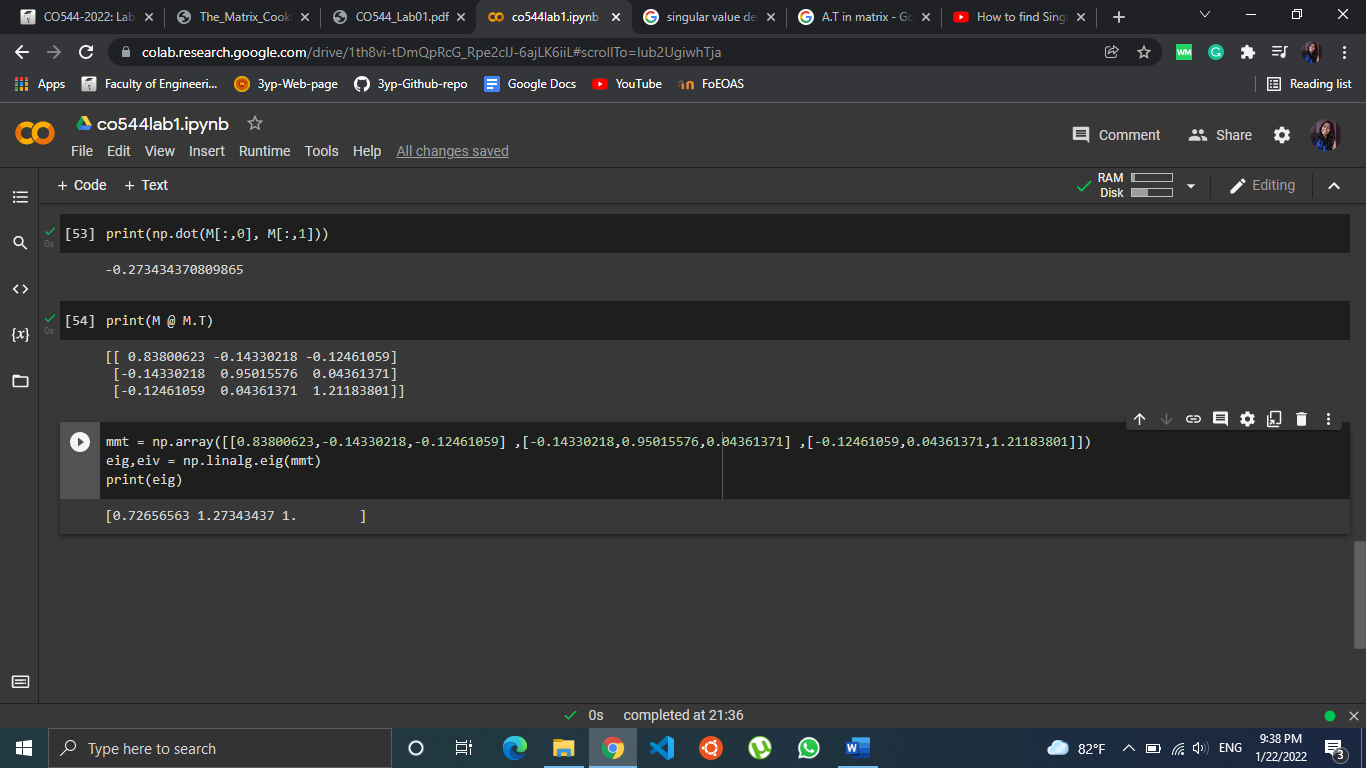


Any nxm matrix can be written as, **A = UDVT** where,

U = Eigen vectors of AAT (AT = A transpose)

D = diag(eig(AAT ))

V = eigenvectors of ATA



####should complete

**Random Numbers and Univariate Distributions**

Choice of the number of bins and the number of samples affects the histogram. When the number of samples go up, the variation within bin counts gets low.

This can be taken as an aspect of Central Limit Theorem (CLT) 🡪 When drawing a single random sample, the larger the sample is the closer the sample mean will be to the population mean. Therefore, when drawing an infinite number of random samples, the variance of the sampling distribution will be lower the larger the size of each sample is.

**Uncertainty in Estimation**

Variance (σ2)

Variance refers to a statistical measurement of the spread between numbers in a data set. More specifically, variance measures how far each number in the set is from the mean and thus from every other number in the set.

σ2 =

*x* = Data point , *µ* = Mean , *N* = Number of data elements(Sample size)

Standard Deviation (σ)

Standard deviation is a measure of how dispersed the data is in relation to the mean. Low standard deviation means data are clustered around the mean, and high standard deviation indicates data are more spread out.

**Gaussian Distributions**

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. In graph form, normal distribution will appear as a bell curve.

Univariate Gaussian Distribution

This consists one random variable. This is used in measurements and error estimating processes, Gaussian filtering in image processing, in matrices and statistics etc.

Eg: Suppose we wish to model the distribution of returns on an asset, such as a holding of stocks; such a model would be a univariate distribution.

Multivariate Gaussian Distribution

This is a multidimensional generalization of the one-dimensional normal distribution. It represents the distribution of a multivariate random variable, that is made up of multiple random variables which can be correlated with each other. Each variable has its own mean and variance.

The multivariate normal distribution is useful in analyzing the relationship between multiple normally distributed variables. So, this has a great impact on economy and biology applications where the relationship between approximately-normal variables is highly considered.

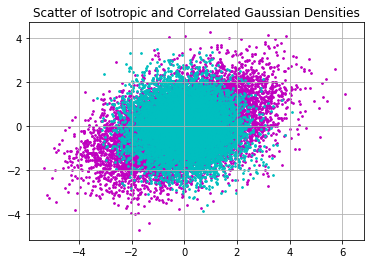
Bi-variate Gaussian distribution

This is a special case of multivariate distributions where, there are two random variables which are normally distributed.

**Cholosky decomposition**

ATA = C where C is the covariance matrix.

Let’s generate 10000 bivariate Gaussian random numbers by X = np.random.randn(1000, 2) Then, Transform each of the two dimensional vectors (rows of X) by Y = X @ A Next, plot the scatter plot of X and Y ]  
Observation:



**Distribution of Projections**

####should complete

