**CO 544 - Machine Learning and Data Mining**

**Lab 01**

**E/17/407**

**Wijesooriya H.D**

* **Ways to run a python file** 
  + Using the command line: python filename.py is the command that we use.
  + Using an IDE
  + Jupyter Notebook / Google colab

1. **Linear Algebra**

**1. Scalar (dot) product: a=x.y:** return the dot product between two vectors. Let the two vectors be X=[x1,x2] and Y=[y1,y2] then the dot product = x1y1+x2y2.

x = np.array([1, 5]) #initialized the vectors

y = np.array([2, 3])

a = np.dot(x, y) #get the dot product

**2. Linear combinations of two vectors:** a vector d is said to be a linear combination of two vectors x and y if there exist scalars 𝛌 and 𝝱, such that d= 𝛌x+𝝱y

**3. Norm of a vector:** The magnitude of a vector is sometimes called the length of a vector or norm of the vector. Let the vector be X=[x1,x2] then, the norm ||c||=

c = np.linalg.norm(x)

**4. Angle between two none zero vectors:** θ = arccos ( x.y / ∥x∥.∥y∥). Here we get the result in radians.

**5.Matrix vector multiplication: v = Au**

**6.. Solve a linear matrix equation, or system of linear scalar equations:** Let’s consider a system with two variables. We can put the coefficients of the variables to one array and the constants to another array and solve them as shown below

2x1 + 5x2 = 3 and 3x1 − 5x2 = 2

d = np.array([[2, 5], [3, -5]]) # coefficients of the variables

e = np.array([3, 2]) #constants

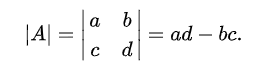
results = np.linalg.solve(d, e)

**7. Inverse of a matrix:** For a matrix A, its inverse is A-1, and A.A-1 = I.

d\_inv = np.linalg.inv(d)

**8. Trace of a matrix:** Trace of a matrix is the sum of the elements in the main diagonal. trace = np.trace(B)

**9. Determinant of a matrix: det(A) or |A|:** It is a [scalar value](https://en.wikipedia.org/wiki/Scalar_(mathematics)) that is a [function](https://en.wikipedia.org/wiki/Function_(mathematics)) of the entries of a [square matrix](https://en.wikipedia.org/wiki/Square_matrix). It allows characterizing some properties of the matrix and the [linear map](https://en.wikipedia.org/wiki/Linear_map) represented by the matrix. The determinant of a 2x2 matrix can be calculated as follow. A\_det = np.linalg.det(A)



**10.Eigenvalues and Eigenvectors: Au = λu :** If Au=λu for u≠0, we say that λ is the *eigenvalue for* u, and that u is an *eigenvector for* λ.

a, M = np.linalg.eig(B) # a- eigen values M-eigen vectors

In order to find the eigen values (𝛌) we need to solve, det(A-𝛌I)=0 where I is the identity matrix. ( I=numpy.identity(n) # nxn identity matrix) If A is an nxn matrix then we have n number of 𝛌 values. By solving (A-𝛌I)u=0 for different 𝛌 values, we can find the corresponding eigen vectors (u).

**11. Singular Value Decomposition (SVD)**

print(M @ M.T) - The ‘@’ sign is an operator which performs the actual matrix multiplication between the matrices M and M.T . Here M.T is the transpose matrix of M. The output of this command is equal to the dot product between the eigen values of the matrix B.

SVD: Any n × m matrix A can be written as, *A* = where,

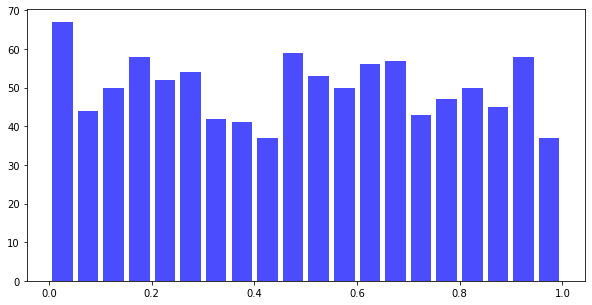
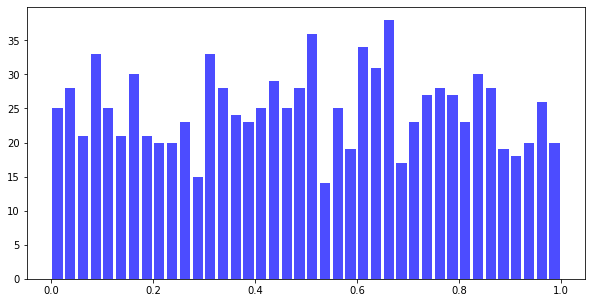
U = eigenvectors of n × n

D = n x m

V = eigenvectors of m x m

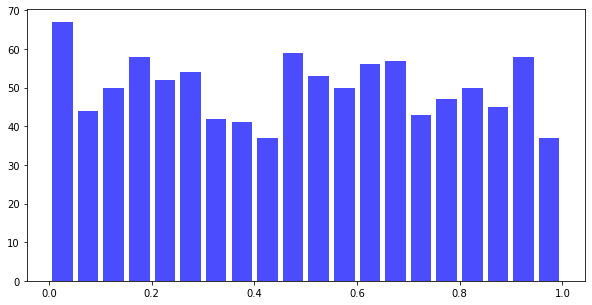
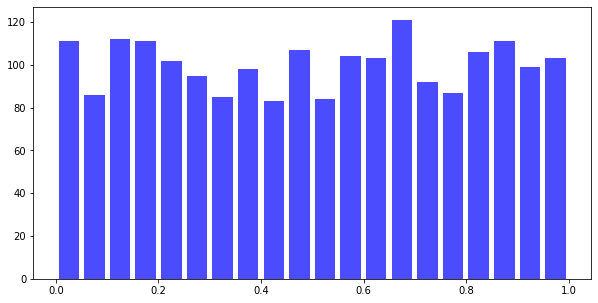
1. **Random Numbers and Univariate Distributions**

**Effect of the number of bins:** the number of bars (columns) in the histogram increases as the number of bins increases.

** **

bins=20 bins=40

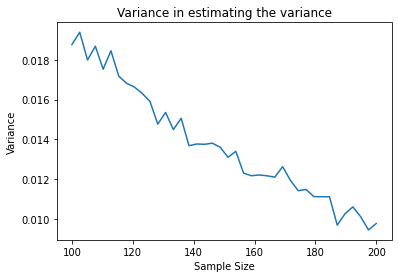
**Effect of the number of samples:**

** **

n=1000 n=2000

1. **Uncertainty in Estimation**

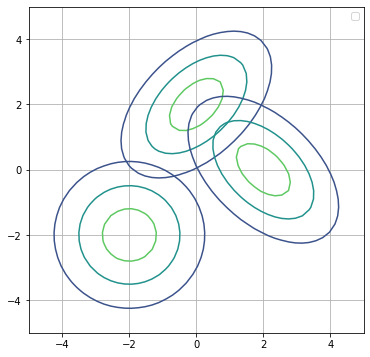
As the sample size becomes larger, the variance becomes smaller. Therefore having a large data set is a good thing.



1. **Bivariate Gaussian Distribution**

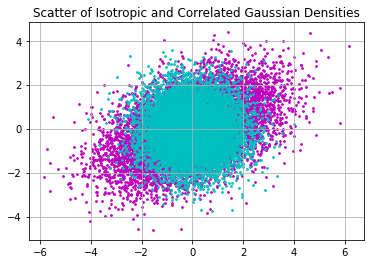
There are 2 random variables and both are normally distributed. A 3D bell curve was obtained.

More about graphs…explain the shapes

****

**5. Sampling from a Multivariate Gaussian Distribution**

There can be two or more random variables.Here we obtained the distribution of projections of the vectors in the space.This concept is used in principal component analysis (PCA). In PCA the dimensions of the data are reduced.So it helps to minimize the data reconstruction cost.



**6. Distribution of Projections**