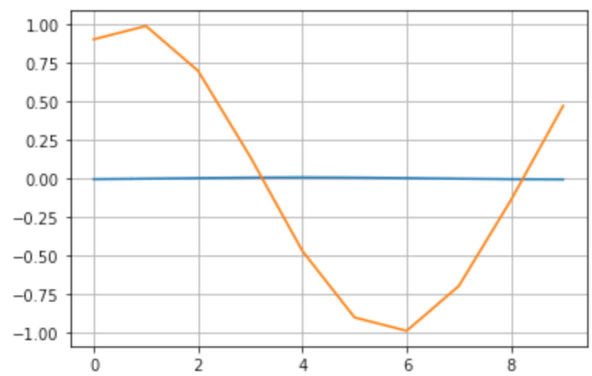
CT303 - Digital Communications
Autumn 2020
Lab *2:Random Processes Stationarity* Group: *A9*

1. (a) We select random values of theta, which is uniformly distributed between –pi and pi for n_exp (=10,000) iterations. In every iteration of the loop we compute the value of

$$X(n) = cos(0.2\pi n + \theta)$$

where n ranges from [0,9].
After generating a n_exp n (10,000 x 10) matrix, we find the mean and auto-correlation. On plotting the mean, it turns out to be constant and the auto correlation(n1,n2) does not change by shift in n. Hence the stochastic function is wide-sense stationary.

```
plt.plot(np.arange(10),means2)
plt.plot(np.arange(N),matrix[3][:])
plt.grid()
plt.show()
```



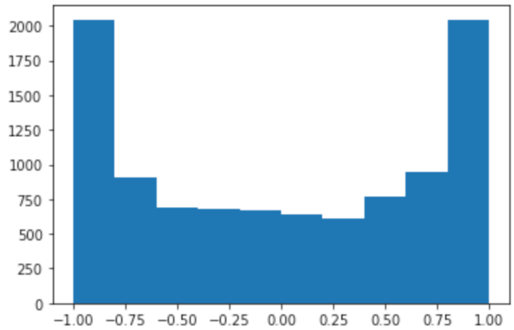

```
R = df.corr()
R
```

	0	1	2	3	4	5	6	7	8	9
0	1.000000	0.809242	0.316306	-0.297704	-0.804798	-1.000000	-0.809242	-0.316306	0.297704	0.804798
1	0.809242	1.000000	0.813281	0.319923	-0.302582	-0.809242	-1.000000	-0.813281	-0.319923	0.302582
2	0.316306	0.813281	1.000000	0.811478	0.308512	-0.316306	-0.813281	-1.000000	-0.811478	-0.308512
3	-0.297704	0.319923	0.811478	1.000000	0.806228	0.297704	-0.319923	-0.811478	-1.000000	-0.806228
4	-0.804798	-0.302582	0.308512	0.806228	1.000000	0.804798	0.302582	-0.308512	-0.806228	-1.000000
5	-1.000000	-0.809242	-0.316306	0.297704	0.804798	1.000000	0.809242	0.316306	-0.297704	-0.804798
6	-0.809242	-1.000000	-0.813281	-0.319923	0.302582	0.809242	1.000000	0.813281	0.319923	-0.302582
7	-0.316306	-0.813281	-1.000000	-0.811478	-0.308512	0.316306	0.813281	1.000000	0.811478	0.308512
8	0.297704	-0.319923	-0.811478	-1.000000	-0.806228	-0.297704	0.319923	0.811478	1.000000	0.806228
9	0.804798	0.302582	-0.308512	-0.806228	-1.000000	-0.804798	-0.302582	0.308512	0.806228	1.000000

The estimated density functions X(k) for k = 3 and k = 4 are shown below

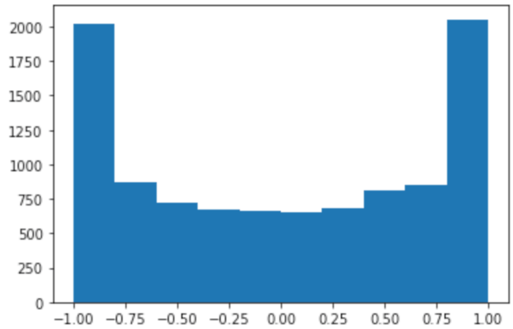
```
plt.hist(df[3])
```

```
(array([2045.,  902.,  687.,  681.,  668.,  644.,  615.,  770.,  944.,
        2044.]),
 array([-9.99999916e-01, -7.99999948e-01, -5.99999981e-01, -4.00000014e-01,
        -2.00000047e-01, -7.96459022e-08,  1.99999888e-01,  3.99999855e-01,
         5.99999822e-01,  7.99999789e-01,  9.99999756e-01]),
 <a list of 10 Patch objects>)
```

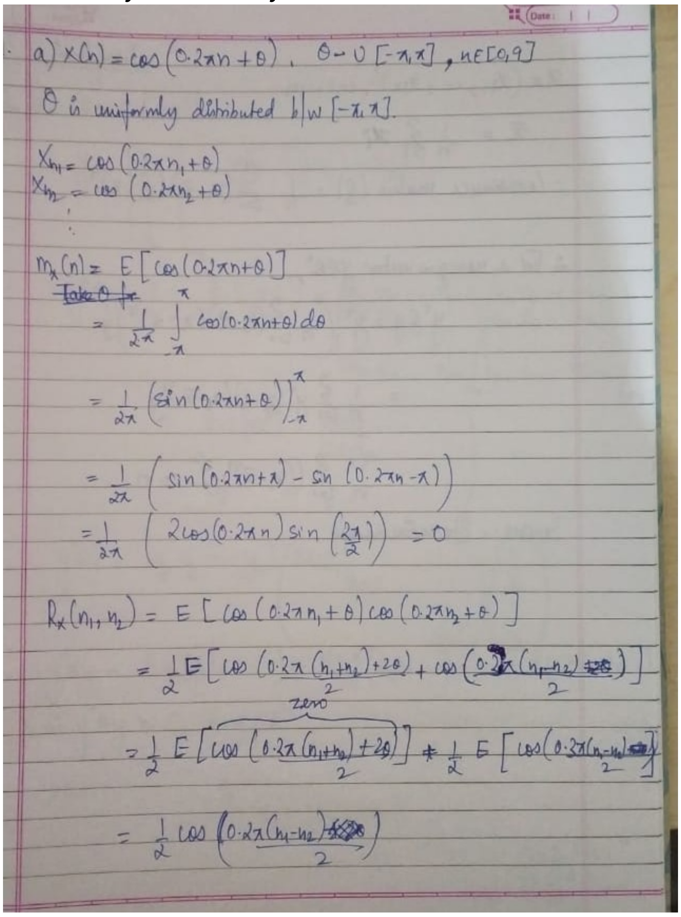



```
plt.hist(df[4])
```

```
(array([2016.,  869.,  722.,  673.,  664.,  655.,  687.,  807.,  854.,
        2053.]),
 array([-9.99999997e-01, -8.00000006e-01, -6.00000015e-01, -4.00000024e-01,
        -2.00000032e-01, -4.11566887e-08,  1.99999950e-01,  3.99999941e-01,
         5.99999933e-01,  7.99999924e-01,  9.99999915e-01]),
 <a list of 10 Patch objects>)
```



The analytical analysis for Q1 A) is as shown below



a) $x(n) = \cos(0.2\pi n + \theta)$, $\theta \sim U[-\pi,\pi]$, $n \in [0,9]$

θ is uniformly distributed b/w $[-\pi,\pi]$.

$X_{n_1} = \cos(0.2\pi n_1 + \theta)$

$X_{n_2} = \cos(0.2\pi n_2 + \theta)$

$\vdots$

$m_x(n) = E[\cos(0.2\pi n + \theta)]$

~~Take θ fr~~

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \cos(0.2\pi n + \theta)\,d\theta$$

$$= \frac{1}{2\pi}\left(\sin(0.2\pi n + \theta)\right)_{-\pi}^{\pi}$$

$$= \frac{1}{2\pi}\left(\sin(0.2\pi n + \pi) - \sin(0.2\pi n - \pi)\right)$$

$$= \frac{1}{2\pi}\left(2\cos(0.2\pi n)\sin\left(\frac{2\pi}{2}\right)\right) = 0$$

$$R_x(n_1,n_2) = E\left[\cos(0.2\pi n_1 + \theta)\cos(0.2\pi n_2 + \theta)\right]$$

$$= \frac{1}{2}E\left[\cos\left(0.2\pi\frac{(n_1+n_2)}{2} + 2\theta\right) + \cos\left(0.2\pi\frac{(n_1-n_2)}{2}\right)\right]$$

$$= \frac{1}{2}E\left[\overbrace{\cos\left(0.2\pi\frac{(n_1+n_2)}{2} + 2\theta\right)}^{\text{zero}}\right] + \frac{1}{2}E\left[\cos\left(0.2\pi\frac{(n_1-n_2)}{2}\right)\right]$$

$$= \frac{1}{2}\cos\left(0.2\pi\frac{(n_1-n_2)}{2}\right)$$

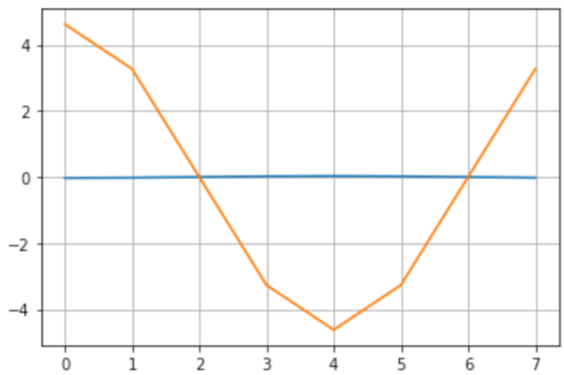
(b) We select random values of A, which is uniformly distributed between –5 and 5 for n_exp (=15,000) iterations. In every iteration of the loop we compute the value of

$$X(n) = A\cos(0.25\pi n)$$

where n ranges from [0,7].

After generating a n_exp n (15,000 x 8) matrix, we find the mean and auto-correlation. On plotting the mean, it turns out to be constant and the auto correlation(n1,n2)changes by shift in n. Hence the stochastic function is not wide-sense stationary.

```python
plt.plot(np.arange(N),means)
plt.plot(np.arange(N),matrix[3][:])
plt.grid()
plt.show()
```



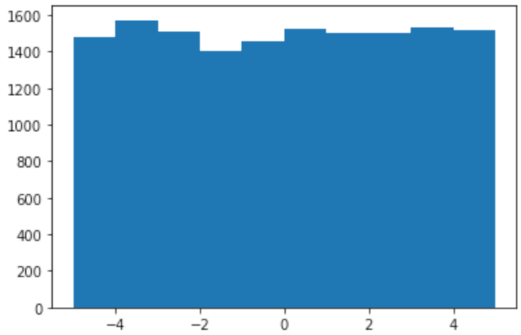
```python
R = df.corr()
R
```

	0	1	2	3	4	5	6	7
0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	1.0
1	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	1.0
2	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	1.0
3	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	-1.0
4	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	-1.0
5	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	-1.0
6	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	-1.0
7	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	1.0

The estimated density functions X(k) for k = 0 is shown below

```python
plt.hist(df[0])
```

```
(array([1480., 1573., 1508., 1402., 1458., 1527., 1500., 1506., 1532.,
        1514.]),
 array([-4.99966840e+00, -3.99973958e+00, -2.99981076e+00, -1.99988194e+00,
        -9.99953123e-01, -2.43031946e-05,  9.99904517e-01,  1.99983334e+00,
         2.99976216e+00,  3.99969098e+00,  4.99961980e+00]),
 <a list of 10 Patch objects>)
```



The analytical analysis for Q1 B) is as shown below

b) $x(n) = A\cos(0.25\pi n)$, $A \sim U[-5,5]$, $n \in [0,7]$
A is randomly generated.

$m_x(n) = E[X(n)]$
$= E[A\cos(0.25\pi n)]$
$= E[A]\cos(0.25\pi n)$, $E[A] = \frac{5+(-5)}{2} = 0$

$R_x(n_1,n_2) = E[A\cos(0.25\pi n_1)\, A\cos(0.25\pi n_2)]$
$= E[A^2]\cos(0.25\pi n_1)\cos(0.25\pi n_2)$
$= \frac{1}{2}(5-(-5))^2 \times \cos\left(\frac{\pi n_1}{4}\right)\cos\left(\frac{\pi n_2}{4}\right)$
$= 50\cos\left(\frac{\pi n_1}{4}\right)\cos\left(\frac{\pi n_2}{4}\right)$

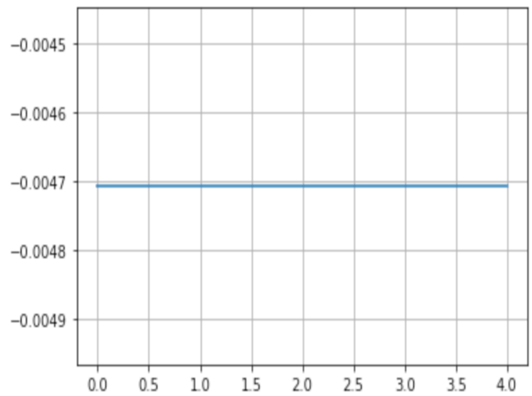
$R_x(n_1,n_2)$ not dependent on $n_1 - n_2$
$\Rightarrow$ Not wide sense stationary.

(c) We generate a random process

$$X(n) = A(n)$$

such that A(n) is normally distributed with mean 0 and variance 1. On finding the mean it turns out to be constant and the auto variance is same for equidistant.

```
plt.plot(np.arange(N),means)
plt.grid()
plt.show()
```



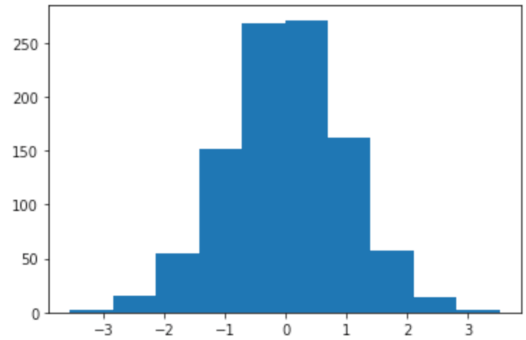

```
R = df.corr()
R
```

	0	1	2	3	4
0	1.0	1.0	1.0	1.0	1.0
1	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0

The estimated density functions X(k) for k = 1 and k = 3 are shown below

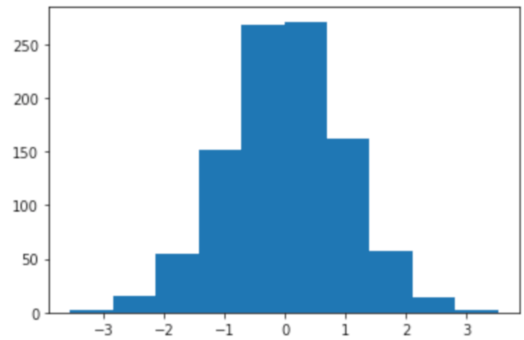
```
plt.hist(df[1])
```

```
(array([  3.,  16.,  55., 151., 268., 271., 162.,  57.,  14.,   3.]),
 array([-3.5489131 , -2.84217061, -2.13542812, -1.42868563, -0.72194314,
        -0.01520065,  0.69154184,  1.39828433,  2.10502682,  2.81176931,
         3.5185118 ]),
 <a list of 10 Patch objects>)
```

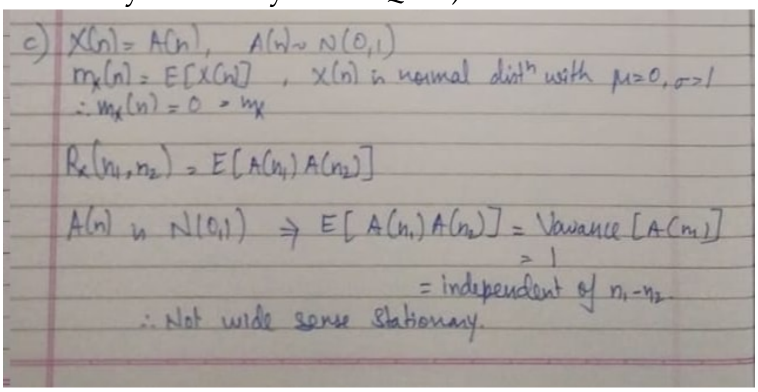



```
plt.hist(df[3])
```

```
(array([  3.,  16.,  55., 151., 268., 271., 162.,  57.,  14.,   3.]),
 array([-3.5489131 , -2.84217061, -2.13542812, -1.42868563, -0.72194314,
        -0.01520065,  0.69154184,  1.39828433,  2.10502682,  2.81176931,
         3.5185118 ]),
 <a list of 10 Patch objects>)
```



The analytical analysis for Q1 C) is as shown below


c) $X(n) = A(n)$, $A(n) \sim N(0,1)$

$m_x(n) = E[X(n)]$, $X(n)$ is normal distn with $\mu = 0, \sigma = 1$

$\therefore m_x(n) = 0 = m_x$

$R_x(n_1, n_2) = E[A(n_1) A(n_2)]$

$A(n)$ is $N(0,1)$ $\Rightarrow E[A(n_1) A(n_2)] = \text{Variance}[A(n_1)]$

$= 1$

= independent of $n_1 - n_2$

$\therefore$ Not wide sense stationary.

2. We created N realizations of every random variable which when combined all together constitute a random process. We took N<=100 values for every month and then took the mean. The mean varied a lot with lowest in the starting and the ending months of the year and highest during the post summer months. Since the mean is not constant we can conclude that the data is not wide sense stationary.

```
mean
```

```
[1.3509803921568628,
 1.1794117647058828,
 1.215686274509804,
 0.719607843137255,
 9.374509803921573,
 85.09019607843136,
 272.6578431372549,
 170.53627450980392,
 112.89607843137256,
 13.887254901960787,
 6.35,
 1.8]
```

```
rainfall_data.cov()
```

	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12
Unnamed: 1	9.064306	-1.263395	-0.431600	0.340971	-2.464628	3.571198	-40.628325	60.424964	-16.963065	-8.740136	4.114554	-1.955446
Unnamed: 2	-1.263395	6.818879	-0.060565	0.033378	5.120262	8.116826	-19.295926	10.987388	0.408928	1.132903	-0.102822	0.514851
Unnamed: 3	-0.431600	-0.060565	10.924108	-0.480707	-6.645339	9.727878	-19.223689	-56.204733	18.392339	-7.232372	-1.133960	0.196337
Unnamed: 4	0.340971	0.033378	-0.480707	1.991097	2.804861	-5.452974	21.952914	28.446905	-1.077249	2.033817	6.600891	-0.178812
Unnamed: 5	-2.464628	5.120262	-6.645339	2.804861	268.046869	114.443213	-174.428115	-34.029066	-193.603962	32.848682	26.885248	0.020594
Unnamed: 6	3.571198	8.116826	9.727878	-5.452974	114.443213	4154.128418	506.391365	-616.898057	-127.901623	-3.113889	-53.069307	15.097228
Unnamed: 7	-40.628325	-19.295926	-19.223689	21.952914	-174.428115	506.391365	18359.130383	-1371.055386	1975.692209	159.258764	-268.352129	-61.004851
Unnamed: 8	60.424964	10.987388	-56.204733	28.446905	-34.029066	-616.898057	-1371.055386	11011.035008	2074.260342	159.720071	275.311832	80.383069
Unnamed: 9	-16.963065	0.408928	18.392339	-1.077249	-193.603962	-127.901623	1975.692209	2074.260342	13024.385331	694.571633	-259.276040	-88.863564
Unnamed: 10	-8.740136	1.132903	-7.232372	2.033817	32.848682	-3.113889	159.258764	159.720071	694.571633	534.132608	-3.770545	-10.950099
Unnamed: 11	4.114554	-0.102822	-1.133960	6.600891	26.885248	-53.069307	-268.352129	275.311832	-259.276040	-3.770545	277.653614	-2.612475
Unnamed: 12	-1.955446	0.514851	0.196337	-0.178812	0.020594	15.097228	-61.004851	80.383069	-88.863564	-10.950099	-2.612475	26.972673

3. (a) We generate a random matrix y . We have already computed the autocorrelation matrix R and autocovariance matrix C for all the random functions generated in the first question. Taking the dot product yî Ry and yî Cy we get positive results for both the equations

$$X(n) = cos(0.2\pi n + \theta)$$

```python
y = np.random.randint(100,size=(N,1))
y_trans = np.transpose(y)

f = np.dot(y_trans,R) # For Correlation Matrix
ans = np.dot(f,y)
print(ans)

f = np.dot(y_trans,C) # For Covariance Matrix
ans = np.dot(f,y)
print(ans)
```

```
[[22577.50913951]]
[[11285.34499064]]
```

$$X(n) = A(n)$$

```python
y = np.random.randint(100,size=(N,1))
y_trans = np.transpose(y)

f = np.dot(y_trans,R) # For Correlation Matrix
ans = np.dot(f,y)
print(ans)

f = np.dot(y_trans,C) # For Covariance Matrix
ans = np.dot(f,y)
print(ans)
```

```
[[41616.]]
[[40460.45580685]]
```

The analytical analysis is as shown below

Page No.

Date: / /

Analytical

3. (a) $R = E[xx^T]$

$\therefore y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n\times 1} \quad y^T = [y_1\ y_2 \cdots y_n]_{1\times n}$

$\therefore y^T R y = ? \quad R = E[xx^T] = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & & \vdots \\ \vdots & \vdots & & \vdots \\ x_{1n} & x_{2n} & & x_{nn} \end{pmatrix}_{n\times n} \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix}_{n\times n}$

$R = E[xx^T] = \begin{pmatrix} \Sigma x_{i1}^2 & \cdots & \cdots \\ \Sigma x_{ij} x_{pk} & & \vdots \\ \vdots & \cdots & \vdots \end{pmatrix}$

$\therefore y^T R y = (y_1\ y_2 \cdots y_n) \begin{pmatrix} \Sigma x_{i1}^2 & & & \\ & \Sigma x_{i2}^2 & & \\ & & \ddots & \\ & & & \Sigma x_{in}^2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$

$y^T R y = \Sigma y_1^2 (x_{i1}^2 + \cdots) + \Sigma y_2^2 (x_{i2}^2 + \cdots) + \cdots + \Sigma y_n^2 (x_{in}^2 + \cdots)$

$\therefore y^T R y \geq 0 \ \forall\, y \in R^n$

$x = (x_{i1}, \cdots, x_{ik})^T$, $i = 1 \cdots n$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$\therefore$ Covariance matrix $(Q) = \frac{1}{n}\sum_{i=1}^{n} (x_i-\bar{x})(x_i-\bar{x})^T$

$\therefore$ For a nonzero vector $y \in \mathbb{R}^k$,

$$y^T Q y = y^T\left(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})^T\right) y$$

$$= \frac{1}{n}\sum_{i=1}^{n} y^T (x_i-\bar{x})(x_i-\bar{x})^T y$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left((x_i-\bar{x})^T y\right)^2 \geq 0$$

Source:- StackExchange

(b) We noticed a pattern, the diagonal has equal values and the corresponding values above and below the diagonal are similar. Hence the matrix is a symmetrical matrix.