به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس سیستم‌های هوشمند

تمرین شماره چهارم

نام و نام خانوادگی:

اشکان جعفری فشارکی

شماره دانشجویی:

۸۱۰۱۹۷۴۸۳

دی ۱۴۰۰

فهرست سوالات

Question#1

A)

According to Figure1, first we choose 2 random centroids and calculate the Euclidean distance for each data and compare the distance between each point and two centroids. For each point the lower calculated distance to the centroid assigns the point to that centroid and cluster.

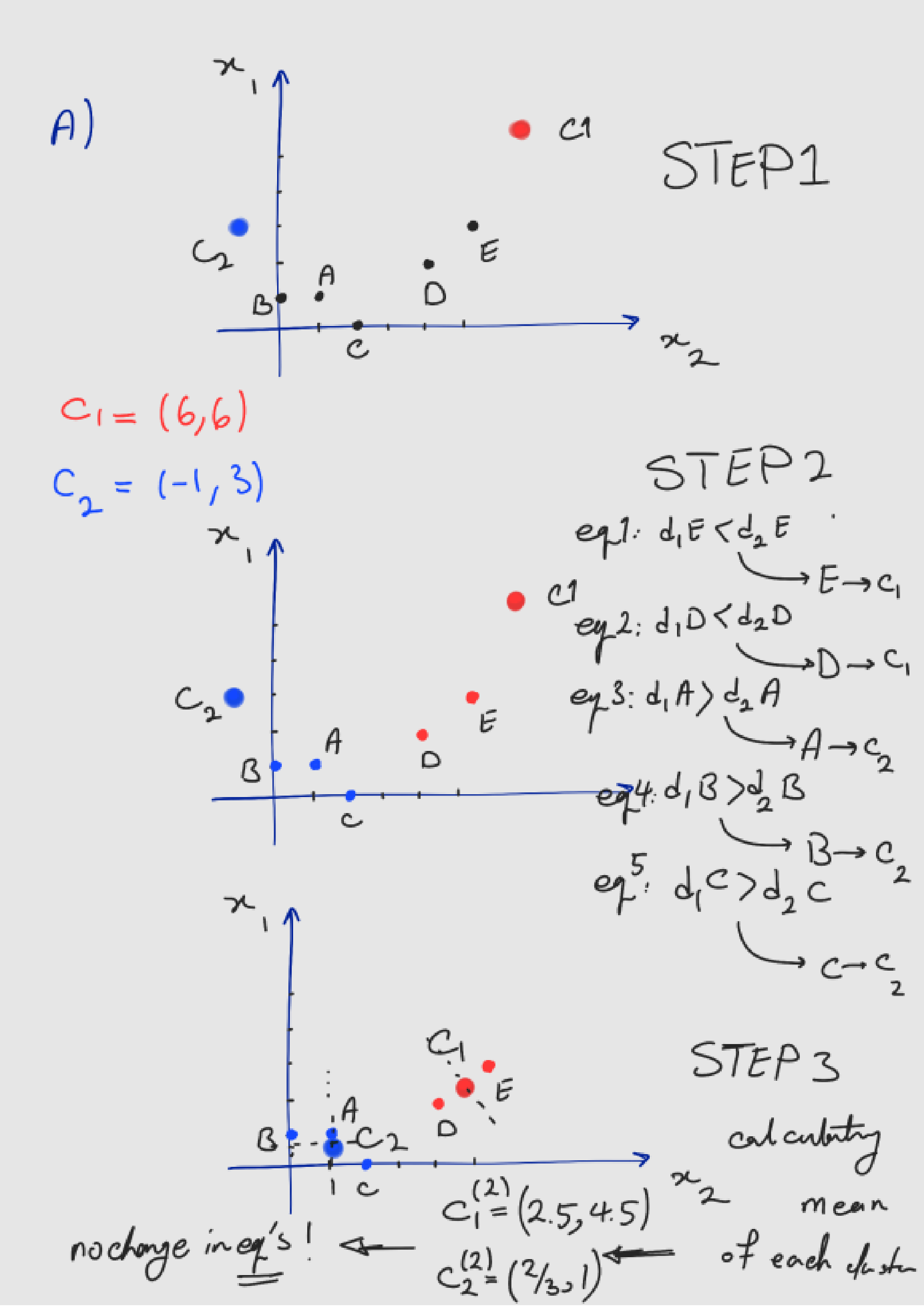
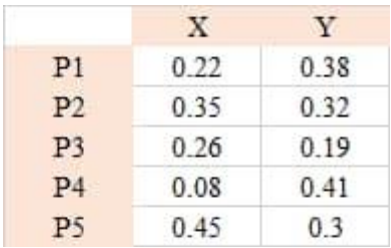


FIGURE1

B)

In this part we want to use hierarchical clustering in order to cluster our data step by step.

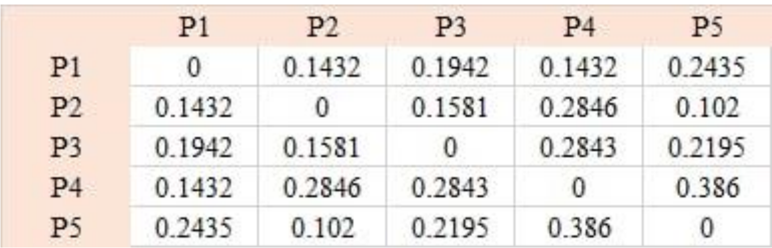
First an overview of our data is shown in Figure2.



	X	Y
P1	0.22	0.38
P2	0.35	0.32
P3	0.26	0.19
P4	0.08	0.41
P5	0.45	0.3

FIGURE2. OVERVIEW OF OUR DATA

In this algorithm, we should calculate the Euclidean distance of each point to all remaining points and then chose the most similar (the lowest distance) to combine that two specific points. Figure3 shows the Euclidean distance between each point.



	P1	P2	P3	P4	P5
P1	0	0.1432	0.1942	0.1432	0.2435
P2	0.1432	0	0.1581	0.2846	0.102
P3	0.1942	0.1581	0	0.2843	0.2195
P4	0.1432	0.2846	0.2843	0	0.386
P5	0.2435	0.102	0.2195	0.386	0

FIGURE3. ONE BY ONE EUCLIDEAN DISTANCE

Before using the algorithm, we could manually cluster the data by short-link. It is shown in Figure 4.

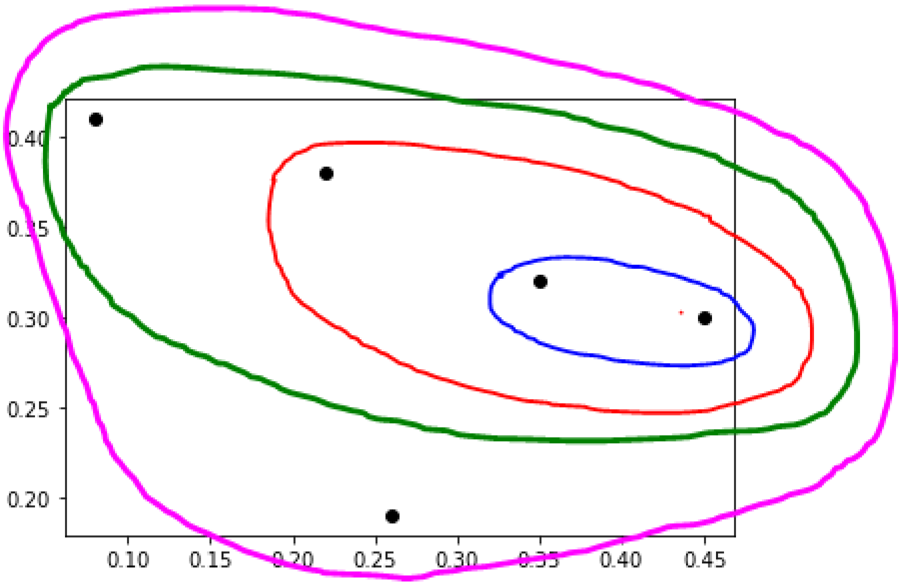


FIGURE4. JUST A GLANCE AT DATA

As you see P2 and P5 have the most similarity, so these two points will be combined to one cluster and actually a new point(P6)!

Now we should compare the distance between P1, P3, P4 and our new point P6 which is the combination of P2 and P5.

Because of using the short-link method to define the next similar one, the next is P1 to join P6 which is actually two points and it is because of the most similarity of P1 to P6 and it would create P7 (recall that for similarity between P6 and P1 we use the nearest one between P2 and P5 in P6). In the next step, P4 will join P7 because of its high similarity with P1 which was in P7 before. At the we can join P3 to P7 and our root in Dendrogram will be created.

1) P2 + P5 = P6
2) P1 + P6 = P7
3) P7 + P4 = P8
4) P8 + P3 = root

The above scenario is represented in Figure4 by a Dendrogram:

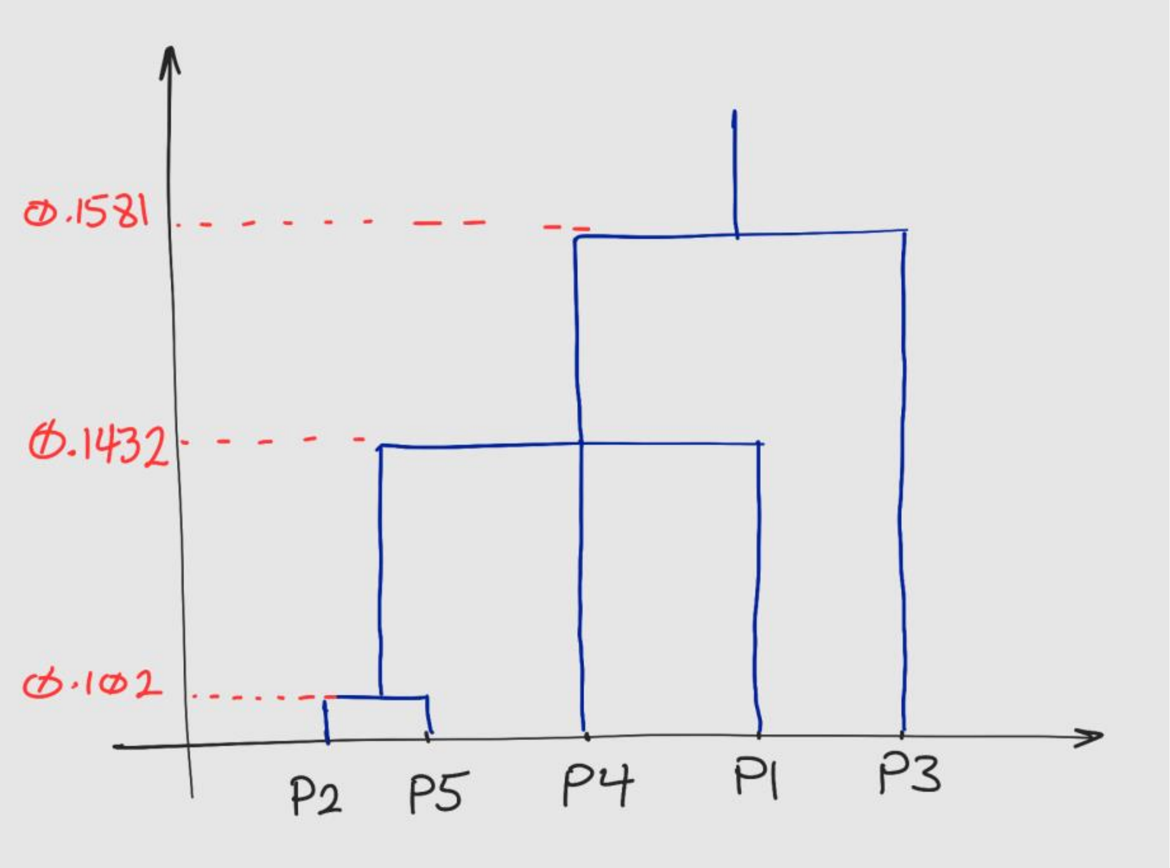
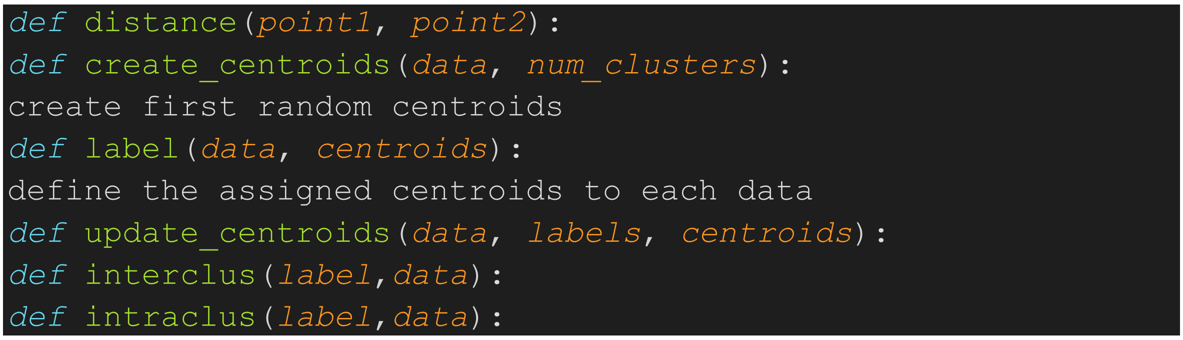


FIGURE 5. DENDROGRAM FOR THIS PROBLEM

Question 2

A)

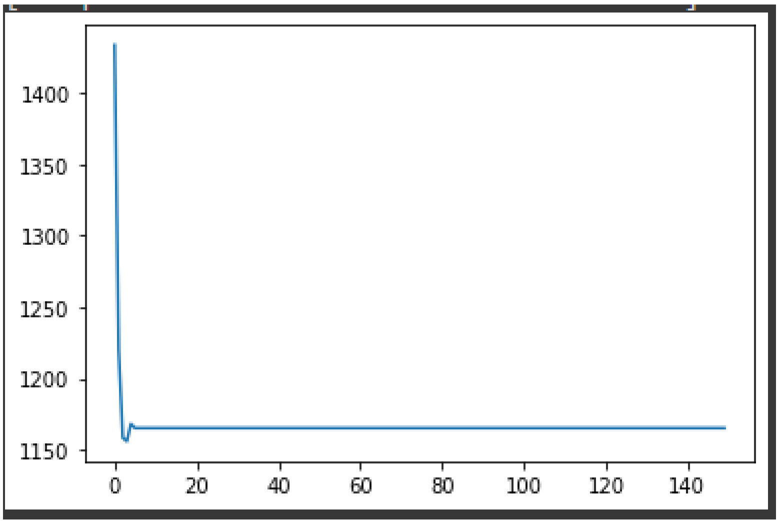
In this part we want to implement K-means algorithm on our iris data. The functions I use is as below:

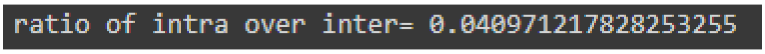


```
def distance(point1, point2):
def create_centroids(data, num_clusters):
create first random centroids
def label(data, centroids):
define the assigned centroids to each data
def update_centroids(data, labels, centroids):
def interclus(label,data):
def intraclus(label,data):
```

As you can see in figures 6 to 9, the cost function over iteration is shown. Also, the ratio of intra-cluster distances over inter-cluster distances is reported.

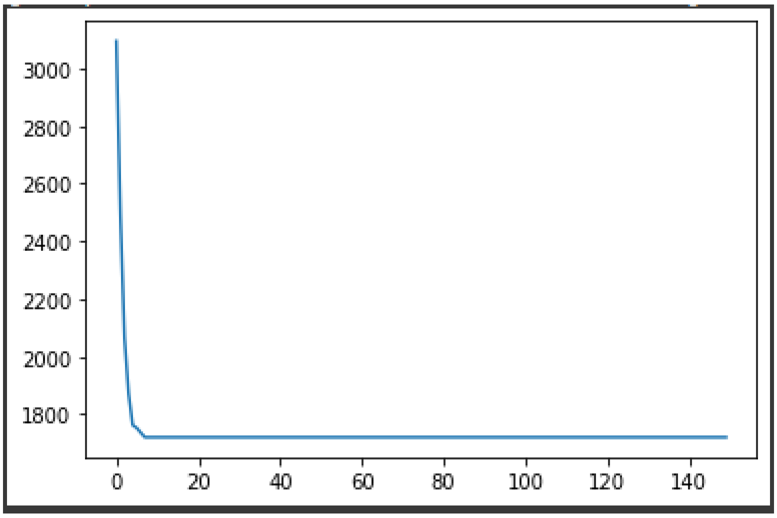
Cost function in my algorithm is intra-cluster distances which should be minimized as our optimization aim!

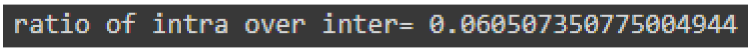




```
ratio of intra over inter= 0.040971217828253255
```

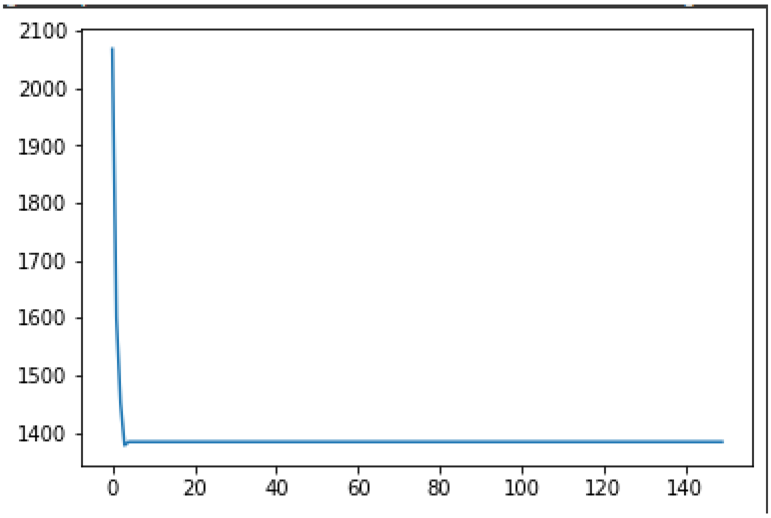
FIGURE 6. COST FUNCTION OVER ITERATION FOR K=10

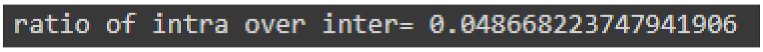




```
ratio of intra over inter= 0.0605073507750049444
```

FIGURE 7. COST FUNCTION OVER ITERATION FOR K=5





```
ratio of intra over inter= 0.048668223747941906
```

FIGURE 8. COST FUNCTION OVER ITERATION FOR K=20

As it is reported in ratio of intra distance over inter distance, it can be concluded that K=10 is the best choice in comparison to K=20 and K=5.

If we make the iterations lower then we can see the elbow pattern as it is shown in figure9.

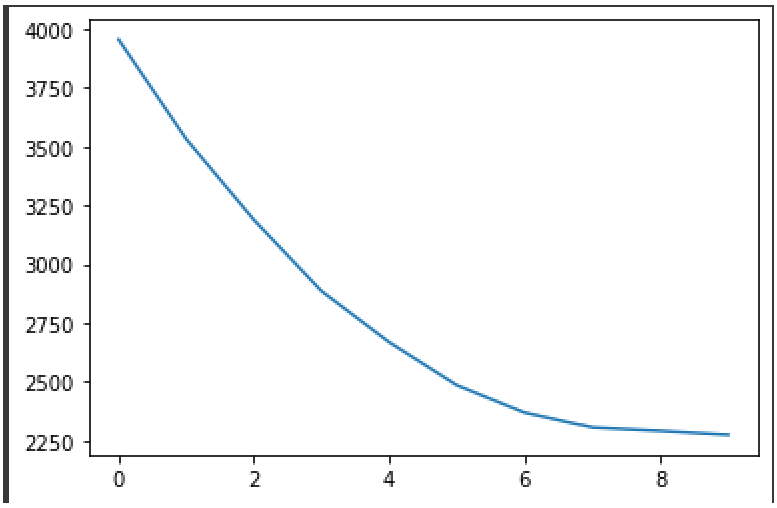
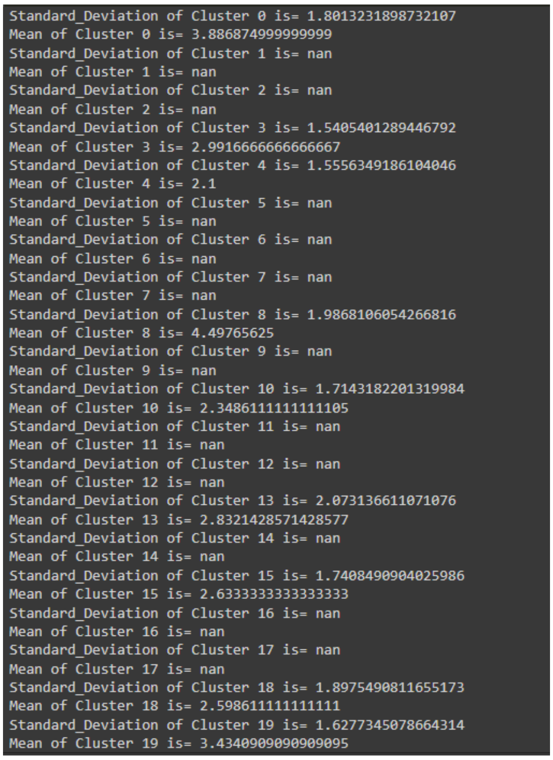


FIGURE 9. COST FUNCTION OVER ITERATION (ELBOW PATTERN)

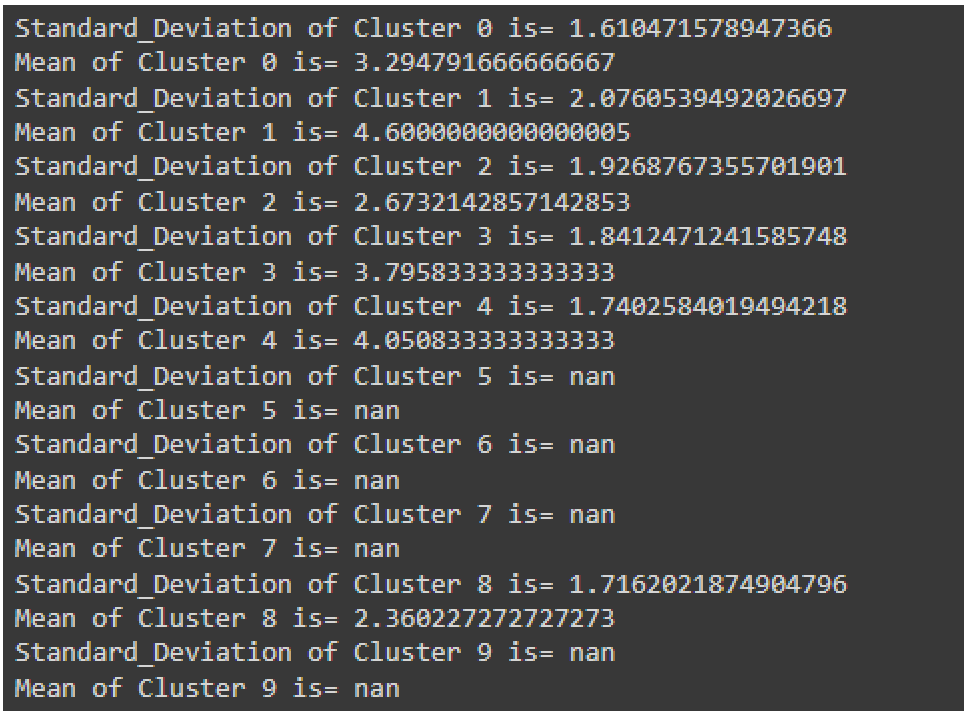
B)

The mean and standard deviation is calculated as it is obligated in the question.



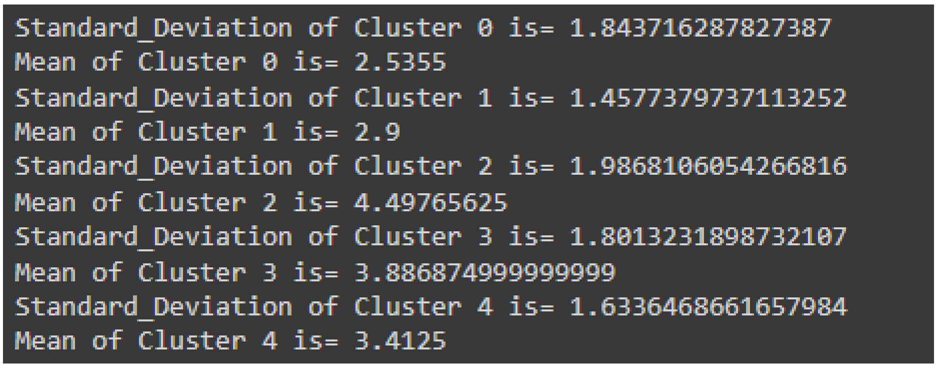
```
Standard_Deviation of Cluster 0 is= 1.8013231898732107
Mean of Cluster 0 is= 3.886874999999999
Standard_Deviation of Cluster 1 is= nan
Mean of Cluster 1 is= nan
Standard_Deviation of Cluster 2 is= nan
Mean of Cluster 2 is= nan
Standard_Deviation of Cluster 3 is= 1.5405401289446792
Mean of Cluster 3 is= 2.9916666666666667
Standard_Deviation of Cluster 4 is= 1.5556349186104046
Mean of Cluster 4 is= 2.1
Standard_Deviation of Cluster 5 is= nan
Mean of Cluster 5 is= nan
Standard_Deviation of Cluster 6 is= nan
Mean of Cluster 6 is= nan
Standard_Deviation of Cluster 7 is= nan
Mean of Cluster 7 is= nan
Standard_Deviation of Cluster 8 is= 1.9868106054266816
Mean of Cluster 8 is= 4.49765625
Standard_Deviation of Cluster 9 is= nan
Mean of Cluster 9 is= nan
Standard_Deviation of Cluster 10 is= 1.7143182201319984
Mean of Cluster 10 is= 2.3486111111111105
Standard_Deviation of Cluster 11 is= nan
Mean of Cluster 11 is= nan
Standard_Deviation of Cluster 12 is= nan
Mean of Cluster 12 is= nan
Standard_Deviation of Cluster 13 is= 2.073136611071076
Mean of Cluster 13 is= 2.8321428571428577
Standard_Deviation of Cluster 14 is= nan
Mean of Cluster 14 is= nan
Standard_Deviation of Cluster 15 is= 1.7408490904025986
Mean of Cluster 15 is= 2.6333333333333333
Standard_Deviation of Cluster 16 is= nan
Mean of Cluster 16 is= nan
Standard_Deviation of Cluster 17 is= nan
Mean of Cluster 17 is= nan
Standard_Deviation of Cluster 18 is= 1.8975490811655173
Mean of Cluster 18 is= 2.598611111111111
Standard_Deviation of Cluster 19 is= 1.6277345078664314
Mean of Cluster 19 is= 3.4340909090909095
```

FIGURE 10. MEAN AND STANDARD DEVIATION OF DATA FOR K=20



```
Standard_Deviation of Cluster 0 is= 1.610471578947366
Mean of Cluster 0 is= 3.294791666666667
Standard_Deviation of Cluster 1 is= 2.0760539492026697
Mean of Cluster 1 is= 4.6000000000000005
Standard_Deviation of Cluster 2 is= 1.9268767355701901
Mean of Cluster 2 is= 2.6732142857142853
Standard_Deviation of Cluster 3 is= 1.8412471241585748
Mean of Cluster 3 is= 3.795833333333333
Standard_Deviation of Cluster 4 is= 1.7402584019494218
Mean of Cluster 4 is= 4.050833333333333
Standard_Deviation of Cluster 5 is= nan
Mean of Cluster 5 is= nan
Standard_Deviation of Cluster 6 is= nan
Mean of Cluster 6 is= nan
Standard_Deviation of Cluster 7 is= nan
Mean of Cluster 7 is= nan
Standard_Deviation of Cluster 8 is= 1.7162021874904796
Mean of Cluster 8 is= 2.360227272727273
Standard_Deviation of Cluster 9 is= nan
Mean of Cluster 9 is= nan
```

FIGURE 11. MEAN AND STANDARD DEVIATION OF DATA FOR K=20



```
Standard_Deviation of Cluster 0 is= 1.843716287827387
Mean of Cluster 0 is= 2.5355
Standard_Deviation of Cluster 1 is= 1.4577379737113252
Mean of Cluster 1 is= 2.9
Standard_Deviation of Cluster 2 is= 1.9868106054266816
Mean of Cluster 2 is= 4.49765625
Standard_Deviation of Cluster 3 is= 1.8013231898732107
Mean of Cluster 3 is= 3.886874999999999
Standard_Deviation of Cluster 4 is= 1.6336468661657984
Mean of Cluster 4 is= 3.4125
```

FIGURE 11. MEAN AND STANDARD DEVIATION OF DATA FOR K=20

Question 3

A)

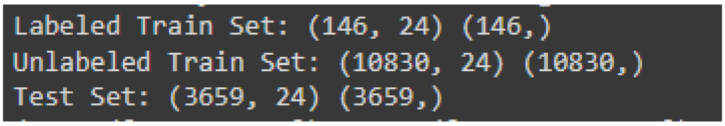
Logistic Regression is used when the dependent variable is categorical.

For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

As it is shown below data is splitted in 1, 74, 25 for label, unlabel and test data.



```
Labeled Train Set: (146, 24) (146,)
Unlabeled Train Set: (10830, 24) (10830,)
Test Set: (3659, 24) (3659,)
```

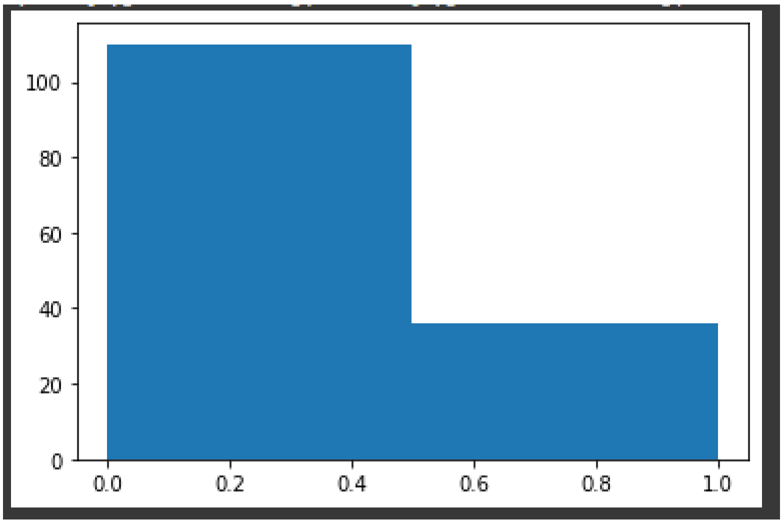
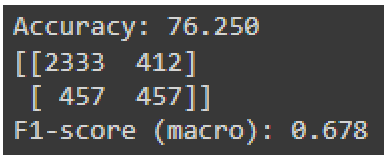


Figure 12. Data Distribution on Target feature

As it is shown above, Data is imbalance cause more than 2 times higher is 0 complication than 1 complication. So, we should know that high accuracy in our classifier is more because of data distribution than our good work!

B)

F1-score and accuracy and confusion matrix of our regressor is as below:



```
Accuracy: 76.250
[[2333  412]
 [ 457  457]]
F1-score (macro): 0.678
```

Figure 13. F1-score and accuracy and confusion matrix

C)

Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data). It is a special instance of weak supervision.

Some methods for semi-supervised learning are not geared to learning from both unlabeled and labeled data, but instead make use of unlabeled data within a supervised learning framework.

Self-training is a wrapper method for semi-supervised learning. First a supervised learning algorithm is trained based on the labeled data only. This classifier is then applied to the unlabeled data to generate more labeled examples as input for the supervised learning algorithm. Generally, only the labels the classifier is most confident in are added at each step.

Brief of the algorithm I use is as below:

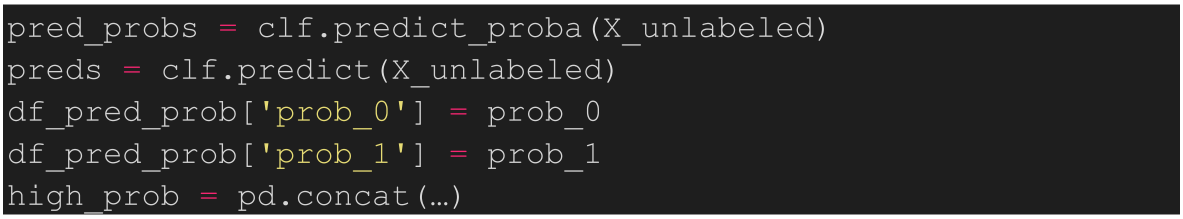
Step 1: First, train a Logistic Regression classifier on the labeled training data.

Step 2: Next, use the classifier to predict labels for all unlabeled data, as well as probabilities for those predictions. In this case, I will only adopt 'pseudo-labels' for predictions with greater than 99% probability.

Step 3: Concatenate the 'pseudo-labeled' data with the labeled training data, and re-train the classifier on the concatenated data.

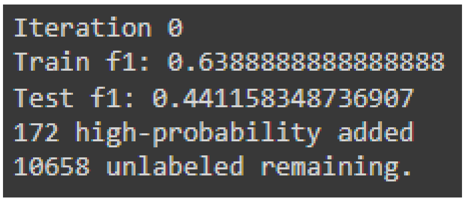
Step 4: Use trained classifier to make predictions for the labeled test data, and evaluate the classifier.

Repeat steps 1 through 4 until no more predictions have greater than 99% probability, or no unlabeled data remains.

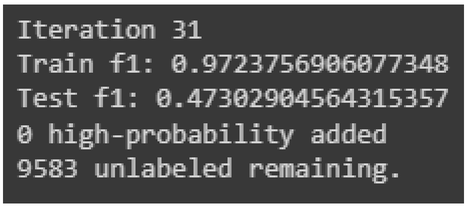


```
pred_probs = clf.predict_proba(X_unlabeled)
preds = clf.predict(X_unlabeled)
df_pred_prob['prob_0'] = prob_0
df_pred_prob['prob_1'] = prob_1
high_prob = pd.concat(…)
```

In figure 14, it is shown that how much iteration there was needed and the comparison between first and last train test F1-score.



```
Iteration 0
Train f1: 0.6388888888888888
Test f1: 0.441158348736907
172 high-probability added
10658 unlabeled remaining.
```



```
Iteration 31
Train f1: 0.9723756906077348
Test f1: 0.47302904564315357
0 high-probability added
9583 unlabeled remaining.
```

FIGURE 14. COMPARISON BETWEEN FIRST AND LAST TRAIN TEST F1-SCORE

It is obvious based on Figure14 that semi-supervised learning helped us through getting higher F1-score both on train and test data.

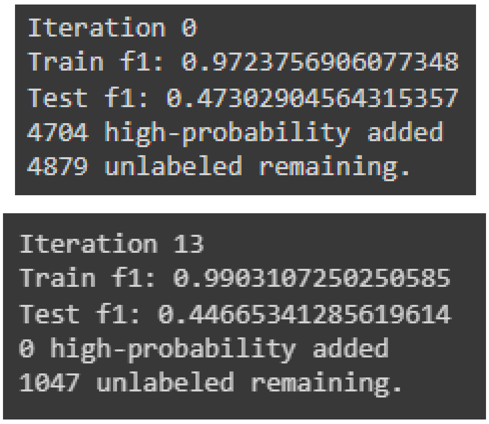
Maybe on test data it is just a bit improvement but it is also great with unlabeled data!!

D)

Our criteria are two things in this algorithm:

1) No more predictions have greater than 99% probability
2) No unlabeled data remains

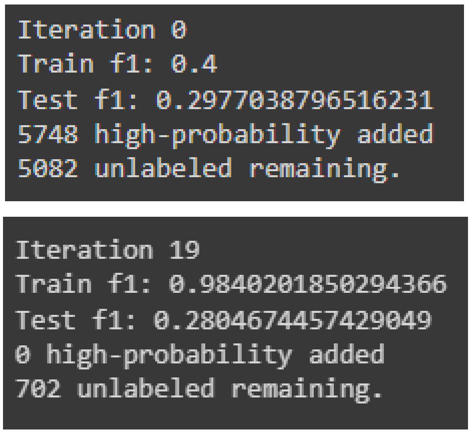
I have changed the limit probability to 89 and 79. The results is as below:



```
Iteration 0
Train f1: 0.9723756906077348
Test f1: 0.47302904564315357
4704 high-probability added
4879 unlabeled remaining.
```

```
Iteration 13
Train f1: 0.9903107250250585
Test f1: 0.44665341285619614
0 high-probability added
1047 unlabeled remaining.
```

FIGURE 15. COMPARISON BETWEEN FIRST AND LAST TRAIN TEST F1-SCORE, P = 89%



```
Iteration 0
Train f1: 0.4
Test f1: 0.2977038796516231
5748 high-probability added
5082 unlabeled remaining.
```

```
Iteration 19
Train f1: 0.9840201850294366
Test f1: 0.2804674457429049
0 high-probability added
702 unlabeled remaining.
```

Figure 16. Comparison between first and last train test F1-score, P = 79%

It shows that P = 99% is a better choice for limitation!

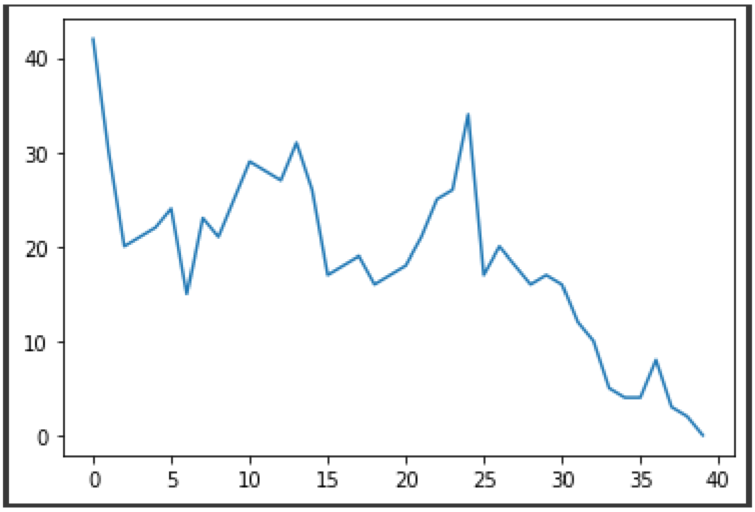


Figure 17. Number of Pseudo-Labels over Iteration

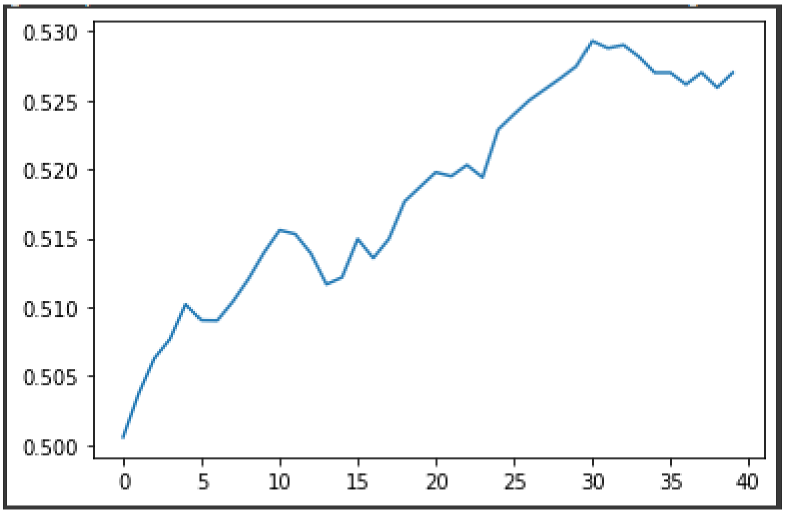


Figure 18. Test F1-score over Iteration

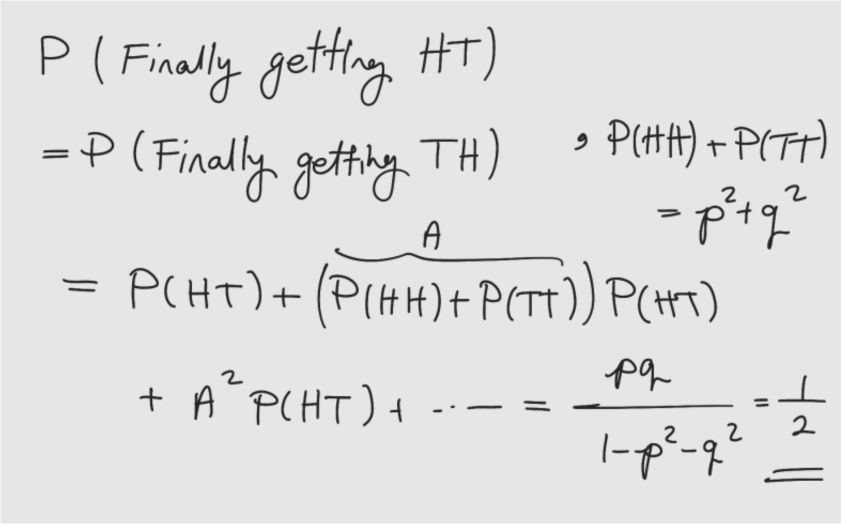
Question 4

A-1)

In this part we want to design an algorithm using an unfair coin which give us a uniform density to choose one person from 2 persons. Briefly, we want to generate p = ½ by using an unfair coin. Von Neumann’s Solution helps us through this way! In figure19, it is shown how the final p is ½.

To generate such a random variable, there is 3 steps:

1. Toss the coin twice.
2. If the result is HT, assign X=0. If the result is TH, assign X=1.
3. If the result is either HH or TT, then discard the previous two tosses and go to step1.



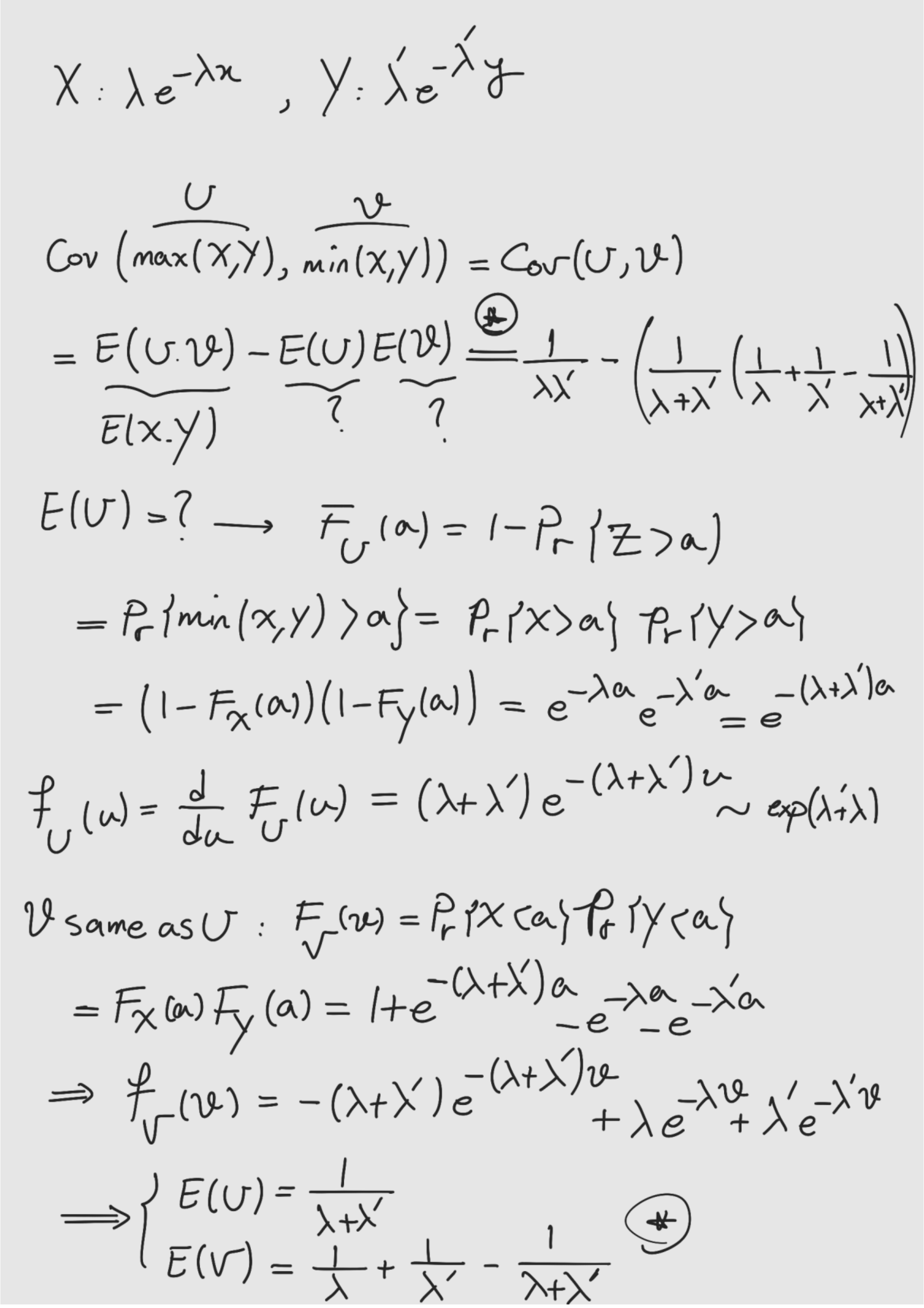
$$P(\text{Finally getting } HT) = P(\text{Finally getting } TH) \quad , \; P(HH)+P(TT) = p^2+q^2$$

$$= P(HT) + \overbrace{\big(P(HH)+P(TT)\big)}^{A} P(HT) + A^2 P(HT) + \cdots = \frac{pq}{1-p^2-q^2} = \frac{1}{2}$$

FIGURE19. PROOF OF VON NEUMANN’S METHOD

A-2)

1)



$X : \lambda e^{-\lambda x} \quad , \quad Y : \lambda' e^{-\lambda' y}$

$$Cov\left(\overbrace{max(X,Y)}^{U}, \overbrace{min(X,Y)}^{v}\right) = Cov(U,v)$$

$$= \underbrace{E(U.v)}_{E(X.Y)} - \underbrace{E(U)}_{?}\underbrace{E(v)}_{?} \overset{\circledast}{=} \frac{1}{\lambda\lambda'} - \left(\frac{1}{\lambda+\lambda'}\left(\frac{1}{\lambda}+\frac{1}{\lambda'}-\frac{1}{\lambda+\lambda'}\right)\right)$$

$$E(U) = ? \longrightarrow F_U(a) = 1 - P_r\{Z > a)$$

$$= P_r\{min(x,y) > a\} = P_r\{X > a\}\, P_r\{Y > a\}$$

$$= (1 - F_X(a))(1 - F_Y(a)) = e^{-\lambda a} e^{-\lambda' a} = e^{-(\lambda+\lambda')a}$$

$$f_U(u) = \frac{d}{du} F_U(u) = (\lambda+\lambda')e^{-(\lambda+\lambda')u} \sim exp(\lambda'+\lambda)$$

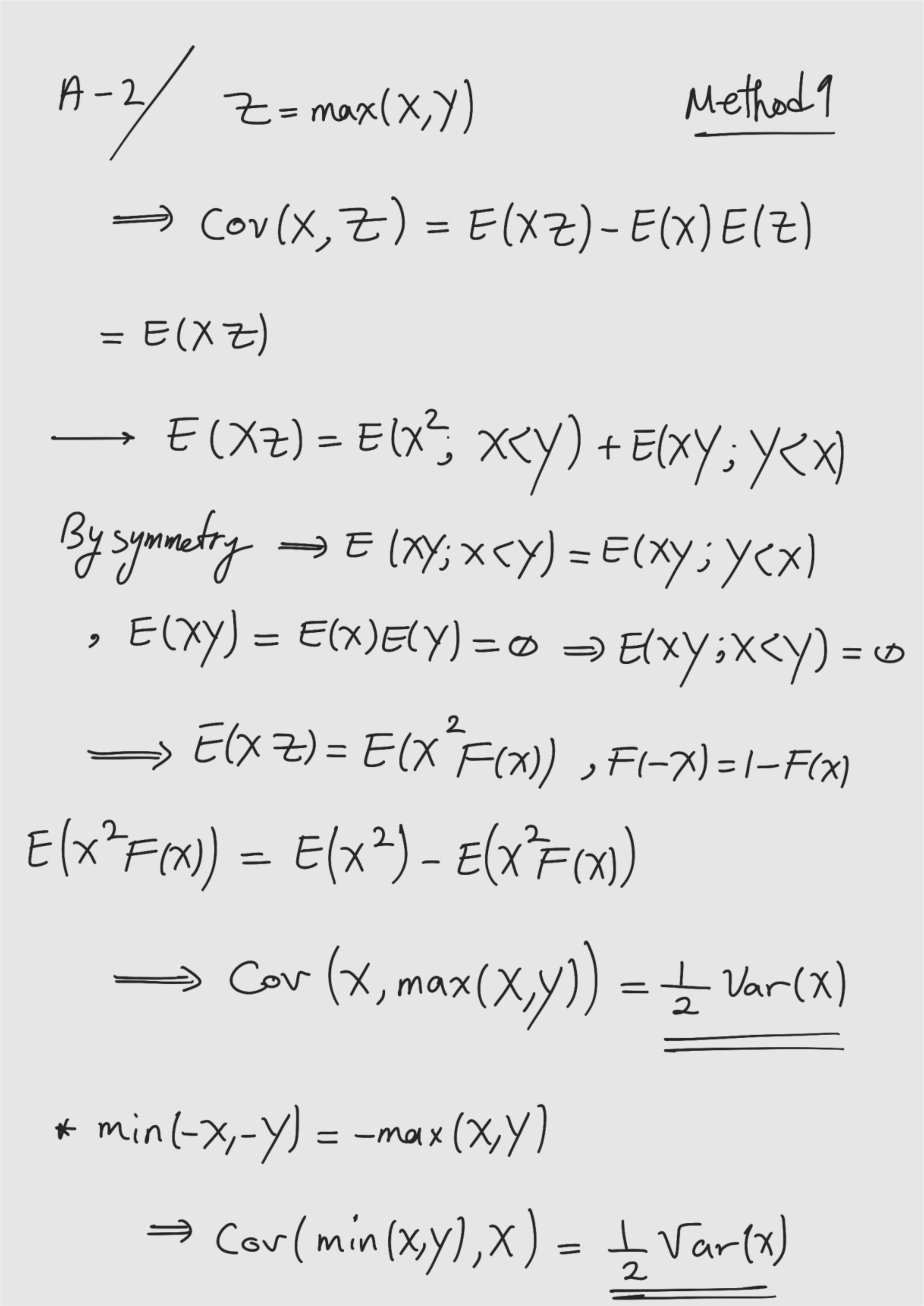
v same as U : $F_V(v) = P_r\{X < a\}\, P_r\{Y < a\}$

$$= F_X(a) F_Y(a) = 1 + e^{-(\lambda+\lambda')a} - e^{-\lambda a} - e^{-\lambda' a}$$

$$\Rightarrow f_V(v) = -(\lambda+\lambda')e^{-(\lambda+\lambda')v} + \lambda e^{-\lambda v} + \lambda' e^{-\lambda' v}$$

$$\Longrightarrow \begin{cases} E(U) = \dfrac{1}{\lambda+\lambda'} \\ E(V) = \dfrac{1}{\lambda} + \dfrac{1}{\lambda'} - \dfrac{1}{\lambda+\lambda'} \end{cases} \circledast$$

2)



A-2/ $Z = \max(X,Y)$ Method 1

$$\Longrightarrow \mathrm{Cov}(X,Z) = E(XZ) - E(X)E(Z)$$

$$= E(XZ)$$

$$\longrightarrow E(XZ) = E(X^2; X<Y) + E(XY; Y<X)$$

By symmetry $\Longrightarrow E(XY; X<Y) = E(XY; Y<X)$

$, E(XY) = E(X)E(Y) = 0 \Rightarrow E(XY; X<Y) = 0$

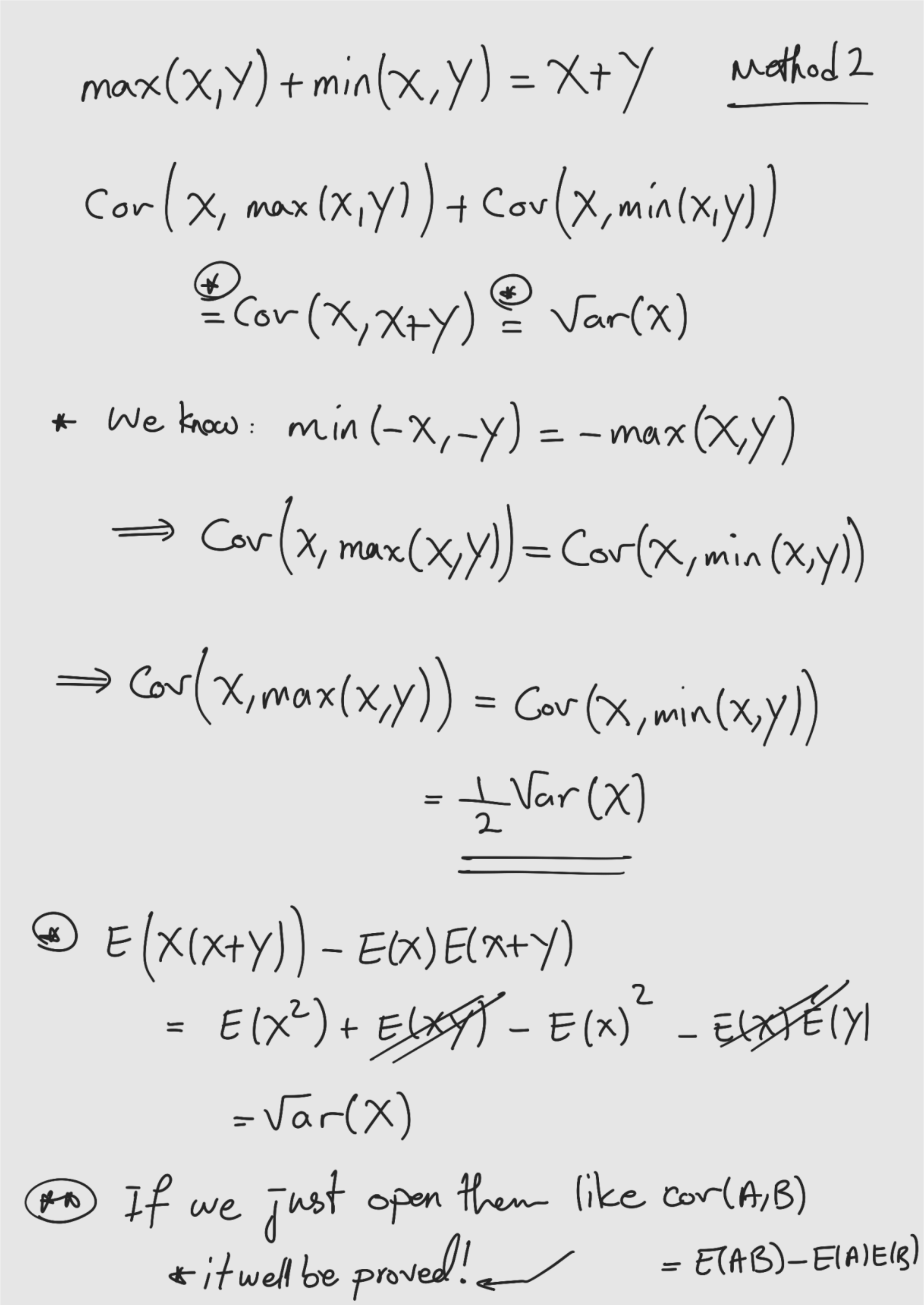
$$\Longrightarrow E(XZ) = E(X^2F(X)) \quad , F(-X) = 1 - F(X)$$

$$E(X^2F(X)) = E(X^2) - E(X^2F(X))$$

$$\Longrightarrow \mathrm{Cov}(X, \max(X,Y)) = \underline{\underline{\tfrac{1}{2}\mathrm{Var}(X)}}$$

$* \min(-X,-Y) = -\max(X,Y)$

$$\Rightarrow \mathrm{Cov}(\min(X,Y), X) = \underline{\underline{\tfrac{1}{2}\mathrm{Var}(X)}}$$



$\max(X,Y) + \min(X,Y) = X+Y$ Method 2

$\text{Cor}\big(X, \max(X,Y)\big) + \text{Cov}\big(X, \min(X,Y)\big)$

$\overset{\circledast}{=} \text{Cov}(X, X+Y) \overset{\circledast}{=} \text{Var}(X)$

* We know: $\min(-X,-Y) = -\max(X,Y)$

$\Longrightarrow \text{Cov}\big(X, \max(X,Y)\big) = \text{Cov}\big(X, \min(X,Y)\big)$

$\Longrightarrow \text{Cov}\big(X, \max(x,y)\big) = \text{Cov}\big(X, \min(x,y)\big)$

$= \underline{\underline{\frac{1}{2}\text{Var}(X)}}$

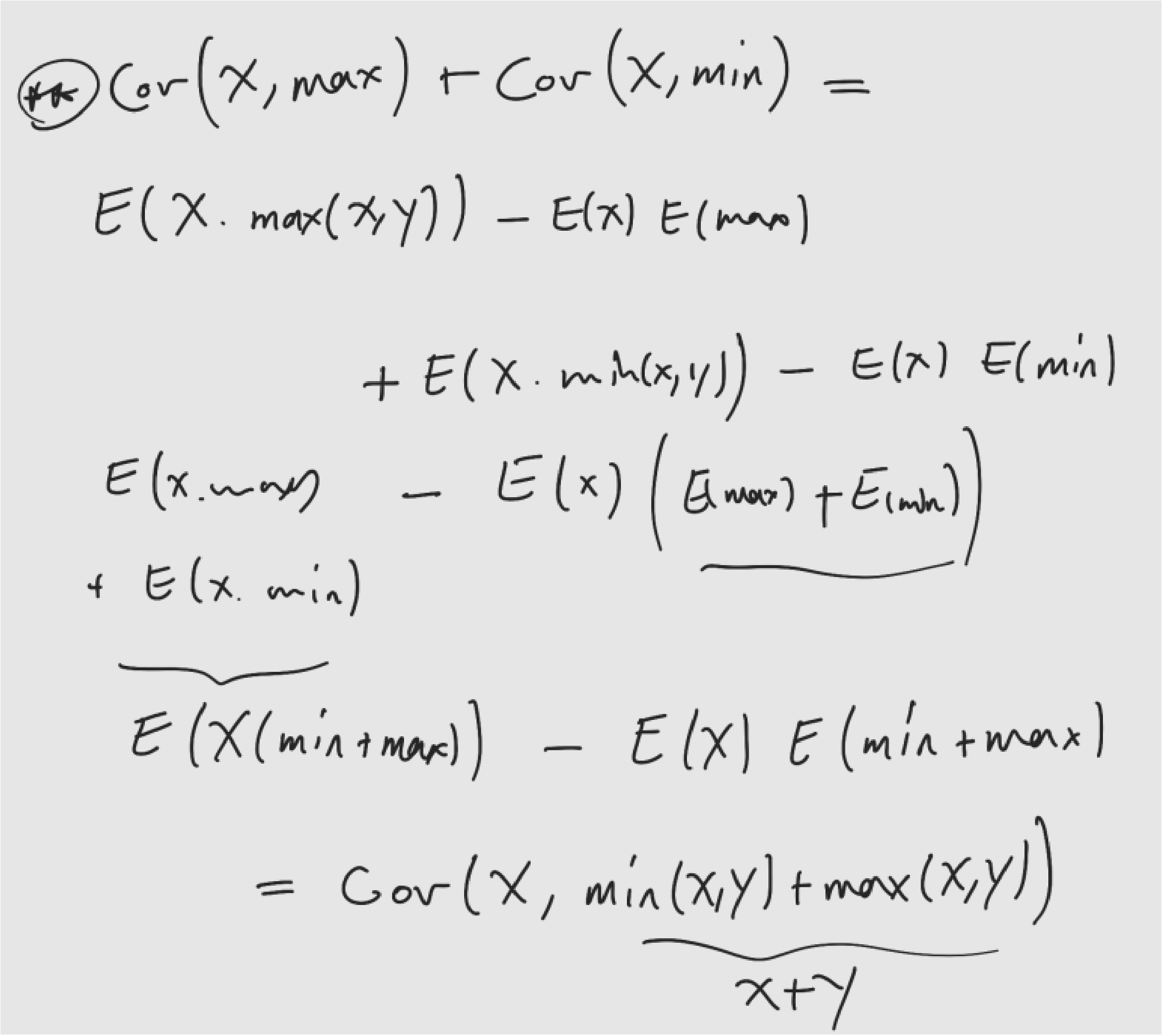
$\circledast$ $E\big(X(X+Y)\big) - E(X)E(X+Y)$

$= E(X^2) + \cancel{E(XY)} - E(x)^2 - \cancel{E(X)E(Y)}$

$= \text{Var}(X)$

$\circledast\circledast$ If we just open them like $\text{cor}(A,B) = E(AB) - E(A)E(B)$

* it well be proved! ←



$$(\ast\ast)\ \mathrm{Cov}(X, \max) + \mathrm{Cov}(X, \min) =$$

$$E(X \cdot \max(X,Y)) - E(X)\,E(\max)$$

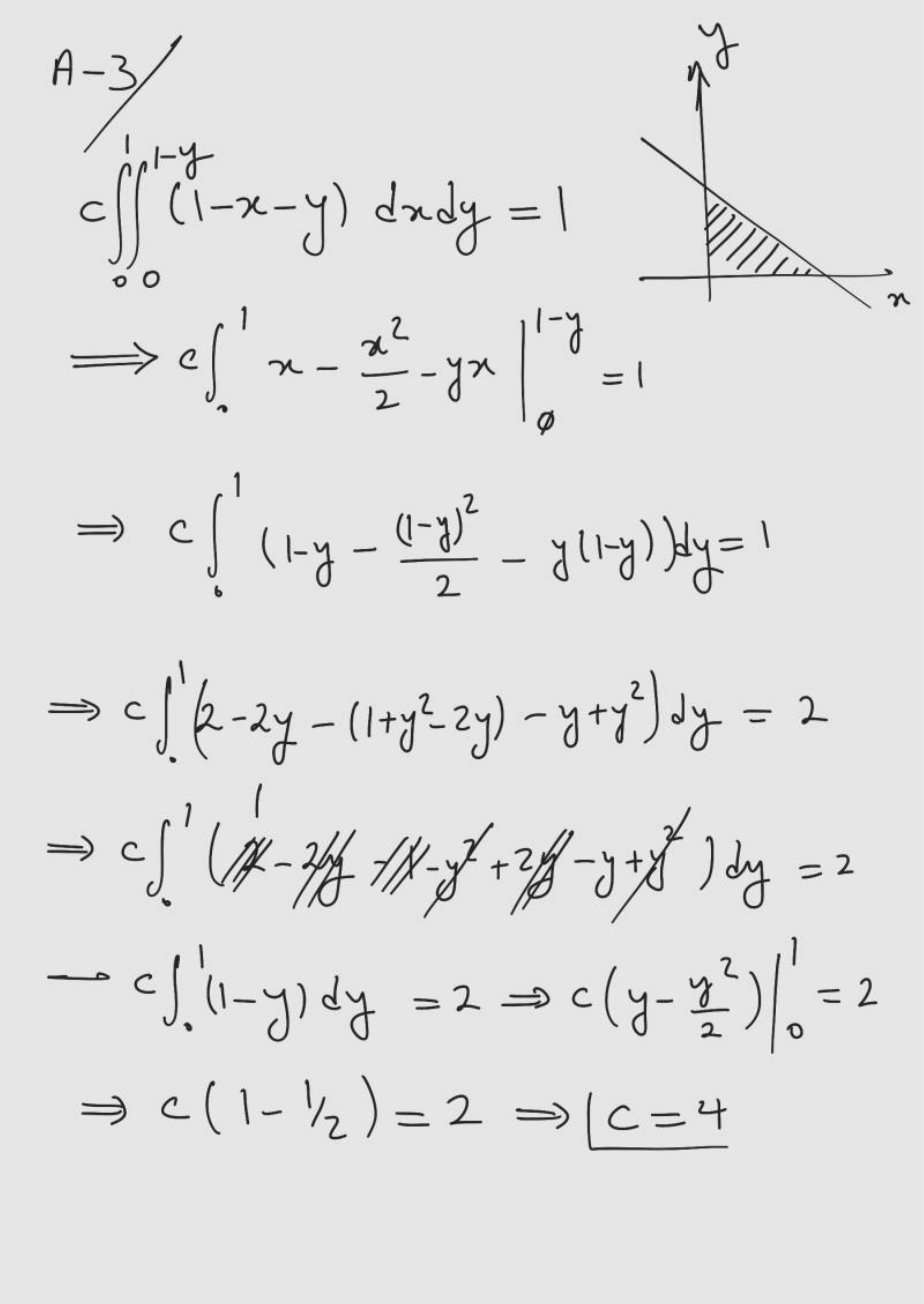
$$+ E(X \cdot \min(X,Y)) - E(X)\,E(\min)$$

$$\underbrace{E(X.\max) + E(X.\min)} - E(X)\left(\underline{E(\max) + E(\min)}\right)$$

$$E(X(\min + \max)) - E(X)\,E(\min + \max)$$

$$= \mathrm{Cov}(X, \underbrace{\min(X,Y) + \max(X,Y)}_{X+Y})$$

A-3)



A-3/

$$c\int_0^1\int_0^{1-y}(1-x-y)\,dx\,dy = 1$$

$$\Longrightarrow c\int_0^1 x - \frac{x^2}{2} - yx \Big|_{\emptyset}^{1-y} = 1$$

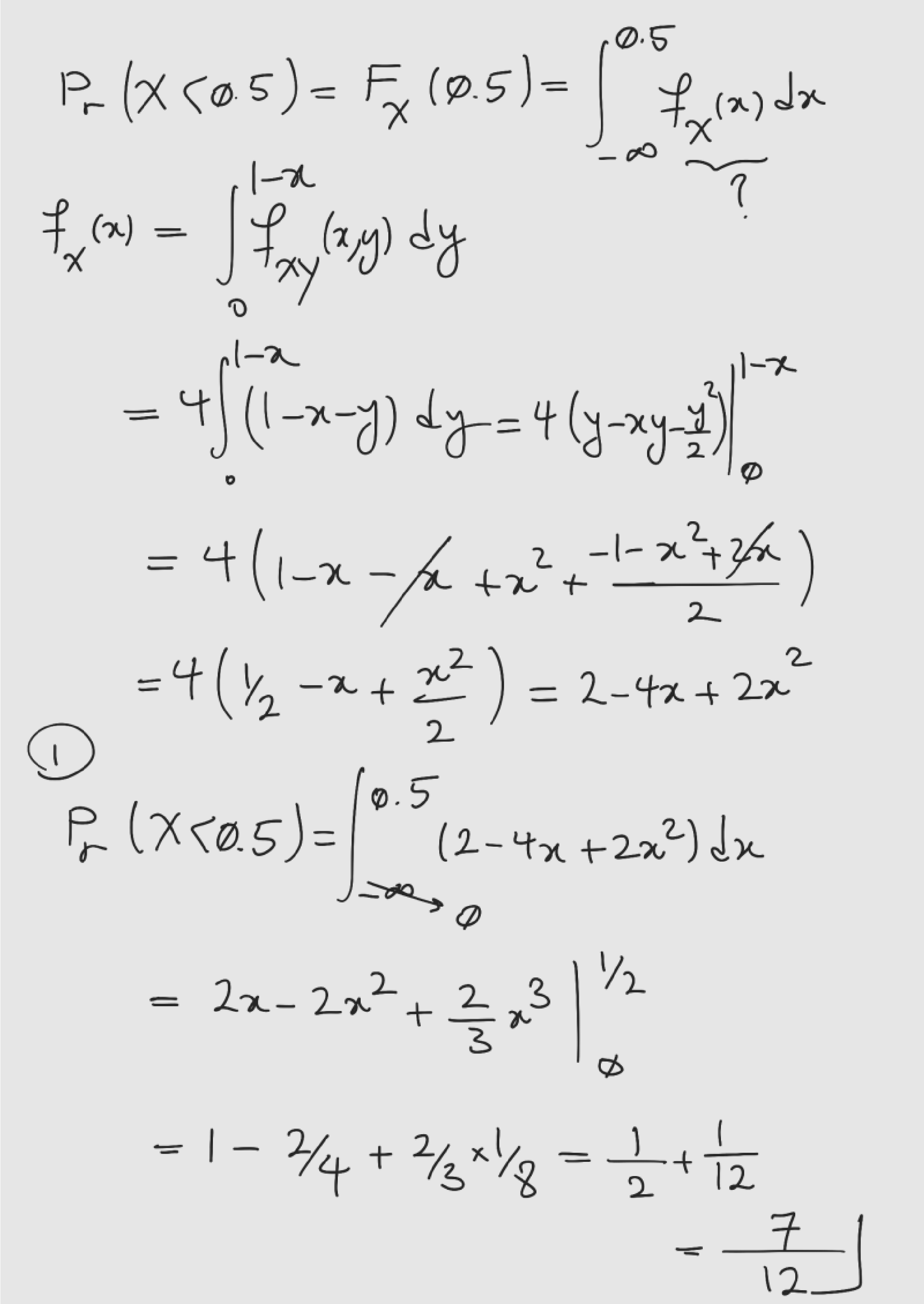
$$\Rightarrow c\int_0^1 \left(1-y - \frac{(1-y)^2}{2} - y(1-y)\right)dy = 1$$

$$\Rightarrow c\int_0^1 \left(2-2y-(1+y^2-2y)-y+y^2\right)dy = 2$$

$$\Rightarrow c\int_0^1 \left(\overset{1}{\cancel{2}} - \cancel{2y} \cancel{-1} - \cancel{y^2} + \cancel{2y} - y + \cancel{y^2}\right)dy = 2$$

$$\longrightarrow c\int_0^1 (1-y)\,dy = 2 \Rightarrow c\left(y-\frac{y^2}{2}\right)\Big|_0^1 = 2$$

$$\Rightarrow c\left(1-\frac{1}{2}\right) = 2 \Rightarrow \boxed{c=4}$$



$$P_r(X<0.5)=F_X(0.5)=\int_{-\infty}^{0.5}\underbrace{f_X(x)}_{?}\,dx$$

$$f_X(x)=\int_0^{1-x}f_{xy}(x,y)\,dy$$

$$=4\int_0^{1-x}(1-x-y)\,dy=4\left(y-xy-\frac{y^2}{2}\right)\Big|_0^{1-x}$$

$$=4\left(1-x-\cancel{x}+x^2+\frac{-1-x^2+\cancel{2x}}{2}\right)$$

$$=4\left(\tfrac{1}{2}-x+\frac{x^2}{2}\right)=2-4x+2x^2$$

①

$$P_r(X<0.5)=\int_{\cancel{-\infty}\rightarrow 0}^{0.5}(2-4x+2x^2)\,dx$$

$$=2x-2x^2+\frac{2}{3}x^3\Big|_0^{1/2}$$

$$=1-\tfrac{2}{4}+\tfrac{2}{3}\times\tfrac{1}{8}=\frac{1}{2}+\frac{1}{12}$$

$$=\frac{7}{12}$$

② $E(X+Y) = E(X)+E(Y)$, $f_X(x) \equiv f_Y(y)$)

because of symmetry!

$$E(X) = \int_{-\infty}^{+\infty} x f_X(x)\, dx$$

$$= \int_0^1 x(2-4x+2x^2)\,dx$$

$$= \int_0^1 (2x-4x^2+2x^3)\,dx = \left(x^2 - \frac{4}{3}x^3 + \frac{1}{2}x^4\right)\Big|_0^1$$

$$= \left(1-\frac{4}{3}+\frac{1}{2}\right) = 3/2 - 4/3 = \frac{1}{6}$$

$\Longrightarrow E(Y) = 1/6$ because of symmetry in every thing

$\Rightarrow E(X+Y) = 1/3$

The first value, shows the probability that first coder dedicates less than 0.5 of hour (or any unit) for company.

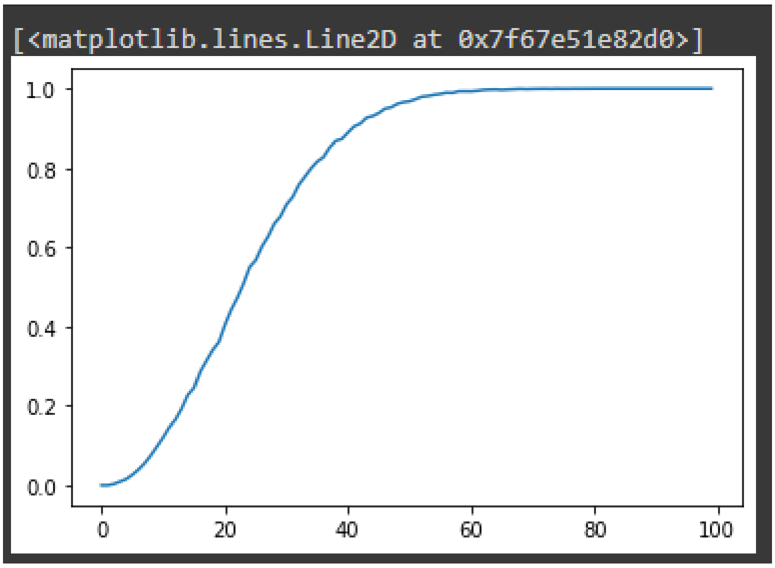
The second value shows that how much time these two guys would spend in average for the company.

B-1)

The functions we use to implement the Birthday problem simulation is as below:

```
def generate_random_birthday():
Every time it is called, it would generate a random time
def generate_k_birthdays(k):
If we call this function the first function would be called
“k” times!
def aloc(birthdays):
Tell us how many of that “k” birthdays have coincidence
def estimate_p_aloc(NUM_PEOPLE):
Calculates the probability of same birthdays in our society
```

It is shown in Figure20 that as we increase the number of our society, the probability of coincidence in birthdays will increase and for n>60, the probability approximately is one.



```
[<matplotlib.lines.Line2D at 0x7f67e51e82d0>]
```

FIGURE20. PROBABILITY OF BIRTHDAY COINCIDENCE FOR N=1 TO 100

B-2)

The central limit theorem states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed. This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually n > 30).

For the random samples we take from the population, we can compute the mean of the sample means:

$$\mu_{\bar{X}} = \mu$$

and the standard deviation of the sample means:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

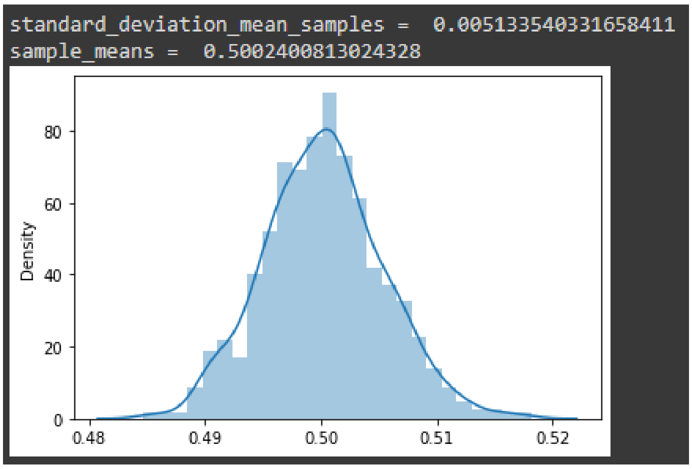
First, we want to check CLT on exponential random variables. So, we decide sample size (n) equals to 10000 and iteration (s) equals to 1000.

EXP Random Variable:

$$\text{MEAN} = \frac{1}{\lambda}$$

$$\text{STD} = \frac{1}{\lambda}$$

Figure21 proved CLT and shows that mean of sample means are same as beginning but standard deviation should be divided by 100 because the root of n is 100.



```
standard_deviation_mean_samples =  0.005133540331658411
sample_means =  0.5002400813024328
```

Figure21. Proving CLT and showing changes in MEAN and STD

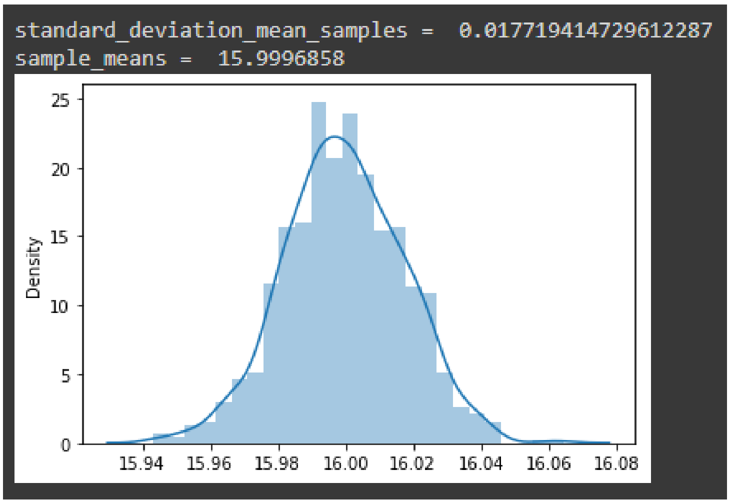
Second, we want to check CLT on binomial random variables. So, we decide sample size (n) equals to 10000 and iteration (s) equals to 1000.

Binomial Random Variable:

$$\text{MEAN} = np$$

$$\text{STD} = \sqrt{np(1-p)}$$

Figure22 proved CLT and shows that mean of sample means are same as beginning but standard deviation should be divided by 100 because the root of n is 100.



```
standard_deviation_mean_samples =  0.017719414729612287
sample_means =  15.9996858
```

FIGURE22. PROVING CLT AND SHOWING CHANGES IN MEAN AND STD

References:

1) https://towardsdatascience.com/a-gentle-introduction-to-self-training-and-semi-supervised-learning-ceee73178b38
2) https://machinelearningmastery.com/semi-supervised-learning-with-label-propagation/
3) Some entities for K-means