**IML -EX1  
Amitay Sicherman -203449004**

[Warm-up - Algebra Recap 2](#_Toc35968831)

[SVD 4](#_Toc35968832)

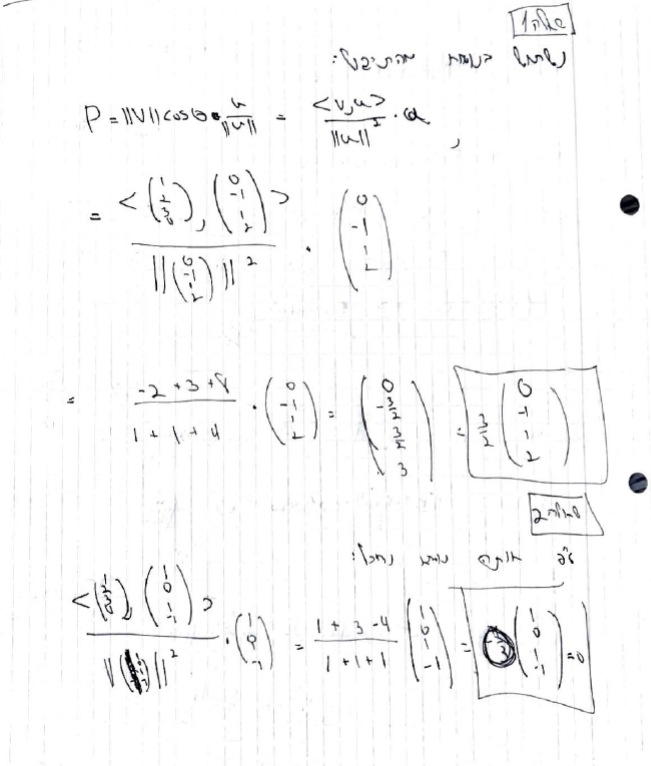
[Multivariate Calculus 7](#_Toc35968834)

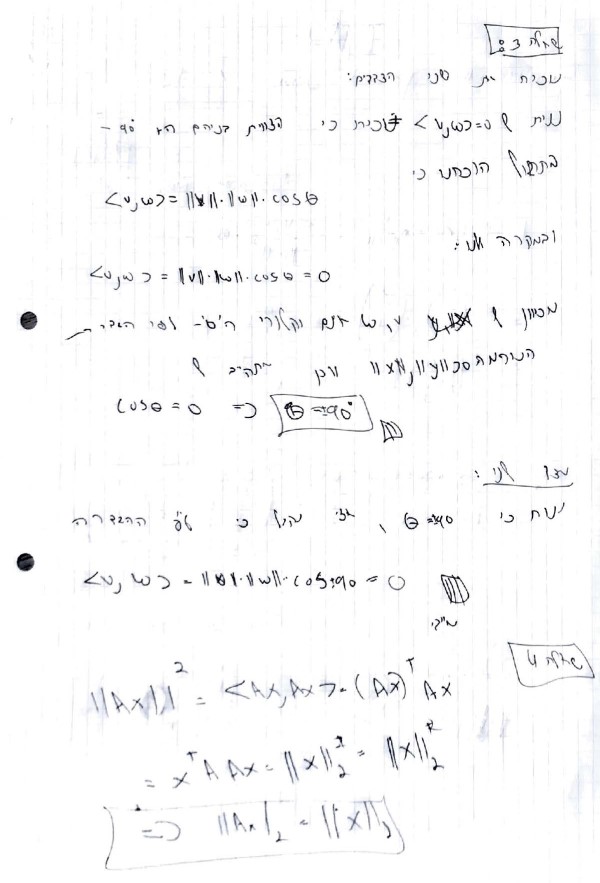
[Multivariate Gaussian- practical question 10](#_Toc35968835)

[Concentration inequalities - practical question 14](#_Toc35968836)

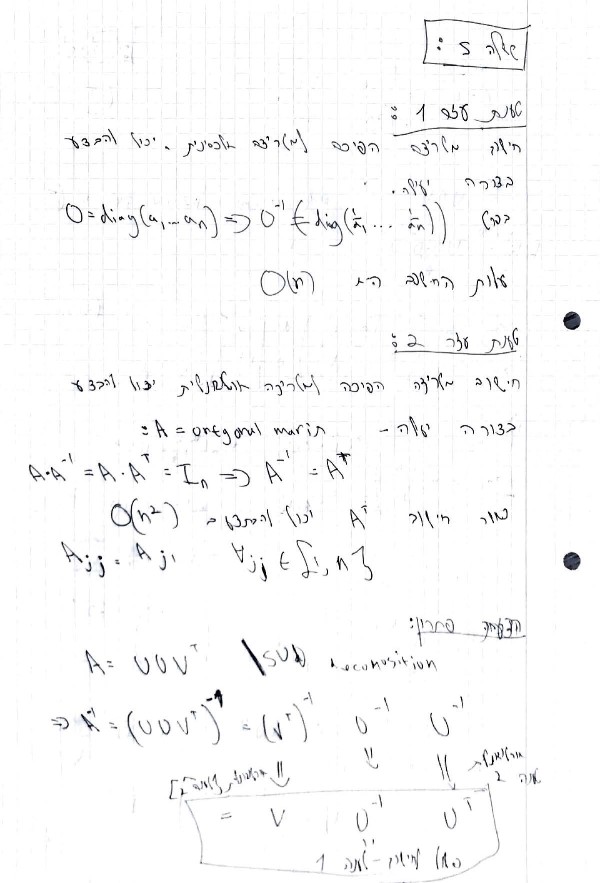
[Python Code 17](#_Toc35968837)

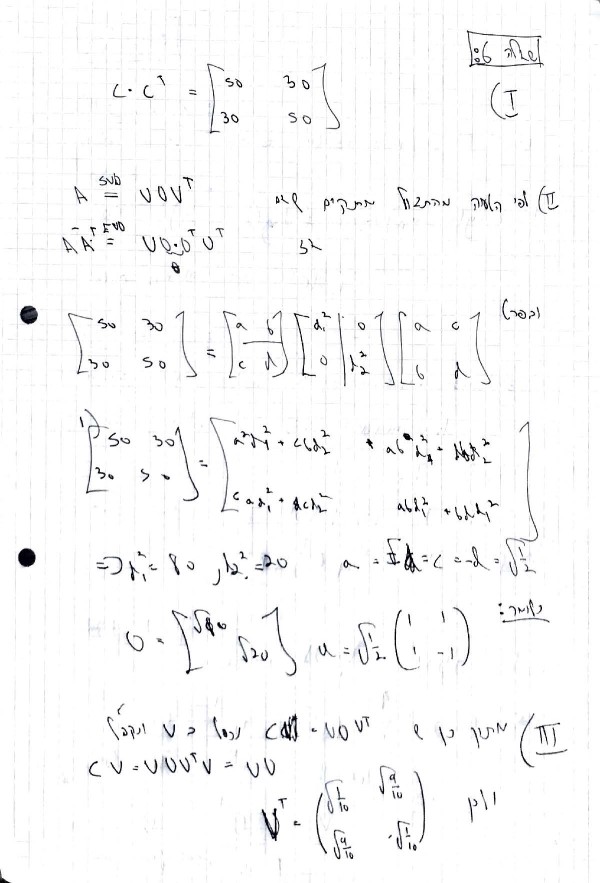
## Warm-up - Algebra Recap





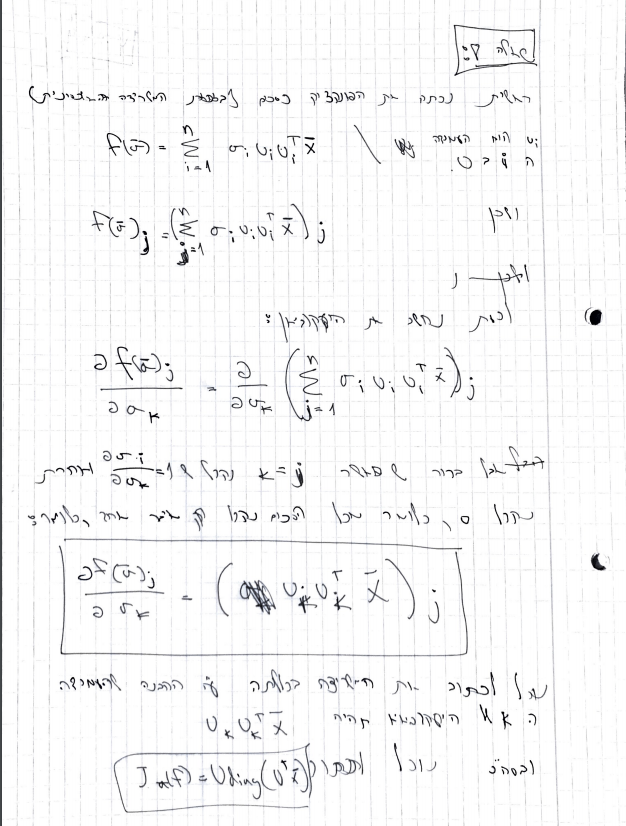
## SVD

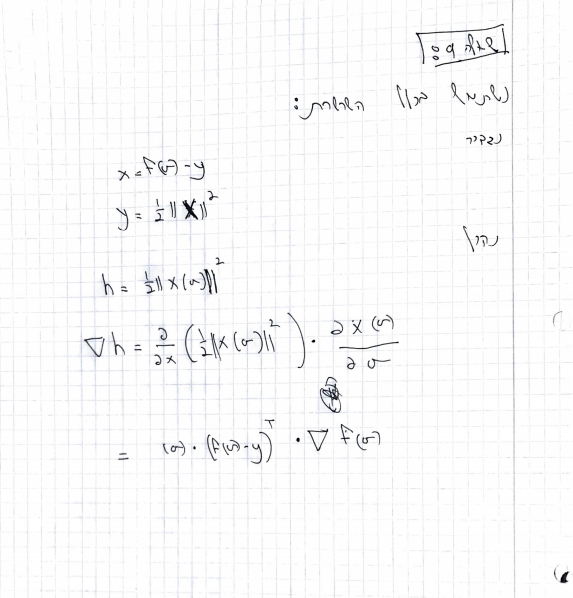


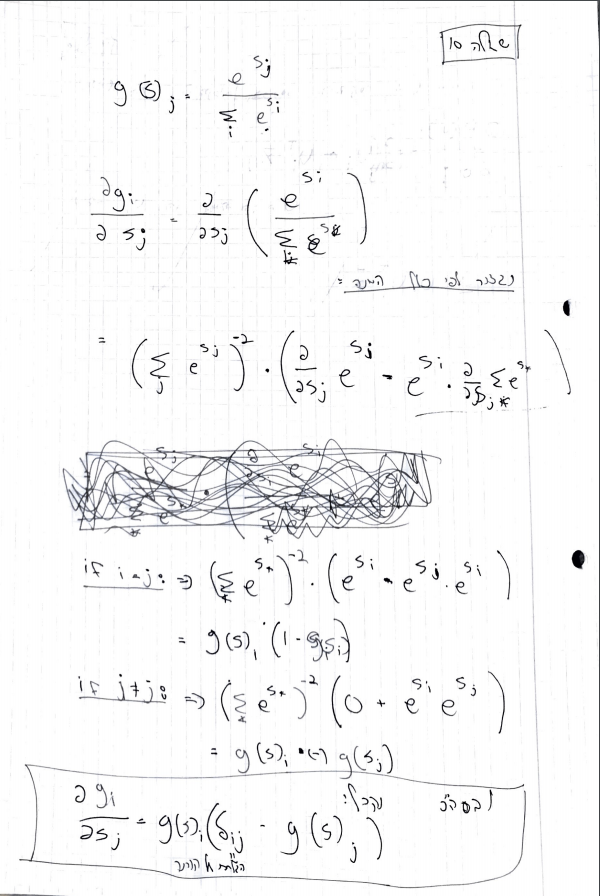


## 

## Multivariate Calculus

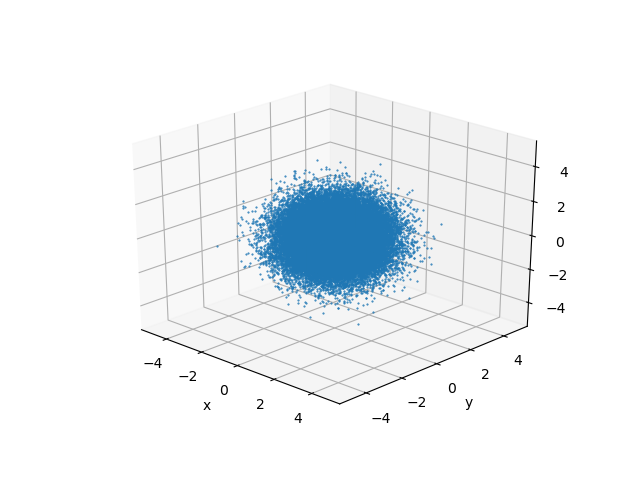


****

****

## Multivariate Gaussian- practical question

**שאלה 11:**



**שאלה 12:**

תמונה שמכילה ציור

התיאור נוצר באופן אוטומטי

מטריצת הCOV

[[0.01 0. 0. ]

[0. 0.25 0. ]

[0. 0. 4. ]]

ניתן לראות כי עדיין אין קורלציה בין הצירים (רק האיברים על האלכסון קיימים) אבל יש מתיחה – כמו שניתן לראות בסרטוט. יש מתיחה אבל אין סיבוב. עדיין מקביל לצירים.

**שאלה 13:**

תמונה שמכילה אובייקט

התיאור נוצר באופן אוטומטי

מטריצת הCOV:

[[0.2012325 0.77954016 0.22139559]

[0.77954016 3.38267323 1.18514263]

[0.22139559 1.18514263 0.67609427]]

המטריצה לא אלכסונית- יש גם מתיחה וגם סיבוב (כמו שרואים בסרטוט)

**שאלה 14:**

נעריך כי הסרטוט יראה כמו גאוסיאן דו מימד סביב אותם ערכים, וזה אכן מה שקיבלנו:  
תמונה שמכילה שולחן, מחשב נישא, מחשב

התיאור נוצר באופן אוטומטי  
**שאלה 15:**נצפה לראות גאוסיאן בעל אותם ערכים, אבל בצפיפות נמוכה יותר. כלומר צפיפות הנקודות דלילה יותר אבל הצורה והערכים זהים- וזה אכן מה שקיבלנו:  
תמונה שמכילה שולחן, צילום, מחשב נישא, ישיבה

התיאור נוצר באופן אוטומטי

## Concentration inequalities - practical question

**שאלה 16 A:**

נצפה לראות ככל שM גדל, שאיפה אסימפטומטית לממוצע האמיתי (אשר מוערך להיות 0.25)

תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטי

ניתן לראות כי כל השורות מתכנסות לערך מסוים ושכל הערכים הם "קרובים" לערך שציפינו לראות. כולם מתכנסים לאזור ה0.25 ("מתקרבים" זה לא מוגדר – ובהמשך השאלה נדייק יותר)  
  
**שאלה 16 B:**\* בגרף הראשון חתכתי את ציר הY בנקודה 5 כאשר הקו הסגול מעליה על מנת לקבל גרף ברור יותר **תמונה שמכילה צילום מסך, מפה

התיאור נוצר באופן אוטומטיתמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטי**  
  
**שאלה 16C:**

נצפה לראות שככל שM גדל נשאף למאה אחוז (1) – כאשר ככל שאפסילון גדול יותר נשאף מהר יותר  
תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטי  
וזה אכן מה שקיבלנו- כאשר בעבור אפסילון קטן מאוד מספר הדגימות לא מספיק

## Python Code

**Python CODE**

def conditional(x\_y\_z):  
 temp=x\_y\_z[0:2,:]  
 temp2=x\_y\_z[2,:]  
 # cond=temp2>-0.4 and temp2<0.1  
 return temp[:,np.logical\_and(temp2>-0.4 , temp2<0.1)]  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 mean = [0, 0, 0]  
 cov = np.eye(3)  
 x\_y\_z = np.random.multivariate\_normal(mean, cov, 50000).T  
 # plot\_3d(x\_y\_z);  
 s=np.diag([0.1,0.5,2])  
 newData=np.matmul(s,x\_y\_z)  
 newCov=np.matmul(np.matmul(s,cov),s.transpose())  
 print("scale cove:")  
 print(newCov)  
 # plot\_3d(newData)  
 ortoM=get\_orthogonal\_matrix(3)  
 ortoData=np.matmul(ortoM,newData)  
 # plot\_3d(ortoData)  
 print("orto cove:")  
 print(np.matmul(np.matmul(ortoM,newCov),ortoM.transpose()))  
  
 # plot\_2d(x\_y\_z[0:2,:])  
 # plot\_2d(newData[0:2,:])  
 # plot\_2d(ortoData[0:2,:])  
  
 plot\_2d(conditional(x\_y\_z))  
 plot\_2d(conditional(newData))  
 plot\_2d(conditional(ortoData))

import numpy as np  
import matplotlib.pyplot as plt  
import math  
row=1000  
col=100000  
data = np.random.binomial(1, 0.25, (col,row))  
epsilons=(0.5, 0.25, 0.1, 0.01, 0.001)  
  
# part a:  
av=np.zeros(row)  
for i in range(5):  
 for m in range(1,row):  
 av[m]=np.mean(data[i,1:m])  
 plt.plot(av)  
plt.show()  
  
# part b:  
# Chebyshev  
i=1  
upperBoundChebyshev=np.zeros(row)  
fig = plt.figure()  
ax = fig.add\_subplot(111)  
  
for epsilon in epsilons:  
 var=np.var(data[i,:])  
 for m in range(1,row):  
  
 upperBoundChebyshev[m]=var/(m\*epsilon\*epsilon)  
 plt.plot(upperBoundChebyshev[5:,],label='epsilon='+str(epsilon))  
legend = ax.legend()  
plt.ylim([0,5])  
plt.title("Chebyshev upper bound")  
  
plt.show()  
  
# Hoeffding  
i=1  
upperBoundHoeffding=np.zeros(row)  
fig = plt.figure()  
ax = fig.add\_subplot(111)  
  
for epsilon in epsilons:  
 var=np.var(data[i,:])  
 for m in range(1,row):  
  
 upperBoundHoeffding[m]=2\*math.exp(-2\*m\*epsilon\*epsilon)  
  
 plt.plot(upperBoundHoeffding[1:,],label='epsilon='+str(epsilon))  
legend = ax.legend()  
plt.title("Hoeffding upper bound")  
plt.show()  
  
# part c  
fig = plt.figure()  
ax = fig.add\_subplot(111)  
pres=np.zeros(row)  
for epsilon in epsilons:  
 print(epsilon)  
 for m in range(1,row):  
 temp=np.mean(data[:,0:m],axis=1)  
 pres[m]=(col-np.count\_nonzero(temp > 0.25+epsilon) -np.count\_nonzero(temp < 0.25-epsilon))/col  
 plt.plot(pres[1:],label='epsilon='+str(epsilon))  
legend = ax.legend()  
plt.title("16 C")  
  
plt.show()