Contents

1 ./3d gausian.py

```
import numpy as np
import pandas as pd
from plotnine import *
import matplotlib.pyplot as plt
from matplotlib import rc

from mpl_toolkits.mplot3d import Axes3D
from scipy.linalg import qr

# rc('text', usetex=True)
mean = [0, 0, 0]
cov = np.eye(3)
x_y_z = np.random.multivariate_normal(mean, cov, 50000).T


def get_orthogonal_matrix(dim):
    H = np.random.randn(dim, dim)
    Q, R = qr(H)
    return Q


def plot_3d(x_y_z):
    '''
    plot points in 3D
    :param x_y_z: the points. numpy array with shape: 3 X num_samples (first dimension for x, y, z
    coordinate)
    '''
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.scatter(x_y_z[0], x_y_z[1], x_y_z[2], s=1, marker='.', depthshade=False)
    ax.set_xlim(-5, 5)
    ax.set_ylim(-5, 5)
    ax.set_zlim(-5, 5)
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')


def plot_2d(x_y):
    '''
    plot points in 2D
    :param x_y_z: the points. numpy array with shape: 2 X num_samples (first dimension for x, y
    coordinate)
    '''
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(x_y[0], x_y[1], s=1, marker='.')
    ax.set_xlim(-5, 5)
    ax.set_ylim(-5, 5)
    ax.set_xlabel('x')
    ax.set_ylabel('y')



def section11():

    plot_3d(x_y_z)
    plt.title("Normal distribution $ \sigma(x) = \sigma(y) = \sigma(z)= 1 $ and $ \mu = 0 $")
    plt.savefig("sec11.png")
```

```python

def section12():
    S = np.diag([0.1 , 0.5, 2])
    S_transform = S.dot( x_y_z )
    plot_3d( S_transform )
    plt.title("transformed Normal distribution $ S \mathcal{N} $")
    plt.savefig("sec12.png")


    S_trandform_cov = np.cov(S_transform)
    print( S_trandform_cov )


def section13():
    S = np.diag([0.1 , 0.5, 2])
    S_transform = S.dot( x_y_z )


    rndmatrix = get_orthogonal_matrix( 3 )

    S_trans_mul_by_orth =\
        rndmatrix.dot( S_transform )

    plot_3d( rndmatrix.dot( x_y_z ) )
    plt.title("Normal distribution multiplied by random Diagonal $ diag \cdot  \mathcal{N} $")
    plt.savefig("sec13mulbyrand_orth.png")


    plot_3d( S_trans_mul_by_orth )
    plt.title("transformed Normal distribution multiplied by random Diagonal $ diag \cdot  S \mathcal{N} $")
    plt.savefig("sec13mulbyrand_orth_S.png")


def plot_xy_proj_histo(x_y_z, sec, _title):
    #ploting 2d proj
    plot_2d(x_y_z[:-1])
    plt.savefig("{0}proj.png".format(sec))
    # iterating over the x any y axes.
    for X, _dim in zip(x_y_z[:-1] , ["x" , "y"]):
        # calculate histogram
        fig = plt.figure()
        x , y  = np.histogram( X, bins = 600 )
        plt.plot( y[:-1], x,
         label="histogram of {0} dim".format(_dim) )
        plt.title(_title.format(_dim))
        plt.xlabel(_dim)
        plt.ylabel('points')
        plt.savefig("{0}proj_hist_{1}.png".format(sec,_dim))

def section14():

    plot_xy_proj_histo(x_y_z, "section14", "Histogram of projection Normal distribution  $ proj( data , {0}) $")



def section15():

    query =  (x_y_z[2] >= -0.4) & (x_y_z[2] <= 0.1)
    filtred = np.array( [ x_y_z[_][query] for _ in range(3) ]
     ,dtype=np.float64)
    print(filtred)
    plot_xy_proj_histo( filtred, "section15", "Histogram of projection Normal distribution for $ z \in (-0.4,0.1) proj( data


def section16():
    length  = 10**5
    samples = 1000
    data = np.random.binomial(1, 0.25, (samples, length))
```

```python
epsilon = [0.5, 0.25, 0.1, 0.01, 0.001]


plt.figure()

def plot_estimate_function_as_m(row , i):
    '''
        ploting the arithmetic mean of given sample.
    '''
    X, Y, _accsum = [0], [0], 0
    for coin, value in enumerate(row):
        if coin > 0 :
            _accsum += value
            X.append( coin )
            Y.append( _accsum / coin  )

    plt.scatter(X, Y, s=1, marker='.', label="{0}`s sample".format(i) )


# plt.title("{0}'s dataset'".format(i))
plt.xlabel("flips")
plt.ylabel("aritmetic mean")
for i in range(5):
    plot_estimate_function_as_m( data[i], i)

plt.legend()
plt.title("estimate as function of flips" )
plt.savefig("section16a.png")

def bound_by_one(X ,Y):
    '''
        bound_by_one - assignment 1's at the greater values.
    '''
    Y [Y > 1] = 1
    return X, Y

def hoffding(eps, length):
    '''
        estimate of the probability by Hoffding.
    '''
    X = np.arange(length)
    return bound_by_one(X , 2* np.exp(-2*(X)*(eps**2)))

def chaviapprox(eps, length):
    '''
        estimate of the probability by Chebyshevs.
    '''
    X = np.arange(length)
    return bound_by_one(X, 0.25  / ( X * eps**2))

def plotingseq_distance_from_bound(eps, length, data):
    '''
        calculate the amount of samples which their aritmatic mean is
        greater than given epsilon.
    '''
    precentage = [ ]
    for m in range(1,length+1):
        count = 0
        for sample in data:
            if ( abs(np.sum(sample[:m]) - 0.25*m) > eps*m):
                count += 1
        precentage.append(count / samples)
    return np.arange(length), precentage


relevantlength = [100 ,100 ,100 ,10**4 ,10**5 ]

funcnaems = [ "Chebyshevs" , "Hoffding"  ]
```

```
    for i, eps in enumerate( epsilon ):
        if i == 4:
            plt.figure()
            for j, func in enumerate( [chaviapprox, hoffding] ):
                plt.scatter( *(func(eps, relevantlength[i] )),
                 s=1, marker='.' ,  label="{0}".format(funcnaems[j]))
            plt.scatter(\
             *plotingseq_distance_from_bound(eps, relevantlength[i], data),\
              s=1, marker='.', label=" $ P( \\overline{{x}} - E[x] > {0})$".format(eps))
            plt.title("upper bound, $ \\varepsilon = {0} $".format(eps))
            plt.xlabel("flips")
            plt.ylabel("probability bound")
            plt.legend()
            plt.savefig("section16b{0}.png".format(i))



if __name__ == "__main__":

    section11()
    section12()
    section13()
    section14()
    section15()
    section16()
```

תרגיל 1

7 באפריל 2020

דויד פונרובסקי 208504050.

1 חימום.

1.1 סעיפים 1 + 2:

$$proj\left(u,v\right)=\frac{\langle u,v\rangle}{||u||}\cdot\frac{u}{||u||}=\begin{cases}(1)\rightarrow & \frac{1}{1^2+(-1)^2+2^2}\begin{bmatrix}1\\2\\3\\4\end{bmatrix}\cdot\begin{bmatrix}0\\-1\\1\\2\end{bmatrix}\begin{bmatrix}0\\-1\\1\\2\end{bmatrix}=\frac{6}{9}\begin{bmatrix}1\\0\\1\\-1\end{bmatrix}\\ (2)\rightarrow & \frac{1}{1^2+(-1)^2+1^2}\begin{bmatrix}1\\2\\3\\4\end{bmatrix}\cdot\begin{bmatrix}1\\0\\1\\-1\end{bmatrix}\begin{bmatrix}1\\0\\1\\-1\end{bmatrix}=\frac{0}{3}\begin{bmatrix}1\\0\\1\\-1\end{bmatrix}=0\end{cases}$$

1.2 סעיף 3:

טענה- $\langle u,v\rangle=0\Leftrightarrow$ הזווית בין u ל v היא $\pm\frac{\pi}{2}$. ל $u,v\in\mathbb{R}^m$. הוכחה, נחשב את $\langle u,v\rangle$:

$$0=\langle u,v\rangle=||u||\cdot||v||\cos\theta$$

מיצד שני נתון כי $u,v\neq\vec{0}$ ולכן $||u||,||v||>0$ זה יכל לקרות אם ורק אם $\cos\theta=0$ ולכן $\theta=\pm\frac{\pi}{2}$ (עד כדי מחזור 2π).

1.3 סעיף 4:

טענה- אם $A\in Orthonormal$ אזי $||Au||_2=||u||_2$. הוכחה: נזכיר כי $A^tA=\mathbb{I}$, ונקבל כי :

$$||Au||_2=\left(\left(Au\right)^tAu\right)^{\frac{1}{2}}=\left(u^tA^tAu\right)^{\frac{1}{2}}=\left(u^tu\right)^{\frac{1}{2}}=||u||_2$$

2 SVD :

2.1 סעיף 5:

נתונה לנו A ביצוג SVD כלומר $A = UDV^T$ כאשר U ו V הן אורטונורמליות. לכן נקבל כי $A^{-1} = \left(UDV^T\right)^{-1} = \left(V^T\right)^{-1} D^{-1}U^{-1} = VD^{-1}U^T$. כשאשר המעבר האחרון $U^{-1} = U^T$ נובע מכך ש U אורטונורמלית (לפי משפט).
נשים לב שחישוב ההופכית על ידי דירוג $[A|\mathbb{I}]$ (גאוס-ג'ורדן) ידרוש מאיתנו $\mathcal{O}\left(n^3\right)$ זמן, כאשר n הוא מספר השורות (והעמודות) של A. לעומת זאת מאחר ו $D \in diag$ ניתן לחשב את D^{-1} ב $\mathcal{O}\left(n\right)$ ואת U^T ניתן לחשב ב $\mathcal{O}\left(n^2\right)$.
את המימוש של U^T ניתן לשפר בכך שבמקום להחליף את U ניתן פשוט להחליף את האיטרטור, כך שבעת גישה ל $\left(U^T\right)_{i,j}$ נקבל את $U_{j,i}$. זאת כמובן ב $\mathcal{O}\left(1\right)$.

2.2 סעיף 6:

נשים לב כי $C^T = \left(UDV^T\right)^T = VD^TU^T$ אבל $D \in diag$ ולכן $D^T = D$ כלומר $C^T = VDU^T$. מכאן נקבל כי $C^TC = VDU^TUDV^T = VD^2V^T$. כלומר נוכל למצוא את D ו V על ידי ליכסון של C^TC.

$$C^TC = \begin{bmatrix} 5 & -1 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix} = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$$

נמצא את הערכים העצמאים:

$$det\left(C^TC - \lambda\mathbb{I}\right) = 0 \Rightarrow (26-\lambda)(74-\lambda) - 18^2 = 0$$

פיתרונות של המשוואה שלעיל הם: $\lambda = 20, 80$ ולכן $D^2 = \begin{bmatrix} 80 & 0 \\ 0 & 20 \end{bmatrix}$ כלומר $D = \begin{bmatrix} \sqrt{80} & 0 \\ 0 & \sqrt{20} \end{bmatrix}$. נמצא את הוקטורים העצמאים:

$$\begin{bmatrix} 26-20 & 18 \\ 18 & 74-20 \end{bmatrix} = \begin{bmatrix} 6 & 18 \\ 18 & 54 \end{bmatrix} = 6\begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} \Rightarrow v = normfactor \cdot \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$
$$\begin{bmatrix} 26-80 & 18 \\ 18 & 74-80 \end{bmatrix} = \begin{bmatrix} -54 & 18 \\ 18 & -6 \end{bmatrix} = 6\begin{bmatrix} -9 & 3 \\ 3 & -1 \end{bmatrix} \Rightarrow v = normfactor \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$
$$V^T = normfactor \cdot \begin{bmatrix} 1 & 3 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{10}} & \sqrt{\frac{9}{10}} \\ \sqrt{\frac{9}{10}} & -\sqrt{\frac{1}{10}} \end{bmatrix}$$

בנוסף אנו יודעים כי $C = UDV^T \Rightarrow U = D^{-1}VC$ ולכן U שווה ל-

$$U = \begin{bmatrix} \frac{1}{\sqrt{80}} & 0 \\ 0 & \frac{1}{\sqrt{20}} \end{bmatrix} \begin{bmatrix} \sqrt{\frac{1}{10}} & \sqrt{\frac{9}{10}} \\ \sqrt{\frac{9}{10}} & -\sqrt{\frac{1}{10}} \end{bmatrix} \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix} =$$
$$= \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} \end{bmatrix}$$

2.3 סעיף 7:

אלגוריתם למציאת מטריצת SVD. נסמן, $C_0 = A^T A$ וב $\lambda_1 ... \lambda_n$, $v_1 ... v_n$ את הע"ע והו"ע של C_0. וב $m = argmax\{\lambda_i\}$ נראה את נכונות הטענה, כלומר שהסדרה $b_{k+1} = \frac{C_0 b_k}{\|C_0 b_k\|} \underset{k\to\infty}{\rightarrow} v_m$ לכל תנאי התחלה עבורם $b_{0_m} \neq 0$.

הוכחה לפי סעיף קודם נזכיר כי לפי התרגיל הקודם : $C_0 = VD^2V^T$ כאשר V אורטונורמלית מכאן-

$$b_{k+1} = \frac{C_0 b_k}{\|C_0 b_k\|} = \frac{VD^2V^T b_k}{\|VD^2V^T b_k\|} = \frac{VD^2V^T \frac{VD^2V^T b_{k-1}}{\|VD^2V^T b_{k-1}\|}}{\left\|VD^2V^T \frac{VD^2V^T b_{k-1}}{\|VD^2V^T b_{k-1}\|}\right\|}$$

ניעזר בכך ש V אורטונורמלית ולכן $V^TV = \mathbb{I}$. בנוסף $\left\|VD^2V^T b_{k-1}\right\| \in \mathbb{R}$ ולכן ניתן לצמצם את הפקטורים כי $||\alpha u|| = \alpha||u||$. נקבל כי

$$b_{k+1} = \frac{VD^4V^T b_{k-1}}{\|VD^4V^T b_{k-1}\|}$$

ניפתח באופן איטרטיבי (אינדוקצה).

$$b_{k+1} = \frac{VD^{2k}V^T b_0}{\|VD^{2k}V^T b_0\|}$$

נפעיל כמה טריקים, אם $b_0 \sim \mathcal{D}$ אז גם $V^T b_0 \sim \mathcal{D}$ (כפל של גודל דטרמינסטי במשתנה מקרי משמר התפלגות). בנוסף מאורטונורמליות V משמרת נורמה, נסמן $\tilde{b_0} = V^T b_0$ וב- $b_{k+1}^{(j)}$ את הקורדינטה ה j של b_{k+1}.ונצטמצם ל-

$$b_{k+1} = \frac{VD^{2k}\tilde{b_0}}{\left\|VD^{2k}\tilde{b_0}\right\|} = \frac{VD^{2k}\tilde{b_0}}{\left\|D^{2k}\tilde{b_0}\right\|}$$

$$\Rightarrow b_{k+1}^{(l)} = \sum_j V_{l,j} \frac{\lambda_j^k b_0^{\tilde{(j)}}}{\left(\sum_i^n \left(\lambda_i^k b_0^{\tilde{(i)}}\right)^2\right)^{\frac{1}{2}}} = \sum_j V_{l,j} \frac{\left[\frac{\lambda_j}{\lambda_m}\right]^k \frac{b_0^{\tilde{(i)}}}{b_0^{\tilde{(m)}}}}{\left(1 + \sum_{i\neq m}^n \left(\left[\frac{\lambda_i}{\lambda_m}\right]^k \frac{b_0^{\tilde{(i)}}}{b_0^{\tilde{(m)}}}\right)^2\right)^{\frac{1}{2}}}$$

כאשר ונקבל שלכל $j \neq m$ הביטוי - $\left[\frac{\lambda_i}{\lambda_m}\right]^k \rightarrow 0$ ולכן $b_{k+1}^{(l)} = V_{l,m}$ כלומר b_{k+1} שווה לעמודת m ב V וזאת בדיוק v_m. (כי V היא המטריצה המלכסנת את A^TA) ∎

שלב שני, **הורדת מימד** ,נגריל עכשיו וקטור חדש כאשר $b_0^{\tilde{(m)}} = 0$ ונשים לב כי $\lambda_m^k b_0^{\tilde{(m)}} = 0$, מכאן נסמן ב$m'$ את הע"ע השני בגודלו ונקבל בדיוק באות הדרך כי b_{k+1} מתכנס ל $v_{m'}$. נמשיך באופן איטרטיבי ונמצא כך את V.

כמובן ש $C_0 = VD^2V^T \Rightarrow V^TC_0V = D^2$ ולכן מצאנו גם את D. ולבסוף נישתמש ב $U = D^{-1}VC_0$ כדי למצוא את U.

זמן ריצה:

נחסום את $||b_k, v_m||$:

$$||b_k, v_m|| = ||\sum_j V_{m,j} \frac{\left[\frac{\lambda_j}{\lambda_m}\right]^k \frac{b_0^{\tilde{(i)}}}{b_0^{\tilde{(m)}}}}{\left(1 + \sum_{i\neq m}^n \left(\left[\frac{\lambda_i}{\lambda_m}\right]^k \frac{b_0^{\tilde{(i)}}}{b_0^{\tilde{(m)}}}\right)^2\right)^{\frac{1}{2}}} - V_{m,j}|| \leq$$

$$\leq ||\sum_j V_{m,j} \left[\frac{\lambda_j}{\lambda_m}\right]^k \frac{b_0^{\tilde{(i)}}}{b_0^{\tilde{(m)}}} - V_{m,j}|| \leq$$

$$\overset{*}{\leq} \sqrt{(n-1)\left(V_{m',j}\left[\frac{\lambda_{m'}}{\lambda_m}\right]^k \frac{b_0^{\tilde{(m')}}}{b_0^{\tilde{(m)}}}\right)^2}$$

כאשר m' הוא האינדקס של הע"ע השני הכי גדול. $*$ כמובן שהאי-שיוון נכון אחל ממקום מסויים (צריך לבחור k מספיק גדול כדי לבטיח שהפקטור $V_{m',j}\frac{b_0^{\tilde{(m')}}}{b_0^{\tilde{(m)}}}$ יהי יחסית לא משמעותי ביחס ל $\left[\frac{\lambda_{m'}}{\lambda_m}\right]$, אינפי 1). נמשיך :

$$\leq \sqrt{n}V_{m',j}\left[\frac{\lambda_{m'}}{\lambda_m}\right]^k\frac{b_0^{(\tilde{m'})}}{b_0^{(\tilde{m})}}\leq\varepsilon$$

ולכן עבור דיוק של לפחות ε נידרוש :

$$k\geq\log_{\frac{\lambda_{m'}}{\lambda_m}}\left(\frac{b_0^{(\tilde{m})}}{b_0^{(\tilde{m'})}V_{m',j}\sqrt{n}}\varepsilon\right)=\log_{\frac{\lambda_m}{\lambda_{m'}}}\left(\frac{b_0^{(\tilde{m'})}V_{m',j}\sqrt{n}}{b_0^{(\tilde{m})}\varepsilon}\right)\geq$$
$$\log_{\frac{\lambda_m}{\min\lambda}}\left(\frac{b_0^{(\tilde{m'})}V_{m',j}\sqrt{n}}{b_0^{(\tilde{m})}\varepsilon}\right)=\mathcal{O}\left(\ln\left(\frac{\sqrt{n}}{\varepsilon}\right)\right)$$

ההיפוך בסיסים ב $\log$ היה חיוני מאחר ו- $\frac{\lambda_{m'}}{\lambda_m}\leq 1$ בהצגה זאת אנחנו לא מרוויחים אינפורמציה אינטואטיבית.
נכחיש כרגע את ההשפעות של שגיאת ε על האיטרציות הבאות. חישוב b^{k+1} מ b^k דורש $\mathcal{O}\left(n\right)$ עבודה (נשים לב שאת הסכום במכנה אנו מחשבים רק פעם אחת עבור כל הקורדינטות של b^{k+1}), ולכן נבקבל שאת V ניתן לחשב ב $\mathcal{O}\left(n^2\ln\left(\frac{\sqrt{n}}{\varepsilon}\right)\right)$. את U ניתן לחשב באותו אופן, נפעיל את אותה וריאציה על $AA^T=C_0^T$. D מורכבת מ העע"מ של A, נעבור עמודה עמודה ב $v_m\in V$ וניפתור את המשווה $Av_m=\lambda v_m$, כאשר מספיק לנו לפתור רק עבור הקורדינטנה הראשונה שאינה שווה ל -0 כלומר $\lambda_m=\frac{\sum A_0^i v_m^i}{v_m^0}$ כלומר חישוב λ_m דורש $\mathcal{O}\left(n\right)$ וחישוב D דורש $\mathcal{O}\left(n^2\right)$ ולכן סך הכל נישאר עם זמן ריצה אסימפטוטי של $\mathcal{O}\left(n^2\ln\left(\frac{\sqrt{n}}{\varepsilon}\right)\right)$.

3 חלק שלישי.

3.1 סעיף 8:

יהיה $x\in\mathbb{R}^n$ וקטור קבוע, ו $U\in\mathbb{R}^{n\times n}$ אורטוגונלית. נחשב את היעקובן של

$$f\left(\sigma\right)=Udiag\left(\sigma\right)U^Tx$$
$$\Rightarrow f\left(\sigma\right)_i=U_i^j\sigma_j\delta_j^k\left(U^T\right)_k^lx_l=U_i^k\sigma_k\left(U^T\right)_k^lx_l$$

$$J\left(f\right)_{i,j}=\frac{\partial f_i}{\partial\sigma_j}=U_i^k\delta_{k,j}\left(U^T\right)_k^lx_l=U_i^j\left(U^T\right)_j^lx_l$$

נשים לב כי רק l אינדקס רץ, ולכן הכתיב שקול ל- $J\left(f\right)=Udiag\left(U^Tx\right)$.

3.2 סעיף 9:

נתון $h\left(\sigma\right)=\frac{1}{2}\left|f\left(\sigma\right)-y\right|^2$ נחשב את הגרדיאנט :

$$\nabla h_i=\frac{\partial h}{\partial\sigma_i}=\frac{\partial h}{\partial f}\frac{\partial f}{\partial\sigma_i}=\left(f\left(\sigma\right)-y\right)\frac{\partial f}{\partial\sigma_i}\Rightarrow\nabla h=\left(f\left(\sigma\right)-y\right)^T\nabla f$$

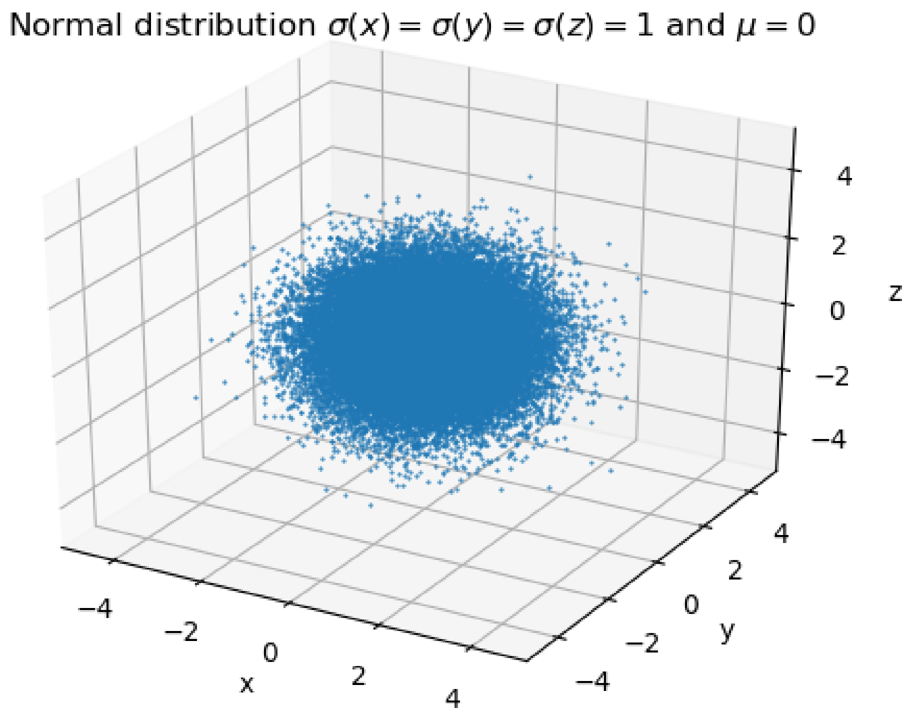
3.3 סעיף 10:

נחשב את היעקבויאן של ה $softmax$: $g\left(\boldsymbol{z}\right)_j=\frac{e^{z_j}}{\sum e^{z_k}}$

$$J\left(g\right)_{i,j}=\frac{\partial g_i}{\partial z_j}=\frac{\delta_{i,j}e^{z_i}\sum e^{z_k}-e^{z_j}e^{z_i}}{\left(\sum e^{z_k}\right)^2}=\frac{\delta_i^je^{z_i}\sum e^{z_k}}{\left(\sum e^{z_k}\right)^2}-\frac{e^{z_j}e^{z_i}}{\left(\sum e^{z_k}\right)^2}=$$
$$=\delta_i^jg_i-g_ig_j=g_i\left(\delta_i^j-g_j\right)$$

4 חלק רביעי.

4.1 סעיף 11:

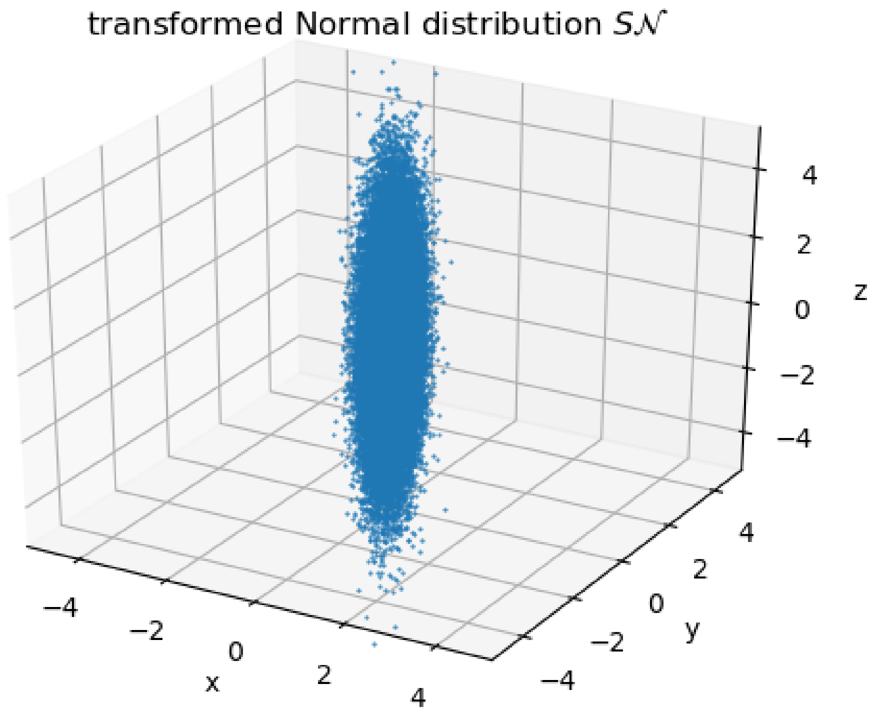


4.2 סעיף 12:

אנליטית: אנו יודעים כי $cov\left(ax,y\right) = a \cdot cov\left(x,y\right)$ ולכן נצפה לקבל את - $cov_{i,i} = diag\left(S_{i,i}^2\right) \Rightarrow S^2$.
באופן נומרי מקבל את:

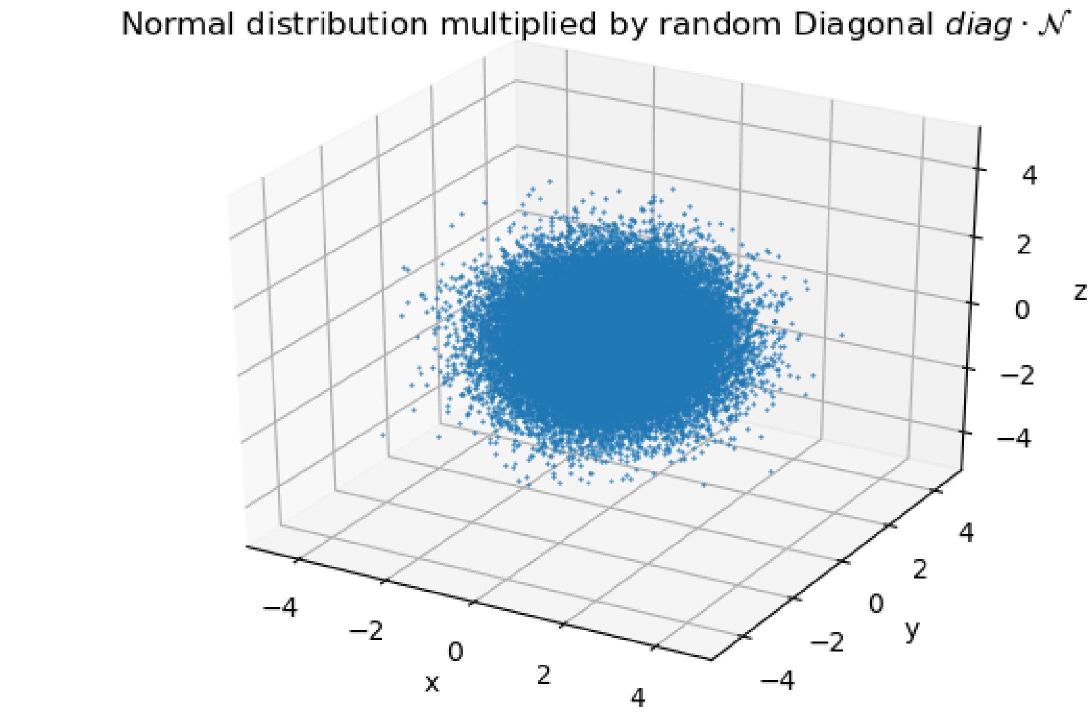
```
davidponar@DESKTOP-M43N0AS MSYS ~/workspace/iml/exercies/ex1 (master)
$ python ./3d_gausian.py
[[1.00755762e-02 1.17976315e-03 2.67707751e-03]
 [1.17976315e-03 2.41934079e-01 1.53245781e-02]
 [2.67707751e-03 1.53245781e-02 3.99995184e+00]]

davidponar@DESKTOP-M43N0AS MSYS ~/workspace/iml/exercies/ex1 (master)
$ python ./3d_gausian.py
[[ 1.00194739e-02  1.16457886e-04 -7.34568479e-04]
 [ 1.16457886e-04  2.51434529e-01 -3.21494511e-03]
 [-7.34568479e-04 -3.21494511e-03  4.03340142e+00]]
```

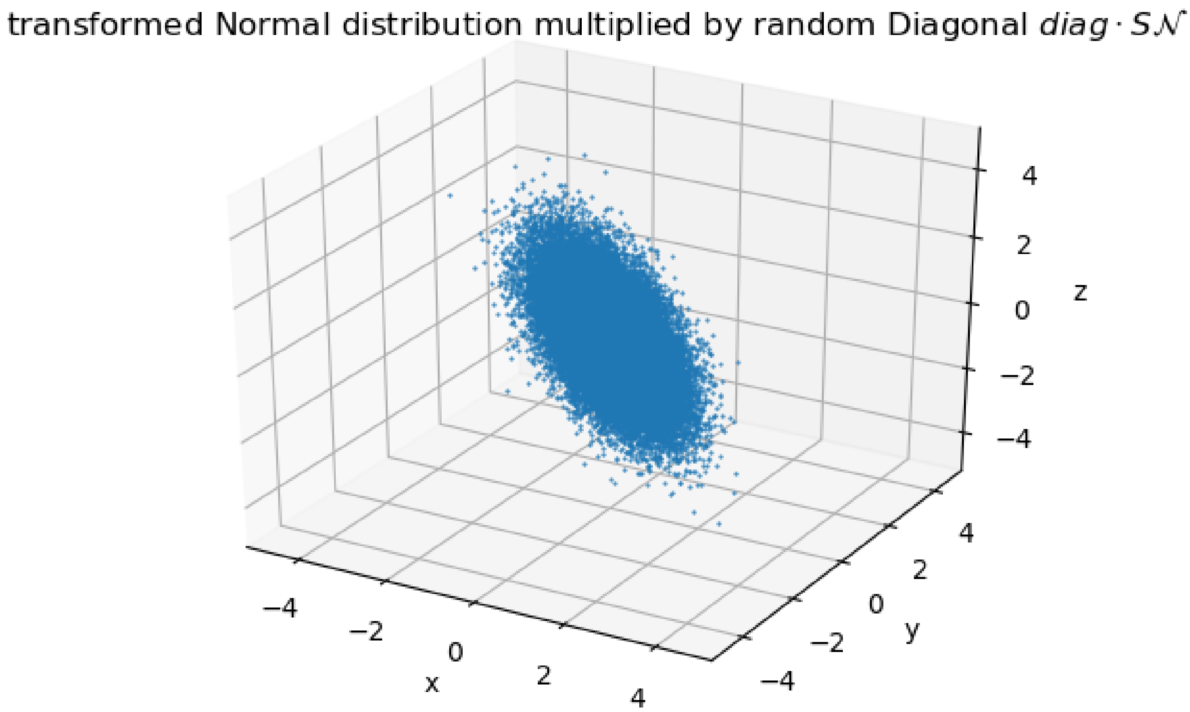
כאשר החישוב הראשון הוא ל $n = 5 \cdot 10^3$ בעוד שהחישוב השני הוא ל $n = 5 \cdot 10^4$ כלומר קל לראות שהמטרציה שואפת ל S^2.
הנקודת עצמן:



4.3 סעיף 13 :

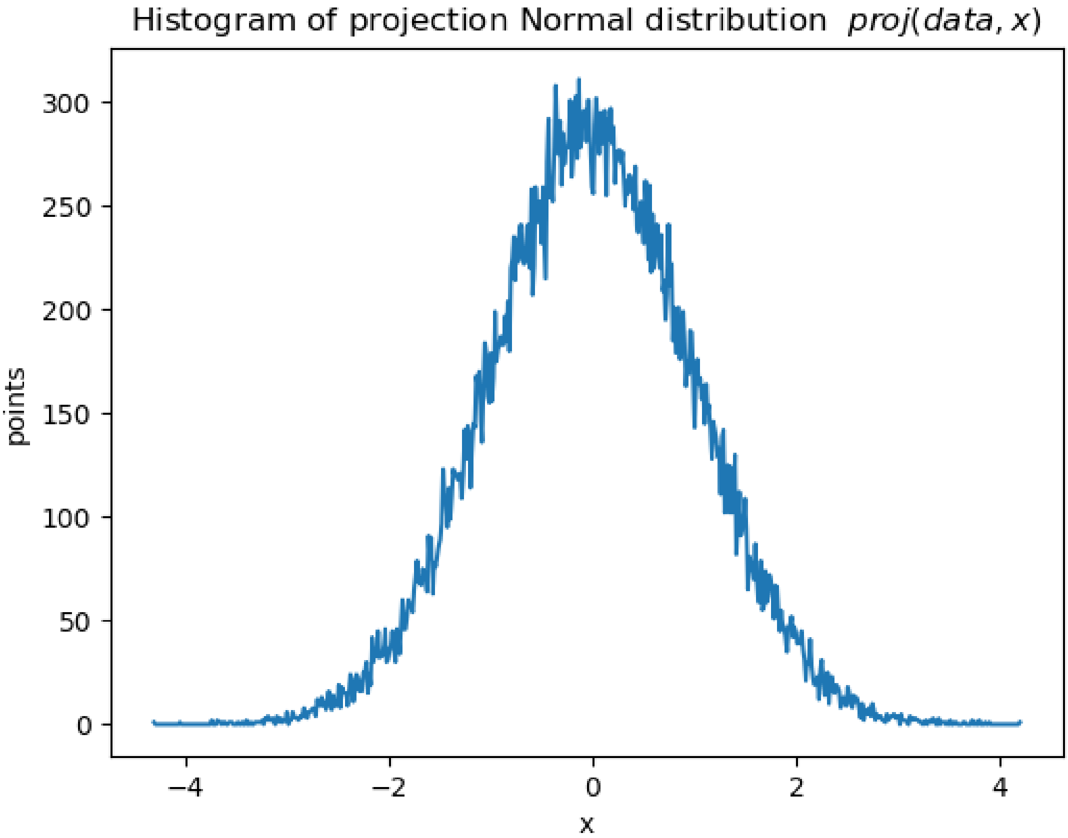
למעשה אנו מסובבים את הדאטה שלנו בזווית רנדומית. נמסן ב X את המידע, וב SX את המידע לאחר המתיחה. לX הייתה סימטריה לסיבובים ($x \sim y \sim z$) ולכן נצפה שלאחר הסיבוב נקבל עדיין כי $cov\left(\mathcal{R}X\right) = cov\left(X\right)$. לעומת זאת $\mathcal{R}\left(SX\right) = \left(\mathcal{R}S\right)X$ ולכן נצפה לקבל סיבוב של המידע ומכאן גם סיבוב של ווקטור השונות, (עכשיו cov כבר לא תיהיה אלכסון).



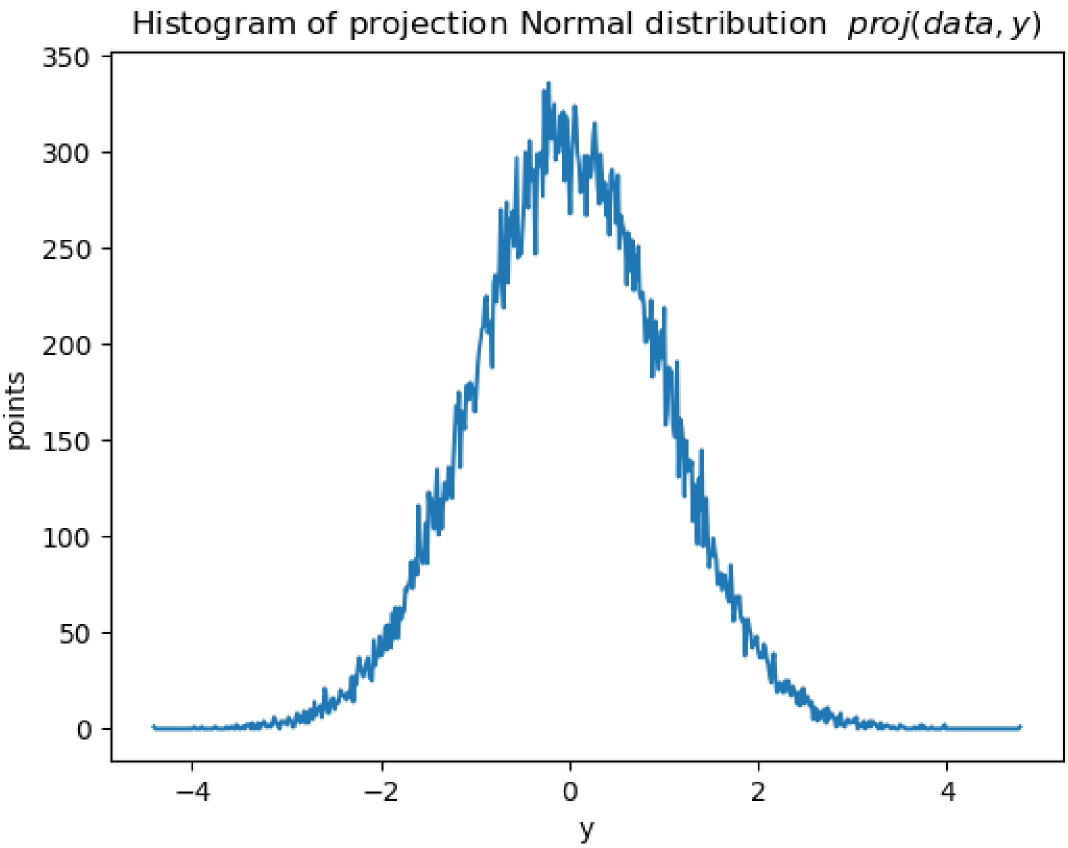


4.4 סעיף 14 :

היסטוגרמת ההטלה על ציר הx :

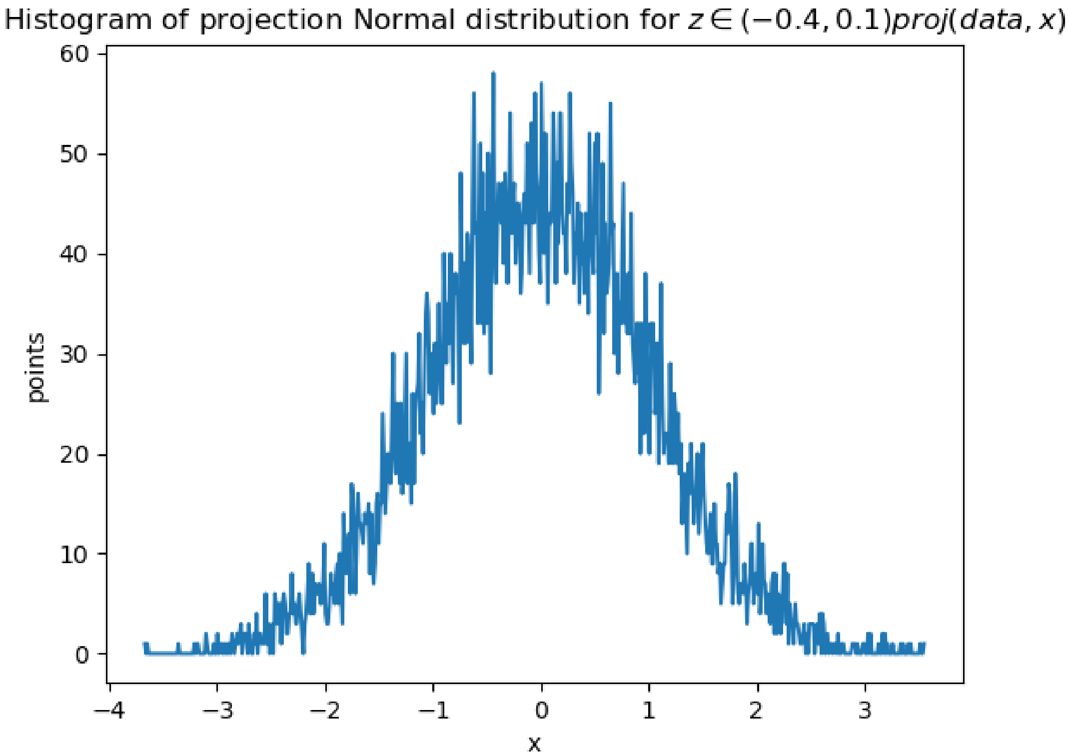


היסטוגרמת ההטלה על ציר ה y :

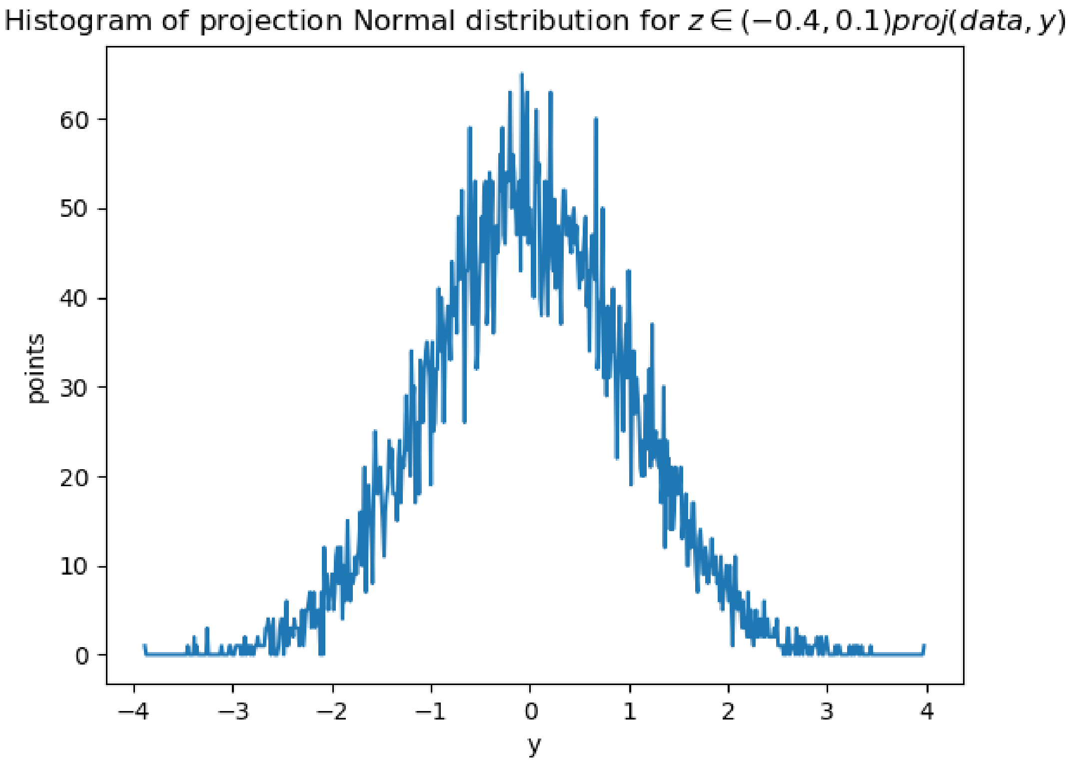


4.5 סעיף 15 :

היסטוגרמת ההטלה על ציר הx :



היסטוגרמת ההטלה על ציר ה y:

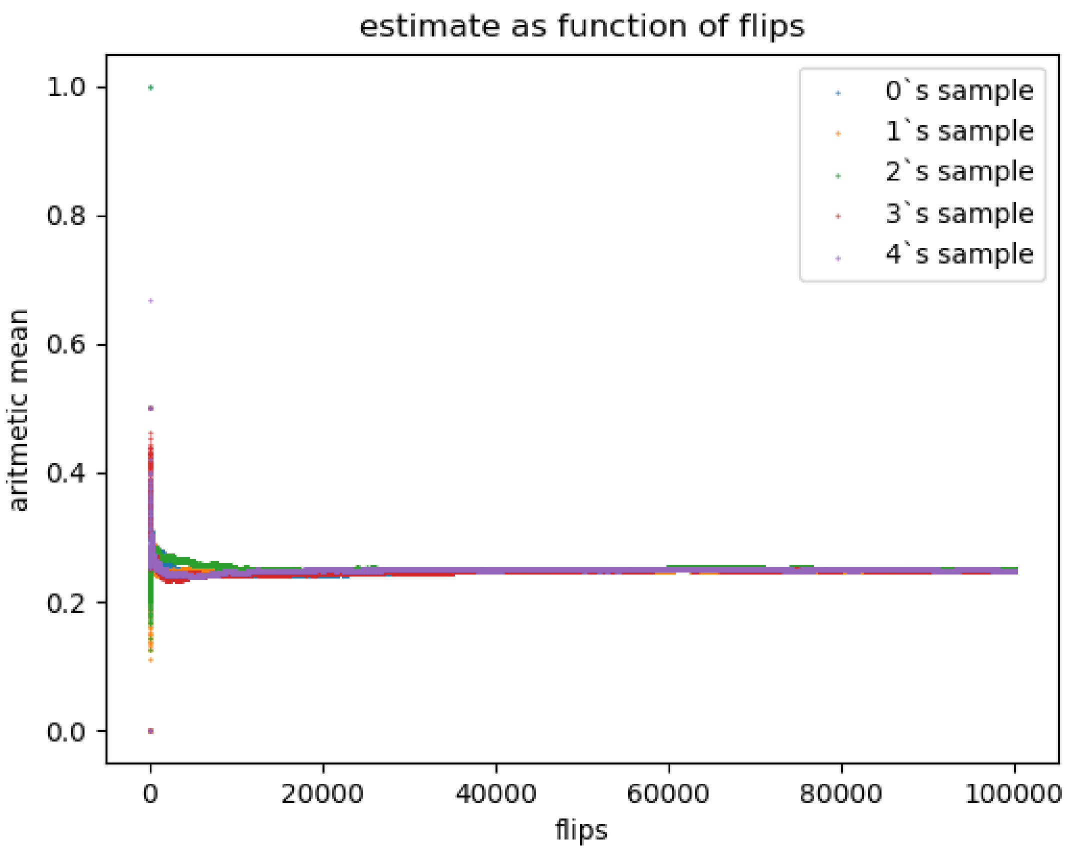


כאשר אובדן הצורה הגאוסנית, נובע מכך שהדאטה שלנו מכיל פחות נקודות כרגע, ולכן רגיש יותר לרעש.

5 חלק חמישי

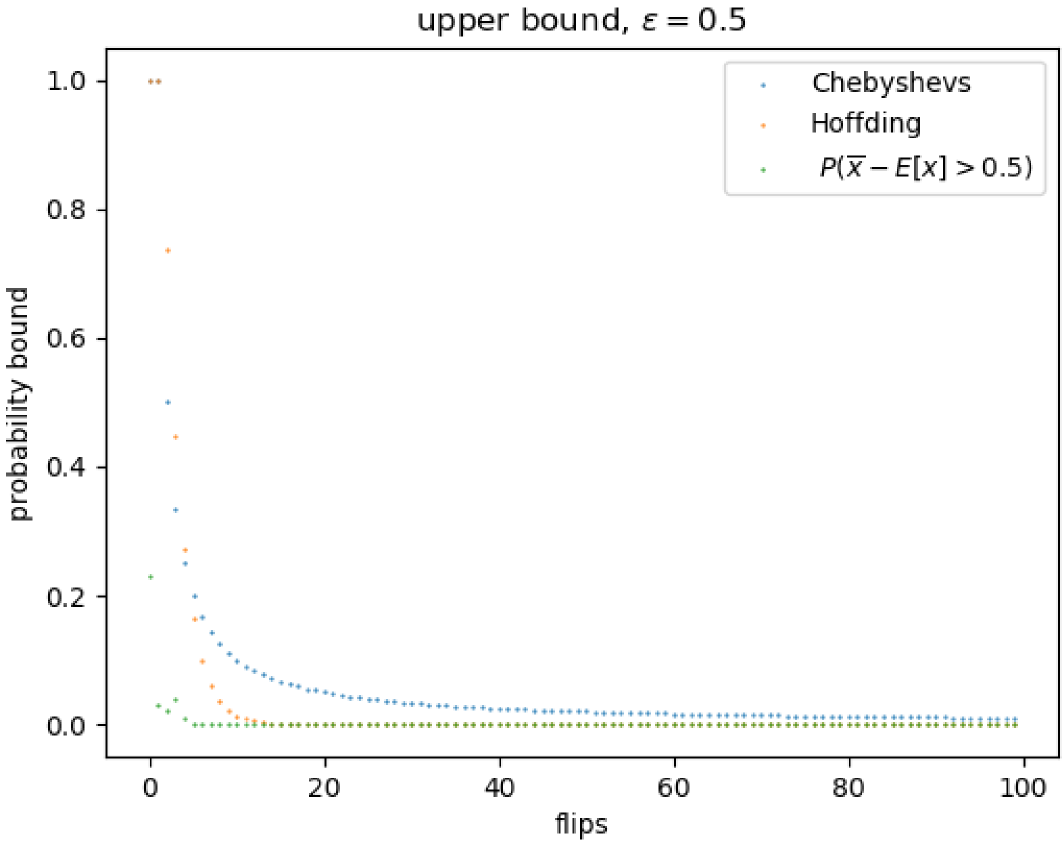
5.1 סעיף 16 :

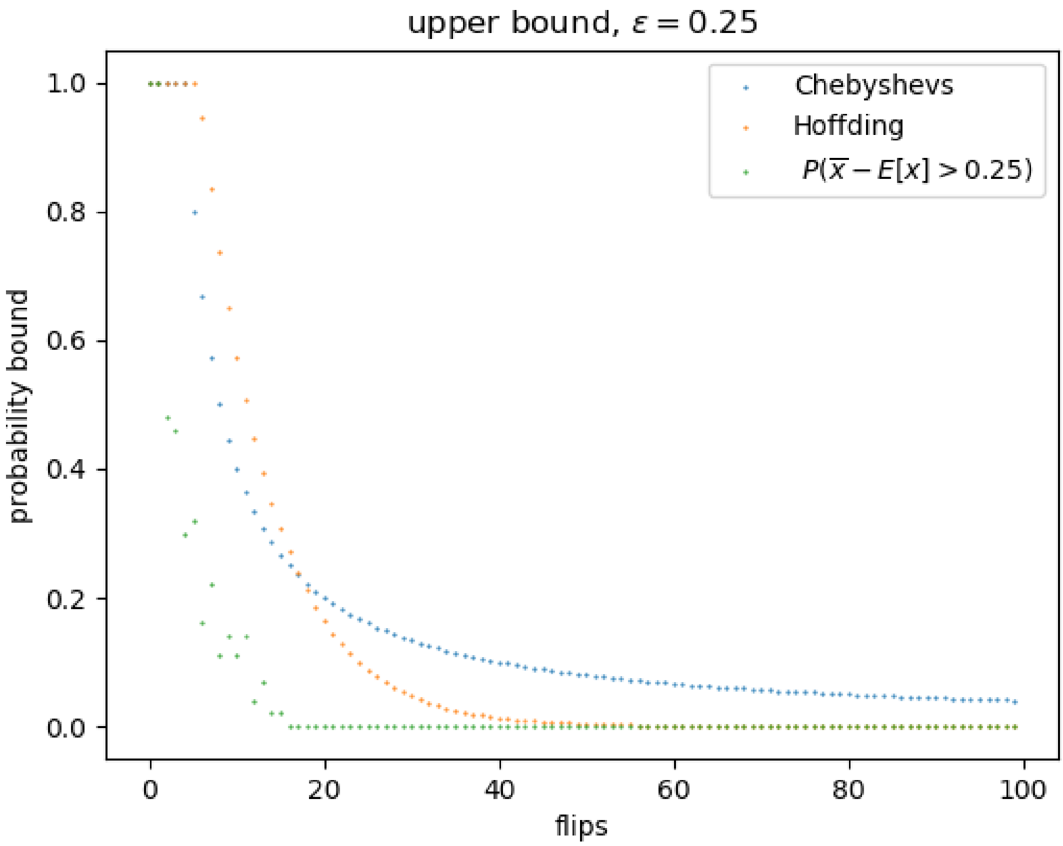
הממוצע הארמטי כפונקציה של מספר ההטלות עבור 5 הניסוים הראשונים, מאחר וההטלות מתפלגות $i.d.d$ ניתן לראות התכנסות לתוחלת של מטבע יחיד :

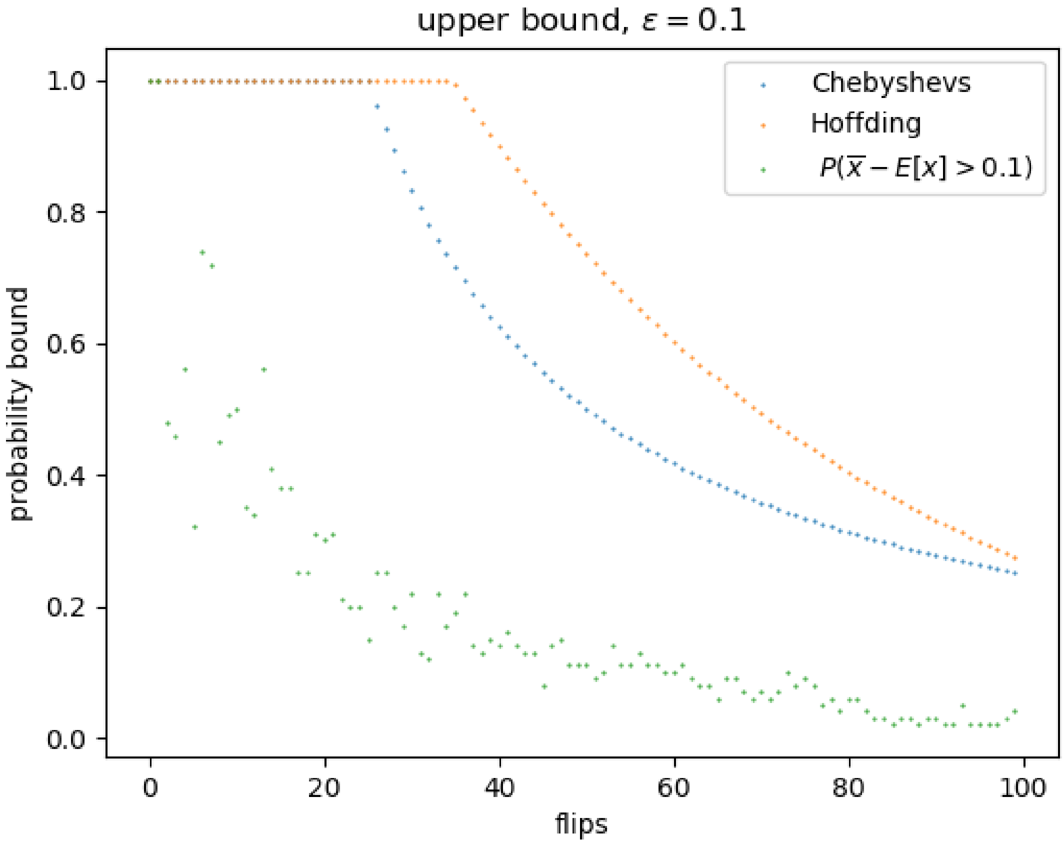


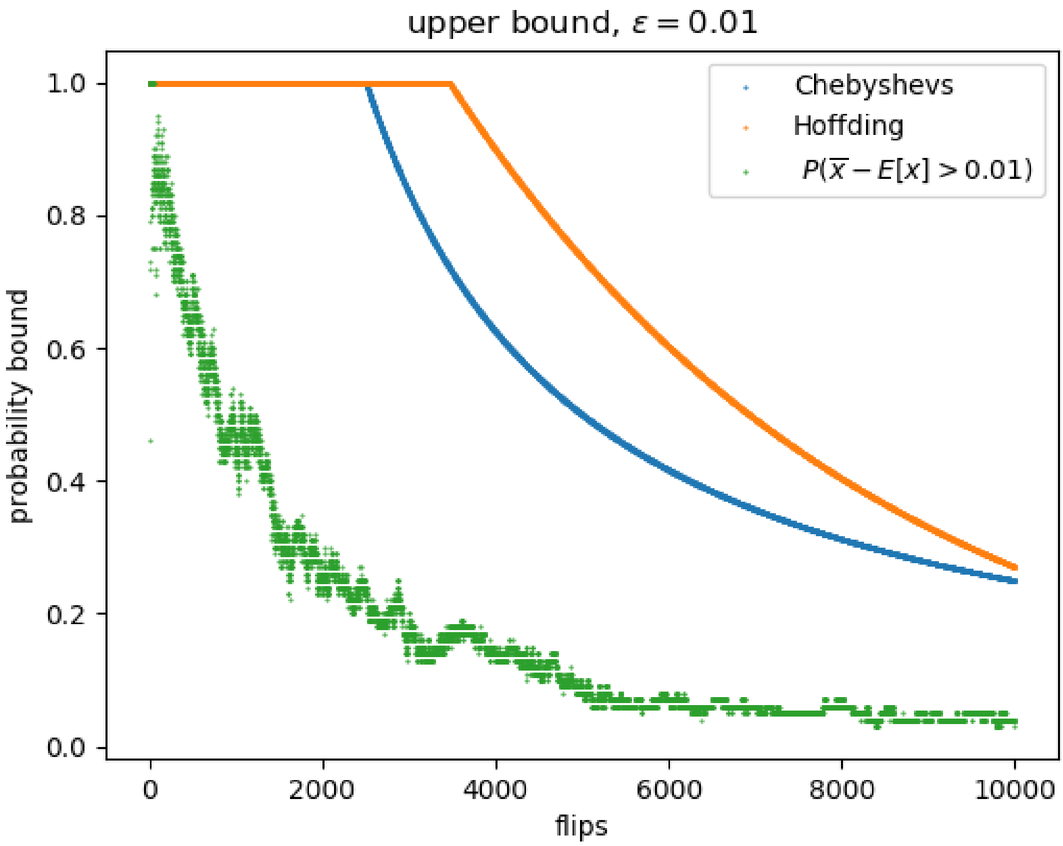
5.2 16 סעיפים ב ו ג:

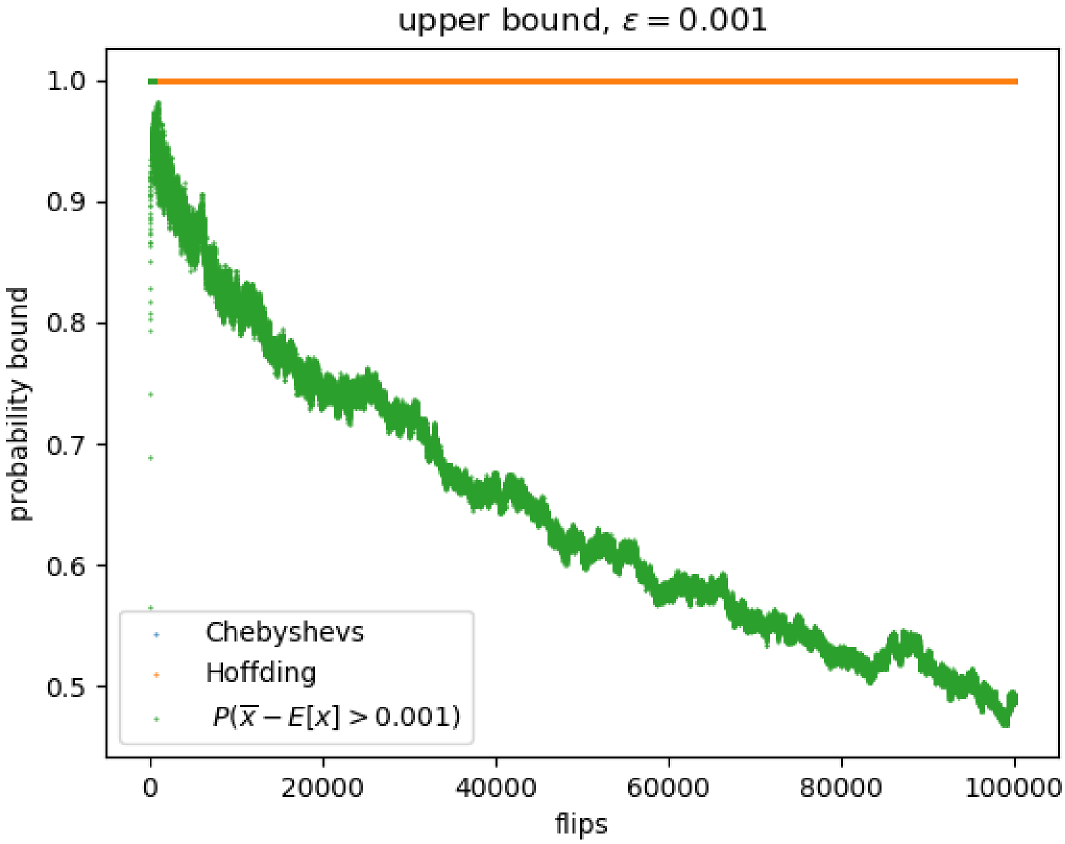
נישתמש בתוצאה שעבור מטבע יחיד $p\left(1-p\right)\leq\frac{1}{4}$ ולכן $\mathcal{D}^{m}\left[\left|p-\hat{p}\right|\geq\varepsilon\right]\leq\frac{1}{4m\varepsilon^{2}}$ וכמובן שהופנדינג מקיים : $\mathcal{D}^{m}\left[\left|p-\hat{p}\right|\geq\varepsilon\right]\leq2e^{-2m\varepsilon^{2}}$ עבור ה εים השונים נצייר את החסם, ונשווה אל ההפרש האמיתי בין הממצוע האריטמי לבין התוחלת הצפויה :


upper bound, $\varepsilon = 0.5$
Chebyshevs
Hoffding
$P(\overline{x} - E[x] > 0.5)$
probability bound
flips
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100


upper bound, $\varepsilon = 0.25$
Chebyshevs
Hoffding
$P(\overline{x} - E[x] > 0.25)$
probability bound
flips
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100


upper bound, $\varepsilon = 0.1$
Chebyshevs
Hoffding
$P(\overline{x} - E[x] > 0.1)$
probability bound
flips
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100


upper bound, $\varepsilon = 0.01$
Chebyshevs
Hoffding
$P(\overline{x} - E[x] > 0.01)$
probability bound
flips
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000


upper bound, $\varepsilon = 0.001$
probability bound
1.0
0.9
0.8
0.7
0.6
0.5
0
20000
40000
60000
80000
100000
flips
Chebyshevs
Hoffding
$P(\overline{x} - E[x] > 0.001)$