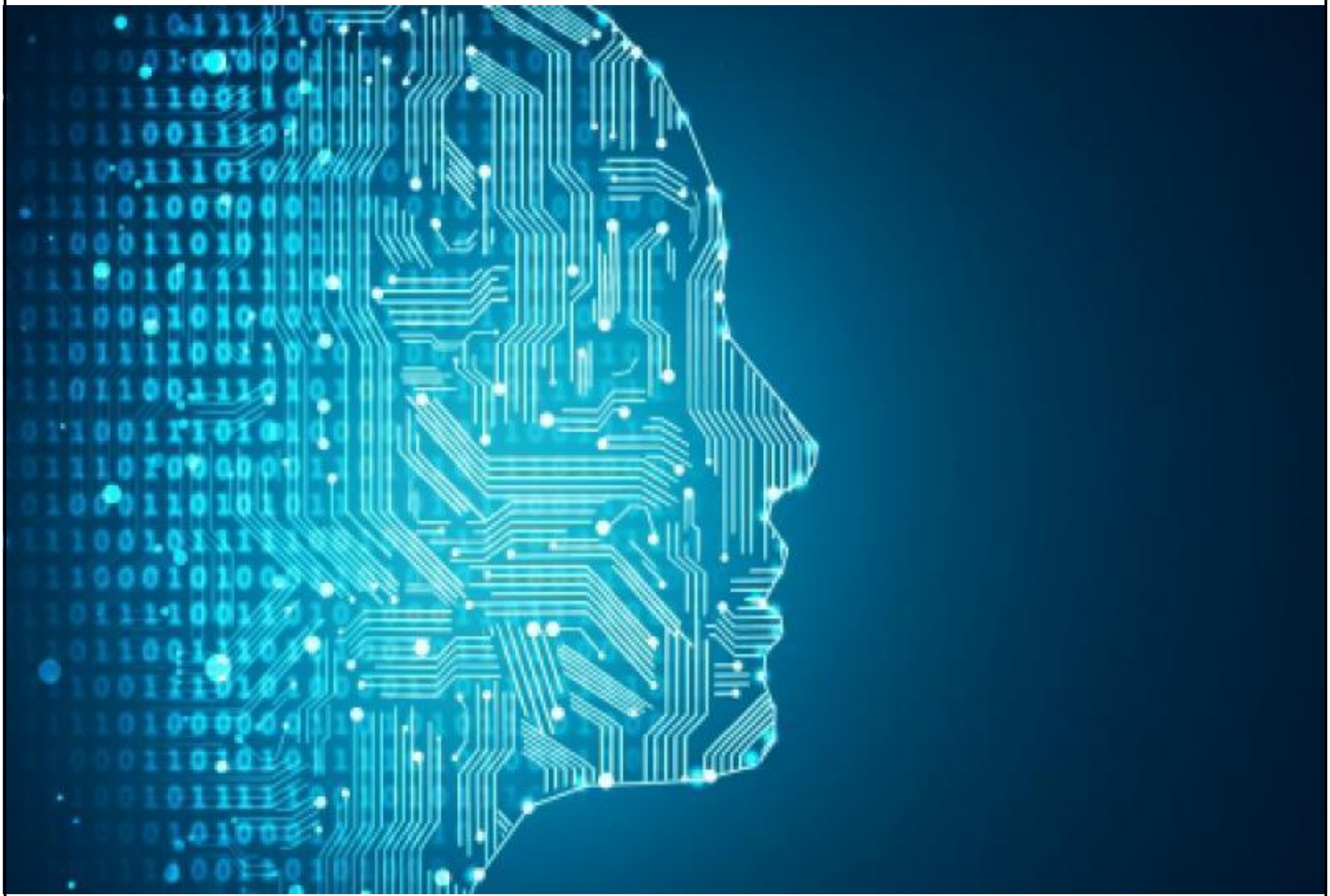


Αναγνώριση Προτύπων

Εργασία 1

Αλβανάκη Παρασκευή

ΑΜ 57286

ΕΡΩΤΗΜΑ 1

Θεωρήστε ένα ζάρι Z1 και z1 μια διακριτή τυχαία μεταβλητή που συμβολίζει το αποτέλεσμα της ρίψης του Z1.

α) Υπολογίστε τη συνάρτηση πυκνότητας πιθανότητας του f(z1) = p(Z=z1).

β) Υπολογίστε την μέση τιμή και τη μεταβλητότητα της τυχαίας μεταβλητής z1.

γ) Υπολογίστε τη skewness και την kurtosis της τυχαίας μεταβλητής z1.

ΑΠΑΝΤΗΣΗ 1

1α

f(z1) = p(Z=z1) = $\frac{1}{6}$

1β

Mean: E[X]=$\sum_k x_k * P(x_k)$= 1*$\frac{1}{6}$ + 2*$\frac{1}{6}$ + 3*$\frac{1}{6}$ +4*$\frac{1}{6}$ + 5*$\frac{1}{6}$ + 6*$\frac{1}{6}$ =3,5

Variance: Var[x]=σ^2= E[$(X-E[X])^2$]=$\frac{(1-3.5)^2+(2-3.5)^2+(3-3.5)^2+(4-3.5)^2+(5-3.5)^2+(6-3.5)^2}{6}$=$\frac{17,5}{6}$

1γ

skewness: E[$\left(\frac{x-\mu}{\sigma}\right)^3$]=$\frac{(1-3.5)^3+(2-3.5)^3+(3-3.5)^3+(4-3.5)^3+(5-3.5)^3+(6-3.5)^3}{\sigma^3} = 0$

kurtosis: E[$\left(\frac{x-\mu}{\sigma}\right)^4$]=$\frac{(1-3.5)^4+(2-3.5)^4+(3-3.5)^4+(4-3.5)^4+(5-3.5)^4+(6-3.5)^4}{\sigma^4}$ =1,73142898

ΕΡΩΤΗΜΑ 2

Το MATLAB μπορεί να δημιουργεί τυχαίες πραγματικές μεταβλητές στο διάστημα [0,1] με τη βοήθεια της εντολής rand(M,N), όπου M, N το μέγεθος του πίνακα που παράγει.
α) Χρησιμοποιήστε την παραπάνω εντολή για να φτιάξετε ένα διάνυσμα 1 x N που να έχει N τυχαίες ρίψεις του ζαριού Z1.

β) Απεικονίστε την προσεγγιστική συνάρτηση πυκνότητας πιθανότητας (ιστόγραμμα) των ρίψεων του ζαριού για N=20, 100 και 1000. Ποια κατανομή φαίνεται να ακολουθεί.
γ) Χρησιμοποιήστε τις αντίστοιχες εντολές του MATLAB για να υπολογίσετε τη μέση τιμή, μεταβλητότητα, skewness και την kurtosis των N ρίψεων για τιμές του N=10, 20, 50, 100, 500 και 1000.

δ) Για ποιες τιμές πλήθους ρίψεων φαίνονται οι πειραματικές τιμές να προσεγγίζουν τις θεωρητικές που έχετε υπολογίσει στο 1.

ε) Για ποια τιμή πλήθους ρίψεων θα λέγαμε ότι ξεκινάμε να έχουμε wide-sense stationarity ?

ΑΠΑΝΤΗΣΗ 2

2α

```
N=20;
dice_throws=ceil(6*rand(1,N));
disp(dice_throws);
```

Άρα δημιουργείται το array:
[4 1 5 3 5 5 6 5 1 6 5 4 6 2 2 6 4 2 3 2]

2β

Η δημιουργία του ιστογράμματος μπορεί να γίνει είτε manually (κώδικα σε σχόλια ως VERSION 1) είτε με τη βοήθεια του 'Normalization pdf' που έχει build in η histogram(). Έχει υλοποιηθεί και με τους 2 τρόπους στον κώδικα που παρατίθεται παρακάτω. Επιπλέον για την αναπαράσταση των ιστογραμμάτων έχει χρησιμοποιηθεί η συνάρτηση stem καθώς έχουμε διακριτές τιμές και είναι πιο ακριβής η αναπαράσταση. Παρ’ όλα αυτά για πιο visual απεικόνιση έχει χρησιμοποιηθεί και η συνάρτηση histogram plot η οποία παραθέτει σε bars τα αποτελέσματα.

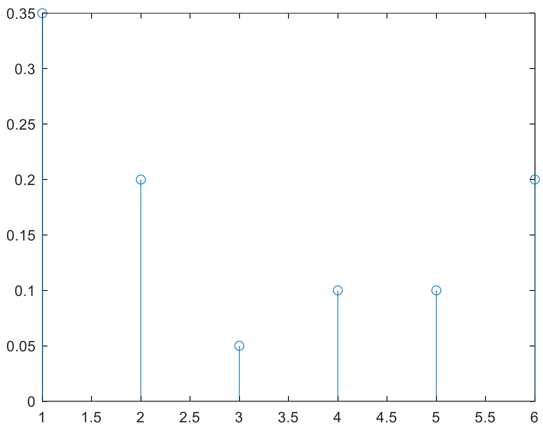
Ακολουθείται η **ομοιόμορφη** κατανομή.

```
%manually (VERSION 1)
rep_N=[20,100,1000];
for i=1:length(rep_N)
    dice_throws=ceil(6*rand(1,rep_N(i)));
    figure
    h1=histogram(dice_throws);
    yt=get(gca,'YTick');
    set(gca,'YTick',yt,'YTickLabel',yt/N(i));
    values1=h1.Values;
    values1=values1./N(i);
    disp(values1)
end

%2b version 2
%or with the help of the build in function
rep_N=[20,100,1000];
disp("Οι pdf τιμες για N=20, N=100, N=1000 για καθε τιμή ζαριού
είναι:");
for i=1:length(rep_N)
    dice_throws=ceil(6*rand(1,rep_N(i)));
    figure('Name',sprintf('For %.0f bars', i),'NumberTitle','off');
    hold on;

    h = histogram(dice_throws,'Normalization','pdf');
```

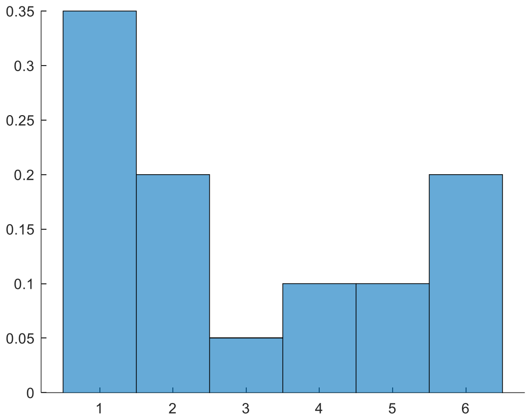
```
    values=h.Values;

    figure('Name',sprintf('For %.0f stem', i),'NumberTitle','off');
    h2 = stem(1:6,h.Values);
    values=h.Values;

    disp(values)
end
```

Για **N=20** έχουμε τα ιστογράμματα:

Απεικόνιση με stem (διακριτές τιμές)

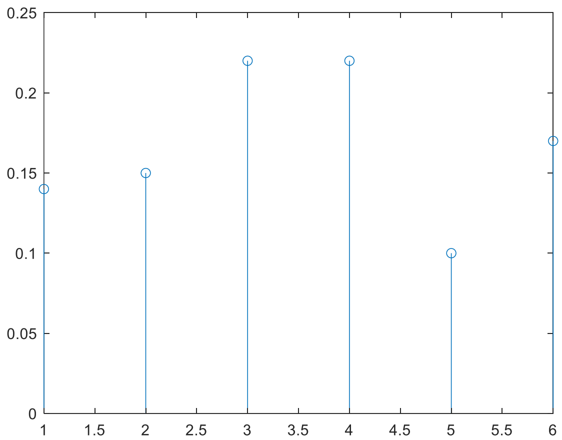


Απεικόνιση με bars

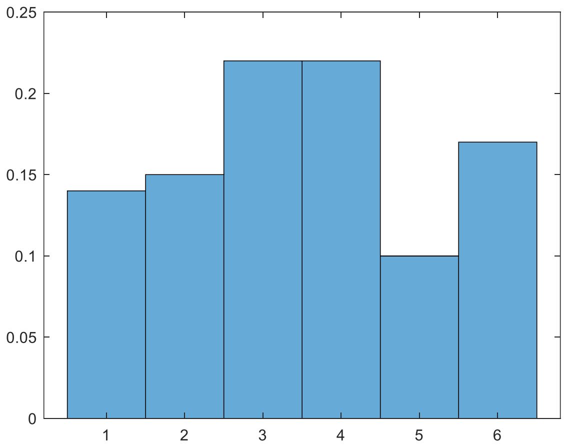


Για **N=100** έχουμε τα ιστογράμματα:

Απεικόνιση με stem (διακριτές τιμές)

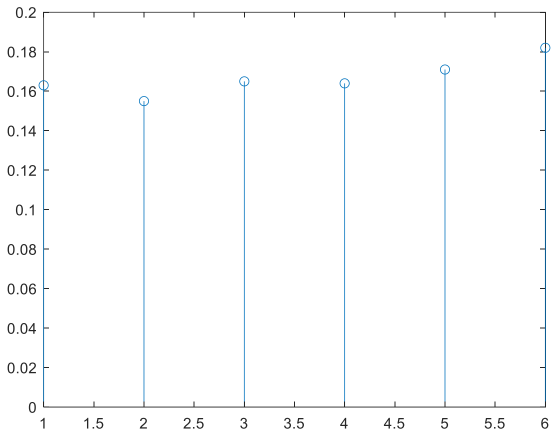


Απεικόνιση με bars

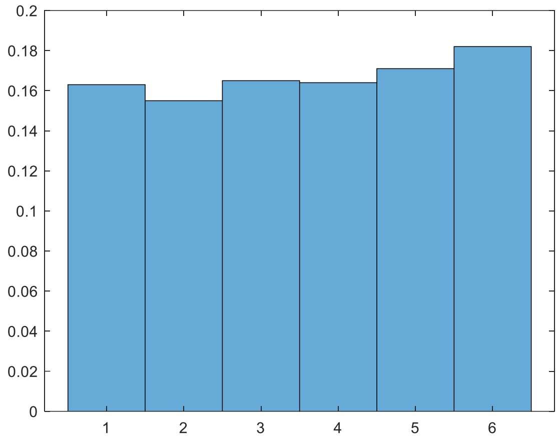


Για **N=1000** έχουμε τα ιστογράμματα:

Απεικόνιση με stem (διακριτές τιμές)



Απεικόνιση με bars



2γ

Ο κώδικας με τον οποίο υλοποιείται το ερώτημα αυτό είναι ο ακόλουθος:

```
%2c
rep_times=[10, 20, 50, 100, 500,1000];
all_values=zeros(length(rep_times),4);
disp("Ο πίνακας με τα mean, var, skewness και kurtosis είναι ο
εξής:");
for k=1:length(rep_times)

    dice_throws=ceil(6*rand(1,rep_times(k)));
    all_values(k,1)=mean(dice_throws);
    all_values(k,2)=var(dice_throws);
    all_values(k,3)= skewness(dice_throws);
    all_values(k,4)=kurtosis(dice_throws);
end
disp(all_values);
```

Έτσι δημιουργούμε τον ακόλουθο πίνακα με γραμμές τα N(rep_times) δηλαδή το πόσες φορές ρίχνουμε τα ζάρια και στήλες τα mean, var, skewness & kurtosis.

```
Ο πίνακας με τα mean, var, skewness και kurtosis ειναι ο εξης:
   3.500000000000000   3.611111111111111   0.358421665253225   1.624852071005917
   3.900000000000000   3.042105263157895  -0.332179930795847   1.685049269046107
   3.720000000000000   2.858775510204083  -0.039388288262197   1.609016761573135
   3.200000000000000   3.151515151515150   0.228632862663813   1.734220907297832
   3.544000000000000   2.953971943887778  -0.045088591499918   1.717901301082966
   3.497000000000000   2.994985985985995  -0.014070920202889   1.700228699787688
```

2δ

Παρατηρούμε πως αρχίζει να προσεγγίζει τις θεωρητικές τιμες στο N=1000

2ε

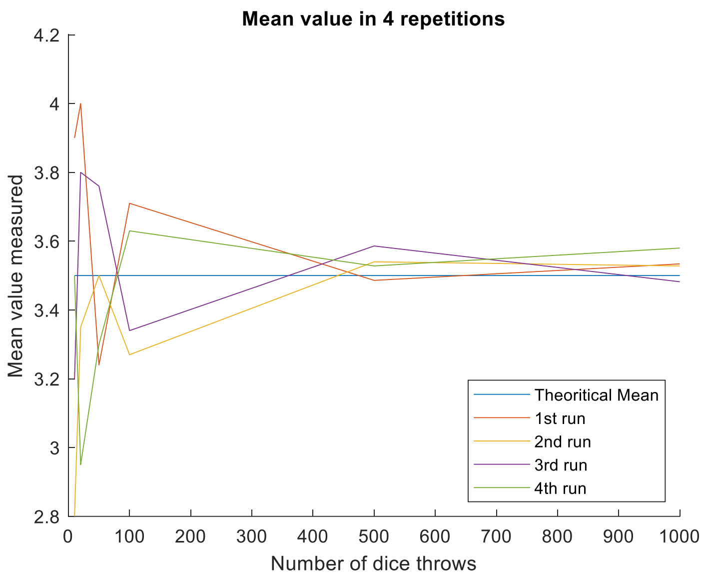
Αρχικά για αυτό το ερώτημα θεώρησα πως πρέπει να βρούμε γενικά που έχουμε wide-sense stationarity ωστόσο μετά στο μάθημα αναφέρθηκε πως μας ενδιαφέρει μόνο για τις τιμές που μας έχουν δοθεί. Για τις τιμές που μας έχουν δοθεί δεν έχουμε wide-sense stationarity σε επαρκή βαθμό. Ακόμα και στις μεγάλες τιμες του N=1000 έχουμε ακόμα αποκλίσεις καθώς τα στατιστικά στοιχεία του σχήματος εξακολουθούν να αλλάζουν. Για την μέση τιμή (1ης τάξης) παρατηρώ ότι αρχίζουν να συγκλίνουν για μικρότερες τιμές του N. Για το 2ης τάξεις variance χρειαζόμαστε μεγαλύτερο αριθμό σημείων (mean,variance,skewness,kurtosis παρατίθενται γραφικά).Σε οποιαδήποτε περίπτωση χρειαζόμαστε N τάξης μεγέθους 10^4 και συγκεκριμένα γύρω στο 10000 για να έχουμε wide-sense stationarity με μία χαμηλή διακύμανση.

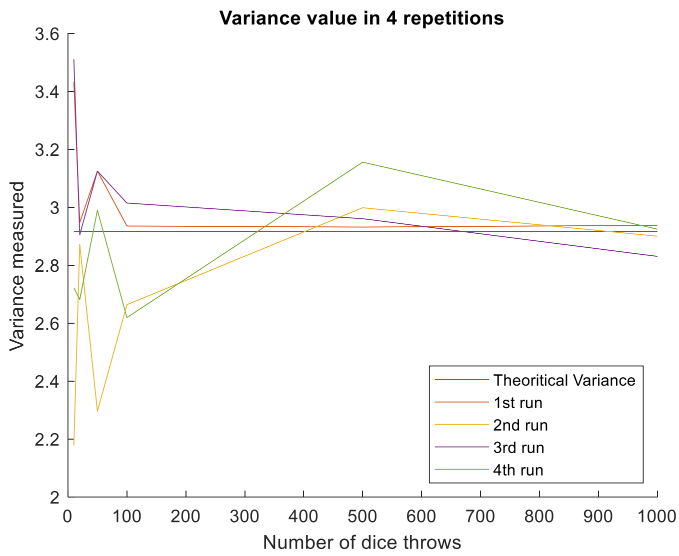
Ο κώδικας για wide-sense stationarity πρέπει να δοκιμάζει διάφορα N και η διαφορά μεταξύ των τιμών είναι μικρότερη από μια τιμή την κρατάει. Στη συνέχεια για να εξακριβώσει πως δεν πρόκειται για «τυχαία» τιμή(λόγω rand) τρέχει τον αλγόριθμο 4 φορές και αν το αποτέλεσμα εξακολουθεί να είναι επαρκές το κρατάει. Τα στοιχεία που έχω παραθέσει στο παράδειγμα είναι ενδεικτικά απλά για να δούμε πως μεταβάλει η τυχαιότητα τα στατιστικά ανάλογα με το N ωστόσο αν κάποιος το τρέξει για μεγάλες τιμες θα δει ότι οι καμπύλες ταυτίζονται.

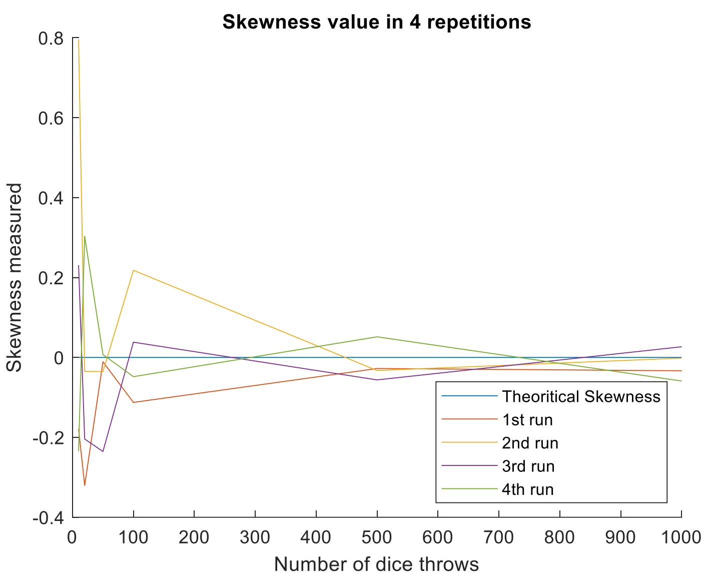
Ο κώδικας είναι ο ακόλουθος:

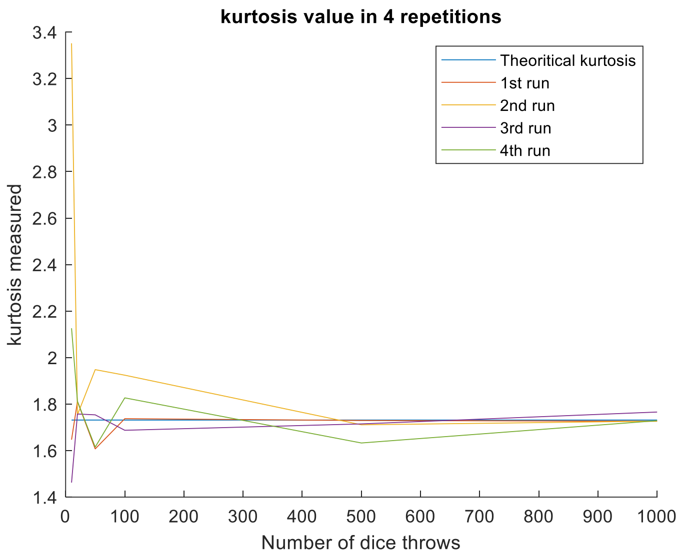
```
%2e
rep_times2=[10, 20, 100, 500,1000,4000];
mean_values=zeros(4,length(rep_times2));
var_values=zeros(4,length(rep_times2));
skewness_values=zeros(4,length(rep_times2));
kurtosis_values=zeros(4,length(rep_times2));

for j=1:4

    for k=1:length(rep_times2)
        dice_throws=ceil(6*rand(1,rep_times2(k)));
        mean_values(j,k)=mean(dice_throws);
        var_values(j,k)=var(dice_throws);
        skewness_values(j,k)= skewness(dice_throws);
        kurtosis_values(j,k)=kurtosis(dice_throws);
    end
end
disp("Ο πίνακας με τα mean(γραμμες οι επαναλήψεις με τα ιδια N
στήλες τα διαφορετικα N)");
disp(mean_values);
disp("Ο πίνακας με τα var(γραμμες οι επαναλήψεις με τα ιδια N
στήλες τα διαφορετικα N)");
disp(var_values);
disp("Ο πίνακας με τα skewness(γραμμες οι επαναλήψεις με τα ιδια N
στήλες τα διαφορετικα N)");
disp(skewness_values);
disp("Ο πίνακας με τα kurtosis(γραμμες οι επαναλήψεις με τα ιδια N
στήλες τα διαφορετικα N)");
disp(kurtosis_values);
```

```
figure('Name','Mean value in 4 repetitions
plot','NumberTitle','off');
hold on;
plot(rep_times2,[3.5 3.5 3.5 3.5 3.5 3.5]);
plot(rep_times2, mean_values);
legend({'Theoritical Mean','1st run','2nd run','3rd run','4th
run',},'Location','southeast');
title('Mean value in 4 repetitions');
xlabel('Number of dice throws');
ylabel('Mean value measured');
hold off;

figure('Name','Variance in 4 repetitions
plot','NumberTitle','off');
hold on;
plot(rep_times2,[2.916666667 2.916666667 2.916666667 2.916666667
2.916666667 2.916666667]);
plot(rep_times2, var_values);
legend({'Theoritical Variance','1st run','2nd run','3rd run','4th
run',},'Location','southeast');
title('Variance value in 4 repetitions');
xlabel('Number of dice throws');
ylabel('Variance measured');
hold off;

figure('Name','Skewness in 4 repetitions
plot','NumberTitle','off');
hold on;
plot(rep_times2,[0 0 0 0 0 0]);
plot(rep_times2, skewness_values);
legend({'Theoritical Skewness','1st run','2nd run','3rd run','4th
run',},'Location','southeast');
title('Skewness value in 4 repetitions');
xlabel('Number of dice throws');
ylabel('Skewness measured');
hold off;

figure('Name','kurtosis in 4 repetitions
plot','NumberTitle','off');
hold on;
plot(rep_times2,[1.73142898 1.73142898 1.73142898 1.73142898
1.73142898 1.73142898]);
plot(rep_times2, kurtosis_values);
legend({'Theoritical kurtosis','1st run','2nd run','3rd run','4th
run',},'Location','northeast');
title('kurtosis value in 4 repetitions');
xlabel('Number of dice throws');
ylabel('kurtosis measured');
hold off;
```









ΕΡΩΤΗΜΑ 3

Χρησιμοποιείστε τον παραπάνω κώδικα για να παράγετε 1000 ρίψεις του ζαριού Z1 και άλλες ρίψεις του ζαριού Z2.

α) Υπολογίστε και απεικονίστε την κοινή συνάρτηση πυκνότητας πιθανότητας (joint pdf) f(z1,z2).

β) Ορίστε μια νέα τυχαία μεταβλητή y ως το άθροισμα των τιμών των 2 ζαριών y = z1+z2. Απεικονίστε την συνάρτηση πυκνότητας πιθανότητας της νέας μεταβλητής. Μπορείτε να εξηγήσετε το αποτέλεσμα ; (Αναζητήστε την ιδιότητα των pdf ότι αν y = z1+z2, τότε f(y)= f(z1)*f(z2), όπου * ο τελεστής της συνέλιξης)

ΑΠΑΝΤΗΣΗ 3

3α

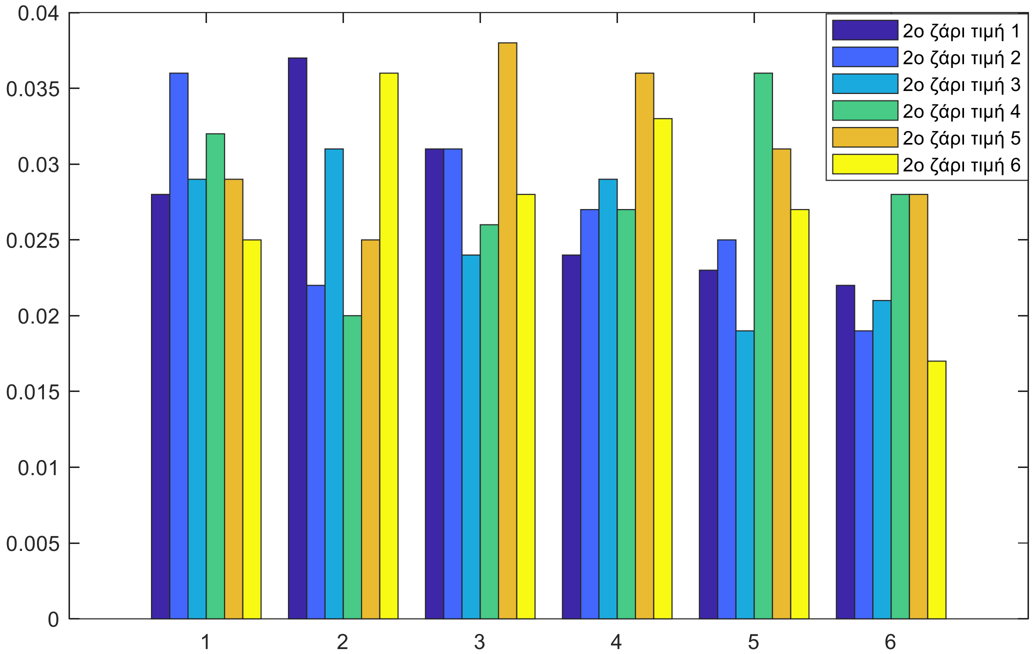
Στο ζητούμενο αυτό έχει δημιουργηθεί ένας 6x6 πίνακας με κάθε κελί να συμβολίζει μια κατάσταση(1 ρίψη και των 2 ζαριών πχ.1,2) με τις γραμμές να συμβολίζουν τις τιμές του 1ου ζαριού και τις στήλες του 2ου.

```
%3a
dice_throws=zeros(6,6);
x=ceil(6*rand(N,1));
y=ceil(6*rand(N,1));

for i=1:N
   dice_throws(x(i),y(i))=dice_throws(x(i),y(i))+1;
end
prob_throws=dice_throws/N;
disp(prob_throws);
figure
bar(prob_throws,'hist');
```

Το αποτέλεσμα είναι ο ακόλουθος πίνακας:

0.028000000000000	0.036000000000000	0.029000000000000	0.032000000000000	0.029000000000000	0.025000000000000
0.037000000000000	0.022000000000000	0.031000000000000	0.020000000000000	0.025000000000000	0.036000000000000
0.031000000000000	0.031000000000000	0.024000000000000	0.026000000000000	0.038000000000000	0.028000000000000
0.024000000000000	0.027000000000000	0.029000000000000	0.027000000000000	0.036000000000000	0.033000000000000
0.023000000000000	0.025000000000000	0.019000000000000	0.036000000000000	0.031000000000000	0.027000000000000
0.022000000000000	0.019000000000000	0.021000000000000	0.028000000000000	0.028000000000000	0.017000000000000

Στην παρακάτω απεικόνιση ο άξονας x συμβολίζει τις τιμές του 1ου ζαριού σε κάθε ρίψη και το κάθε bar με διαφορετικό χρώμα συμβολίζει τις τιμές του 2ου ζαριού στην ίδια ρίψη.

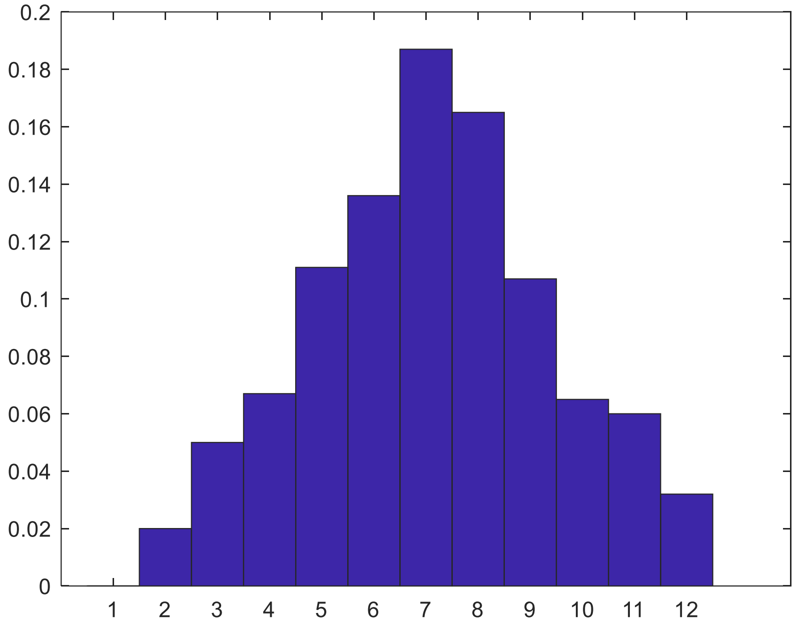


3β

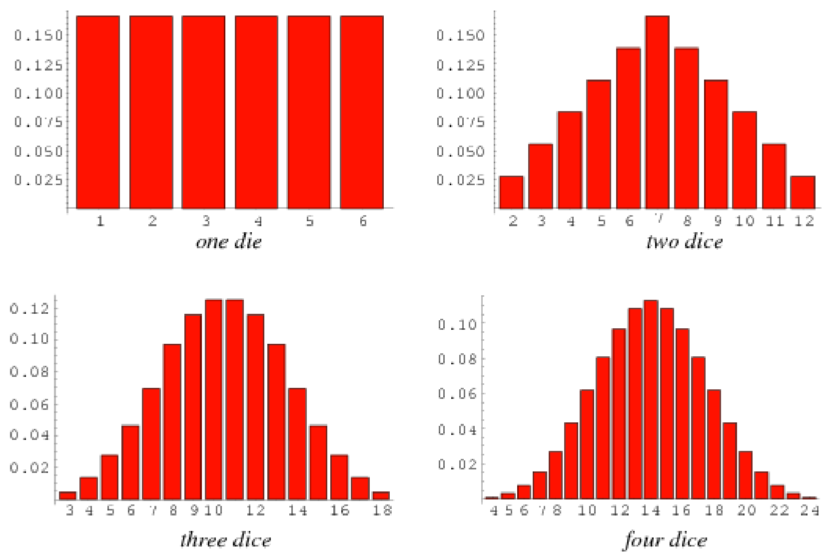
Στο ζητούμενο ουσιαστικά αθροίζουμε τις τιμές των 2 ζαριών σε κάθε ρίψη και προκύπτει ένα array διαστάσεων 1x12

```
%3b
x=ceil(6*rand(N,1));
y=ceil(6*rand(N,1));
dice_throws=zeros(1,12);
for i=1:N
    sum=x(i)+y(i);
   dice_throws(sum)=dice_throws(sum)+1;
end
prob_throws=dice_throws/N;
disp(prob_throws);
figure
bar(prob_throws,'hist');
format long
```

Η αποικόνιση της pdf είναι η εξής:



Γνωρίζουμε πως για $y = x_1 + x_2$ έχουμε $f_y(x) = f_{x1}(x)^{*} f_{x2}(x)$ Σύμφωνα με το θεώρημα κεντρικού ορίου η κατανομή του πληθυσμού των μέσων τιμών τείνει προς την κανονική κατανομή, όσο αυξάνει το μέγεθος Ν του πληθυσμιακού δείγματος. Αυτό μπορεί ναι εξηγηθεί καθώς οι κεντρικές τιμές μπορούν να δημιουργηθούν με πολλούς διαφορετικούς συνδιασμούς ζαριών(μεσαίες ζαριές μεταξύ τους, υψηλές με χαμηλές κλπ.). Η pdf τείνει σε κανονική κατανομή. Όσο αυξάνεται το πλήθος των ζαριών τόσο πιο πολύ τείνει στην κατανομή αυτή. Ενδεικτικά για μέχρι 4 ζάρια :



ΕΡΩΤΗΜΑ 4

Έστω η συνάρτηση f(x) = (x-5)4 +3x και θέλουμε να βρούμε την τιμή του x στην οποία βρίσκεται το ελάχιστο της συνάρτησης.

α) Χρησιμοποιείστε την ιδιότητα της παραγώγου για να βρεθεί απευθείας η τιμή του ελαχίστου.

β) Υλοποιείστε στο MATLAB την αριθμητική τεχνική του gradient descent για να βρείτε το ελάχιστο της συνάρτησης. Ξεκινήστε από μια τυχαία τιμή, βάλτε μια αυθαίρετη τιμή για το βήμα η και εφαρμόστε τη σχέση του gradient descent επαναληπτικά, μέχρι ότου η τιμή του ελαχίστου να μην αλλάζει. Μπορεί να χρειαστεί να δοκιμάσετε διάφορες τιμές για το η. Σε πόσες επαναλήψεις έχετε βρει το ελάχιστο.

γ) Υλοποιείστε στο MATLAB την αριθμητική τεχνική του Newton method για να βρείτε το ελάχιστο της συνάρτησης. Ξεκινήστε από μια τυχαία τιμή και εφαρμόστε τη σχέση του Newton method επαναληπτικά, μέχρι ότου η τιμή του ελαχίστου να μην αλλάζει. Σε πόσες επαναλήψεις έχετε βρει το ελάχιστο.

δ) Συγκρίνετε τις 2 μεθοδολογίες. Ποια δίνει τη γρηγορότερη σύγκλιση ?

ΑΠΑΝΤΗΣΗ 4

4α

Για το ελάχιστο έχουμε παράγωγο 0.

f''(z1) = $4 * (z1 - 5)^3 + 3$ = …=> z1= 4.091439703583929

4β

Επιλέχθηκε learning rate 0.02 αρχικό σημείο το 0 και όριο error 10^(-6).
Το σημείο προσδιορίζεται μετά από 55 επαναλήψεις.
Ο κώδικας που υλοποιήθηκε είναι ο παρακάτω:

```
%4b gradient descent
lr=0.01;
x=0;
reps=1;
new_x=x-lr*(4*(x-5)^3+3);
disp(abs(new_x-x));
while (abs(new_x-x)>(10^(-6)))
    x=new_x;
    new_x=x-lr*(4*(x-5)^3+3);
    disp(new_x);
    reps=reps+1;
end
disp(new_x);
disp(reps);
```

4γ

Επιλέχθηκε αρχικό σημείο το 0 και όριο error 10^(-6).
Το σημείο προσδιορίζεται μετά από 9 επαναλήψεις.
Ο κώδικας που υλοποιήθηκε είναι ο παρακάτω:

```
%4c

x=0;
reps=1;
new_x=x-1/(((x-5)^2)*12)*(4*(x-5)^3+3);
disp(abs(new_x-x));
while (abs(new_x-x)>(10^(-6)))
    x=new_x;
    new_x=x-1/(((x-5)^2)*12)*(4*(x-5)^3+3);
    disp(new_x);
    reps=reps+1;
end
disp(new_x);
disp(reps);
%}
```

4δ

Η μέθοδος Newton είναι φανερά πιο γρήγορη από την gradient descent καθώς ο ρυθμός μάθησης αλλάζει αυτόματα, επιταχύνει όσο είναι μακριά από το ακρότατο, επιβραδύνει όσο πλησιάζει στο ακρότατο χρησιμοποιώντας το αντίστροφο της $2^{ης}$ παραγώγου. Ωστόσο στη μέθοδος Newton έχει μειονεκτήματα όπως η πολυπλοκότητα της και το γεγονός ότι μπορεί να απειριστεί (αν $2^{η}$ παράγωγος 0 αρά 1/0=inf) επειδή γίνεται τοπικά στο σημείο προσέγγιση παραβολής της συνάρτησης.