Table of Contents

```
% Peter Purnyn & Lenny Knittle
% Section 61
% Group 9

clc; %clear command window
clear; %clear variables
close all; %close all open figures
```

Problem 1 -

Define variables, r as radius, A as area, C as circumference

```
r = 12;
A = pi*r.^2;
C = pi*2*r;

%Print solution
display('1.')
fprintf('A) %4.2f\n', A);
fprintf('B) %4.2f\n', C);
```

```
1.
A) 452.39
B) 75.40
```

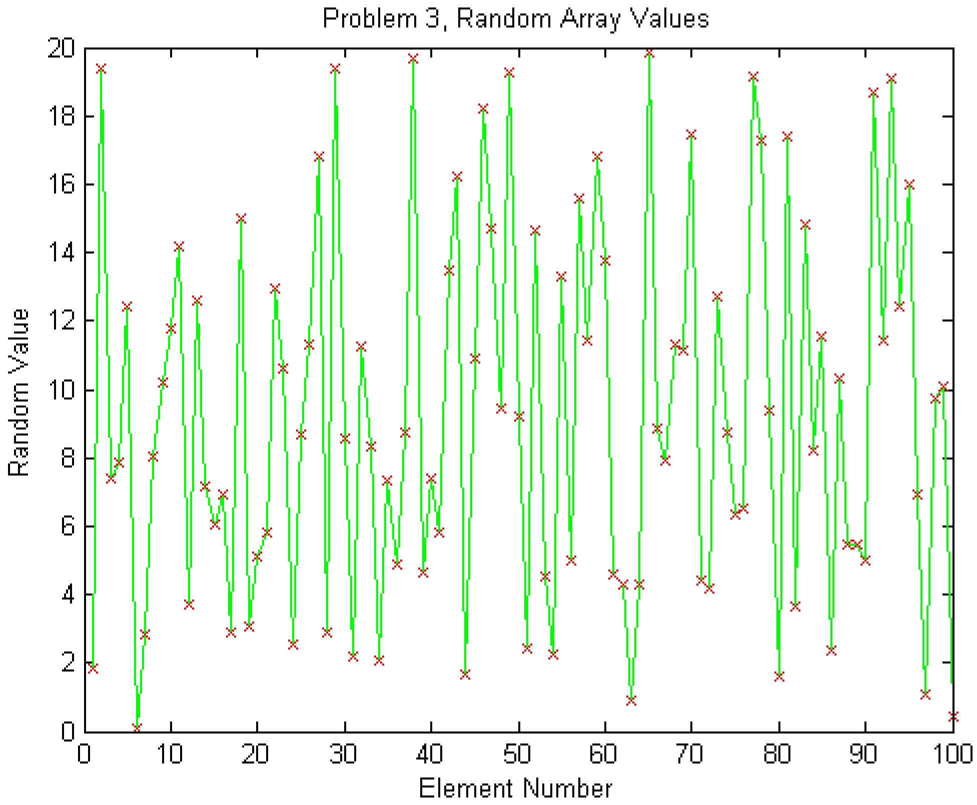
Problem 2 -

```
fprintf('\n2.\n');
odd=[1:2:19]; % create odd vector
even=(0:2:19)'; % create even vector

%A Transpose collumn to row
even1 = even'; %transpose even vector
fprintf('A) ');
fprintf('< ')
fprintf('%d ',even1)
fprintf('>\n')
%B Find size of vectors
oddsize = size(odd); %get size of odd vector
evensize = size(even1); %get size of even vector
fprintf('B) size of odd vector  = ');
fprintf('%d ',oddsize);
fprintf('\n   size of even vector = ');
fprintf('%d ',evensize);
%C Find sum of vectors
result1=odd+even1; %add odd and even vectors
fprintf('\nC) < ');
fprintf('%d ',result1);
fprintf('>');
%D Find product of vectors
result2=odd*even; %multiply vectors
fprintf('\nD) %d ',result2);
```

```
2.
A) < 0 2 4 6 8 10 12 14 16 18 >
B) size of odd vector  = 1 10
   size of even vector = 1 10
C) < 1 5 9 13 17 21 25 29 33 37 >
D) 1230
```

Problem 3 -

```
%Generate Random Numbers
Rand1 = (rand(100,1))*20;

%Create Plot
plot(Rand1,'g') %plot random numbers as line graph
title('Problem 3, Random Array Values'); %title graph
%label axis
xlabel('Element Number');
ylabel('Random Value');
hold on; %plot next graph on same figure
plot(Rand1,'rx') %plot same random numbers as red Xs
```



Problem 4 -

```
fprintf('\n\n4.');
%Define Sigma, Mu
Sigma = 50;
Mu = 0;
%Generate random numbers
Rand3 = normrnd(Mu,Sigma,[1,1000]);
%Pick random number from aray
RandNum = Rand3(floor(1000*rand(1)));
%Initialize variables
greaterCount=0;
lessThanCount=0;
equalsCount=0;
% iterate through random numbers and count
for (i=1:1000)
    if(Rand3(i)>50) % is this # greater than 50?
        greaterCount=greaterCount+1;
    elseif(Rand3(i)<50) %is this # greater less than 50?
            lessThanCount=lessThanCount+1;
    else %is this # = 50?
            equalsCount=equalsCount+1;
    end
end
% print the counts divided by total numbers
% in order to get the percentage
fprintf('\nA) %g',greaterCount/1000);
fprintf('%%');
fprintf('\nB) %g',lessThanCount/1000);
fprintf('%%');
fprintf('\nC) %g',equalsCount/1000);
fprintf('%%');
```

```
4.
A) 0.198%
B) 0.802%
C) 0%
```

Problem 5

```
fprintf('\n\n5.');
%Generate random numbers
Rand4 = rand(1,1000)*500;
%Pick random number from aray
RandNum = Rand4(floor(1000*rand(1)));
%Initialize Variables
greaterCount1=0;
lessThanCount1=0;
equalsOrGreaterCount=0;
% iterate through all the random #s
for (i=1:1000)
    if(Rand4(i)>150)
        %is this # greater than 150?
        greaterCount1=greaterCount1+1;
        equalsOrGreaterCount=equalsOrGreaterCount+1;
    elseif(Rand4(i)==150)
        %is this # = 150
        equalsOrGreaterCount=equalsOrGreaterCount+1;
    elseif(Rand4(i)<150 && Rand4(i)>=120)
        % is this # less than 150 and greater than or equal to 120?
        lessThanCount1=lessThanCount1+1;
    end
end
% print the counts divided by total numbers
% in order to get the percentage
fprintf('\nA) %g',greaterCount1/1000);
fprintf('%%');
fprintf('\nB) %g',lessThanCount1/1000);
fprintf('%%');
fprintf('\nC) %g',equalsOrGreaterCount/1000);
fprintf('%%');
```

```
5.
A) 0.697%
B) 0.061%
```

```
C) 0.697%
```

Problem 6 -

```
fprintf('\n\n6.');
%Define Sigma, Mu
Sigma = 25;
Mu = 0;
greaterThan25Count=0;
lessThan15Count=0;
sum=0;
mean1=0;
sum25=0;
mean25=0;
%Generate random numbers with normal distribution
Rand5 = normrnd(Mu,Sigma,[100,100]);
for i=1:100 %iterate through all the rows
    for j=1:100 %iterate through all the collumns
        if(Rand5(i,j)>25)
            %is the # > 25?
            greaterThan25Count=greaterThan25Count+1;
            %finds the sum of all numbers great than 25
            sum25=sum25+Rand5(i,j);
        elseif(Rand5(i,j)<15)
            %is the # < 15
            lessThan15Count=lessThan15Count+1;
        end
        %finds sum of all the #
        sum=sum+Rand5(i,j);

    end
end
% print the sum divided by total numbers
% in order to get the arithmetic average
mean1 = sum/10000;
% finds the arithmetic average of the numbers
% larger than 25
mean25 = sum25/greaterThan25Count;

%prints the answers
fprintf('\nA) %g',greaterThan25Count);
fprintf('\nB) %g',lessThan15Count);
fprintf('\nC) %g',sum);
fprintf('\nD) %g',mean1);
fprintf('\nE) %g',mean25);
```

```
6.
A) 1515
B) 7348
C) -6554.73
D) -0.655473
```

```
E) 38.0401
```

Problem 7

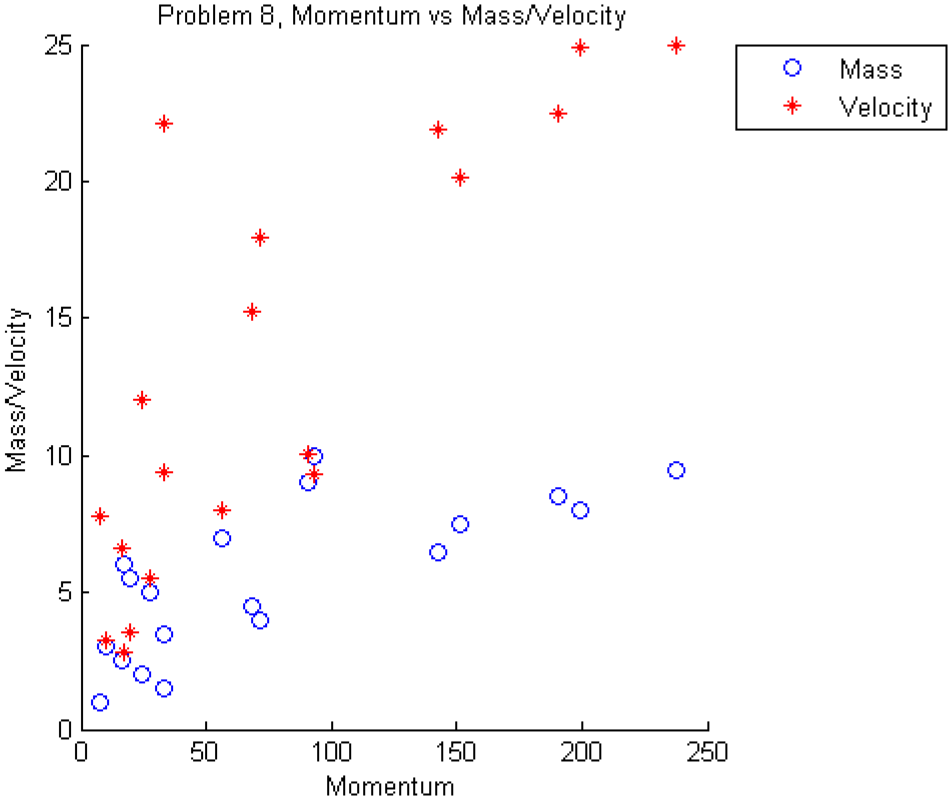
```
fprintf('\n\n7.');
%Generate random numbers with uniform distribution
Rand6 = rand(100,100)*500;
%Pick random number from array
RandNum = Rand6(floor(100*rand(1)),floor(100*rand(1)));
greaterThan150=0;
sum2=0;
mean2=0;
sumGreaterThan=0;
for i=1:100 % iterate through the rows
    for j=1:100 % iterate through the collums
        if(Rand6(i,j)>150)
            % is # > 150?
            greaterThan150=greaterThan150+1;
            if(Rand6(i,j)<130)
                % sum the numbers larger than 130
                sumGreaterThan=sumGreaterThan+Rand6(i,j);
            end
        end
        % sums all the #s
        sum2 = sum2+Rand6(i,j);
    end
end
% finds the arithmetic mean of all the #s
mean2=sum2/10000;
% print the answers
fprintf('\nA) %g',greaterThan150);
fprintf('\nB) %f',sum2);
fprintf('\nC) %g',mean2);
fprintf('\nD) %g',sumGreaterThan);
```

```
7.
A) 7012
B) 2500913.148136
C) 250.091
D) 0
```

Problem 8

```
fprintf('\n\n8.');
m=[1:.5:10]; %creates a vector from 1 to 10 with a step size of .5
fprintf('\nA) < ');
fprintf('%g ',m);
fprintf('>');
msize=size(m); % gets the size of m
fprintf('\nB) size of m vector = ');
fprintf('%d ',msize);
fprintf('\n   the vector is a row vector because the size is 1xN ');
fprintf('\nC) ');
v=25*rand(19,1); %generates a vector the same size as m, but with random numbers
fprintf('\nD) < ');
fprintf('%g ',v);
fprintf('>');
fprintf('\n    x has to be 19 so that the v vector and m vector have the same size
p=v'.*m; % multiplies the velocities and masses to get a vector of momentums
figure2=figure('name','Problem 8');
scatter(p,m); % plots a scatterplot of momentum vs masses
title('Problem 8, Momentum vs Mass/Velocity');
xlabel('Momentum');
ylabel('Mass/Velocity');
hold on;
scatter(p,v,'r*') % plots a scatterplot of momentum vs velocities
legend('Mass','Velocity','Location','BestOutside');
```

```
8.
A) < 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 >
B) size of m vector = 1 19
   the vector is a row vector because the size is 1xN
C)
D) < 7.78509 22.0954 12.0393 6.62835 3.24659 9.3849 17.9225 15.2555 5.5440
    x has to be 19 so that the v vector and m vector have the same size
```

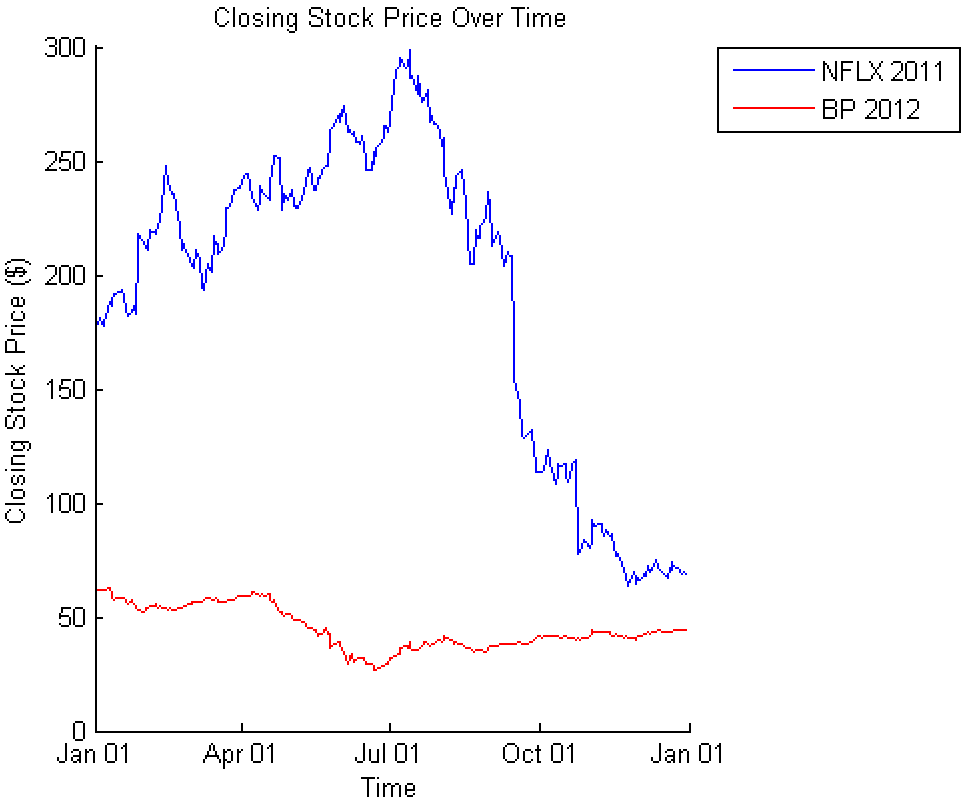


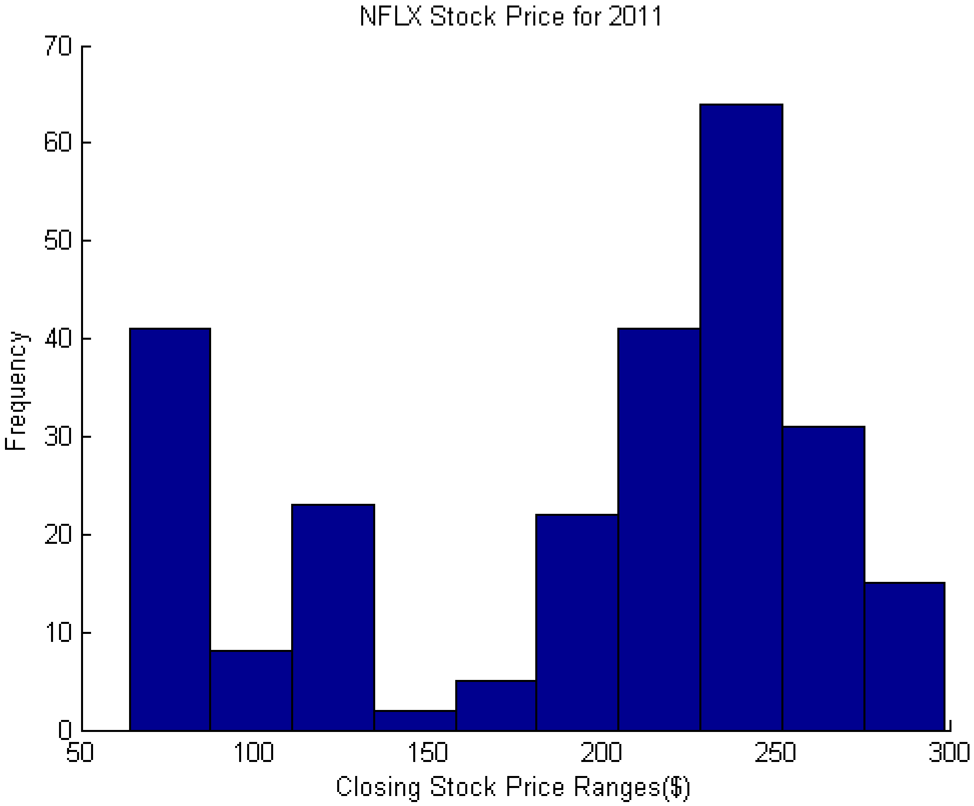
Problem 9

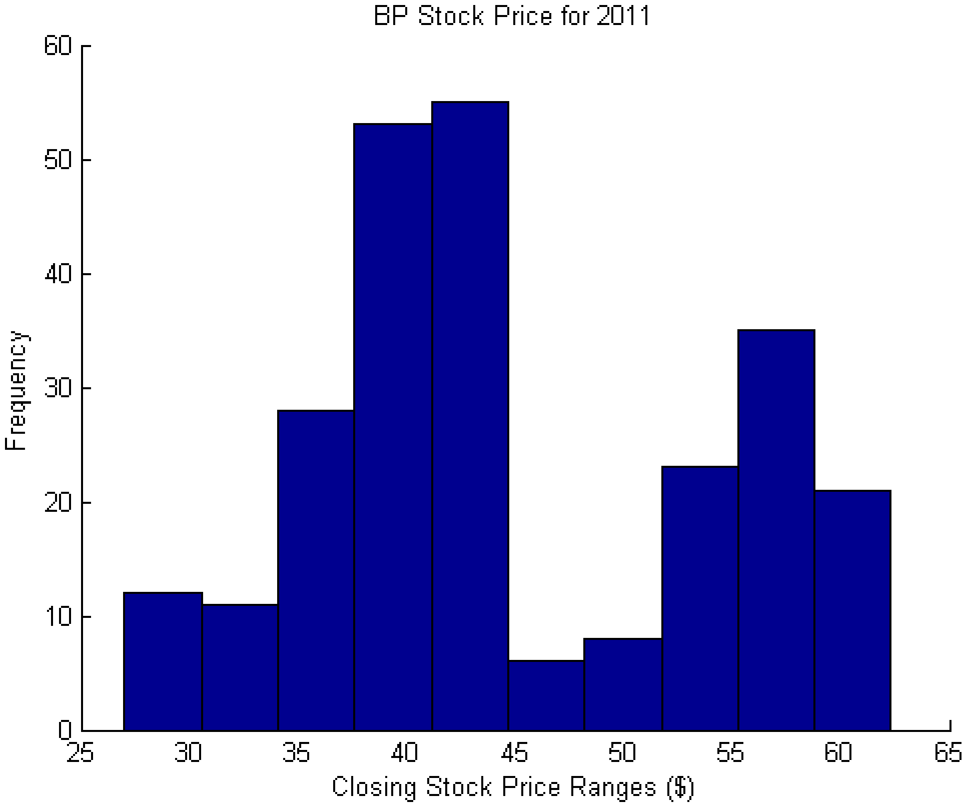
```
fprintf('\n\n9.\n');
% imports the NFLX stock xls dates column
[nflxNdata, stockDates, Nflxalldata] = xlsread('nflx_engr202.xls','A2:A253');
% imports the NFLX stock xls closing price column
nflxData = xlsread('nflx_engr202.xls','E2:E253');
% imports the BP stock xls dates column
bpData = xlsread('bp_engr202.xls','E2:E253');
%converst dates to matlab form
dates = datenum(stockDates, 'mmm dd');
%Descriptive Statistics for NFLX stock
nflxMin = min(nflxData); % finds the minimum
nflxMax = max(nflxData); % finds the maximum
nflxRange = range(nflxData); %finds the range
nflxMean = mean(nflxData); %finds the arithmetic average
%Descriptive Statistics for BP stock
bpMin = min(bpData); % finds the minimum
bpMax = max(bpData); % finds the maximum
bpRange = range(bpData); %finds the range
bpMean = mean(bpData); %finds the arithmetic average
% Prints NFLX answers
fprintf('A) NFLX stock\n');
fprintf('Min: %g\n',nflxMin);
fprintf('Max: %g\n',nflxMax);
fprintf('Range: %g\n',nflxRange);
fprintf('Mean: %g\n',nflxMean);
fprintf('B) BP Stock\n');
fprintf('Min: %g\n',bpMin);
fprintf('Max: %g\n',bpMax);
fprintf('Range: %g\n',bpRange);
fprintf('Mean: %g\n',bpMean);
%creates new figure to put NFLX and BP data on.
figure3=figure('name','Problem 9 Stocks');
hold on;
title('Closing Stock Price Over Time');
xlabel('Time');
ylabel('Closing Stock Price ($)');
plot(dates, nflxData); %plots nflx data
plot(dates, bpData,'r'); %plots bp data
datetick('x','mmm dd'); %sets the tick marks to be month and day
legend('NFLX 2011','BP 2012','Location','BestOutside'); %adds a legend
%creates new figure for NFLX histogram
figure4=figure('name','Problem 9 NFLX histogram');
hold on;
title('NFLX Stock Price for 2011');
xlabel('Closing Stock Price Ranges($)');
ylabel('Frequency');
hist(nflxData,10); %plots NFLX histogram
%creates new figure for BP histogram
figure5=figure('name','Problem 9 BP histogram');
hold on;
title('BP Stock Price for 2011');
```

```
xlabel('Closing Stock Price Ranges ($)');
ylabel('Frequency');
hist(bpData,10);%plots bp histogram
```

```
9.
A) NFLX stock
Min: 63.86
Max: 298.73
Range: 234.87
Mean: 192.452
B) BP Stock
Min: 27.02
Max: 62.32
Range: 35.3
Mean: 44.9964
```




NFLX Stock Price for 2011
Frequency
0
10
20
30
40
50
60
70
50
100
150
200
250
300
Closing Stock Price Ranges($)


BP Stock Price for 2011
Frequency
0
10
20
30
40
50
60
25
30
35
40
45
50
55
60
65
Closing Stock Price Ranges ($)

Published with MATLAB® R2013a