Problem Set 1 Solutions

Tyler Ransom

June 5, 2012

1 Main Problem Set Code

```
%%% Problem Set 1 Code
%%% Written by Tyler Ransom, Duke University
clear all;
clc;
%% Problem 1
% (a) Create the following four matrices of random numbers, setting the ...
    seed to '1234'To set the seed, type the following code: ...
    rand('seed',1234); and randn('seed',1234);. Name the matrices and set ...
    the dimensions as noted
% i.   A_{10x7}  - random numbers distributed U[-5,10]
% ii.  B_{10x7}  - random numbers distributed N(-2,15)  [st dev is 15 ]
% iii. C_{5x7}   - the first 5 rows and first 5 columns of A  and the last ...
    two columns and first 5 rows of B
% iv.  D_{10x7}  - where D_{i,j}=A_{i,j}  if A_{i,j}≤0 , or 0  otherwise
rand('seed',1234); randn('seed',1234);
A = 15*rand(10,7)-5;
% different way to do this: A = unifrnd(-5,10,10,7)

B = -2+15*randn(10,7);
% different way to do this: B = normrnd(-2,15,10,7)

C = [A(1:5,1:5) B(1:5,end-1:end)];
D = A.*(A≤0);

% (b) Use a built-in Matlab function to list the number of elements of A
numel(A);

% (c) Use a series of built-in Matlab functions to list the number of ...
    unique elements of D
numel(unique(D));

% (d) Using the reshape command, create a new matrix called E  which is ...
    the 'vec' operator applied to B . Can you find an easier way to ...
    accomplish this?
E = reshape(B,size(B,1)*size(B,2),1);
E1 = reshape(B,numel(B),1); %this is another way to do it
```

```
E2 = B(:); %this is the easiest way to do it

% (e) Create a new matrix called F  which is 3-dimensional and contains A  ...
   in the first column of the third dimension and B  in the second column ...
   of the third dimension
F = zeros(10,7,2);
F(:,:,1)=A;
F(:,:,2)=B;

% (f) Use the permute function to twist F  so that it is now ...
   F_{2\times10\times7}  instead of F_{10\times7\times2}
F=permute(F,[3 1 2]);

% (g) Create a matrix G  which is the Kronecker product of B  and C . What ...
   happens when you try C\otimes F ?
G=kron(B,C);
% kron(C,F) gives an error because the Kron product is only defined for ...
   2-D (not 3-D)

% (h) Save the matrices A , B , C , D , E , F  and G  as a .mat file named ...
   matrixpractice.
save matrixpractice A B C D E F G

% (i) Save only the matrices A , B , C , and D  as a .mat file called ...
   firstmatrix.
save firstmatrix A B C D

% (j) Export C  as a .csv file called Cmatrix.
csvwrite('Cmatrix.csv',C);

% (k) Export D  as a tab-delimited .dat file called Dmatrix.
dlmwrite('Dmatrix.dat',D,'delimiter','\t');
%% Problem 2
% (a) Write a loop that computes the element-by-element product of A and B
AB = zeros(size(A));
for i=1:size(A,1)
    for j=1:size(A,2)
        AB(i,j)=A(i,j)*B(i,j);
    end
end
AB2 = A.*B;

% (b) Write a loop that creates a column vector called Cprime which ...
   contains only the elements of C that are between -5 and 5 (inclusive).
Cprime=[]; %initialize Cprime as an empty matrix
for j=1:size(C,2)
    for i=1:size(C,1)
        if C(i,j)≥-5 && C(i,j)≤5
            Cprimetemp = C(i,j);
            Cprime = [Cprime;Cprimetemp]; %if C is between -5 and 5, ...
               append it to Cprime
        end
    end
end
```

```
Cprime2 = C((C≥-5)&(C≤5)); %This is a MUCH easier way to accomplish the ...
    same thing

% (c) (c) Create a 3-dimensional matrix called X  that is of dimension % N ...
    x K x T  where N=15,169 , K=6 , and T=5 .
n=15169;
k=6;
T=5;
X=zeros(n,k,T);
X(:,:,1)=[ones(n,1) (rand(n,1)<.75) 15+5*randn(n,1) ...
    pi/3+(exp(-1))*randn(n,1) binornd(20,.6,n,1) binornd(20,.5,n,1)];
for t=1:T;
    X(:,1,t) = ones(n,1);
    X(:,2,t) = (rand(n,1)<.75*(6-t)/5);
    X(:,3,t) = 15+(t-1)+5*(t-1)*randn(n,1);
    X(:,4,t) = pi*(6-t)/3+(exp(-1))*randn(n,1);
    X(:,5,t) = X(:,5,1);
    X(:,6,t) = X(:,6,1);
end

% (d) Use loops to create a matrix beta which is K x T  and whose elements ...
    evolve across time
beta = zeros(k,T);
for t=1:T
    beta(1,t)=1+.25*(t-1);
    beta(2,t)=log(t);
    beta(3,t)=-sqrt(t);
    beta(4,t)=exp(t)-exp(t+1);
    beta(5,t)=t;
    beta(6,t)=t/3;
end

% (e) Use loops to create a matrix Y  which is N\times T  defined by ...
    Y_{t}=X_{t}\beta_{t}+\varepsilon_{t} , where ...
    \varepsilon_{t}\overset{iid}{\sim}N\left(0,\sigma=.36\right)
Y = zeros(n,T);
for t=1:T
    Y(:,t) = X(:,:,t)*beta(:,t)+ .36*randn(n,1);
end
%% Problem 3
% (a) Clear the workspace and import the file nlsw88.csv into Matlab. Make ...
    sure you appropriately convert missing values and variable names.
clear all;
data = csvread('nlsw88_use.csv',1); %skip the first line (which has ...
    variable names)
% I converted missings to NaNs in Excel, then saved the file as ...
    'nlsw88_use.csv' so I could directly import it to Matlab
idcode = data(:,1);
age = data(:,2);
race = data(:,3);
married = data(:,4);
never_married = data(:,5);
grade = data(:,6);
collgrad = data(:,7);
```

```
120 south = data(:,8);
121 smsa = data(:,9);
122 c_city = data(:,10);
123 industry = data(:,11);
124 occupation = data(:,12);
125 union = data(:,13);
126 wage = data(:,14);
127 hours = data(:,15);
128 ttl_exp = data(:,16);
129 tenure = data(:,17);
130 clear data
131 save nlsw88
132
133 % (b) What percentage of the sample has never been married? What ...
       percentage are college graduates?
134 nanmean(never_married);
135 nanmean(collgrad);
136
137 % (c) Use the tabulate command to report what percentage of the sample is ...
       in each race category
138 tabulate(race);
139
140 % (d) Create a matrix called summarystats which lists the mean, median, ...
       standard deviation, min, max, number of unique elements, and ...
       interquartile range (75th percentile minus 25th percentile) of grade ...
       (highest grade completed).
141 summarystats = [nanmean(grade) nanmedian(grade) nanstd(grade) ...
       nanmin(grade) nanmax(grade) iqr(grade)];
142 missing_pct = 100*sum(isnan(grade))/length(grade);
143
144 % (e) Graphically show the joint distribution of industry and occupation
145 hist3([industry occupation],[12 13]);
146
147 % (f) Tabulate the mean wage over industry categories and union status
148 meanwage = grpstats(wage,[industry occupation],'mean');
149 %% Problem 4
150 % (a) Clear the workspace and load firstmatrix.mat.
151 clear all; load firstmatrix
152
153 % (b) Write a function called matrixops that takes as inputs the matrices ...
       A  and B  from question (a) of problem 1 and has three outputs: (i) ...
       the element-by-element product of the inputs, (ii) the product ...
       A^{\prime}B , and (iii) the sum of all the elements of A+B .
154 % see "matrixops.m"
155
156 % (c) Starting on line 2 of the function, write a comment that explains ...
       what matrixops does.
157 % see "matrixops.m"
158
159 % (d) In the command window, type help matrixops. What comes up?
160 % The comment from the first line(s) of the function come up
161
162 % (e) Evaluate matrixops.m using A  and B  from question (a) of problem 1
163 [elem,product,summation]=matrixops(A,B);
```

```
164
165 % (f) Just before the first executable line of matrixops.m (i.e. right ...
      after the first-line comments), write an if statement which gives an ...
      error if the two inputs are not the same size. Have the error say ...
      Şinputs must have the same size.Ť
166 % see "matrixops.m"
167
168 % (g) Evaluate matrixops.m using C  and D  from question (a) of problem 1. ...
      What happens?
169 % [elem,product,summation]=matrixops(C,D);
170 % I get the error message that I typed in the function matrixops
171
172 % (h) Now evaluate matrixops.m using ttl_exp and wage from nlsw88.mat.
173 clear all; load nlsw88;
174 [elem2,product2,summation2]=matrixops(ttl_exp,wage);
```

2 Function Code

```
function [element,product,summation]=matrixops(mat1,mat2)
% [element,product,summation]=MATRIXOPS(mat1,mat2) is a function which takes
% as inputs two marices (must be of the same size) and outputs the
% element-by-element product, the product mat1'*mat2, and the summation
% of all elements of mat1 and mat2
if size(mat1)≠size(mat2)
    error('mat1 and mat2 must have the same size')
end
element = mat1.*mat2;
product = mat1'*mat2;
summation = sum(sum(mat1+mat2));
```