
Table of Contents

.....	1
Question 1.	2
Question 2	2
Question 3	5
Question 4	5
Extra Credit	7

```
clear all;  
close all;
```

```
awfuldata = readtable('Neonatal_Mortality.xlsx', 'ReadVariableNames', ...  
    true, 'Range', 'A1:CA195');
```

```
% This mess of ugly corrects the random sequential order year based data  
% was arranged in the original dataset, as well as correcting the  
% importation as a string value as opposed to a numerical/double
```

```
% Cooper explained how to make this work more elegantly, but... meh.
```

```
awfulnum(1,:) = str2double(awfuldata.x1990);  
awfulnum(2,:) = str2double(awfuldata.x1991);  
awfulnum(3,:) = str2double(awfuldata.x1992);  
awfulnum(4,:) = str2double(awfuldata.x1993);  
awfulnum(5,:) = str2double(awfuldata.x1994);  
awfulnum(6,:) = str2double(awfuldata.x1995);  
awfulnum(7,:) = str2double(awfuldata.x1996);  
awfulnum(8,:) = str2double(awfuldata.x1997);  
awfulnum(9,:) = str2double(awfuldata.x1998);  
awfulnum(10,:) = str2double(awfuldata.x1999);  
awfulnum(11,:) = str2double(awfuldata.x2000);  
awfulnum(12,:) = str2double(awfuldata.x2001);  
awfulnum(13,:) = str2double(awfuldata.x2002);  
awfulnum(14,:) = str2double(awfuldata.x2003);  
awfulnum(15,:) = str2double(awfuldata.x2004);  
awfulnum(16,:) = str2double(awfuldata.x2005);  
awfulnum(17,:) = str2double(awfuldata.x2006);  
awfulnum(18,:) = str2double(awfuldata.x2007);  
awfulnum(19,:) = str2double(awfuldata.x2008);  
awfulnum(20,:) = str2double(awfuldata.x2009);  
awfulnum(21,:) = str2double(awfuldata.x2010);  
awfulnum(22,:) = str2double(awfuldata.x2011);  
awfulnum(23,:) = str2double(awfuldata.x2012);  
awfulnum(24,:) = str2double(awfuldata.x2013);  
awfulnum(25,:) = str2double(awfuldata.x2014);  
awfulnum(26,:) = str2double(awfuldata.x2015);
```

```
awful2015 = str2double(awfuldata.x2015);
```

```
awful1990 = str2double(awfuldata.x1990);
```

Warning: Variable names were modified to make them valid MATLAB identifiers

Question 1.

```
numCountries = length(unique(awfuldata.Country))
units = unique(awfuldata.Units)

% There are 194 different countries in this database.
% The units for Neonatal mortality are deaths/1000 live births

numCountries =

    194

units =

    'per 1,000 live births'
```

Question 2

```
figure;

hist(awful1990, 8:8:80)
g = findobj(gca, 'Type', 'patch');
set(g, 'FaceColor', 'r', 'EdgeColor', 'b', 'FaceAlpha', 0.75);

hold on;

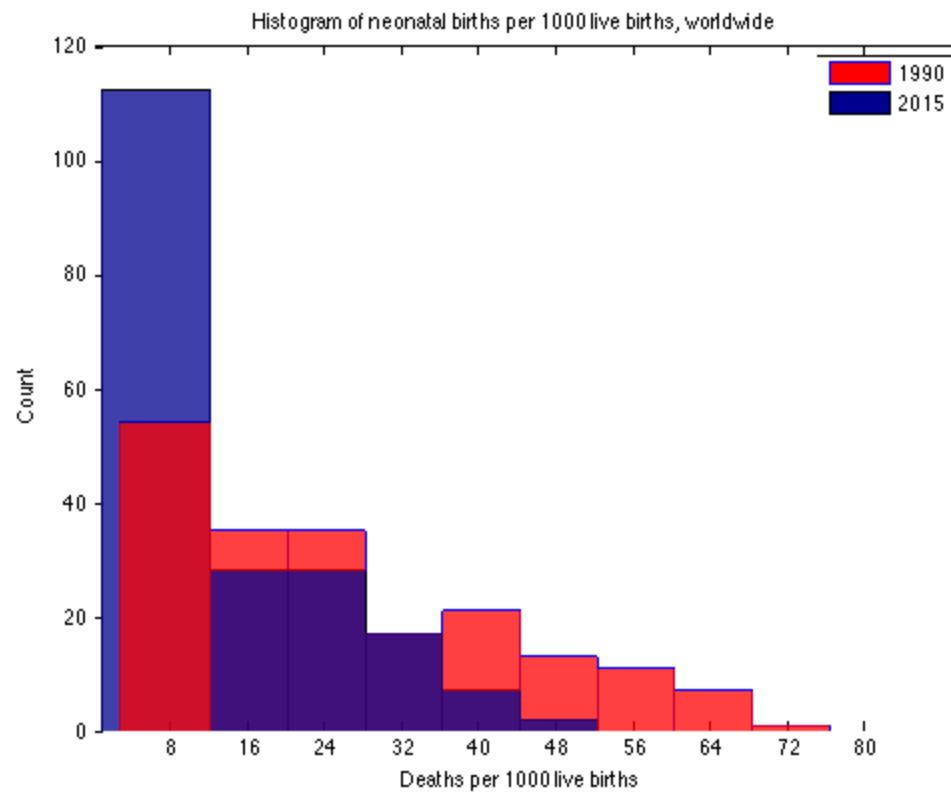
hist(awful2015, 8:8:80)
h = findobj(findobj(gca, 'Type', 'patch'));
set(h, 'FaceAlpha', 0.75);
hold off;
xlabel('Deaths per 1000 live births')
ylabel('Count')
legend('1990', '2015');
title('Histogram of neonatal births per 1000 live births, worldwide')

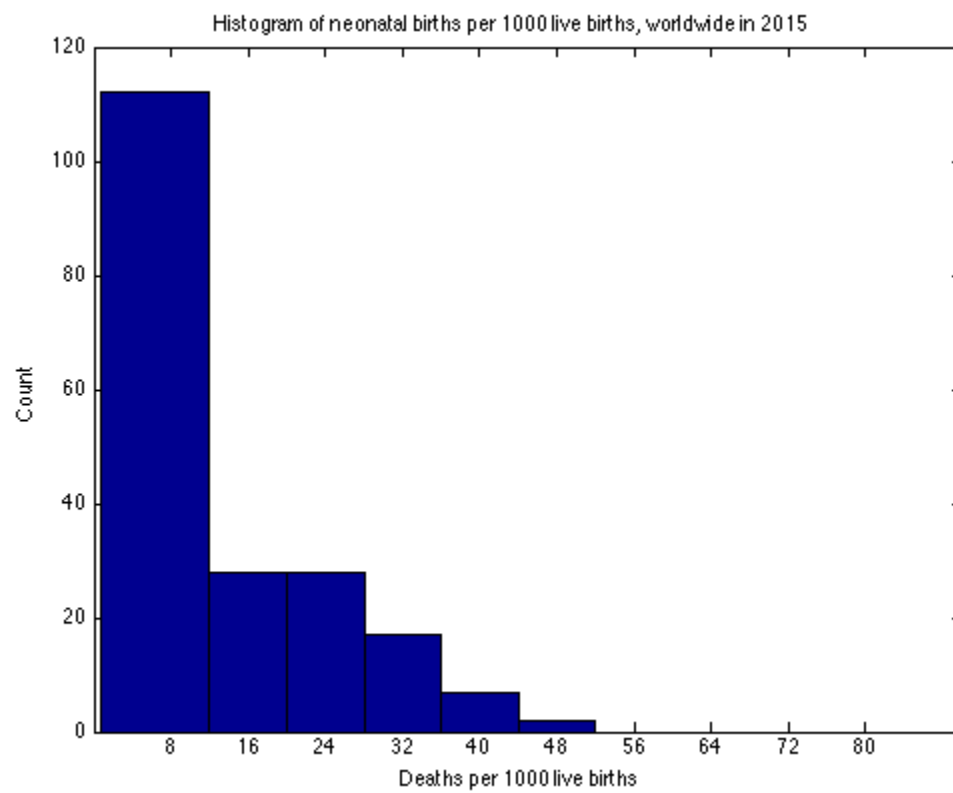
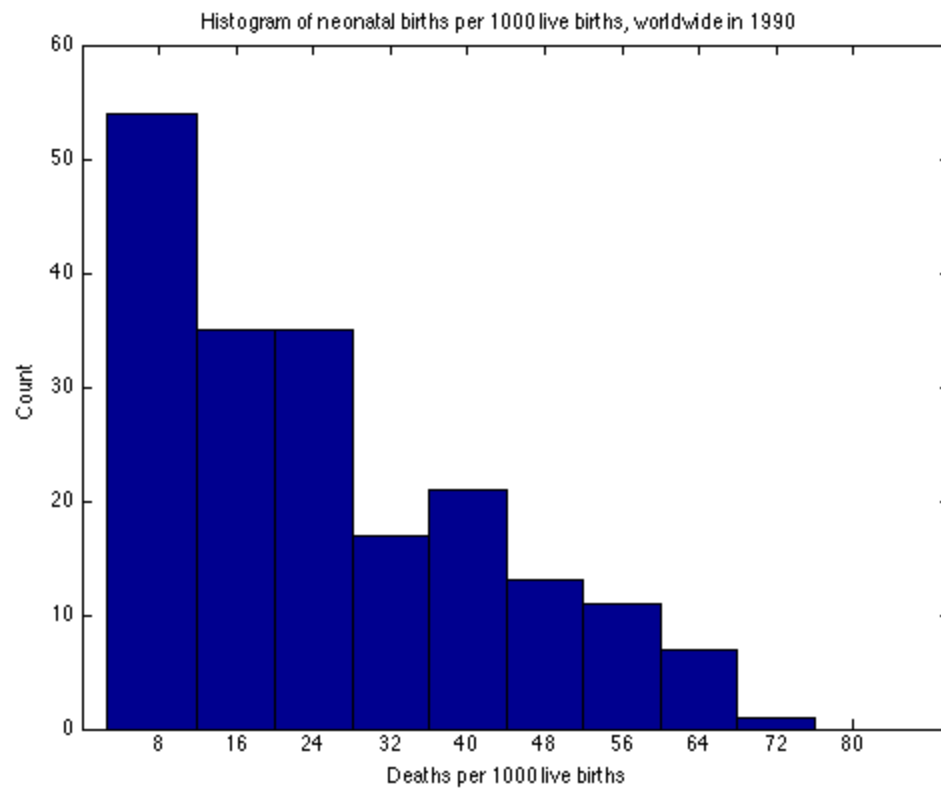
% I talked to Dr. Brunton and she gave me permission to also turn in the
% two histograms on separate plots, since I have Matlab R2014a, which does
% not have the histogram() command (Came out in R2014b). The above roughly
% approximates the shading histogram() would provide, but the first bin is
% a little wonky?
%

figure;
hist(awful1990, 8:8:80)
xlabel('Deaths per 1000 live births')
```

```
ylabel('Count')
title('Histogram of neonatal births per 1000 live births, worldwide in 1990')

figure;
hist(awful2015, 8:8:80)
xlabel('Deaths per 1000 live births')
ylabel('Count')
title('Histogram of neonatal births per 1000 live births, worldwide in 2015')
```





Question 3

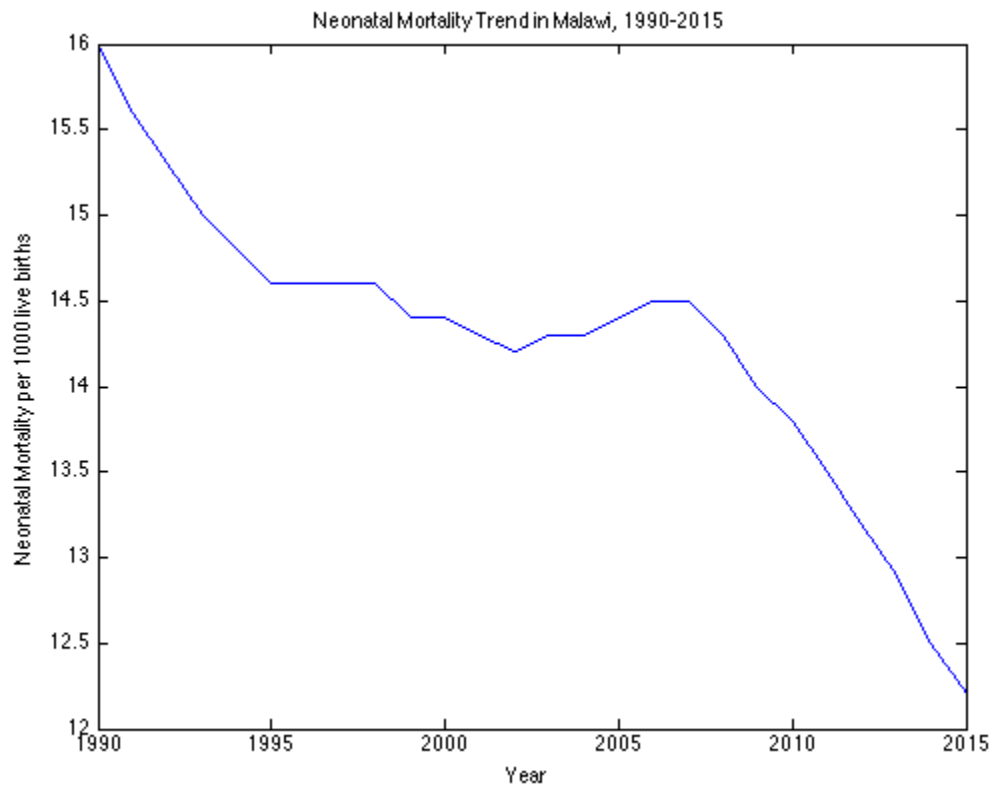
```
awfulcountryid = find(strcmp('MW', awfuldata.CountryRegionId))

malawidata = awfulnum(:,awfulcountryid);

figure;
plot(1990:2015, malawidata')
ylabel('Neonatal Mortality per 1000 live births');
xlabel('Year');
title('Neonatal Mortality Trend in Malawi, 1990-2015');
```

```
awfulcountryid =
```

```
174
```



Question 4

```
awfulnum = awfulnum';
lowinc = find(strcmp('Low income', awfuldata.CountryIncomeGroup));
uncatinc = find(strcmp('-', awfuldata.CountryIncomeGroup));

awfulhighincl = find(strcmp('High income: OECD', awfuldata.CountryIncomeGroup)) ;
```

```
awfulhighinc2 = find(strcmp('High income: nonOECD', awfuldata.CountryIncomeGroup));

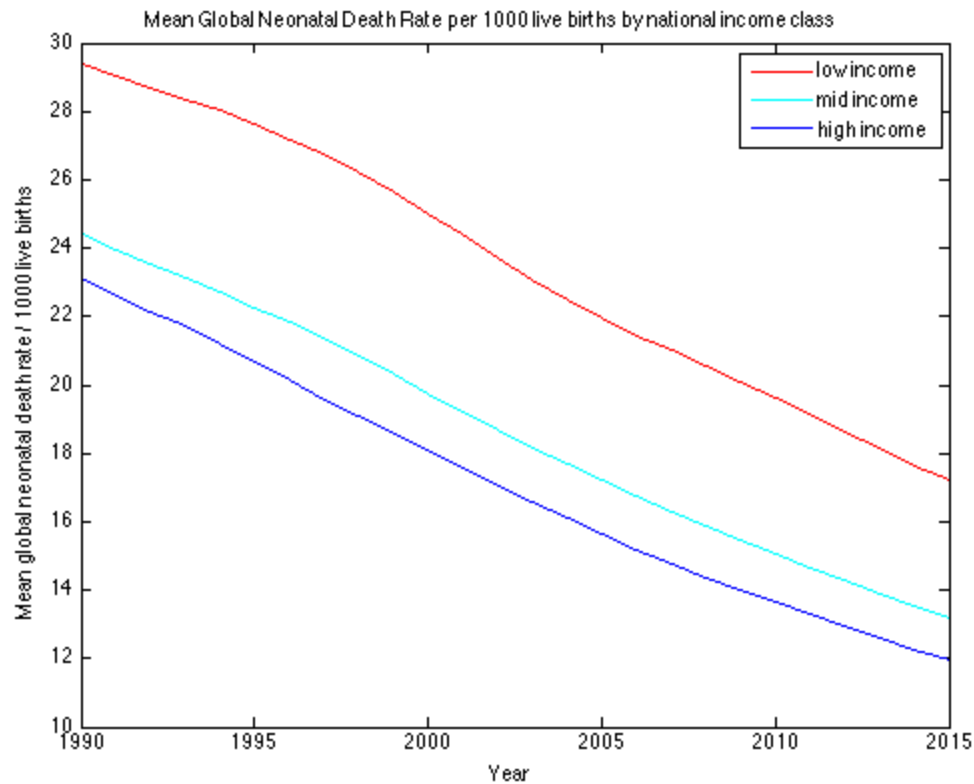
awfulmidinc1 = find(strcmp('Lower middle income', awfuldata.CountryIncomeGroup));
awfulmidinc2 = find(strcmp('Upper middle income', awfuldata.CountryIncomeGroup));

highinc = vertcat(awfulhighinc1, awfulhighinc2);
midinc = vertcat(awfulmidinc1, awfulmidinc2);

lowdata = awfulnum(lowinc, :);
middata = awfulnum(midinc, :);
highdata = awfulnum(highinc, :);

lowmean = mean(lowdata,1);
midmean = mean(middata,1);
highmean = mean(highdata,1);

figure;
plot(1990:2015, lowmean, 'r')
hold on;
plot(1990:2015, midmean, 'c');
plot(1990:2015, highmean, 'b');
hold off;
legend('low income', 'mid income', 'high income');
xlabel('Year');
ylabel('Mean global neonatal death rate / 1000 live births');
title('Mean Global Neonatal Death Rate per 1000 live births by national income cla
```



Extra Credit

```

initialdata = awfulnum(:,20); % Year 2000 Neonatal death rates
trainingindices = randsample(1:194, 125); % random numbers to generate test data
testindices = 1:194; % Lame hack to get the indices that aren't in the training

testindices(ismember(testindices,trainingindices)) = []; % deletes any values t

trainingdata = initialdata(trainingindices, :); % Creates vector of neonatal deat
testdata = initialdata(testindices, :); % Creates vector of neonatal death rates

initincomedata = awfuldata.CountryIncomeGroup; % Creates vector of income group da

for(i = 1:length(initincomedata)) % Cleans up income data group to just have High

    if(isequal(initincomedata{i}, 'High income: OECD'))
        initincomedata{i} = 'High';
    elseif(isequal(initincomedata{i}, 'High income: nonOECD'))
        initincomedata{i} = 'High';
    elseif(isequal(initincomedata{i}, 'Lower middle income'))
        initincomedata{i} = 'Mid';
    elseif(isequal(initincomedata{i}, 'Upper middle income'))
        initincomedata{i} = 'Mid';
    elseif(isequal(initincomedata{i}, 'Low income'))

```

```

        initincomedata{i} = 'Low';
    end;
end;

niter = 400;
val = zeros(niter,1);

for(i = 1:niter)
    trainingindices = randsample(1:194, 125); % random numbers to generate test data
    testindices = 1:194; % Lame hack to get the indices that aren't in the training

    testindices(ismember(testindices,trainingindices)) = []; % deletes any values t

    trainingdata = initialdata(trainingindices, :); % Creates vector of neonatal deat
    testdata = initialdata(testindices, :); % Creates vector of neonatal death rates

    trainingincome = initincomedata(trainingindices); % Creates a vector of income gr
    testincome = initincomedata(testindices);

    test = categorical(trainingincome); % Turns trainingincome into a catagorical vari
    classify(testdata, trainingdata, test', 'Linear'); % Actual classify call taht ret

    val(i,1) = mean(ismember(classify(testdata, trainingdata, test', 'Linear'), testin

end;

mean(val)

ans =

    0.9336

```

Published with MATLAB® R2014a