Solutions to Homework 2 (covering Statistics Lectures 1 and 2)

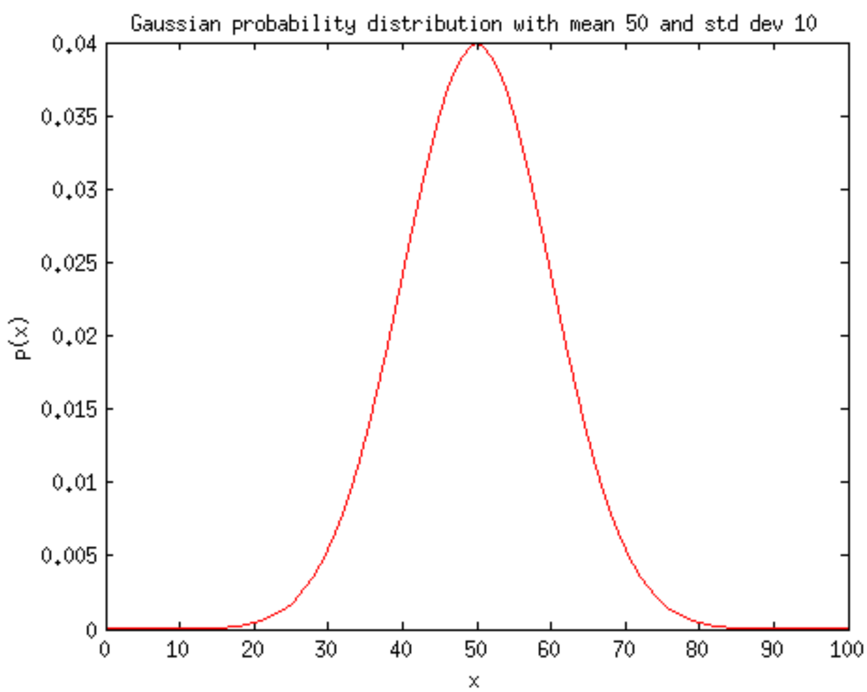
Contents

Problem 0

```
load('Homework2.mat');
```

Problem 1

```
% define some stuff
sigma = 10;  % standard deviation
mn = 50;     % mean
x = 0:100;   % x-values to evaluate function at

% evaluate the Gaussian function at all x-values
y = 1/(sigma*sqrt(2*pi)) * exp(-(x-mn).^2/(2*sigma^2));

% visualize the results
figure;
plot(x,y,'r-');
xlabel('x');
ylabel('p(x)');
title(sprintf('Gaussian probability distribution with mean %d and std dev %d',mn,sigma));
```



Problem 2

```
% compute the metric on the actual data
actualval = std(mean(data1,2));

% the null hypothesis is that all three groups actually reflect
% the same underlying probability distribution.  so, let's
% aggregate all the values together.
alldata = data1(:);

% in our randomization test, we will randomly divide our data
% into three groups and compute the metric.
vals = zeros(1,10000);  % initialize vector of results
for p=1:10000

  % randomly shuffle the data
  datashuffle = alldata(randperm(length(alldata)));

  % reshape into three groups
  datashuffle = reshape(datashuffle,[3 20]);

  % compute the metric and record the result
  vals(p) = std(mean(datashuffle,2));

end

% in what fraction of simulations are values observed
% that are more extreme than the actual value?
pval = sum(vals > actualval) / 10000;

% report the result
pval
```

```
pval =

                    0.0095
```

Problem 3

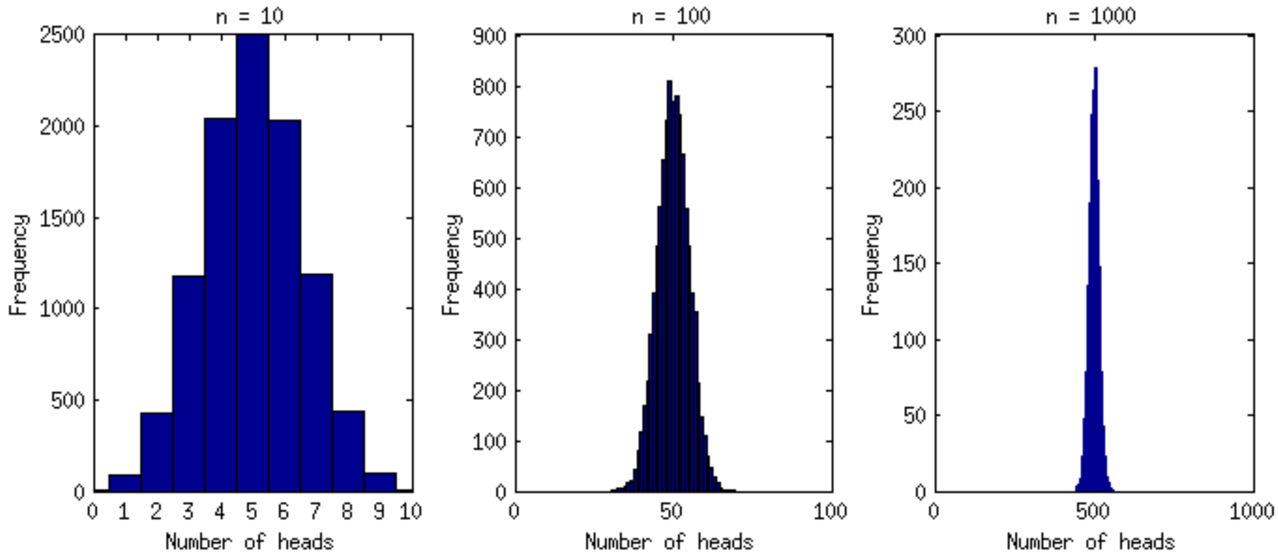
```
% compute the difference between the post- and pre-manipulation measurements
x = data2b - data2a;

% what is the actually observed difference?
actualval = mean(x);

% the null hypothesis is that the values in x come from a probability
% distribution that has a mean of 0.  (in other words, the experimental
% manipulation doesn't really increase or decrease the measured values.)

% let's use the observed x values as an empirical probability distribution
% that conforms to the null hypothesis.  however, we first need to
% subtract off the mean of x.
xcentered = x - mean(x);

% now, let's use bootstrapping to see what sort of x-values are likely
% to be observed given our sample size
vals = zeros(1,10000);
for p=1:10000
  ix = ceil(length(xcentered)*rand(1,length(xcentered)));  % bootstrap indices
  vals(p) = mean(xcentered(ix));  % record the result
end
```

```
% in what fraction of simulations are the differences that are observed
% more extreme than the actually observed difference?
pval = sum(abs(vals) > abs(actualval)) / 10000;

% report the result
pval
```

```
pval =

                    0.0363
```

Problem 4

```
% define number of coin flips
ns = [10 100 1000];

% create a new figure and resize it
figure;
set(gcf,'Position',[100 100 800 300]);

% loop over number of coin flips
for p=1:length(ns)

  % perform 10000 simulations of random coin flips
  flips = round(rand(10000,ns(p)));

  % count number of heads in each simulation
  dist = sum(flips==0,2);

  % initalize subplot
  subplot(1,length(ns),p);

  % plot histogram with bin centers at each possible number of heads
  hist(dist,0:1:ns(p));

  % make sure there are good axis bounds
  ax = axis;
  axis([0 ns(p) ax(3:4)]);

  % label axes and give a title
  xlabel('Number of heads');
  ylabel('Frequency');
  title(sprintf('n = %d',ns(p)));

end
```



Published with MATLAB® R2012b