

STA113 Lab Simulation Session
Spring 2005
I. H. Dinwoodie

The goal is to introduce a few Matlab commands from the Statistics Toolbox. There are no data files to read in, and nothing to print out.

The time required to type in and execute the commands below should be less than 15 minutes.

Start Matlab on a Unix workstation with the command

```
%matlab  
.....  
>> exit
```

The % character within Matlab starts a comment that is ignored by Matlab.

Week 1 Jan 14

unifrnd, hist, boxplot, mean, median, prctile, std

Generate 1000 pseudo-random numbers in the interval (0,1) and 1000 in the interval (0,2) and compare them.

```
>>help unifrnd
>>a=unifrnd(0,1,10,1);
>>b=unifrnd(0,2,10,1)
>>x=[a,b]
>>whos
>>mean(x)
>>std(x)
>>median(x)
>>prctile(x,95)
>>prctile(x, 5)
>>hist(x)
>>boxplot(x)
```

Week 2 Jan 21

regress, corrcoef, plot, polyfit, lsline

Fit a straight line and a quadratic curve to data.

```
>>x=(1:10)'  
>>y=x+unifrnd(0,1,10,1)  
>>plot(x,y,'.')  
>>corrcoef(x,y)  
>>lsline % adds the regression line to the graph  
>>legend('Data Points','Regression Line',0)  
>>regress(y,[ones(10,1),x]) % add the column of 1's to get an intercept  
>>polyfit(x,y,1)  
>>regress(y,[ones(10,1),x,x.^2]) % fits a quadratic curve, same as below  
>>polyfit(x,y,2)
```

What is the intercept of the fitted line?

Week 3 Jan 28

robustfit

Fit a line to data with some very wrong measurements.

```
>>x=(1:10)'           % ' makes a column vector
>>y=x+unifrnd(0,1,10,1)
>>y(3)=10
>>figure             % opens a new graphics window
>>plot(x,y, '.' )
>>lsline             % adds the least squares line
>>b=robustfit(x,y)   % fits line in a robust way
>>hold               % keeps the same picture for the next plot
>>line(x,b(2)*x+b(1))
```

Week 4 Feb 4

crosstab

Generate biallelic genetic data from Hardy-Weinberg equilibrium and tabulate it into genotype data (unordered pairs). The allele labels are 0 and 1, not A, a. There are 20 individuals, with simulated genetic pairs in the rows of the 20x2 matrix below.

```
>>data=binornd(1,.7,20,2)
>>table=crosstab(data(:,1),data(:,2))
>>g=table+table'      % this is double on the diagonal
>>genotypes=triu(g-tril(triu(table))) % subtract the diagonal
```

Week 5 Feb 11

binornd, binocdf, binoinv, binopdf

```
>>x=binornd(10,.5,50,1)      % flip a coin 10 times, do it fifty times
>>hist(x)
>>mean(x)                   % should be 10*.5
>>var(x)                    % should be 10*.5*.5
>>binocdf(5,10,.5)
>>binoinv(.5,10,.5)        % about the 50th percentile
>>binopdf(5,10,.5)        % (10 choose 5)*.5^10
```

Week 6 Feb 18

exprnd, normrnd, trnd, qqplot

```
>>times=exprnd(2,1000,1);
>>mean(times)
>>std(times)
>>n=normrnd(2,3,1000,1)
>>qqplot(n)
>>t=trnd(3, 1000,1)      % 3 degrees of freedom in the t-distribution
>>hist(t)
>>qqplot(t)             % the qqplot reveals the impostor
```

Week 7 Feb 25

diff, cumsum, ./, Poisson process

```
>>intertimes=exprnd(10,100,1); % Matlab uses the mean as the parameter!
>>times=cumsum(intertimes);
>>times=[0;times];
>>plot(times,0:100,'.') % rate 1/10 per second Poisson process
>>xlabel('Time (s)')
>>ylabel('Cumulative Count of Arrivals')
>>title('Rate 1/10s Poisson Process')
>>mean(diff(times))
>>flag=(0<times & times<=100) % flag the arrival times that are before
% or equal 100 seconds, 0 is not an arrival
>>sum(flag) % count the number of arrivals before 100 seconds
```

Week 8 Mar 4

Compute pi with a Monte Carlo Method

The number $\pi/4$ is the fraction of area in the square $(0,1) \times (0,1)$ occupied by the quarter circle. Simulate 10000 pairs of numbers randomly chosen from the square, put each pair in a row of a 10000x2 matrix. "mag" is the magnitude of the pairs.

```
>>x=unifrnd(0,1,10000,2);  
>>mag=sqrt(x(:,1).^2+x(:,2).^2);  
>>4*sum(mag < 1)/10000
```

This is not very accurate. Is there a way to use the sample size stuff on p. 296 to get 3 decimal places of accuracy?

Week 9 Mar 11
Central Limit Theorem

We show how adding 10 independent $\text{uniform}(0,1)$ random variables changes the shape of the pdf to something close to a Normal density. Below each column is a sample of size 10 from the $\text{unif}(0,1)$ distribution. We will need the `histp.m` m-file for probability histograms for the best picture.

```
>>unifarray=unifrnd(0,1,10,1000); % 1000 samples each of size 10 from unif(0,1)
>>histp(unifarray(1,1:1000)); % the shape of the pdf for unif(0,1)
>>sums=sum(unifarray); % add the 10 numbers in each of 1000 columns
>>histp(sums) % mean is 10*.5, variance=10*1/12
>>var(sums)
>>hold
>>histp(sums/10) % the sample means, variance (1/12)/10.
>>histp((sums-10*.5)/sqrt(10*1/12))
% N(0,1), sums centered around 0, divided by
% their standard deviation (10*1/12)^.5.
```

Week 10 Mar 25

mle

```
>>x=normrnd(2,6,1000,1); % 6 is the standard deviation
>>mle('normal',x)
>>[est,ci]=mle('normal',x)
>> ci(1,1)<2 & 2<ci(2,1) % check if the mean 2 is in the confidence interval
>>[h,p,ci]=ttest(x) % another way to get the confidence interval for mu
```

The confidence interval will usually contain the number it is supposed to trap, but every once in a while it will miss.

Week 12 Apr 8

paired data, unidrnd

Below the vector of shifts are individual effects on "before" (x) and "after" (y) measurements due to an individual property such as age or size.

The difference removes the shift so the one sample method is valid.

Below we have 100 individuals, with an individual property in the range

1 to 20 in the "shifts" vector. Then x is a vector of "before" measurements, a random perturbation of the individual baseline in "shifts", and y is an "after" measurement, another random perturbation of the baseline. The difference of x and y removes the individual effect.

```
>>shifts=unidrnd(20,100,1)
```

```
>>x=shifts+normrnd(-1,1,100,1)
```

```
>>y=shifts+normrnd(2,1,100,1)
```

```
>>d=x-y
```

```
>>hist(x) % cannot assume x is normally distributed
```

```
>>hist(d) % d is more normal
```

```
>>[h,p,ci]=ttest(d) % compares the means of "before" and "after" effects.
```

Week 13 Apr 15

p-values

```
>>x=normrnd(0,1,10,1)
>>help ttest
>>[h,p,ci]=ttest(x,1,.10, 'left')    % test true mean=1, vs mean<1, alpha=.10
>>tstat=(mean(x)-1)/(std(x)/sqrt(10))
>>tcdf(tstat,10-1)                    % lower tail area
>>[h,p,ci]=ttest(x,1,.10,'both')     % two-sided test
>>2*(1-tcdf(abs(tstat),10-1))        % p value is the sum of both tail areas
```

Week 14 Apr 22

==, clusterdata, crosstab

Generate numbers that come from two different probability distributions, corresponding to two different species or categories, and separate them.

```
>>clusterlabel=unidrnd(2,20,1)           % true labels on two populations
>>x=clusterlabel+unifrnd(-.75,.75,20,1)% add randomness to labels for confusion
>>labelestimate=clusterdata(x,2)        % try to separate the two groups
>>misclass=crosstab(labelestimate,clusterlabel)
```

The number of misclassifications is the sum of the off-diagonal entries if the labels are matched 1-1, 2-2, or the sum of the diagonal entries if the labels are matched 1-2, 2-1. The lower number is the right one.