Stat 579 Statistical Computing II Homework 2

Erik Erhardt

February 21, 2006

All code can be found in the Appendix.

Exercise 1 *Inverse CDF method.*

If $X \sim \text{Exp}(\lambda)$, the pdf is $f(x|\lambda) = \lambda e^{-\lambda x}, x \geq 0, \lambda > 0$, with cdf $F(x|\lambda) = 1 - e^{-\lambda x} = v$, where we will have $v \sim \text{Unif}(0, 1)$. Solving for x, $x = -\frac{1}{\lambda} \ln(1 - v)$ and we know $x \sim \text{Exp}(\lambda)$. Let $u = 1 - v$, since $v \sim \text{Unif}(0, 1)$, $1 - v = u \sim \text{Unif}(0, 1)$, so $x = -\frac{1}{\lambda} \ln(u)$ is simplified.

This method works because by construction, the probability of producing a number less than x is the same as the definition of the cdf, that the probability of an exponential number being less than x is $F(x)$. Because these are equal, the probability of generating any range of numbers is correct.

Exercise 2 *Multinomial tests.*

(a) `pearsonp.m` *and* `likelig2.m` *in appendix*

(b) `multinomial_script.m` *in appendix*

(c) *Dean recommentation vs. professor's prerogative.*

```
>> multinomial_script
Multinomial script                                          by Erik Barry Erhardt
This script prompts for:
                   k = the number of cells (>1)
            x_1..x_k = the k cell counts (>0)
  theta_10..theta_k0 = the k cell multinomial probabilities (sum to 1)
Values returned include:
  Pearson P and likelihood G^2 statistics and p-values under chi2 with k-1 df
Input k: 5
Enter x vector of cell counts
Input x(1):16
Input x(2):50
Input x(3):31
Input x(4):11
Input x(5):9
```

```
Enter theta0 vector of multinomial probabilities
Input theta0(1):.1
Input theta0(2):.2
Input theta0(3):.4
Input theta0(4):.2
Input theta0(5):.1
For cell counts:
         16            50            31            11             9
have expected values
         11.7          23.4          46.8          23.4          11.7
with null multinomial probabilities
          0.1           0.2           0.4           0.2           0.1
Pearson P and p-value
       44.346  5.4367e-009
Likelihood G^2 and p-value
       39.078  6.7124e-008
```

There is extremely strong evidence that the professor is not following the Dean's recommendation.

(d) *Multiple samples*

Below I input a 2x4 matrix X, and returned are 2x1 vectors for the test statistics and p-values.

```
>> x=[4 5 0 6;3 2 1 0];theta0=[.5 .2 .1 .2];[P,pval]=likelig2(x,theta0)
P =
       8.3972
        3.065
pval =
     0.038479
      0.38172
>> x=[4 5 0 6;3 2 1 0];theta0=[.5 .2 .1 .2];[P,pval]=pearsonp(x,theta0)
P =
       7.4667
            2
pval =
     0.058421
      0.57241
```

Exercise 3 *Simulate from multinomial.*

(a) `multrnd.m` *in appendix.*

(b) *Samples are correct.*

The plot in Figure 1 on page 3 suggests the simulations are accurate.

(c) *Plot clock-time of generating samples with* `multrnd.m`.

The plot in Figure 2 on page 3 shows the total times for simulations and average time per sample. The plot illustrates that the total number of counts for each sample makes the biggest difference for amount of time, with time increasing roughly linearly with the number of samples. The length of time also increases roughly linearly with the number of multinomial bins. The length of time nearly doubles from 500 to 1000 samples. Thus, the length of time increases proportionally with the number of samples n, the number of bins k, and the count number m.

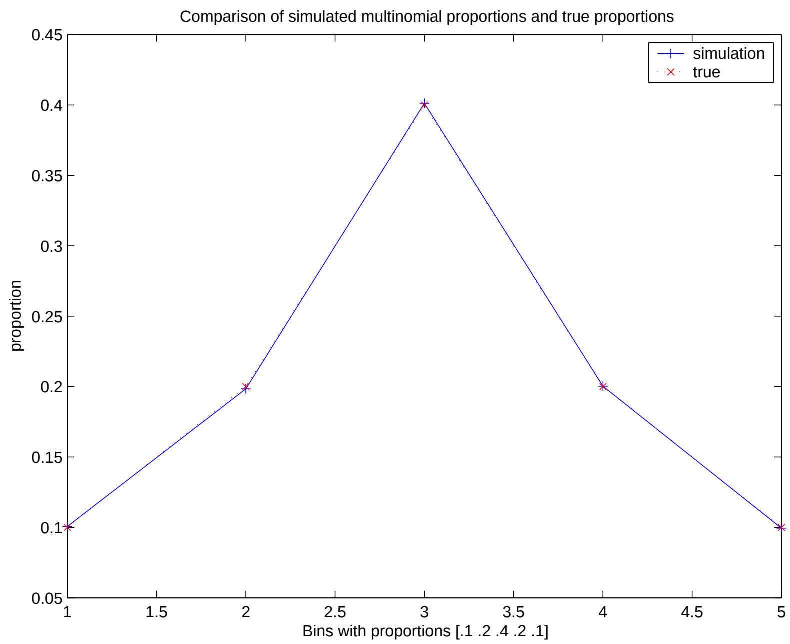

Figure 1: 3b. Simulation replicates true values.

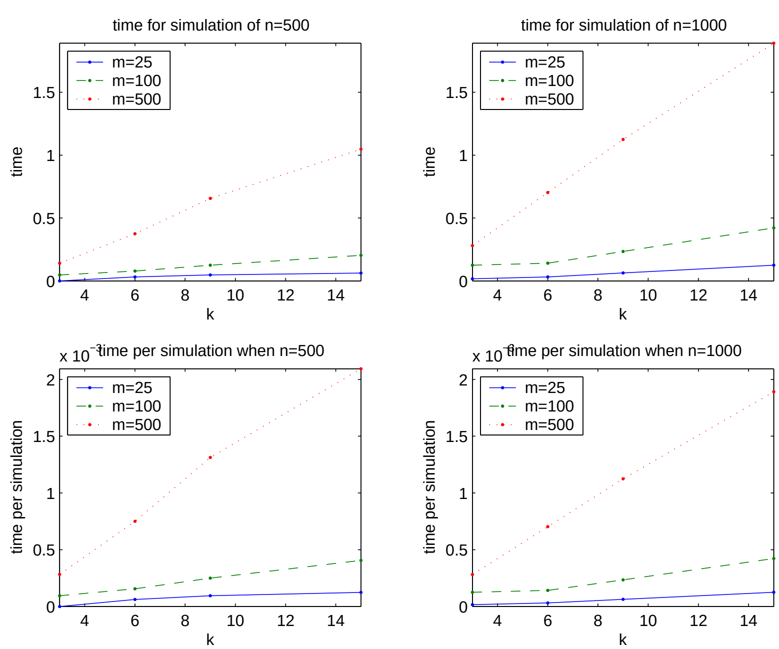

Figure 2: 3c. Total times for simulations and average time per sample.

(d) `multigen.m`

Function `multigen` works because it randomly distributes m items to k bins, with bin i having probability θ_i.

(e) `multigen.m` *is much faster than* `multrnd.m`

`multigen.m` is much faster than `multrnd.m` for a number of reasons. Two of the main ones are because there is no `for` loop in `multigen.m`, and the binning in `histc.m` is in a `mex` file, not in interpreted code.

(f) *n samples at a time*

I had originally did this, rather than use a `for` loop in the sampling design code. I suspect that had I used a `for` loop in the sampling before the two methods would have been about the same, proportionally, depending on how much of actual processing is taken up by the `for` loop.

Exercise 4 *Control variates.*

Table 1 on page 4 gives the results of the MC estimates based on the crude $\hat{\mu}_c$ and control variate $\hat{\mu}_{cv}$, with 95% CIs, and correlation between the Pearson P and Likelihood G^2 statistics. The table is sorted by scenario, k, then n.

n was selected for the MC error to be small, using $c = 0.01$ giving $n = ((4/c)^2/(k-1))$. In those cases, the sample size is larger, but the CI is wider than the specified $4c = 0.04$.

m	k	n	scen.	$\hat{\mu}_c$	(95% CI)	$\hat{\mu}_{cv}$	(95% CI)	ρ
200	10	2000	1	9.2518	(9.0564, 9.4471)	9.0736	(9.0497, 9.0975)	0.99249
200	10	17778	1	9.0715	(9.007, 9.136)	9.0873	(9.0797, 9.0948)	0.99309
200	25	2000	1	24.451	(24.138, 24.765)	24.606	(24.54, 24.672)	0.97786
200	25	6667	1	24.58	(24.406, 24.754)	24.609	(24.571, 24.646)	0.97609
200	50	2000	1	52.056	(51.584, 52.528)	52.23	(52.071, 52.388)	0.94251
200	50	3266	1	52.357	(51.981, 52.732)	52.198	(52.078, 52.319)	0.94748
200	10	2000	2	10.282	(10.083, 10.481)	10.278	(10.157, 10.4)	0.82338
200	10	17778	2	10.273	(10.205, 10.341)	10.267	(10.226, 10.307)	0.82791
200	25	2000	2	26.571	(26.348, 26.795)	26.979	(26.769, 27.189)	0.85594
200	25	6667	2	26.787	(26.66, 26.914)	26.812	(26.69, 26.933)	0.85783
200	50	2000	2	45.822	(45.572, 46.072)	46.224	(45.822, 46.627)	0.97593
200	50	3266	2	45.827	(45.636, 46.018)	46.221	(45.907, 46.535)	0.97446

Table 1: 4. MC estimates of crude $\hat{\mu}_c$ and control variate $\hat{\mu}_{cv}$.

Exercise 5 *Goodness-of-fit upper tail probability approximation.*

Table 2 on page 5 gives the MC proportions of Pearson's P, $\Pr(P \geq t)$, and the Likelihood G^2, $\Pr(G^2 \geq t)$, for $\alpha = \Pr(\chi^2_{k-1} \geq \chi^2_{k-1,1-\alpha} = t)$. Also, is diff $= \Pr(P \geq t) - \Pr(G^2 \geq t)$ and its standard error.

n was selected for the MC error to be small, using $c = 0.01$ giving $n = \alpha(1-\alpha)/c^2$. In those cases, the sample size is larger, and the SE of the difference is roughly $c\sqrt{2}$.

m	k	n	scen.	α	$\Pr(P \geq t)$	$\Pr(G^2 \geq t)$	diff	SE(diff)
200	10	99	1	0.01	0.030303	0.020202	0.010101	0.022288
200	10	475	1	0.05	0.048421	0.054737	−0.0063158	0.01435
200	10	901	1	0.1	0.10655	0.11432	−0.0077691	0.014766
200	25	99	1	0.01	0.020202	0.010101	0.010101	0.017348
200	25	475	1	0.05	0.048421	0.056842	−0.0084211	0.014487
200	25	901	1	0.1	0.096559	0.11543	−0.018868	0.014496
200	50	99	1	0.01	0	0.010101	−0.010101	0.01005
200	50	475	1	0.05	0.048421	0.090526	−0.042105	0.016442
200	50	901	1	0.1	0.09101	0.17869	−0.08768	0.015959
200	10	99	2	0.01	0.020202	0.010101	0.010101	0.017348
200	10	475	2	0.05	0.077895	0.10316	−0.025263	0.018601
200	10	901	2	0.1	0.11654	0.16093	−0.044395	0.016252
200	25	99	2	0.01	0.040404	0.010101	0.030303	0.022195
200	25	475	2	0.05	0.10737	0.063158	0.044211	0.018065
200	25	901	2	0.1	0.14317	0.10655	0.036626	0.01555
200	50	99	2	0.01	0.040404	0	0.040404	0.01979
200	50	475	2	0.05	0.12	0.0042105	0.11579	0.015203
200	50	901	2	0.1	0.16648	0.0022198	0.16426	0.012509

Table 2: 5. Upper-tail probabilities.

Exercise 6 *Importance Sampling.*

The plots in Figure 3 on page 7 indicate that the method using the Beta distribution has extremely small standard errors, but the importance sampling method based on the uniform distribution has large and unpredicable standard errors.

The plot in Figure 4 on page 8 plots the ratio of standard errors.

The plot in Figure 4 on page 8 plots the beta distributions we would like to draw from. If this is not possible, we perform importance sampling. The distribution we draw from should mimic these beta distributions as closely as possible.

For the Beta$(.25, .25)$ distribution, I would use two semi-triangular distributions and a uniform distribution in the middle, to mimic the U-shape. For the Beta$(3, 3)$ distribution, a triangular distribution might work well. For the Beta$(.25, 3)$ distribution, a truncated exponential distribution. For the Beta$(3, .25)$ distribution, a reversed truncated exponential distribution.

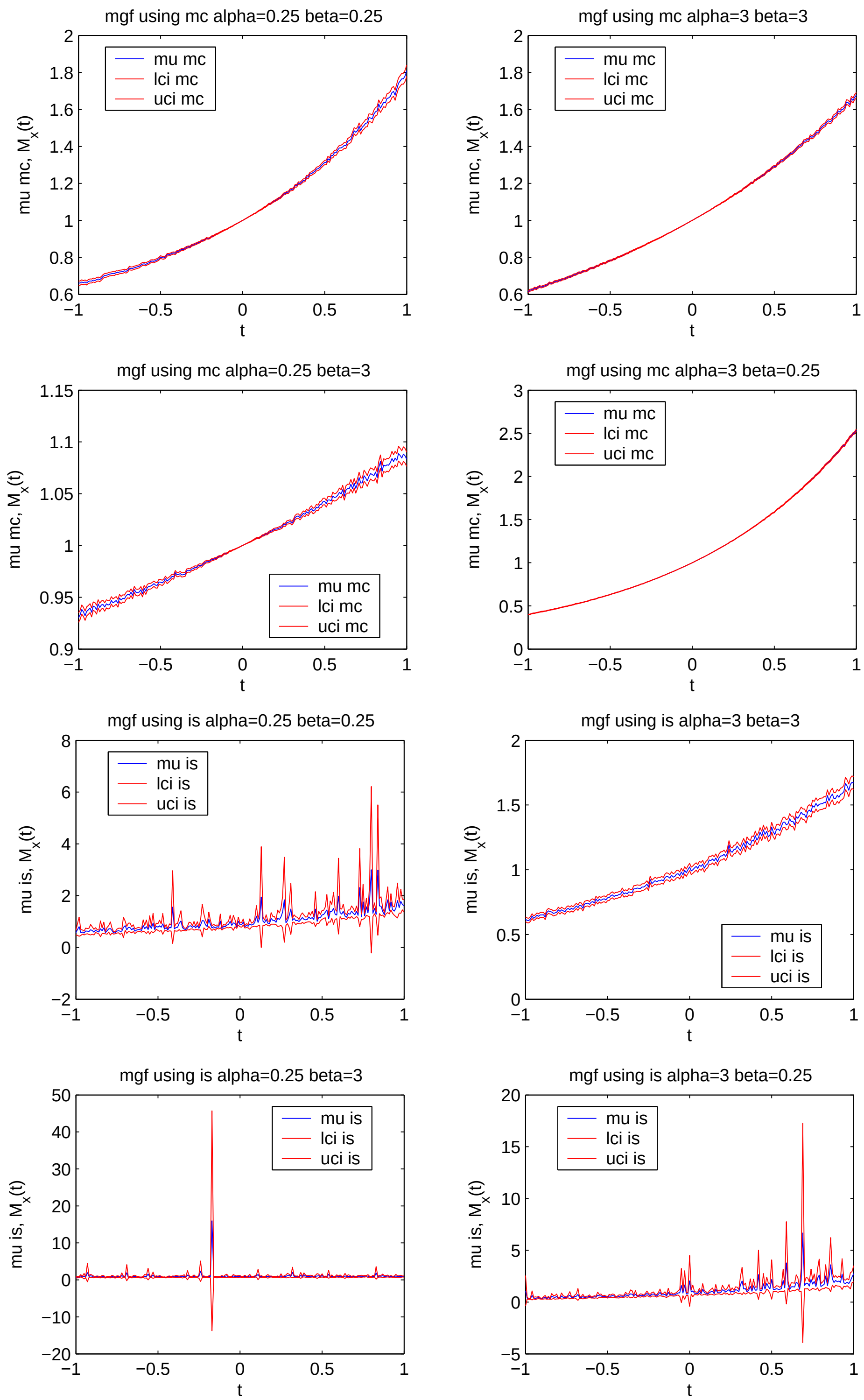
mgf using mc alpha=0.25 beta=0.25
mgf using mc alpha=3 beta=3
mgf using mc alpha=0.25 beta=3
mgf using mc alpha=3 beta=0.25
mgf using is alpha=0.25 beta=0.25
mgf using is alpha=3 beta=3
mgf using is alpha=0.25 beta=3
mgf using is alpha=3 beta=0.25
mu mc
lci mc
uci mc
mu is
lci is
uci is
mu mc, M_x(t)
mu is, M_x(t)
t

Figure 3: 6c.

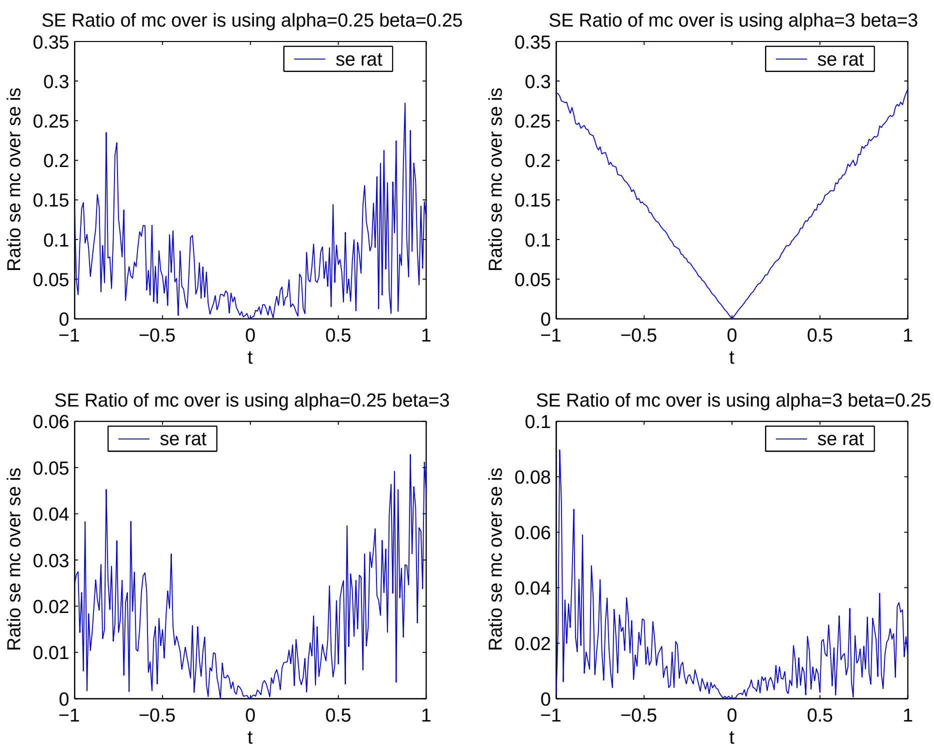

Figure 4: 6d. Ratio of crude MC and uniform IS standard errors.

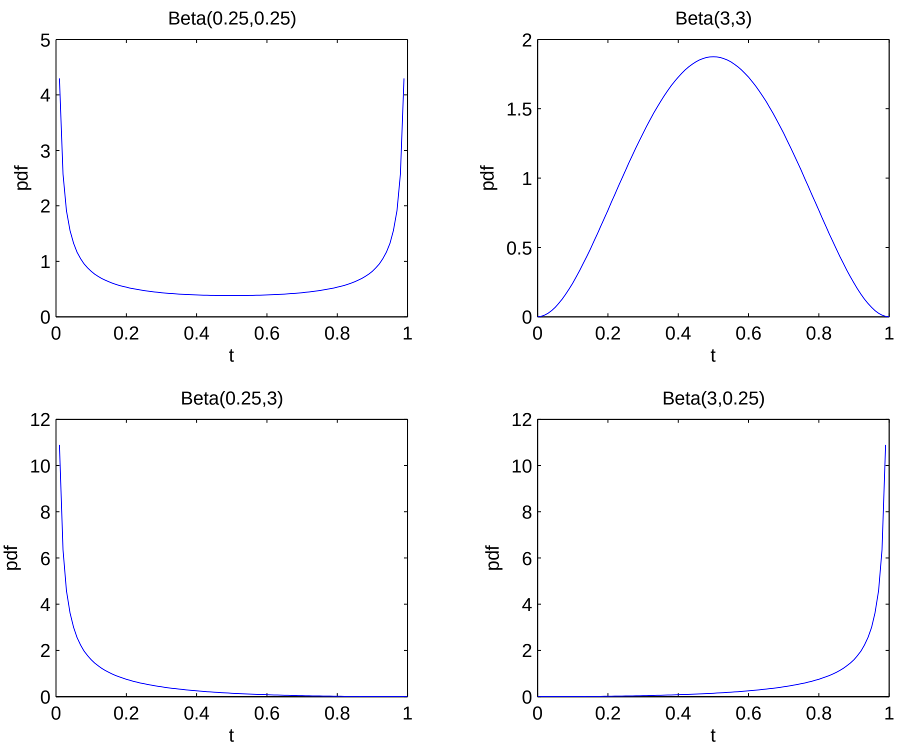

Figure 5: 6d. Beta distributions considered in this problem.

Appendix

Matlab code used for the analysis above

```
%%% Exercise 2 =================================================================
% 2a           ========================
function [P,pval]=pearsonp(x,theta0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  pearsonp, Stat 579 Statistical Computing II Homework 2
%  Create a M-file to calculate the pearson P statistic,
%     and p-value under chi2,
%  input (x,theta0) where
%     x is a matrix with row vectors of counts (each row an observation), and
%     theta0 is a row vector of proportions under H0.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:
%  Erik Barry Erhardt, M.S., Ph.D. student Statistics
%  Dept of Math & Stats, Univ. of New Mexico, Albuquerque, NM 87131, U.S.A.
%  Office: Humanities 328, MSC03 2150
%  erike AT stat.unm.edu    http://www.stat.unm.edu/~erike/
%  (505)277-0757            Fax: (505)277-5505
%  American Statistical Association Albuquerque Chapter Representative
%
% Date: 2/18/06
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin~=2;error('PEARSONP requires 2 arguments');return;end;
xs=size(x);ts=size(theta0);
if xs(2) ~= ts(2);error('Number columns for x and theta0 must be the same');return;end;
if length(find(x<0)) > 0;error('Counts in x must all be nonnegative');return;end;
if length(find(theta0<=0)) > 0 | length(find(theta0>1)) > 0;error('Proportions in theta0 must between 0 and 1');return
eps=1e-10;
if abs(sum(theta0)-1) > eps;error('Sum of theta0 vector is not 1');return;end;
m=sum(x,2);                               % sum of counts in x
k=xs(2);                                  % number of categories
P=sum(((x-m*theta0.*ones(xs)).^2)./(m*theta0),2);   % Pearson P statistic
pval=1-chi2cdf(P,k-1);                    % p-value based on chi2 with k-1 df
% EOF

function [g2,pval]=likelig2(x,theta0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  likelig2, Stat 579 Statistical Computing II Homework 2
%  Create a M-file to calculate the likelihood statistic,
%     and p-value under chi2,
%  input (x,theta0) where
%     x is a matrix with row vectors of counts (each row an observation), and
%     theta0 is a row vector of proportions under H0.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:
%  Erik Barry Erhardt, M.S., Ph.D. student Statistics
%  Dept of Math & Stats, Univ. of New Mexico, Albuquerque, NM 87131, U.S.A.
%  Office: Humanities 328, MSC03 2150
%  erike AT stat.unm.edu    http://www.stat.unm.edu/~erike/
%  (505)277-0757            Fax: (505)277-5505
%  American Statistical Association Albuquerque Chapter Representative
%
% Date: 2/18/06
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin~=2;error('PEARSONP requires 2 arguments');return;end;
xs=size(x);ts=size(theta0);
if xs(2) ~= ts(2);error('Number columns for x and theta0 must be the same');return;end;
if length(find(x<0)) > 0;error('Counts in x must all be nonnegative');return;end;
if length(find(theta0<=0)) > 0 | length(find(theta0>1)) > 0;error('Proportions in theta0 must between 0 and 1');return
eps=1e-10;
if abs(sum(theta0)-1) > eps;error('Sum of theta0 vector is not 1');return;end;
m=sum(x,2);                                  % sum of counts in x
k=xs(2);                                     % number of categories
xp=x;xp(find(x==0))=1; % xp is x but replace 0 counts with 1 for log(argument)
g2=2*sum(x.*log(xp./(m*theta0.*ones(xs))),2); % likelihood G^2 statistic
pval=1-chi2cdf(g2,k-1);                      % p-value based on chi2 with k-1 df
% EOF

% 2b           ========================
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  multinomial_script, Stat 579 Statistical Computing II Homework 2
%  Script for input of
%     number of cells
%     cell counts
%     null probabilities
%  then calculate P and G2 statistics with p-values.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:
%  Erik Barry Erhardt, M.S., Ph.D. student Statistics
%  Dept of Math & Stats, Univ. of New Mexico, Albuquerque, NM 87131, U.S.A.
%  Office: Humanities 328, MSC03 2150
%  erike AT stat.unm.edu    http://www.stat.unm.edu/~erike/
%  (505)277-0757            Fax: (505)277-5505
%  American Statistical Association Albuquerque Chapter Representative
%
% Date: 2/18/06
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
format short g;
format compact;
disp(' ');
disp('Multinomial script                                       by Erik Barry Erhardt');
disp(' ');
disp('This script prompts for:');
disp('                   k = the number of cells (>1)');
disp('            x_1..x_k = the k cell counts (>0)');
disp('  theta_10..theta_k0 = the k cell multinomial probabilities (sum to 1)');
disp(' ');
disp('Values returned include:');
disp('  Pearson P and likelihood G^2 statistics and p-values under chi2 with k-1 df');
disp(' ');
% Enter k
k=0;
while k<2;
   k=input('Input k: ');
end
disp(' ');
disp('Enter x vector of cell counts');
x=-1*ones(1,k); % set default values to invalid values (-1)
theta0=-1*ones(1,k); % set default values to invalid values (-1)
for i=1:k;
   while x(i) < 0;
x(i)=input(strcat('Input x(',num2str(i),'): '));
if x(i) < 0;
warning('x values must be nonnegative integers');
end
   end
end
disp(' ');
disp('Enter theta0 vector of multinomial probabilities');
eps=1e-10;
while abs(sum(theta0)-1) > eps;
for i=1:k;
    while theta0(i) < 0 | theta0(i) > 1;;
theta0(i)=input(strcat('Input theta0(',num2str(i),'): '));
if theta0(i) < 0 | theta0(i) > 1;
warning('theta0 values must be between 0 and 1');
end
end
   end
if abs(sum(theta0)-1) > eps;warning('RENTER, Sum of theta0 vector is not 1');theta0=-1*ones(1,k);end;
end
[P,Ppval]  =pearsonp(x,theta0);
[G2,G2pval]=likelig2(x,theta0);
disp(' ');
disp('For cell counts: ');
disp(x);
disp('have expected values');
disp(sum(x)*theta0);
disp('with null multinomial probabilities ');
disp(theta0);
disp(' ');
disp('Pearson P and p-value ');
disp([P,Ppval]);
```

```
disp('Likelihood G^2 and p-value ');
disp([G2,G2pval]);
format
% EOF
% 2c           ========================
% 2d           ========================
%%% Exercise 3 ================================================================
% 3a           ========================
function x=multrnd(m,theta0,n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  multrnd (inefficient), Stat 579 Statistical Computing II Homework 2
%  Create a M-file to simulate n samples from a multinomial distribution
% with a total count of m,
%     theta0 is a row vector of proportions.
%  input (m,theta0) where
%     m is a scaler, and
%     theta0 is a row vector of proportions under H0.
%  returns a matrix with each sample in a row.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:
%  Erik Barry Erhardt, M.S., Ph.D. student Statistics
%  Dept of Math & Stats, Univ. of New Mexico, Albuquerque, NM 87131, U.S.A.
%  Office: Humanities 328, MSC03 2150
%  erike AT stat.unm.edu    http://www.stat.unm.edu/~erike/
%  (505)277-0757            Fax: (505)277-5505
%  American Statistical Association Albuquerque Chapter Representative
%
% Date: 2/18/06
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin<2;error('MULTRND requires 2 arguments');return;end;
if nargin==2;n=1;end;
if length(find(theta0<=0)) > 0 | length(find(theta0>1)) > 0;error('Proportions in theta0 must between 0 and 1');return
eps=1e-10;
if abs(sum(theta0)-1) > eps;error('Sum of theta0 vector is not 1');return;end;
k=length(theta0);
x=zeros(n,k);
ind=(1:n)';
while (length(ind) > 0);
x(ind,1)=binornd(m, theta0(1), length(ind), 1);
if k>2;
for i=2:(k-1);
x(ind,i)=binornd(m-sum(x(ind,1:(i-1)),2), theta0(i)/(1-sum(theta0(1:(i-1)))));%, length(ind), 1);
end
end
x(ind,k)=m-sum(x(ind,:),2);
ind=find(m-sum(x(ind,:),2) < 0);
x(ind,:)=0;
end
% EOF

% 3b           ========================
n=1000;
m=100;
theta0=[.1 .2 .4 .2 .1]
x=multrnd(m,theta0,n);
xs=sum(x)/sum(sum(x));
plot([1:length(theta0)],xs,'+-' ,[1:length(theta0)],theta0,'rx:');
legend('simulation','true')
title('Comparison of simulated multinomial proportions and true proportions');
xlabel('Bins with proportions [.1 .2 .4 .2 .1]');
ylabel('proportion');
print -depsc2 sc2hw02_3b.eps
% 3c           ========================
n=[500, 1000];
m=[25, 100, 500];
k=[3,6,9,15];
time=zeros(length(m),length(k),length(n));
for m_i=m
   im=find(m==m_i);
   for k_i=k
      ik=find(k==k_i);
      theta0=ones(1,k_i)./k_i;
      for n_i=n
```

```
         in=find(n==n_i);
         tic;
         multrnd(m_i,theta0,n_i);
         time(im,ik,in)=toc;
      end
   end
end
subplot(2,2,1)
plot(k,time(1,:,1),'.-' ,k,time(2,:,1),'.--' ,k,time(3,:,1),'.:');
axis([min(k) max(k) 0 max(max(max(time)))])
legend('m=25','m=100','m=500',2);
title('time for simulation of n=500');
xlabel('k');ylabel('time');

subplot(2,2,2)
plot(k,time(1,:,2),'.-' ,k,time(2,:,2),'.--' ,k,time(3,:,2),'.:');
axis([min(k) max(k) 0 max(max(max(time)))])
legend('m=25','m=100','m=500',2);
title('time for simulation of n=1000');
xlabel('k');ylabel('time');

timeper=time;
timeper(:,:,1)=timeper(:,:,1)/n(1);
timeper(:,:,2)=timeper(:,:,2)/n(2);
subplot(2,2,3)
plot(k,timeper(1,:,1),'.-' ,k,timeper(2,:,1),'.--' ,k,timeper(3,:,1),'.:');
axis([min(k) max(k) 0 max(max(max(timeper)))])
legend('m=25','m=100','m=500',2);
title('time per simulation when n=500');
xlabel('k');ylabel('time per simulation');
subplot(2,2,4)
plot(k,timeper(1,:,2),'.-' ,k,timeper(2,:,2),'.--' ,k,timeper(3,:,2),'.:');
axis([min(k) max(k) 0 max(max(max(timeper)))])
legend('m=25','m=100','m=500',2);
title('time per simulation when n=1000');
xlabel('k');ylabel('time per simulation');
print -depsc2 sc2hw02_3c.eps

% 3d          ========================
function x=multigen(m,theta0,n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  multigen (efficient), Stat 579 Statistical Computing II Homework 2
%  Create a M-file to simulate n samples from a multinomial distribution
% with a total count of m,
%     theta0 is a row vector of proportions.
%  input (m,theta0) where
%     m is a scaler, and
%     theta0 is a row vector of proportions under H0.
%  returns a matrix with each sample in a row.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author:
%  Erik Barry Erhardt, M.S., Ph.D. student Statistics
%  Dept of Math & Stats, Univ. of New Mexico, Albuquerque, NM 87131, U.S.A.
%  Office: Humanities 328, MSC03 2150
%  erike AT stat.unm.edu    http://www.stat.unm.edu/~erike/
%  (505)277-0757            Fax: (505)277-5505
%  American Statistical Association Albuquerque Chapter Representative
%
% Date: 2/18/06
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin<2;error('MULTRND requires 2 arguments');return;end;
if nargin==2;n=1;end;
if length(find(theta0<=0)) > 0 | length(find(theta0>1)) > 0;error('Proportions in theta0 must between 0 and 1');return
eps=1e-10;
if abs(sum(theta0)-1) > eps;error('Sum of theta0 vector is not 1');return;end;
t = [0 cumsum(theta0)]; % partitions [0,1] into sections each theta(i) long.
u = unifrnd(0,1,n,m); % generates m unif(0,1) random deviates
x = histc(u',t)'; % counts the number of the m deviates in u which fall into the k bins specified by the partions in t
x = x(:,1:length(theta0)); % remove k+1 entry of x, which is the count of >1, of which there are none.
% EOF

n=1000;
m=100;
theta0=[.1 .2 .4 .2 .1]
```

```
x=multigen(m,theta0,n);
xs=sum(x)/sum(sum(x));
plot([1:length(theta0)],xs,'+-' ,[1:length(theta0)],theta0,'rx:');
legend('simulation','true')
title('Comparison of simulated multinomial proportions and true proportions');
xlabel('Bins with proportions [.1 .2 .4 .2 .1]');
ylabel('proportion');

print -depsc2 sc2hw02_3d.eps

% 2e          ========================

n=[500, 1000];
m=[25, 100, 500];
k=[3,6,9,15];
time=zeros(length(m),length(k),length(n));
time2=zeros(length(m),length(k),length(n));

for m_i=m
   im=find(m==m_i);
   for k_i=k
      ik=find(k==k_i);
      theta0=ones(1,k_i)./k_i;
      for n_i=n
         in=find(n==n_i);
         tic;
         multrnd(m_i,theta0,n_i);
         time(im,ik,in)=toc;
         tic;
         multigen(m_i,theta0,n_i);
         time2(im,ik,in)=toc;
      end
   end
end

subplot(2,2,1)
plot(k,time(1,:,1),'b+-' ,k,time(2,:,1),'b+--' ,k,time(3,:,1),'b+:' ,k,time2(1,:,1),'rx-' ,k,time2(2,:,1),'rx--' ,k,tim
axis([min(k) max(k) 0 max(max(max(max(time))),max(max(max(time2))))])
legend('multrnd: m=25','multrnd: m=100','multrnd: m=500','multigen: m=25','multigen: m=100','multigen: m=500',2);
title('time for simulation of n=500');
xlabel('k');ylabel('time');

subplot(2,2,2)
plot(k,time(1,:,2),'b+-' ,k,time(2,:,2),'b+--' ,k,time(3,:,2),'b+:' ,k,time2(1,:,2),'rx-' ,k,time2(2,:,2),'rx--' ,k,tim
axis([min(k) max(k) 0 max(max(max(max(time))),max(max(max(time2))))])
legend('multrnd: m=25','multrnd: m=100','multrnd: m=500','multigen: m=25','multigen: m=100','multigen: m=500',2);
title('time for simulation of n=1000');
xlabel('k');ylabel('time');

timeper=time;
timeper(:,:,1)=timeper(:,:,1)/n(1);
timeper(:,:,2)=timeper(:,:,2)/n(2);
timeper2=time2;
timeper2(:,:,1)=timeper2(:,:,1)/n(1);
timeper2(:,:,2)=timeper2(:,:,2)/n(2);

subplot(2,2,3)
plot(k,timeper(1,:,1),'b+-' ,k,timeper(2,:,1),'b+--' ,k,timeper(3,:,1),'b+:' ,k,timeper2(1,:,1),'rx-' ,k,timeper2(2,:,1
axis([min(k) max(k) 0 max(max(max(max(timeper))),max(max(max(timeper2))))])
legend('multrnd: m=25','multrnd: m=100','multrnd: m=500','multigen: m=25','multigen: m=100','multigen: m=500',2);
title('time per simulation when n=500');
xlabel('k');ylabel('time per simulation');

subplot(2,2,4)
plot(k,timeper(1,:,2),'b+-' ,k,timeper(2,:,2),'b+--' ,k,timeper(3,:,2),'b+:' ,k,timeper2(1,:,2),'rx-' ,k,timeper2(2,:,2
axis([min(k) max(k) 0 max(max(max(max(timeper))),max(max(max(timeper2))))])
legend('multrnd: m=25','multrnd: m=100','multrnd: m=500','multigen: m=25','multigen: m=100','multigen: m=500',2);
title('time per simulation when n=1000');
xlabel('k');ylabel('time per simulation');

print -depsc2 sc2hw02_3e.eps

% 3f          ========================

%%% Exercise 4 ================================================================

% 4a          ========================

n=[2000, 0];
m=[200];
k=[10, 25, 50];
t=[1 2];

for m_i=m
   im=find(m==m_i);
   for k_i=k
```

```
      ik=find(k==k_i);
      for n_i=n
         in=find(n==n_i);
         if n_i==0; c=0.01; n_i=ceil((4/c)^2/(k_i-1)); end;
         for t_i=t
            it=find(t==t_i);

            if t_i == 1;theta0=ones(1,k_i)./k_i;end;
            if t_i == 2;theta0=.1*ones(1,k_i)./k_i;theta0(end)=.9+theta0(end);end;

            x=multigen(m_i,theta0,n_i);   % simulation
            P =pearsonp(x,theta0);        % Pearson's P
            G2=likelig2(x,theta0);        % likelihood G^2

            cv=G2-P;

            mu_c=mean(G2);
            v_c =var(G2);
            moe_c=2*sqrt(v_c/n_i);
            lci_c=mu_c-moe_c;
            uci_c=mu_c+moe_c;

            mu_cv=mean(cv)+(k_i-1);
            v_cv =var(cv);
            moe_cv=2*sqrt(v_cv/n_i);
            lci_cv=mu_cv-moe_cv;
            uci_cv=mu_cv+moe_cv;

            corr=corrcoef(G2,P);
            corr=corr(1,2);

            disp([m_i k_i n_i t_i mu_c lci_c uci_c mu_cv lci_cv uci_cv corr])
         end
      end
   end
end

%%% Exercise 5 ==================================================================
n=[0]; % determined by c, alpha, and k
m=[200];
k=[10, 25, 50];
t=[1 2];
alpha=[.01 .05 .1];

for m_i=m
   im=find(m==m_i);
   for k_i=k
      ik=find(k==k_i);
      for n_i=n
         in=find(n==n_i);
         for t_i=t
            it=find(t==t_i);
            for alpha_i=alpha
               ialpha=find(alpha==alpha_i);

               c=0.01; n_i=ceil(alpha_i*(1-alpha_i)/c^2);

               tt=chi2inv(1-alpha_i,k_i-1);

               if t_i == 1;theta0=ones(1,k_i)./k_i;end;
               if t_i == 2;theta0=.1*ones(1,k_i)./k_i;theta0(end)=.9+theta0(end);end;

               x=multigen(m_i,theta0,n_i);   % simulation
               P =pearsonp(x,theta0);        % Pearson's P
               G2=likelig2(x,theta0);        % likelihood G^2

               P_t = length(find(P>=tt))/n_i;
               G2_t = length(find(G2>=tt))/n_i;

               dif=P_t-G2_t;
               dif_se=sqrt(P_t*(1-P_t)/n_i+G2_t*(1-G2_t)/n_i);

               disp([m_i k_i n_i t_i alpha_i P_t G2_t dif dif_se])
            end
         end
      end
   end
end

%%% Exercise 6 ==================================================================
% 6a           ========================

ab=[.25 .25;3 3;.25 3;3 .25];
n=2000;
t=-1:0.01:1;

init = zeros(length(t),length(ab));
mu_mc = init;
se_mc = init;
mu_is = init;
```

```
se_is = init;
clear init;
for ab_i=1:length(ab)
   for t_i=t
      it=find(t==t_i);
      x=betarnd(ab(ab_i,1),ab(ab_i,2),n,1);
      mu_mc(it,ab_i) = sum(exp(t_i.*x))/n;
      se_mc(it,ab_i) = sqrt(var(exp(t_i.*x))/n);
      x=unifrnd(0,1,n,1);
      fb=betapdf(x,ab(ab_i,1),ab(ab_i,2));
      mu_is(it,ab_i) = sum(exp(t_i.*x).*fb)/n;
      se_is(it,ab_i) = sqrt(var(exp(t_i.*x).*fb)/n);
   end
end
lci_mc = mu_mc - 1.96*se_mc;
uci_mc = mu_mc + 1.96*se_mc;
lci_is = mu_is - 1.96*se_is;
uci_is = mu_is + 1.96*se_is;
se_rat = se_mc./se_is;
% 6c          ========================
for iplot=1:4
   subplot(2,2,iplot)
   plot(t,mu_mc(:,iplot),'b-' ,t,lci_mc(:,iplot),'r-' ,t,uci_mc(:,iplot),'r-' );
   legend('mu mc','lci mc','uci mc',0);
   tit=strcat('mgf using mc alpha=',num2str(ab(iplot,1)),' beta=',num2str(ab(iplot,2)));
   title(tit);
   xlabel('t');ylabel('mu mc, M_x(t)');
end
print -depsc2 sc2hw02_6mc.eps
for iplot=1:4
   subplot(2,2,iplot)
   plot(t,mu_is(:,iplot),'b-' ,t,lci_is(:,iplot),'r-' ,t,uci_is(:,iplot),'r-' );
   legend('mu is','lci is','uci is',0);
   tit=strcat('mgf using is alpha=',num2str(ab(iplot,1)),' beta=',num2str(ab(iplot,2)));
   title(tit);
   xlabel('t');ylabel('mu is, M_x(t)');
end
print -depsc2 sc2hw02_6is.eps
% 6d          ========================
for iplot=1:4
   subplot(2,2,iplot)
   plot(t,se_rat(:,iplot),'b-');
   legend('se rat',0);
   tit=strcat('SE Ratio of mc over is using alpha=',num2str(ab(iplot,1)),' beta=',num2str(ab(iplot,2)));
   title(tit);
   xlabel('t');ylabel('Ratio se mc over se is');
end
print -depsc2 sc2hw02_6se.eps

% 6e          ========================
ab=[.25 .25;3 3;.25 3;3 .25];
t=0:0.01:1;
for iplot=1:4
   subplot(2,2,iplot)
   plot(t,betapdf(t,ab(iplot,1),ab(iplot,2)),'b-');
   tit=strcat('Beta(',num2str(ab(iplot,1)),', ',num2str(ab(iplot,2)),')');
   title(tit);
   xlabel('t');ylabel('pdf');
end
print -depsc2 sc2hw02_6beta.eps
```