%=========================================================================

%

% Program to generate realized random numbers from the

% 1. Time invariant model

% 2. Count model

% 3. Linear regression model

% 4. Exponential regression model

% 5. Autoregressive model

% 6. Bilinear model

% 7. Autoregressive and heteroskedastic model

% 8. ARCH model

%

% The realizations of the random draws from the models will be

% different to those reported in the text as the random numbers

% generated by the Gauss random number generator differ from those

% generated by Matlab. Consequently the realizations are not printed

% in this program and the PICTURE WILL BE DIFFERENT TO THE FIGURES

% IN CHAPTER 1.

%

%=========================================================================

clear all;

clc;

% Set seed of random number generator

RandStream.setDefaultStream( RandStream('mt19937ar','seed',1234567) )

t = 5;

% Time invariant model

mu = 0; % Mean

sig = 2; % Standard deviation

z = randn(t,1); % Standard normal random numbers

y = sig\*z; % Generate realizations of y

s = (-10:0.1:10)';

fz = normpdf(s); % Probability distribution of z

fy = normpdf((s-mu)/sig)\*(1/sig); % Probability distribution of y

s1 = s;

y1 = y;

fz1 = fz;

fy1 = fy;

% Count model

theta = 2;

y = poissrnd(theta,t,1);

s = 0:1:9;

s2 = s;

y2 = y;

fy2 = poisspdf(s,theta);

% Linear regression model

beta = 3;

sig = 2;

z = randn(t,1);

x = (0:1:t-1)';

y = beta\*x + sig\*z;

s = (-10:0.1:20)';

fz = normpdf(s);

tmp0 = repmat(s,1,length(beta\*x'));

tmp1 = repmat(beta\*x',length(s),1);

fy = normpdf( (tmp0-tmp1)./sig )\*(1/sig);

s3 = s;

y3 = y;

fz3 = fz;

fy3 = fy;

% Exponential regression model

b0 = 1;

b1 = 2;

x = (0:1:t-1)';

mu = b0 + b1\*x;

z = rand(t,1);

y = -mu.\*log(1 - z);

s = (0.1:0.1:20)';

tmp0 = repmat( s,1,length(mu') );

tmp1 = repmat( mu',length(s),1 );

fy = exp(-tmp0./tmp1)./tmp1;

s4 = s;

y4 = y;

fy4 = fy;

% Autoregressive model

rho = 0.8;

sig = 2;

u = sig\*randn(t+1,1);

y0(1) = 0;

for i = 2:length(u+1)

y0(i) = rho\*y0(i-1)+u(i-1);

end

y = y0(2:end)';

yl = y0(1:end-1)';

s = (-10:0.1:10)';

fu = normpdf(s/sig)\*(1/sig);

tmp0 = repmat(s,1,length(rho\*yl'));

tmp1 = repmat(rho\*yl',length(s),1);

fu = normpdf(s/sig)\*(1/sig);

tmp1 = repmat(rho\*yl',length(s),1);

fy = normpdf( (tmp0-tmp1)./sig )\*(1/sig);

s5 = s;

y5 = y;

fu5 = fu;

fy5 = fy;

% Bilinear time series model

rho = 0.8;

gam = 0.4;

sig = 2;

u = sig\*randn(t,1);

mu = zeros(t,1);

y = zeros(t,1);

for i=2:t

mu(i) = rho\*y(i-1) + gam\*y(i-1)\*u(i-1);

y(i) = mu(i) + u(i);

end

s = (-10:0.1:10)';

fu = normpdf(s/sig)\*(1/sig);

tmp0 = repmat(s,1,length(mu'));

tmp1 = repmat(mu',length(s),1);

fy = normpdf( (tmp0-tmp1)./sig )\*(1/sig);

s6 = s;

y6 = y;

fu6 = fu;

fy6 = fy;

% Autoregressive model with heteroskedasticity

rho = 0.8;

a0 = 0.8;

a1 = 0.8;

w = rand(t+1,1);

sig2 = a0 + a1\*w;

sig = sqrt(sig2);

u = sig.\*rand(t+1,1);

y0(1) = 0.0;

for i = 2:length(u+1)

y0(i) = rho\*y0(i-1)+u(i-1);

end

y = y0(2:end)';

yl = y0(1:end-1)';

sig = sig(2:end);

s = (-10:0.1:10)';

fz = normpdf(s);

tmp0 = repmat(s,1,length(rho\*yl'));

tmp1 = repmat(rho\*yl',length(s),1);

tmp2 = repmat(sig',length(s),1);

fu = normpdf(tmp0./tmp2).\*(1./tmp2);

fy = normpdf( (tmp0-tmp1)./tmp2 ).\*(1./tmp2);

s7 = s;

y7 = y;

fz7 = fz;

fy7 = fy;

% ARCH model

beta = 3.0;

alpha0 = 4.0;

alpha1 = 0.9;

z = randn(t+1,1);

x = (-1:1:t-1)';

y0 = zeros(t,1);

u = zeros(t+1,1);

sig2 = zeros(t+1,1) + alpha0;

sig = sqrt(sig2);

for i=2:t

sig2(i) = alpha0 + alpha1\*u(i-1)^2;

sig(i) = sqrt(sig2(i));

u(i) = sig(i)\*z(i);

y0(i) = beta\*x(i) + u(i);

end

y = y0(2:end);

x = x(2:end);

sig = sig(2:end);

s = (-10:0.1:20)';

fz = normpdf(s);

tmp0 = repmat(s,1,length(beta\*x'));

tmp1 = repmat(beta\*x',length(s),1);

tmp2 = repmat(sig',length(s),1);

fy = normpdf((tmp0-tmp1)./tmp2 ).\*(1./tmp2);

s8 = s;

y8 = y;

fz8 = fz;

fy8 = fy;

% Plot the data

figure(1);

clf;

%--------------------------------------------------------%

% Panel (a)

subplot(4,2,1)

plot(s1,fz1,'-k',...

s1,fy1,'--k',...

'LineWidth',0.75);

title('(a) Time Invariant Model');

ylabel('f(y)');

xlabel('y');

legend('z','y','Location','NorthEast')

legend('boxoff')

%--------------------------------------------------------%

% Panel (b)

subplot(4,2,2)

bar(s2,fy2);

title('(b) Count Model');

ylabel('f(y)');

xlabel('y');

axis([-0.5 9.5 0 0.35])

h = findobj(gca,'Type','patch');

set(h,'FaceColor','w','EdgeColor','k')

%--------------------------------------------------------%

% Panel (c)

subplot(4,2,3)

plot(s3,fy3(:,1),'-k',...

s3,fy3(:,3),'--k',...

s3,fy3(:,5),':k',...

'LineWidth',0.75);

title('(c) Linear Regression Model');

ylabel('f(y)');

xlabel('y');

%legend('z','y1','y3','y5','Location','NorthEast')

%legend('boxoff')

%--------------------------------------------------------%

% Panel (d)

s4 = (0.1:0.1:20)';

subplot(4,2,4)

plot(s4,fy4(:,1),'-k',...

s4,fy4(:,3),'--k',...

s4,fy4(:,5),':k',...

'LineWidth',0.75);

title('(d) Exponential Regression Model');

ylabel('f(y)');

xlabel('y');

%legend('y1','y3','y5','Location','NorthEast')

%legend('boxoff')

%legend2latex(f1)

%--------------------------------------------------------%

% Panel (e)

subplot(4,2,5)

plot(s5,fy5(:,1),'-k',...

s5,fy5(:,3),'--k',...

s5,fy5(:,5),':k',...

'LineWidth',0.75);

title('(a) Autoregressive Model');

ylabel('f(y)');

xlabel('y');

%--------------------------------------------------------%

% Panel (f)

subplot(4,2,6)

plot(s6,fy6(:,1),'-k',...

s6,fy6(:,3),'--k',...

s6,fy6(:,5),':k',...

'LineWidth',0.75);

title('(b) Bilinear Model');

ylabel('f(y)');

xlabel('y');

%--------------------------------------------------------%

% Panel (g)

subplot(4,2,7)

plot(s7,fy7(:,1),'-k',...

s7,fy7(:,3),'--k',...

s7,fy7(:,5),':k',...

'LineWidth',0.75);

title('(c) Autoregressive Heteroskedastic Model');

ylabel('f(y)');

xlabel('y');

%legend('y1','y3','y5','Location','NorthEast')

%legend('boxoff')

% set(gca,'XTick',-10:5:20)

% set(gca,'YTick',0:0.05:0.5)

% xlim([-10,10])

ylim([0,0.5])

%--------------------------------------------------------%

% Panel (d)

subplot(4,2,8)

plot(s8,fy8(:,1),'-k',...

s8,fy8(:,3),'--k',...

s8,fy8(:,5),':k',...

'LineWidth',0.75);

title('(d) ARCH Model');

ylabel('f(y)');

xlabel('y');

% set(gca,'XTick',-10:5:20)

% set(gca,'YTick',0:0.1:0.45)

% xlim([-10,20])

% ylim([0,0.45])

%=========================================================================

%

% Program to estimate an Poisson model and plot the

% log-likelihood function.

%

%=========================================================================

clear all;

clc;

% Data

y = [ 8, 3, 4 ];

% y = [ 6, 2, 3, 1 ] ; % Data used in Poisson exercies

t = length(y);

theta = mean(y);

lnl\_t = y\*log(theta) - theta - log(factorial(y));

g\_t = y/theta - 1;

h\_t = -y/theta^2;

disp( ['Sum of y = ', num2str( sum(y) )] );

disp( ['Mean of y = ', num2str( theta )] );

disp(['Log-likelihood function = ', num2str(mean(lnl\_t)) ] );

disp('');

disp( ' yt lnlt gt ht ');

disp( [y' lnl\_t' g\_t' h\_t' ]);

% \*\*\* Generate graph \*\*\*

theta = 0.01:0.01:15;

lnl = zeros(length(theta),1);

for i = 1:length(theta)

lnl(i) = mean( y\*log(theta(i)) - theta(i) - log(factorial(y)) );

end

% Switch off TeX interpreter and clear figure

set(0,'defaulttextinterpreter','none');

figure(1);

clf;

figure1 = figure;

% Create subplot

subplot1 = subplot(1,2,1,'Parent',figure1);

hold(subplot1,'all');

plot(theta,lnl,'Parent',subplot1,'Color',[0 0 0]);

title('(a) Log-likelihood function');

ylabel('$\ln L\_T(\theta)$');

xlabel('$\theta$');

box off;

subplot2 = subplot(1,2,2,'Parent',figure1);

view(subplot2,[0 -90]);

hold(subplot2,'all');

bar(y,lnl\_t,'Parent',subplot2);

title('(b) Log-density function');

ylabel('$\ln f(y\_t;5)$','VerticalAlignment','bottom','Rotation',90,...

'HorizontalAlignment','center');

xlabel('$y\_t$','VerticalAlignment','cap','HorizontalAlignment','center');

set(gca,'xtick',[1 2 3 4 5 6 7 8 9 10])

laprint(figure1,'poisson','options','factory');

%=========================================================================

%

% Program to estimate an Poisson model and plot the

% log-likelihood function.

%

%=========================================================================

clear all;

clc;

% Data

y = [ 8, 3, 4 ];

% y = [ 6, 2, 3, 1 ] ; % Data used in Poisson exercies

t = length(y);

theta = mean(y);

lnl\_t = y\*log(theta) - theta - log(factorial(y));

g\_t = y/theta - 1;

h\_t = -y/theta^2;

disp( ['Sum of y = ', num2str( sum(y) )] );

disp( ['Mean of y = ', num2str( theta )] );

disp(['Log-likelihood function = ', num2str(mean(lnl\_t)) ] );

disp('');

disp( ' yt lnlt gt ht ');

disp( [y' lnl\_t' g\_t' h\_t' ]);

% \*\*\* Generate graph \*\*\*

theta = 0.01:0.01:15;

lnl = zeros(length(theta),1);

for i = 1:length(theta)

lnl(i) = mean( y\*log(theta(i)) - theta(i) - log(factorial(y)) );

end

% Switch off TeX interpreter and clear figure

set(0,'defaulttextinterpreter','none');

figure(1);

clf;

figure1 = figure;

% Create subplot

subplot1 = subplot(1,2,1,'Parent',figure1);

hold(subplot1,'all');

plot(theta,lnl,'Parent',subplot1,'Color',[0 0 0]);

title('(a) Log-likelihood function');

ylabel('$\ln L\_T(\theta)$');

xlabel('$\theta$');

box off;

subplot2 = subplot(1,2,2,'Parent',figure1);

view(subplot2,[0 -90]);

hold(subplot2,'all');

bar(y,lnl\_t,'Parent',subplot2);

title('(b) Log-density function');

ylabel('$\ln f(y\_t;5)$','VerticalAlignment','bottom','Rotation',90,...

'HorizontalAlignment','center');

xlabel('$y\_t$','VerticalAlignment','cap','HorizontalAlignment','center');

set(gca,'xtick',[1 2 3 4 5 6 7 8 9 10])

laprint(figure1,'poisson','options','factory');

%=======================================================================

%

% Program to estimate a Normal model with unknown mean

% and known variance equal to one

%

%=======================================================================

clear all;

clc;

% Read in the data for y

y = [ 1; 2; 5; 1; 2; ];

t = length( y ); % Define the sample size

% Compute the MLE

theta = mean(y);

disp('MLE');

disp('---');

disp(theta);

% Define the log of the likelihood at each observation

lnlt = -0.5\*log(2\*pi) - 0.5\*(y - theta).^2;

% Evaluate log like at each obs

disp('Log like at MLE for each obs');

disp('----------------------------');

disp([y lnlt])

% Evaluate log likelihood

disp('Log Likelihood at MLE');

disp('---------------------');

disp(mean(lnlt));

% Evaluate the gradient at the MLE for each obs

g\_t = y - theta;

disp('Gradient of the log like at MLE for each obs');

disp('--------------------------------------------');

disp(g\_t);

% Evaluate the gradient at the MLE

g = mean(g\_t);

disp('Gradient of the log likelihood at MLE');

disp('-------------------------------------');

disp(g);

% Evaluate the Hessian at the MLE for each obs

h = -t;

disp('Hessian of the log likelihood at MLE');

disp('------------------------------------');

disp(h);

%=========================================================================

%

% Program to estimate the parameters of a normal distribution by

% maximum likelihood and plot the log of the likelihood function.

%

%=========================================================================

clear all;

clc;

% Read in data

y = [ 5, -1, 3, 0, 2, 3]';

t = length(y); % Define the sample size

% Compute the MLEs

mu\_mle = mean(y);

sig2\_mle = mean( (y - mu\_mle).^2 );

disp('MLE(mu)');

disp('-------');

disp(mu\_mle);

disp('MLE(sig2)');

disp('-------');

disp(sig2\_mle);

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%\*\*

%\*\* Generate graph

%\*\*

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

mu = 1.0:0.05:3.00;

sig2 = 3.0:0.05:5.0;

lnl = zeros( length(mu),length(sig2) );

for i = 1:length(mu);

for j = 1:length(sig2)

lnl(i,j) = -0.5\*t\*log(2\*pi)-0.5\*t\*log(sig2(j))-sum( (y - mu(i)).^2/sig2(j) );

end

end

% Switch off TeX interpreter and clear figure

set(0,'defaulttextinterpreter','none');

figure(1);

clf;

mesh(mu,sig2,lnl,'EdgeColor','black')

xlabel('$\mu$');

ylabel('$\sigma^2$');

zlabel('$\mathrm{ln} L( \mu,\, \sigma^2)$');

set(gca,'ztick',[])

axis tight

grid 'off'

box 'off'

%saveas(gcf,'D:\Stan\Current\Vancebook\mle-basic\Graphics\m\_c1fig5.eps','eps')

laprint(1,'normlike','options','factory');

%=========================================================================

%

% Program to model the number of strikes per annum in the U.S.

% Data are from Kennan (1985)

%

%=========================================================================

clear all;

clc;

% Number of strike per annum, 1968 to 1976

yt = [8; 6; 11; 3; 3; 2; 18; 2; 9];

% Maximum likelihood estimate of the number of strikes

theta = mean(yt);

disp( ['MLE of mean number of strikes (in years) = ' num2str(theta)]);

% Plot the estimated distribution of strike numbers

y = 0:1:20;

f = poisspdf(y,theta);

figure(1)

bar(y,f);

% Plot the histogram

figure(2)

hist(yt,11);

poisspdf

Poisson probability density function

[expand all in page](javascript:void(0);?browser=F1help)

**Syntax**

Y = poisspdf(X,lambda)

**Description**

Y = poisspdf(X,lambda) computes the Poisson pdf at each of the values in X using mean parameters in lambda. X and lambda can be vectors, matrices, or multidimensional arrays that all have the same size. A scalar input is expanded to a constant array with the same dimensions as the other input. The parameters in lambda must all be positive.

The Poisson pdf is

*f*(*x*↓↓*λ*)=*λxx*!*e*−*λ* ; *x*=0,1,2,…,∞ .

The density function is zero unless *x* is an integer.

**Examples**

A computer hard disk manufacturer has observed that flaws occur randomly in the manufacturing process at the average rate of two flaws in a 4 GB hard disk and has found this rate to be acceptable. What is the probability that a disk will be manufactured with no defects?

In this problem, λ = 2 and *x* = 0.

p = poisspdf(0,2)

p =

0.1353

%=========================================================================

%

% Program to model the duration of strikes per annum in the U.S.

% Data are from Kennan (1985)

%

%=========================================================================

clear all;

clc;

% Read the data: US strike data from 1968 to 1976

load strike.mat;

yt = data(:,1);

% Maximum likelihood estimate of strike duration

theta = mean(yt);

disp( ['MLE of mean strike duration (in days) = ' num2str(theta)]);

% Plot the estimated distribution of strike durations

y = 0:10:300;

f = (1/theta)\*exp(-y/theta);

figure(1)

plot(y,f);

% Plot the histogram

figure(2)

hist(yt,21);

%=========================================================================

%

% Program to compute the maximum likelihood estimates

% of the asset price model.

%

% The data consist of the Australian, Singapore and NASDAQ stock

% market indexes for the period 3 January 1989 to 31 December 2009.

%

%=========================================================================

clear all;

clc;

% Load data

load assetprices.mat

pt = pt\_aust; % Choose the asset price

pt = pt/1000;

% MLE based on the log-normal distribution of the share index

% Transitions are from log price at t-1 to log price at t

y = trimr(log(pt),1,0) - trimr(log(pt),0,1);

alpha = mean(y);

sig2 = mean( (y - alpha).^2 );

disp(['MLE of alpha based on the log-normal distribution = ', num2str(alpha)]);

disp(['MLE of sig2 based on the log-normal distribution = ', num2str(sig2)]);

%=========================================================================

% Returns a matrix (or vector) stripped of the specified rows

%

% Inputs:

% x = input matrix (or vector) (n x k)

% rb = first n1 rows to strip

% re = last n2 rows to strip

%

% Mimics the Gauss routine.

%=========================================================================

function z = trimr(x,rb,re)

n = length(x);

if (rb+re) >= n;

error('Attempting to trim too much');

end

z = x(rb+1:n-re,:);

end

%=========================================================================

%

% Maximum likelihood estimation of the stationary distribution of the

% Vasciek model of interest rates using Ait Sahalia's (1996) data.

%

%=========================================================================

clear all;

clc;

% Load data (5505x4 array called eurodata, 1 Jun 1973 - 25 Feb 1995)

load eurodata.mat

rt = eurodata(:,4)\*100;

% Maximum likelihood estimates of the stationary distribution

mu\_r = mean( rt );

sig2\_r = mean( (rt - mu\_r).^2 );

disp( ['MLE of mean of stationary distribution: ', num2str(mu\_r)]);

disp( ['MLE of variance of stationary distribution: ', num2str(sig2\_r)]);

% Compute stationary density from -5% to 35%

r = -5:0.1:25;

fnorm = normpdf( ( r - mu\_r )/sqrt(sig2\_r) )/sqrt(sig2\_r);

prob = normcdf( ( 0 - mu\_r )/sqrt(sig2\_r) );

disp( ['Probability of a negative interest rate: ', num2str(prob)]);

%\*\*\* Generate graph \*\*\*%

% Switch off TeX interpreter and clear figure

% First figure

set(0,'defaulttextinterpreter','none');

figure(1);

clf;

plot(r,fnorm,'-k')

ylabel('f(r)')

xlabel('Interest Rate');

set(gca,'ytick',[]);

axis tight;

% Print the tex file to the relevant directory

%laprint(1,'statdensity','options','factory');

%=========================================================================

%

% Maximum likelihood estimation of the transitional distribution of the

% Vasciek model of interest rates using Ait Sahalia's (1996) data.

%

%=========================================================================

clear all;

clc;

% Load data (5505x4 array called eurodata, 1 Jun 1973 - 25 Feb 1995)

load eurodata.mat

rt = eurodata(:,4)\*100;

% Regress r(t) on r(t-1)

y = trimr(rt,1,0);

x = [ones( length( y ),1 ) trimr(rt,0,1) ];

theta = x\y;

e = y - x\*theta;

sig2 = mean(e.^2);

disp( ['MLE of alpha: ', num2str( theta(1) )]);

disp( ['MLE of rho: ', num2str( theta(2) )]);

disp( ['MLE of beta: ', num2str( theta(2)-1 )]);

disp( ['MLE of sigma^2: ', num2str( sig2 )]);

disp( ' ' );

disp( ['Minimum interest rate: ', num2str( min( rt ) )]);

disp( ['Median interest rate: ', num2str( median( rt ) )]);

disp( ['Maximum interest rate: ', num2str( max( rt ) )]);

disp( ' ' );

disp( ['MLE of mean based on the transitional distribution: ', num2str( -theta(1)/(theta(2)-1) )]);

disp( ['MLE of variance based on the transitional distribution: ', num2str( -sig2/((theta(2)-1)\*(2+theta(2)-1)) )]);

% Compute transitional densities

r = 0:0.1:30;

mu\_min = theta(1) + theta(2)\*min(rt);

fmin = normpdf( ( r - mu\_min )/sqrt(sig2) )/sqrt(sig2);

mu\_med = theta(1) + theta(2)\*median(rt);

fmed = normpdf( ( r - mu\_med )/sqrt(sig2) )/sqrt(sig2);

mu\_max = theta(1) + theta(2)\*max(rt);

fmax = normpdf( ( r - mu\_max )/sqrt(sig2) )/sqrt(sig2);

%\*\*\* Generate graph \*\*\*%

% Switch off TeX interpreter and clear figure

set(0,'defaulttextinterpreter','none');

f1=figure(1);

clf;

plot(r,fmin,'--k',r,fmed,'-k',r,fmax,':k','LineWidth',0.75);

ylabel('f(r)')

xlabel('r');

set(gca,'ytick',[ ]);

axis tight;

%legend('Minimum','Median','Maximum','Location','Best')

%legend2latex( f1 )

% Print the tex file to the relevant directory

%laprint(1,'transdensity','options','factory');