Problem Set 3 Solutions

Tyler Ransom

June 11, 2012

1 Main Problem Set Code

```
clear all;
clc;
load nhanes2d
rand('seed',1234);
%% Problem 1

% (a)
a1 = figure('visible','off');
ksdensity(hct);
xlabel('Hematocrit percentage')
ylabel('Density')
title('Kernel Density Plot of Hematocrit Percentage','FontWeight','bold')
exportfig(a1, 'fig1a.eps')

% (b)
b1 = figure('visible','off');
histfit(hct);
h = get(gca,'Children');
set(h(2),'FaceColor',[.8 .8 1])
xlabel('Hematocrit percentage')
ylabel('Frequency')
title('Histogram of Hematocrit Percentage with Normal ...
    Fit','FontWeight','bold')
exportfig(b1, 'fig1b.eps')


% hist3([region heartatk],[numel(unique(region)) 2])
% hist3([race heartatk],[numel(unique(race)) 2])

% (c)
c1 = figure('visible','off');
[f,x] = ecdf(hct(sex==1));
stairs(x,f);
hold on
[f,x] = ecdf(hct(sex==2));
stairs(x,f,'--');
```

```
xlabel('Hematocrit percentage (hct)')
ylabel('F(hct)')
title('Empirical CDF of Hematocrit Percentage by Gender','FontWeight','bold')
legend('Male','Female')
exportfig(c1, 'fig1c.eps')
hold off

% (d)
d1 = figure('visible','off');
[f,x] = ecdf(hct(region==1));
stairs(x,f);
hold on
[f,x] = ecdf(hct(region==2));
stairs(x,f,'--');
[f,x] = ecdf(hct(region==3));
stairs(x,f,':');
[f,x] = ecdf(hct(region==4));
stairs(x,f,'-.');
xlabel('Hematocrit percentage (hct)')
ylabel('F(hct)')
title('Empirical CDF of Hematocrit Percentage by Region','FontWeight','bold')
legend('Northeast','Midwest','South','West')
exportfig(d1, 'fig1d.eps')
hold off

% (e)
e1 = figure('visible','off');
[f,x] = ecdf(hct(race==1));
stairs(x,f);
hold on
[f,x] = ecdf(hct(race==2));
stairs(x,f,'--');
[f,x] = ecdf(hct(race==3));
stairs(x,f,'-.');
xlabel('Hematocrit percentage (hct)')
ylabel('F(hct)')
title('Empirical CDF of Hematocrit Percentage by Race','FontWeight','bold')
legend('White','Black','Other')
exportfig(e1, 'fig1e.eps')
hold off
%% Problem 2

% (a) i.
y = hct;
X = [ones(size(hct)) age race==2 race==3 heartatk sex==2 highbp region==1 ...
    region==2 region==3 smsa==2 smsa==4 height weight houssiz];
%%% create a vector that is 1 if all obs are there; 0 otherwise:
subset2 = ¬isnan(hct)&¬isnan(age)&¬isnan(race)&¬isnan(heartatk)...
          &¬isnan(sex)&¬isnan(highbp)&¬isnan(region)&¬isnan(smsa)...
          &¬isnan(height)&¬isnan(weight)&¬isnan(houssiz);
y = hct(subset2); %drop missing observations from y
X = X(subset2,:); %drop missing observations from X
nb = size(X,2);   %initialize the number of regressors for later use
%%% Initialize the baseline closed-form OLS formulas (for later comparison)
```

```
beta = (X'*X)\X'*y;

ia2 = figure('visible','off');
ezmesh(@(x,y)beta(1)+beta(end-2).*x+beta(end-1).*y,[nanmin(height),nanmax(height)],[nanmin
xlabel('Height (cm)')
ylabel('Weight (kg)')
zlabel('Hematocrit Percentage')
title('Marginal OLS plane predicting hematocrit percentage by height and ...
    weight')
exportfig(ia2, 'fig2ai.eps')

% (b) i.
test = figure('visible','off');
hist3([height weight],[20,20]);
xlabel('Height (cm)')
ylabel('Weight (kg)')
zlabel('Frequency')
exportfig(test, 'hwdist.eps')

ib2 = figure('visible','off');
scatter3(height,weight,hct,'filled')
hold on
ezmesh(@(x,y)beta(1)+beta(end-2).*x+beta(end-1).*y,[nanmin(height),nanmax(height)],[nanmin
xlabel('Height (cm)')
ylabel('Weight (kg)')
zlabel('Hematocrit Percentage')
title('Marginal OLS plane predicting hematocrit percentage by height and ...
    weight, with actual data values')
exportfig(ib2, 'fig2bi.eps')
hold off
%% Problem 3

% (a)
y =highbp;
X = [ones(size(highbp)) age race==2 race==3 heartatk sex==2 hct region==1 ...
     region==2 region==3 smsa==2 smsa==4 height weight houssiz];
y = highbp(subset2); %drop missing observations from y
X = X(subset2,:); %drop missing observations from X
btest = glmfit(X,y,'binomial','constant','off');
bptest = glmfit(X,y,'binomial','link','probit','constant','off');
options = ...
    optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6,'TolX',1e-8,'TolFun',1
[blogit, llogit, ~,~,~,h] = ...
    fminunc('binaryMLE',rand(size(X,2),1),options,X,y,'logit');
selogit = sqrt(diag(inv(h)));
resultlogit=[[blogit;-llogit] [selogit;NaN]];

% (b)
[bprobit,lprobit]      = fminsearch('binaryMLE',blogit ,options,X,y,'probit');
[bprobit bptest]
[bprobit,lprobit,~,~,~,h] = fminunc('binaryMLE',bprobit,options,X,y,'probit');
seprobit = sqrt(diag(inv(h)));
[bprobit bptest]
resultprobit=[[bprobit;-llogit] [seprobit;NaN]];
```

```
140 save results resultprobit resultlogit
141
142 % (c)
143 compare = [blogit./1.6 bprobit];
144
145 % (d)
146 Phatlogit   = exp(X*blogit)./(1+exp(X*blogit));
147 Phatprobit  = normcdf(X*bprobit,0,1);
148 mean(highbp)
149 mean(Phatlogit)
150 mean(Phatprobit)
151 -llogit
152 -lprobit
```

2 MLE Function Code

```
function like = binaryMLE(b,X,y,type)
% BINARYMLE(b,X,y,type) estimates a simple logit or probit model for
% regressors in X and dependent variable y
% 'type' is a string variable (either 'logit' or 'probit') which dictates
% the distributional assumption of the error term
switch type
    case {'logit',{}}
        logP0 =      -log(1+exp(X*b));
        logP1 = (X*b)-log(1+exp(X*b));
    case 'probit'
        logP0 = log(1-normcdf(X*b,0,1));
        logP1 = log(  normcdf(X*b,0,1));
        % try to trap errors where Matlab tries to evaluate log(0):
        logP0(1-normcdf(X*b,0,1)==0)=log(1e-220);
        logP1(  normcdf(X*b,0,1)==0)=log(1e-220);
end
like = -sum((y==0).*logP0 + (y==1).*logP1);
```