Econ 673
Problem Set #2
Answer Sheet

For each of the following problems, please provide detailed written responses. Also, you should (a) attach copies of computer programs and output if required to answer a question and (b) e-mail me a copy of your code.

1. **Critically evaluate the following claim, explaining whether it is either true or false and why.**

   "It is known that, in both countries A and B, the decision to purchase a home follows a standard probit specification; i.e.,

$$\Pr\left[y_i^k = 1 \mid X_{1i}^k, X_{2i}^k; \beta\right] = \Phi\left(\beta_0 + \beta_1 X_{1i}^k + \beta_2 X_{2i}^k\right) \qquad k = A, B$$

   where $\Phi(\bullet)$ denotes the standard normal cdf and the coefficients are known to be the same for the two countries. However, the proportion of homeowners in country A is 50%, while in country B only 25% of households own homes. Using this information, you can conclude that the marginal impact of a change in X_{1i}^k will be larger in country A than in country B."

   This statement is false. The basic idea here is that we know that the marginal effect of a variable, such as X_{1i}^k, is largest for individuals who *index* is close to zero (and hence with choice probabilities near 0.5). This is true since

$$\frac{\partial \Pr\left[y_i^k = 1 \mid X_{1i}^k, X_{2i}^k; \beta\right]}{\partial X_{1i}^k} = \phi\left(\beta_0 + \beta_1 X_{1i}^k + \beta_2 X_{2i}^k\right)\beta_1,$$

   which is largest when the index $\left(\beta_0 + \beta_1 X_{1i}^k + \beta_2 X_{2i}^k\right)$ is zero. The average marginal effect in country A is then given by:

$$\tfrac{1}{N_A}\sum_{i\in A}\frac{\partial \Pr\left[y_i^k = 1 \mid X_{1i}^k, X_{2i}^k; \beta\right]}{\partial X_{1i}^k} = \frac{\beta_1}{N_A}\sum_{i\in A}\phi\left(\beta_0 + \beta_1 X_{1i}^k + \beta_2 X_{2i}^k\right).$$

   Similarly, the average effect in country B is given by

$$\tfrac{1}{N_B}\sum_{i\in B}\frac{\partial \Pr\left[y_i^k = 1 \mid X_{1i}^k, X_{2i}^k; \beta\right]}{\partial X_{1i}^k} = \frac{\beta_1}{N_B}\sum_{i\in B}\phi\left(\beta_0 + \beta_1 X_{1i}^k + \beta_2 X_{2i}^k\right).$$

   However, knowing the proportion of homeowners in the two countries says nothing about these terms. For example, Country A might be equally divided between two types of individuals (those who absolutely hate owning homes and those who absolutely love living in homes). While this would lead to a home ownership rate of 50%, for these extreme types, the marginal effect of any variable is close to zero. On the other hand, country B might be made up entirely of individuals who are largely indifferent between homeownership and non-homeownership. In this case, the marginal effects will be relatively large, even though only 25% are actually observed to choose home ownership.

2. **It is illegal in the U.S. to discriminate on the basis of race in deciding whether or not to approve a home mortgage application. As an analyst, you decide to investigate this issue. Before proceeding, however, you want to be sure that your program to estimate your specified model is working properly, so you conduct a pseudo-data experiment. Specifically, you set up a data-generating process in which you assume that whether or not an individual's loan is approved depends upon the following factors:**

- **C_i: The household's credit score;**
- **R_i: The ratio of total monthly debt payments to total monthly income;**
- **D_{Ai}: A dummy variable that equals 1 if the applicant is an African-American and equals 0 otherwise; and**
- **D_{Pi}: A dummy variable that equals 1 if the applicant has any public record of credit problems (e.g., bankruptcy, etc.) and equals zero otherwise.**

You assume that there is a latent variable that determines loan approval, with

$$y_i^* = \beta_0 + \beta_C C_i + \beta_R R_i + \beta_A D_{Ai} + \beta_P D_{Pi} - \eta_i \qquad \textbf{(1)}$$

where η_i follows the standard normal distribution (i.e., $\eta_i \sim N(0,1)$). The observed discrete choice variable is given by.

$$y_i = \begin{cases} 1 & y_i^* > 0 \text{ (loan approved)} \\ 0 & y_i^* \le 0 \text{ (load denied)} \end{cases} \qquad \textbf{(2)}$$

Finally, assume that in the population:

- $C_i \sim TN_{(3.00,8.50)}(6.75, 0.64)$**; i.e. truncated normal (truncated below at 3.00 and above at 8.50.**
- $R_i \sim TN_{(0,0.9)}(0.25, 0.04)$
- $D_{Ai} = \begin{cases} 1 & \text{with probability } 0.15 \\ 0 & \text{with probability } 0.85 \end{cases}$
- $D_{Pi} = \begin{cases} 1 & \text{with probability } 0.10 \\ 0 & \text{with probability } 0.90 \end{cases}$

a) **Generate a sample of 1000 observations on y_i^*, y_i, C_i, R_i, D_{Ai} and D_{Pi} assuming that**

$$\begin{aligned} \beta_0 &= -10 \\ \beta_C &= 2 \\ \beta_R &= -1.5 \\ \beta_A &= -1.0 \\ \beta_P &= -0.8 \end{aligned}$$

Provide summary statistics (i.e., means, minimums, maximums, and standard deviations) for each of these variables.

The program at the end of this answer sheet generates the requisite data. The resulting summary statistics are:

	mean	std	min	max
y_i	0.9260	0.2619	0.0000	1.0000
C_i	6.7217	0.7864	4.2926	8.4976
R_i	0.2939	0.1753	0.0019	0.8151
D_{Ai}	0.1490	0.3563	0.0000	1.0000
D_{Pi}	0.1000	0.3002	0.0000	1.0000

b) Write a GAUSS program to estimate these parameters using the maximum likelihood routine MAXLIK. Your program does not have to provide fancy output, but I would like you to provide me with both a copy of your code and its output, with the relevant estimates highlighted.

Again, the attached program carries out the requested estimation for both the probit and logit specifications, with resulting parameter estimates:

Variable	Probit	Logit
Intecept	-8.716 (1.059)	-15.794 (1.925)
C_i	1.790 (0.184)	3.251 (0.339)
R_i	-1.825 (0.499)	-3.500 (0.956)
D_{Ai}	-0.795 (0.208)	-1.474 (0.380)
D_{Pi}	-0.768 (0.245)	-1.392 (0.460)

As we would expect given the different scaling underling the two models, the parameter estimates are different. However, the relative size and signs of the parameters are very similar across the two models, as is the statistical significance of the estimates.

c) Using your parameter estimates, compute point estimates and standard errors for the following:

- **The probability that the following "types" of individuals would have their loan approved:**
  - **Type I:** $C_i = 5.50, R_i = 0.6, D_{Ai} = 1, D_{Pi} = 1$
  - **Type II:** $C_i = 7.50, R_i = 0.2, D_{Ai} = 0, D_{Pi} = 0$
  - **Type III:** $C_i = 6.50, R_i = 0.3, D_{Ai} = 1, D_{Pi} = 0$

The fitted choice probabilities for the probit and logit models follow directly from the parameter estimates. The standard errors for these fitted choice probabilities are calculated in the attached code using a simulation procedure in which we (a) draw from the normal approximation to the sampling distribution for the ML estimators in each case, (b) calculated the fitted choice probabilities for each draw, and (c) calculate the standard deviation of these fitted probabilities across the draws. Doing so yields the following choice probabilities

Type	Probit	Logit
I	0.053 (0.034)	0.1717 (0.0858)
II	1.000 (<0.001)	0.9893 (0.0075)
III	0.935 (0.027)	0.8214 (0.0538)

As expected, Type I households have low probability of having their loan approved, as they have all of the characteristics associated loan denial: low credit score, past history of credit problems, a high debt to income ratio, and being African-American. Type II households are at the other end of the spectrum and, consequently, we find their predicted probability of loan approval to lie somewhere in between. The two models yield similar, though not exactly the same, predictions.

- **The marginal effect that the person's credit score has on the probability that each of the three "types" of individuals will have their loan approved.**

The marginal effects of credit score on the approval probability are given by

Type	Probit	Logit
I	0.201 (0.099)	0.316 (0.119)
II	<0.001 (<0.001)	0.025 (0.017)
III	0.242 (0.075)	0.336 (0.078)

The marginal effects are, as we saw in question 1, smallest for individuals at the extreme of the choice probabilities (in this case Type II's). Again, the logit and probit models yield similar implications.

- **The impact that a sudden public bad credit event would have on the** $\Pr\left[y_i = 1\right]$ **for Type II and Type III individuals.**

The impact of a sudden bad credit event for individuals of Type II and Type III are found by comparing the fitted choice probabilities for these individuals (show at the top of this page) with the fitted choice probabilities in which $D_{Pi} = 1$. The resulting change in the fitted probabilities are:

Type	Probit	Logit
II	<-0.001 (<0.001)	-0.0135 (0.0130)
III	-0.170 (0.073)	-0.1503 (0.0947)

- **The racial bias in the loan application decision implied by your model.**

Finally, this question asks you to assess the racial bias in the loan application decision implied by your model. There are several ways to do this. The way I do it is to compute the change in the loan probabilities for the entire sample when the racial indicator is D_{Pi} switched between 1 and 0. This holds constant for each individual in the sample *all other attributes.* The resulting bias estimated is given in the following tables

Probit	Logit
0.073 (0.026)	0.080 (0.049)

d) Using the *same* data, repeat parts (b) and (c) using a logit specification. Compare and contrast the implications of your probit and logit results.

The results for the logit model are integrated into the answers above.

GAUSS Code:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Problem Set #2 - Econ 673 - Fall 2008                                  @
@       Question 2                                                             @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Note: MAXLIK is a special routine and you have to tell GAUSS that your @
@             will be using it                                                 @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
library maxlik;
#include maxlik.ext;
maxset;

output  file = Prob2q2.out reset;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       The first step in this program is to generate the data                 @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
/*       specify parameter values */
beta0   = -10.0;
betac   = 2;
betar   = -1.5;
betaa   = -1.0;
betap   = -0.8;
beta    = beta0|betac|betar|betaa|betap;

/*       generate explanatory variables */
nobs    = 1000;
ac      = 3.0;
bc      = 8.5;
muc     = 6.75;
sigc    = 0.8;
uac     = cdfn((ac - muc)/sigc);
ubc     = cdfn((bc - muc)/sigc);
uc      = uac + (ubc - uac)*rndu(nobs,1);
c       = muc + sigc*cdfni(uc);                  /*credit score*/

ar      = 0;
br      = 0.9;
mur     = 0.25;
sigr    = 0.2;
uar     = cdfn((ar - mur)/sigr);
ubr     = cdfn((br - mur)/sigr);
ur      = uar + (ubr - uar)*rndu(nobs,1);
r       = mur + sigr*cdfni(ur);                  /*credit score*/

Da      = (rndu(nobs,1).>0.85);                /*college graduate*/

Dp      = (rndu(nobs,1).>0.90);                /*membership in env. group*/

X       = ones(nobs,1)~c~r~Da~Dp;
k       = cols(X);
ystar   = X*beta - rndn(nobs,1);
y       = (ystar.>0);
q       = 2*y - 1;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Generate Summary Statistics                                            @
```

```
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
z       = y~X;
meanz   = meanc(z)';
stdz    = stdc(z)';
maxz    = maxc(z)';
minz    = minc(z)';
print   ;
print   "summary statistics";
print   "*******************************************";
print   "     mean       std       min         max     ";
format  /m1 /rd 10,4;
length  = cols(meanz);
for     i (1,length,1);
        print meanz[1,i] stdz[1,i] minz[1,i] maxz[1,i];
endfor;
print   ;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Estimate the model using maxlik and probit specification               @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
b0      = beta;
b0      = {1,0,0,0,0};
dta     = y~X;
/*_max_GradProc = &gradprbt;*/
{b_ml,logl,g,varcov,ret}=maxprt(maxlik(dta,0,&prbt,b0));
stderr  = sqrt(diag(varcov));
print   ;
print   "maxlik results of estimation";
print   "************************************";
print   "  estimate     std. err.";
print   b_ml~stderr;


@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Compute fitted probabilities and marginal effects                      @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
riter   = 1000;
XA      = {1 5.5 0.6 1 1};
XB      = {1 7.5 0.2 0 0};
XC      = {1 6.5 0.3 1 0};
XB1     = {1 7.5 0.2 0 1};
XC1     = {1 6.5 0.3 1 1};
Xwhite  = ones(nobs,1)~c~r~Da~zeros(nobs,1);
Xblack  = ones(nobs,1)~c~r~Da~ones(nobs,1);
pfitA   = zeros(riter,1);
pfitB   = zeros(riter,1);
pfitC   = zeros(riter,1);
mfitA   = zeros(riter,1);
mfitB   = zeros(riter,1);
mfitC   = zeros(riter,1);
dfitB   = zeros(riter,1);
dfitC   = zeros(riter,1);
rfit    = zeros(riter,1);
betar   = beta' + rndn(riter,5) * chol(varcov);
for     i (1,riter,1);
        pfitA[i,1] = cdfn(XA*betar[i,.]');
        pfitB[i,1] = cdfn(XB*betar[i,.]');
        pfitC[i,1] = cdfn(XC*betar[i,.]');
        mfitA[i,1] = pdfn(XA*betar[i,.]')*betar[riter,2];
        mfitB[i,1] = pdfn(XB*betar[i,.]')*betar[riter,2];
```

```
        mfitC[i,1] = pdfn(XC*betar[i,.]')*betar[riter,2];
        dfitB[i,1] = cdfn(XB1*betar[i,.]') - pfitB[i,1];
        dfitC[i,1] = cdfn(XC1*betar[i,.]') - pfitC[i,1];
        rfit[i,1]  = meanc(cdfn(Xwhite*betar[i,.]'))
                   - meanc(cdfn(Xblack*betar[i,.]'));
endfor;
print   ;
print   "probit fitted probabilities";
print   "*************************************";
print   "  type     estimate     std. err. ";
print   "   I      " meanc(pfitA) stdc(pfitA);
print   "  II      " meanc(pfitB) stdc(pfitB);
print   " III      " meanc(pfitC) stdc(pfitC);

print   ;
print   "probit marginal Impact of credit score  ";
print   "*************************************";
print   "  type     estimate     std. err.";
print   "   I      " meanc(mfitA) stdc(mfitA);
print   "  II      " meanc(mfitB) stdc(mfitB);
print   " III      " meanc(mfitC) stdc(mfitC);

print   ;
print   "probit change in Pr(y=1) from a bad credit event";
print   "*******************************************";
print   "  type     estimate     std. err. ";
print   "  II      " meanc(dfitB) stdc(dfitB);
print   " III      " meanc(dfitC) stdc(dfitC);

conf   = 0.95;
print  ;
print  "probit estimated racial bias";
print  "*****************************************************";
print  "                      " 100*conf "percent confidence interval";
print  "estimate   std. err.   lower bound   upper bound";
srebate = sortc(rfit,1);
lb     = srebate[floor((1 - conf)*riter/2),1];
ub     = srebate[ceil((1 + conf)*riter/2),1];
print  meanc(rfit) stdc(rfit) lb ub;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                               @
@       Estimate the model using maxlik and probit specification                @
@                                                                               @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
b0      = beta;
b0      = {1,0,0,0,0};
dta     = y~X;
/*_max_GradProc = &gradlogit;*/
{b_ml,logl,g,varcov,ret}=maxprt(maxlik(dta,0,&logit,b0));
stderr  = sqrt(diag(varcov));
print   ;
print   "maxlik results of estimation";
print   "*************************************";
print   "  estimate     std. err.";
print   b_ml~stderr;


@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                               @
@       Compute fitted probabilities and marginal effects                       @
@                                                                               @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
riter   = 1000;
betar   = beta' + rndn(riter,5) * chol(varcov);
```

```
for     i (1,riter,1);
        pfitA[i,1] = 1./(1+exp(-XA*betar[i,.]'));
        pfitB[i,1] = 1./(1+exp(-XB*betar[i,.]'));
        pfitC[i,1] = 1./(1+exp(-XC*betar[i,.]'));
        mfitA[i,1] = exp(XA*betar[i,.]')./((1+exp(XA*betar[i,.]')).^2)
                     .*betar[riter,2];
        mfitB[i,1] = exp(XB*betar[i,.]')./((1+exp(XB*betar[i,.]')).^2)
                     .*betar[riter,2];
        mfitC[i,1] = exp(XC*betar[i,.]')./((1+exp(XC*betar[i,.]')).^2)
                     .*betar[riter,2];
        dfitB[i,1] = 1./(1+exp(-XB1*betar[i,.]')) - pfitB[i,1];
        dfitC[i,1] = 1./(1+exp(-XC1*betar[i,.]')) - pfitC[i,1];
        rfit[i,1]  = meanc(1./(1+exp(-Xwhite*betar[i,.]')))
                   - meanc(1./(1+exp(-Xblack*betar[i,.]')));
endfor;
print   ;
print   "logit fitted probabilities";
print   "**********************************";
print   "  type     estimate     std. err. ";
print   "   I      " meanc(pfitA) stdc(pfitA);
print   "  II      " meanc(pfitB) stdc(pfitB);
print   " III      " meanc(pfitC) stdc(pfitC);

print   ;
print   "logit Marginal Impact of credit score  ";
print   "**********************************";
print   "  type     estimate     std. err.";
print   "   I      " meanc(mfitA) stdc(mfitA);
print   "  II      " meanc(mfitB) stdc(mfitB);
print   " III      " meanc(mfitC) stdc(mfitC);

print   ;
print   "logit Change in Pr(y=1) from a bad credit event";
print   "****************************************";
print   "  type     estimate     std. err. ";
print   "  II      " meanc(dfitB) stdc(dfitB);
print   " III      " meanc(dfitC) stdc(dfitC);

conf   = 0.95;
print  ;
print  "logit estimated racial bias";
print  "**************************************************";
print  "                      " 100*conf "percent confidence interval";
print  "estimate   std. err.   lower bound   upper bound";
srebate = sortc(rfit,1);
lb     = srebate[floor((1 - conf)*riter/2),1];
ub     = srebate[ceil((1 + conf)*riter/2),1];
print  meanc(rfit) stdc(rfit) lb ub;
end;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                               @
@       Procedure to compute loglikelihood value for probit model              @
@                                                                               @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
proc    prbt(b,dta);
local   yvar, xvar, qvar, llik, nobs, k;
yvar    = dta[.,1];
qvar    = 2 * yvar - 1;
nobs    = rows(dta);
k       = cols(dta);
xvar    = dta[.,2:k];
llik    = ln(cdfn(qvar .* xvar * b));      /* Likelihood function for probit */
retp    (llik);
endp;
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Procedure to compute loglikelihood value for logit model               @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
proc    logit(b,dta);
local   yvar, xvar, qvar, llik, nobs, k;
yvar    = dta[.,1];
qvar    = 2 * yvar - 1;
nobs    = rows(dta);
k       = cols(dta);
xvar    = dta[.,2:k];
llik    = -ln(1+exp(-qvar .* xvar * b));      /* Likelihood function for probit
*/
retp    (llik);
endp;

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                                                                              @
@       Procedure to compute loglikelihood value for logit model               @
@                                                                              @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
proc    logt(b,dta);
local   yvar, xvar, qvar, arg, llik;
yvar    = dta[.,1];
qvar    = 2 * yvar - 1;
xvar    = dta[.,2:3];
arg     = exp(qvar .* xvar * b);
llik    = ln(arg./(1 + arg));      /* Likelihood function for probit */
retp    (llik);
endp;
```