7 APPENDIX

The IML programs for examples are given in the Appendix and appear under the appropriate chapter heading and example number.

CHAPTER 2

EXAMPLE 2.1

```
proc iml; reset nolog;
y={80, 15, 5};
b={80,0.1875};
diff=1;
j=0;
do while (diff>0.000001); j=j+1;
  q=j(2,1,1);
  q[1]=-(1+b[2]+b[2]*b[2])+y[+]/b[1];
  q[2]=-b[1]*(1+2*b[2])+(y[2]+2*y[3])/b[2];
  H=j(2,2,1);
  H[1,1]=-y[+]/(b[1]*b[1]); H[1,2]=-(1+2*b[2]);
  H[2,1]=H[1,2]; H[2,2]=-2*b[1]-(y[2]+2*y[3])/(b[2]*b[2]);
  b1=b-inv(H)*q;
  diff=(b-b1)`*(b-b1);
  b=b1;
end;
m=j(3,1,0);
m[1]=b[1]; m[2]=b[1]*b[2]; m[3]=b[1]*b[2]*b[2];
print j b m;
```

EXAMPLE 2.2

```
proc iml; reset nolog;
y={80, 15, 5};
x={1 0, 1 1, 1 2};
b=ginv(x`*x)*x`*y;
b={80,0.1875};
diff=1;
j=0;
do while (diff>0.000001); j=j+1;
  m=exp(x*b);
  b1=b+ginv(x`*diag(m)*x)*x`*(y-m);
  diff=sqrt((b-b1)`*(b-b1));
  b=b1;
end;
m=exp(x*b); print j b m;
```

EXAMPLE 2.3

```
proc iml; reset nolog;
y={80, 15, 5}; ybegin=y;
b={80,0.1875};
diff=1;
j=0;
do while (diff>0.000001); j=j+1;
  q=j(2,1,1);
  q[1]=-(1+b[2]+b[2]*b[2])+ybegin[+]/b[1];
  q[2]=-b[1]*(1+2*b[2])+(ybegin[2]+2*ybegin[3])/b[2];
  Inf=j(2,2,1);
  Inf[1,1]=y[+]/(b[1]*b[1]); Inf[1,2]=(1+2*b[2]);
  Inf[2,1]=Inf[1,2]; Inf[2,2]=2*b[1]+(y[2]+2*y[3])/(b[2]*b[2]);
  b1=b+inv(Inf)*q;
  diff=sqrt((b-b1)`*(b-b1));
  b=b1;
  y[1]=b[1]; y[2]=b[1]*b[2]; y[3]=b[1]*b[2]*b[2];
end;
m=j(3,1,0); m[1]=b[1]; m[2]=b[1]*b[2]; m[3]=b[1]*b[2]*b[2];
print j b m;
```

EXAMPLE 2.9

```
proc iml; reset nolog;
Gm=j(1,3,0);  Gy=j(1,3,0);
y={80,15,5}; ybegin=y; m=y; muhat=y;
i=0; j=0;
diff1=1; diff2=1;
do while (diff1>0.000001);
i=i+1; j=0;
diff2=1;
Dm=diag(m);
Gm[1]=m[3]; Gm[2]=-2*m[2]; Gm[3]=m[1];
y=ybegin;
  do while (diff2>0.000001);
  j=j+1;
  g=y[1]*y[3]-y[2]*y[2];
  Gy[1]=y[3]; Gy[2]=-2*y[2]; Gy[3]=y[1];
  muhat=y-(Gm*Dm)`*ginv(Gy*Dm*Gm`)*g;
  diff2=sqrt((muhat-y)`*(muhat-y));
  y=muhat;
  end;
diff1=sqrt((muhat-m)`*(muhat-m));
m=muhat;
end;
print i j m;
```

EXAMPLE 2.10

```
proc iml; reset nolog;
Gy=j(1,3,0);
y={80,15,5};
j=0;
diff1=1;
do while (diff1>0.000001);
  j=j+1;
  Gy[1]=1/y[1]; Gy[2]=-2/y[2]; Gy[3]=1/y[3];
  GmDm={1 -2 1};
  g=log(y[1]*y[3]/(y[2]*y[2]));
  muhat=y-GmDm`*ginv(Gy*GmDm`)*g;
  diff1=sqrt((muhat-y)`*(muhat-y));
  y=muhat;
end;
print j y;
```

or

```
proc iml; reset nolog;
y={80, 15, 5};      m=y;
x={1 0, 1 1, 1 2};
p=i(3)-x*ginv(x`*x)*x`;
diff=1;
j=0;
do while (diff>0.000001); j=j+1;
  idy=inv(diag(y));
  muhat=y-p*ginv(p*idy*p)*p*log(y);
  diff=sqrt((muhat-y)`*(muhat-y));
  y=muhat;
end;
print j muhat;
```

EXAMPLE 2.11

```
proc iml;
reset nolog;
yobs={125,18,20,34}; mu=yobs;
x={1 -1 -1 -3,
   0  1 -1  0};
diff=1; r=0;
do while (diff>1e-10);
r=r+1;
v=diag(mu)-(1/197)#mu*mu`;
mu1=yobs-(x*v)`*ginv(x*v*x`)*x*yobs;
diff=(mu-mu1)`*(mu-mu1);
mu=mu1;
end;
print r mu;
pi=mu[4]/197*4; print pi;
```

CHAPTER 3

EXAMPLE 3.1 : Proc Catmod for reduced Loglinear model

```
data verdict;
input m v f n @@;
cards;
1 1 1 42   1 1 2 23
1 2 1  4   1 2 2 11
2 1 1 79   2 1 2 65
2 2 1 12   2 2 2 41
3 1 1 32   3 1 2 17
3 2 1  8   3 2 2 24
;
proc catmod;
weight n;
model m*v*f=_response_/ml nogls noprofile pred=freq;
loglin m v f m*v v*f;
run;
```

CATMOD PROCEDURE

Response: M*V*F	Response Levels (R)= 12
Weight Variable: N	Populations (S)= 1
Data Set: VERDICT	Total Frequency (N)= 358
Frequency Missing: 0	Observations (Obs)= 12

MAXIMUM-LIKELIHOOD ANALYSIS

Iteration	Sub Iteration	-2 Log Likelihood	Convergence Criterion
0	0	1779.1932	1.0000
1	0	1621.7743	0.0885
2	0	1590.2147	0.0195
3	0	1590.0846	0.0000819
4	0	1590.0846	1.2263E-8
5	0	1590.0846	4.29E-16

Parameter Estimates

Iteration	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0
1	-0.3296	0.6508	0.4413	-0.0112	-0.0223	0.3212	0.2793
2	-0.4090	0.6050	0.5376	-0.1947	0.2463	0.007680	0.3846
3	-0.4219	0.6068	0.5518	-0.1941	0.2509	0.0178	0.3823
4	-0.4221	0.6067	0.5520	-0.1941	0.2512	0.0178	0.3823
5	-0.4221	0.6067	0.5520	-0.1941	0.2512	0.0178	0.3823

MAXIMUM-LIKELIHOOD ANALYSIS-OF-VARIANCE TABLE

Source	DF	Chi-Square	Prob
M	2	55.92	0.0000
V	1	56.51	0.0000
F	1	8.50	0.0036
M*V	2	8.60	0.0135
V*F	1	32.99	0.0000
LIKELIHOOD RATIO	4	2.81	0.5898

ANALYSIS OF MAXIMUM-LIKELIHOOD ESTIMATES

Effect	Parameter	Estimate	Standard Error	Chi-Square	Prob
M	1	-0.4221	0.1062	15.81	0.0001
	2	0.6067	0.0811	55.92	0.0000
V	3	0.5520	0.0734	56.51	0.0000
F	4	-0.1941	0.0666	8.50	0.0036
M*V	5	0.2512	0.1062	5.60	0.0180
	6	0.0178	0.0811	0.05	0.8266
V*F	7	0.3823	0.0666	32.99	0.0000

MAXIMUM-LIKELIHOOD PREDICTED VALUES FOR RESPONSE FUNCTIONS AND FREQUENCIES

Sample	M	V	F	Function Number	Observed Function	Observed Standard Error	Predicted Function	Predicted Standard Error	Residual
1				1	0.55961579	0.25588316	0.46056655	0.22902508	0.09904924
				2	-0.0425596	0.29179604	0.08408898	0.23545775	-0.1266486
				3	-1.7917595	0.54006172	-1.9103652	0.3908212	0.11860574
				4	-0.7801586	0.36410954	-0.7576857	0.31291637	-0.0224729
				5	1.19139402	0.23307701	1.25599258	0.20979113	-0.0645986
				6	0.99633344	0.2388541	0.87951501	0.21679525	0.11681843
				7	-0.6931472	0.35355339	-0.6481235	0.32394827	-0.0450237
				8	0.53551824	0.25701539	0.50455601	0.22387033	0.03096223
				9	0.28768207	0.27003086	0.17799958	0.2397416	0.10968249
				10	-0.3448405	0.31700189	-0.198478	0.24589408	-0.1463625
				11	-1.0986123	0.40824829	-1.1526795	0.23414645	0.05406722
	1	1	1	F1	42	6.0887294	38.5465116	4.76034564	3.45348837
	1	1	2	F2	23	4.63921829	26.4534884	3.57260801	-3.4534884
	1	2	1	F3	4	1.98879543	3.6	1.11274377	0.4
	1	2	2	F4	11	3.26527352	11.4	2.95150493	-0.4
	2	1	1	F5	79	7.84646666	85.3953488	7.04763999	-6.3953488
	2	1	2	F6	65	7.29371812	58.6046512	5.80126523	6.39534884
	2	2	1	F7	12	3.4055492	12.72	2.77929216	-0.72
	2	2	2	F8	41	6.02531902	40.28	5.58608559	0.72
	3	1	1	F9	32	5.39811678	29.0581395	4.13758176	2.94186047
	3	1	2	F10	17	4.02402006	19.9418605	3.04155594	-2.9418605
	3	2	1	F11	8	2.79664604	7.68	1.88314117	0.32
	3	2	2	F12	24	4.73191943	24.32	4.32421628	-0.32

EXAMPLE 3.1 : ML Estimation with the Newton-Raphson algorithm

```
proc iml;
reset nolog;
y={42, 23, 4, 11, 79, 65, 12, 41, 32, 17, 8, 24};
x={1     1     0     1     1     1     0     1,
   1     1     0     1    -1     1     0    -1,
   1     1     0    -1     1    -1     0    -1,
   1     1     0    -1    -1    -1     0     1,
   1     0     1     1     1     0     1     1,
   1     0     1     1    -1     0     1    -1,
   1     0     1    -1     1     0    -1    -1,
   1     0     1    -1    -1     0    -1     1,
   1    -1    -1     1     1    -1    -1     1,
   1    -1    -1     1    -1    -1    -1    -1,
   1    -1    -1    -1     1     1     1    -1,
   1    -1    -1    -1    -1     1     1     1};
m=y;
b=ginv(x`*x)*x`*log(m);
m=exp(x*b);
diff=1; i=0;

do while (diff>1e-15); i=i+1;
b1=b+ginv(x`*diag(m)*x)*x`*(y-m);
diff=(b-b1)`*(b-b1);
b=b1;
m=exp(x*b);
end;

sebhat=sqrt(vecdiag(ginv(x`*diag(m)*x)));
print i b sebhat;
```

EXAMPLE 3.1 : ML Estimation under constraints

```
proc iml;
reset nolog;
y={42, 23, 4, 11, 79, 65, 12, 41, 32, 17, 8, 24}; ybegin=y;
m=y;
x={1   1   0   1   1   1   0   1   0   1   1   0,
   1   1   0   1  -1   1   0  -1   0  -1  -1   0,
   1   1   0  -1   1  -1   0   1   0  -1  -1   0,
   1   1   0  -1  -1  -1   0  -1   0   1   1   0,
   1   0   1   1   1   0   1   0   1   1   0   1,
   1   0   1   1  -1   0   1   0  -1  -1   0  -1,
   1   0   1  -1   1   0  -1   0   1  -1   0  -1,
   1   0   1  -1  -1   0  -1   0  -1   1   0   1,
   1  -1  -1   1   1  -1  -1  -1  -1   1  -1  -1,
   1  -1  -1   1  -1  -1  -1   1   1  -1   1   1,
   1  -1  -1  -1   1   1   1  -1  -1  -1   1   1,
   1  -1  -1  -1  -1   1   1   1   1   1  -1  -1};

c={0 0 0 0 0 0 0 1 0 0 0 0,
   0 0 0 0 0 0 0 0 1 0 0 0,
   0 0 0 0 0 0 0 0 0 0 1 0,
   0 0 0 0 0 0 0 0 0 0 0 1};
acp=c*ginv(x`*x)*x`; ac=acp`;
gy=ac`*log(y);
wald=gy`*ginv(ac`*diag(1/y)*ac)*gy;
diff=1; i=0;

do while (diff>1e-10);
y1=y-ac*ginv(ac`*diag(1/y)*ac)*ac`*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;
bhat=ginv(x`*x)*x`*log(y);
print bhat;

v=diag(y)-y*y`/y[+];
v=diag(y);
covy=v-ac*ginv(ac`*diag(1/y)*ac)*ac`;
se_y=sqrt(vecdiag(covy));
print y se_y;

est1=diag(1/y)*covy*diag(1/y);
cov_bhat=ginv(x`*x)*x`*est1*x*ginv(x`*x);
se_bhat=sqrt(vecdiag(cov_bhat));
print bhat se_bhat;

chi2=sum((ybegin-y)#(ybegin-y)/y);
dev=2#ybegin`*log(ybegin/y); print dev;
print chi2 dev wald;
```

EXAMPLE 3.2 : Proc Logistic and Proc Genmod

```
data blood;
input pressure ypres yabs;
events=ypres;
trials=ypres+yabs;
cards;
111.5  3 153
121.5 17 235
131.5 12 272
141.5 16 255
151.5 12 127
161.5  8  77
176.5 16  83
191.5  8  35
;
proc logistic;
model events/trials=pressure;
run;

proc genmod;
model events/trials=pressure/link=logit dist=bin;
run;
```

```
                        The LOGISTIC Procedure

  Data Set: WORK.BLOOD
  Response Variable (Events): EVENTS
  Response Variable (Trials): TRIALS
  Number of Observations: 8
  Link Function: Logit
```

Response Profile

Ordered Value	Binary Outcome	Count
1	EVENT	92
2	NO EVENT	1237

Model Fitting Information and Testing Global Null Hypothesis BETA=0

Criterion	Intercept Only	Intercept and Covariates	Chi-Square for Covariates
AIC	670.831	648.718	.
SC	676.024	659.102	.
-2 LOG L	668.831	644.718	24.113 with 1 DF (p=0.0001)
Score	.	.	26.556 with 1 DF (p=0.0001)

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
INTERCPT	1	-6.0820	0.7243	70.5098	0.0001	.	.
PRESSURE	1	0.0243	0.00484	25.2523	0.0001	0.269349	1.025

Association of Predicted Probabilities and Observed Responses

Concordant	= 56.8%	Somers' D	= 0.273
Discordant	= 29.5%	Gamma	= 0.316
Tied	= 13.7%	Tau-a	= 0.035
(113804 pairs)		c	= 0.636

The GENMOD Procedure

Model Information

Description	Value
Data Set	WORK.BLOOD
Distribution	BINOMIAL
Link Function	LOGIT
Dependent Variable	EVENTS
Dependent Variable	TRIALS
Observations Used	8
Number Of Events	92
Number Of Trials	1329

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	6	5.9092	0.9849
Scaled Deviance	6	5.9092	0.9849
Pearson Chi-Square	6	6.2899	1.0483
Scaled Pearson X2	6	6.2899	1.0483
Log Likelihood	.	-322.3590	.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Std Err	ChiSquare	Pr>Chi
INTERCEPT	1	-6.0820	0.7243	70.5076	0.0001
PRESSURE	1	0.0243	0.0048	25.2513	0.0001

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Std Err	ChiSquare	Pr>Chi
SCALE	0	1.0000	0.0000	.	.

NOTE: The scale parameter was held fixed.

EXAMPLE 3.2 : ML Estimation using the Newton-Raphson algorithm

```
proc iml;
reset nolog;
x={1 111.5, 1  121.5, 1  131.5, 1  141.5, 1  151.5, 1 161.5, 1  176.5, 1 191.5};
y={3 153,   17 235,   12 272,   16 255,   12 127,    8 77,    16 83,    8 35};
xr=nrow(x);

yi=y[,1]; yi0=yi;
ni=y[,1]+y[,2];
pi=yi/ni;
e=j(xr,1,1);

logit=log(pi/(e-pi));
bhat=ginv(x`*x)*x`*logit;

diff=1;  i=0;
do while (diff>1e-10); i=i+1;
  pi=exp(x*bhat)/(e+exp(x*bhat));
  var=ni#pi#(e-pi); v=diag(var); ivar=1/var;
  yi1=ni#pi;
  bhat1=bhat+ginv(x`*v*x)*x`*(yi-yi1);
  diff=(bhat-bhat1)`*(bhat-bhat1);
  bhat=bhat1;
end;

sebhat=sqrt(vecdiag(ginv(x`*v*x)));
print i bhat sebhat;
```

EXAMPLE 3.2 : ML Estimation under constraints

```
proc iml;
reset nolog;
x={1 111.5, 1  121.5, 1  131.5, 1  141.5, 1  151.5, 1 161.5, 1  176.5, 1 191.5};
y={3 153,    17 235,  12 272,   16 255,   12 127,   8 77,     16 83,    8 35};
xr=nrow(x);
p=i(xr)-x*ginv(x`*x)*x`;

yi=y[,1]; yi0=yi;
ni=y[,1]+y[,2];
e=j(xr,1,1);

diff=1; i=0;
do while (diff>1e-10); i=i+1;
  pi=yi/ni;
  logit=log(pi/(e-pi));
  var=ni#pi#(e-pi); v=diag(var); ivar=1/var;
  g=p*diag(ivar);
  yi1=yi-p*ginv(p*diag(ivar)*p)*p*logit;
  diff=(yi1-yi)`*(yi1-yi);
  yi=yi1;
end;

bhat=ginv(x`*x)*x`*logit;
sebhat=sqrt(vecdiag(ginv(x`*v*x)));
print i yi0 yi1;
print bhat sebhat;

pi=yi/ni;
var=ni#pi#(e-pi); v=diag(var); iv=diag(1/var);
covy=v-p*ginv(p*iv*p)*p;
se_y=sqrt(vecdiag(covy));

est1=iv*covy*iv;
cov_bhat=ginv(x`*x)*x`*est1*x*ginv(x`*x);
se_bhat=sqrt(vecdiag(cov_bhat));
print bhat se_bhat;

chi2=sum((yi0-yi)#(yi0-yi)/yi)+sum((yi-yi0)#(yi-yi0)/(ni-yi));
dev=2#yi0`*log(yi0/yi)+2#(ni-yi0)`*log((ni-yi0)/(ni-yi));

print chi2 dev;
```

EXAMPLE 3.3: Proc Catmod, Proc Logistic and Proc Genmod

```
data verdict;
input m v f n @@;
cards;
1 1 1 42   1 1 2 23
1 2 1  4   1 2 2 11
2 1 1 79   2 1 2 65
2 2 1 12   2 2 2 41
3 1 1 32   3 1 2 17
3 2 1  8   3 2 2 24
;
proc catmod;
weight n;
model v=m f/ml nogls noprofile;
run;

data verdict;
input m1 m2 f1 guilty n_guilty @@;
events=guilty;
trials=guilty+n_guilty;
cards;
 1  0  1 42 4
 0  1  1 79 12
-1 -1  1 32 8
 1  0 -1 23 11
 0  1 -1 65 41
-1 -1 -1 17 24
;
proc logistic;
model events/trials=m1 m2 f1;
run;

proc genmod;
model events/trials=m1 m2 f1/link=logit dist=bin;
run;
```

The CATMOD Procedure

Data Summary

Response	v	Response Levels	2
Weight Variable	n	Populations	6
Data Set	VERDICT	Total Frequency	358
Frequency Missing	0	Observations	12

Maximum Likelihood Analysis

Iteration	Sub Iteration	-2 Log Likelihood	Convergence Criterion	Parameter Estimates 1	2	3	4
0	0	496.29338	1.0000	0	0	0	0
1	0	382.97715	0.2283	0.8530	0.3128	0.1136	0.5613
2	0	378.42388	0.0119	1.0465	0.4339	0.1224	0.7443
3	0	378.34289	0.000214	1.0776	0.4548	0.1211	0.7732
4	0	378.34285	1.0572E-7	1.0783	0.4553	0.1210	0.7739
5	0	378.34285	2.749E-14	1.0783	0.4553	0.1210	0.7739

Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	53.91	<.0001
m	2	8.38	0.0152
f	1	32.61	<.0001
Likelihood Ratio	2	0.26	0.8801

Analysis of Maximum Likelihood Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		1.0783	0.1469	53.91	<.0001
m	1	0.4553	0.2226	4.18	0.0408
	2	0.1210	0.1717	0.50	0.4809
f	1	0.7739	0.1355	32.61	<.0001

The LOGISTIC Procedure

Data Set: WORK.VERDICT
Response Variable (Events): EVENTS
Response Variable (Trials): TRIALS
Number of Observations: 6
Link Function: Logit

Response Profile

Ordered Value	Binary Outcome	Count
1	EVENT	258
2	NO EVENT	100

Model Fitting Information and Testing Global Null Hypothesis BETA=0

Criterion	Intercept Only	Intercept and Covariates	Chi-Square for Covariates
AIC	426.100	386.343	.
SC	429.981	401.865	.
-2 LOG L	424.100	378.343	45.758 with 3 DF (p=0.0001)
Score	.	.	43.571 with 3 DF (p=0.0001)

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate	Odds Ratio
INTERCPT	1	1.0783	0.1469	53.9107	0.0001	.	.
M1	1	0.4553	0.2226	4.1840	0.0408	0.168558	1.577
M2	1	0.1210	0.1717	0.4968	0.4809	0.054754	1.129
F1	1	0.7739	0.1355	32.6053	0.0001	0.427234	2.168

```
Association of Predicted Probabilities and Observed Responses
          Concordant = 62.6%          Somers' D = 0.434
          Discordant = 19.2%          Gamma     = 0.531
          Tied       = 18.2%          Tau-a     = 0.175
          (25800 pairs)               c         = 0.717


                    The GENMOD Procedure

                     Model Information
         Description                    Value
         Data Set                       WORK.VERDICT
         Distribution                   BINOMIAL
         Link Function                  LOGIT
         Dependent Variable             EVENTS
         Dependent Variable             TRIALS
         Observations Used              6
         Number Of Events               258
         Number Of Trials               358


              Criteria For Assessing Goodness Of Fit
       Criterion                DF          Value       Value/DF
       Deviance                  2         0.2554         0.1277
       Scaled Deviance           2         0.2554         0.1277
       Pearson Chi-Square        2         0.2552         0.1276
       Scaled Pearson X2         2         0.2552         0.1276
       Log Likelihood            .      -189.1714              .


                  Analysis Of Parameter Estimates
    Parameter   DF    Estimate     Std Err    ChiSquare   Pr>Chi
    INTERCEPT    1      1.0783      0.1469      53.9106   0.0001
    M1           1      0.4553      0.2226       4.1840   0.0408
    M2           1      0.1210      0.1717       0.4968   0.4809
    F1           1      0.7739      0.1355      32.6053   0.0001
    SCALE        0      1.0000      0.0000            .        .

NOTE:  The scale parameter was held fixed.
```

EXAMPLE 3.3 : ML Estimation under constraints and using the Newton-Raphson algorithm

```
proc iml;
reset nolog;
x={1  1  0  1,
   1  0  1  1,
   1 -1 -1  1,
   1  1  0 -1,
   1  0  1 -1,
   1 -1 -1 -1};
y={42 4, 79 12, 32 8, 23 11, 65 41, 17 24};
xr=nrow(x);

yi=y[,1]; yi0=yi;
ni=y[,1]+y[,2];
pi=yi/ni; pi0=pi;
e=j(xr,1,1);

print 'ML ESTIMATION SUBJECT TO CONSTRAINTS';
p=i(xr)-x*ginv(x`*x)*x`;
diff=1; i=0;
do while (diff>1e-10); i=i+1;
  pi=yi/ni;
  logit=log(pi/(e-pi));
  var=ni#pi#(e-pi); v=diag(var); ivar=1/var;
  g=p*diag(ivar);
  yi1=yi-p*ginv(p*diag(ivar)*p)*p*logit;
  diff=(yi1-yi)`*(yi1-yi);
  yi=yi1;
end;

bhat=ginv(x`*x)*x`*logit;
sebhat=sqrt(vecdiag(ginv(x`*v*x)));
print i yi0 yi1; print bhat sebhat;

pi=yi/ni; var=ni#pi#(e-pi); v=diag(var); iv=diag(1/var);
covy=v-p*ginv(p*iv*p)*p; se_y=sqrt(vecdiag(covy));

est1=iv*covy*iv;
cov_bhat=ginv(x`*x)*x`*est1*x*ginv(x`*x); se_bhat=sqrt(vecdiag(cov_bhat));
print bhat se_bhat;

chi2=sum((yi0-yi)#(yi0-yi)/yi)+sum((yi-yi0)#(yi-yi0)/(ni-yi));
dev=2#yi0`*log(yi0/yi)+2#(ni-yi0)`*log((ni-yi0)/(ni-yi));
print chi2 dev;

print 'NEWTON-RAPHSON ALGORITHM';
logit=log(pi0/(e-pi0)); bhat=ginv(x`*x)*x`*logit;

diff=1;  i=0;
do while (diff>1e-10); i=i+1;
  pi=exp(x*bhat)/(e+exp(x*bhat));
  var=ni#pi#(e-pi); v=diag(var); ivar=1/var;
  yi1=ni#pi;
  bhat1=bhat+ginv(x`*v*x)*x`*(yi-yi1);
  diff=(bhat-bhat1)`*(bhat-bhat1);
  bhat=bhat1;
end;

sebhat=sqrt(vecdiag(ginv(x`*v*x)));
print i bhat sebhat;
```

CHAPTER 4

EXAMPLE 4.1

```
proc iml; reset nolog;
/************************************************************/
/* Give the observed values of y from the square table     */
/************************************************************/
y={50  45   8  18   8
   28 174  84 154  55
   11  78 110 223  96
   14 150 185 714 447
    3  42  72 320 411};
/* y={11607 100 366 124 87 13677 515 302 172 225 17819 270 63 176 286 10192}; */
/* y={1520 266 124 66 234 1512 432 78 117 362 1772 205 36 82 179 492}; */

y=y`; ybeg=y;

/************************************************************/
/* Create C matrix for the test under constraints          */
/************************************************************/
n=sqrt(nrow(y)); nn=n#(n-1)/2;
C=j(nn,n*n,0);
r=0;
do j=1 to (n-1);
k1begin=(j-1)*(n+1)+2; k1end=n*j;
lc=0;
  do k1=k1begin to k1end;
  lc=lc+1;
  r=r+1; k2=k1+(n-1)*lc;
  C[r,k1]=1; C[r,k2]=-1;
  end;
end;

/***************************************************/
/* 1 Test for CS model under constraints          */
/***************************************************/
print 'Model CS';
x=j(nn,1,1);
P=I(nn)-x*ginv(x`*x)*x`;
K=P*C;
diff=1;
i=0;

do while (diff>1e-10); i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(x`*x)*x`*C*log(y));
print delta chi2 g2;
print ybeg y;
```

```
/*************************************************/
/* 2 Test for S model under constraints          */
/*************************************************/
print 'Model S';
y=ybeg;
diff=1;
i=0;

do while (diff>1e-10);
i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-C`*ginv(C*Di*C`)*C*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
print chi2 g2;
print ybeg y;

/*************************************************/
/* 3 Test for DPS model under constraints        */
/*************************************************/
print 'Model DPS';
y=ybeg;

X=I(n-1);
do h=2 to n-1;
YY=I(n-h)||j(n-h,h-1,0);
X=X//YY;
free YY;
end; print X;

P=I(nn)-X*ginv(X`*X)*X`;
K=P*C;

diff=1;
i=0;

do while (diff>1e-10);
i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(X`*X)*X`*C*log(y));
print delta chi2 g2;
print ybeg y;
```

```
/************************************************/
/* 4 Test for LDPS model under constraints      */
/************************************************/
print 'Model LDPS';
y=ybeg;

X1=I(n-1);
do h=2 to n-1;
YY=I(n-h)||j(n-h,h-1,0);
X1=X1//YY;
free YY;
end;
L=1;
do h=2 to n-1;
L=L//h;
end;
X=X1*L; print X;

P=I(nn)-X*ginv(X`*X)*X`;
K=P*C;

diff=1;
i=0;

do while (diff>1e-10);
i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(X`*X)*X`*C*log(y));
print delta chi2 g2;
print ybeg y;

/************************************************/
/* 5 Test for ALDPS model under constraints     */
/************************************************/
print 'Model ALDPS';
y=ybeg;

X1=I(n-1);
do h=2 to n-1;
YY=I(n-h)||j(n-h,h-1,0);
X1=X1//YY;
free YY;
end;
L=1;
do h=2 to n-1;
L=L//h;
end;
```

```
X=j(2*n,1,n)-X1*L; print X;

P=I(nn)-X*ginv(X`*X)*X`;
K=P*C;

diff=1;
i=0;

do while (diff>1e-10);
i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(X`*X)*X`*C*log(y));
print delta chi2 g2;
print ybeg y;

/***************************************************/
/* 6 Test for 2RPS model under constraints        */
/***************************************************/
print 'Model 2RPS';
y=ybeg;

X1=I(n-1);
do h=2 to n-1;
YY=I(n-h)||j(n-h,h-1,0);
X1=X1//YY;
free YY;
end;
L=0;
do h=1 to n-2;
L=L//h;
end;
X2=X1*L; X3=j(n*2,1,1); X=X3||X2; print X;

P=I(nn)-X*ginv(X`*X)*X`;
K=P*C;

diff=1; i=0;

do while (diff>1e-10); i=i+1;
Dy=diag(y); Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;
chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(X`*X)*X`*C*log(y));
print delta chi2 g2;
print ybeg y;
```

```
/************************************************/
/* 7 Test for QS model under constraints        */
/************************************************/
print 'Model QS';
y=ybeg;

free X;
plusi=I(n-1); mini=-plusi;
een=j(n-1,1,1); mineen=-een;
X=een||mini||mineen||plusi;

do k=1 to n-2;
nul=j(n-k-1,k,0);
plusi=I(n-k-1); mini=-plusi;
een=j(n-k-1,1,1); mineen=-een;
YY=nul||een||mini||nul||mineen||plusi;
X=X//YY;
free YY;
end;


P=I(nn)-X*ginv(X`*X)*X`;
K=P*C;

diff=1;
i=0;

do while (diff>1e-10);
i=i+1;
Dy=diag(y);
Di=inv(Dy);
y1=y-K`*ginv(K*Di*K`)*K*log(y);
diff=(y1-y)`*(y1-y);
y=y1;
end;

chi2=(ybeg-y)`*((1/y)#(ybeg-y));
g2=2*ybeg`*log(ybeg/y);
delta=exp(ginv(X`*X)*X`*C*log(y));
print delta chi2 g2;
print ybeg y;
```

CHAPTER 5

EXAMPLE 5.2 and EXAMPLE 5.3

```
/********************************************************/
/* ML estimation of cell probabilities for incomplete   */
/* IxJ contingency tables if data is missing on either  */
/* categories and the missing data mecahnism is         */
/* ignorable                                            */
/********************************************************/

proc iml; reset nolog;

/**********************************************************/
/* ENTER FREQUENCY VECTORS A, B and C:                    */
/* A: both row and column categories observed             */
/*    enter rowwise                                       */
/* B: row category observed and column category missing   */
/* C: column category observed and row category missing   */
/**********************************************************/

/***************/
/* Example 5.2 */
/***************/
A={392,55,76,38};
B={33,9};
C={31,7};

/***************/
/* Example 5.3 */
/***************/
A={287,39,38,18,6,4,91,22,23};
B={279,27,201};
C={59,18,26};

i=nrow(B);
j=nrow(C);
na=nrow(A); nb=i; nc=j;

y=A//B//C;
sy=y[+];
ya=y[1:na,]; yb=y[na+1:na+nb,]; yc=y[na+nb+1:na+nb+nc,];
som_ya=ya[+]; som_yb=yb[+]; som_yc=yc[+];
pa=ya/ya[+]; pb=yb/yb[+]; pc=yc/yc[+];
p=pa//pb//pc; tot=p[+];
p0=p; pbegin=p;

ij=i+j;
ej=J(1,j,1); ei=J(1,i,1);
i_i=I(i);    i_j=I(j);
i_ij=-I(ij);
c_row=i_i@ej;
c_col=ei@i_j;
g1=c_row//c_col;
G=g1||i_ij;
```

```
diff=1; t=0;

do while (diff>1e-20); t=t+1;

  pa=p[1:na,]; pb=p[na+1:na+nb,]; pc=p[na+nb+1:na+nb+nc,];
  cova=diag(pa)/som_ya-pa*pa`/som_ya;
  covb=diag(pb)/som_yb-pb*pb`/som_yb;
  covc=diag(pc)/som_yc-pc*pc`/som_yc;
  V=block(cova,covb,covc);

  p=pbegin;
    print g; print p;
  gp=G*p;
  pt=p-(G*V)`*ginv(G*V*G`)*gp;
  diff=(pt-p0)`*(pt-p0);
  p0=pt;
  p=pt;

end;

stderr=sqrt(vecdiag(v-(g*v)`*ginv(g*v*g`)*g*v));
print pt stderr;
```

EXAMPLE 5.2: GENMOD

```
data one;
input count p11 p12 p21 off;
cards;
392  561    0    0   0
 55    0  561    0   0
 76    0    0  561   0
 38 -561 -561 -561 561
 33   42   42    0   0
  9  -42  -42    0  42
 31   38    0   38   0
  7  -38    0  -38  38
;
proc genmod data=one;
model count=p11 p12 p21/dist=poi link=id offset=off noint;
run;
```

The GENMOD Procedure

Model Information

Description	Value
Data Set	WORK.ONE
Distribution	POISSON
Link Function	IDENTITY
Dependent Variable	COUNT
Offset Variable	OFF
Observations Used	8

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	5	0.1125	0.0225
Scaled Deviance	5	0.1125	0.0225
Pearson Chi-Square	5	0.1149	0.0230
Scaled Pearson X2	5	0.1149	0.0230
Log Likelihood	.	2642.6805	.

Analysis Of Parameter Estimates

Parameter	DF	Estimate	Std Err	ChiSquare	Pr>Chi
INTERCEPT	0	0.0000	0.0000	.	.
P11	1	0.6971	0.0187	1389.0323	0.0001
P12	1	0.0986	0.0124	63.5830	0.0001
P21	1	0.1358	0.0141	92.2715	0.0001
SCALE	0	1.0000	0.0000	.	.

NOTE: The scale parameter was held fixed.

Lagrange Multiplier Statistics

Parameter	ChiSquare	Pr>Chi
Intercept	0.0309	0.8605

EXAMPLE 5.3: GENMOD

```
data one;
input count p11 p12 p13 p21 p22 p23 p31 p32 off;
cards;
287   528     0     0     0     0     0     0     0     0
39      0   528     0     0     0     0     0     0     0
38      0     0   528     0     0     0     0     0     0
18      0     0     0   528     0     0     0     0     0
6       0     0     0     0   528     0     0     0     0
4       0     0     0     0     0   528     0     0     0
91      0     0     0     0     0     0   528     0     0
22      0     0     0     0     0     0     0   528     0
23   -528  -528  -528  -528  -528  -528  -528  -528   528
279   507   507   507     0     0     0     0     0     0
27      0     0     0   507   507   507     0     0     0
201  -507  -507  -507  -507  -507  -507     0     0   507
59    103     0     0   103     0     0   103     0     0
18      0   103     0     0   103     0     0   103     0
26   -103  -103     0  -103  -103     0  -103  -103   103
;
proc genmod data=one;
model count=p11 p12 p13 p21 p22 p23 p31 p32/dist=poi link=id offset=off noint;
run;
```

```
                        The GENMOD Procedure
                         Model Information
              Description                   Value
              Data Set                      WORK.ONE
              Distribution                  POISSON
              Link Function                 IDENTITY
              Dependent Variable            COUNT
              Offset Variable               OFF
              Observations Used             15

               Criteria For Assessing Goodness Of Fit
         Criterion              DF         Value       Value/DF
         Deviance                7       36.0006         5.1429
         Scaled Deviance         7       36.0006         5.1429
         Pearson Chi-Square      7       36.8259         5.2608
         Scaled Pearson X2       7       36.8259         5.2608
         Log Likelihood          .     4471.6801              .

                    Analysis Of Parameter Estimates
       Parameter    DF     Estimate     Std Err    ChiSquare   Pr>Chi
       INTERCEPT     0       0.0000      0.0000            .        .
       P11           1       0.4747      0.0174     748.0998   0.0001
       P12           1       0.0701      0.0102      47.1384   0.0001
       P13           1       0.0742      0.0107      47.7219   0.0001
       P21           1       0.0327      0.0064      25.9330   0.0001
       P22           1       0.0120      0.0045       6.9284   0.0085
       P23           1       0.0087      0.0041       4.4891   0.0341
       P31           1       0.2060      0.0158     169.1698   0.0001
       P32           1       0.0558      0.0106      28.0104   0.0001
       SCALE         0       1.0000      0.0000            .        .
NOTE:  The scale parameter was held fixed.
                     Lagrange Multiplier Statistics
                     Parameter     ChiSquare   Pr>Chi
                     Intercept        0.0571   0.8111
```

EXAMPLE 5.3 (Fully Classified cases)

```
data wheeze;
input smoke status f @@;
cards;
1 1 287 1 2 39 1 3 38
2 1  18 2 2  6 2 3  4
3 1  91 3 2 22 3 3 23
;
proc catmod data=wheeze;
weight f;
model smoke*status= _response_/ml noprofile pred=prob;
loglin smoke status smoke*status;
run;
```

CATMOD PROCEDURE

MAXIMUM-LIKELIHOOD PREDICTED VALUES FOR RESPONSE FUNCTIONS AND PROBABILITIES

				-------Observed-------		-------Predicted------		
			Function		Standard		Standard	
Sample	SMOKE	STATUS	Number	Function	Error	Function	Error	Residual
1			1	2.523988	0.21670852	2.523988	0.2167086	0
			2	0.52806743	0.26290547	0.52806743	0.26290557	0
			3	0.50209194	0.26418564	0.50209194	0.26418573	0
			4	-0.2451225	0.31469639	-0.2451225	0.31469648	0
			5	-1.3437347	0.45841567	-1.3437347	0.45840475	-1.8143E-9
			6	-1.7491999	0.54173634	-1.7491999	0.5417358	0
			7	1.37536529	0.23338224	1.37536529	0.23338233	0
			8	-0.0444518	0.29821604	-0.0444518	0.29821615	0
	1	1	P1	0.54356061	0.02167697	0.54356061	0.02167697	0
	1	2	P2	0.07386364	0.01138245	0.07386364	0.01138246	0
	1	3	P3	0.0719697	0.01124706	0.0719697	0.01124706	0
	2	1	P4	0.03409091	0.00789715	0.03409091	0.00789715	0
	2	2	P5	0.01136364	0.00461275	0.01136364	0.00461261	0
	2	3	P6	0.00757576	0.0037735	0.00757576	0.0037735	0
	3	1	P7	0.17234848	0.01643654	0.17234848	0.01643655	0
	3	2	P8	0.04166667	0.00869632	0.04166667	0.00869633	0
	3	3	P9	0.04356061	0.00888298	0.04356061	0.00888298	0

EXAMPLE 5.4: Model {SPC}

```
proc iml; reset nolog;
yc={3,176,4,293,17,197,2,23}; som_yc=yc[+]; pc=yc/som_yc;
ym={10,150,5,90};             som_ym=ym[+]; pm=ym/som_ym;
G={1 0 0 0 1 0 0 0 -1  0  0  0,
   0 1 0 0 0 1 0 0  0 -1  0  0,
   0 0 1 0 0 0 1 0  0  0 -1  0,
   0 0 0 1 0 0 0 1  0  0  0 -1};

y=yc//ym;
p=pc//pm; p0=p; pbegin=p;

diff=1; t=0;

do while (diff>1e-20); t=t+1;
 pc=p[1:8,]; pm=p[9:12,];
 covc=diag(pc)/som_yc-pc*pc`/som_yc;
 covm=diag(pm)/som_ym-pm*pm`/som_ym;
 V=block(covc,covm);
 p=pbegin;
 gp=G*p;
 pt=p-(G*V)`*ginv(G*V*G`)*gp;
 diff=(pt-p0)`*(pt-p0);
 p0=pt; p=pt;
end;
pc=100*pt[1:8,]; print pc;
```

EXAMPLE 5.4: (ML estimation with EM algorithm: Model SC PC)

```
proc iml;
reset nolog;
/************************************/
/* design matrix: S P C SP SC PC SPC */
/************************************/
X={1  1  1  1  1  1  1  1,
   1 -1  1  1 -1 -1  1 -1,
   1  1 -1  1 -1  1 -1 -1,
   1 -1 -1  1  1 -1 -1  1,
   1  1  1 -1  1 -1 -1 -1,
   1 -1  1 -1 -1  1 -1  1,
   1  1 -1 -1 -1 -1  1  1,
   1 -1 -1 -1  1  1  1 -1};

/** model:SP,SC,PC **/  ah=X[,8];
/** model:SP,SC    **/  ah=X[,7:8];
/** model:SC,PC    **/  ah=X[,5]||X[,8];
y={3,176,4,293,17,197,2,23,10,150,5,90};

ya=y[1:8,];  na=ya[+];  pa=ya/na;     yabeg=ya; ya1=ya;

diff2=1;  r=0;
do while (diff2>1e-10);
diff1=1;

/*******************************************************/
/* First iteration:   Starting values of EM algorithm */
/* Higher iterations: M-Step of EM algorithm          */
/*******************************************************/
do while (diff1>1e-20);
  yt=ya-ah*ginv(ah`*diag(1/ya)*ah)*ah`*log(ya);
  diff1=(yt-ya)`*(yt-ya);
  ya=yt;
end;

/************************************/
/* E-Step of EM algorithm          */
/************************************/
r=r+1;
pa=ya/ya[+];
pfill=j(2,2,1)@i(4)*pa;
yb=j(2,1,1)@y[9:12,];
ya=yabeg+yb#pa/pfill;
ya2=ya;

diff2=(ya2-ya1)`*(ya2-ya1);
ya1=ya2;
end;
print r pa;

sig=diag(ya)-ah*ginv(ah`*diag(1/ya)*ah)*ah`;
cov=ginv(X`*X)*X`*(diag(1/ya)*sig*diag(1/ya))*X*ginv(X`*X);
var=diag(cov);

pa=100*pa;
/*print pa var;*/
```

EXAMPLE 5.4: (ML estimation under constraints: Model CS CP)

```
proc iml; reset nolog;
y={3,176,4,293,17,197,2,23,10,150,5,90}; ybegin=y; y0=y; n=y[+]; mu0=y;
ya=y[1:8,];  na=ya[+];
yb=y[9:12,]; nb=yb[+];
i=2; j=2; k=2; jk=j*k; ijk=i*j*k;
x={1  1  1  1  1  1  1  1,
   1 -1  1  1 -1 -1  1 -1,
   1  1 -1  1 -1  1 -1 -1,
   1 -1 -1  1  1 -1 -1  1,
   1  1  1 -1  1 -1 -1 -1,
   1 -1  1 -1 -1  1 -1  1,
   1  1 -1 -1 -1 -1  1  1,
   1 -1 -1 -1  1  1  1 -1};
xu=x[,1:4]||x[,6:7];
p1=i(8)-xu*ginv(xu`*xu)*xu`;
cr=(1/na)#j(1,i,1)@i(jk);

diff1=1; j1=0;
diff2=1; j2=0;

do while (diff1>1e-10); j1=j1+1; j2=0; diff2=1;
 ya=y[1:8,]; yb=y[9:12,];
 cov=diag(y)-1/n#y*y`;
 gmu1=(p1*diag(1/ya))||j(8,4,0);
 gmu2=cr||((-1/nb)#i(jk));
 gmu=gmu1//gmu2;
 y=ybegin;

do while (diff2>1e-10); j2=j2+1;
 ya=y[1:8,]; yb=y[9:12,];
 g1=p1*log(ya);
 g2=(cr||((-1/nb)#i(jk)))*y;
 g=g1//g2;
 gy1=(p1*diag(1/ya))||j(ijk,jk,0);
 gy2=cr||((-1/nb)#i(jk));
 gy=gy1//gy2;

 yt=y-(gmu*cov)`*ginv(gy*cov*gmu`)*g;
 diff2=(yt-y0)`*(yt-y0);
 y0=yt;
 y=yt;
end;

mut=yt;
diff1=(mut-mu0)`*(mut-mu0);
mu0=mut;
end;

ya=y[1:8,];
pa=ya/na; pb=yb/nb;
print i j pa pb;
```

```
cov=diag(y)-1/n#y*y`;
gmu1=(p1*diag(1/ya))||j(ijk,jk,0);
gmu2=cr||((-1/nb)#i(jk));
gmu=gmu1//gmu2;

sig=sqrt(1/na#1/na#vecdiag(cov-(gmu*cov)`*ginv(gmu*cov*gmu`)*gmu*cov));
sig=sig[1:8,];
p=p[1:8,];
print ybegin yt pa sig;
```

I	J	PA	PB
2	2	0.0049631	0.0317501
		0.254203	0.538353
		0.0075794	0.010518
		0.3882079	0.4193789
		0.026787	
		0.28415	
		0.0029385	
		0.0311711	

YBEGIN	YT	PA	SIG
3	3.5486225	0.0049631	0.0015509
176	181.75517	0.254203	0.0155327
4	5.4193025	0.0075794	0.002343
293	277.56862	0.3882079	0.0158979
17	19.152686	0.026787	0.0051063
197	203.16723	0.28415	0.0160086
2	2.1010385	0.0029385	0.0007932
23	22.287327	0.0311711	0.0061783
10	8.0962709		
150	137.28002		
5	2.6820797		
90	106.94163		