References

Agresti, A. (1984) *Analysis of ordinal categorical data.* New York: Wiley.

Agresti, A. (1984) *Categorical data analysis.* New York: Wiley.

Bishop, Y.V.V., Fienberg S.E. and Holland, P.W. (1975) *Discrete multivariate analysis.* Cambridge, MA: MIT Press.

Central Statistical Service, Republic of South Africa (1985). *Survey of household expenditure.* report No. 01-11-02(1985).

Elderton, W.P. and Johnson, N.L. (1969) *Systems of frequency curves.* Cambridge University Press.

Johnson, N.L. and Kotz, S. (1970) *Continuous univariate distributions-1. Distributions in statistics.* John Wiley & Sons

Joubert, H.M. and Crowther, N.A.S. (1992) Fitting of distributions in the case of survey data. *HSRC Report WS-50.*

Kendall, M.G. and Stuart, A. (1958) *The advanced theory of statistics.* Griffin.

Matthews, G.B. and Crowther, N.A.S. (1995) A maximum likelihood estimation procedure when modelling in terms of constraints. *S.A Statist. J.*, **29**, *29-51.*

Matthews, G.B. and Crowther, N.A.S. (1997) A maximum likelihood estimation procedure when modelling categorical data in terms of cross-product ratios. *S.A Statist. J.*, **31**, *161-184.*

Matthews, G.B. and Crowther, N.A.S. (1998) A maximum likelihood estimation procedure for the generalized linear model with restrictions. *S.A Statist. J.*, **32**, *119-144.*

Muirhead, R.J. (1982) *Aspects of Multivariate statistical theory.* John Wiley & Sons

Neter, J., Kutner, M.H., Nachtsheim, C.J. and Wasserman, W. (1996) *Applied linear statistical models.* Mc Graw-Hill.

Pearson, K. (1924). On the mean error of frequency distributions. *Biometrika*, **16**, *198-200*

Part V

Appendix

Appendix A

SAS programs: Part I

A.1 EXP1.SAS

```
proc iml worksize= 60;
f={17,14,31,26,12}; n=f[+];
x={12.5,25,50,100};
C={1 0 0 0,
   1 1 0 0,
   1 1 1 0,
   1 1 1 1};
CI=inv(C);
k=nrow(f); k1=k-1;
v1=J(k1,1,1);
Px=x*inv(x‘*x)*x‘;
p=C*f[1:k1]/n;
i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
  i=i+1; pi=p; p=p0;
  V=(C*diag(CI*pi)*C‘-pi*pi‘)/n;
  thetapi=-(x‘*log(v1-pi))/(x‘*x); mupi=1/thetapi;
  Dpi=-inv(diag(v1-pi));
```

```
  Gpi=-diag(exp(-thetapi*x))*Px*Dpi - I(k1);
  j=0; diff=1;
  do while (diff > 1e-9);
    j=j+1; pv=p;
    thetap=-(x'*log(v1-p))/(x'*x); mup=1/thetap;
    Dp=-inv(diag(v1-p));
    Gp=-diag(exp(-thetap*x))*Px*Dp - I(k1);
    g=(v1-exp(-thetap*x))-p;
    print i j g pi p thetapi mupi thetap mup;
    p=p-(Gpi*V)'*ginv(Gp*V*Gpi')*g;
    diff=sqrt((p-pv)'*(p-pv));
  end;
  diff1=sqrt((p-pi)'*(p-pi));
end;
```

A.2 EXP2.SAS

```
proc iml worksize= 60;
f={17,14,31,26,12}; n=f[+];
x={12.5,25,50,100};
k=nrow(f); k1=k-1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
p=C*f[1:k1]/n;
v1=J(k1,1,1);
Q=I(k1)-x*inv(x'*x)*x';
i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
    i=i+1; pi=p; p=p0;
    Dpi=inv(diag(pi-v1));
    Gpi=Q*Dpi;
    V=(C*diag(CI*pi)*C'-pi*pi')/n;
    j=0; diff=1;
```

```
    do while (diff > 1e-9);
        j=j+1; pv=p;
        Dp=inv(diag(p-v1));
        Gp=Q*Dp;
        g=Q*log(v1-p);
        print i j p pi;
        p=p-(Gpi*V)‘*ginv(Gp*V*Gpi‘)*g;
        diff=sqrt((p-pv)‘*(p-pv));
        if i=1 & j=1 then do;
            Wald=g‘*ginv(Gp*V*Gp‘)*g;
            GpV=Gp*V;
            df=trace(GpV*ginv(GpV‘*GpV)*GpV‘);
            discr=wald/n;
        end;
    end;
    diff1=sqrt((p-pi)‘*(p-pi));
end;
Cov_pi=V-(Gpi*V)‘*ginv(Gpi*V*Gpi‘)*(Gpi*V);
theta=-(x‘*log(v1-pi))/(x‘*x);
Var_theta=((x‘*Dpi)/(x‘*x))*Cov_pi*((x‘*Dpi)/(x‘*x))‘;
mu=1/theta;
SE_mu=sqrt(Var_theta/(theta**4));

e=(CI*pi*n)//(n-(CI*pi*n)[+]);
Pearson=(((f-e)##2)/e)[+];
P_pvalue=1-probchi(Pearson,df);
W_pvalue=1-probchi(Wald,df);
print mu SE_mu Pearson P_pvalue Wald W_pvalue df;
```

A.3 EXPSIM.SAS

```
proc iml;
rep=1000; n=100; theta0=50;
matrix=J(rep,4,0);
x={12.5,25,50,100};
xl=0//x;
xu=x//250;
mid=(xl+xu)/2;
mlb=J(n,1,1)@xl‘;
mub=J(n,1,1)@xu‘;
k=nrow(xu); k1=k-1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
Q=I(k1)-x*inv(x‘*x)*x‘;
do r=1 to rep;
  y=theta0#ranexp(J(n,1,r));
  my=y@J(k,1,1)‘;
  t=((my>mlb)=(my<=mub));
  f=t[+,]‘;
  p=C*f[1:k1]/n;
  i=0; p0=p; diff1=1;
  do while (diff1 > 1e-9);
    i=i+1; pi=p; p=p0;
    Dpi=inv(diag(pi-v1));
    Gpi=Q*Dpi;
    V=(C*diag(CI*pi)*C‘-pi*pi‘)/n;
    j=0; diff=1;
    do while (diff > 1e-9);
      j=j+1; pv=p;
      Dp=inv(diag(p-v1));
      Gp=Q*Dp;
      g=Q*log(v1-p);
```

```
      p=p-(Gpi*V)‘*ginv(Gp*V*Gpi‘)*g;
      diff=sqrt((p-pv)‘*(p-pv));
      if i=1 & j=1 then do;
        Wald=g‘*ginv(Gp*V*Gp‘)*g;
        GpV=Gp*V;
        df=trace(GpV*ginv(GpV‘*GpV)*GpV‘);
        pvalue=1-probchi(Wald,df);
        discr=wald/n;
      end;
    end;
    diff1=sqrt((p-pi)‘*(p-pi));
  end;
  theta=-(x‘*log(v1-pi))/(x‘*x);
  mu=1/theta;
  Cov_pi=V-(Gpi*V)‘*ginv(Gpi*V*Gpi‘)*(Gpi*V);
  Var_theta=((x‘*Dpi)/(x‘*x))*Cov_pi*((x‘*Dpi)/(x‘*x))‘;
  SE_mu=sqrt(Var_theta/(theta**4));
  e=(CI*pi*n)//(n-(CI*pi*n)[+]);
  Pearson=(((f-e)##2)/e)[+];
  matrix[r,1]=mu;
  matrix[r,2]=SE_mu;
  matrix[r,3]=Pearson;
  matrix[r,4]=Wald;
end;
create d from matrix[colname={’mu’ ’SE_mu’ ’Pearson’ ’Wald’}];
append from matrix;

proc means data=d n mean std p5 p50 p95;
var mu SE_mu wald;
run;

proc univariate data=d normal plot;
var Pearson;
output out=pp pctlpts=5 10 25 50 75 90 95 pctlpre=pp;
```

```
run;

proc univariate data=d normal plot;
var Wald;
output out=pw pctlpts=5 10 25 50 75 90 95 pctlpre=pw;
run;

proc print data=pp;
run;
proc print data=pw;
run;
```

A.4 NORM1.SAS

```
proc iml worksize= 60;
f={9,26,24,27,14}; n=f[+];
x={40,50,60,75};
k=nrow(f); k1=k-1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
XD=x||J(k1,1,-1);
XXX=inv(XD‘*XD)*XD‘;
Px=XD*inv(XD‘*XD)*XD‘;
p=C*f[1:k1]/n;
i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
  i=i+1; pi=p; p=p0;
  V=(C*diag(CI*pi)*C‘-pi*pi‘)/n;
  alphapi=XXX*probit(pi);
  mupi=alphapi[2]/alphapi[1]; sigmapi=1/alphapi[1];
  zpi=XD*alphapi;
  Dpi=inv(diag(pdf(’normal’,probit(pi))));
```

```
  Gpi=diag(pdf('normal',zpi))*Px*Dpi - I(k1);
  j=0; diff=1;
  do while (diff > 1e-9);
    j=j+1; pv=p;
    alphap=XXX*probit(p);
    mup=alphap[2]/alphap[1]; sigmap=1/alphap[1];
    zp=XD*alphap;
    Dp=inv(diag(pdf('normal',probit(p))));
    Gp=diag(pdf('normal',zp))*Px*Dp - I(k1);
    g=probnorm(zp)-p;
    print alphap i j g pi[format=6.4] p[format=6.4] mupi sigmapi mup sigmap;
    p=p-(Gpi*V)`*ginv(Gp*V*Gpi`)*g;
    diff=sqrt((p-pv)`*(p-pv));
  end;
  diff1=sqrt((p-pi)`*(p-pi));
end;
```

A.5 NORM2.SAS

```
proc iml worksize= 60;
f={9,26,24,27,14}; n=f[+];
x={40,50,60,75};
n=f[+];
k=nrow(f); k1=k-1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
p=C*f[1:k1]/n;
v1=J(k1,1,1);
XD=x||J(k1,1,-1);
XXX=inv(XD`*XD)*XD`;
Px=XD*inv(XD`*XD)*XD`;
Q=I(k1)-Px;
```

```
*** Theoretical value ***;
*p=(probnorm((x-58)/15));
***;

i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
  i=i+1; pi=p; p=p0;
  Dpi=inv(diag(pdf('normal',probit(pi))));
  Gpi=Q*Dpi;
  V=(C*diag(CI*pi)*C`-pi*pi`)/n;
  j=0; diff=1;
  do while (diff > 1e-9);
    j=j+1; pv=p;
    Dp=inv(diag(pdf('normal',probit(p))));
    Gp=Q*Dp;
    g=Q*probit(p);
    p=p-(Gpi*V)`*ginv(Gp*V*Gpi`)*g;
    print i j p pi;
    diff=sqrt((p-pv)`*(p-pv));
    if i=1 & j=1 then do;
      Wald=g`*ginv(Gp*V*Gp`)*g;
      GpV=Gp*V;
      df=trace(GpV*ginv(GpV`*GpV)*GpV`);
      discr=wald/n;
    end;
  end;
  diff1=sqrt((p-pi)`*(p-pi));
end;
Cov_pi=V-(Gpi*V)`*ginv(Gpi*V*Gpi`)*(Gpi*V);
alpha=XXX*probit(pi);
Cov_alpha=(XXX*Dpi)*Cov_pi*(XXX*Dpi)`;
mu=alpha[2]/alpha[1]; sigma=1/alpha[1];
print mu sigma;
beta=mu//sigma;
```

```
B=J(2,2,0);
B[1,1]=-alpha[2]/((alpha[1])**2);
B[1,2]=1/(alpha[1]);
B[2,1]=-1/((alpha[1])**2);
Cov_beta=B*Cov_alpha*B';
SE_beta=sqrt(diag(Cov_beta));

e=(CI*pi*n)//(n-(CI*pi*n)[+]);
Pearson=(((f-e)##2)/e)[+];
P_pvalue=1-probchi(Pearson,df);
W_pvalue=1-probchi(Wald,df);
print beta Cov_beta SE_beta, mu sigma, Pearson P_pvalue Wald W_pvalue df;

probitp=probit(p);
Pprobitp=Px*probitp;
Qprobitp=Q*probitp;
print probitp [format=9.7] Pprobitp[format=9.7] Qprobitp[format=9.7];
```

A.6 NORMSIM.SAS

```
proc iml worksize= 60;
rep=1000; n=100; mu0=58; sigma0=15; x={40,50,60,75};
matrix=J(rep,8,0);
k1=nrow(x); k=k1+1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
XD=x||J(k1,1,-1);
XXX=inv(XD'*XD)*XD';
Px=XD*inv(XD'*XD)*XD';
Q=I(k1)-Px;
xl=0//x; xu=x//100;
mlb=J(n,1,1)@xl'; mub=J(n,1,1)@xu';
```

```
start data;
  sp=mu0*J(n,1,1)+sigma0*rannor(J(n,1,r));
  xbar=sp[+]/n;
  xstd=sqrt(sp‘*sp/n-xbar**2);
  sss=sp@J(1,k,1);
  t=(sss>mlb)=(sss<=mub);
  f=t[+,]‘;
  p=C*f[1:k1]/n;
finish;

start fit;
  i=0; p0=p; diff1=1;
  do while (diff1 > 1e-9);
    i=i+1; pi=p; p=p0;
    Dpi=inv(diag(pdf(’normal’,probit(pi))));
    Gpi=Q*Dpi;
    V=(C*diag(CI*pi)*C‘-pi*pi‘)/n;
    j=0; diff=1;
    do while (diff > 1e-9);
      j=j+1; pv=p;
      Dp=inv(diag(pdf(’normal’,probit(p))));
      Gp=Q*Dp;
      g=Q*probit(p);
      p=p-(Gpi*V)‘*ginv(Gp*V*Gpi‘)*g;
      diff=sqrt((p-pv)‘*(p-pv));
      if i=1 & j=1 then do;
        Wald=g‘*ginv(Gp*V*Gp‘)*g;
        GpV=Gp*V;
        df=trace(GpV*ginv(GpV‘*GpV)*GpV‘);
        discr=wald/n;
      end;
    end;
    diff1=sqrt((p-pi)‘*(p-pi));
```

```
  end;
  Cov_pi=V-(Gpi*V)'*ginv(Gpi*V*Gpi')*(Gpi*V);
  alpha=XXX*probit(pi);
  Cov_alpha=(XXX*Dpi)*Cov_pi*(XXX*Dpi)';
  mu=alpha[2]/alpha[1]; sigma=1/alpha[1];
  beta=mu//sigma;
  B=J(2,2,0);
  B[1,1]=-alpha[2]/((alpha[1])**2);
  B[1,2]=1/(alpha[1]);
  B[2,1]=-1/((alpha[1])**2);
  Cov_beta=B*Cov_alpha*B';
  SE_beta=diag(sqrt(diag(Cov_beta)));

  e=(CI*pi*n)//(n-(CI*pi*n)[+]);
  Pearson=(((f-e)##2)/e)[+];
  P_pvalue=1-probchi(Pearson,df);
  W_pvalue=1-probchi(Wald,df);

  matrix[r,1]=xbar;
  matrix[r,2]=xstd;
  matrix[r,3]=mu;
  matrix[r,4]=(SE_beta[1,1]);
  matrix[r,5]=sigma;
  matrix[r,6]=(SE_beta[2,2]);
  matrix[r,7]=Pearson;
  matrix[r,8]=Wald;
finish;

do r=1 to rep;
  run data;
  run fit;
end;

create d from
```

```
  matrix[colname={'xbar' 'xstd' 'mu' 'SE_mu' 'sigma' 'SE_sigma' 'Pearson' 'Wald'}];
append from matrix;

proc means data=d maxdec=3 n mean std p5 p50 p95;
var xbar xstd mu SE_mu sigma SE_sigma;
run;

proc means data=d maxdec=4 p5 p10 p25 p50 p75 p90 p95;
var Pearson Wald;
run;
```

A.7 FIT.SAS

```
proc iml worksize= 60;
*************************;
*    Exponential ='E'   *;
*    Normal      ='N'   *;
*    Weibull     ='W'   *;
*    Log-logistic='L'   *;
*    Pareto      ='P'   *;
*************************;

*===>; distr='W';
*===>; f={9,37,67,63,30}; x={40,75,125,175}; x=x-0.5;

n=f[+];
k=nrow(f); k1=k-1;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
p=C*f[1:k1]/n;

start X;
```

```
      if distr='E' then XD=-x;
      if distr='N' then XD=x||(-v1);
      if distr='W' then XD=log(x)||(-v1);
      if distr='L' then XD=log(x)||v1;
      if distr='P' then XD=-log(x)||v1;
finish;

start h;
      if distr='E' then h=log(v1-p);
      if distr='N' then h=probit(p);
      if distr='W' then h=log(-log(v1-p));
      if distr='L' then h=log(p/(v1-p));
      if distr='P' then h=log(v1-p);
finish;

start D(Dp,p) global(distr,v1);
      if distr='E' then Dp=inv(diag(p-v1));
      if distr='N' then Dp=inv(diag(pdf('normal',probit(p))));
      if distr='W' then Dp=-inv(diag(log(v1-p)))*inv(diag(v1-p));
      if distr='L' then Dp=inv(diag(p))+inv(diag(v1-p));
      if distr='P' then Dp=-inv(diag(v1-p));
finish;

start beta;
      if distr='E' then beta=1/alpha;
      if distr='N' then do;
            beta[1]=alpha[2]/alpha[1];
            beta[2]=1/alpha[1];
      end;
      if (distr='W' | distr='P') then do;
            beta[1]=alpha[1];
            beta[2]=exp(alpha[2]/alpha[1]);
      end;
      if distr='L' then beta=alpha;
```

```
finish;

start B;
      if distr='E' then B=-1/(alpha**2);
      if distr='N' then do;
            B[1,1]=-alpha[2]/((alpha[1])**2);
            B[1,2]=1/(alpha[1]);
            B[2,1]=-1/((alpha[1])**2);
      end;
      if (distr='W' | distr='P') then do;
            B[1,1]=1;
            B[2,1]=-(alpha[2])/((alpha[1])**2)*exp((alpha[2])/(alpha[1]));
            B[2,2]=inv(alpha[1])*exp((alpha[2])/(alpha[1]));
      end;
      if distr='L' then B=I(nrow(alpha));
finish;

start wald;
      Wald=g`*ginv(Gp*V*Gp`)*g;
      GpV=Gp*V;
      df=trace(GpV*ginv(GpV`*GpV)*GpV`);
finish;

start mu;
      if distr='E' then mu=beta;
      if distr='N' then mu=beta[1];
      if distr='W' then mu=beta[2]*(gamma(1+1/beta[1]));
      if distr='L' then mu=exp(-beta[2]/beta[1])
                           *gamma(1+1/beta[1])*gamma(1-1/beta[1]);
      if distr='P' then mu=(beta[1]*beta[2])/(beta[1]-1);
finish;

start sigma;
      if distr='E' then sigma=beta;
```

```
      if distr='N' then sigma=beta[2];
      if distr='W' then sigma=sqrt(beta[2]**2
                           *(gamma(1+2/beta[1])-(gamma(1+1/beta[1]))**2));
      if distr='L' then sigma=sqrt(exp(-2*beta[2]/beta[1])
                           *gamma(1+2/beta[1])*gamma(1-2/beta[1])
                           - (gamma(1+1/beta[1])*gamma(1-1/beta[1]))**2));
      if distr='P' then sigma=sqrt((beta[1]*beta[2]**2)
                           /((beta[1]-1)**2*(beta[1]-2)));
finish;

run X;
Q=I(k1)-XD*inv(XD`*XD)*XD`;

i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
      i=i+1; pi=p; p=p0;
      run D(Dpi,pi);
      Gpi=Q*Dpi;
      V=(C*diag(CI*pi)*C`-pi*pi`)/n;
      j=0; diff=1;
      do while (diff > 1e-9);
            j=j+1; pv=p;
            run D(Dp,p);
            run h;
            Gp=Q*Dp;
            g=Q*h;
            print i j p pi g;
            p=p-(Gpi*V)`*ginv(Gp*V*Gpi`)*g;
            diff=sqrt((p-pv)`*(p-pv));
            if i=1 & j=1 then run wald;
      end;
      diff1=sqrt((p-pi)`*(p-pi));
end;
Cov_pi=V-(Gpi*V)`*ginv(Gpi*V*Gpi`)*(Gpi*V);
```

```
alpha=inv(XD‘*XD)*XD‘*h;
Cov_alpha=(inv(XD‘*XD)*XD‘*Dpi)*Cov_pi*(inv(XD‘*XD)*XD‘*Dpi)‘;
SE_alpha=sqrt(diag(Cov_alpha)*J(nrow(alpha),1,1));
print alpha Cov_alpha SE_alpha;

beta=J(nrow(alpha),1,0); run beta;
B=J(nrow(alpha),nrow(alpha),0); run B;
Cov_beta=B*Cov_alpha*B‘;
SE_beta=sqrt(diag(Cov_beta)*J(nrow(beta),1,1));
print beta Cov_beta SE_beta;

run mu; run sigma;
print mu sigma;

e=(CI*pi*n)//(n-(CI*pi*n)[+]);
Pearson=(((f-e)##2)/e)[+];
P_pvalue=1-probchi(Pearson,df);
W_pvalue=1-probchi(Wald,df);
discr=wald/n;
print Pearson P_pvalue Wald W_pvalue df discr;
```

Appendix B

SAS programs: Part II

B.1 FACTOR1.SAS

```
data d;
set phdabc.wisk;
if jaar=2003 & vlak=1 & wisk in('A','B','C','D','E') & 0<=finaal<=108;
maths=wisk;
if 0<=eksamen<40   then stats=40;
if 40<=eksamen<50  then stats=50;
if 50<=eksamen<60  then stats=60;
if 60<=eksamen<75  then stats=75;
if 75<=eksamen<=108 then stats=108;
keep maths stats;
run;

proc freq data=d noprint;
tables maths / out=factor1;
tables stats / out=class;
tables maths*stats / out=freq;
run;
```

```
*** Start: Empty cells ***;
data t;
maths='A'; stats=39; count=0;
output;
run;
data freq; set freq t;
run;
proc sort data=freq;
by maths stats;
run;
*** Finish: Empty cells ***;

proc transpose data=freq out=freq prefix=c;
by maths;
var count;
run;

proc iml worksize=200 symsize=2000;
use freq; read all var{c1 c2 c3 c4 c5} into freq;
use class; read all var{stats} into class;
use factor1; read all var{maths} into factor1;

n=freq[+];
nt=nrow(freq);
k=nrow(freq); k1=k-1;
x=class[1:k1]; x=x-0.5;
nn=freq[,+];
f=colvec(freq[,1:k1]);  f=f<>0.0001;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
po=inv(diag(nn)@I(k1))*f;
p=(I(nt)@C)*po;
print freq factor1 class x;
```

```
XD=x||-v1;
XXX=inv(XD‘*XD)*XD‘; XXX1=XXX[1,]; XXX2=XXX[2,];
Px=XD*inv(XD‘*XD)*XD‘;
Q=I(k1)-Px;
nor=(I(nt)@Q);

H=J(nt-1,1,1)||-I(nt-1);
var=H*(I(nt)@XXX1);

nfac1=nrow(factor1);
*Yar={2,1,0,-1,-2};                    *<=== Factor A: ordinal ***;
Yar={90,75,65,55,45};                  *<=== Factor A: linear  ***;
YD=J(nt,1,1)||Yar;
YYY=inv(YD‘*YD)*YD‘;
Qr=I(nt)-YD*inv(YD‘*YD)*YD‘;
reg=Qr*(I(nt)@XXX2);

*ZD=nor;                               *<=== Model 1;
*ZD=nor//var;                          *<=== Model 2;
ZD=nor//var//reg;                     *<=== Model 3-4;

i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
        i=i+1; pi=p; p=p0;
        Dpi=inv(diag(pdf('normal',probit(pi))));
        Gpi=ZD*Dpi;
    pio=(I(nt)@CI)*pi;
        Vo=inv(diag(nn)@I(k1))*(diag(pio)
           -(diag(pio))*(I(nt)@(v1*v1‘))*(diag(pio)));
        V=(I(nt)@C)*Vo*(I(nt)@C)‘;
        j=0; diff=1;
        do while (diff > 1e-9);
                j=j+1; pv=p;
```

```
                Dp=inv(diag(pdf('normal',probit(p))));
                Gp=ZD*Dp;
                hp=probit(p);
                g=ZD*hp;
*                print i j p pi g;
                p=p-(Gpi*V)'*ginv(Gp*V*Gpi')*g;
                diff=sqrt((p-pv)'*(p-pv));
                if i=1 & j=1 then do;
                        Wald=g'*ginv(Gp*V*Gp')*g;
                        GpV=Gp*V;
                        df=trace(GpV*ginv(GpV'*GpV)*GpV');
                        discr=wald/n;
                end;
        end;
        diff1=sqrt((p-pi)'*(p-pi));
end;
Cov_pi=V-(Gpi*V)'*ginv(Gpi*V*Gpi')*(Gpi*V);

alpha=(I(nt)@XXX)*hp;
alpha1=(I(nt)@XXX1)*hp;
alpha2=(I(nt)@XXX2)*hp;
Cov_alpha=((I(nt)@XXX)*Dpi)*Cov_pi*((I(nt)@XXX)*Dpi)';

mu=alpha2/alpha1;
sigma=1/alpha1;
beta=(mu@{1,0})+(sigma@{0,1});
B11=-alpha2/(alpha1#alpha1);
B12=1/alpha1;
B21=-1/(alpha1#alpha1);
I11=J(2,2,0);I12=J(2,2,0);I21=J(2,2,0);
I11[1,1]=1;I12[1,2]=1;I21[2,1]=1;
B=(diag(B11)@I11)+(diag(B12)@I12)+(diag(B21)@I21);
Cov_beta=B*Cov_alpha*B';
B1=(diag(B11)@{1 0})+(diag(B12)@{0 1});
```

```
B2=(diag(B21)@{1 0});
Cov_mu=B1*Cov_alpha*B1';
Cov_sigma=B2*Cov_alpha*B2';
SE_mu=sqrt(diag(Cov_mu)*J(nrow(mu),1,1));
SE_sigma=sqrt(diag(Cov_sigma)*J(nrow(sigma),1,1));
print mu SE_mu, sigma SE_sigma;

gamma=YYY*mu;
Cov_gamma=YYY*Cov_mu*YYY';
SE_gamma=sqrt(diag(Cov_mu)*J(nrow(gamma),1,1));
print gamma SE_gamma;

Za=designf(cusum(J(nfac1,1,1)));

LD=J(nt,1,1)||Za;
LLL=inv(LD'*LD)*LD';
lambda=LLL*mu;
lambda=choose(abs(lambda)<1e-9,0,lambda);
Cov_lambda=LLL*Cov_mu*LLL';
Cov_lambda=choose(abs(Cov_lambda)<1e-9,0,Cov_lambda);
SE_lambda=sqrt(diag(Cov_lambda)*J(nrow(lambda),1,1));
print lambda SE_lambda;

TTT=block(1,Za);
tau=TTT*lambda;
Cov_tau=TTT*Cov_lambda*TTT';
SE_tau=sqrt(diag(Cov_tau)*J(nrow(tau),1,1));
print tau SE_tau;

count=cusum(1//nfac1);
tau0=tau[count[1]:count[1]];    SE_tau0=SE_tau[count[1]:count[1]];
tau1=tau[count[1]+1:count[2]];  SE_tau1=SE_tau[count[1]+1:count[2]];
print tau0 SE_tau0, tau1 SE_tau1;
```

```
piom=(shape(pio,nt));
exp1=piom#(repeat(nn,1,k1));
exp2=nn-exp1[,+];
exp=exp1||exp2;

Pearson=(((freq-exp)##2)/exp)[+];
P_pvalue=1-probchi(Pearson,df);
W_pvalue=1-probchi(Wald,df);
print freq exp, Pearson P_pvalue Wald W_pvalue df;
```

B.2 FACTOR2.SAS

```
proc freq data=phdabc.sbib noprint;
tables product / out=product;
tables agegrp / out=agegrp;
tables agec / out=agec;
tables premium / out=class;
tables agegrp*product*premium / out=b;
run;

proc transpose data=b out=freq prefix=c;
by agegrp product;
var count;
run;

proc iml worksize=200 symsize=2000;
use freq; read all var{c1 c2 c3 c4 c5} into freq;
use class; read all var{premium} into class;
use agegrp; read all var{agegrp} into factor1;
use product; read all var{product} into factor2;
print freq factor1 factor2; print class;
```

```
n=freq[+];
nt=nrow(freq);
k=nrow(class); k1=k-1;
x=class[1:k1];
nn=freq[,+];
f=colvec(freq[,1:k1]);  f=f<>0.0001;
C=J(k1,1,1)@cusum(J(1,k1,1))<=J(1,k1,1)@cusum(J(k1,1,1));
CI=inv(C);
v1=J(k1,1,1);
po=inv(diag(nn)@I(k1))*f;
p=(I(nt)@C)*po;

XD=log(x)||v1;
XXX=inv(XD'*XD)*XD'; XXX1=XXX[1,]; XXX2=XXX[2,];
Px=XD*inv(XD'*XD)*XD';
Qx=I(k1)-Px;

nfac1=nrow(factor1);
nfac2=nrow(factor2);
print n nt k x, f po p ;

*Y1=designf(cusum(J(nfac1,1,1))@J(nfac2,1,1));  *<=== Factor A: dummy;
Y1={24.5,34.5,44.5,54.5}@J(nfac2,1,1);          *<=== Factor A: linear;
Y2=designf(J(nfac1,1,1)@cusum(J(nfac2,1,1)));
Y12=hdir(Y1,Y2);
*YD=J(nt,1,1)||Y1||Y2;              *<=== Only main effects;
YD=J(nt,1,1)||Y1||Y2||Y12;          *<=== Main effects with interaction;
Py=YD*inv(YD'*YD)*YD';
Qy=I(nt)-Py;

start GGG(p,g,GG) global(nt,v1,Qx,XXX,XXX1,XXX2,Qy,h,D,kappa,theta,nu,A,Y12);
  h=log(p/((J(nt,1,1)@v1)-p));
  D=inv(diag(p))+inv(diag((J(nt,1,1)@v1)-p));
```

```
  glog=(I(nt)@Qx)*h;
  GGlog=(I(nt)@Qx)*D;

  kappa=(I(nt)@XXX1)*h;
  theta=(I(nt)@XXX2)*h;
  nu=exp(-theta/kappa);
   A1=nu#(theta/(kappa#kappa));
   A2=nu#(-1/kappa);
   A=diag(A1)@{1 0} + diag(A2)@{0 1};
  greg=Qy*nu;
   GGreg=Qy*A*(I(nt)@XXX)*D;

*  g=glog;          *<=== Model 1;
*  GG=GGlog;        *<=== Model 1;
  g=glog//greg;    *<=== Model 2-4;
  GG=GGlog//GGreg; *<=== Model 2-4;
finish;

i=0; p0=p; diff1=1;
do while (diff1 > 1e-9);
  i=i+1; pi=p; p=p0;
    pio=(I(nt)@CI)*pi;
  Vo=inv(diag(nn)@I(k1))*(diag(pio)- (diag(pio))*(I(nt)@(v1*v1‘))*(diag(pio))‘);
  V=(I(nt)@C)*Vo*(I(nt)@C)‘;
  run GGG(pi,gpi,GGpi);
  j=0; diff=1;
  do while (diff > 1e-9);
    j=j+1; pv=p;
    run GGG(p,gp,GGp);
    print i j p pi gp;
    p=p-(GGpi*V)‘*ginv(GGp*V*GGpi‘)*gp;
    diff=sqrt((p-pv)‘*(p-pv));
    if i=1 & j=1 then do;
      Wald=gp‘*ginv(GGp*V*GGp‘)*gp;
```

```
      GpV=GGp*V;
      df=trace(GpV*ginv(GpV'*GpV)*GpV');
      discr=wald/n;
    end;
  end;
  diff1=sqrt((p-pi)'*(p-pi));
end;
nnm=shape(nn,nfac1);
thetam=shape(theta,nfac1);
kappam=shape(kappa,nfac1);

Cov_pi=V-(GGpi*V)'*ginv(GGpi*V*GGpi')*(GGpi*V);
Cov_alpha=((I(nt)@XXX)*D)*Cov_pi*((I(nt)@XXX)*D)';

mu=exp(-theta/kappa)#gamma(J(nt,1,1)+1/kappa)#gamma(J(nt,1,1)-1/kappa);
sigma=sqrt(exp(-2*theta/kappa)
  #(gamma(J(nt,1,1)+2/kappa)#gamma(J(nt,1,1)-2/kappa)
  -(gamma(J(nt,1,1)+1/kappa)#gamma(J(nt,1,1)-1/kappa))##2));
mum=shape(mu,nfac1);
sigmam=shape(sigma,nfac1);
print mum sigmam;

Cov_nu=A*Cov_alpha*A';
SE_nu=sqrt(diag(Cov_nu)*J(nrow(nu),1,1));
num=shape(nu,nfac1); SE_num=shape(SE_nu,nfac1);
print num SE_num;

YYY=inv(YD'*YD)*YD';
gamma=YYY*nu;
Cov_gamma=YYY*Cov_nu*YYY';
SE_gamma=sqrt(diag(Cov_gamma)*J(nrow(gamma),1,1));
print gamma SE_gamma;

D1=designf(cusum(J(nfac2,1,1)));
```

```
DDD=block(1,1,D1,D1);
delta=DDD*gamma;
Cov_delta=DDD*Cov_gamma*DDD';
SE_delta=sqrt(diag(Cov_delta)*J(nrow(delta),1,1));
print delta SE_delta;

Za=designf(cusum(J(nfac1,1,1))@J(nfac2,1,1));  *<=== Factor A: dummy;
Zb=designf(J(nfac1,1,1)@cusum(J(nfac2,1,1)));
Zab=hdir(Za,Zb);
LD=J(nt,1,1)||Za||Zb||Zab;          *<== saturated model;
LLL=inv(LD'*LD)*LD';
lambda=LLL*nu;
lambda=choose(abs(lambda)<1e-9,0,lambda);
Cov_lambda=LLL*Cov_nu*LLL';
Cov_lambda=choose(abs(Cov_lambda)<1e-9,0,Cov_lambda);
print LD lambda;

S1=designf(cusum(J(nfac1,1,1)));
S2=designf(cusum(J(nfac2,1,1)));
S12=S1@S2;
S=block(1,S1,S2,S12);
tau=S*lambda;
Cov_tau=S*Cov_lambda*S';
SE_tau=sqrt(diag(Cov_tau)*J(nrow(tau),1,1));
print tau SE_tau;

count=cusum(1//nfac1//nfac2//(nfac1*nfac2));
tau0=tau[1:1];                  SE_tau0=SE_tau[1:1];
tau1=tau[count[1]+1:count[2]];  SE_tau1=SE_tau[count[1]+1:count[2]];
tau2=tau[count[2]+1:count[3]];  SE_tau2=SE_tau[count[2]+1:count[3]];
tau12=tau[count[3]+1:count[4]]; SE_tau12=SE_tau[count[3]+1:count[4]];
tau12m=shape(tau12,nfac1);      SE_tau12m=shape(SE_tau12,nfac1);
print tau0 tau1 tau2 tau12m, SE_tau0 SE_tau1 SE_tau2 SE_tau12m;
```

```
piom=(shape(pio,nt));
exp1=piom#(repeat(nn,1,k1));
exp2=nn-exp1[,+];
exp=exp1||exp2;

Pearson=(((freq-exp)##2)/exp)[+];
P_pvalue=1-probchi(Pearson,df);
W_pvalue=1-probchi(Wald,df);
print freq exp, Pearson P_pvalue Wald W_pvalue df;

*** Start: Graph ***;
*** Eerste fig: 4.5 en 3.5cm - Tweede fig: 4 en 3cm;
xl=0.5//class[1:k1];
xu=class;
width=xu-xl;
print xl xu x width;
```

Appendix C

SAS Programs: Part III

C.1 Phi0.SAS

```
proc iml;
*===>;a=1; b=2; rho=0.5;

pi=(gamma(0.5))**2;
diff=1; Phi0=0; i=0;
do while (diff>1e-8);
  vorige=Phi0;
  Phi0=Phi0 + ((2*rho)##i*sqrt(1-rho##2))/(4*pi*gamma(i+1))*gamma((i+1)/2)##2
     * probgam((a##2/(2*(1-rho##2))),(i+1)/2)
     * probgam((b##2/(2*(1-rho##2))),(i+1)/2);
  i=i+1;
  diff=abs(Phi0-vorige);
end;
check=probbnrm(a,b,rho)-probbnrm(a,0,rho)-probbnrm(0,b,rho)+probbnrm(0,0,rho);
print Phi0 check;
```

C.2 Phi.SAS

```
proc iml;
*===>; a=1; b=2; rho=0.5;

pi=(gamma(0.5))**2;

start Phi0(Phi0,a,b,rho) global(pi);
diff=1; Phi0=0; i=0;
do while (diff>1e-8);
  vorige=Phi0;
  Phi0=Phi0 + ((2*rho)##i *sqrt(1-rho##2))/(4*pi*gamma(i+1))*gamma((i+1)/2)##2
     * probgam((a##2/(2*(1-rho##2))),(i+1)/2)
     * probgam((b##2/(2*(1-rho##2))),(i+1)/2);
  i=i+1;
  diff=abs(Phi0-vorige);
end;
finish;

if a<0 & b<0 then do;
  run Phi0(Phi01,10,10,rho);
  run Phi0(Phi02,-a,10,rho);
  run Phi0(Phi03,10,-b,rho);
  run Phi0(Phi04,-a,-b,rho);
  Phi=Phi01-Phi02-Phi03+Phi04;
end;
if a<0 & b>=0 then do;
  run Phi0(Phi01,10,10,rho);
  run Phi0(Phi02,-a,10,rho);
  run Phi0(Phi03,10,b,-rho);
  run Phi0(Phi04,-a,b,-rho);
  Phi=Phi01-Phi02+Phi03-Phi04;
end;
```

```
if a>=0 & b<0 then do;
  run Phi0(Phi01,10,10,rho);
  run Phi0(Phi02,a,10,-rho);
  run Phi0(Phi03,10,-b,rho);
  run Phi0(Phi04,a,-b,-rho);
  Phi=Phi01+Phi02-Phi03-Phi04;
end;
if a>=0 & b>=0 then do;
  run Phi0(Phi01,10,10,rho);
  run Phi0(Phi02,a,10,-rho);
  run Phi0(Phi03,10,b,-rho);
  run Phi0(Phi04,a,b,rho);
  Phi=Phi01+Phi02+Phi03+Phi04;
end;
check=probbnrm(a,b,rho);
print Phi check;
```

C.3 BVN.SAS

```
proc iml;
pie=gamma(0.5)##2;

freq={106  90  35  5 ,
      57   73  59  22,
      15   40  57  27,
       2   14  45  99};

x={59.5,69.5,79.5};
y={49.5,59.5,74.5};

n=freq[+];
nfr=freq[,+];
nfc=freq[+,];
nr=nrow(freq); nr1=nr-1; Er=J(nr,1,1); Er1=J(nr1,1,1);
nc=ncol(freq); nc1=nc-1; Ec=J(nc,1,1); Ec1=J(nc1,1,1);
rc=nr*nc;
Cr=J(nr,1,1)@cusum(J(1,nr,1))<=J(1,nr,1)@cusum(J(nr,1,1));
Cc=J(nc,1,1)@cusum(J(1,nc,1))<=J(1,nc,1)@cusum(J(nc,1,1));
C=Cr@Cc; CI=inv(C);
fxy=colvec(freq);
p=C*fxy/n;

XD=x||J(nr1,1,-1);
XXX=inv(XD‘*XD)*XD‘;
PmX=XD*inv(XD‘*XD)*XD‘;

YD=y||J(nc1,1,-1);
YYY=inv(YD‘*YD)*YD‘;
PmY=YD*inv(YD‘*YD)*YD‘;

IV=cusum(j(rc,1,1)); IM=shape(IV,nr);
```

```
xx=IM[1:nr1,nc]; yy=IM[nr,1:nc1]; xy=IM[1:nr1,1:nc1];
Gmx=J(nr1,rc,0); Gmy=J(nc1,rc,0); Gmxy=J(nr1*nc1,rc,0);
ij=0;
do i=1 to nr1; Gmx[i,xx[i]]=1; end;
do j=1 to nc1; Gmy[j,yy[j]]=1; end;
do i=1 to nr1; do j=1 to nc1;
  ij=ij+1;
  Gmxy[ij,xy[i,j]]=1;
end; end;

start F0(F0,z1,z2,rho,k,l) global(pie);
i=1; diff2=1;
F0= 2**((k+l)/2) * (1-rho**2)**((k+l+1)/2) / (4*pie)
  * gamma((k+1)/2) * gamma((l+1)/2)
  * probgam((z1**2/(2*(1-rho**2))),(k+1)/2)
  * probgam((z2**2/(2*(1-rho**2))),(l+1)/2);
do while (diff2>1e-9);
  vF0=F0;
  F0= F0+2**((k+l)/2)*(1-rho**2)**((k+l+1)/2) / (4*pie)*(2*rho)**i
    * gamma((i+k+1)/2) * gamma((i+l+1)/2) / gamma(i+1)
    * probgam((z1**2/(2*(1-rho**2))),(i+k+1)/2)
    * probgam((z2**2/(2*(1-rho**2))),(i+l+1)/2);
  diff2=abs(vF0-F0);
  i=i+1;
end;
finish;

start F (F,rho,zy,zx,k,l);
  if zx<0 & zy<0 then do;
    run F0(F1,10,10,rho,k,l);
    run F0(F2,-zx,10,rho,k,l);
    run F0(F3,10,-zy,rho,k,l);
    run F0(F4,-zx,-zy,rho,k,l);
    F=F1-F2-F3+F4;
```

```
  end;
  if zx<0 & zy>=0 then do;
    run F0(F1,10,10,rho,k,l);
    run F0(F2,-zx,10,rho,k,l);
    run F0(F3,10,zy,-rho,k,l);  F3=F3*(-1)**k;
    run F0(F4,-zx,zy,-rho,k,l);  F4=F4*(-1)**k;
    F=F1-F2+F3-F4;
  end;
  if zx>=0 & zy<0 then do;
    run F0(F1,10,10,rho,k,l);
    run F0(F2,zx,10,-rho,k,l);  F2=F2*(-1)**l;
    run F0(F3,10,-zy,rho,k,l);
    run F0(F4,zx,-zy,-rho,k,l);  F4=F4*(-1)**l;
    F=F1+F2-F3-F4;
  end;
  if zx>=0 & zy>=0 then do;
    run F0(F1,10,10,rho,k,l);
    run F0(F2,zx,10,-rho,k,l);  F2=F2*(-1)**l;
    run F0(F3,10,zy,-rho,k,l);  F3=F3*(-1)**k;
    run F0(F4,zx,zy,rho,k,l);
    F=F1+F2+F3+F4;
  end;
finish;

start prob (pp,x1,x2,y1,y2,rho);
  pp=probbnrm(x2,y2,rho)-probbnrm(x2,y1,rho)
    -probbnrm(x1,y2,rho)+probbnrm(x1,y1,rho);
finish;

start volume(p,rho,vv,zx,zy,II,JJ) global(nr,nc,nr1,nc1,Er,Ec,CI,pie);
  zx1=-10//zx; zx2=zx//10;
  zy1=-10//zy; zy2=zy//10;

  run prob(ppIJ,zx[II-1],zx[II],zy[JJ-1],zy[JJ],rho);
```

```
run prob(ppIJ1,zx[II-1],0,zy[JJ-1],0,rho);
run prob(ppIJ2,zx[II-1],0,0,zy[JJ],rho);
run prob(ppIJ3,0,zx[II],zy[JJ-1],0,rho);
run prob(ppIJ4,0,zx[II],0,zy[JJ],rho);

run prob(ppI,((zx[II-1])*Ec),((zx[II])*Ec),zy1,zy2,rho);
run prob(ppI1,((zx[II-1])*Ec),(0*Ec),zy1,zy2,rho);
run prob(ppI2,(0*Ec),((zx[II])*Ec),zy1,zy2,rho);
run prob(ppJ,zx1,zx2,((zy[JJ-1])*Er),((zy[JJ])*Er),rho);
run prob(ppJ1,zx1,zx2,((zy[JJ-1])*Er),(0*Er),rho);
run prob(ppJ2,zx1,zx2,(0*Er),((zy[JJ])*Er),rho);

volc1=J(nr,nc,0);volc2=J(nr,nc,0);volc3=J(nr,nc,0);volc4=J(nr,nc,0);

volc1[1:II-1,1:JJ-1]=1;
volc2[1:II-1,JJ+1:nc]=1;
volc3[II+1:nr,1:JJ-1]=1;
volc4[II+1:nr,JJ+1:nc]=1;

volc1[II,1:JJ-1]=(ppI1[1:JJ-1]/ppI[1:JJ-1])‘;
volc2[II,JJ+1:nc]=(ppI1[JJ+1:nc]/ppI[JJ+1:nc])‘;
volc3[II,1:JJ-1]=(ppI2[1:JJ-1]/ppI[1:JJ-1])‘;
volc4[II,JJ+1:nc]=(ppI2[JJ+1:nc]/ppI[JJ+1:nc])‘;

volc1[1:II-1,JJ]=(ppJ1[1:II-1]/ppJ[1:II-1]);
volc2[1:II-1,JJ]=(ppJ2[1:II-1]/ppJ[1:II-1]);
volc3[II+1:nr,JJ]=(ppJ1[II+1:nr]/ppJ[II+1:nr]);
volc4[II+1:nr,JJ]=(ppJ2[II+1:nr]/ppJ[II+1:nr]);

volc1[II,JJ]=ppIJ1/ppIJ;
volc2[II,JJ]=ppIJ2/ppIJ;
volc3[II,JJ]=ppIJ3/ppIJ;
volc4[II,JJ]=ppIJ4/ppIJ;
```

```
  v1=colvec(volc1);
  v2=colvec(volc2);
  v3=colvec(volc3);
  v4=colvec(volc4);
  vv=(v1+v4)-(v2+v3);

  vol1=v1`*CI*p;
  vol2=v2`*CI*p;
  vol3=v3`*CI*p;
  vol4=v4`*CI*p;
finish;

start rho (p,rhop,vvp,zxp,zyp,IIp,JJp,drdp) global(pie,CI);
  rhop=0; diffr=1;
  do while (diffr > 1e-10);
    rhov=rhop;
    run volume(p,rhop,vvp,zxp,zyp,IIp,JJp);
    rhop=sin(pie/2*(vvp`*CI*p));
    diffr=sqrt((rhop-rhov)**2);
    drdp=cos(pie/2*(vvp`*CI*p))*pie/2*vvp`*CI;
  end;
finish;

start GGxy(p,rho,zx,zy,Dx,Dy,drdp,GGxy) global(nr1,nc1,rc,Pmx,Pmy,Gmx,Gmy,Gmxy);
  ZZx=zx@J(1,nc1,1);
  ZZy=zy`@J(nr1,1,1);
  dFdzxm=diag(pdf('normal',zx))*probnorm((ZZy-rho*ZZx)/sqrt(1-rho**2));
  dFdzym=probnorm((ZZx-rho*ZZy)/sqrt(1-rho**2))*diag(pdf('normal',zy));
  do i=1 to nr1;
    EEr=J(nr1,nr1,0);
    EEr[i,i]=1;
    tyd=colvec(EEr*dFdzxm);
    if i=1 then dFdzx=tyd;
    else dFdzx=dFdzx||colvec(tyd);
```

```
  end;
  do j=1 to nc1;
    EEc=J(nc1,nc1,0);
    EEc[j,j]=1;
    tyd=colvec(dFdzym*EEc);
    if j=1 then dFdzy=colvec(tyd);
    else dFdzy=dFdzy||colvec(tyd);
  end;
  dFdr=J(nr1,nc1,0); rho2=rho**2;
  do i=1 to nr1; do j=1 to nc1;
    run F(F00,rho,zx[i],zy[j],0,0);
    run F(F20,rho,zx[i],zy[j],2,0);
    run F(F11,rho,zx[i],zy[j],1,1);
    run F(F02,rho,zx[i],zy[j],0,2);
    dFdr[i,j]=rho/(1-rho2)*F00 - rho/((1-rho2)**2)*F20
        + (1+rho2)/((1-rho2)**2)*F11 - rho/((1-rho2)**2)*F02;
  end; end;
  dFdr=colvec(dFdr);
  dzxdp=Pmx*Dx*Gmx;
  dzydp=Pmy*Dy*Gmy;
  GGxy=(dFdzx||dFdzy||dFdr)*(dzxdp//dzydp//drdp) - Gmxy;
finish;

start marginal(px,alphaxp,muxp,sigmaxp,zxp,nr,IIp,Dxp,Gmx,XD,XXX,Pmx,GGxp);
  alphaxp=XXX*probit(px);
  zxp=XD*alphaxp;
  muxp=alphaxp[2]/alphaxp[1];
  sigmaxp=1/alphaxp[1];
  do IIp=1 to nr until (zxp[IIp]>=0); end;
  Dxp=inv(diag(pdf('normal',probit(px))));
  GGxp=diag(pdf('normal',zxp))*Pmx*Dxp*Gmx - Gmx;
finish;

i=0; p0=p; diff1=1;
```

```
do while (diff1 > 1e-8);
  i=i+1; pi=p; p=p0;
  V=(C*diag(CI*pi)*C'-pi*pi')/n;
  matrixpi=shape(pi,nr);
  pix=matrixpi[1:nr1,nc];
  piy=matrixpi[nr,1:nc1]';
  pixy=colvec(matrixpi[1:nr1,1:nc1]);
  run marginal(pix,alphaxpi,muxpi,sigmaxpi,zxpi,nr,IIpi,Dxpi,Gmx,XD,XXX,Pmx,GGxpi);
  run marginal(piy,alphaypi,muypi,sigmaypi,zypi,nc,JJpi,Dypi,Gmy,YD,YYY,Pmy,GGypi);
  run rho(pi,rhopi,vvpi,zxpi,zypi,IIpi,JJpi,drdpi);
  run GGxy(pi,rhopi,zxpi,zypi,Dxpi,Dypi,drdpi,GGxypi);
  GGpi=GGxpi//GGypi//GGxypi;
  j=0; diff=1;
  do while (diff > 1e-8);
    j=j+1; pv=p;
    matrixp=shape(p,nr);
    px=matrixp[1:nr1,nc];
    py=matrixp[nr,1:nc1]';
    pxy=colvec(matrixp[1:nr1,1:nc1]);
    run marginal(px,alphaxp,muxp,sigmaxp,zxp,nr,IIp,Dxp,Gmx,XD,XXX,Pmx,GGxp);
    run marginal(py,alphayp,muyp,sigmayp,zyp,nc,JJp,Dyp,Gmy,YD,YYY,Pmy,GGyp);
    run rho(p,rhop,vvp,zxp,zyp,IIp,JJp,drdp);
    run GGxy(p,rhop,zxp,zyp,Dxp,Dyp,drdp,GGxyp);
    GGp=GGxp//GGyp//GGxyp;
    gx=probnorm(zxp)-px;
    gy=probnorm(zyp)-py;
    gxy=probbnrm(zxp@Ec1,Er1@zyp,rhop)-pxy;
    g=gx//gy//gxy;
    print i j g pi p, matrixp zxp zyp,
              rhopi muxpi sigmaxpi muypi sigmaypi,
              rhop muxp sigmaxp muyp sigmayp;
    p=p-(GGpi*V)'*ginv(GGp*V*GGpi')*g;
    if i=1 & j=1 then do;
      Wald=g'*ginv(GGp*V*GGp')*g;
```

```
      GGpV=GGp*V;
      df=trace(GGpV*ginv(GGpV'*GGpV)*GGpV');
      pvalue=1-probchi(Wald,df);
      discr=wald/n;
      Cov_rho=drdpi*V*drdpi';
      SE_rho=sqrt(Cov_rho);
    end;
    diff=sqrt((p-pv)'*(p-pv));
  end;
  diff1=sqrt((p-pi)'*(p-pi));
end;
mux=muxp; sigmax=sigmaxp;
muy=muyp; sigmay=sigmayp;
rho=rhop;

Cov_pi=V-(GGpi*V)'*ginv(GGpi*V*GGpi')*(GGpi*V);

alphax=alphaxp;
Cov_alphax=(XXX*Dxpi*Gmx)*Cov_pi*(XXX*Dxpi*Gmx)';
Ax=J(2,2,0);
Ax[1,1]=-alphax[2]/((alphax[1])**2);
Ax[1,2]=1/(alphax[1]);
Ax[2,1]=-1/((alphax[1])**2);
Cov_musigx=Ax*Cov_alphax*Ax';
SE_mux=sqrt(Cov_musigx[1,1]);
SE_sigmax=sqrt(Cov_musigx[2,2]);

alphay=alphayp;
Cov_alphay=(YYY*Dypi*Gmy)*Cov_pi*(YYY*Dypi*Gmy)';
Ay=J(2,2,0);
Ay[1,1]=-alphay[2]/((alphay[1])**2);
Ay[1,2]=1/(alphay[1]);
Ay[2,1]=-1/((alphay[1])**2);
Cov_musigy=Ay*Cov_alphay*Ay';
```

```
SE_muy=sqrt(Cov_musigy[1,1]);
SE_sigmay=sqrt(Cov_musigy[2,2]);

print mux SE_mux sigmax SE_sigmax,
      muy SE_muy sigmay SE_sigmay,
      rho SE_rho;

t_rho=rho/SE_rho;
p_rho=(1-probnorm(t_rho))*2;
print t_rho p_rho;

alpha_xy=mux-muy*rho*sigmax/sigmay;
beta_xy=rho*sigmax/sigmay;
alpha_yx=muy-mux*rho*sigmay/sigmax;
beta_yx=rho*sigmay/sigmax;
print alpha_xy beta_xy alpha_yx beta_yx;

exp=shape((CI*pi*n),nr);
pearson=(((freq-exp)##2)/exp)[+];
print n freq exp Pearson Wald;
```

C.4 BVNSIM.SAS

```
proc iml worksize=30000 symsize=30000;
pie=gamma(0.5)##2;

n=1000;
rep=10;
number=1;
x={8,10,12}; y={45,50,55};
mu={11,48};
sig={9  -16.8,
     -16.8  64};
call eigen(L,H,SIG);
sig12=H*diag(sqrt(L))*H‘;

nr=nrow(x)+1; nr1=nr-1; Er=J(nr,1,1); Er1=J(nr1,1,1);
nc=nrow(y)+1; nc1=nc-1; Ec=J(nc,1,1); Ec1=J(nc1,1,1);
rc=nr*nc;
Cr=J(nr,1,1)@cusum(J(1,nr,1))<=J(1,nr,1)@cusum(J(nr,1,1));
Cc=J(nc,1,1)@cusum(J(1,nc,1))<=J(1,nc,1)@cusum(J(nc,1,1));
C=Cr@Cc; CI=inv(C);

XD=x||J(nr1,1,-1);
XXX=inv(XD‘*XD)*XD‘;
PmX=XD*inv(XD‘*XD)*XD‘;

YD=y||J(nc1,1,-1);
YYY=inv(YD‘*YD)*YD‘;
PmY=YD*inv(YD‘*YD)*YD‘;

IV=cusum(j(rc,1,1)); IM=shape(IV,nr);
xx=IM[1:nr1,nc]; yy=IM[nr,1:nc1]; xy=IM[1:nr1,1:nc1];
Gmx=J(nr1,rc,0); Gmy=J(nc1,rc,0); Gmxy=J(nr1*nc1,rc,0);
ij=0;
```

```
do i=1 to nr1; Gmx[i,xx[i]]=1; end;
do j=1 to nc1; Gmy[j,yy[j]]=1; end;
do i=1 to nr1; do j=1 to nc1;
    ij=ij+1;
    Gmxy[ij,xy[i,j]]=1;
end; end;

/*
*** Begin: Theoretical values ***;
poprho=sig[1,2]/sqrt(sig[1,1]*sig[2,2]);
popzx=((x-mu[1,1])/sqrt(sig[1,1]))//10;
popzy=((y-mu[2,1])/sqrt(sig[2,2]))//10;
poppi=probbnrm((popzx)@J(nc,1,1),J(nr,1,1)@(popzy),poprho);
freq=shape(CI*poppi*n,nr);
fxy=colvec(freq);
p=C*fxy/freq[+];
*** End: Theoretical values ***;
*/

start data;
sp=sig12*rannor(J(2,n,number))+ mu*J(1,n,1);

smu=sp[,+]/n;
ssig=sp*sp`/n-smu*smu`;
D=inv(sqrt(diag(ssig)));
scor=D*ssig*D;
smux=smu[1]; smuy=smu[2];
ssigx=sqrt(ssig[1,1]); ssigy=sqrt(ssig[2,2]);
srho=ssig[1,2]/sqrt(ssig[1,1]*ssig[2,2]);

sp=sp`;
spx=sp[,1];
spy=sp[,2];
f=j(nr,nc,0);
```

```
do k=1 to n;
    t=j(nr,nc,0);
    do III=1 to nr1 until (spx[k] <= x[III]); end;
    do JJJ=1 to nc1 until (spy[k] <= y[JJJ]); end;
    t[III,JJJ]=1;
    f=f+t;
end;
freq=f;
freq=freq<>1e-6;
fxy=colvec(freq);
p=C*fxy/freq[+];
finish;

start F0(F0,z1,z2,rho,k,l) global(pie);
i=1; diff2=1;
F0= 2**((k+l)/2) * (1-rho**2)**((k+l+1)/2) / (4*pie)
    * gamma((k+1)/2) * gamma((l+1)/2)
    * probgam((z1**2/(2*(1-rho**2))),(k+1)/2)
    * probgam((z2**2/(2*(1-rho**2))),(l+1)/2);
do while (diff2>1e-9);
    vF0=F0;
    F0= F0+2**((k+l)/2) *(1-rho**2)**((k+l+1)/2) / (4*pie) * (2*rho)**i
        * gamma((i+k+1)/2) * gamma((i+l+1)/2) / gamma(i+1)
        * probgam((z1**2/(2*(1-rho**2))),(i+k+1)/2)
        * probgam((z2**2/(2*(1-rho**2))),(i+l+1)/2);
    diff2=abs(vF0-F0);
    i=i+1;
end;
finish;

start F (F,rho,zy,zx,k,l);
    if zx<0 & zy<0 then do;
        run F0(F1,10,10,rho,k,l);
        run F0(F2,-zx,10,rho,k,l);
```

```
        run F0(F3,10,-zy,rho,k,l);
        run F0(F4,-zx,-zy,rho,k,l);
        F=F1-F2-F3+F4;
    end;
    if zx<0 & zy>=0 then do;
        run F0(F1,10,10,rho,k,l);
        run F0(F2,-zx,10,rho,k,l);
        run F0(F3,10,zy,-rho,k,l);    F3=F3*(-1)**k;
        run F0(F4,-zx,zy,-rho,k,l);    F4=F4*(-1)**k;
        F=F1-F2+F3-F4;
    end;
    if zx>=0 & zy<0 then do;
        run F0(F1,10,10,rho,k,l);
        run F0(F2,zx,10,-rho,k,l);    F2=F2*(-1)**l;
        run F0(F3,10,-zy,rho,k,l);
        run F0(F4,zx,-zy,-rho,k,l);    F4=F4*(-1)**l;
        F=F1+F2-F3-F4;
    end;
    if zx>=0 & zy>=0 then do;
        run F0(F1,10,10,rho,k,l);
        run F0(F2,zx,10,-rho,k,l);    F2=F2*(-1)**l;
        run F0(F3,10,zy,-rho,k,l);    F3=F3*(-1)**k;
        run F0(F4,zx,zy,rho,k,l);
        F=F1+F2+F3+F4;
    end;
finish;

start prob (pp,x1,x2,y1,y2,rho);
    pp=probbnrm(x2,y2,rho)-probbnrm(x2,y1,rho)
      -probbnrm(x1,y2,rho)+probbnrm(x1,y1,rho);
finish;

start volume(p,rho,vv,zx,zy,II,JJ) global(nr,nc,nr1,nc1,Er,Ec,CI,pie);
    zx1=-10//zx; zx2=zx//10;
```

```
zy1=-10//zy; zy2=zy//10;

run prob(ppIJ,zx[II-1],zx[II],zy[JJ-1],zy[JJ],rho);
run prob(ppIJ1,zx[II-1],0,zy[JJ-1],0,rho);
run prob(ppIJ2,zx[II-1],0,0,zy[JJ],rho);
run prob(ppIJ3,0,zx[II],zy[JJ-1],0,rho);
run prob(ppIJ4,0,zx[II],0,zy[JJ],rho);

run prob(ppI,((zx[II-1])*Ec),((zx[II])*Ec),zy1,zy2,rho);
run prob(ppI1,((zx[II-1])*Ec),(0*Ec),zy1,zy2,rho);
run prob(ppI2,(0*Ec),((zx[II])*Ec),zy1,zy2,rho);
run prob(ppJ,zx1,zx2,((zy[JJ-1])*Er),((zy[JJ])*Er),rho);
run prob(ppJ1,zx1,zx2,((zy[JJ-1])*Er),(0*Er),rho);
run prob(ppJ2,zx1,zx2,(0*Er),((zy[JJ])*Er),rho);

volc1=J(nr,nc,0);volc2=J(nr,nc,0);volc3=J(nr,nc,0);volc4=J(nr,nc,0);

volc1[1:II-1,1:JJ-1]=1;
volc2[1:II-1,JJ+1:nc]=1;
volc3[II+1:nr,1:JJ-1]=1;
volc4[II+1:nr,JJ+1:nc]=1;

volc1[II,1:JJ-1]=(ppI1[1:JJ-1]/ppI[1:JJ-1])‘;
volc2[II,JJ+1:nc]=(ppI1[JJ+1:nc]/ppI[JJ+1:nc])‘;
volc3[II,1:JJ-1]=(ppI2[1:JJ-1]/ppI[1:JJ-1])‘;
volc4[II,JJ+1:nc]=(ppI2[JJ+1:nc]/ppI[JJ+1:nc])‘;

volc1[1:II-1,JJ]=(ppJ1[1:II-1]/ppJ[1:II-1]);
volc2[1:II-1,JJ]=(ppJ2[1:II-1]/ppJ[1:II-1]);
volc3[II+1:nr,JJ]=(ppJ1[II+1:nr]/ppJ[II+1:nr]);
volc4[II+1:nr,JJ]=(ppJ2[II+1:nr]/ppJ[II+1:nr]);

volc1[II,JJ]=ppIJ1/ppIJ;
volc2[II,JJ]=ppIJ2/ppIJ;
```

```
    volc3[II,JJ]=ppIJ3/ppIJ;
    volc4[II,JJ]=ppIJ4/ppIJ;

    v1=colvec(volc1);
    v2=colvec(volc2);
    v3=colvec(volc3);
    v4=colvec(volc4);
    vv=(v1+v4)-(v2+v3);

    vol1=v1`*CI*p;
    vol2=v2`*CI*p;
    vol3=v3`*CI*p;
    vol4=v4`*CI*p;
finish;

start rho (p,rhop,vvp,zxp,zyp,IIp,JJp,drdp) global(pie,CI);
    i=0;
    rhop=0; diffr=1;
    do while ((diffr > 1e-10) & (i<100));
    i=i+1;
        rhov=rhop;
        run volume(p,rhop,vvp,zxp,zyp,IIp,JJp);
        rhop=sin(pie/2*(vvp`*CI*p));
        diffr=sqrt((rhop-rhov)**2);
        drdp=cos(pie/2*(vvp`*CI*p))*pie/2*vvp`*CI;
    end;
finish;

start GGxy(p,rho,zx,zy,Dx,Dy,drdp,GGxy) global(nr1,nc1,rc,Pmx,Pmy,Gmx,Gmy,Gmxy);
    ZZx=zx@J(1,nc1,1);
    ZZy=zy`@J(nr1,1,1);
    dFdzxm=diag(pdf('normal',zx))*probnorm((ZZy-rho*ZZx)/sqrt(1-rho**2));
    dFdzym=probnorm((ZZx-rho*ZZy)/sqrt(1-rho**2))*diag(pdf('normal',zy));
    do i=1 to nr1;
```

```
        EEr=J(nr1,nr1,0);
        EEr[i,i]=1;
        tyd=colvec(EEr*dFdzxm);
        if i=1 then dFdzx=tyd;
        else dFdzx=dFdzx||colvec(tyd);
    end;
    do j=1 to nc1;
        EEc=J(nc1,nc1,0);
        EEc[j,j]=1;
        tyd=colvec(dFdzym*EEc);
        if j=1 then dFdzy=colvec(tyd);
        else dFdzy=dFdzy||colvec(tyd);
    end;
    dFdr=J(nr1,nc1,0); rho2=rho**2;
    do i=1 to nr1; do j=1 to nc1;
        run F(F00,rho,zx[i],zy[j],0,0);
        run F(F20,rho,zx[i],zy[j],2,0);
        run F(F11,rho,zx[i],zy[j],1,1);
        run F(F02,rho,zx[i],zy[j],0,2);
        dFdr[i,j]=rho/(1-rho2)*F00 - rho/((1-rho2)**2)*F20
                + (1+rho2)/((1-rho2)**2)*F11 - rho/((1-rho2)**2)*F02;
    end; end;
    dFdr=colvec(dFdr);
    dzxdp=Pmx*Dx*Gmx;
    dzydp=Pmy*Dy*Gmy;
    GGxy=(dFdzx||dFdzy||dFdr)*(dzxdp//dzydp//drdp) - Gmxy;
finish;

start marginal(px,alphaxp,muxp,sigmaxp,zxp,nr,IIp,Dxp,Gmx,XD,XXX,Pmx,GGxp);
    alphaxp=XXX*probit(px);
    zxp=XD*alphaxp;
    muxp=alphaxp[2]/alphaxp[1];
    sigmaxp=1/alphaxp[1];
    do IIp=1 to nr until (zxp[IIp]>=0); end;
```

```
    Dxp=inv(diag(pdf('normal',probit(px))));
    GGxp=diag(pdf('normal',zxp))*Pmx*Dxp*Gmx - Gmx;
finish;

start fit;
i=0; p0=p; diff1=1;
do while (diff1 > 1e-8);
    i=i+1; pi=p; p=p0;
    V=(C*diag(CI*pi)*C`-pi*pi`)/n;
    matrixpi=shape(pi,nr);
    pix=matrixpi[1:nr1,nc];
    piy=matrixpi[nr,1:nc1]`;
    pixy=colvec(matrixpi[1:nr1,1:nc1]);
    run marginal(pix,alphaxpi,muxpi,sigmaxpi,zxpi,nr,IIpi,Dxpi,Gmx,XD,XXX,Pmx,GGxpi);
    run marginal(piy,alphaypi,muypi,sigmaypi,zypi,nc,JJpi,Dypi,Gmy,YD,YYY,Pmy,GGypi);
    run rho(pi,rhopi,vvpi,zxpi,zypi,IIpi,JJpi,drdpi);
    run GGxy(pi,rhopi,zxpi,zypi,Dxpi,Dypi,drdpi,GGxypi);
    GGpi=GGxpi//GGypi//GGxypi;
    j=0; diff=1;
    do while (diff > 1e-8);
        j=j+1; pv=p;
        matrixp=shape(p,nr);
        px=matrixp[1:nr1,nc];
        py=matrixp[nr,1:nc1]`;
        pxy=colvec(matrixp[1:nr1,1:nc1]);
        run marginal(px,alphaxp,muxp,sigmaxp,zxp,nr,IIp,Dxp,Gmx,XD,XXX,Pmx,GGxp);
        run marginal(py,alphayp,muyp,sigmayp,zyp,nc,JJp,Dyp,Gmy,YD,YYY,Pmy,GGyp);
        run rho(p,rhop,vvp,zxp,zyp,IIp,JJp,drdp);
        run GGxy(p,rhop,zxp,zyp,Dxp,Dyp,drdp,GGxyp);
        GGp=GGxp//GGyp//GGxyp;
        gx=probnorm(zxp)-px;
        gy=probnorm(zyp)-py;
        gxy=probbnrm(zxp@Ec1,Er1@zyp,rhop)-pxy;
        g=gx//gy//gxy;
```

```
        *print r i j g pi p,
              rhopi muxpi sigmaxpi muypi sigmaypi,
              rhop muxp sigmaxp muyp sigmayp;
        p=p-(GGpi*V)'*ginv(GGp*V*GGpi')*g;
        if i=1 & j=1 then do;
            Wald=g'*ginv(GGp*V*GGp')*g;
            GGpV=GGp*V;
            df=trace(GGpV*ginv(GGpV'*GGpV)*GGpV');
            pvalue=1-probchi(Wald,df);
            discr=wald/n;
            Cov_rho=drdpi*V*drdpi';
            SE_rho=sqrt(Cov_rho);
        end;
        diff=sqrt((p-pv)'*(p-pv));
    end;
    diff1=sqrt((p-pi)'*(p-pi));
end;
mux=muxp; sigmax=sigmaxp;
muy=muyp; sigmay=sigmayp;
rho=rhop;

Cov_pi=V-(GGpi*V)'*ginv(GGpi*V*GGpi')*(GGpi*V);

alpha_xy=mux-muy*rho*sigmax/sigmay;
beta_xy=rho*sigmax/sigmay;
alpha_yx=muy-mux*rho*sigmay/sigmax;
beta_yx=rho*sigmay/sigmax;

alphax=alphaxp;
Cov_alphax=(XXX*Dxpi*Gmx)*Cov_pi*(XXX*Dxpi*Gmx)';
Ax=J(2,2,0);
Ax[1,1]=-alphax[2]/((alphax[1])**2);
Ax[1,2]=1/(alphax[1]);
Ax[2,1]=-1/((alphax[1])**2);
```

```
Cov_musigx=Ax*Cov_alphax*Ax';
SE_mux=sqrt(Cov_musigx[1,1]);
SE_sigmax=sqrt(Cov_musigx[2,2]);

alphay=alphayp;
Cov_alphay=(YYY*Dypi*Gmy)*Cov_pi*(YYY*Dypi*Gmy)';
Ay=J(2,2,0);
Ay[1,1]=-alphay[2]/((alphay[1])**2);
Ay[1,2]=1/(alphay[1]);
Ay[2,1]=-1/((alphay[1])**2);
Cov_musigy=Ay*Cov_alphay*Ay';
SE_muy=sqrt(Cov_musigy[1,1]);
SE_sigmay=sqrt(Cov_musigy[2,2]);
exp=shape((CI*pi*n),nr);
Pearson=(((freq-exp)##2)/exp)[+];

print r freq exp matrixp,
      mux SE_mux sigmax SE_sigmax,
      muy SE_muy sigmay SE_sigmay,
      rho SE_rho;
finish;

start write;
stats[r,1]=number;
stats[r,2]=i;
stats[r,3]=j;
stats[r,4]=smux;
stats[r,5]=ssigx;
stats[r,6]=smuy;
stats[r,7]=ssigy;
stats[r,8]=srho;
stats[r,9]=mux;
stats[r,10]=SE_mux;
stats[r,11]=sigmax;
```

```
stats[r,12]=SE_sigmax;
stats[r,13]=muy;
stats[r,14]=SE_muy;
stats[r,15]=sigmay;
stats[r,16]=SE_sigmay;
stats[r,17]=rho;
stats[r,18]=SE_rho;
stats[r,19]=Pearson;
stats[r,20]=Wald;
finish;

stats=J(rep,20,0);
do r=1 to rep;
     run data;
    run fit;
     run write;
    number=number+1;
end;

create a from stats [colname={'number' 'i' 'j' 'smux' 'ssigx' 'smuy' 'ssigy'
'srho' 'mux' 'SE_mux' 'sigmax' 'SE_sigmax' 'muy' 'SE_muy' 'sigmay' 'SE_sigmay'
'rho' 'SE_rho' 'Pearson' 'Wald'}];
append from stats;

quit;

proc means data=a mean std p5 p50 p95;
run;
```