Appendix 2

A 2.1 Program to find Power of the test for the parameter π in Zero Inflated Poisson distribution

```
# include <stdio.h>
# include <math.h>
# include <conio.h>
# include <time.h>
#include <stdlib.h>
void main()
{       int i, x[100], n, k, noofzero, sim, nosim = 25000;
        float mu, u, prod, p, xbar, sumx, fmu, fdmu, newmu, oldmu, phat,
        ybar;
        float power, muh0, avarph0, z;
        clrscr(); randomize();
        n=10; mu=5.0;
        for (p=0.5; p<=1.00;p = p + 0.1)
  {      power=0.0;
        for(sim=1; sim<=nosim; sim++)
         {  for (i=1;i<=n; i++)
         { u=(float)random(RAND_MAX)/RAND_MAX;
           if (u<=p)
         { prod=1.0;
             k=0;
            do{ k=k+1;
            u=(float)random(RAND_MAX)/RAND_MAX;
            prod=prod*u;
  } while (prod>exp(-mu));
        x[i]=k-1;
 }
        else{x[i]=0; }
 }
         noofzero=0;sumx=0.0;
         for(i=1;i<=n;i++)
         {if(x[i]==0) {noofzero=noofzero+1;}
         else {sumx=sumx+x[i];}
  }
        muh0=sumx/(float)(n-noofzero);
        newmu=muh0;
        do{oldmu=newmu;
        fmu=sumx-((n-noofzero)*oldmu/(1-exp(-oldmu)));
```

```
          fdmu=(n-noofzero)*(oldmu*exp(-oldmu) + exp(-oldmu)-1.0)/((1-
         exp(-oldmu))*(1-exp(-oldmu)));
         newmu=oldmu-(fmu/fdmu);
 }       while(fabs(newmu-oldmu)>0.000001);
          avarph0=exp(-newmu)/(n*(1-exp(-newmu)));
          phat=(n-noofzero)/(n*(1-exp(-newmu)));
          z=(phat-1)/sqrt(avarph0);
          if(fabs(z)>=2.326) power=power+1;
 }
          printf("\np=%f   Power=%f ",p,100*power/nosim);
 }
         getch();
 }
```

A 2.2 Program to find Power of the test for the parameter π in Zero Inflated Negative Binomial Distribution

```
// program to find power of the ZINBD
 # include <stdio.h>
 # include <math.h>
 # include <conio.h>
 # include <time.h>
 #include <stdlib.h>
 void main()
 {      int w,i,j,x[100],m=10,k,r,noofzero,sim,nosim=25000;
        float u,p,pi,xbar,sum,sumx,hp,hdp,newp,oldp,pihat,ybar;
        float ph0,power,avarpi0,z,pr1,pr2;
         clrscr(); randomize();
         r=10;p=0.5;
         for(pi=0.5;pi<=1.01;pi=pi+.1)
 {        power=0.0;
         for(sim=1;sim<=nosim;sim++)
 {      for(i=1;i<=m;i++)
 {      u=(float)random(RAND_MAX)/RAND_MAX;
         if (u<=pi)
 {      sum=0.0;
        for(j=1;j<=r;j++)
     {  u=(float)random(RAND_MAX)/RAND_MAX;
         printf("\n log(u)=%f  ",log(u));
        w=log(u)/log(1-p);
        sum=sum+w;
      }
        x[i]=sum;
    }
         else{x[i]=0; }
 }
        noofzero=0;sumx=0.0;
        for(i=1;i<=m;i++)
 {      if(x[i]==0) {noofzero=noofzero+1;}
        else {sumx=sumx+x[i];}
      }
        ph0=sumx/(float)(m-noofzero);
         ph0=1.0/((ph0/r)+1);
         newp=ph0;
         printf("\n newp=%f",newp);
 do {  oldp=newp;
        pr1=pow(oldp+1,-r);
         pr2=pow(oldp+1,-r-1);
        printf("\n pr1=%f pr2=%f", pr1,pr2);
        getch();
         hp=sumx-((m-noofzero)*r*oldp/(1-pr1));
         hdp=(m-noofzero)*r*((r*oldp*pr2)+(pr1-1));
         hdp=hdp/pow((1-pr1),2);
         newp=oldp-(hp/hdp);
}       while(fabs(newp-oldp)>0.000001);
         avarpi0=pr1/(m*(1-pr1));
        pihat=(m-noofzero)/(m*(1-pr1));
        z=(pihat-1)/sqrt(avarpi0);
         if(fabs(z)>=2.326) power=power+1;
}        printf("\np=%f   Power=%f ",pi,100*power/nosim);
        getch();
}
         getch();
        getch();
}
```

A 2.3 Program to find Power of the test for the parameter π in Zero Inflated Binomial Distribution

```
// Program to find the power of the ZIBIPI Distribution.
 # include <stdio.h>
 # include <math.h>
 # include <conio.h>
 # include <time.h>
 #include <stdlib.h>
 void main()
 {
        int i,j,m=10,x[100],n,k,noofzero,sim,nosim=25000,sum;
         float th,u,prod,p,xbar,sumx,hth,hdth,newth,oldth,phat,pi;
         float power,thh0,avarpi0,z,ath,aphat,nth;
         clrscr();       randomize();
         n=10; th=0.50;
        ath=0.0;        aphat=0.0;
         for(pi=0.5;pi<=1.01;pi=pi+.10)
 {      power=0.0;
         for(sim=1;sim<=nosim;sim++)
 {
        for(i=1;i<=m;i++)
 {
         u=(float)random(RAND_MAX)/RAND_MAX;
         if (u<=pi)
 {      sum=0;
         for(j=1;j<=n;j++)
        {
        u=(float)random(RAND_MAX)/RAND_MAX;
        if(u<=th)
        sum=sum+1;
        }
          x[i]=sum;
   }
         else{x[i]=0; }
         printf(" i=%d      x(I)=%d",i,x[i]);
        getch();
 }
         noofzero=0;
        for(i=1;i<=m;i++)
 {      if(x[i]==0) {noofzero=noofzero+1;}
 }
         sumx=0.0;
         for(i=1;i<=m;i++)
 {      sumx=sumx+x[i];}
```

```
         xbar=sumx/(m*n);
         printf("\n Zero O   bs are=%d xbar=%f",noofzero,xbar);
         newth=xbar;
  do{   oldth=newth;
         hth=sumx-n*newth*(m-noofzero)/(1-pow((1-newth),n));
         hdth=n*(m-noofzero)*((newth*n*pow((1-newth),n-1))+pow((1-
        newth),n)-1.0);
         hdth=hdth/((1-pow((1-newth),n))*(1-pow((1-newth),n)));
         newth=oldth-(hth/hdth);
         printf("\n th=%f hth/hdth=%f ",newth,hth/hdth);
        getch();
 }      while(fabs(newth-oldth)>0.000001);
        printf("\n thhat=%f ",newth);
        nth=pow((1-newth),n)  ;
        phat=(1-(noofzero/(float)m))/(1-nth);
        avarpi0=nth/(m*(nth-1.0)*(nth-1.0) +(m*nth*(1-nth)));
        z=(phat-1)/sqrt(avarpi0);
        if(fabs(z)>=1.96) power=power+1;
        printf("   Phat=%f ",phat);
        ath=ath+newth;  aphat=aphat+phat;
  }
        ath=ath/nosim;  aphat=aphat/nosim;
        printf("\n pi=%f Power=%f ",pi,100*power/nosim);
  }
        getch();
  }
```

A2.4 Program to find Power of the test for the parameter θ in Zero Inflated Poisson distribution

```
 # include <stdio.h>
 # include <math.h>
 # include <conio.h>
 # include <time.h>
 #include <stdlib.h>
 void main()
 {       int i,x[100],n,k,noofzero,sim,nosim=25000;
         double mu,u,prod,p,xbar,sumx,fmu,fdmu,newmu,oldmu,phat,ybar,pi0;
         double power1,muh0,avarm0,z,z3,mu0,ph0,z2,power2,power3;
         double Num,Deno,a,b,c,d,alpha;
         FILE*fp;
          fp=fopen("mk_si1.xls","w");
                 clrscr(); randomize();
           n=50;mu0=5.0;alpha=0.05;
         fprintf(fp,"\n n=%d \t mu0=%f \t alpha=%f  \t
         no.of.simulation=%d ",n, mu0, alpha, nosim);
         for(p=0.30;p<=0.70;p=p+0.10)
{
                 for(mu=5.0;mu<=9.2;mu=mu+0.2)
          {       power1=power2=power3=0.0;
                 for(sim=1;sim<=nosim;sim++)
          {        for(i=1;i<=n;i++)
          {      u=(float)random(RAND_MAX)/RAND_MAX;
                 if (u<=p)
         {       prod=1.0;
                 k=0;
                 do{ k=k+1;
                  u=(float)random(RAND_MAX)/RAND_MAX;
                  prod=prod*u;
            }     while (prod>exp(-mu));
                  x[i]=k-1;
            }
                 else{x[i]=0; }
          }
                 noofzero=0;sumx=0.0;
                 for(i=1;i<=n;i++)
         {       if(x[i]<1) {noofzero=noofzero+1;}
                  else {sumx=sumx+x[i];}
          }
                  muh0=sumx/(float)(n-noofzero);
                  newmu=muh0;
         do{oldmu=newmu;
        fmu=sumx-((n-noofzero)*oldmu/(1-exp(-oldmu)));
        fdmu=(n-noofzero)*(oldmu*exp(-oldmu) + exp(-oldmu)-
        1.0)/((1-exp(-oldmu))*(1-exp(-oldmu)));
        newmu=oldmu-(fmu/fdmu);
         printf("\n fmu =%lf  fdmu=%lf",fmu,fdmu);
}       while(fabs(newmu-oldmu)>0.000001);
         ph0=(1.0-(noofzero/(float)n))/(1-exp(-mu0));
         Num = mu0*noofzero*((1-exp(-mu0))*(1-exp(-mu0)));
        a=(n-noofzero);b=mu0*exp(-mu0);c=(-noofzero+n*exp(-
        mu0));d=noofzero*(1-exp(-mu0));
         Deno=a*(b*c+d);
         if(noofzero==0) exit;
        avarm0=Num/Deno;
          printf("\n No of Zero=%d   ",noofzero);
         getch();
        z=(newmu-mu0)/sqrt(avarm0);
         printf("\n z=%lf   ",z);
        if(fabs(z)>=1.96) power1=power1+1;
         xbar=sumx/(float)n;
         z2=sqrt(n)*(xbar-mu0)/(sqrt(xbar));
          if(fabs(z2)>=1.96) power2=power2+1;
          pi0=(n-noofzero)/(n*(1-exp(-mu0)));
         z3= sqrt(n)*pi0*(xbar/pi0-mu0);
         z3=z3/sqrt(mu0*(1+mu0*(1-pi0)));
         if(fabs(z3)>=1.96) power3=power3+1;
        printf("\n sim no=%d power1=%lf power2=%lf mu=%f
        ",sim,100*power1/sim,100*power2/sim,mu);
 }
         fprintf(fp,"\n p=%3.2f \t mu=%4.2f  \t %lf  \t %lf \t %lf  ",p,
        mu,
        100*power1/sim, 100*power2/sim,100*power3/sim);
        printf("\n p=%3.2f  mu=%4.2f  power1=%lf  power2=%lf
        power3=%lf ",p, mu,  100*power1/sim,
        100*power2/sim,100*power3/sim);
         printf("\n p=%f   Power1=%f ",p,100*power1/nosim);
}
}
         fclose(fp);
        getch();
}
```