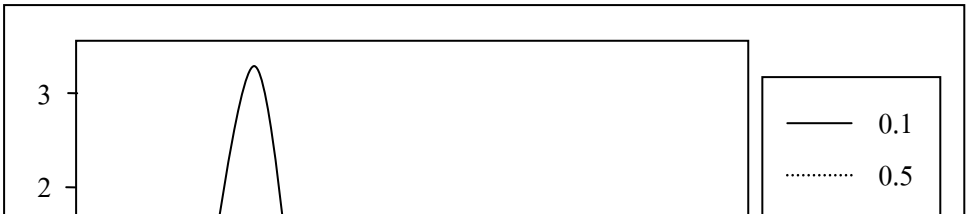
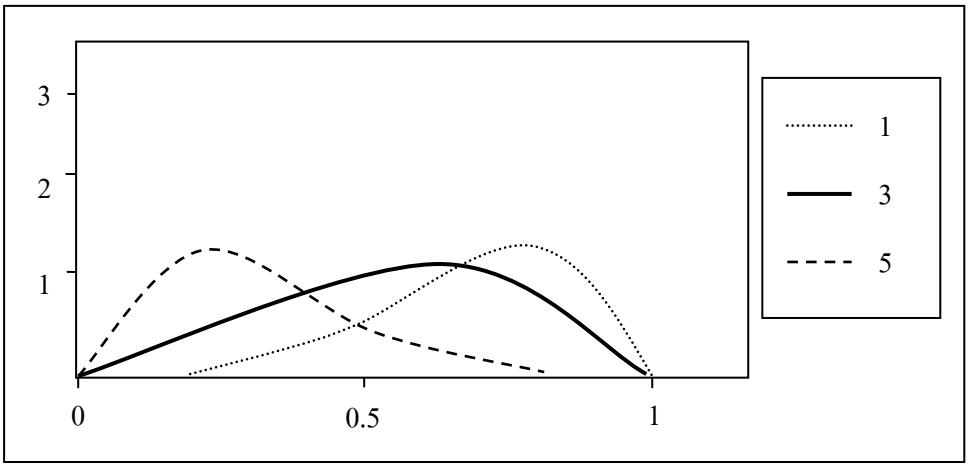
ภาคผนวก

ภาคผนวก ก

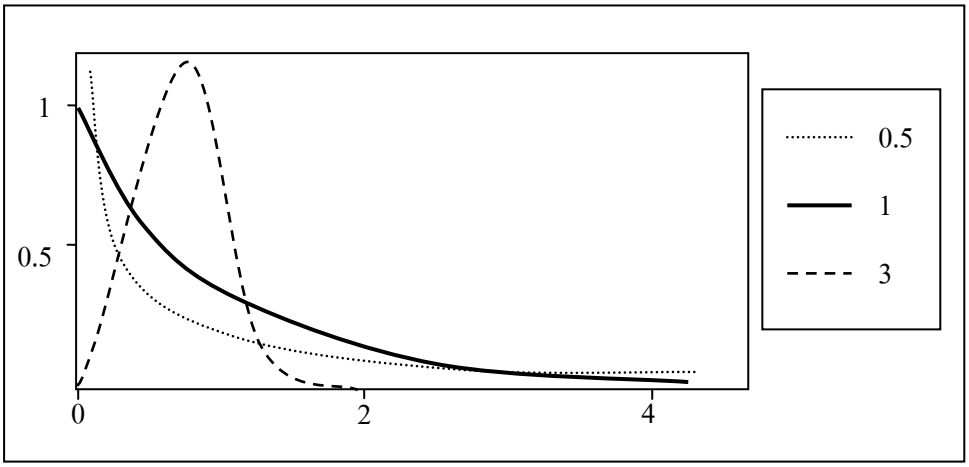
กราฟแสดงลักษณะเส้นโค้งของการแจกแจงแบบต่างๆ



ภาพผนวกที่ ก1 กราฟแสดงลักษณะเส้นโค้งของการแจกแจงแบบลอกนอร์มอลที่มีค่าพารามิเตอร์ σ เท่ากับ 0.1, 0.5, 1 และ 1.5 ตามลำดับ



ภาพผนวกที่ ก2 กราฟแสดงลักษณะเส้นโค้งของการแจกแจงแบบเบต้าที่มีค่าพารามิเตอร์ β เท่ากับ 1, 3 และ 5 ตามลำดับ



ภาพผนวกที่ ก3 กราฟแสดงลักษณะเส้นโค้งของการแจกแจงแบบไวบูลล์ที่มีค่าพารามิเตอร์ β เท่ากับ 0.5, 1 และ 3 ตามลำดับ

ภาคผนวก ข

โปรแกรมที่ใช้ในการวิจัย

```
/*************************************************
```

```
*                    A generator for random  numbers                    *
*********************************************************/
do i=1 to n;
        x(i ) = ranuni(seed);
end;

/*********************************************************
*                  A generator for log-normal distribution              *
*********************************************************/
mean = 4;
s2 = sqrt(1);
do i=1 to n;
        z(i) = 0+sqrt(1)*rannor(seed);
        u(i) = mu+(s2*z(i));
        y(i) = exp(u(i));
end;

/*********************************************************
*                    A generator for  beta  distribution                *
*********************************************************/
do i=1 to n;
        g1=rangam(seed,4);
        g2=rangam(seed,1);
        y(i)=g1/(g1+g2);
end;

/*********************************************************
*                    A generator for  weibull  distribution             *
*********************************************************/
mean = 5;

s2 = 0.5;
```

```
do i=1 to n;
        v(i) = (-1/mu)*log(x(i));
        y(i) = v(i)*exp(1/s2);
end;

/*****************************************************
*                        least – square  method                        *
*****************************************************/
options pageno=1 nodate;
title ‘least – square  method';
dm log 'clear'; dm output 'clear';
%let nobs=1000;
data simulation;
        array x x1-x10;
        array y y1-y10;
        array xy xy1-xy10;
        array xx xx1-xx10;
        array sxy sxy1-sxy10;
        array sx sx1-sx10;
        array sy sy1-sy10;
        array sxx sxx1-sxx10;
        array xbar xbar1-xbar10;
        array ybar ybar1-ybar10;
        array xbarx xbarx1-xbarx10;
        array nx_y nx_y1-nx_y10;
        array nx_x nx_x1-nx_x10;
        array c c1-c10;
        array d d1-d10;
        array b b1-b10;
        array e e1-e10;

        array a a1-a10;
```

```
        array yp yp1-yp10;
        n = 10;
        seed = 12345;

do id=1 to &nobs;
do i=1 to n;
        xy(i) = x(i)*y(i);
        xx(i) = x(i)*x(i);
        sxy(i) = sum(of  xy1-xy10);
        sx(i) = sum(of  x1-x10);
        sy(i) = sum(of  y1-y10);
        sxx(i) = sum(of  xx1-xx10);
        xbar(i) = sx(i)/n;
        ybar(i) = sy(i)/n;
        xbarx(i) = xbar(i)*xbar(i);
        nx_y(i) = n*xbar(i)*ybar(i);
        nx_x(i) = n*xbarx(i);
        c(i) = sxy(i)-nx_y(i);
        d(i) = sxx(i)-nx_x(i);
        b(i) = c(i)/d(i);
        e(i) = b(i)*xbar(i);
        a(i) = ybar(i)-e(i);
        yp(i) = a10+(b10*x(i));
end;

/*****************************************************
*                least absolute deviations regression              *
*****************************************************/
options pageno=1 nodate;
title 'least  absolute  deviations  regression';

dm log 'clear'; dm output 'clear';
```

```
%let nobs=1000;
data simulation;
        array x x1-x10;
        array y y1-y10;
        array yd yd1-yd10;
        array xd xd1-xd10;
        array ydx ydx1-ydx10;
        array axd axd1-axd10;
        array rydx rydx1-rydx10;
        array raxd raxd1-raxd10;
        array cus cus1-cus10;
        array yo yo1-yo10;
        array xo xo1-xo10;
        array xx xx1-xx10;
        array yy yy1-yy10;
        array Newy Newy1-Newy10;
        array Newx Newx1-Newx10;
        array yp yp1-yp10;
        n = 10;
        seed = 12345;

do id=1 to &nobs;

/*A generator for log-normal distribution*/
/*A generator for beta distribution*/
/*A generator for weibull distribution*/

do i=1 to n;
            yd(i) = y(i)-y1;
            xd(i) = x(i)-x1;

            ydx(i) = yd(i)/xd(i);                                    /*****slope*****/
```

```
        axd(i) = abs(xd(i));
        rydx(i) = ydx(i);
        raxd(i) = axd(i);
        xx(i) = x(i);
        yy(i) = y(i);                                   /*****create variable*****/
end;

do i=1 to n-1;
do j=i+1 to n;
        if  rydx(i) >= rydx(j) then do;
            ntemp = rydx(i);
            mtemp = raxd(i);
            xtemp = xx(i);
            ytemp = yy(i);
            rydx(i) = rydx(j);
            raxd(i) = raxd(j);                          /*****sort  data*****/
            xx(i) = xx(j);
            yy(i) = yy(j);
            rydx(j) = ntemp;
            raxd(j) = mtemp;
            xx(j) = xtemp;
            yy(j) = ytemp;
        end;
end;
end;

do i=1 to n;
do j=i+1 to n;
        cus(i) = raxd(i);                               /*****create variable*****/
        if cus(i) >= 0 then do;                         /*****cumulative sum*****/

           ntemp = cus(i)+raxd(j);
```

```
            cus(j) = ntemp;
          end;
end;
end;

do i=1 to n;
          ndiv = cus10/2;
          if cus(i)  >  ndiv then do;
            yo1 = yy(i);                          /*****yo1=chack end loop*****/
            xo1 = xx(i);                          /*****xo1=chack end loop*****/
            Newy(1) = yo1;
            Newx(1) = xo1;
            i = n;
          end;
end;

b = (y1-yold)/(x1-xold);                          /*****slope b*****/
a = y1-(b*x1);                                    /*****constant a*****/

do i=1 to n;
          yp(i) = a+(b*x(i));                     /*****regression line*****/
end;

/*******************************************************
*                          M  regression                *
*******************************************************/
options pageno=1 nodate;
title ' M  regression';
dm log 'clear'; dm output 'clear';
%let nobs=1000;

data simulation;
```

```
array x x1-x10;
array y y1-y10;
array xy xy1-xy10;
array xx xx1-xx10;
array sxy sxy1-sxy10;
array sx sx1-sx10;
array sy sy1-sy10;
array sxx sxx1-sxx10;
array xbar xbar1-xbar10;
array ybar ybar1-ybar10;
array xbarx xbarx1-xbarx10;
array nx_y nx_y1-nx_y10;
array nx_x nx_x1-nx_x10;
array c c1-c10;
array d d1-d10;
array b b1-b10;
array e e1-e10;
array a a1-a10;
array yo yo1-yo10;
array eo eo1-eo10;
array abseo abseo1-abseo10;
array euvo euvo1-euvo10;
array elvo elvo1-elvo10;
array ea ea1-ea10;
array yea yea1-yea10;
array yp yp1-yp10;
n = 10;
seed = 12345;

do id=1 to &nobs;
```

```
/*A generator for log-normal distribution*/
/*A generator for beta distribution*/
/*A generator for weibull distribution*/

do i=1 to n;
        xy(i) = x(i)*y(i);
        xx(i) = x(i)*x(i);
        sxy(i) = sum(of xy1-xy10);
        sx(i) = sum(of x1-x10);
        sy(i) = sum(of y1-y10);
        sxx(i) = sum(of xx1-xx10);
        xbar(i) = sx(i)/n;
        ybar(i) = sy(i)/n;
        xbarx(i) = xbar(i)*xbar(i);
        nx_y(i) = n*xbar(i)*ybar(i);
        nx_x(i) = n*xbarx(i);
        c(i) = sxy(i)-nx_y(i);
        d(i) = sxx(i)-nx_x(i);
        b(i) = c(i)/d(i);                       /*****slope b*****/
        e(i) = b(i)*xbar(i);
        a(i) = ybar(i)-e(i);                    /*****constant a*****/
        yo(i) = a10+(b10*x(i));                 /*****old  regression line*****/
        eo(i) = y(i)-yo(i);
        abseo(i) = abs(eo(i));
        median = median(of abseo:);
        vo = 1.483*median;                      /*****calculate variance*****/
        uvo = 1.5*vo;
        lvo = -1.5*vo;
        euvo(i) = uvo;

        elvo(i) = lvo;                          /*****create variable*****/
```

```
        ea(i) = eo(i);
        if eo(i) > euvo(i) then ea(i) = euvo(i);                    /*****create  condition*****/
        else do;
        if eo(i) < elvo(i) then ea(i) = elvo(i);
        else ea(i) = eo(i);
        end;
        yea(i) = yo(i)+ea(i);
        yp(i) = a10+(b10*x(i));                                     /*****new  regression line*****/
end;

/*****************************************************
*     nonparametric regression using the ranks of number      *
*****************************************************/
options pageno=1 nodate;
title ' nonparametric regression using the ranks of number';
dm log 'clear'; dm output 'clear';
%let nobs=1000;
data simulation;
        array x x1-x10;
        array y y1-y10;
        array ba ba1-ba10;
        array bb bb1-bb10;
        array bc bc1-bc10;
        array bd bd1-bd10;
        array be be1-be10;
        array bf bf1-bf10;
        array bg bg1-bg10;
        array bh bh1-bh10;
        array bi bi1-bi10;
        array wa wa1-wa10;

         array wb wb1-wb10;
```

```
array wc wc1-wc10;
array wd wd1-wd10;
array we we1-we10;
array wf wf1-wf10;
array wg wg1-wg10;
array wh wh1-wh10;
array wi wi1-wi10;
array wwa wwa1-wwa10;
array wwb wwb1-wwb10;
array wwc wwc1-wwc10;
array wwd wwd1-wwd10;
array wwe wwe1-wwe10;
array wwf wwf1-wwf10;
array wwg wwg1-wwg10;
array wwh wwh1-wwh10;
array wwi wwi1-wwi10;
array b b1-b45;
array w w1-w45;
array rb rb1-rb45;
array rw rw1-rw45;
array cus cus1-cus45;
array sumy sumy1-sumy10;
array ybar ybar1-ybar10;
array sumx sumx1-sumx10;
array xbar xbar1-xbar10;
array yp yp1-yp10;
n = 10;
seed = 12345;
m = (n*(n-1))/2;


do id=1 to &nobs;
```

```
/*A generator for log-normal distribution*/
/*A generator for beta distribution*/
/*A generator for weibull distribution*/

calx = x(1);
caly = y(1);

do i=2 to n;
do j = i-1 to n-1;
        ba(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=3 to n;
do j = i-1 to n-2;
        bb(j) = (caly-y(i))/(calx-x(i));
end;
end;
do i=4 to n;
do j = i-1 to n-3;
        bc(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=5 to n;
do j = i-1 to n-4;
        bd(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=6 to n;
```

```
do j = i-1 to n-5;
        be(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=7 to n;
do j = i-1 to n-6;
        bf(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=8 to n;
do j = i-1 to n-7;
        bg(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=9 to n;
do j = i-1 to n-8;
        bh(j) = (caly-y(i))/(calx-x(i));
end;
end;

do i=10 to n;
do j = i-1 to n-9;
        bi(j) = (caly-y(i))/(calx-x(i));
end;
end;
```

/*****set b-value*****/

```
b1=ba1;   b2=ba2;   b3=ba3;   b4=ba4;   b5=ba5;
b6=ba6;   b7=ba7;   b8=ba8;   b9=ba9;
b10=bb1;  b11=bb2;  b12=bb3;  b13=bb4;  b14=bb5;
b15=bb6;  b16=bb7;  b17=bb8;
b18=bc1;  b19=bc2;  b20=bc3;  b21=bc4;  b22=bc5;
b23=bc6;  b24=bc7;
b25=bd1;  b26=bd2;  b27=bd3;  b28=bd4;  b29=bd5;
b30=bd6;
b31=be1;  b32=be2;  b33=be3;  b34=be4;  b35=be5;
b36=bf1;  b37=bf2;  b38=bf3;  b39=bf4;
b40=bg1;  b41=bg2;  b42=bg3;
b43=bh1;  b44=bh2;
b45=bi1;

do i=2 to n;
do j = i-1 to n-1;
        wa(j) = abs(calx-x(i));
end;
end;

do i=3 to n;
do j = i-1 to n-2;
        wb(j) = abs (calx-x(i));
end;
end;

do i=4 to n;
do j = i-1 to n-3;
        wc(j) = abs (calx-x(i));
end;

end;
```

```
do i=5 to n;
do j = i-1 to n-4;
        wd(j) = abs(calx-x(i));
end;
end;

do i=6 to n;
do j = i-1 to n-5;
        we(j) = abs(calx-x(i));
end;
end;

do i=7 to n;
do j = i-1 to n-6;
        wf(j) = abs(calx-x(i));
end;
end;

do i=8 to n;
do j = i-1 to n-7;
        wg(j) = abs(calx-x(i));
end;
end;

do i=9 to n;
do j = i-1 to n-8;
        wh(j) = abs(calx-x(i));
end;
end;

do i=10 to n;
```

```
do j = i-1 to n-9;
        wi(j) = abs(calx-x(i));
end;
end;

swa = sum(of wa1-wa9);
swb = sum(of wb1-wb8);
swc = sum(of wc1-wc7);
swd = sum(of wd1-wd6);
swe = sum(of we1-we5);
swf = sum(of wf1-wf4);
swg = sum(of wg1-wg3);
swh = wh1+wh2;
swi = wi1;
sw  = swa+swb+swc+swd+swe+swf+swg+swh+swi;
do i=1 to n;
        wwa(i) = wa(i)/sw;
        wwb(i) = wb(i)/sw;
        wwc(i) = wc(i)/sw;
        wwd(i) = wd(i)/sw;
        wwe(i) = we(i)/sw;
        wwf(i) = wf(i)/sw;
        wwg(i) = wg(i)/sw;
        wwh(i) = wh(i)/sw;
        wwi(i) = wi(i)/sw;
end;

/*****set w-value*****/
w1=wwa1;   w2=wwa2;   w3=wwa3;   w4=wwa4;   w5=wwa5;


w6=wwa6;   w7=wwa7;   w8=wwa8;   w9=wwa9;
```

```
w10=wwb1;  w11=wwb2;  w12=wwb3;  w13=wwb4;  w14=wwb5;
w15=wwb6;  w16=wwb7;  w17=wwb8;
w18=wwc1;  w19=wwc2;  w20=wwc3;  w21=wwc4;  w22=wwc5;
w23=wwc6;  w24=wwc7;
w25=wwd1;  w26=wwd2;  w27=wwd3;  w28=wwd4;  w29=wwd5;
w30=wwd6;
w31=wwe1;  w32=wwe2;  w33=wwe3;  w34=wwe4;  w35=wwe5;
w36=wwf1;  w37=wwf2;  w38=wwf3;  w39=wwf4;
w40=wwg1;  w41=wwg2;  w42=wwg3;
w43=wwh1;  w44=wwh2;
w45=wwi1;

do i=1 to m;
        rb(i) = b(i);
        rw(i) = w(i);                                   /*****create variable*****/
        cus(i) = rw(i);
end;

do i=1 to m-1;
do j=i+1 to m;
        if  rb(i) >= rb(j) then do;
            ntemp = rb(i);
            mtemp = rw(i);
            rb(i) = rb(j);                              /*****sort  data*****/
            rw(i) = rw(j);
            rb(j) = ntemp;
            rw(j) = mtemp;
        end;
end;
end;

do i=1 to m;
```

```
do j=i+1 to m;
        if  cus(i) >= 0 then do;
            ntemp = cus(i)+rw(j);                    /*****cumulative sum*****/
            cus(j) = ntemp;
        end;
end;
end;

do i=1 to m;
        if  cus(i) > 0.5 then do;                    /*****create  condition*****/
            cij = cus(i);
            bij = rb(i);                             /*****slope b*****/
            i = m;
        end;
        sumy(i) = sum(of  y1-y10);
        ybar(i) = sumy(i)/n;
        sumx(i) = sum(of  x1-x10);
        xbar(i) = sumx(i)/n;
        aij = ybar(i)-(bij*xbar(i));                 /*****constant a*****/
        yp(i) = aij+(bij*x(i));                      /*****regression line*****/
end;

/*****************************************************
*                        print  data                 *
*****************************************************/
proc print data=simulation;
run;
```