```
proc bspline
opt int  dian 1
opt choice salida 1 deta nodet
opt choice overnight 1 si no
option choice vto  1 enteros fechas

dec rect[series]n_v
dec rect[rect] final b_qsp db_qsp dtij ddtij dij_tp ddij_tp
dec rect resul cf tp ones BSP dBSP bsp1 dbsp1 bpk dbpk wr $
    djs cup dil pr tij mae rmse ytemp1 ytemp2 ptemp gridt desc ndesc nbsp nbsp1 nbpk ndbpk lambdas bpks
prod1 prod unos unos1  tp1 efe piest
dec vect temp seqa fmh yyyy1 temp1 np mh m c dd1 mm1 madurez_r nodos $
    vned tps one y00 y0 y yobs entr wm


dim resul(16,1)


 *FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF*
 *                 Fecha de transacción a estimar                      *
                        com fecha = dian
 *FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF*


 ************ Lectura de datos ***************

dim n_v(1,4000)
com nf=0
do i=1,%cols(n_v)
 set n_v(1,i) = i{0}
end do i


do i=2,%cols(n_v)
 set pru = n_v(1,i)-n_v(1,1)
 acc pru
 com nf=(nf+1)         ;*<<<<<< Número de días de transacción  ****
 if(%maxvalue(pru).eq.0.0) {
  break
  }
end do i
dim final(1,fix(nf))
 make tdate 1 1
 #fecha

 dis 'Estimaciones para' tdate
 make datos 2 *
 #fecha fecha+nf fecha+(2*nf) fecha+(3*nf)
 dis 'datos' datos
********************************************
 dim np(%rows(datos)) mh(%rows(datos)) fmh(%rows(datos)) yyyy1(%rows(datos)) m(%rows(datos)) $
    c(%rows(datos)) dd1(%rows(datos)) mm1(%rows(datos)) madurez_r(%rows(datos)) pr(%rows(datos),1)

***************** Calculo de TP y CF ***********************
 ewise m(i)=datos(i,1)
 ewise c(i)=datos(i,2)

 comp n = %rows(m)

 *** paso de la fecha de transacción a real ***
 com  x13 = tdate(1,1)
 com yyyy = fix(x13)
 com mm = fix( 100*(x13-yyyy) );
 com dd = 10000*(x13 - yyyy - float(mm)/100);
 com tdate_r  = yyyy + (mm-1.0)/12 + (dd-1.0)/365;
 if(vto.eq.1){

  ewi madurez_r(i) = m(i)/365
  ewise  m(i) = m(i)/365
  }
 else {
 *** paso de las fecha de maduración a reales ***
  ewi yyyy1(i) = fix(m(i))
  ewi mm1(i) = fix( 100*(m(i)-yyyy1(i)) )
  ewi dd1(i) = 10000*(m(i) - yyyy1(i) - (mm1(i)+0.0)/100)
  ewi madurez_r(i) = yyyy1(i) + (mm1(i)-1.0)/12 + (dd1(i)-1.0)/365
 *** descuento de la fecha de transacción ****
 ewise  m(i) = madurez_r(i) - tdate_r
  }

com fechas_r=m
```

```
 comp h = 1.0
 comp mh = m*h
 ewise fmh(j) = fix(mh(j))
 do fil=1,n
   comp uno = 0.0
   if (mh(fil).gt.fmh(fil)); comp uno = 1.0
   comp np(fil) = fmh(fil) + uno;
 end do fil

 comp k = %MAXVALUE(np)
 dim cf(n,fix(k)) tp(n,fix(k))
 *** Valores iniciales de CF y TP ****
 ewise cf(i,j) = 0.0
 ewise tp(i,j) = 999.0

 **** Calculo de CF y TP ****
 comp j = 1
 while (j.le.n)  {
   if (np(j).gt.1) {
     dim ones(1,fix(np(j))-1)
     ewise ones(i,j) = 1.0
     comp temp = (c(j)/h)*ones
     do col=1,(fix(np(j))-1)
       comp cf(j,col) = temp(col)
     end do col
   }
   comp cf(j,fix(np(j))) = 100 + c(j)/h
   dim seqa(fix(np(j)))
   ewise seqa(j) = 0.0
   do i=2,fix(np(j))
     comp seqa(i) =  seqa(i-1) + 1/h
   end do i
   dim temp1(fix(np(j)))
   comp temp2 = m(j) - fix(m(j)*h)/h + ( 1/h - (m(j) > fix(m(j)*h)/h)/h )
   ewise temp1(i) = temp2 + seqa(i)
   do i=1,fix(np(j))
     comp  tp(j,i) = temp1(i)
   end do i
   comp j = j+1
  }

 if(salida.eq.1) {
  dis "cf" cf "tp" tp
  }
 dim b_qsp(%rows(tp),2) db_qsp(%rows(tp),2) dtij(%rows(tp),1) ddtij(%rows(tp),1) $
     dij_tp(%rows(tp),2) ddij_tp(%rows(tp),2) tij(1,%cols(tp)) y00(%rows(cf)) $
     y0(%rows(cf)) y(%rows(cf)) ytemp1(%cols(cf),%rows(cf)) ytemp2(%cols(cf),%rows(cf)) $
     ptemp(%cols(cf),%rows(cf))



 ********* funciones b-spline*****************

 com k=3     ;* GRADO DE LA FUNCION B-SPLINE

 ***<<<<<Cálculo del número y posición de los nodos internos al estilo McCulloch>>>>>>*****************
 com ni=%if(sqrt(%rows(datos))-
fix(sqrt(%rows(datos))).le.0.5,fix(sqrt(%rows(datos))),fix(sqrt(%rows(datos)))+1)
 if(vto.eq.1) {
  set madurez = madurez_r(t)
  }
  else {
   set madurez = madurez_r(t) - tdate_r
  }
 make mat
 #madurez
 dim djs(fix(ni),1) nodos(fix(k)+%rows(djs)) vned(fix(k)) tps(fix(%rows(djs)+2*k))
dis 'ni' fix(ni)
dis 'mat' mat
 do i=1,fix(ni)
  com l=fix(((i-1)*%rows(mat))/((ni)-1))
  com theta=(((i-1)*%rows(mat))/((ni)-1))-l
  if (l.eq.0.0) {
   comp dj=0.0
   }
  else {
   if (theta.eq.0.0) {
     com dj = mat(fix(l),1)
     }
   else {
     com dj = mat(fix(l),1) + theta * (mat(fix(l+1),1)-mat(fix(l),1))
     }
```

```
  }
 com djs(i,1)=dj
end do i
com ni=%rows(djs)-1

if(salida.eq.1) {
 dis ' Número de segmentos (n)' fix(ni)
}

ewi djs(i,j)=%if(djs(i,j).eq.0.0,djs(i,j),fix(djs(i,j))+1)
com epsilon = 0.001
do i=2,%rows(djs)
  com djs(i,1)=%maxvalue(madurez)*((i-1.0)/(%rows(djs)-1.0))
end do i
com djs(%rows(djs),1)= djs(%rows(djs),1) + epsilon
if(salida.eq.1) {
 dis 'Posición de los nodos' djs
 }

com k=3
com ele=0.0


do cc=fix(-k),fix(ni-1)
 com ele=ele+1
end do cc
dim dil(%rows(cf),fix(ele)) BSP(%cols(tp),fix(ele)) dBSP(%cols(tp),fix(ele))
com nei=k
com ned=k
com tdjs=tr(djs)
ewi nodos(j)= %if(j.le.k,-k+j-1,tdjs(1,fix(j-k)))

******************vector de nodos extras a la derecha*********************
ewi vned(j)=%maxvalue(madurez)*(j+1.0)
*****************************************************************************
ewi tps(j)=%if(j.le.fix(%rows(djs)+k),nodos(j),vned(fix(j-(%rows(djs)+k))))
dis 'tps' tps
*****************************
com dimf=%if(k.eq.1,ele,(2*k))
dim bsp1(fix(dimf),1) dbsp1(fix(dimf),1) bpk(fix(dimf),fix(k)) dbpk(fix(dimf),fix(k)) $

ewi bsp(i,j) =0.0
ewi dbsp(i,j)=0.0
*_____________________________________*
* para crear el vector W de la restricción *
*_____________________________________*
dim entr(1)
set mintp 1 %rows(tp) = tp(t,1)
extremum(noprint) mintp
com entr(1) = %minent
dim wr(1,fix(ele))
dim wm(1)
ewi wm(j)=1  ;*((datos(fix(entr(1)),4)/100))
*_____________________________________*

com con=0
do ache=fix(-k),fix(-k+k+1-1)
 com con=con+1
end do ache
dim bpks(1,con)
com con=0

do l=fix(-k),fix(-k+k+1)
 com con=con+1
end do l
dim prod(1,con-1)
dim prod1(1,con)
dim unos(con,1)
ewi unos(i,j)=1.0

como con=0
do p=fix(-k),fix(ni-1)
 como con=con+1
end do p
dim bpk(1,con)
dim efe(%rows(tp),%cols(tp))
dim unos1(con,1)
dim tp1(%rows(tp)+1,%cols(tp))
ewi tp1(i,j)=%if(i.eq.%rows(tp)+1,%if(j.eq.1,0.0,999.0),tp(i,j))

do ii=1,%rows(tp1)
 dim cup(1,%cols(cf)) ;ewi cup(i,j)=%na
```

```
  do jj=1,%cols(tp1)
   ewi bpk(i,j) =0.0
   com pos=-1
   comp con=0
   comp noif = 1
   do p=fix(-k),fix(ni-1)
    comp con=con+1
    com pos=pos+1
    comp noif = 0
     com con1=0
     do l=p,fix(p+k+1)
      com con1=con1+1
      com con2=0
      com con3=0
      do ache=p,fix(p+k+1)
       com con2=con2+1
       if(ache.ne.l) {
       com con3=con3+1
       com bpks(1,con3)=1.0/(tps(con2+pos)-tps(con1+pos))
       }
      end do ache
      ewi prod(i,j)=%if(j.eq.1,bpks(1,j),prod(1,j-1)*bpks(1,j))
      com prod1(1,con1)=prod(1,%cols(prod))*%max(0.0,(tp1(ii,jj)-tps(con1+pos)))**k
     end do l
     com bpkx=prod1*unos
     com
bpk(1,con)=%if(tp1(ii,jj).ge.tps(fix(pos+1)).AND.tp1(ii,jj).le.tps(fix(pos+k+1+1)),%scalar(bpkx),0.0)
  end do p
   do hh=1,fix(ele)
    if(ii.le.%rows(tp)) {
     comp bsp(jj,hh)=bpk(1,hh)
     }
    else {
     if(jj.eq.1) {
      com wr(1,hh)= bpk(1,hh)
      }
     }
   end do hh
   com cup(1,jj)=cf(ii,jj)
   com tij(1,jj)=tp(ii,jj)
  end do jj
  if(ii.le.%rows(tp)) {
   com b_qsp(ii,1)=bsp
   com dij_tp(ii,2)=tr(tij)
   com suma=cup*bsp
  do ww=1,fix(ele)
   com dil(ii,ww)=suma(1,ww)
  end do ww
  }
 end do ii

dis 'dil' dil

 ewi pr(i,j)=datos(i,4)
 dim one(%rows(pr)) mae(%rows(mat),1) rmse(%rows(pr),1)
 ewi one(j)=1.0
 com alfa=inv(tr(dil)*dil)*tr(dil)*pr


 if(overnight.eq.1) {
  matrix alfar=alfa+(inv(tr(dil)*dil))*tr(wr)*inv(wr*inv(tr(dil)*dil)*tr(wr))*(wm-wr*alfa)
  matrix sigmaur=(tr(pr-dil*alfar)*(pr-dil*alfar))*inv(%scalar(%rows(pr)-ele))
  com s2r1=%scalar(sigmaur)
  matrix sigmawr=s2r1*(inv(tr(dil)*dil))-
s2r1*inv(tr(dil)*dil)*tr(wr)*inv(wr*inv(tr(dil)*dil)*tr(wr))*wr*inv(tr(dil)*dil)
  do i=1,%rows(b_qsp)
   com b_qsp(i,2)=b_qsp(i,1)*sigmawr*tr(b_qsp(i,1))
  end do i
  do i=1,%rows(b_qsp)
   com dtij(i,1)=b_qsp(i,1)*(alfar)
  end do i

  matrix piest = dil*alfar
  if(salida.eq.1) {
   dis 'Vector estimado de parámetros restringidos' alfar
   dis 'Precios estimados' piest
   dis 'Precios observados' pr
  }
 }
 else {
  if(salida.eq.1) {
   dis  'Vector estimado de parámetros' alfa
```

```
   dis 'Precios estimados' piest
   dis 'Precios observados' pr
   }
  matrix sigmau=(tr(pr-dil*alfa)*(pr-dil*alfa))*inv(%scalar(%rows(pr)-ele))
  com s2r1=%scalar(sigmau)
  matrix sigmaw=s2r1*(inv(tr(dil)*dil))
  do i=1,%rows(b_qsp)
   com b_qsp(i,2)=b_qsp(i,1)*sigmaw*tr(b_qsp(i,1))
  end do i
  do i=1,%rows(b_qsp)
   com dtij(i,1)=b_qsp(i,1)*(alfa)
  end do i
  matrix piest=dil*alfa
  }

 do i=1,%rows(tp)
  com dij_tp(i,1) =dtij(i,1)
 end do i
 set opi = pr(t,1)
 stat(noprint) opi
 com desv=sqrt(%variance)

 ewi mae(i,j)= abs(pr(i,1)-piest(i,1))
 comp mae=(tr(one)*mae)*%scalar((1/float(%rows(pr))))
 dis 'MAE de Precios' mae

 ewi rmse(i,j)= (pr(i,1)-piest(i,1))**2
 comp rmse=sqrt(%scalar(tr(one)*rmse)*%scalar((1/float(%rows(pr)))))
 dis 'RMSE de Precios' rmse

 set dcto = %na
 set st = %na
 set tj = %na
 set ft = %na
com con=0
 do i=1,%rows(dij_tp)
  do ii=1,%rows((dij_tp)(i,1))
   if (((dij_tp)(i,1)(ii,1)).ne.0) {
    com con=con+1
    com dcto(fix(con)) = (dij_tp)(i,1)(ii,1)
    com st(fix(con)) = ((1.0/(dij_tp)(i,1)(ii,1))**(1.0/(dij_tp)(i,2)(ii,1)))-1.0
    com tj(fix(con)) = dij_tp(i,2)(ii,1)+tdate(1,1)
}
  end do ii
 end do i

 ************************************************************************************
 *** Cálculo de los rendimientos (Yt) dado precios (Pt),    utilizando el algoritmo     ***
 ***                              de Newton-Raphson                                   ***
 ************************************************************************************
 comp Pt = piest
 comp h_ncomp = 1
 comp ytol    = 0.00000000001

 ewise y00(i) = 100*h_ncomp*( (cf(i,1)/pt(i,1))**(1/(tp(i,1)*h_ncomp)) - 1 )
 ewise y0(i)  = (cf(i,2)==0)*y00(i) + (cf(i,2)>0)*cf(i,1)*h_ncomp
 comp eps     = 1.0
 while eps > ytol {
    ewise ytemp1(i,j) = (1.0+y0(j)/(100*h_ncomp))**(-tp(j,i)*h_ncomp-1)
    ewise ytemp2(i,j) = ((-tp(j,i)*h_ncomp)*ytemp1(i,j)) / (100*h_ncomp)
    comp Dp           = %xdiag(cf * ytemp2)
    ewise ptemp(i,j)  = (1.0+y0(j)/(100*h_ncomp))**(-tp(j,i)*h_ncomp)
    comp p0           = %xdiag(cf*ptemp)
    ewise y(i)        = y0(i) + (cf(i,2) > 0)*(pt(i,1)-p0(i))/Dp(i)
    comp eps          = %MAXVALUE( %abs(y-y0) )
    comp y0           = %abs(y); ** to prevent crash **
  }
 comp Yt = y

 ************************************************************************************
 dim yobs(%rows(yt))
 ewise yobs(i)=datos(i,3)

 ***** Estadísticas MAE y RMSE para rendimientos *****
 comp mae_y  = %avg( %abs(yt - yobs) )
 comp rmse_y = %sqrt(tr(yt - yobs)*(yt - yobs)*(1.0/%rows(yt)))

dis 'MAE de Rendimientos' mae_y

dis 'RMSE de Rendimientos' rmse_y
```

```
dis 'vencimientos' m
dis 'Rendim. al vencim. observado' yobs
dis 'Rendim. al vencim. Estmado' yt
*********************************************************************************************
com incremento  = 0.01
com con = 0
com epsilon = 0.0000001

com tiemp = epsilon
while(tiemp.le.(djs(%rows(djs),1)+epsilon)) {
 com con=con+1
 comp tiemp = %minvalue(tp)+tiemp
 }

dim gridt(fix(con),1)
dim ndesc(con,1)
com epsilon = 0.0000001
com con = 0
com tiemp = epsilon
while(tiemp.le.(djs(%rows(djs),1)+epsilon)) {
 com con=con+1
 comp tiemp = %minvalue(tp)+tiemp    ;*tiemp + incremento
 com gridt(fix(con),1)=tiemp
 }


dim BSP(%cols(gridt),fix(ele)) ;*ewi bsp(i,j)=0.0

com dimf=%if(k.eq.1,ele,(2*k))
dim bsp1(fix(dimf),1)
ewi bpk(i,j)=0.0

 do ii=1,%rows(gridt)
  com pos=-1
  comp con=0
  comp noif = 1
  do p=fix(-k),fix(ni-1)
   comp con=con+1
   com pos=pos+1
   comp noif = 0
   com con1=0
    do l=p,fix(p+k+1)
      com con1=con1+1
      com con2=0
      com con3=0
      do ache=p,fix(p+k+1)
       com con2=con2+1
       if(ache.ne.l) {
       com con3=con3+1
       com bpks(1,con3)=1.0/(tps(con2+pos)-tps(con1+pos))
       }
      end do ache
      ewi prod(i,j)=%if(j.eq.1,bpks(1,j),prod(1,j-1)*bpks(1,j))
      com prod1(1,con1)=prod(1,%cols(prod))*%max(0.0,(gridt(ii,1)-tps(con1+pos)))**k
     end do l
     com bpkx=prod1*unos
     com bpk(1,con)=%scalar(bpkx)
   end do p

   do hh=1,fix(ele)
    comp bsp(1,hh)=bpk(1,hh)
   end do hh
  if(overnight.eq.1) {
   com desc=tr(alfar)*tr(bsp)
   }
  else {
   com desc=tr(alfa)*tr(bsp)
  }
  com ndesc(ii,1)=desc(1,1)
 end do ii



 set Descuento 1 %rows(gridt) = %if(t.le.%rows(gridt),ndesc(t,1),%na)
 set Tspot 1 %rows(gridt) = %if(t.le.%rows(gridt),(((1.0/(Descuento))**(1.0/(gridt(t,1))))-1.0),%na)
 set Tforward 2 %rows(gridt) = %if(t.gt.1.and.t.le.%rows(gridt),((Descuento{1}/Descuento)-1.0)*100,%na)
 set Anos 1 %rows(gridt) = gridt(t,1)

 set anos 1 %rows(gridt) = %if(t.eq.1,fechas_r(1),fechas_r(1)+gridt(t-1,1))
 set mes 1 %rows(gridt) = fix((12.0*(anos-fix(anos)))+1.0)
 set dia 1 %rows(gridt) = ((365.0*(anos-(1.0/12.0* (fix((12.0*anos)+1.0)-1.0) )))+1.0)
 set nfecha 1 %rows(gridt) = %if(%CLOCK(fix(nfecha(t)),4).lt.4.and.mes(t).eq.2.and.dia(t).ge.28,$
```

```
                                %round((fix(anos)+(mes+1)/100+(dia(t+4)-1)/10000),4),$
                                (%if(%CLOCK(fix(nfecha(t)),4).eq.4.and.mes(t).eq.2.and.dia(t).gt.30,$
                                %round((fix(anos)+(mes+1)/100+(dia(t+2)-1)/10000),4),$
                                (%if(mes.eq.2.and.dia.ge.31.or.mes.eq.4.and.dia.ge.31.or.$
                                mes.eq.6.and.dia.ge.31.or.mes.ge.9.and.dia.ge.31.or.$
                                mes.eq.11.and.dia.ge.31,%round((fix(anos)+(mes+1)/100+(dia(t+1)-
1)/10000),4),$
                                %round((fix(anos)+mes/100+dia/10000),4))))))

 if(salida.eq.1) {
  print / anos Tspot Tforward
 }



 sca(hticks=100,header="Función de Descuento",subheader="Metodología B-spline cúbica",vlabel='Tasas
compuestas anuales (%)',hlabel='Vencimiento',style=line) 1
 # anos Descuento
 sca(hticks=100,header="Tasa spot",subheader="Metodología B-spline cúbica",vlabel='Tasas compuestas
anuales (%)',hlabel='Vencimiento',style=line) 1
 # anos Tspot
 sca(hticks=100,header="Tasa forward",subheader="Metodología B-spline cúbica",vlabel='Tasas compuestas
anuales (%)',hlabel='Vencimiento',style=line) 1
 # anos Tforward

end bspline
```