```
(* tuto_536.nb. 21/8/06.
  OHM4 EICL version imported from t535 but done correctly here I hope. Started
  with t517c. Update equations and their order changed and parameters from
  first page of MPS handout. Mirror effect and spacing effect predictions and
  all of Glanzer regularities. Options for calculator version as well. *)
(* Terminology. For bivariate spacing effects we need four Glenberg or G
  matrices for study test or st and continuous task or con and High Frequency or
  HF items and Low Frequency or LF items. Call them GHFst and GHFcon and GLFst
  and GLFcon. We also need the number of rows and columns call them N1st and
  N2st and N1con and N2con. These are the same for HF and LF items. The actual
  spacing intervals are functions of rows and columns so we need slopes and
  intercepts call them stHFslope and stHFint and conHFslope and conHFint. For
  spacing effect computations find qqq for HF items and uuu for LF items. *)
(* For study test N1 is 4 and N2 is 4. Space2slope is 4 and space2int
  is 1 leading to P1 to P2 spacing intervals of 1 and 5 and 11 and 16.
  Space3slope is 25 and space3int is 12 leading to P2 to P3 retention
  intervals of 12 and 37 and 62 and 87. From t517f. For continuous task
  N1 is 3 and N2 is 4. space2slope is 8 and space2 int is 3 leading to
  spacing intervals of 3 and 11 and 19. space3slope is 4 and space3int is 2
  leading to p2 to p3 retention intervals of 2 and 6 and 14. From t517g. *)
(* Timing. For Nsub of 20 and LL of 80 about 50 sec. Nsub of 100 about 10
  min. 70 minutes for Nsub of 500 and LL of 40 so Nrank is perforce 20. Four
  hours for Nsub of 1000 or Nsub of 500 and LL of 80. 40 minutes for Nsub
  of 400 and Nrank of 20. There is a separate lexicon for each of the four
  conditions namely AN and BN and BO and AO but this is nested within subjects
  so same update parameter values for each subject.  This will not scale up
  because of the forced choice which is roughly proportional to N squared. *)
(* Cannnot reorder update equations to natural order
  as in t534 because they are linked. *)
(* 13/8/06.  BOrep and AOrep for extra processing of BO and AO items. Number
  of extra times the update equations were run before forgetting decrement. No
  good even when 10 to 0 split. Has been removed. Used cHF and cLF instead. *)
(* Variance regularities. Just three. HF to LF for new and
  O/N ratios for HF and LF. See ohm_45a p. 17. *)
(* Note differences from t535. 7000 rather than 1000 types,
 Zsf of 10 rather than 6 and LL of 80 rather than 40 and
  LLHF and LLLF of 500 and 3500 rather than 340 and 540. *)
(* We have done BN and BO and must repeat for AN and AO when we figure
  out what is wrong. Find "BBM" below where  insert must go. *)
(* To debug and to explore the parameter space using the check option is not
  too helpful. Set debug to False and also set all "V" to .01 of their means. *)
(* I may have screwed thing up for the spacing effect in changing HF
  to LF. Do we do it twice once at the end of BO and then again at the
  end of AO? Check later after we get the mirror effect if we do. *)
(* Fudge xcr for extra c ratio. This is the ratio by which c is increased
  for LF items. Put it in all update equations for LF items. *)
(* Coefficients of Variation. Do as a package. From t536a. Reset LL to 80
  and Nsub to whatever when debugged. xcr all bonehead. Check each step. Also
  Nrep variability should be proportional to meanrep. Lag or space too? Will
  have to change LL for continuous task. Maybe other things too. *)
```

```
(* kV is an easy way to scale the variance. In particular make kV
  very small to reduce or eliminate parameter variability. *)
(* Present problems. Contentering weak. The LF ROC plot is
  very shakey.The old new HF ROC has the wrong variance tho
  slope pretty close to one. Otherwise quite respectable. *)
(* It has one free parameter which is meanxcr. It increases the value
  of c by some factor currently 2. Vxcr is a fixed parameter. *)

(* Load packages  *)
<< Statistics`ContinuousDistributions`
<< Statistics`DiscreteDistributions`
<< Graphics`MultipleListPlot`

(* Start preliminaries. From EICLpre which came from t534d. *)
(* Print heading *)
name = "tuto_536.nb";
heading = "Program to calculate the predictions for the original version of EICL
    for the mirror effect and the spacing effect but done correctly. Gives
    predictions for spacing effect for study test and continuous task.  ";
start = Date[];
Print[name, "  ", start];
Print[heading];

(* Set debug and check *)
debug = False; check = False;

(* Set SeedRandom
   startingseed=123;
 SeedRandom[startingseed];  *)
Print["debug, check, starting seed: ", {debug, check, startingseed}];

(* Round x to n decimal places for printing *)
round[x_, n_] := N[Round[10^n x] / 10^n, n];

(* Define ntyp and compute ntok.  *)
ntyp = 7000; ntok = 0;
Do[
  ntok += Floor[ntyp / i];
  , {i, 1, ntyp}];

(* Set list length LL and Zipf scale factor Zsf *)
LL = 80; Zsf = 7;
Print["ntyp, ntok, Zsf: ", {ntyp, ntok, Zsf}];

(* Define update equations.  *)
xp[x_, space_] := α^space x + a (1 - α^space x);
yp[y_, space_] := β^space y + b (1 - β^space y);
zp[z_, x_, c_, space_] := (1 - γ^space) z + c (1 - α^space x);


(*  New from t536. Set mean parameter values in array Pmean. *)
```

```
meanα = .99995; meanα = .9997; meanα = .9999; meanβ = .9999;
meanβ = .999997; meanγ = .97; meanγ = .99; meana = .09; meanb = .008; meanc = .2;
Pmean = {meanα, meanβ, meanγ, meana, meanb, meanc};
Print["Pmean: ", Pmean];

(* New from t536a. Set Coefficients of Variation in
  array kPV. Must set kV first. This scales the variability. *)
kV = 2.0; Vα = .00005; Vβ = .00005; Vγ = .1; Vγ = .01; Va = 1; Vb = .1;
Vc = .1; kPV = kV {Vα, Vβ, Vγ, Va, Vb, Vc};
Print["kV, kPV: ", {kV, kPV}];

(* Set xcr parameters for later normal distribution. *)
meanxcr = 2; Vxcr = .01; SDxcr = meanxcr Vxcr;
Print["meanxcr, Vxcr, SDxcr: ", {meanxcr, Vxcr, SDxcr}];

(* Set standard deviations for the forgetting parameters and the LRCs and xcr.
    SDα=meanα Vα;SDβ= meanβ Vβ;SDγ= meanγ Vγ;
 SDa= meana Va;SDb= meanb Vb;SDc=meanc Vc;SDxcr=meanxcr Vxcr;
 Print[" SDα, SDβ, SDγ, SDa, SDb, SDc, SDxcr: ",
  {SDα, SDβ, SDγ, SDa, SDb, SDc, SDxcr}];  *)

(* New from t534a. Set standard deviations in array PSD *)
PSD = {meanα Vα, meanβ Vβ, meanγ Vγ, meana Va, meanb Vb, meanc Vc};
Print["PSD: ", PSD];

(*  New from t534a. Set Normal Distributions *)
NDα = NormalDistribution[Pmean[[1]], PSD[[1]]];
NDβ = NormalDistribution[Pmean[[2]], PSD[[2]]];
NDγ = NormalDistribution[Pmean[[3]], PSD[[3]]];
NDa = NormalDistribution[Pmean[[4]], PSD[[4]]];
NDb = NormalDistribution[Pmean[[5]], PSD[[5]]];
NDc = NormalDistribution[Pmean[[6]], PSD[[6]]];
NDxcr = NormalDistribution[meanxcr, SDxcr];

(* Set Vrep and and Vlag. Vrep determines the
  repetition variability and Vlag determines the space variability
  for the lexicon. They are the same for HF and LF items. *)
Vrep = 1.0; Vlag = 1.0;
Print["Vrep, Vlag: ", { Vrep, Vlag}];

(* Set LLHF and LLLF to use for setting rank. Also
  set Nsub for number of subjects and Nrank. Nsub is the outer
  loop and Nrank under that. Delay is for concentering. *)
LLHF = 500; LLLF = 2500; Nsub = 500; Nrank = LL / 2; delay = 100;

(* Overwrite to check *)
Nsub = If[check, 1, Nsub];
Nrank = If[check, 1, Nrank];
Print["LLHF, LLLF, Nsub, Nrank, delay for concentering: ",
  {LLHF, LLLF, Nsub, Nrank, delay}];
```

```
(* Set BOrep and AOrep for extra processing of AO and BO items. Now out.
    BOrep=0;AOrep=10;
 Print["BOrep, AOrep: ",{BOrep, AOrep}]; *)

(* Set NrepHFcheck and NrepLFcheck and spaceHFcheck and spaceLFcheck for lexicon *)
NrepHFcheck = 60; NrepLFcheck = 19; spaceHFcheck = 702; spaceLFcheck = 2216;
If[check, Print["NrepHFcheck, NrepLFcheck, spaceHFcheck, spaceLFcheck: ",
   {NrepHFcheck, NrepLFcheck, spaceHFcheck, spaceLFcheck}], Null];

(* Initialize BN and AN and BO and AO to save data for mirror effect. *)
AN = {}; BN = BO = AO = AN;

(* Initialize for the spacing effect. First set Number of rows and
  Number of columns for study test or st procedure and continuous task
  or con for High Frequency or HF and Low Frequency or LF items. *)
N1st = 4; N2st = 4; N1con = 3; N2con = 4;

Print["N1st, N2st, N1con, N2con: ", {N1st, N2st, N1con, N2con}];

(* Now initialize the four Glenberg or G matrices. High
  frequency or low frequency and study test or continuous task. *)
GHFst = Table[0, {N1st}, {N2st}];
GHFcon = Table[0, {N1con}, {N2con}];
GLFst = Table[0, {N1st}, {N2st}];
GLFcon = Table[0, {N1con}, {N2con}];

(* We also need slopes and interceptsfor the two word
  frequencies and the two procedures. From t517f and t517g.  *)
space2slopest = 4; space2intst = 1; space3slopest = 25; space3intst = 12;
space2slopecon = 8; space2intcon = 3; space3slopecon = 4; space3intcon = 2;
Print["space2slopest, space2intst, space3slopest, space3intst: ",
  {space2slopest, space2intst, space3slopest, space3intst}];
Print["space2slopecon, space2intcon, space3slopecon, space3intcon: ",
  {space2slopecon, space2intcon, space3slopecon, space3intcon}];

(* Initialize all delays for concentering *)
BNdelay = {}; BOdelay = ANdelay = AOdelay = BNdelay;

(* End of preliminaries *)

(* Loop for subjects with index isub and upper limit Nsub starts here *)
Do[

  (* Set parameter values for this subject. They are all random variables.  *)
  α = 99; While[Not[0 < α < 1.0], α = Random[NDα]];
  β = 99; While[Not[0 < β < 1.0], β = Random[NDβ]];
  γ = 99; While[Not[0 < γ < 1.0], γ = Random[NDγ]];
a = -99; While[Not[0 < a < 1.0], a = Random[NDa]];
b = -99; While[Not[0 < b < 1.0], b = Random[NDb]];
  c = -99; While[Not[0 < c < 1.0], c = Random[NDc]];
  xcr = Random[NDxcr];
```

```
  (* Overwrite to check *)
  α = If[check, meanα, α];
  β = If[check, meanβ, β];
  γ = If[check, meanγ, γ];
  a = If[check, meana, a];
  b = If[check, meanb, b];
  c = If[check, meanc, c];
  xcr = If[check, meanxcr, xcr];

  (* Debug  *)
  If[debug, Print["isub, α, β, γ, a, b, c, xcr: ", {isub, α, β, γ, a, b, c, xcr}], Null];

  (*  BN and BO start here. Loop for rank with index irank and upper limit Nrank. *)
  Do[

   (* BN starts here. Initialize x and y. *)
   x = 0; y = 0;

   (* Set meanrep and Nrep *)
meanrep = Floor[Zsf ntyp / (LLHF + irank)]; sd = Vrep meanrep;
temp = -99;
While[temp < 2, temp = Random[NormalDistribution[meanrep, Vrep meanrep]]];
   Nrep = Floor[temp];

   (* Overwrite to check *)
   yIf[check, NrepHFcheck, Nrep];

   (* Loop for Nrep no index *)
Do[

    (* Set up lag. This will be different for every
      subject and every rank frequency both in HF condition and in the
      LF condition and for every repetition in forming the lexicon.  *)
    meanlag = Zsf ntok / Nrep; SDlag = Vlag meanlag;

    (* Set up Normal Distribution for this lag and every rank frequency. *)
    NDlag = NormalDistribution[meanlag, Vlag meanlag];

    (* Now set up space  *)
temp = -99;
    While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

    (* Overwrite to check *)
    space = If[check, spaceHFcheck, space];

    (* Update  *)
    x = xp[x, space];
    y = yp[y, space];

    If[debug, Print["irank, meanrep, Nrep, meanlag, SDlag, space, x, y: ",
```

```
      round[{irank, meanrep, Nrep, meanlag, SDlag, space, x, y}, 3]], Null];

    (* End loop for Nrep for BN for this rank and subject *)
    , {Nrep}];

  (* Save *)
  BN = Append[BN, x - y];

  If[check, Print["x and y for BN at end of lexicon: ", round[{x, y}, 3]], Null];

  (* Decrement for concentering *)
  xD = x α^delay;
  yD = y α^delay;
  BNdelay = Append[BNdelay, xD];
  BNdelay = Append[BNdelay, yD];

  (* BN ends here. Now we must repeat all of the above and add the study update and
    the study test decrement and insert it here for BO. So BO starts here. *)

  (* Initialize x and y and z *)
  x = 0; y = 0; z = 0;

  (* Set meanrep and Nrep *)
meanrep = Floor[Zsf ntyp / (LLHF + irank)]; sd = Vrep meanrep;
temp = -99;
While[temp < 2, temp = Random[NormalDistribution[meanrep, Vrep meanrep]]];
  Nrep = Floor[temp];

  (* Overwrite to check *)
  If[check, NrepHFcheck, Nrep];

(* Loop for Nrep no index *)
  Do[

    (* Set up lag. This will be different for every
      subject and every rank frequency both in HF condition and in the
      LF condition and for every repetition in forming the lexicon.  *)
    meanlag = Zsf ntok / Nrep; SDlag = Vlag meanlag;

    (* Set up Normal Distribution for this lag and every rank frequency. *)
    NDlag = NormalDistribution[meanlag, Vlag meanlag];

    (* Now set up space  *)
temp = -99;
    While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

    (* Overwrite to check *)
    space = If[check, spaceHFcheck, space];

    (* Update  *)
    x = xp[x, space];
```

```
 y = yp[y, space];

 If[debug, Print["irank, meanrep, Nrep, meanlag, SDlag, space, x, y: ",
   round[{irank, meanrep, Nrep, meanlag, SDlag, space, x, y}, 3]], Null];

 (* End loop for Nrep for BO for this rank and subject *)
 , {Nrep}];

(* Print *)
If[check, Print["BO:"], Null];
If[check, Print["x and y for BO at end of lexicon: ", round[{x, y}, 3]], Null];

(* Now turn new items into old. Learning increment for old HF items *)

(* Now set up space  *)
temp = -99;
While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

(* Overwrite to check *)
space = If[check, spaceHFcheck, space];

(* Update  *)
z = zp[z, x, c, space];
x = xp[x, space];
y = yp[y, space];

If[check, Print["x and y and z for BO at after 2P: ", round[{x, y, z}, 3]], Null];

(* Repeat BOrep times
  Do[
   z=zp[z,x,c,space];
   x=xp[x,space];
   y=yp[y,space];
     *)

(* Print *)
If[check, Print["x and y and z for BO after update: ", round[{x, y, z}, 3]], Null];

(* Strength at test for old HF items *)
space = Random[] LL / 2 + Random[] LL;

(* Set STspacecheck. 11/8/06. *)
STspacecheck = 10;
If[check, Print["STspacecheck: ", STspacecheck], Null];

(* Overwrite to check *)
space = If[check, STspacecheck, space];
x = x α^space;
y = y β^space;
z = z ɣ^space;
```

```
If[check, Print["x and y and z for BO at test: ", round[{x, y, z}, 3]], Null];

(* Save *)
BO = Append[BO, x - y + z];

If[check, Print["x and y and z for BO at test: ", round[{x, y,  z}, 3]], Null];

(* Now delay for concentering. *)
(* Decrement for concentering *)
xD = x α^delay;
yD = y β^delay;
zD = z γ^delay;
BOdelay = Append[BOdelay, xD - yD + zD];

(*  BO ends here. Now spacing effect for HF items for this subject
  and this rank for study test and continuous task procedures.  Set
  up GHFst and then GHFcon for study test and continuous task
  respectively. Study test spacing effect for HF words starts here *)

(* Loop for P1 to P2 interval *)
Do[
 (* Loop for P2 to test interval *)
 Do[
   space2 = space2slopest (i - 1) + space2intst;
   space3 = space3slopest (j - 1) + space3intst;

   (* Optional print. There could be a lot. *)
   If[debug, Print["i, j, space2, space3: ", {i, j, space2, space3}], Null];

   (* Update P1 to P2 *)
   z2 = zp[z, x, c, space2];
   x2 = xp[x, space2];
   y2 = yp[y, space2];


   (* Update P2 to test *)
   x3 = x2 α^space3;
   y3 = y2 β^space3;
   z3 = z2 γ^space3;

   If[debug, Print["x3, y3, z3 for stHF: ", round[{x3, y3, z3}, 3]], Null];

   (* Save in GHFst. *)
   GHFst[[i, j]] = GHFst[[i, j]] + x3 - y3 + z3;

   (* End of loop for space3 *)
   , {j, 1, N2st}];

 (* End of loop for space2 *)
 , {i, 1, N1st}];
```

```
    (* End of spacing effect for study test for HF
      items. Repeat for spacing effect for continuous task for HF items. *)

    (* Loop for P1 to P2 interval *)
    Do[
     (* Loop for P2 to test interval *)
     Do[
       space2 = space2slopecon (i - 1) + space2intcon;
       space3 = space3slopecon (j - 1) + space3intcon;

       (* Optional print. There could be a lot. *)
       If[debug, Print["i, j, space2, space3: ", {i, j, space2, space3}], Null];

       (* Update P1 to P2 *)
       z2 = zp[z, x, c, space2];
       x2 = xp[x, space2];
       y2 = yp[y, space2];

       (* Update P2 to test *)
       x3 = x2 α^space3;
       y3 = y2 β^space3;
       z3 = z2 γ^space3;

       If[debug, Print["x3, y3, z3 for stLF: ", round[{x3, y3, z3}, 3]], Null];

       (* Must save in GHFcon here.  *)
       (* Must save in GHFst here. *)
       GHFcon[[i, j]] = GHFcon[[i, j]] + x3 - y3 + z3;

       (* End of loop for space3 *)
       , {j, 1, N2con}];

     (* End of loop for space2 *)
     , {i, 1, N1con}];

    (* End loop for rank. We now have BN and BO and GHFst
      and GHFcon. BN and BO are lists and GHFst and GHFcon are arrays *)
    , {irank, 1, Nrank}];

   (* Check *)
   If[check, Print["BN: ", BN], Null];
   If[check, Print["BO: ", BO], Null];
If[check, Print["TGHGFst: ", TableForm[round[Transpose[GHFst] / (Nsub LL), 3]]], Null];
If[check, Print["TGHGFcon: ", TableForm[round[Transpose[GHFcon] / (Nsub LL), 3]]], Null];


(* BBM. For LF items we must repeat all of the above and
    change HF to LF wherever it occurs and B to A in BN and BO.  *)

   (*  AN and AO start here. Loop for rank with index irank and upper limit Nrank. *)
   Do[
```

```
   (* AN starts here. Initialize x and y. *)
   x = 0; y = 0;

   (* Set meanrep and Nrep *)
meanrep = Floor[Zsf ntyp / (LLLF + irank)]; sd = Vrep meanrep;
temp = -99;
While[temp < 2, temp = Random[NormalDistribution[meanrep, Vrep meanrep]]];
   Nrep = Floor[temp];

   (* Overwrite to check *)
   yIf[check, NrepLFcheck, Nrep];

   (* Loop for Nrep no index *)
Do[

    (* Set up lag. This will be different for every
      subject and every rank frequency both in HF condition and in the
      LF condition and for every repetition in forming the lexicon.  *)
    meanlag = Zsf ntok / Nrep; SDlag = Vlag meanlag;

    (* Set up Normal Distribution for this lag and every rank frequency. *)
    NDlag = NormalDistribution[meanlag, Vlag meanlag];

    (* Now set up space  *)
temp = -99;
    While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

    (* Overwrite to check *)
    space = If[check, spaceLFcheck, space];

    (* Update  *)
    x = xp[x, space];
    y = yp[y, space];

    If[debug, Print["irank, meanrep, Nrep, meanlag, SDlag, space, x, y: ",
      round[{irank, meanrep, Nrep, meanlag, SDlag, space, x, y}, 3]], Null];

    (* End loop for Nrep for AN for this rank and subject *)
    , {Nrep}];

   (* Save *)
   AN = Append[AN, x - y];

   If[check, Print["x and y for AN at end of lexicon: ", round[{x, y}, 3]], Null];

   (* Decrement for concentering *)
   xD = x α^delay;
   yD = y α^delay;
   ANdelay = Append[ANdelay, xD];
   ANdelay = Append[ANdelay, yD];
```

```
   (* AN ends here. Now we must repeat all of the above and add the study update and
     the study test decrement and insert it here for AO. So AO starts here. *)

   (* Initialize x and y and z *)
   x = 0; y = 0; z = 0;

   (* Set meanrep and Nrep *)
meanrep = Floor[Zsf ntyp / (LLLF + irank)]; sd = Vrep meanrep;
temp = -99;
While[temp < 2, temp = Random[NormalDistribution[meanrep, Vrep meanrep]]];
   Nrep = Floor[temp];

   (* Overwrite to check *)
   If[check, NrepLFcheck, Nrep];

(* Loop for Nrep no index *)
   Do[

    (* Set up lag. This will be different for every
      subject and every rank frequency both in LF condition and in the
      LF condition and for every repetition in forming the lexicon.  *)
    meanlag = Zsf ntok / Nrep; SDlag = Vlag meanlag;

    (* Set up Normal Distribution for this lag and every rank frequency. *)
    NDlag = NormalDistribution[meanlag, Vlag meanlag];

    (* Now set up space  *)
temp = -99;
    While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

    (* Overwrite to check *)
    space = If[check, spaceLFcheck, space];

    (* Update  *)
    x = xp[x, space];
    y = yp[y, space];

    If[debug, Print["irank, meanrep, Nrep, meanlag, SDlag, space, x, y: ",
      round[{irank, meanrep, Nrep, meanlag, SDlag, space, x, y}, 3]], Null];

    (* End loop for Nrep for AO for this rank and subject *)
    , {Nrep}];

   (* Print *)
   If[check, Print["AO:"], Null];
   If[check, Print["x and y for AO at end of lexicon: ", round[{x, y}, 3]], Null];

   (* Now turn new items into old. Learning increment for old LF items *)

   (* Now set up space  *)
```

```
temp = -99;
While[temp < 0, temp = Random[NormalDistribution[meanlag, Vlag]]]; space = Floor[temp];

(* Overwrite to check *)
space = If[check, spaceLFcheck, space];

(* Update  *)
z = zp[z, x, xcr c, space];
x = xp[x, space];
y = yp[y, space];

If[check, Print["x and y and z for AO at after 2P: ", round[{x, y, z}, 3]], Null];

(* Repeat AOrep times
  Do[
   z=zp[z,x,xcr c,space];
   x=xp[x,space];
   y=yp[y,space];
     *)

(* Print *)
If[check, Print["x and y and z for AO after update: ", round[{x, y, z}, 3]], Null];

(* Strength at test for old LF items *)
space = Random[] LL / 2 + Random[] LL;

(* Set STspacecheck. 11/8/06. *)
STspacecheck = 10;
If[check, Print["STspacecheck: ", STspacecheck], Null];

(* Overwrite to check *)
space = If[check, STspacecheck, space];
x = x α^space;
y = y β^space;
z = z γ^space;

If[check, Print["x and y and z for AO at test: ", round[{x, y, z}, 3]], Null];

(* Save *)
AO = Append[AO, x - y + z];

If[check, Print["x and y and z for AO at test: ", round[{x, y, z}, 3]], Null];

(* Now delay for concentering. *)
(* Decrement for concentering *)
xD = x α^delay;
yD = y β^delay;
zD = z γ^delay;
AOdelay = Append[AOdelay, xD - yD + zD];

(*  AO ends here. Now spacing effect for LF items for this subject
```

```
  and this rank for study test and continuous task procedures.  Set
  up GLFst and then GLFcon for study test and continuous task
  respectively. Study test spacing effect for LF words starts here *)

(* Loop for P1 to P2 interval *)
Do[
 (* Loop for P2 to test interval *)
 Do[
   space2 = space2slopest (i - 1) + space2intst;
   space3 = space3slopest (j - 1) + space3intst;

   (* Optional print. There could be a lot. *)
   If[debug, Print["i, j, space2, space3: ", {i, j, space2, space3}], Null];

   (* Update P1 to P2 *)
   z2 = zp[z, x, xcr c, space2];
   x2 = xp[x, space2];
   y2 = yp[y, space2];


   (* Update P2 to test *)
   x3 = x2 α^space3;
   y3 = y2 β^space3;
   z3 = z2 γ^space3;

   If[debug, Print["x3, y3, z3 for stLF: ", round[{x3, y3, z3}, 3]], Null];

   (* Save in GLFst. *)
   GLFst[[i, j]] = GLFst[[i, j]] + x3 - y3 + z3;

   (* End of loop for space3 *)
   , {j, 1, N2st}];

 (* End of loop for space2 *)
 , {i, 1, N1st}];

(* End of spacing effect for study test for LF
  items. Repeat for spacing effect for continuous task for LF items. *)

(* Loop for P1 to P2 interval *)
Do[
 (* Loop for P2 to test interval *)
 Do[
   space2 = space2slopecon (i - 1) + space2intcon;
   space3 = space3slopecon (j - 1) + space3intcon;

   (* Optional print. There could be a lot. *)
   If[debug, Print["i, j, space2, space3: ", {i, j, space2, space3}], Null];

   (* Update P1 to P2 *)
   z2 = zp[z, x, xcr c, space2];
```

```
      x2 = xp[x, space2];
      y2 = yp[y, space2];

      (* Update P2 to test *)
      x3 = x2 α^space3;
      y3 = y2 β^space3;
      z3 = z2 γ^space3;

      If[debug, Print["x3, y3, z3 for stLF: ", round[{x3, y3, z3}, 3]], Null];

      (* Must save in GLFcon here.  *)
      (* Must save in GLFst here. *)
      GLFcon[[i, j]] = GLFcon[[i, j]] + x3 - y3 + z3;

      (* End of loop for space3 *)
      , {j, 1, N2con}];

    (* End of loop for space2 *)
    , {i, 1, N1con}];

   (* End loop for rank. We now have AN and AO and GLFst
     and GLFcon. AN and AO are lists and GHFst and GHFcon are arrays *)
   , {irank, 1, Nrank}];


  (* End of loop for subjects *)
  , {isub, 1, Nsub}];

(* For grid search *)
mirror = {Mean[AN], Mean[BN], Mean[BO], Mean[AO]};
Print["Mirror effect ", round[mirror, 3]];

Print["space 2 and space 3 for st and con for HF need to be checked. If they
    are correct than LF should be OK because they should be the same."];

(* Write out results *)
mirrorMeans = {Mean[AN], Mean [BN], Mean[BO], Mean[AO]};
mirrorSDs = {StandardDeviation[AN],
   StandardDeviation [BN], StandardDeviation[BO], StandardDeviation[AO]};
If[! check, Print["mirrorMeans: ", round[mirrorMeans, 3]], Null];
If[! check, Print["mirrorSDs:    ", round[mirrorSDs, 3]], Null];

(* The three variance regularities. HF to LF for new and O/N ratios for HF and LF. *)
Print["Variance regularities. HF to LF for new and O/N ratios for HF and LF: ",
  round[{StandardDeviation[BN] / StandardDeviation[AN], StandardDeviation[BO] /
     StandardDeviation[BN], StandardDeviation[AO] / StandardDeviation[AN]}, 3]];

(* Get distributions call them fAN and fBN and fBO and
  fAO. From t340 first then t524. First normalize the lists. *)
allmax = Max[{Max[AN], Max[BN], Max[BO], Max[AO]}] + .001;
xAN = AN / allmax; xBN = BN / allmax; xBO = BO / allmax; xAO = AO / allmax;
```

```
If[! check, Print["allmax: ", round[allmax, 3]], Null];

Do[
  xAN[[i]] = If[xAN[[i]] < 0, .001, xAN[[i]]];
  xBN[[i]] = If[xBN[[i]] < 0, .001, xBN[[i]]];
  xAO[[i]] = If[xAO[[i]] < 0, .001, xAO[[i]]];
  xBO[[i]] = If[xBO[[i]] < 0, .001, xBO[[i]]];
  , {i, 1, Nsub Nrank}];

(* Now compute distribution. First fAN. *)
fAN = Table[0, {20}];
Do[k = Floor[20 xAN[[i]]]; fAN[[k + 1]] += 1;, {i, 1, Dimensions[xAN][[1]]}];

(* Now fBN *)
fBN = Table[0, {20}];
Do[k = Floor[20 xBN[[i]]]; fBN[[k + 1]] += 1;, {i, 1, Dimensions[xBN][[1]]}];

(* Now fBO *)
fBO = Table[0, {20}];
Do[k = Floor[20 xBO[[i]]]; fBO[[k + 1]] += 1;, {i, 1, Dimensions[xBO][[1]]}];

(* Now fAO *)
fAO = Table[0, {20}];
Do[k = Floor[20 xAO[[i]]]; fAO[[k + 1]] += 1;, {i, 1, Dimensions[xAO][[1]]}];

(* Plot *)
If[! check, Print["Mirror Distributions"], Null];
If[! check, MultipleListPlot[fAN, fBN, fBO, fAO,
   PlotLegend → {"AN", "BN", "BO", "AO"}, PlotJoined → {True, True, True, True}], Null];

(* ROC curves. From t522d via t524. 28/3/06. ROC slopes for slope
  regularity. Imported from t522d. Dropped the "all" in fAN etc. *)

(* Set up pcumfAN which is the reverse cumulative probability needed
  for ROC curve. But then we need to reverse that for plotting. *)
cumfAN = Table[fAN[[20]], {20}];
cumfBN = Table[fBN[[20]], {20}];
cumfBO = Table[fBO[[20]], {20}];
cumfAO = Table[fAO[[20]], {20}];
Do[
  cumfAN[[i]] = cumfAN[[i + 1]] + fAN[[i]], {i, 19, 1, -1}];
pcumfAN = cumfAN / (Nsub Nrank);
RpcumfAN = Reverse[pcumfAN];

(* Repeat for BN *)
cumfBN = Table[fBN[[20]], {20}];
Do[
  cumfBN[[i]] = cumfBN[[i + 1]] + fBN[[i]], {i, 19, 1, -1}];
pcumfBN = cumfBN / (Nsub Nrank);
RpcumfBN = Reverse[pcumfBN];
```

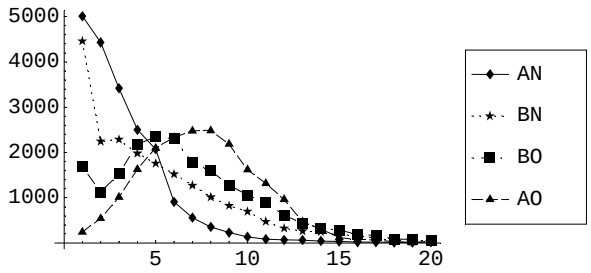
```
(* Repeat for BO *)
Do[
  cumfBO[[i]] = cumfBO[[i + 1]] + fBO[[i]], {i, 19, 1, -1}];
pcumfBO = cumfBO / (Nsub Nrank);
RpcumfBO = Reverse[pcumfBO];

(* Repeat for AO *)
Do[
  cumfAO[[i]] = cumfAO[[i + 1]] + fAO[[i]], {i, 19, 1, -1}];
pcumfAO = cumfAO / (Nsub Nrank);
RpcumfAO = Reverse[pcumfAO];

(* Rename for simplicity *)
T1 = RpcumfAN; T2 = RpcumfBN; T3 = RpcumfBO; T4 = RpcumfAO;

(* Set up ROC1. This is the LF plot. *)
ROC1 = {{T1[[1]], T4[[1]]}, {T1[[2]], T4[[2]]}, {T1[[3]], T4[[3]]},
   {T1[[4]], T4[[4]]}, {T1[[5]], T4[[5]]}, {T1[[6]], T4[[6]]}, {T1[[7]], T4[[7]]},
   {T1[[8]], T4[[8]]}, {T1[[9]], T4[[9]]}, {T1[[10]], T4[[10]]}, {T1[[11]], T4[[11]]},
   {T1[[12]], T4[[12]]}, {T1[[13]], T4[[13]]}, {T1[[14]], T4[[14]]},
   {T1[[15]], T4[[15]]}, {T1[[16]], T4[[16]]}, {T1[[17]], T4[[17]]},
   {T1[[18]], T4[[18]]}, {T1[[19]], T4[[19]]}, {T1[[20]], T4[[20]]}};
Print["LF ROC plot:"]; ListPlot[ROC1, {PlotJoined → True,
  AspectRatio → Automatic, Frame → True}];

(* Set up ROC2 *)
ROC2 = {{T2[[1]], T3[[1]]}, {T2[[2]], T3[[2]]}, {T2[[3]], T3[[3]]},
   {T2[[4]], T3[[4]]}, {T2[[5]], T3[[5]]}, {T2[[6]], T3[[6]]}, {T2[[7]], T3[[7]]},
   {T2[[8]], T3[[8]]}, {T2[[9]], T3[[9]]}, {T2[[10]], T3[[10]]}, {T2[[11]], T3[[11]]},
   {T2[[12]], T3[[12]]}, {T2[[13]], T3[[13]]}, {T2[[14]], T3[[14]]},
   {T2[[15]], T3[[15]]}, {T2[[16]], T3[[16]]}, {T2[[17]], T3[[17]]},
   {T2[[18]], T3[[18]]}, {T2[[19]], T3[[19]]}, {T2[[20]], T3[[20]]}};
Print["HF ROC plot:"];
ListPlot[ROC2, {PlotJoined → True, AspectRatio → Automatic, Frame → True}];

(* Slope regularities.  Superceded by variance regularities. See OHM45a.
SR={StandardDeviation[AO]/StandardDeviation[BN],
   StandardDeviation[AO]/StandardDeviation[AN],
   StandardDeviation[BO]/StandardDeviation[BN],
   StandardDeviation[BO]/StandardDeviation[AN]};
Print["Slope regularities: ",round[SR,3]]; *)

(* Multiple list plots
MultipleListPlot[ROC2,ROC1,{PlotJoined→True,AspectRatio→Automatic,Frame→True},
   PlotLegend→{"ROC2","ROC1"}];
MultipleListPlot[ROC3,ROC2,{PlotJoined→True,AspectRatio→Automatic,Frame→True},
  PlotLegend→{"ROC3","ROC2"}];
MultipleListPlot[ROC4,ROC3,{PlotJoined→True,AspectRatio→Automatic,Frame→True},
  PlotLegend→{"ROC4","ROC3"}]; *)
```

```
(* Write out mean concentering data *)
concenter = {Mean[ANdelay], Mean[BNdelay], Mean[BOdelay], Mean[AOdelay]};
Print["mirrorMeans:  ", round[mirrorMeans, 3]]
Print["Mean contentering: ", round[concenter, 3]];

(*  Simple forced choice routine.  No threshhold. ANBN here. *)
low = AN; high = BN; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
ANBN = sum / (Nlow Nhigh);

(* BOAO here *)
low = BO; high = AO; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
BOAO = sum / (Nlow Nhigh);

(* BNBO here *)
low = BN; high = BO; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
BNBO = sum / (Nlow Nhigh);

(* ANBO here *)
low = AN; high = BO; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
ANBO = sum / (Nlow Nhigh);

(* BNAO here *)
low = BN; high = AO; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
```

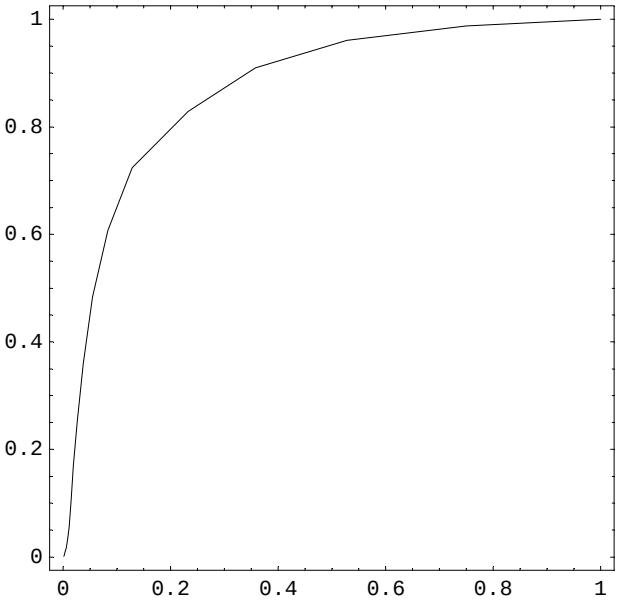
```
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
BNAO = sum / (Nlow Nhigh);

(* ANAO here *)
low = AN; high = AO; Nlow = Dimensions[low][[1]]; Nhigh = Dimensions[high][[1]];
sum = 0;
Do[
  Do[
    sum = If[high[[i]] > low[[j]], sum + 1, sum];
    , {j, 1, Nlow}];
  , {i, 1, Nhigh}];
ANAO = sum / (Nlow Nhigh);

(* Define Forced Choice and print result *)
FC = {ANBN, BOAO, BNBO, ANBO, BNAO, ANAO};
Print["Forced Choice: ", round[FC, 3]];

(* Plot predicted spacing effect for Hockley
  study test for HF and LF items. Lifted from t517f. *)

(* Get Transposes  *)
TGHFst = Transpose[GHFst];
TGLFst = Transpose[GLFst];

(* Bivariate HF and LF plots. The abscissa is the N1 P1 to P2 spacing
  intervals and the parameter is the N2 P2 to test retention intervals. *)
Print["Hockley study-test bivariate plots for HF (top) and LF (bottom) items:"];
MultipleListPlot[TGHFst[[1]], TGHFst[[2]], TGHFst[[3]],
  TGHFst[[4]], PlotJoined → True, PlotLegend → {RI1, RI2, RI3, RI4}];
MultipleListPlot[TGLFst[[1]], TGLFst[[2]], TGLFst[[3]], TGLFst[[4]],
  PlotJoined → True, PlotLegend → {RI1, RI2, RI3, RI4}];

(* Repeat for Oddson continuous task. From t517g. *)

(* Get Transposes  *)
TGHFcon = Transpose[GHFcon];
TGLFcon = Transpose[GLFcon];

(* Bivariate HF and LF plots for continuous task. The abscissa is the N1 P1 to P2
  spacing intervals and the parameter is the N2 P2 to tecon retention intervals. *)
Print["Oddson bivariate plots for continuous task
    for HF (top) and LF (bottom) items:"];
MultipleListPlot[TGHFcon[[1]], TGHFcon[[2]], TGHFcon[[3]], TGHFcon[[4]],
  PlotJoined → True, PlotLegend → {RI1, RI2, RI3, RI4}];
MultipleListPlot[TGLFcon[[1]], TGLFcon[[2]], TGLFcon[[3]],
  TGLFcon[[4]], PlotJoined → True, PlotLegend → {RI1, RI2, RI3, RI4}];

(* Word Frequency Effect for study test procedure *)
WFEst = (TGLFst - TGHFst) / (Nsub Nrank);
Print["WFE for study test procedure: ", TableForm[round[WFEst, 3]]];
```

```
(* Word Frequency Effect for continuous task procedure *)
WFEcon = (TGLFcon - TGHFcon) / (Nsub Nrank);
Print["WFE for continuous task procedure: ", TableForm[round[WFEcon, 3]]];

(* Time *)
end = Date[];
Print["Start: ", start];
Print["End:   ", end];
```

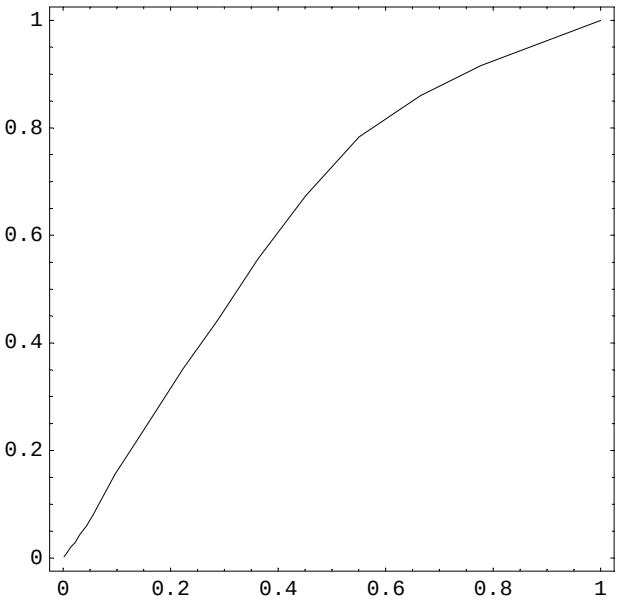
```
tuto_536.nb  {2006, 8, 21, 18, 7, 7}

Program to calculate the predictions for the original version
  of EICL for the mirror effect and the spacing effect but done correctly.
  Gives predictions for spacing effect for study test and continuous task.

debug, check, starting seed: {False, False, startingseed}

ntyp, ntok, Zsf: {7000, 63071, 7}

Pmean: {0.9999, 0.999997, 0.99, 0.09, 0.008, 0.2}

kV, kPV: {2., {0.0001, 0.0001, 0.02, 2., 0.2, 0.2}}

meanxcr, Vxcr, SDxcr: {2, 0.01, 0.02}

PSD: {0.000049995, 0.0000499999, 0.0099, 0.09, 0.0008, 0.02}

Vrep, Vlag: {1., 1.}

LLHF, LLLF, Nsub, Nrank, delay for concentering: {500, 2500, 500, 40, 100}

N1st, N2st, N1con, N2con: {4, 4, 3, 4}

space2slopest, space2intst, space3slopest, space3intst: {4, 1, 25, 12}

space2slopecon, space2intcon, space3slopecon, space3intcon: {8, 3, 4, 2}

Mirror effect {0.128, 0.198, 0.285, 0.335}

space 2 and space 3 for st and con for HF need to be checked.
  If they are correct than LF should be OK because they should be the same.

mirrorMeans: {0.128, 0.198, 0.285, 0.335}

mirrorSDs:    {0.122, 0.208, 0.196, 0.148}

Variance regularities. HF to LF for new and O/N ratios for HF and LF: {1.702, 0.945, 1.21}

allmax: 0.955

Mirror Distributions
```



LF ROC plot:



HF ROC plot:



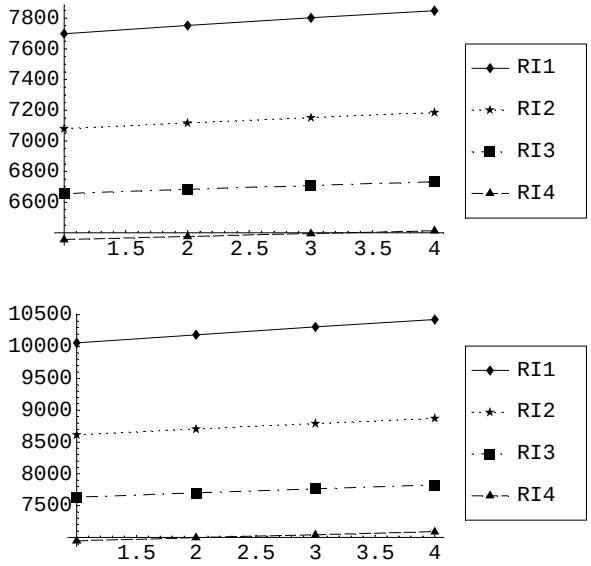
mirrorMeans: {0.128, 0.198, 0.285, 0.335}

Mean contentering: {0.089, 0.188, 0.236, 0.223}

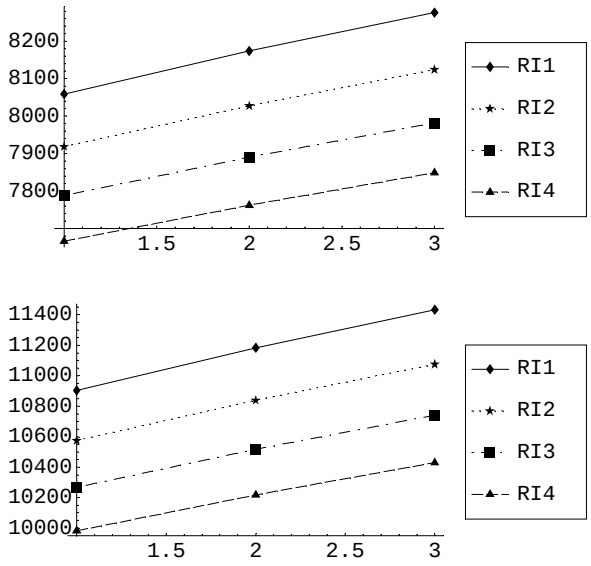
Internal counter overflow.
Some expression evaluations may not run to their
final fixed points. Try redoing your computation
on a 64-bit enhanced version of Mathematica.

Forced Choice: {0.61, 0.596, 0.638, 0.776, 0.727, 0.876}

Hockley study–test bivariate plots for HF (top) and LF (bottom) items:



Oddson bivariate plots for continuous task for HF (top) and LF (bottom) items:



WFE for study test procedure:

0.118	0.122	0.125	0.129
0.077	0.079	0.082	0.084
0.049	0.051	0.053	0.055
0.029	0.031	0.032	0.034

```
                                  0.142     0.15      0.158
WFE for continuous task procedure: 0.133     0.141     0.148
                                  0.124     0.131     0.138
                                  0.116     0.123     0.129

Start: {2006, 8, 21, 18, 7, 7}

End:   {2006, 8, 21, 22, 17, 47}
```