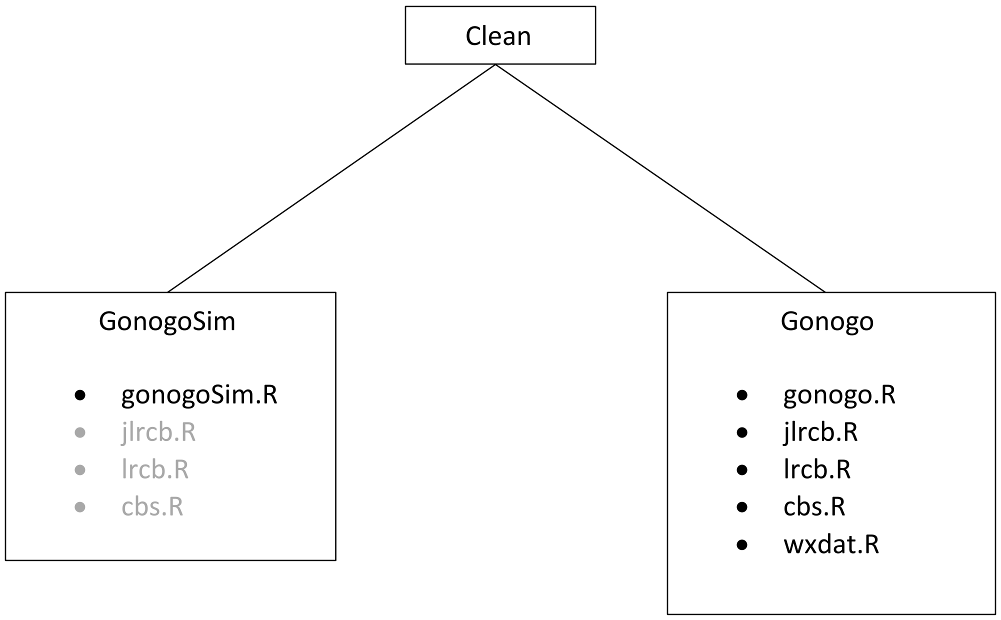
1. Suggest making 2 folders in a new (i.e., clean) folder in your R workspace. One will be for console use, the other for simulation use. I call the console version folder "Gonogo" and the simulation folder "GonogoSim".

2. Drop gonogo.R, jlrcb.R, lrcb.R, cbs.R and wxdat.R into the console folder.
   Drop gonogoSim.R, jlrcb.R, lrcb.R and cbs.R into the simulation folder.



3. My six "originals" appear in their locations in black. The three scripts in grey indicate they are copies of the originals.

4. Open R (I use 64 bit), navigate to each folder, and run all of the .R files using source("filename.R").

5. Sensitivity tests are conducted in the console folder.
   A 3pod test is begun by:
   w=gonogo(mlo,mhi,sg)
   A Neyer test is begun by:
   w=gonogo(mlo,mhi,sg,neyer=T)

   The full call to "gonogo" is: gonogo=function(mlo=0,mhi=0,sg=0,newz=T,reso=0,ln=F,neyer=F).
   The "ln=T" quietly conducts the test in log(X) units whereas testing is done in X units. The ln=T option is experimental, especially if you continue on with a Phase III. The starting values for tau[1] and beta need to be looked at in more detail.
   A "reso=.125" option, e.g., recommends test levels rounded to the nearest one eighth.

6. A word about the "title" entry: make it short & sweet. No need to say it's a 3pod or a Neyer test. No need to say it's done in the log scale either. This info, captured in the call, is automatically PREPENDED to the hopefully succinct title you provide.

7. A test is gracefully suspended by entering: an (X,Y) pair with Y=-1; a negative sample size (n2 or n3); or an invalid (p, λ) pair (i.e., $p < 0$, $p > 1$ or $\lambda < 0$).

CAUTION: when asked for 2 entries, make two entries, otherwise the test ends UNGRACEFULLY. Tests suspended gracefully are saved in the list "w" and may be resumed by:

z=w; w=gonogo(newz=F)

The above is actually 2 commands, separated by the semi colon.
NOTE: the variable named "z" has a special meaning for the gonogo script. With the newz=F option, it looks for a list, saved as "z", in the immediate environment. So, reserve z for that use.
TIP: In R, the "up arrow" allows you to scroll through and repeat previous commands. This feature also allows you to edit a similar line ("left arrow", "right arrow) without too much typing.

8. A successfully suspended (or completed) test is saved in a named list. If the list is named w, then w contains all pertinent information about the test.

9. Eight plots are included. A history plot of X's and Y's is produced at (practically) every console entry or upon completion of a simulation (with plt=1 option). The history plot can be generated from the list associated with the test, as can the remaining 7 plots. The command formats are v=ptest(w,i) (console version) or v=pStest(w,i) (simulation version), for various i. Defining v in this way captures any tabular output generated by the call. Details are provided as follows:

i	Graphic	Syntax	Alternate Syntax
1	History plot	ptest(w,1) or pStest(w,1)	pdat1(w) or pSdat1(w)
2	MLE's of mu and sigma	ptest(w,2) or pStest(w,2)	pdat2(w) or pSdat2(w)
3	Response curve, with data Pooled Adjacent violators solution 95% 1-Sided GLM ***confidence bounds***	ptest(w,3) or pStest(w,3)	pdat3(w) or pSdat3(w)
4	A simple visual of the data	ptest(w,4) or pStest(w,4)	picdat(w)
5	Joint likelihood ratio (LR) multi-***confidence bounds***	ptest(w,5) or pStest(w,5)	jlrcb(w)
6	Joint & Individual LR multi-***confidence bounds***	ptest(w,6) or pStest(w,6)	lrcb(w)
* 7	Joint and/or Individual LR ***confidence bound***	ptest(w,7) or pStest(w,7)	cbs(w,1)
8	***Confidence bounds*** on Probability (p) and Quantile (q) computed via 3 methods: Likelihood Ratio (LR), Fisher Matrix (FM) and General Linear Model (GLM)	ptest(w,8) or pStest(w,8)	cbs(w,2)

* The LR tools (graphics and tabulations) focus on ***confidence bounds*** as featured in Neyer's SenTest™ program. ***Confidence bounds*** computed via other more commonly used methods (FM & GLM) are also included in cbs(w,2)**.** LR ***confidence bounds*** graphed with cbs(w,1) are restricted to data sets having interval overlap and conf1 < c1max. Less restrictive (including point overlap, no overlap, conf1 > c1max) LR ***confidence bounds*** may be graphed with jlrcb(w) & lrcb(w).
An aside: Critical decisions are often based upon ***Confidence bounds.*** Interestingly, Appendix G of MIL-STD-331C [7] ignores the subject. Guidance on decision making based on ***confidence bounds*** seems needed. Some have a view that ***confidence bounds*** in this setting cannot be taken very seriously. Employing ***confidence bounds*** to choose the test sequence, e.g., in a D-optimal test, is an example of a useful application.

The following table gives additional detail about these graph-producing functions in terms of calls to ptest(w,i) (or pStest(w,i)), for i= 5, 6,7, or 8:

A hodgepodge of graphs and tabulations provided by:					**ptest(w,i) or pStest(w,i)**			
input	method	joint conf region	overlap	output	i=5 jlrcb(w)	i=6 lrcb(w)	i=7 cbs1(w,1)	i=8 cbs2(w,2)
conf1								X
conf1 or conf2					X	X	X	
conf1's or conf2's					X	X		
p 0 or 0 q						X	X	
P 0 or 0 q or 0 0							X	
	LR	bounded	Y		X	X	X*	X*
			N		X	X		
		unbounded	Y		X	X	X*	X*
			N		X	X		
	FM		Y					X*
	GLM		Y					X*
				joint LRCB(s)	X	X	X	
				individual LRCB's		X	X	
				(x,qnorm(p)), x=q, qlo & qhi		X		X
				(q,x), x=q, qlo & qhi		X		X
				(q,qnorm(x)), x=p, plo & phi		X		X
				100*(p,x), x=p, plo & phi		X		X
				list returned	X			
				text file		X	X	X

* interval overlap only

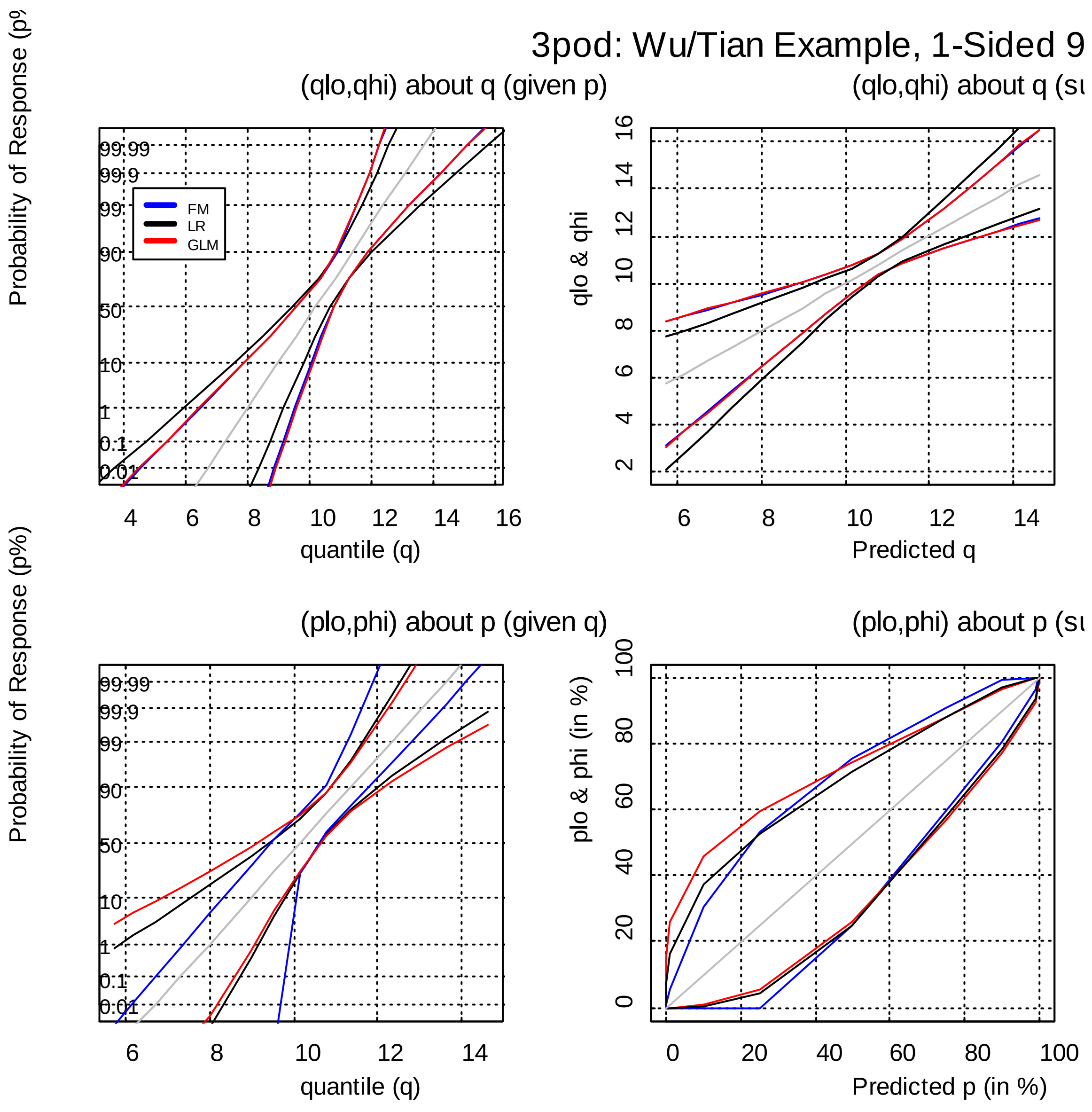
10. A few words about ptest(w,8): It asks for a conf1 entry, and gives 4 plots of the 3 different confidence intervals (FM,LR and GLM).

    NOTE The conf1 entry applies to the 2-sided situation. The plot produced depicts TWO 1-sided confidence limits each having (1+conf1)/2 confidence. So, if you're interested in 1-sided limits only, say at 90% confidence, then you would want to enter .80 at the prompt for conf1.

    Besides a plot, ptest(w,8) also creates a text file cbs.txt in your R working directory. What you may want do with cbs.txt (outside of R) is: copy its contents, paste it into a word document, and highlight what you just pasted. Next, select the Table tab; select Table; and select Convert Text to Table.

    This will produce a table that will yield: vertical limits (pl,pu) about p (for various p's) and horizontal limits (ql,qu) about q (for various q's). The first 15 lines are FM limits; the next 15 lines are LR limits; the last 15 lines are GLM limits.

    The results of plotting the Wu/Tian example (Table 1 of [1]) and processing the cbs.txt as described above (after emboldening a few borders, centering the headings, and right justifying the entries) are depicted below:



ql	q	qh	pl	p	pu
2.36084	5.73071	9.10058	0	0.000001	0.000019
3.10264	6.186992	9.271344	0	0.00001	0.000158
3.930343	6.696829	9.463315	0	0.0001	0.001273
4.881624	7.284103	9.686581	0	0.001	0.009661
6.032461	7.997557	9.962654	0	0.01	0.066076
7.589243	8.973379	10.357515	0	0.1	0.360084
8.473017	9.540364	10.607712	0	0.25	0.613152
9.407776	10.170326	10.932876	0.174284	0.5	0.825716
10.213007	10.800288	11.387569	0.550185	0.75	0.949815
10.72964	11.367273	12.004906	0.780187	0.9	1
11.28389	12.343095	13.402299	0.959775	0.99	1
11.596206	13.05655	14.516893	0.993735	0.999	1
11.834502	13.643823	15.453144	0.999133	0.9999	1
12.034566	14.15366	16.272754	0.999888	0.99999	1
12.210354	14.609942	17.00953	0.999986	0.999999	1

0.472863	5.73071	8.080301	0	0.000001	0.026286
1.390702	6.186992	8.336611	0	0.00001	0.045457
2.413505	6.696829	8.624101	0	0.0001	0.078944
3.589674	7.284103	8.957453	0	0.001	0.138181
5.012541	7.997557	9.367295	0	0.01	0.2456
6.934724	8.973379	9.945905	0.001106	0.1	0.454013
8.022127	9.540364	10.304244	0.022298	0.25	0.597941
9.159269	10.170326	10.755014	0.185137	0.5	0.762818
10.120303	10.800288	11.387803	0.523806	0.75	0.90666
10.765501	11.367273	12.256497	0.744514	0.9	0.981139
11.522703	12.343095	14.066164	0.909089	0.99	0.999921
11.965315	13.05655	15.462946	0.960622	0.999	1
12.311225	13.643823	16.628065	0.981742	0.9999	1
12.6053	14.15366	17.64554	0.991219	0.99999	1
12.865596	14.609942	18.558826	0.995677	0.999999	1
2.247471	5.73071	9.213949	0	0.000001	0.152923
2.998869	6.186992	9.375115	0	0.00001	0.197268
3.837256	6.696829	9.556402	0	0.0001	0.255488
4.800771	7.284103	9.767434	0	0.001	0.3331
5.966306	7.997557	10.028809	0.000003	0.01	0.439783
7.5426	8.973379	10.404158	0.002451	0.1	0.598846
8.437002	9.540364	10.643727	0.031738	0.25	0.693874
9.38197	10.170326	10.958682	0.199312	0.5	0.800688
10.193057	10.800288	11.407519	0.509708	0.75	0.907355
10.708022	11.367273	12.026524	0.717592	0.9	0.976561
11.248176	12.343095	13.438013	0.875757	0.99	0.999766
11.547035	13.05655	14.566064	0.929762	0.999	0.999999
11.773613	13.643823	15.514033	0.956975	0.9999	1
11.96327	14.15366	16.344051	0.972551	0.99999	1
12.129632	14.609942	17.090252	0.982038	0.999999	1

Here are a few examples of how the table may be used (with GLM entries):

Pr[Y=1 response | X = 10.8] ~ .75
A lower 90% confidence limit on the above probability is .509.
A lower 90% confidence limit of the 1 in a 1000 point is 4.800771

11. Of course, after running a viable test one may want certain statistics. At the moment there's no one function that summarizes everything one might want. For now, here's how to use the function ***nyqrda*** to retrieve some basic information about a test w and the ML fit that was obtained:
In your R session, execute u=nyqrda(w$d0,ln=w$d0). Then,

u$a is a (intercept)	(- 10.8892 for the Wu/Tian example)
u$b is b (slope)	(1.070684 for the Wu/Tian example)
u$mu is mu	(10.17033 for the Wu/Tian example)
u$sig is sig	(.9339827 for the Wu/Tian example)

Additionally, u$xglm is also returned, and R's ***predict*** could be used to compute vertical confidence limits, and ***summary***(u$xglm) will produce the following type of output (shown below for the Wu/Tian example):

```
Call:
glm(formula = y ~ x, family = binomial(link = probit), maxit = maxitt,
    epsilon = eps)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.96144  -0.00057   0.21608   0.39464   1.53622

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.8892     4.5469  -2.395  0.01663 *
x             1.0707     0.4138   2.587  0.00967 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.795  on 29  degrees of freedom
Residual deviance: 14.139  on 28  degrees of freedom
AIC: 18.139

Number of Fisher Scoring iterations: 7
```

12. 3pod and Neyer have different Phase I's but their Phase II's are the same. A typical Neyer ends in Phase II. An option to continue in Phase III is provided. A 3pod flow chart is included in Appendix 1 (taken from [1]). Gonogo utilizes a system of identification using a Phase, Stage, Step notation defined in [1]. A Neyer test flow chart using Block numbers to describe Phase I computations is included in Appendix 2 (taken from [5]).

13. Modifications to the original 3pod are incorporated, specifically in

    Steps (i) and (ii) of Phase I's Stage 1 (I1(i) and I1(ii)) ;
    Phase III RMJ, which is now a special case of a Skewed RMJ Phase III (with $\lambda = 1$);
    Like Phases I and II, Phase III, is now scale-free (verify with gonogoSim with different M's)

14. Phase III, described in {1], [2] and [3], is now a skewed RMJ procedure calling for two parameters, p and λ. For p near 0, a large λ is recommended. This forces an upward bias whereby bigger upward (than downward) steps are taken, increasing the chance of y=1 response. For p near 1, a small λ is recommended. This forces a downward bias whereby bigger downward steps are taken, increasing the chance of y=0 response. Regarding choosing λ given p: "Our simulations do not tell us exactly what value of λ should be chosen. The optimal choice of λ is a complicated function of various factors, and theory does not help here. Simulations do suggest that a wide range of λ values give good performance. To find a good λ value requires some tweaking on the part of the experimenter". [3] The simulation program gonogoSim is useful to visually see how phase III unfolds for various choices of p and λ.

15. An error correction feature is provided with a handy function: fixw. Suppose an error was inadvertently entered at the nth previous console read. The syntax to go back, fix and continue is:

    z=fixw(w,n); w=gonogo(newz=F)

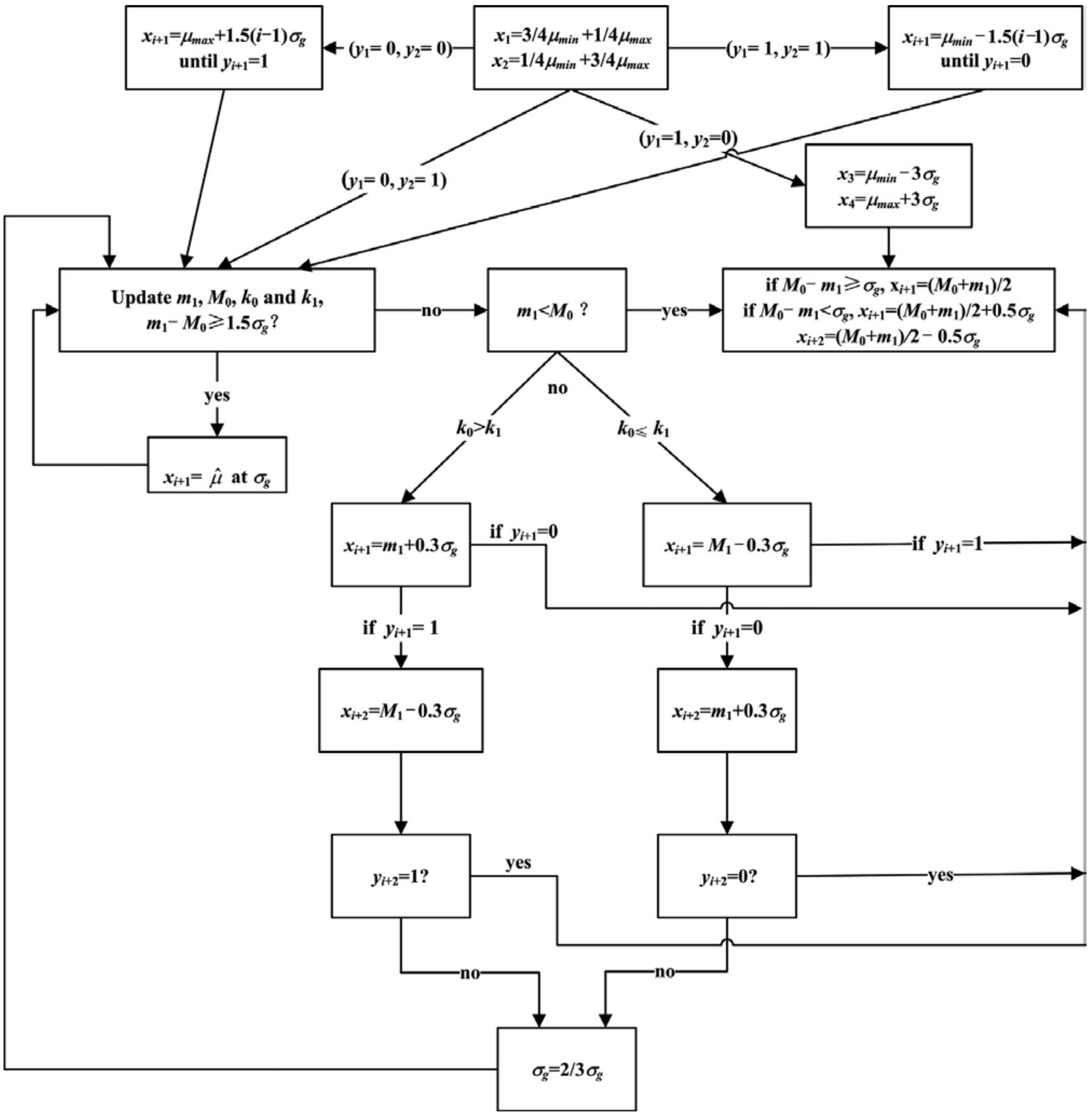
16. Twenty five sample sensitivity tests (as lists) are provided. Access to them is gained by: wx=wxdat(i) (for I between 1 and 25). The various wx's are useful as arguments to ptest. Some of the trivial data sets (very small n, single point overlap, no overlap) provide much insight into the behavior of the joint LR confidence bounds. Plots obtained by ptest(wx,4) and ptest(wx,5) are included in **jlrbc.docx**.

17. The function lrmax(w) returns several important characteristics of the test data, including: the overlap category "one23" (1 for interval overlap; 2 for point overlap; and 3 for no overlap); c1max and c2max, which indicate whether the LR joint confidence regions are bounded (conf1 < c1max, or conf2 < c2max); and con = # of responses/total tested (which determines the axis of the joint LR confidence region). conf1 and conf2 are related by: c1max=pchisq(qchisq(c2max,2),1).

18. The programs dropped into the "simulation" folder perform much the same way as their console counterparts. The list w is gotten in one fell swoop of the simulation program. For example, w=gonogoSim(u,mlo,mhi,sg,n2,n3,p,lam) produces the list w and its history plot. It could be reproduced also by: pStest(w,1). The full call to gonogoSim is:
gonogoSim(u,mlo,mhi,sg,n2,n3,p,lam,dm=0,ds=0,ln=F,plt=0,neyer=F,iseed=-1,IIgo=T,M=1)
M: the simulation ALWAYS multiplies mlo, mhi, sg, dm and ds by M. The default is M=1. This option, along with a fixed iseed, allows you to demonstrate that the procedures are indeed scale-free.
dm is the deviation from the true mean (mu_true), which is taken to be (mlo+mhi)/2.
ds is the deviation from the true standard deviation (sigma_true), which is taken to be sg.
plt: plt=0 for no plot
iseed: An iseed unequal to -1 allows one to repeatedly generate the same sequence of X's & Y's.
IIgo: IIgo = T means we're continuing into Phase II (IIgo = F would be useful to compare 3pod and Neyer Phase I). 3pod's Phase I sample size (n1) was further broken down into two parts (n11 and n12). n11, being the appropriate sample size to compare to Neyer's n1, is the stopping point for the IIgo=F option for the 3pod case.
NOTE: For gonogoSim, the reso option was disabled, i.e., reso is set to zero right at startup. However, titles and many function calls retain reference to reso as if it was still an option (done this way to facilitate adding the option back in at some future time).

19. There are eighty functions involved in this R code. A summary of the calls for each is included in Appendix 3.

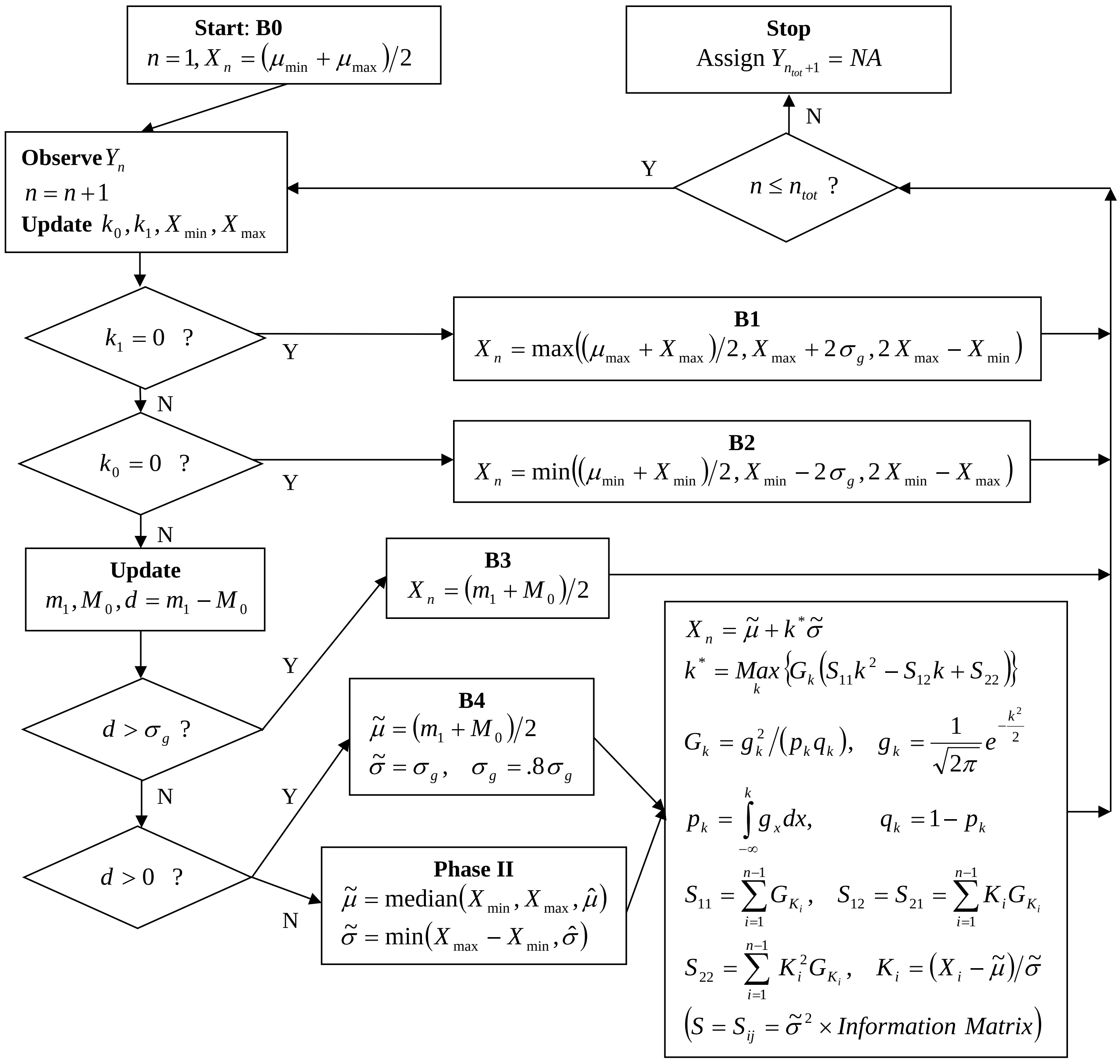
Appendix 1



$x_{i+1}=\mu_{max}+1.5(i-1)\sigma_g$ until $y_{i+1}=1$
$(y_1=0, y_2=0)$
$x_1=3/4\mu_{min}+1/4\mu_{max}$
$x_2=1/4\mu_{min}+3/4\mu_{max}$
$(y_1=1, y_2=1)$
$x_{i+1}=\mu_{min}-1.5(i-1)\sigma_g$ until $y_{i+1}=0$
$(y_1=1, y_2=0)$
$(y_1=0, y_2=1)$
$x_3=\mu_{min}-3\sigma_g$
$x_4=\mu_{max}+3\sigma_g$
Update m_1, M_0, k_0 and k_1, $m_1-M_0\geqslant 1.5\sigma_g$?
no
$m_1<M_0$?
yes
if $M_0-m_1\geqslant\sigma_g$, $x_{i+1}=(M_0+m_1)/2$
if $M_0-m_1<\sigma_g$, $x_{i+1}=(M_0+m_1)/2+0.5\sigma_g$
$x_{i+2}=(M_0+m_1)/2-0.5\sigma_g$
yes
$x_{i+1}=\hat{\mu}$ at σ_g
no
$k_0>k_1$
$k_0\leqslant k_1$
$x_{i+1}=m_1+0.3\sigma_g$
if $y_{i+1}=0$
$x_{i+1}=M_1-0.3\sigma_g$
if $y_{i+1}=1$
if $y_{i+1}=1$
if $y_{i+1}=0$
$x_{i+2}=M_1-0.3\sigma_g$
$x_{i+2}=m_1+0.3\sigma_g$
$y_{i+2}=1?$
yes
$y_{i+2}=0?$
yes
no
no
$\sigma_g=2/3\sigma_g$

3pod Phase I flow chart

Appendix 2


Start: **B0**
$n=1, X_n=(\mu_{\min}+\mu_{\max})/2$
Stop
Assign $Y_{n_{tot}+1}=NA$
N
Y
$n \le n_{tot}$?
Observe Y_n
$n=n+1$
Update $k_0, k_1, X_{\min}, X_{\max}$
$k_1=0$?
Y
B1
$X_n=\max((\mu_{\max}+X_{\max})/2, X_{\max}+2\sigma_g, 2X_{\max}-X_{\min})$
N
$k_0=0$?
Y
B2
$X_n=\min((\mu_{\min}+X_{\min})/2, X_{\min}-2\sigma_g, 2X_{\min}-X_{\max})$
N
Update
$m_1, M_0, d=m_1-M_0$
B3
$X_n=(m_1+M_0)/2$
Y
$d>\sigma_g$?
N
B4
$\tilde{\mu}=(m_1+M_0)/2$
$\tilde{\sigma}=\sigma_g, \quad \sigma_g=.8\sigma_g$
Y
$d>0$?
N
Phase II
$\tilde{\mu}=\text{median}(X_{\min}, X_{\max}, \hat{\mu})$
$\tilde{\sigma}=\min(X_{\max}-X_{\min}, \hat{\sigma})$
$X_n=\tilde{\mu}+k^*\tilde{\sigma}$
$k^*=\underset{k}{Max}\{G_k(S_{11}k^2-S_{12}k+S_{22})\}$
$G_k=g_k^2/(p_k q_k), \quad g_k=\frac{1}{\sqrt{2\pi}}e^{-\frac{k^2}{2}}$
$p_k=\int_{-\infty}^{k} g_x dx, \qquad q_k=1-p_k$
$S_{11}=\sum_{i=1}^{n-1} G_{K_i}, \quad S_{12}=S_{21}=\sum_{i=1}^{n-1} K_i G_{K_i}$
$S_{22}=\sum_{i=1}^{n-1} K_i^2 G_{K_i}, \quad K_i=(X_i-\tilde{\mu})/\tilde{\sigma}$
$(S=S_{ij}=\tilde{\sigma}^2\times Information\ Matrix)$

Neyer flow chart (with "gonogo" Phase I Block identifiers)

Appendix 3

i	gonogo	gonogoSim	lrcb	jlrcb	cbs	wxdat	functions
1			X				clim0(rx,ry,m,s,levb)
2			X				clim(rx,ry,m,s,uu,levb)
3			X				abllik(data,mu,sig)
4			X				lrcb(dat)
5			X	X			xyllik(rx,ry,m,s)
6				X			stopQuietly(...)
7				X			calcblim(bl,ul)
8				X			mkb0(confv,nam)
9				X			unbd(rx,ry,levs,mh1,mh2,es,mlim)
10				X			uliknext(rx,ry,levs0,em1,es,em2)
11				X			ulik(rx,ry,levs0,em,es)
12				X			otherpoint(rx,ry,muhat,levs0,con)
13				X			jlik(rx,ry,levs0,ms,op,one23)
14				X			**jlrcb(dat)**
15				X		X	**picdat(dat)**
16				X			simp(dat)
17	X						**gonogo(mlo=0,mhi=0,sg=0,newz=T,reso=0,ln=F,neyer=F)**
18	X						nphaseI(dat0,mlo,mhi,sg,reso,about,titl,unit,ln)
19	X						phaseI1(dat0,mlo,mhi,sg,reso,about,titl,unit,ln)
20	X						phaseI2(d0,dat0,sg,reso,about,titl,unit,ln)
21	X						phaseI3(d0,dat0,sg,reso,about,titl,unit,ln)
22	X						getd0(xx,d0,dat0,ID,reso,about,titl,unit,ln,cab=F)
23	X						getxr(x,nd0,reso,ln)
24	X						prd0(z)
25	X	X					n.update(dat)
26	X	X			X		m.update(dat)
27	X						phaseII(d0,dat0,n2,reso,about,titl,unit,ln)
28	X	X					skewL(c1,nu,tau2,p,be)
29	X						sphaseIII(d0,dat0,n3,p,reso,about,titl,unit,ln,lam=0)
30	X	X			X		glmmle(mydata, response = 1, maxitt = 10., eps = 1e-006, lgit = F)
31	X	X			X		llik(mydata, mu, sig, response = 1)
32	X	X			X		tauf(x,y,response=1)
33	X	X			X		yinfomat(dat, mu, sig)
34	X	X					ykpm(dat, m, es, iplot = 0)
35	X	X					ydeldet(kn, b)
36	X	X					**nyqrda**(dat, lgit = F, ln = F, xmin = -9999., xmax = 9999, conf = 0.95, small = F, response = 0., labx = "", laby = "PROBABILITY OF RESPONSE", maxitt = 10., eps = 1e-006, zee = 0)
37	X	X					pavdf(data.df, ln, plotit = F, response = 0, lineit = F, labx = "STIMULUS", laby = "PROBABILITY OF RESPONSE", titl = "PAV SOLUTION")
38	X						chabout(about,s47,loc)
39	X						wabout(vv)
40	X						blrb3()

41	X						blrb2()
42	X						about4(x)
43	X						**fixw(w,k=1)**
44	X						fixw1(w)
45	X	X					f38(x,l)
46	X	X					f3point8(l)
47	X	X					fgs(mlo,mhi,sg)
48	X	X					ifg(fg0,fg1)
49	X	X					addneyr(dtt,ylm,sim=F)
50	X	X					add3pod(dtt,ylm,sim=F)
51	X						**ptest(dat,plt)**
52	X						pdat1(dat)
53	X						pdat2(dat)
54	X						pdat3(dat)
55	X	X		X	X		reset()
56					X		dose.p(obj,p)
57					X		mixed(dat)
58					X		graf1(limx,t1,k,big,legnd)
59			X				grafl(limx)
60					X		ml.lims(dat,conf,maxitt=10,response=1,rd=6)
61					X		sp.lims(dat,conf,maxitt=10,response=1,rd=6)
62					X		lr.lims(dat,conf1)
63					X	X	**lrmax(w,plt=F)**
64					X		**cbs(w,plt,maxitt=10,response=1)**
65		X					**gonogoSim(mlo,mhi,sg,n2,n3,p,lam,dm=0,ds=0,ln=F,plt=0,neyer=F,iseed=-1,IIgo=T,M=1)**
66		X					pI1(mlo,mhi,sg,tmu,tsig,reso,ln,iseed,dat0=data.frame(numeric(0)))
67		X					pI2(d0,dat0,sg,tmu,tsig,reso,ln,iseed)
68		X					pI3(d0,dat0,sg,tmu,tsig,reso,ln,iseed)
69		X					npI(mlo,mhi,sg,tmu,tsig,reso,ln,iseed,dat0=data.frame(numeric(0)))
70		X					pII(d0,dat0,tmu,tsig,n2,reso,ln,iseed)
71		X					spIIIsim(d0,dat0,tmu,tsig,n3,p,lam,reso,ln,iseed)
72		X					gd0(xx,d0,dat0,ID,mu,sig,reso,ln,iseed=-1)
73		X					gxr(x,mu,sig,reso,ln,iset)
74		X					ntau(dat,response=1)
75		X					wabout13(M,cmlo,cmhi,csg,p,n11,n12,n2,n3,lam,reso)
76		X					**pStest(dat)**
77		X					pSdat1(dat)
78		X					pSdat2(dat)
79		X					pSdat3(dat)
80						X	wxdat(i)

Table of function calls

References

1. Wu, C. F. J, Tian, Y., 2014, Three-phase optimal design of sensitivity experiments Journal of Statistical Planning and Inference 149, 1-15

2. Wang, D. P., Tian, Y. B., Wu, C. F. J., 2015, Comprehensive comparisons of major procedures for sensitivity testing,(Preprint)

3. Wang, D. P., Tian, Y., Wu, C. F. Jeff, 2015, A Skewed Version of the Robbins-Munro-Joseph Procedure for Binary Response, Statistica Sinica, 1679-1689

4. Neyer, Barry T., 1994, A D-Optimality-Based Sensitivity Test, Technometrics, 36, 61-70

5. Ray, D. M., Roediger, P. A., and Neyer, B. T., 2014, Commentary: Three-phase optimal design of sensitivity experiments, Journal of Statistical Planning and Inference, 149, 20-25

6. http://neyersoftware.com/SensitivityTest/SensitivityTestFlyer.htm

7. MIL-STD-331C, Appendix G, Statistical Methods to Determine the Initiation Probability of One-Shot Devices

8. Joseph, V. R., Efficient Robbins-Monro Procedure for Binary Data, Biometrika, 91, 461-470