EFSA GMO Panel opinion on

Statistical considerations for the safety evaluation of GMOs

Presentation of the full statistical code used for the example

The EFSA GMO Panel opinion on *Statistical considerations for the safety evaluation of GMOs (http://www.efsa.europa.eu/en/scdocs/scdoc/1250.htm)* includes a worked-out example (section 5) to illustrate how to implement the statistical procedure described in the opinion itself in the frame of the comparative assessment. In the opinion the EFSA GMO Panel presents Genstat and SAS program fragments to give the essential central mixed model calculation. The opinion does not provide the full code needed to implement the whole procedure from beginning to end.

Many stakeholders have expressed a wish to have the full code of the program used to develop the example, in order to gain a comprehensive insight into the statistical details and steps needed to implement the recommended procedure.

In order to provide the best possible scientific support, EFSA is now publishing the full statistical code used for the example presented in the EFSA GMO Panel opinion. This code is provided in both Genstat and SAS language with the clear understanding that this code is only applicable to the specific example presented in the opinion and it is not of general applicability to other cases. The code may be freely adapted for other specific needs. The Genstat and SAS codes produce tabular results, in addition the Genstat code produces the graphical output as shown in the opinion.

Genstat 12 code used in the example presented in the EFSA GMO Panel opinion on *Statistical considerations for the safety evaluation of GMOs*"

Genstat Release 12 code for the example reported in:

1. EFSA (2010). Statistical considerations for the safety evaluation of GMOs, EFSA Journal 8(1): 1250.
2. van der Voet, H., Perry, J.N., Amzal, B. and Paoletti, C. (submitted). A statistical assessment of differences and equivalences between genetically modified and reference plant varieties.

Note:

EFSA wishes to point out that the seds and the Kenward-Roger degrees of freedom are found slightly different between the two programs (Genstat and SAS). This of course influences the derived calculations, but, in practical terms, there is very little difference. In this example there is one change in the classification of analytes: using SAS Tryptophan is classified Type 1 rather than Type 2. The lower confidence limit for the difference test of Tryptophan is 1.0002 with Genstat (see Table 3 in Opinion, which also gives the 44.0 df, an sed of 0.02851 and an lsd of 0.04791), and with SAS this is 0.999987 (with 44.1 df, an sed of 0.02862, and an lsd of 0.04816).

Data are expected in Example.xls. First worksheet should contain columns analyte, site, rep, genotype, y.
Genotypes 1 and 2 are interpreted as conventional counterpart and GMO, respectively, others are refs.
Nondetects (measurements reported as 'less than RL') should be entered in y as negative values -RL

```
"Define analytes for detailed output"
text [val='Niacin','16:0 Palmitic'] showa
text [val='none'] showa

"Set output format"
output [print=*; width=120] 1

"Read data from Excel file"
import 'Example.xls'

"Define factors for site, replication and genotype"
group [rede=y] site,rep,genotype

"Define three-level factor genotypegroup"
fact [lev=3; lab=!T(comp,gmo,ref)] genotypegroup
calc genotypegroup=genotype*(genotype.in.!(1,2))+3*(genotype>=3)

"Define indicator variable for references"
calc indref=(genotypegroup==3)

"Define contrast conventional counterpart to GMO and refs"
fact [labels=!T(comp,GMOref)] comp_aside
calc comp_aside=1*(genotypegroup==1)+2*(genotypegroup.in.!(2,3))

"Define contrast conventional counterpart and GMO to refs"
fact [labels=!T(compGMO,ref)] ref_aside
calc ref_aside=1*(genotypegroup.in.!(1,2))+2*(genotypegroup==3)

"Define factor for analytes in dataset"
group [rede=y] analyte; lab=analall

"Set up structures for results"
calc na=nval(analall)
vari [val=#na(0)] analkeep
vari [na] seddif,sedeq,sedteq,dfdif,dfeq,dfteq,vare1,vare2,varg,varge,var0,\
meancon,meangmo,meanref,lsddif,lsdeq,lsdteq
text [na] case
calc na2=2*na
calc nsite=nlev(site)
calc nsite1=nsite+1
calc nsite2=2*nsite
vari [na2] column[1...nsite2]
```

```
"Loop for all analytes"
for a=#analall; i=1...na

"Print analyte name and create label"
prin [squa=y] a
prin [ch=case$[i]; ipri=*; squa=y] a; fiel=1; deci=0

"Restrict dataset to data of the current analyte"
restrict y,genotype,genotypegroup,comp_aside,ref_aside,indref,site,rep; analyte.in.a

"Set nondetects (coded in dataset as negative values, -limit) to 0.5*limit"
calc nlessthan=sum(y<=0)
 if nlessthan>0
prin [squa=y; orie=a] nlessthan; deci=0
calc y=y*(y>0)-0.5*y*(y<=0)
endif

"Log-transform data"
calc y=log(y)

"Insert missing values for outliers (identified previously)"
 if a.in.'16:1 Palmitoleic'
   calc y=mvinsert(y; genotypegroup==3.and. y<-6)
 elsif a.in.'17:0 Heptadecanoic'
   calc y=mvinsert(y; genotypegroup==3.and. y>-6)
 elsif a.in.'18:0 Stearic'
   calc y=mvinsert(y; genotypegroup==1.and. y>-2.8)
 elsif a.in.'18:2 Linoleic'
   calc y=mvinsert(y; genotypegroup==1.and. y>0.7)
 elsif a.in.'Copper'
   calc y=mvinsert(y; genotypegroup==1.and. y>0.4)
 elsif a.in.'Ferulic Acid'
   calc y=mvinsert(y; genotypegroup==1.and. y<7.1)
 elsif a.in.'Phytic Acid'
   calc y=mvinsert(y; genotypegroup==1.and. y<-2)
 endif

"Exclude cases with zero variance"
 if var(y)==0
   prin [squa=y; ipri=*] '     zero variance'
 else
   calc analkeep$[i]=1

"Mixed model analysis for differences ----------------------------------------------------------------"
vcomp [fixed=ref_aside/genotypegroup; cadjust=none]\
site + site.rep + indref.genotype; constraint=pos
reml [prin=*] y
if a.in.showa  "show detailed output "
prin 'Mixed model for differences ',a
vdisplay  [prin=#,means; pse=alldiff]
endif

"Keep means and variance components"
vkeep ref_aside.genotypegroup; means=means
vari [val=#means] means1
calc (meancon,meangmo,meanref)$[i]=means1$[1,2,6]
vkeep [sigma2=v0] site + site.rep + indref.genotype; ve1,ve2,vg
scal vge; 0
calc (vare1,vare2,varg,varge,var0)$[i]=ve1,ve2,vg,vge,v0
```

```
"Keep sed, lsd, df"
vkeep ref_aside.genotypegroup; sedmeans=sed; ddf=df
calc seddif$[i]=sed$[2;1]
calc lsddif$[i]=edt(0.95;df)*sed$[2;1]
calc dfdif$[i]=df

"Mixed model including interaction gmocomp with site ----------------------------------------------"
vcomp [fixed=ref_aside/genotypegroup +\
ref_aside.site + ref_aside.genotypegroup.site; cadjust=none]\
site + site.rep + indref.genotype ; constraint=pos
 reml [prin=*] y
if a.in.showa  "show detailed output"
prin 'Mixed model for differences (with GxE interaction) ',a
vdisplay [prin=#,mean; pterms=ref_aside.genotypegroup.site; pse=allest]
endif

"Keep site means and standard errors"
vkeep ref_aside.genotypegroup.site; means=ge; varmeans=covge
matr [8;3] mge
equa ge; mge
calc ii=2*i-1
calc ii1=ii+1
calc column[1...nsite]$[ii]=mge$[1,3,5,7;1]
calc column[1...nsite]$[ii1]=mge$[1,3,5,7;2]
calc column[nsite1...nsite2]$[ii]=sqrt(covge$[1,7,13,19;1,7,13,19])
calc column[nsite1...nsite2]$[ii1]=sqrt(covge$[2,8,14,20;2,8,14,20])

"Mixed model analysis for setting equivalence limits ----------------------------------------------"
vcomp [fixed=comp_aside/genotypegroup]\
site + site.rep + genotype ; constraint=pos
reml [prin=*] y
if a.in.showa  "show detailed output "
prin 'Mixed model for setting equivalence limits ',a
vdisplay  [prin=#,means; pse=alldiff]
endif

"Keep sed, lsd, df"
vkeep comp_aside.genotypegroup; sedmeans=sed; ddf=df
calc sedeq$[i]=sed$[6;5]
calc lsdeq$[i]=edt(0.975;df)*sed$[6;5]
calc dfeq$[i]=df

"Mixed model analysis for testing (non-)equivalence ----------------------------------------------"
vcomp [fixed=comp_aside/genotypegroup; cadjust=none]\
site + site.rep + indref.genotype  ; constraint=pos
reml [prin=*] y
if a.in.showa  "show detailed output "
prin 'Mixed model for equivalence tests ',a
vdisplay  [prin=#,means; pse=alldiff]
endif

"Keep sed, lsd, df"
vkeep comp_aside.genotypegroup; sedmeans=sed; ddf=df
calc sedteq$[i]=sed$[6;5]
calc lsdteq$[i]=edt(0.95;df)*sed$[6;5]
calc dfteq$[i]=df
endif

"Unrestrict dataset"
rest y,genotype,genotypegroup,comp_aside,ref_aside,indref,site,rep

endfor "End of loop through analytes"
```

"Tables"

"Omit results for analytes with zero variance"
```
duplicate analall; anal
subset [analkeep==1] anal,case,seddif,sedeq,sedteq,dfdif,dfeq,dfteq,\
vare1,vare2,varg,varge,var0,meancon,meangmo,meanref,lsddif,lsdeq,lsdteq
calc nanal=nval(anal)
prin nanal
```

"Geometric means (Opinion 2010, Table 1)"
```
calc gmcomp,gmgmo,gmref=exp(meancon,meangmo,meanref)
prin anal,gmcomp,gmgmo,gmref; deci=3
```

"Variance components (Opinion 2010, Table 2)"
```
prin varg,vare1,vare2,var0,dfdif,dfteq,dfeq,case; fiel=8
```

"Difference assessment (Opinion 2010, Table 3)"
```
calc dgc=meangmo-meancon
calc dlow,dupp=dgc+(-1,1)*lsddif
calc ratgc,loratgc,upratgc=exp(dgc,dlow,dupp)
prin ratgc,loratgc,upratgc,seddif,dfdif,lsddif,case; fiel=8
```

"Equivalence assessment at scale of ratio to comparator (Opinion 2010, Table 4)"
```
calc dgr=meangmo-meanref
calc eqlow,equpp=dgr+ (-1,+1)*lsdeq
calc ratgr,lorat,uprat=exp(dgr,eqlow,equpp)
prin ratgr,lorat,uprat,sedeq,dfeq,lsdeq,case; fiel=8
```

"Equivalence assessment on scale GMO to comparator (Opinion 2010, Table 5)"
```
calc fac=lsddif/lsdteq
calc eqlowadj,equppadj=meangmo-meancon + (meanref-meangmo + (-1,+1)*lsdeq) * fac
calc bteqlowadj,btequppadj=exp(eqlowadj,equppadj)
prin sedteq,dfteq,lsdteq,fac,bteqlowadj,btequppadj,case; fiel=8
```

"GM and GSEM per site (Opinion 2010, Table 6 and 7)"
```
vari [val=2(#analkeep)] analkeep2
subset [analkeep2==1] column[]
text [val=2(#anal)] anal2
text [val=(comp,GMO)#nanal] vari2
calc gm[1...4],gsem[1...4]=exp(column[])
prin gm[],gsem[]; fiel=8(9); deci=4
```

"Generate difference/equivalence classes"

"criterion for doing no equivalence test"
```
scal tol; 1e-6
calc noequivtest=varg<=tol    "variance component genotype very small or zero"
calc sigdif=(dlow>0).or.(dupp<0)            "significant difference"
calc equiv=dlow>=eqlowadj.and.dupp<=equppadj      "unambiguously equivalent"
calc warning=(dgc<eqlowadj).or.(dgc>equppadj) "point estimate outside equivalence limits"
calc nonequiv=dlow<eqlowadj.or.dupp>equppadj      "unambiguously non-equivalent"
calc out1=.not.sigdif .and. equiv
calc out2=sigdif .and. equiv
calc out3=.not.sigdif .and. .not.equiv .and. .not.noequivtest .and. .not.warning
calc out4=sigdif .and. .not.equiv .and. .not.noequivtest .and. .not.warning
calc out5=.not.sigdif .and. .not.noequivtest .and. warning
calc out6=sigdif .and. .not.noequivtest .and. warning .and. .not.nonequiv
calc out7=sigdif .and. .not.noequivtest .and. warning .and. nonequiv
calc out8= .not.sigdif .and. .not.equiv .and. noequivtest
calc out9=sigdif .and. .not.equiv .and. noequivtest
```

"Number of expected signif. differences under equivalence hypothesis "

```
calc nsigdif=sum(sigdif)
scal niter; 1000
grandom [normal; niter] difeq[1...nanal]
calc difeq[]=sqrt(2*#varg)*difeq[]
calc pdif[1...nanal]=2*CUT(abs((difeq[])/#seddif);#dfdif)
vari [nanal] psig,meannsig
calc psig$[1...nanal]=mean(pdif[])
calc sig[1...nanal]=pdif[]<0.10
calc meannsig$[1...nanal]=mean(sig[])
calc nsig=vsum(sig)
hist nsig
calc mnsig=mean(nsig)
calc nsiglow,nsigupp=percentile(nsig;2.5,97.5)
prin nsigdif,mnsig,nsiglow,nsigupp
```

"Graphs"

"Plots emphasizing tests of difference and equivalence (Opinion 2010, Figure 3 and 4)"

```
"optional: adapt classification for minimum thresholds"
if 0
calc tu=log(5/4)
calc tl=-tu
prin tl,tu
calc eqlow=eqlow*(eqlow<=tl)+tl*(eqlow>tl)
calc equpp=equpp*(equpp>=tu)+tu*(equpp<tu)
else
scal tl,tu; 0
endif
```

"Initialize graphics"

```
text title; 'Comparative analysis'
conc [case1] case,' '
conc [case2] case,'   <<< Sig. dif.'
conc [case3] case,'   <<< Equiv. more likely than not'
conc [case4] case,'   <<< Sig. dif. and equiv. more likely than not'
conc [case5] case,'   <<< Non-equiv. more likely than not'
conc [case6] case,'   <<< Sig. dif. and non-equiv. more likely than not'
conc [case7] case,'   <<< Sig. dif. and sig. non-equiv.'
conc [case8] case,'   <<< Vg=0 (no conclusion on equiv.)'
conc [case9] case,'   <<< Sig. dif. and Vg=0 (no conclusion on equiv.)'
calc na=nval(case)
text [case; val=#na(' ')] casei
vari [val=1...na] index
calc index=na-index+1
calc hival=max(equpp)+0.1*(max(equpp)-min(eqlow))
vari [val=#na(hival)] label
calc indexm1=min(index)-1
calc indexp1=max(index)+1
vari [val=indexm1,#index,indexp1] index1
calc zero=index1-index1
calc llow=zero+tl
calc uupp=zero+tu
calc indexe=index-0.2
scal npergraph; 27
calc ngraph=int(na/npergraph-0.00001)+1
calc ngraph1=ngraph-1
prin na,ngraph; deci=0
frame 1; xmupp=0.5;xmlow=0.01;xlow=0;xupp=1.0;ylow=0;yupp=1
axes 1; ylab=*; xtit='Ratio to comparator'; styl=x
```

```
xaxis 1;transform=log10
calc label,dgc,eqlow,equpp,zero,llow,uupp,dlow,dupp,eqlowadj,equppadj=\
  exp(label,dgc,eqlow,equpp,zero,llow,uupp,dlow,dupp,eqlowadj,equppadj)
calc xlow=min(!(#label,#dgc,#eqlow,#equpp,#eqlowadj,#equppadj,#zero,#llow,#uupp,#dlow,#dupp))
calc xupp=max(!(#label,#dgc,#eqlow,#equpp,#eqlowadj,#equppadj,#zero,#llow,#uupp,#dlow,#dupp ))+0.1
axes 1; xlow=xlow; xupp=xupp
pen 5; meth=line; lines=1; col='black'; symb=0
scal s; 0.7
scal l; 1.25

"Loop over plots"
for j=ngraph,ngraph1...1
calc ilow,iupp=(j-1)*npergraph+1,npergraph
calc ylow,yupp=ilow,iupp+(-1,1)
axes 1; ylow=ylow; yupp=yupp; ymarks=!(-1,99999)
text [val=clear] kc

"type 1"
calc cond=index>=ilow.and.index<=iupp.and.out1
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case1
text col; 'green'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=1
pen 4; symb=0; label=case1; col=col; size=1; thick=1
  pen 6; col=col; size=1; thick=1
calc igr=ngraph-j+1
if ngraph>1
calc igr=ngraph-j+1
prin [ch=titlei; ipri=*; squa=y] title,' (',igr,')'; deci=0; fiel=1; skip=0
else
prin [ch=titlei; ipri=*; squa=y] title
endif
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 2"
calc cond=index>=ilow.and.index<=iupp.and.out2
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case2
text col; 'blue'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=2
pen 4; symb=0; label=case2; col=col; size=1; thick=2
pen 6; col=col; size=1; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 3"
calc cond=index>=ilow.and.index<=iupp.and.out3
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case3
text col; 'black'
```

```
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=2
pen 4; symb=0; label=case3; col=col; size=1; thick=2
pen 6; col=col; size=1; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 4"
calc cond=index>=ilow.and.index<=iupp.and.out4
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case4
text col; 'black'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=2
pen 4; symb=0; label=case4; col=col; size=1; thick=2
pen 6; col=col; size=1; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 5"
calc cond=index>=ilow.and.index<=iupp.and.out5
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case5
text col; 'red'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,1,1; thick=2
pen 4; symb=0; label=case5; col=col; size=1; thick=2
pen 6; col=col; size=l; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 6"
calc cond=index>=ilow.and.index<=iupp.and.out6
if sum(cond)>0
rest index,indexe,casei; cond
conc casei,case6
text col; 'red'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,1,1; thick=2
pen 4; symb=0; label=case6; col=col; size=1; thick=2
pen 6; col=col; size=l; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 7"
calc cond=index>=ilow.and.index<=iupp.and.out7
if sum(cond)>0
rest index,indexe,casei; cond
```

```
conc casei,case7
text col; 'red'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,1,1; thick=2
pen 4; symb=0; label=case7; col=col; size=1; thick=2
pen 6; col=col; size=l; thick=2
dgraph [win=1; key=0; title=titlei; screen=#kc]\
(index)4,(index1)3; label,dgc,eqlowadj,equppadj,zero,llow,uupp; pen=4,1,2,3,3(5);\
xlower=*,dlow,2(*),3(*); xupper=*,dupp,2(*),3(*); xbarpen=6
rest index,indexe,casei
text [val=keep] kc
endif

"type 8"
calc cond=index>=ilow.and.index<=iupp.and.out8
if sum(cond)>0
rest index,casei; cond
conc casei,case8
text col; 'black'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=2
pen 4; symb=0; label=case8; col=col; size=1; thick=2
pen 6; col=col; size=1; thick=2
dgraph [win=1; key=0; screen=#kc]\
(index)2,(index1)3; label,dgc,zero,llow,uupp; pen=4,1,3(5);\
xlower=*,dlow,3(*); xupper=*,dupp,3(*); xbarpen=6
rest index,casei
text [val=keep] kc
endif

"type 9"
calc cond=index>=ilow.and.index<=iupp.and.out9
if sum(cond)>0
rest index,casei; cond
conc casei,case9
text col; 'black'
pen 1,2,3; meth=point; symb= 2,18,18; col=col; cfill=col; size=1,s,s; thick=2
pen 4; symb=0; label=case9; col=col; size=1; thick=2
pen 6; col=col; size=1; thick=2
dgraph [win=1; key=0; screen=#kc]\
(index)2,(index1)3; label,dgc,zero,llow,uupp; pen=4,1,3(5);\
xlower=*,dlow,3(*); xupper=*,dupp,3(*); xbarpen=6
rest index,casei
text [val=keep] kc
endif
endfor "summary plots"

"Boxplots (Opinion 2010, Figures 5a, 5b, 6 and 7)"

"Backtransform data"
calc eqlow,equpp=meanref + (-1,+1)*lsdeq
calc y,eqlow,meanref,equpp=exp(y,eqlow,meanref,equpp)

"Initialize graphics"
frame 1; xmupper=*
axes 1; styl=xy; xlow=*; xupp=*; xtit=*; ylow=*; yupp=*; ymarks=*; ylab=*
xaxis 1; transform=iden
pen 1...4; symb=1; lab=*; col='black'; cfill='black'
pen 5; symb=0; meth=line; line=1; col='black'; thick=4
pen -10...-1; size=0.8
pen 10; symb=0; line=0
vari [val=0.7,1.3] ybar
calc outbox11= out2 .and. !(1...nanal)<=27   "sig dif"
calc outbox12= out2 .and. !(1...nanal)>27   "sig dif"
```

```
calc outbox2= out3.or.out4  "equiv. more likely than not"
calc outbox3= out5.or.out6.or.out7.or.out8.or.out9    "non equiv more likely than not + Vg==0 "

"Loop boxplots"
for outbox=outbox11,outbox12,outbox2,outbox3
if sum(outbox)>0
subset [outbox]\
anal,casei,eqlow,equpp,meanref; analshow,caseis,lows,upps,meanrefs
prin 'Reference GM and equivalence limits on natural scale:'
prin lows,meanrefs,upps,analshow
calc nshow=nval(analshow)
calc nrow=int((nshow+1)/2)
fframe [row=nrow; col=2; rskip=0.03; ymlow=0.03; ymupp=0.02; xmlow=0.08] ngraph; sscreen=kcp

"Loop over selected analytes"
for a=#analshow; c=#caseis; yl=#lows; yu=#upps; mr=#meanrefs; i=1...nshow
subset [(analyte.in.a) .and. (y-y==0)] y,genotypegroup; y1,genotypegroup1

"Blank plot including all data (to guarantee enough space)"
calc nn=nobs(y1)
vari [val=#y1,yl,yu] data1
axes i; ymark=!(1...3); ylab=!T(ref,gmo,comp); style=box
dgraph [wind=i; screen=#kcp[i]; key=0; title=c] !(#nn(2),0.5,3.5); data1; pen=10

"Three boxplots per window"
boxplot [wind=i; screen=k;  title=c; orie=across; meth=schematic; bar%=40]\
y1; group=genotypegroup1; unitlab=*

"Add lines for equivalence limits"
vari [val=2(yl)] ll
vari [val=2(yu)] uu
vari [val=2(mr)] mm
dgraph [wind=i; screen=keep; key=0] 3(ybar);  ll,uu,mm; pen=5
dele [rede=y] y1,yl,yu,data1,type1,genotypegroup1
endfor "selected analyte in boxplot"
dele [rede=y] analshow,caseis,lows,upps,meanrefs
endif
endfor  "boxplots"

stop
```

SAS 9.1 code used in the example presented in the EFSA GMO Panel opinion on *Statistical considerations for the safety evaluation of GMOs*

```
* ------------------------------------------------------------------------------------------------
SAS 9.1 code for the example reported in:
1. EFSA (2010). Statistical considerations for the safety evaluation of GMOs, EFSA Journal 8(1): 1250.
2. van der Voet, H., Perry, J.N., Amzal, B. and Paoletti, C. (submitted). A statistical assessment of differences and equivalences between
   genetically modified and reference plant varieties.
 ------------------------------------------------------------------------------------------------
Note:
EFSA wishes to point out that the seds and the Kenward-Roger degrees of freedom are found slightly different between the two programs
(GenStat and SAS). This of course influences the derived calculations, but, in practical terms, there is very little difference. In this example
there is one change in the classification of analytes: using SAS Tryptophan is classified Type 1 rather than Type 2. The lower confidence
limit for the difference test of Tryptophan is 1.0002 with Genstat (see Table 3 in Opinion, which also gives the 44.0 df, an sed of
0.02851 and an lsd of 0.04791), and with SAS this is 0.999987 (with 44.1 df, an sed of 0.02862, and an lsd of 0.04816).
------------------------------------------------------------------------------------------------
Data are expected in Example.xls. First worksheet should contain columns analyte, site, rep, genotype, y.
Genotypes 1 and 2 are interpreted as conventional counterpart and GMO, respectively, others are refs.
Nondetects (measurements reported as 'less than RL') should be entered in y as negative values -RL
 ------------------------------------------------------------------------------------------------
;
*** read example data from Excel file;
proc import out=example
        datafile="D:\Data\GMO\Opinion\SAS\example.xls"
        dbms=Excel replace;
    getnames=yes; mixed=no; scantext=yes; usedate=yes; scantime=yes;
run;

*** sort data;
proc sort data=example;
by analyte genotype site rep;
run;

*** preprocess data;
data example1;
set example;
* define three-level factor genotypegroup;
if genotype eq 1 then genotypegroup='comp';
if genotype eq 2 then genotypegroup='gmo';
if genotype ge 3 then genotypegroup='ref';
* define indicator variable indref;
if genotypegroup eq 'ref' then indref=1;
                    else indref=0;
* replace nondetects by 0.5 times the reporting limit;
if y<0 then y= -y/2;
* log transform;
y=log(y);
* outliers;
if analyte='16:1 Palmitoleic' and genotypegroup='ref' and y<-6 then delete;
if analyte='17:0 Heptadecanoic' and genotypegroup='ref' and y>-6 then delete;
if analyte='18:0 Stearic' and genotypegroup='comp' and y>-2.8 then delete;
if analyte='18:2 Linoleic' and genotypegroup='comp' and y>0.7 then delete;
if analyte='Copper' and genotypegroup='comp' and y>0.4 then delete;
if analyte='Ferulic Acid' and genotypegroup='comp' and y<7.1 then delete;
if analyte='Phytic Acid' and genotypegroup='comp' and y<-2 then delete;
run;

*** output settings;
ods html close;
run;
```

***** Mixed model analysis for differences**;

```
proc mixed data=example1 CL=WALD;
by analyte;
class site rep genotype genotypegroup;
model y =  genotypegroup /s covb outp=out  ddfm=kenwardroger;
random site site*rep indref*genotype;
estimate 'gmo_comp' genotypegroup -1 1 0 / CL alpha=0.1;
lsmeans genotypegroup;
ods output lsmeans=lsmeans_g estimates=estdif covparms=covparms;
run;
* Geometric means;
data lsmeans_g2;
set lsmeans_g;
gm = exp(estimate);
keep analyte genotypegroup gm;
proc print data=lsmeans_g2;
run;
* Assessment of differences;
data estdif1;
set estdif;
ratiod=exp(estimate); lowd=exp(lower);  uppd=exp(upper);
dgc=estimate; sed=stderr; dfd=df;
lsddif=sed*tinv(0.95,dfd);
keep analyte dgc ratiod lowd uppd sed dfd lsddif;
proc print data=estdif1;
run;
```

***** Mixed model for setting equivalence limits**;

```
proc mixed data=example1 CL=WALD;
by analyte;
class site rep genotype genotypegroup;
model y =  genotypegroup /s covb outp=out  ddfm=kenwardroger;
random site site*rep genotype;
estimate 'gmo_ref' genotypegroup 0 1 -1 / CL alpha=0.05;
ods output estimates=esteq covparms=covparms;
run;
* keep estimated variance componetns for genotype;
data varg;
set covparms(where=(covparm='genotype'));
varg=estimate;
keep analyte varg;
run;
* 95% equivalence limits ratio GMO/ref;
data esteq1;
merge esteq varg;
ratioe=exp(estimate); lowe=exp(lower); uppe=exp(upper);
dgr=estimate; see=stderr; dfe=df;
lsdeq=see*tinv(0.975,dfe);
by analyte;
keep analyte varg dgr ratioe lowe uppe see dfe lsdeq;
proc print data=esteq1;
run;
```

***** Mixed model analysis for testing (non-)equivalence**;

```
proc mixed data=example1 CL=WALD;
by analyte;
class site rep genotype genotypegroup;
model y =  genotypegroup /s covb outp=out  ddfm=kenwardroger;
random site site*rep indref*genotype;
estimate 'gmo_ref' genotypegroup 0 1 -1 / CL alpha=0.1;
ods output estimates=estteq;
run;
```

```
* Assessment of equivalences (adjusted scale);
data estteq1;
merge estdif1 esteq1 estteq;
lsdteq=stderr*tinv(0.95,df);
fac=lsddif/lsdteq;
lower=-dgr-lsdeq; upper=-dgr+lsdeq;
adjlow=dgc+fac*lower; adjupp=dgc+fac*upper;
adjlow=exp(adjlow); adjupp=exp(adjupp);
by analyte;
keep analyte dgc stderr df lsdteq fac adjlow lowd uppd adjupp varg;
proc print data=estteq1;
run;

*** Classification of results;
data classification;
set estteq1;
tol=1e-6;
noequivtest=(varg<=tol);
sigdif=((lowd>1) or (uppd<1));
equiv=((lowd>=adjlow) and (uppd<=adjupp));
warning=((exp(dgc)<adjlow) or (exp(dgc)>adjupp));
nonequiv=((uppd<adjlow) or (lowd>adjupp));
if ((not sigdif) and equiv) then type=1;
if (    sigdif  and equiv) then type=2;
if ((not sigdif) and (not equiv) and (not noequivtest) and (not warning)) then type=3;
if (    sigdif  and (not equiv) and (not noequivtest) and (not warning)) then type=4;
if ((not sigdif) and (not equiv) and (not noequivtest) and warning) then type=5;
if (    sigdif  and (not equiv) and (not noequivtest) and warning and (not nonequiv)) then type=6;
if (    sigdif  and             (not noequivtest) and warning and nonequiv) then type=7;
if (                            noequivtest) then type=8;
keep analyte noequivtest sigdif equiv warning nonequiv type;
proc sort data=classification;
by type analyte;
proc print data=classification;
run;

*** Mixed model including GxE interaction;
proc mixed data=example1 CL=WALD;
by analyte;
class site rep genotype genotypegroup;
model y =  genotypegroup genotypegroup*site /s covb outp=out  ddfm=kenwardroger;
random site site*rep indref*genotype;
lsmeans genotypegroup*site;
ods output lsmeans=lsmeans_gs;
run;
* Geometric means per site;
data lsmeans_gs2;
set lsmeans_gs;
 gm = exp(estimate);
keep analyte genotypegroup site gm;
proc print data=lsmeans_gs2;
run;
```