Supplement 1. Simulation-based Assessment of the Distribution of $d_{jk} = \dfrac{o_{jk} - i_{jk}}{o_{jk}}$.

Recall from Section 2 that to define a ratio that has finite variance, a truncated normal can be used as the data model in Eq. (2) for o_{jk} in $d_{jk} = 1 - \dfrac{i_{jk}}{o_{jk}}$, which is equal in distribution to $1 - \dfrac{\mu_{jk}(1+\delta_{TI} z_1)}{\mu_{jk}(1+\delta_{TO} z_2)} = 1 - R$, which involves a ratio $R = \dfrac{(1+\delta_{TI} z_1)}{(1+\delta_{TO} z_2)}$ of independent normal random variables z_1 and z_2 (for the case of one measurement per group; multiple measurements per group is treated similarly). Section 2 claimed that provided $\delta_{TO} \le 0.02$ and $\delta_{TI} \le 0.05$, the distribution of the truncated version of the ratio $d_{jk} = \dfrac{o_{jk} - i_{jk}}{o_{jk}}$ is extremely close to a normal distribution.

Supplement 1 provides 4 example numerical simulation results involving the distribution (*O-I*)/*O*, with O assumed to be a truncated normal, with truncation occurring only if O is at least 25 standard deviations from its mean value. Example 1 is the approximate variance result. Example 2 is a tolerance interval example with random normal error (no systematic error) for which there is an exact expression for the tolerance interval coverage factor, so simulation using a normal and a normal divided by a truncated normal can be compared. Example 3 is example density plots and normal probability plots with error bars showing that *O-I*)/*O* with a truncated O is approximately normal provided $\delta_{TO} \le 0.02$. Example 4 investigates the variances of the estimators $\hat{\delta}_R^2$ and $\hat{\delta}_S^2$ that arise from applying ANOVA to $d_{jk} = \dfrac{o_{jk} - i_{jk}}{o_{jk}}$.

Example S1.1: Approximate variance result for a ratio of a normal to a truncated normal

In R, the # sign denotes a comment. Comments are inserted below in red to explain.

```
nsim = 10^6; kx = 5; ky = 5. # factors to increase deltaor, deltaor, deltair, deltais
deltaor = .01*kx; deltaos = .01*kx; deltair = .01*ky; deltais = .01*ky
deltaot = (deltaor^2+deltaos^2)^.5; deltait = (deltair^2+deltais^2)^.5
temptrue =100; N = 100
temp1 = numeric(nsim); temp2 = numeric(nsim)
check = matrix(0,nrow=nsim,ncol=2)
for(isim in 1:nsim) {
  x = temptrue*(1+deltaor*rnorm(N) + deltaos*rnorm(N))  # note N for sys, so 1 obs per group
  x1 = pmax(lboundx,x).  # truncation: assume operator measurement is truncated normal
  y = temptrue*(1+deltair*rnorm(N) + deltais*rnorm(N))
  y1 = pmax(lboundy,y) # truncation not necessary for inspector, but does no harm
  temp1[isim] = var((x-y)/x)^.5 # non-truncated version
  temp2[isim] = var((x1-y1)/x1)^.5. # truncated version
}
```

```
temptot = (deltaor^2+deltaos^2+deltair^2+deltais^2)^.5
c(nsim,N,deltaor,deltaos,deltair,deltais,temptot)
[1] 1e+06 1e+02 1e-02 1e-02 1e-02 1e-02 2e-02. # approximation is 0.02
c(mean(temp1),mean(temp2))
[1] 0.01996 0.01996 # actual rounds to 0.02 for untruncated or truncated with 10^6 simulations
c(nsim,N,deltaor,deltaos,deltair,deltais,temptot)
[1] 1e+06 1e+02 2e-02 2e-02 2e-02 2e-02 4e-02. # approximation is 0.04
c(mean(temp1),mean(temp2))
[1] 0.03999 0.03999   # actual via simulation rounds to 0.04 for untruncated or truncated
temptot = (deltaor^2+deltaos^2+deltair^2+deltais^2)^.5
c(nsim,N,deltaor,deltaos,deltair,deltais,temptot)
[1] 1e+06 1e+02 5e-02 5e-02 5e-02 5e-02 1e-01 # exact is 0.10
c(mean(temp1),mean(temp2))
1] 0.1011 0.1011  # actual via simulation rounds to 0.10, truncated or not
c(nsim,N,deltaor,deltaos,deltair,deltais,temptot)
[1] 1e+06 1e+02 1e-01 1e-01 1e-01 1e-01 2e-01. # approximation is 0.20
c(mean(temp1),mean(temp2))
[1] 0.2118575 0.2118575  # actual rounds to 0.21, so the approximation begins to show error
[1] 1.0e+06 1.0e+02 1.5e-01 1.5e-01 1.5e-01 1.5e-01 3.0e-01 # approximation is 0.30 > 0.20
c(mean(temp1),mean(temp2),mean(temp3))
[1] 0.3587435 0.3587435 # actual rounds to 0.36, which is unacceptably different from 0.30
```

Example S1.2: Approximate normality of the ratio of a normal to a truncated normal

This example computes a tolerance interval coverage factor using either a normally distributed variate, or a ratio of normal variates in the one-sided normal case. In this one-sided one-group normal case, the exact coverage factor is known analytically (this is the only such case where the exact tolerance interval coverage factor is known analytically). This example is a "bottom-line" normality check in the context of this paper: and essentially the same result is obtained using normal or using a ratio of a normal to a truncated normal. Compare the boldface numbers below (all three are equal to within the simulation error in using a finite but large (10^6) number of simulations). The simulation results reported use (O-I)/O to compute the tolerance intervals in structured data (random and systematic errors).

```
n1 = 30; p1 = .05; p2 = .01  # p1 is 0.05 coverage, p2 is 99% confidence
del = qnorm(p=1-p1)*n1^.5; sd1 = .05;sd2 = .02; tsd = (sd1^2 + sd2^2)^.5
nsim = 10^6; mu = 0; k.n = 10^3; kseq = seq(2.3,2.7,length=k.n) # after initial run to zoom kseq
tempmat1 <- matrix(0,nrow=nsim,ncol=k.n); tempmat2 <- matrix(0,nrow=nsim,ncol=k.n)
for(isim in 1:nsim) {
 temp1 = mu + rnorm(n=n1,sd=tsd)
 truncated.normal =pmax(0.5,1+rnorm(n=n1,sd=sd2)) # this truncation will almost never occur
 temp1a = (1+ rnorm(n=n1,sd=sd1))/truncated.normal
 temp1a = temp1a-1
```

```
  temp2 = mean(temp1) + kseq*var(temp1)^.5
  temp2a = mean(temp1a) + kseq*var(temp1a)^.5
  tempmat1[isim,] = as.numeric(temp2 >= qnorm(1-p1,sd=tsd))
  tempmat2[isim,] = as.numeric(temp2a >= qnorm(1-p1,sd=tsd))
}

c(n1,p1,p2,del,sd1,sd2,tsd,nsim)
min(kseq[apply(tempmat1,2,mean)>= 1-p2])
min(kseq[apply(tempmat2,2,mean)>= 1-p2])
#rep1 of 10^6 simulations
c(n1,p1,p2,del,sd1,sd2,tsd,nsim)
3.00e+01 5.00e-02 1.00e-02 9.01e+00 5.00e-02 2.00e-02 5.39e-02 1.00e+06
min(kseq[apply(tempmat1,2,mean)>= 1-p2])
2.516617. # simulation-based, using a normal
min(kseq[apply(tempmat2,2,mean)>= 1-p2])
2.515015 # simulation-based, using a ratio of a normal to a truncated normal
#rep2.of 10^6 simulations to be sure that 10^6 is enough simulations to ignore simulation error
min(kseq[apply(tempmat1,2,mean)>= 1-p2])
2.517417 # simulation-based, using a normal
min(kseq[apply(tempmat2,2,mean)>= 1-p2])
2.517017# simulation-based, using a ratio of a normal to a truncated normal
# exact for 1 sided. # the exact k value is only available for the 1-sided normal tolerance interval
del <- qnorm(p=1-p1)*n1^.5
# this is k:
qt(p=1-p2,df=n1-1,ncp=del)/n1^.5
2.515486. # exact, essentially the same as those above from simulation.
```

Example S1.3: Example normality checks for the ratio

A large number (10^4) observations were simulated from a normal and from a ratio of a normal to a truncated normal. Figures S1.1, S1.2, and S1.3 illustrate that the ratio is extremely close to normal in distribution provided $\delta_{TO} \leq 0.02$ and $\delta_{TI} \leq 0.05$.

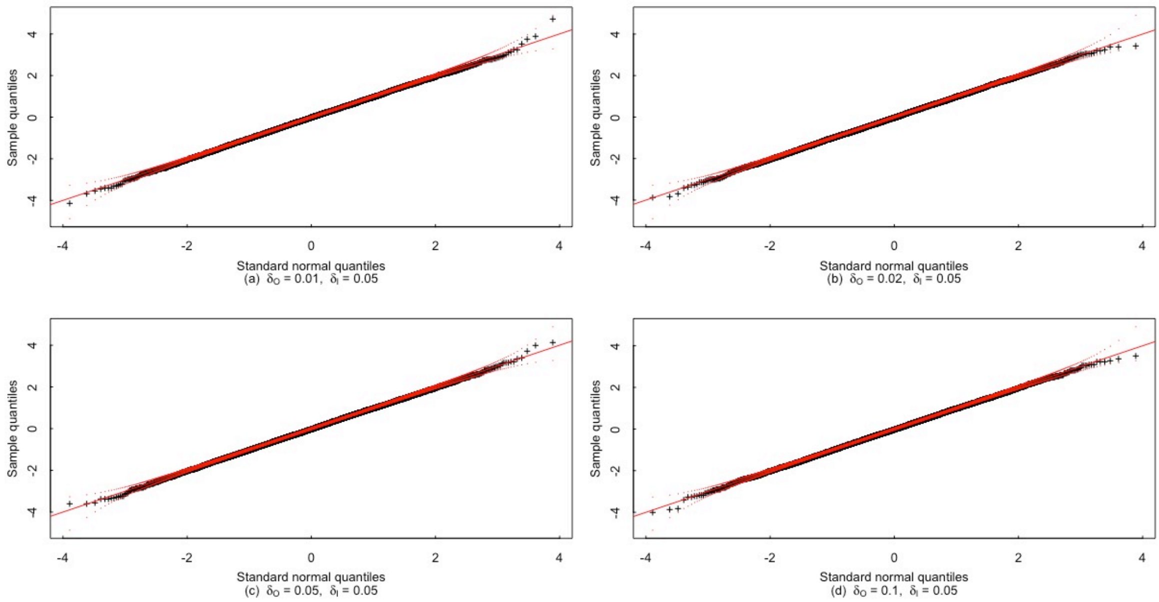

Figure S1.1. Normality checks using normal probability plots and “error bars” (based on simulation) for a normal random variable using 10^4 observations. As expected, normal data “passes” this normality test. In all plots (Figures a-d), $\delta_{TI} = \sqrt{\delta_{RI}^2 + \delta_{SI}^2} = 0.05$ as an example.

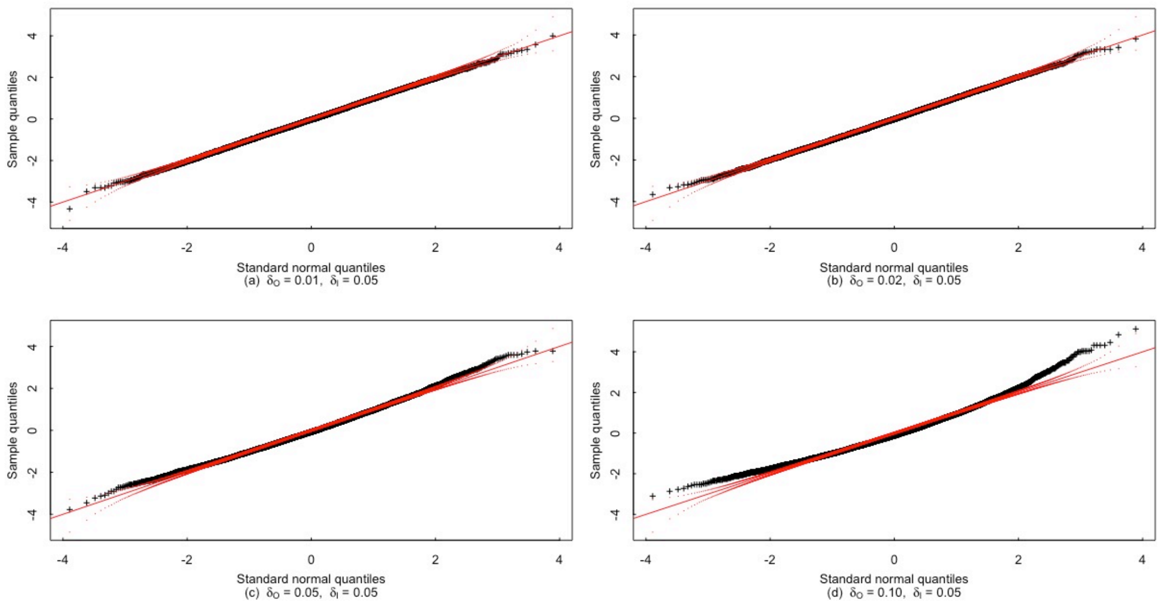

Figure S1.2. Normality checks using normal probability plot sand “error bars” for a ratio of a normal to a truncated normal using 10^4 observations. This ratio data “passes” this normality test provided $\delta_{TO} = \sqrt{\delta_{RO}^2 + \delta_{SO}^2} \leq 0.02$ (top two plots), and begins to show departure from normality if $\delta_{TO} = \sqrt{\delta_{RO}^2 + \delta_{SO}^2} = 0.05$ in these plots with $\delta_{TI} = \sqrt{\delta_{RI}^2 + \delta_{SI}^2} = 0.05$.

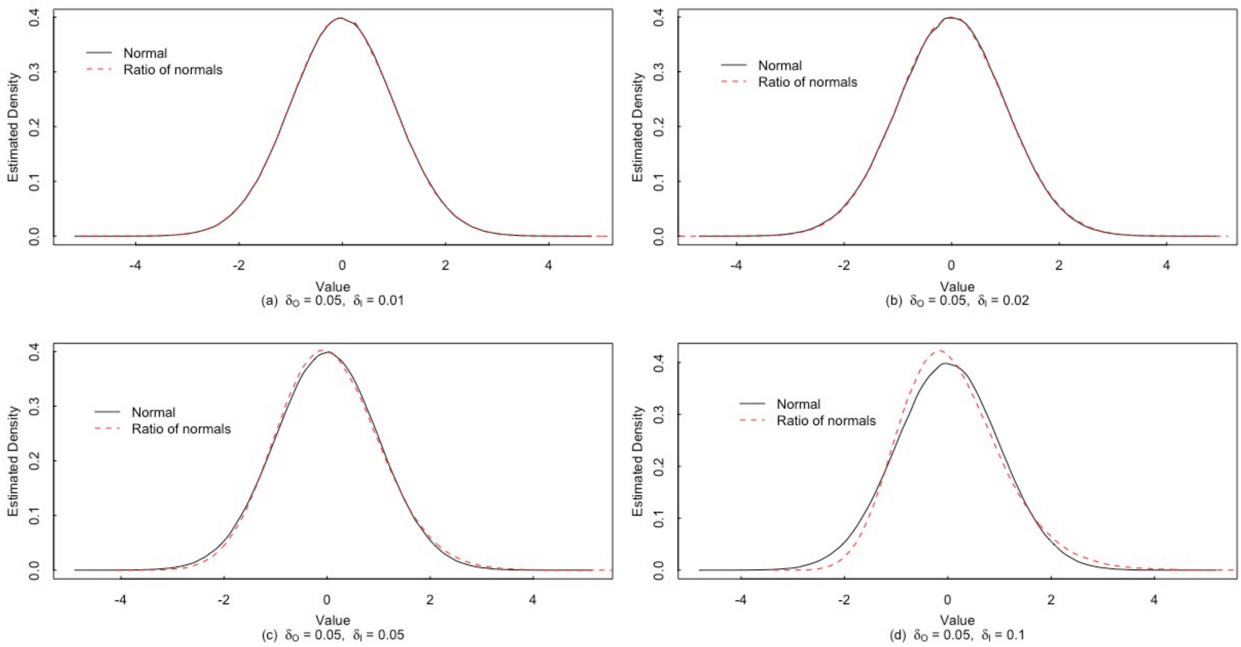

Figure S1.3. The estimated probability density for the same 4 cases as in Figure 2.

Example S1.4. The Variances of the ANOVA-based Estimators $\hat{\delta}_R^2$ and $\hat{\delta}_S^2$

Example 4 investigates the variances of the estimators $\hat{\delta}_R^2$ and $\hat{\delta}_S^2$ that arise from applying ANOVA to $d_{jk} = \frac{o_{jk} - i_{jk}}{o_{jk}}$. The point of this example is that it is defensible to assume that $d_{ij} = (o_{jk} - i_{jk})/o_{jk}$ is approximately normal under Eq. (2), with a variance that is well approximately by linear propagation of error variance (Example S1.1) ,and that the variance of the variance estimates are also well approximated as follows:

```
# columns 1 and 2 are $\hat{\delta}_R^2$and $\hat{\delta}_{S,}^2$,   respectively,  for $d_{ij} = (o_{jk} - i_{jk})/o_{jk}$
# columns 3 and 4 are $\hat{\delta}_R^2$and $\hat{\delta}_{S,}^2$,   respectively,  for $d_{ij} = (o_{jk} - i_{jk})/\square_{jk}$
# so, columns 3 and 4 are the same as an additive model, as in standard ANOVA with normal data
nsim = 10^5; check.mat =matrix(0,nrow=nsim,ncol=4)
for(isim in 1:nsim) {
# simulate 3 groups of 10 measurements per group from Eq. (2):
temp1 =
generate.data(ngroups=3,nvec=rep(10,3),sigma.r.o=0.01,sigma.r.i=0.01,sigma.s.o=0.005,sigma.s.i=0.01)
# compute d:
  dtemp = (temp1[,3]-temp1[,2])/temp1[,2]
# use the usual ANOVA estimates of random and systematic error variances:
  temp2 = estvars0(groups=temp1[,1],d=dtemp). # gives same result as lmer() in R
  check.mat[isim,1:2] = temp2[1:2]
  temp1 = generate.data(ngroups=3,nvec=rep(10,3),
```

```
  sigma.r.o=0,sigma.r.i=rtotsd,sigma.s.o=0,sigma.s.i=stotsd)
  dtemp = (temp1[,3]-temp1[,2])/temp1[,2]
  temp2 = estvars0(groups=temp1[,1],d=dtemp)
  check.mat[isim,3:4] = temp2[1:2]
}
# compare approximation to "exact' (nearly exact with 10^5 simulations)
apply(check.mat,2,mean)
0.0002001705 0.0001255540 0.0002001561 0.0001255132.
# column 1 (ratio) is approximately the same as column 3 (normal) and
# column 2 (ratio) is approximately the same as column 4 (normal).
stotvar = (.01^2+.005^2); rtotvar = (.01^2+.01^2)
c(rtotvar,stotvar)
0.000200 0.000125 # agrees with simulation

apply(check.mat,2,var)^.5
5.462549e-05 1.455530e-04 5.468783e-05 1.455614e-04
# column 1 (ratio) is approximately the same as column 3 (normal) and
# column 2 (ratio) is approximately the same as column 4 (normal).
# rep2 of 10^5 simulations:
apply(check.mat,2,mean)
0.0001999014 0.0001239302 0.0002000479 0.0001243459
apply(check.mat,2,var)^.5
5.439924e-05 1.437925e-04 5.456276e-05 1.438669e-04
```