**Chapter 9: DBMH Analysis: Online Appendices**

# Online Appendix 9.A: The Satterthwaite degree of freedom

This Appendix is for those who have asked the author questions like "*why is ddf not an integer*" and "*what is the Satterthwaite approximation*"? Since in the author's opinion this is not adequately dealt with in the ROC literature, the following explanation is based on the original papers by Satterthwaite[1](#_ENREF_1),[2](#_ENREF_2), dating to the early 1940s.

Suppose  is a continuous random variable sampled from a zero-mean normal distribution with variance :

 .

Letbe the total number of independent realized samples , . The (unbiased) estimate of sample variance  is defined by:

 .

Satterthwaite calls a *simple estimate of variance*; it has degrees of freedom *N* - 1. Let  denoted the corresponding random variable (an estimate is a realization of a random variable). It can be shown[3](#_ENREF_3) that  follows a chi-square distribution with  degrees of freedom (Theorem 7.3.3 in “*An Introduction to Mathematical Statistics and Its Applications*” and see software demonstration below, file mainLarsenMarx455.R):

 .

Therefore the distribution of is:

 .

Since the mean of a chi-square distribution with  degrees of freedom is  (7.3.7 ibid) it follows that the expected value of  is , i.e., . Therefore,

 .

Consider the distribution of a *linear combination of simple estimates of variance*, i.e., the problem addressed by Satterthwaite. Let and  denote two simple estimates of variance with expected values and , respectively, with corresponding (integer) degrees of freedom  and . This implies that  is based on observations, and likewise  is based on observations. Consider the sum:

 .

It turns out that even for this simplest of cases the distribution of **, while computable, is too complex for practical use (while true in 1941 it is probably not true now). Satterthwaite's idea was to approximate the exact distribution with a chi-square distribution with an appropriately chosen degree of freedom , not necessarily restricted to integers (subscript *Sat* = Satterthwaite). To determine  he imposed the condition that the theoretical and approximate distributions of **have the same variance. Since ** is itself an estimate of variance, one is entering Alice's world of a possibly endless quest of estimating the variance of a variance. The variance of  is given by (see explanation below):

 .

This equation is derived as follows (Satterthwaite's 1941 paper is missing a factor of two which does not affect the final result). From Eqn. the distribution of  (*i* = 1,2) is:

 .

One does not need to subtract 1 as the degree of freedom is based on observations. Eqn. implies:

 .

Since the variance of a chi-square distribution is twice the number of degrees of freedom (7.3.7 ibid),

 .

Finally, and here comes the great leap of faith: analogous to Eqn. the hope is that for :

 .

Since the sum of variances of independent random variables is the sum of the individual variances:

 .

This implies (the twos cancel):

 .

Since the *expected* values are generally unknown (one would need the distributions of the variables, and the ability to perform the necessary integrations to calculate the expected values) in practice one uses *estimates* (i.e., observed values) in place of expected values:

 .

[The physicist in the author thinks of this as adding resistors in parallel.] This equation allows one to estimate (do not be surprised if is not an integer – an integer value would be an incredible coincidence), specifically:

 .

Eqn. is readily extended to a linear combination of several simple estimates of variance:

 .

Using estimates (i.e., observed values) in place of expected values:

 .

The mean squares calculated in ANOVA are variance-like quantities, so the Satterthwaite approximation can be applied to them:

 .

This approximation is relevant in the DBM F-statistic, book Equation 9.18, for testing the null hypothesis that the treatment effect is zero (ignoring the Hillis modification for now):

 .

The distribution of this statistic under the null hypothesis is, book Equation 9.19:

 .

The numerator degree of freedom is  - no approximation needed here. The denominator is a linear combination of mean squares, with degrees of freedom (in order of the appearance of the terms in the denominator): ,  and , respectively. Therefore, applying Satterthwaite's approximation, on obtains book Equation 9.22:

 .

### Online Appendix 9.A.1 Code illustration of Eqn.

First the author illustrates the theorem implicit in Eqn. , namely,  follows the chi-square distribution with *N*-1 degrees of freedom, where *N* is the number of samples over which the variance is estimated. The filename is mainLarsenMarx455.R, reflecting the page number of the book, where this theorem is to be found[3](#_ENREF_3) .

### Online Appendix 9.A.2: Code Listing

# mainLarsenMarx455.R

# this code illustrates theorem that estimated variance times (N-1)

# divided by true variance follows the chi-square distribution with (N-1) df

# see Larsen and Marx, page 455??

rm(list = ls());seed <- 1;set.seed(seed)

require(stats)

library(ggplot2)

sigma <- 0.4#true value

N <- 10#number of samples over which variance will be estimated

S <- 1000 #number of independent simulations over which variance of variance will be estimated

Var <- array(dim = S)# to hold individual variance estimates

scaledS2 <- array(dim = S)#to hold (N-1)\*Var[s]/sigma^2 estimates

for (s in 1:S) {

Var[s] <- var(rnorm(N, sd = sigma))# sample variance for this simulation

scaledS2[s] <- (N-1)\*Var[s]/sigma^2

}

# preferred way of testing a model

x <- scaledS2

qqData <- data.frame(x = sort(x), quantile = sort(qchisq(ppoints(x), df = N-1)))

qqPlot <- ggplot(data = qqData, mapping = aes(x = x, y = quantile)) +

geom\_point() + geom\_abline(slope = 1, color = "red", linetype = 2)

print(qqPlot)

# if you insist on using histogram...

pdfData <- data.frame(x = x, y = dchisq(x, df = N-1))

histogram <- ggplot(data = qqData, mapping = aes(x = x)) + geom\_histogram(mapping = aes(y = ..density..), color = "black", fill = "grey") +

geom\_line(mapping = aes(y = y), data = pdfData, color = "red", linetype = 2)

print(histogram)

Line 9 sets the true value of to 0.4. Line 10 sets , the number of samples over which the individual estimates of variances was obtained, to 10. Line 12 sets the number of simulations S to 1000, which is the number of independent trials for each of which the variance of 10 samples from  will be estimated. Line 14-15 creates arrays for holding the S values of Var and scaledS2, where scaledS2 represents the left hand side of Eqn. , i.e. the variable that is expected to be distributed  . Lines 18-19 perform the sampling, variance estimation and the assignments. For example, line 18 uses rnorm() to obtain N samples from a normal distribution with standard deviation sigma = 0.4, and then calculates the variance of the samples, which is assigned to Var[s]. The next line assigns the properly scaled variance estimate to scaledS2[s]. Line 23-26 shows the preferred way of visually testing a model, namely a quantile-quantile or *Q-Q* plot[4](#_ENREF_4). Line 23 assigns the 1000 element array scaledS2 to x. Line 24 defines the qqData frame.The next two lines calls the ggplot() function. Insert a break point at line 29 and clicking on Source yielding Figure 1 (A). The overlaid dotted straight line, created by the geom\_abline() option to ggplot(), has unit slope and passes through the origin. The visual quality of the fit is excellent. This shows that the scaled variance, namely the left hand side of Eqn. indeed behaves like a chi-square random variable with N - 1 = 9 degrees of freedom.

|  |  |
| --- | --- |
| **(A)** | **(B)** |

Figure 1: (A) Q-Q plot showing fit of left hand side of Eqn. against a chi-square random variable with 9 degrees of freedom. (B) Histogram of left hand side of Eqn. with superposed pdf of chi-squared distribution with df = 9. These plots were generated by

Click Continue to execute the rest of the code. These generate the corresponding histogram with the overlaid *pdf*, Figure 1 (B). Again, the fit looks good, but perhaps not as compelling as the presentation in Figure 1 (A). If a quantitative goodness of fit statistic is desired, the starting point would be the histogram, splitting up the x-axis into bins, and computing the chi-square goodness of fit statistic by comparing actual counts to expected counts; i.e., as in book chapter 6.

### Online Appendix 9.A.3: Code illustration of Satterthwaite approximation

This code explores the adequacy of the Satterthwaite approximation. Open the file mainSatterthwaite1941.R.

### Online Appendix 9.A.3.1: Code Listing

#this illustrates Satterthewaites' approximation as described in his 1941 paper

# mainSatterthwaite1941.R

rm(list = ls());seed <- 1;set.seed(seed)

library(ggplot2)

sigma1 <- 1;sigma2 <- 4

N1 <- 5;N2 <- 2 # expected to give bad approximation; use N1 <- 15;N2 <- 20 for better fit

cat("sigma1 = ", sigma1, "sigma2 = ", sigma2, "N1 = ", N1, "N2 = ", N2, "\n")

r1 <- N1-1;r2 <- N2-1

# s1Sqd and s2Sqd are what Satterthwaite calls "simple estimates of variance"

s1Sqd <- var(rnorm(N1, sd = sigma1)) # same as sum((xx1-mean(xx1))^2)/(N1-1)

s2Sqd <- var(rnorm(N2, sd = sigma2))

rSat <- (s1Sqd+s2Sqd)^2/(s1Sqd^2/r1+s2Sqd^2/r2) # Satterthwaite approximation

cat("Satterthwaite df = ", rSat, ", approx. accuracy (should be close to unity) = ",

r1\*s2Sqd/(r2\*s1Sqd), "\n") #if second qnty is close to one, approximation should work

S <- 1000

v1 <- array(dim = S);v2 <- array(dim = S)

for (s in 1:S) {

v1[s] <- var(rnorm(N1, sd = sigma1))# simple estimate of variance

v2[s] <- var(rnorm(N2, sd = sigma2))# do

}

scaledS2 <- rSat\*(v1 + v2)/(sigma1^2+sigma2^2)# scaled values of v1+v2

x <- scaledS2

qqData <- data.frame(scaledS2 = sort(x), quantile = sort(qchisq(ppoints(v1), df = rSat-1)))

qqPlot <- ggplot(data = qqData, mapping = aes(x = scaledS2, y = quantile)) + geom\_point() + geom\_abline(slope = 1, color = "red", linetype = 2) +

xlab(label = "Scaled S2") + ylab(label = "Chi-square Quantile")

print(qqPlot)

Line 5 defines  and , line 6 defines the corresponding number of samples  and : if not set to these values, do set them to the stated values. Line 7 prints these values, line 8 defines the degrees of freedom  and , line 10-11 calculate the "simple estimate" variances s1Sqd and s2Sqd, corresponding to and . Line 12 implements the Satterthwaite approximation Eqn. , where rSat is the Satterthwaite degrees of freedom. Line 13-14 prints rSat and the quantity:

 .

According to Satterthwaite, this quantity, termed "approx. accuracy", should be close to unity (assuming it is larger than one; if not use its reciprocal) if the approximation is to have a reasonable chance of success. The approximation also gets better the larger the values of  and . Clicking on Source yields the following output:

### Online Appendix 9.A.3.2: Code Output

> source(...)

sigma1 = 1 sigma2 = 4 N1 = 5 N2 = 2

Satterthwaite df = 1.138241 , approx. accuracy (should be close to unity) = 59.26727

For the chosen values, the approximation is expected to be quite poor. The approximation accuracy is 59.3, which is much larger than one. Line 16-27 sets up simulations to visually test the quality of the approximation. Line 19 obtains N1 samples from a zero-mean normal distribution with standard deviation sigma1, computes the variance of the sampled values, and assigns the result to v1[s], where s is the simulation index, ranging from 1 to S = 1000. Line 20 obtains N2 samples from a normal distribution with standard deviation sigma2, computes the variance of the sampled values, and assigns the result to v2[s]. Line 22 computes the sum of the two arrays v1 and v2, each containing simple estimates of variance, and scales it appropriately for comparison to a chi-square distribution, and assigns the result to scaledS2. [One is using the left hand side of Eqn. , i.e., multiplying the sum of v1 and v2 by rSat and dividing by the true variance of the sum.] Line 22 implements the Q-Q plot. Sourcing this file yields Figure 2 (A), evidence of a poor fit. Now change line 6 to read N1 <- 15;N2 <- 20 and source the file. Under these conditions the fit is quite good, Figure 2 (B), even though approx. accuracy = 8.5 is not that close to unity.

### Online Appendix 9.A.3.3: Code Output

> source(...)

sigma1 = 1 sigma2 = 4 N1 = 15 N2 = 20

Satterthwaite df = 22.2176 , approx. accuracy (should be close to unity) = 8.48024

|  |  |
| --- | --- |
| **(A)** | **(B)** |

Figure 2: (A) Q-Q plot showing quality of Satterthwaite approximation with N1 = 5 and N2 = 2; the approximation is very poor. (B) Q-Q plot showing quality of Satterthwaite approximation with N1 = 15 and N2 = 20; the approximation is good.

# Online Appendix 9.B: Demonstration of significance testing formulae

The following code implements the significance testing formulae. It is the longer version of the code in mainDBMHBrief.R. The longer version allows one to see what is going on and thus gain better understanding of the analysis. It still uses the RJafroc package to perform some of the tasks, e.g., reading an input file, calculating pseudovalues and mean squares, etc., but the rest of the analysis is shown more transparently. Open the file mainDBMH.R:

### Online Appendix 9.B.1: Code Listing

rm(list = ls()) # mainDBMH.R

library(RJafroc)

alpha <- 0.05

#fileName <- "Franken1.lrc"

fileName <- "VanDyke.lrc"

rocData <- ReadDataFile(fileName, format = "MRMC")

pseudoValues <- PseudoValues(rocData, FOM = "Wilcoxon")

I <- dim(pseudoValues)[1]

J <- dim(pseudoValues)[2]

K <- dim(pseudoValues)[3]

retMS <- MeanSquares(rocData, FOM = "Wilcoxon")

FOM <- FigureOfMerit(rocData, FOM = "Wilcoxon")

cat("\nRandom-reader random-case analysis\n")

MS\_DEN\_DIFF\_FOM\_RRRC <- retMS$msTR+max(retMS$msTC - retMS$msTRC,0)

F\_DBMH <- retMS$msT / MS\_DEN\_DIFF\_FOM\_RRRC

ndf <- (I-1)

ddfH <- MS\_DEN\_DIFF\_FOM\_RRRC^2/(retMS$msTR^2/((I-1)\*(J-1)))

cat("Hillis ddfH = ", ddfH, "\n")

FCrit <- qf(1 - alpha, ndf, ddfH);cat("F statistic is ", F\_DBMH, ", the critical value of F is ", FCrit, "\n")

pValueH <- 1 - pf(F\_DBMH, ndf, ddfH);cat("p-value is ", pValueH, "\n")

trtMeans <- array(dim = I)

for (i in 1:I) trtMeans[i] <- mean(FOM[i,])

trtDiff <- array(dim = c(I,I))

for (i1 in 1:(I-1)) {

for (i2 in (i1+1):I) {

trtDiff[i1,i2] <- trtMeans[i1]- trtMeans[i2]

}

}

trtDiff <- trtDiff[!is.na(trtDiff)]

std\_DIFF\_FOM\_RRRC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_RRRC/J/K)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_RRRC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_RRRC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_RRRC[i,2] <- qt(alpha/2,df = ddfH)\*std\_DIFF\_FOM\_RRRC + trtDiff[i]

CI\_DIFF\_FOM\_RRRC[i,3] <- qt(1-alpha/2,df = ddfH)\*std\_DIFF\_FOM\_RRRC + trtDiff[i]

cat("mean diff is ", CI\_DIFF\_FOM\_RRRC[i,1], " and 95% CI is ", CI\_DIFF\_FOM\_RRRC[i,2], CI\_DIFF\_FOM\_RRRC[i,3], "\n")

}

cat("\nFixed-reader random-case analysis\n")

MS\_DEN\_DIFF\_FOM\_FRRC <- retMS$msTC

FDbmFR <- retMS$msT / MS\_DEN\_DIFF\_FOM\_FRRC

ndf <- (I-1)

ddf <- (I-1)\*(K-1)

cat("ddf = ", ddf, "\n")

FCrit <- qf(1 - alpha, ndf, ddf);cat("F statistic is ", FDbmFR, "and critical value of F is ", FCrit, "\n")

pValue <- 1 - pf(FDbmFR, ndf, ddf);cat("p-value is ", pValue, "\n")

std\_DIFF\_FOM\_FRRC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_FRRC/J/K)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_FRRC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_FRRC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_FRRC[i,2] <- qt(alpha/2,df = ddf)\*std\_DIFF\_FOM\_FRRC + trtDiff[i]

CI\_DIFF\_FOM\_FRRC[i,3] <- qt(1-alpha/2,df = ddf)\*std\_DIFF\_FOM\_FRRC + trtDiff[i]

cat("mean diff is ", CI\_DIFF\_FOM\_FRRC[i,1], " and 95% CI is ", CI\_DIFF\_FOM\_FRRC[i,2], CI\_DIFF\_FOM\_FRRC[i,3], "\n")

}

cat("\nRandom-reader fixed-case analysis\n")

FDbmFC <- retMS$msT / retMS$msTR

ndf <- (I-1)

ddf <- (I-1)\*(J-1)

cat("ddf = ", ddf, "\n")

FCrit <- qf(1 - alpha, ndf, ddf);cat("F statistic is ", FDbmFC, "and critical value of F is ", FCrit, "\n")

pValue <- 1 - pf(FDbmFC, ndf, ddf);cat("p-value is ", pValue, "\n")

MS\_DEN\_DIFF\_FOM\_RRFC <- retMS$msTR

std\_DIFF\_FOM\_RRFC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_RRFC/J/K)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_RRFC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_RRFC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_RRFC[i,2] <- qt(alpha/2,df = ddf)\*std\_DIFF\_FOM\_RRFC + trtDiff[i]

CI\_DIFF\_FOM\_RRFC[i,3] <- qt(1-alpha/2,df = ddf)\*std\_DIFF\_FOM\_RRFC + trtDiff[i]

cat("mean diff is ", CI\_DIFF\_FOM\_RRFC[i,1], " and 95% CI is ", CI\_DIFF\_FOM\_RRFC[i,2], CI\_DIFF\_FOM\_RRFC[i,3], "\n")

}

Line 2 loads the RJafroc package. Line 4 defines the maximum allowed Type I error rate . Line 6 defines fileName as the Van Dyke[5](#_ENREF_5) data file "VanDyke.lrc". Line 7 uses the RJafroc function DfReadDataFile() to read the contents of the Van Dyke data file and saves it to the dataset object rocData. Line 8 calculates the pseudovalues, using function UtilPseudoValues(), using the empirical AUC as the FOM. Line 12 calculates the mean squares, using Eqn. (9.12), implemented in the function UtilMeanSquares(). Line 13 calculates, using the function UtilFigureOfMerit(), the *I* x *J* array of FOM values. All calculations are done using the Wilcoxon statistic.

Put a breakpoint (red dot) to the left of line 15 and click on Source. Try printing out some values (highlight variable followed by Run), using the following as a guide.

### Online Appendix 9.B.2: Code Snippets

Browse[2]> FOM

Rdr - 0 Rdr - 1 Rdr - 2 Rdr - 3 Rdr - 4

Trt - 0 0.9196457 0.8587762 0.9038647 0.9731079 0.8297907

Trt - 1 0.9478261 0.9053140 0.9217391 0.9993559 0.9299517

Browse[2]> retMS

$msT

[1] 0.5467634

$msR

[1] 0.4373268

$msC

[1] 0.3968699

$msTR

[1] 0.06281749

$msTC

[1] 0.09984808

$msRC

[1] 0.06450106

$msTRC

[1] 0.0399716

$msCSingleT

[1] 0.3362943 0.1604237

$msCSingleR

[1] 0.12224980 0.21274824 0.13653391 0.01728937 0.16605281

One sees that the list variable retMS contains the desired mean squares.

Line 15 prints out a message signaling the beginning of random-reader random-case analysis. Line 16 calculates the denominator of the F-statistic, by implementing the denominator of book Equation (9.23). The variable MS\_DEN\_DIFF\_FOM\_RRRC is an abbreviation for "*mean square denominator difference-FOM random reader random case*". Line 17 calculates the observed value of the F-statistic, book Equation (9.23), denoted F\_DBMH. Line 18 trivially calculates *ndf* and line 19 implements book Equation (9.24) to calculate  and line 20 prints out . Line 21-22 calculates the critical value of the F-statistic and prints it out and line 23 calculates the p-value and prints it.

Click on Stop to exit debug mode. Remove the breakpoint at line 15 (by clicking on the red dot; it should disappear); insert a new breakpoint at line 25 and click on Source.

> debugSource(...)

Random reader random case analysis

Hillis ddfH = 15.25967

F statistic is 4.456319 , the critical value of F is 4.529639

p-value is 0.05166569

…

The observed F-statistic, 4.46, is slightly smaller than the critical value, 4.530, consistent with the observed p-value (0.0517) being slightly greater than 0.05. The same values were obtained earlier using the code in mainDBMHBrief.R, as can be appreciated by comparing these values to those in book Table 9.3. For this data set the p-value is not significant.

Line 25 initializes the length *I* array trtMeans to the reader averaged FOMs for each treatment (implemented using the mean() function). Click repeatedly on Next until the cursor gets to line 27, then highlight trtMeans and click on Run:

### Online Appendix 9.B.3: Code Snippets

Browse[2]> trtMeans

[1] 0.8970370 0.9408374

These are the reader averaged empirical AUCs for the two treatments. Lines 28 – 33 calculates inter-treatment FOM differences; in the current example there is only one difference, that between treatment 1 and treatment 2. The longer code is needed to handle more than two treatments; the array dimensioning is wasteful of memory, but simpler to follow. Line 33 extracts differences where they exist (the remaining elements of the I x I trtDiff array are filled with NAs). Line 35 calculates the multiplier of the term in book Equation (9.31), called std\_DIFF\_FOM\_RRRC, an abbreviation for "*standard deviation difference-FOM random-reader random-case*". The variable CI\_DIFF\_FOM\_RRRC (for "9*5% confidence interval difference FOM random-reader random-case*") is an nDiffs by 3 array, where nDiffs is defined at line 36 (= *I(I-1)/2 =* one for *I = 2*) and for each difference there are three terms, the 1st is the value of the difference (the lower numeric indexed treatment minus the higher numeric indexed treatment), the 2nd is the lower limit and the 3rd is the upper limit of the 95% confidence interval. These are calculated in lines 39 – 41, where the last two lines perform the necessary multiplications by  shown in book Equation (9.31). Line 42-43 prints out the value of the difference and the lower and upper 95% CIs.

Exit Debug mode by clicking on Stop, remove the breakpoint at line 25 insert one at line 46, and click on Source.

### Online Appendix 9.B.4: Code Output

> debugSource('~/book2/03 B Statistics of ROC analysis/B2 DBMH Analysis/software/mainDBMH.R')

Random reader random case analysis

Hillis ddfH = 15.25967

F statistic is 4.456319 , the critical value of F is 4.529639

p-value is 0.05166569

mean diff is -0.04380032 and 95% CI is -0.0879595 0.0003588544

...

Highlight CI\_DIFF\_FOM\_RRRC and click Run.

### Online Appendix 9.B.5: Code Snippets

Browse[2]> CI\_DIFF\_FOM\_RRRC

[,1] [,2] [,3]

[1,] -0.04380032 -0.0879595 0.0003588544

The 1st is the value of the FOM difference ("1-2"), the 2nd is the lower limit and the 3rd is the upper limit of the 95% confidence interval. Compare the two code snippets above for consistency.

Lines 46 – 54 perform fixed-reader random-case analysis. Line 48 calculates the F-statistic, where the numerator is the same as before, i.e., *MST*, but the denominator is MSTC, see book Equation (9.38). In the code the denominator is called MS\_DEN\_DIFF\_FOM\_FRRC (abbreviation for "*mean square denominator difference-FOM fixed reader random case*"). Line 52 - 54 calculates the critical value of the F-distribution and the p-value, respectively, and prints them. Lines 56 – 65 calculates the reader-averaged treatment differences and the upper and lower limits of the CIs.

Lines 67 – 86 perform random-reader fixed-case analysis. Click on Stop, remove any existing breakpoints, Ctrl-L to clear the Console screen, and click on Source:

### Online Appendix 9.B.6: Code Output

> source(...)

Random reader random case analysis

Hillis ddfH = 15.25967

F statistic is 4.456319 , the critical value of F is 4.529639

p-value is 0.05166569

mean diff is -0.04380032 and 95% CI is -0.0879595 0.0003588544

Fixed reader random case analysis

ddf = 113

F statistic is 5.475953 and critical value of F is 3.925076

p-value is 0.02103497

mean diff is -0.04380032 and 95% CI is -0.08088303 -0.006717613

Random reader fixed case analysis

ddf = 4

F statistic is 8.704 and critical value of F is 7.708647

p-value is 0.04195875

mean diff is -0.04380032 and 95% CI is -0.08502022 -0.00258042

For random-reader random-case (RRRC) analysis the observed value of the F-statistic is smaller than the critical value, the p-value is greater than 0.05 and the 95% confidence interval includes zero, all which tell us that the NH cannot be rejected at the 5% significance level. As one might expect, if one "freezes" reader or case variability, the difference becomes significant, whether viewed from the point of view of the F-statistic exceeding the critical value, the observed p-value being smaller than alpha or the 95% CI for the difference FOM not including zero. Also, compare these results to those obtained using mainDBMHBrief.R, specifically compare the current results to those shown in book Table 9.3, book Table 9.4 and book Table 9.5.

# Online Appendix 9.C: Text output file listing

The following lists the contents of the traditional format output file "VanDykeOutput.txt" created by sourcing the code in mainDBMHBrief.R. The output format is closely modeled on earlier versions of DBM-MRMC software from the University of Iowa ROC website.

RJafroc SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS

IN THE SOFTWARE.

================================================================================

Package build stats: R 3.2.3; x86\_64-apple-darwin13.4.0; 2016-04-24 20:19:18 UTC; unix

Run date: Jun 23 2016 Thu 22:00:57 EDT

FOM selected : Wilcoxon

Input Data : VanDyke.lrc

Output Data Filename : VanDykeOutput.txt

================================================================================

Significance testing method: DBM-MRMC HILLIS SIGNIFICANCE TESTING

Number of Readers : 5

Number of Treatments : 2

Number of Normal Cases : 69

Number of Abnormal Cases : 45

Fraction of Normal Cases : 0.605263

================================================================================

====================================================================

\*\*\*\*\* Overview \*\*\*\*\*

====================================================================

Three analyses are presented:

(1) Analysis 1 treats both readers and cases as random samples

--results apply to the reader and case populations;

(2) Analysis 2 treats only cases as a random sample

--results apply to the population of cases but only for the

readers used in this study; and

(3) Analysis 3 treats only readers as a random sample

--results apply to the population of readers but only for the

cases used in this study.

For all three analyses, the null hypothesis of equal treatments is

tested in part (a), treatment difference 95% confidence intervals

are given in part (b), and treatment 95% confidence intervals are

given in part (c). Parts (a) and (b) are based on the treatment x

reader x case ANOVA while part (c) is based on the reader x case

ANOVA for the specified treatment; these ANOVA tables are displayed

before the analyses. Different error terms are used as indicated

for parts (a), (b), and (c) according to whether readers and cases

are regarded as fixed or random factors. Note that the treatment

confidence intervals in part (c) are based only on the data for the

specified treatment, rather than the pooled data. Treatment

difference 95% confidence intervals for each reader are presented

in part (d) of Analysis 2; each interval is based on the treatment

x case ANOVA table (not included) for the specified reader.

===========================================================================

\*\*\*\*\* Estimates \*\*\*\*\*

===========================================================================

TREATMENT

-----------------------

READER 0 1

---------- ---------- ----------

0 0.91964573 0.94782609

1 0.85877617 0.90531401

2 0.90386473 0.92173913

3 0.97310789 0.99935588

4 0.82979066 0.92995169

TREATMENT MEANS (averaged across readers)

---------- -----------------------------

0 0.89703704

1 0.94083736

TREATMENT MEAN DIFFERENCES

---------- ---------- -----------

0 - 1 -0.04380032

===========================================================================

\*\*\*\*\* ANOVA Tables \*\*\*\*\*

===========================================================================

TREATMENT X READER X CASE ANOVA

Source SS df MS

------ -------------------- ------ ------------------

T 0.54676344 1 0.54676344

R 1.74930720 4 0.43732680

C 44.84629692 113 0.39686988

TR 0.25126996 4 0.06281749

TC 11.28283352 113 0.09984808

RC 29.15447929 452 0.06450106

TRC 18.06716464 452 0.03997160

Total 105.89811497 1139

TREATMENT X READER X CASE ANOVA

Mean Squares

Source df 0 1

------ --- ---------- ----------

R 4 0.35141967 0.14872462

C 113 0.33629430 0.16042367

RC 452 0.06043607 0.04403659

===========================================================================

\*\*\*\*\* Variance Components Estimates \*\*\*\*\*

===========================================================================

DBM variance component and covariance estimates

DBM Component Estimate

----------------------- ----------------

Var(R) 0.00153500

Var(C) 0.02724923

Var(T\*R) 0.00020040

Var(T\*C) 0.01197530

Var(R\*C) 0.01226473

Var(Error) 0.03997160

===========================================================================

\*\*\*\*\* Analysis 1: Random-readers and Random-cases \*\*\*\*\*

===========================================================================

(Results apply to the population of readers and cases)

a) Test for H0: Treatments have the same Wilcoxon figure of merit.

Source df Mean Square F value Pr > F

---------- ------ --------------- ------- -------

Treatment 1 0.54676344 4.46 0.0517

Error 15.26 0.12269397

Error term: MS(TR) + max[MS(TC) - MS(TRC), 0]

Conclusion: The Wilcoxon FOMs of treatments are not significantly different,

F(1,15.26) = 4.46, p = 0.0517.

b) 95% confidence intervals for treatment differences

Treatment Estimate StdErr df t Pr > t 95% CI

---------- ---------- -------- -------- ------- ------ ------- -------------------

0 - 1 -0.04380 0.02075 15.26 -2.11 0.0517 -0.08796 , 0.00036

H0: the two treatments are equal.

Error term: MS(TR) + max[MS(TC) - MS(TRC), 0]

c) 95% treatment confidence intervals based on reader x case ANOVAs

for each treatment (each analysis is based only on data for the

specified treatment

Treatment Area Std Error df 95% Confidence Interval

---------- ---------- ---------- ------- -------------------------

0 0.89703704 0.03317360 12.74 (0.82522360 , 0.96885048)

1 0.94083736 0.02156637 12.71 (0.89413783 , 0.98753689)

Error term: MS(R) + max[MS(C) - MS(RC), 0]

===========================================================================

\*\*\*\*\* Analysis 2: Fixed-readers and Random-cases \*\*\*\*\*

===========================================================================

(Results apply to the population of cases but only for the readers

used in this study)

a) Test for H0: Treatments have the same Wilcoxon figure of merit.

Source df Mean Square F value Pr > F

---------- ------ --------------- ------- -------

Treatment 1 0.54676344 5.48 0.0210

Error 113.00 0.09984808

Error term: MS(TC)

Conclusion: The Wilcoxon FOMs of treatments are not equal,

F(1,113.00) = 5.48, p = 0.0210.

b) 95% confidence intervals for treatment differences

Treatment Estimate StdErr df t Pr > t 95% CI

---------- ---------- -------- -------- ------- ------ ------- -------------------

0 - 1 -0.04380 0.01872 113.00 -2.34 0.0210 -0.08088 , -0.00672

H0: the two treatments are equal.

Error term: MS(TC)

c) 95% treatment confidence intervals based on reader x case ANOVAs

for each treatment (each analysis is based only on data for the

specified treatment

Treatment Area Std Error df 95% Confidence Interval

---------- ---------- ---------- ------- -------------------------

0 0.89703704 0.02428971 113.00 (0.84891474 , 0.94515933)

1 0.94083736 0.01677632 113.00 (0.90760044 , 0.97407428)

Error term: MS(C)

TREATMENT X CASE ANOVAs for each reader

Sum of Squares

Source df 0 1 2 3 4

------ --- ----------- ----------- ----------- ----------- -----------

T 1 0.0452655 0.1234489 0.0182112 0.0392705 0.5718372

C 113 13.8142273 24.0405509 15.4283316 1.9536987 18.7639677

TC 113 4.1922466 4.4557943 6.2738049 1.9257860 12.5023663

Mean Squares

Source df 0 1 2 3 4

------ --- ----------- ----------- ----------- ----------- -----------

T 1 0.0452655 0.1234489 0.0182112 0.0392705 0.5718372

C 113 0.1222498 0.2127482 0.1365339 0.0172894 0.1660528

TC 113 0.0370995 0.0394318 0.0555204 0.0170424 0.1106404

d) Treatment-by-case ANOVA CIs for each reader

(each analysis is based only on data for the specified reader)

Reader Treatment Estimate StdErr df t Pr > t 95% CI

---------- ---------- ---------- -------- -------- ------- ------ ------- -------------------

0 0 -1 -0.02818 0.02551 113.00 -1.10 0.2717 -0.07872 , 0.02236

1 0 -1 -0.04654 0.02630 113.00 -1.77 0.0795 -0.09865 , 0.00557

2 0 -1 -0.01787 0.03121 113.00 -0.57 0.5680 -0.07971 , 0.04396

3 0 -1 -0.02625 0.01729 113.00 -1.52 0.1318 -0.06051 , 0.00801

4 0 -1 -0.10016 0.04406 113.00 -2.27 0.0249 -0.18745 , -0.01288

===========================================================================

\*\*\*\*\* Analysis 3: Random-readers and Fixed-cases \*\*\*\*\*

===========================================================================

(Results apply to the population of readers but only for the cases used in this study)

a) Test for H0: Treatments have the same Wilcoxon figure of merit.

Source df Mean Square F value Pr > F

---------- ------ --------------- ------- -------

Treatment 1 0.54676344 8.70 0.0420

Error 4.00 0.06281749

Error term: MS(TR)

Conclusion: The Wilcoxon FOMs of treatments are not equal,

F(1,4.00) = 8.70, p = 0.0420.

b) 95% confidence intervals for treatment differences

Treatment Estimate StdErr df t Pr > t 95% CI

---------- ---------- -------- -------- ------- ------ ------- -------------------

0 - 1 -0.04380 0.01485 4.00 -2.95 0.0420 -0.08502 , -0.00258

H0: the two treatments are equal.

c) Reader-by-case ANOVAs for each treatment (each analysis is based only on data for the

specified treatment

Treatment Area Std Error df 95% Confidence Interval

---------- ---------- ---------- ------- -------------------------

0 0.89703704 0.02482994 4.00 (0.82809808 , 0.96597599)

1 0.94083736 0.01615303 4.00 (0.89598936 , 0.98568536)

# Online Appendix 9.D: Excel ANOVA Tables

Table 1: This shows the contents of the worksheet ANOVA in the Excel output file VanDykeOutput.xlsx.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FOM variance components** | |  |  |  |  |  |
| **Var(R)** | 0.00153 |  |  |  |  |  |
| **Var(C)** | 0.02725 |  |  |  |  |  |
| **Var(T\*R)** | 0.00020 |  |  |  |  |  |
| **Var(T\*C)** | 0.01198 |  |  |  |  |  |
| **Var(R\*C)** | 0.01226 |  |  |  |  |  |
| **Var(Error)** | 0.03997 |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **TREATMENT X READER X CASE ANOVA** | | | |  |  |  |
| **Source** | **SS** | **df** | **MS** |  |  |  |
| **T** | 0.54676 | 1 | 0.54676 |  |  |  |
| **R** | 1.74931 | 4 | 0.43733 |  |  |  |
| **C** | 44.84630 | 113 | 0.39687 |  |  |  |
| **TR** | 0.25127 | 4 | 0.06282 |  |  |  |
| **TC** | 11.28283 | 113 | 0.09985 |  |  |  |
| **RC** | 29.15448 | 452 | 0.06450 |  |  |  |
| **TRC** | 18.06716 | 452 | 0.03997 |  |  |  |
| **Total** | 105.89811 | 1139 |  |  |  |  |
|  |  |  |  |  |  |  |
| **READER X CASE ANOVA for each Trt** | | | |  |  |  |
| **Source** | **df** | **Trt - 1** | **Trt - 2** |  |  |  |
| **R** | 4.00000 | 0.35142 | 0.14872 |  |  |  |
| **C** | 113.00000 | 0.33629 | 0.16042 |  |  |  |
| **RC** | 452.00000 | 0.06044 | 0.04404 |  |  |  |
|  |  |  |  |  |  |  |
| **TREATMENT X CASE ANOVAs (MS) for each reader, assuming fixed-reader analysis** | | | | | | |
| **Source** | **df** | **Rdr - 1** | **Rdr - 2** | **Rdr - 3** | **Rdr - 4** | **Rdr - 5** |
| **T** | 1 | 0.04527 | 0.12345 | 0.01821 | 0.03927 | 0.57184 |
| **C** | 113 | 0.12225 | 0.21275 | 0.13653 | 0.01729 | 0.16605 |

The variance components, specifically the ones with "treatment" in their names, are needed for sample size estimation, book Chapter 11 The values listed under treatment x reader x case ANOVA allow one to test for interactions, see Online Section 9.H.

# Online Appendix 9.E: Code for Validating DBMH analysis

Here is the code, file mainRejectRate.R, which performs NH testing. The corresponding cluster version, not listed, is in file mainRejectRateC.R.

### Online Appendix 9.E.1: Code Listing

# mainRejectRate.R

# Simulating testing of analysis method

rm( list = ls( all = TRUE ) )

library(RJafroc)

source( 'RoeMetzZSamples.R' )

nomAlpha <- 0.05;seed <- 1;set.seed( seed )

I <- 2;J <- 5;K1 <- 52;K2 <- 50;VarStrString <- "LH1"

VarStr <- RoeMetzVarStr( VarStrString )

cat("Var Str = ", unlist(VarStr), "\n")

mu2 <- sqrt(2)\*qnorm(VarStr$auc);tau22 <- 0.0\*mu2 # zero multipler sets NH condition

isBinned <- TRUE;DesiredNumBins <- 5 # data binning, R = 5

S <- 2000;reject <- 0

for (s in 1 : S) {

# following line creates z-samples

zSamplesInit <- InitRandomSamples( I, J, K1, K2 ) # assuming all components of VarStr are unity

# following line includes the effect of the variance components and bins data, if needed

zSamplesRaw <- RoeMetzZSamples(

I, J, K1, K2, mu2, tau22, VarStr, zSamplesInit)

NL <- array(-Inf,dim=c(I,J,(K1+K2),1));NL[,,1:K1,1] <- zSamplesRaw$FP

LL <- array(-Inf,dim=c(I,J,K2,1));LL[,,1:K2,1] <- zSamplesRaw$TP

dataset <- ToRJafrocDataset(NL, LL, dataType = "ROC")

if (isBinned) {

dataset <- BinDataset(dataset,DesiredNumBins)

}

ret <- DBMHAnalysis(dataset, FOM = "Wilcoxon", option = "RRRC")

if (ret$pRRRC < nomAlpha) reject <- reject + 1

}

empAlpha <- reject/S

ciWidth <- qnorm(1-nomAlpha/2)\*sqrt(nomAlpha\*(1-nomAlpha)/S) # assuming true value is nomAlpha

CI <- c(nomAlpha-ciWidth, nomAlpha+ciWidth)

cat("NH rejection rate over", S, "simulations: \n")

cat("95% CI lower = ", CI[1] , ", empAlpha = ", empAlpha, ", 95% CI upper = ", CI[2], "\n")

if (!((CI[1] < nomAlpha) & (CI[2] > nomAlpha))) cat("failed NH test.\n")

The simulator is implemented in RoeMetzZSamples.R sourced at line 5. The reader should examine this file and be convinced that it is a straightforward application of book Equation (9.55) and (9.56). Line 8 specifies the size of the datasets that will be generated for each trial, namely two treatments, five readers, 52 non-diseased and 50 diseased cases[[1]](#footnote-1). At line 8 the variance structure "LH1" is specified for the simulator (the notation follows that in the original Roe-Metz publication[6](#_ENREF_6)). Line 9 obtains the variance structure using the function RoeMetzVarStr() which is part of the RoeMetzZSamples.R file. Line 10 prints out the values of the variance components, in the order , , , ,  and , followed by AUC and the name of the simulator. Note the usage of the unlist() function to assure everything prints out in one line, specifically, the printout is Var Str = 0.011, 0.011, 0.1, 0.1, 0.2, 0.6, 0.702, LH1. Commas have been inserted for readability (they are not present in the code output). The first 6 values are the z-sample variance components, followed by AUC = 0.702 and the name of the simulator, LH1.

Line 11 defines the fixed effects in the simulator; AUC is converted to a separation parameter and tau22 is set to zero (by the zero multiplier, which assures the null hypothesis condition; if the multiplier is non-zero one is simulating the alternative hypothesis). The notation corresponds closely to that for the fixed effects in book Equation 9.55.

Line 12 defines isBinned = TRUE, which ensures that the data is binned into DesiredNumBins = 5 bins at line 24. Line 13 sets the simulator to conduct 2000 simulations and initializes reject, the number of rejects (i.e., NH rejections) to zero. Lines 14 – 28 conduct the 2000 simulations inside a for-loop. Line 16 calls InitRandomSamples() to initialize all requisite samples in book Equation (9.55) with samples from unit normal distributions. The multiplication by the correct variances is done in line 18-19, using function RoeMetzZSamples(). At this point one has the raw z-samples, denoted zSamplesRaw. Lines 20 -22 constructs an RJafroc dataset object using the simulated data, which permits ready application of package functions: a binning function at line 24 and the DBMH analysis function at line 26. Line 27 examines the p-value and if smaller than nomAlpha, for nominal alpha[[2]](#footnote-2), increments the reject counter variable by one. After 2000 simulations one arrives at line 29, which calculates empirical alpha, called empAlpha. Line 30 calculates the 95% confidence interval and the rest contain print statements.

This code takes a while to execute. To see what is going on the reader should reduce the number of simulations S to 2 and step through the code by clicking on Run. When done, reset S to 2000 and click Source.

### Online Appendix 9.E.2: Code Output

> source(...)

RMVarStr = 0.011 0.011 0.1 0.1 0.2 0.6 0.702 LH1

NH rejection rate over 2000 simulations:

95% CI lower = 0.04044832

empAlpha = 0.054

95% CI upper = 0.05955168

The method passes the null hypothesis validity test for this simulator. Empirical alpha, 0.054, is close to the nominal value, and the confidence interval (0.04, 0.06) includes the nominal value.

A faster version of the code that uses cluster computing, is in file mainRejectRateC.R. This code is harder to run in "slow-motion" to see what is going on.

# Online Appendix 9.F: Simulator for validating fixed-reader and fixed-case analyses

## Online Appendix 9.F.1: Fixed reader simulator

To simulate Z-samples for fixed-reader one sets= = 0 which does not affect the normalization requirement + +  +  = 1. The simulation model is:

. .

## Online Appendix 9.F.2: Fixed case simulator

To generate data for fixed cases one sets == 0. However, since + < 1, to restore normalization the variances  and  implicit in Eqn. need to be inflated by dividing by + , resulting in new variances  and  defined by

. .

Note that  +  = 1 ensuring that the simulator is properly normalized. The simulation model is:

. .

where

. .

# Online Appendix 9.G: Code illustrating the meaning of pseudovalues

This relates to book Section 9.13. The relevant file is mainMeaningPseudoValues.R.

rm(list = ls()) # mainMeaningPseudoValues.R

library(RJafroc)

options(digits = 6)

if (TRUE) {

#fileName <- "Franken1.lrc"

fileName <- "VanDyke.lrc"

rocData <- ReadDataFile(fileName, format = "MRMC", renumber = "TRUE")

} else {

fileName <- "CXRinvisible3-20mm.xlsx"

frocData <- ReadDataFile(fileName, format = "JAFROC", renumber = "TRUE")

rocData <- FROC2HrROC(frocData)

rm(frocData)

}

K <- dim(rocData$NL)[3];K2 <- dim(rocData$LL)[3];K1 <- K-K2

selectTreatment <- 1

selectReader <- 1

cat("data file = ", fileName, "\n")

cat("selected Treatment = ", selectTreatment, ", selected Reader = ", selectReader, ", number of non-diseased cases = ", K1,

", number of diseased cases = = ", K2, "\n")

rocData <- ExtractDataset(rocData, trts=selectTreatment, rdrs = selectReader)

FOM <- FigureOfMerit(rocData,FOM = "Wilcoxon")

zk1 <- rocData$NL[1,1,1:K1,1];zk2 <- rocData$LL[1,1,1:K2,1]

pseudoVals <- PseudoValues(rocData, FOM = "Wilcoxon")

pseudoVals <- pseudoVals[1,1,]

psdVlsNor <- pseudoVals[1:K1]

psdVlsAbn <- pseudoVals[(K1+1):(K1+K2)]

print(table(psdVlsNor))

print(table(zk1))

cat("\n")

print(table(psdVlsAbn))

print(table(zk2))

Line 16 defines the selected treatment to one and selected reader to one. Line 23 extracts the data for the selected treatment and reader and the next line calculates the FOMs. Line 25 extracts the two ratings arrays, zk1 and zk2, for non-diseased and diseased cases, respectively. Line 26 calculates the pseudovalues for the single-treatment single-reader dataset object rocData, using the Wilcoxon figure of merit. Line 27 removes unnecessary indices of unit length, and lines 28 and 29 extracts the pseudovalues for non-diseased and diseased cases to two arrays, psdVlsNor and psdVlsAbn, respectively. The table() function used in Line 30 is quite useful. It *orders the pseudovalues into unique values and computes the number of occurrences of each unique value*. Line 30 prints this out for *non-diseased case pseudovalues* for while Line 31 does the same for non-diseased cases. Lines 33 and 34 do the same for diseased case pseudovalues and ratings, respectively.

For this dataset there is one situation where the correspondences between the number of occurrences of pseudovalues and ratings appears to be violated. For treatment 2 and reader 5, Table 2, the number of unique pseudovalues and unique ratings are different. However, if one cumulates the two highest ratings counts, the direct correspondence is true. What actually happened in this situation is that the number of unique pseudovalues does not equal the number of ratings bins. For non-diseased cases the reader used only the 3 lowest ratings bins while for diseased cases all 5 ratings bins were used.

Table 2: Data for treatment 2 and reader 5 for *diseased cases* of the Van Dyke data set: ordered unique pseudovalues, row 1, the number of occurrences of each distinct pseudovalue, row 2, and corresponding values for ratings, rows 3 and 4. Note the direct correspondence between the ordering of the ratings and the ordering of the pseudovalues.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pseudovalues | -1.068 | 0.049 | 0.942 | 1.110 | NA | 0.919646 |
| # occurrences | 1 | 4 | 10 | 30 | NA |
| Ratings | 1 | 2 | 3 | 4 | 5 | 4.2667 |
| # occurrences | 1 | 4 | 10 | 4 | 26 |

# Online Appendix 9.H: Testing for interactions

In book Section 9.5 it was stated that between-reader variance could be treatment dependent, i.e., the readers interpretations might exhibit greater variability in one modality than in the other. For the between-reader variance to depend on treatment, then, with reference to book Equation 9.4, the variance  must be significantly greater than zero. The actual value of  = 0.00020, is quite small (ANOVA table in Excel output file, Table 1). To determine if this value is significantly different from zero, consider the F-ratio  =  (also contained in cited ANOVA table). The observed value comes from an F‑distribution with *ndf* = 4 and *ddf* = 452 (see ANOVA worksheet rows 14 and 17). Therefore, the p-value for testing for a treatment-reader effect is:

 .

Calculating this in R, one has 1 ‑ pf(0.06282/0.03997,4,452) = 0.1806607. As the p-value exceeds 0.05, the treatment-reader effect is not significant at alpha = 0.05, consistent with the rather small estimate for . A similar method can be used to test for a treatment-case effect. From the ANOVA output,  = 0.01198, which is much larger than . The relevant F-ratio is  and the p-value is calculated using:

 .

The treatment-case term is significantly different from zero. It is shown in book Section 9.13 that the pseudovalues on the left hand side of book Equation 9.4 are interpretable as the FOMs of individual cases. With this interpretation in mind,  is the case-variance of the FOM for each treatment, averaged over treatments, and  indicates how much the case variance depends on treatment. It is possible for the same case-set to appear more homogenous in one treatment and more heterogeneous in the other treatment, as with this dataset, with treatment #1 having larger variability (0.33629 > 0.16042, ANOVA sheet row 23).

# References

1. Satterthwaite FE. Synthesis of variance. *Psychometrika.* 1941 6(5):309–316.

2. Satterthwaite FE. An Approximate Distribution of Estimates of Variance Components. *Biometrics Bulletin.* 1946;2(6):110–114.

3. Larsen RJ, Marx ML. *An Introduction to Mathematical Statistics and Its Applications.* 3rd ed. Upper Saddle River, NJ: Prentice-Hall Inc; 2001.

4. Field A. *Discovering statistics using SPSS.* Sage publications; 2009.

5. Van Dyke CW, White RD, Obuchowski NA, Geisinger MA, Lorig RJ, Meziane MA. Cine MRI in the diagnosis of thoracic aortic dissection. *79th RSNA Meetings.* 1993.

6. Roe CA, Metz CE. Dorfman-Berbaum-Metz Method for Statistical Analysis of Multireader, Multimodality Receiver Operating Characteristic Data: Validation with Computer Simulation. *Acad Radiol.* 1997;4:298-303.

1. The reason for the slightly unequal numbers of cases is to be assured that array dimensioning is working as expected; it is the author's programming style. [↑](#footnote-ref-1)
2. Also termed the size of the test. [↑](#footnote-ref-2)