RJafroc Documentation

Dev P. Chakraborty, PhD

2020-03-17

Contents

Preface

- This book, an extended documentation of the **RJafroc** package, is undergoing extensive edits.
- It should not be used by the casual user until I give the go ahead.
- It bypasses the file size limits of **CRAN**, currently 5 MB, which severely limits the extent of the documentation that can be included with the CRAN version of the package.
- I welcome corrections and comments by the not-so-casual-user.
- Please use the GitHub website to raise issues and comments:
  - https://github.com/dpc10ster/RJafrocBook

Chapter 1

Introduction

- This is the book desribing the **RJafroc** package.
- The name of the book is RJafrocBook
- Modality and treatment are used interchangeably.
- Reader is a generic radiologist, or a computer aided detection algorithm, or any algorithmic "reader"
- TBA

1.1 References

Chapter 2

ROC DATA FORMAT

$$\theta = \frac{1}{N_L N_N} \sum\nolimits_k \sum\nolimits_{k'} \sum_{r=1}^{n_k^L} \sum_{r'=1}^{n_{k'}^N} \psi(X_{kr}, Y_{k'r'})$$

$$\frac{d}{dx}\left(\int_a^x f(u)\,du\right) = f(x)$$

$$\theta = \frac{1}{N_L N_N}$$

2.1 Introduction

- The purpose of this vignette is to explain the data format of the input Excel file and to introduce the capabilities of the function `DfReadDataFile()`. Background on observer performance methods are in my book (Chakraborty, 2017).
- I will start with Receiver Operating Characteristic (ROC) data (Metz, 1978), as this is by far the simplest paradigm.
- In the ROC paradigm the observer assigns a rating to each image. A rating is an ordered numeric label, and, in our convention, higher values represent greater certainty or **confidence level** for presence of disease. With human observers, a 5 (or 6) point rating scale is typically used, with 1 representing highest confidence for *absence* of disease and 5 (or 6) representing highest confidence for *presence* of disease. Intermediate values represent intermediate confidence levels for presence or absence of disease.
- Note that location information associated with the disease, if applicable, is not collected.

- There is no restriction to 5 or 6 ratings. With algorithmic observers, e.g., computer aided detection (CAD) algorithms, the rating could be a floating point number and have infinite precision. All that is required is that higher values correspond to greater confidence in presence of disease.

2.2 Note to existing users

- The Excel file format has recently undergone changes resulting in 4 extra `list` members in the final created `dataset` object (i.e., 12 members instead of 8).
- Code should run on the old format Excel files as the 4 extra list members are simply ignored.
- Reasons for the change will become clearer in these vignettes
- Basically they are needed for generalization to other data collection paradigms instead of crossed, for example to the split-plot data acquisition paradigm, and for better data entry error control.

2.3 The Excel data format

- The Excel file has three worksheets.
- These are named
  - `Truth`,
  - `NL` (or `FP`),
  - `LL` (or `TP`).

2.4 Illustrative toy file

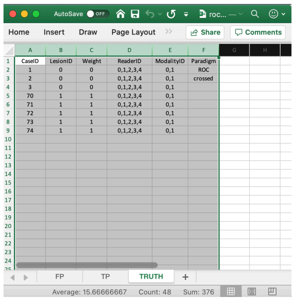
- *Toy files* are artificial small datasets intended to illustrate essential features of the data format.

- The examples shown in this vignette corresponds to Excel file `inst/extdata/toyFiles/ROC/rocCr.xlsx` in the project directory.
- To view these files one needs to `clone` the source files from `GitHub`.

2.5 The `Truth` worksheet

- The `Truth` worksheet contains 6 columns: `CaseID`, `LesionID`, `Weight`, `ReaderID`, `ModalityID` and `Paradigm`.

- For ROC data the first five columns contain as many rows as there are cases (images) in the dataset.
- CaseID: unique integers, one per case, representing the cases in the dataset.
- LesionID: integers 0 or 1, with each 0 representing a non-diseased case and each 1 representing a diseased case.
- In the current toy dataset, the non-diseased cases are labeled 1, 2 and 3, while the diseased cases are labeled 70, 71, 72, 73 and 74. The values do not have to be consecutive integers; they need not be ordered; the only requirement is that they be **unique**.
- Weight: Not used for ROC data, a floating point value, typically filled in with 0 or 1.
- ReaderID: a **comma-separated** listing of reader labels, each represented by a **unique string**, that have interpreted the case. In the example shown below each cell has the value 0, 1, 2, 3, 4 meaning that each of the readers, represented by the strings "0", "1", "2", "3" and "4", have interpreted all cases (hence the "crossed" design). **With reader names that could be confused with integers, each cell in this column has to be text formatted as otherwise Excel will not accept it.** [Try entering 0, 1, 2, 3, 4 in a numeric formatted Excel cell.]
- The reader names could just as well have been Rdr0, Rdr1, Rdr2, Rdr3, Rdr4. The only requirement is that they be unique strings.
- Look in in the inst/extdata/toyFiles/ROC directory for files rocCrStrRdrsTrts.xlsx and rocCrStrRdrsNonUnique.xlsx for examples of data files using longer strings for readers. The second file generates an error because the reader names are not unique.
- ModalityID: a comma-separated listing of modalities (one or more modalities), each represented by a **unique string**, that are applied to each case. In the example each cell has the value "0", "1". **With treatment names that could be confused with integers, each cell has to be text formatted as otherwise Excel will not accept it.**
- The treatment names could just as well have been Trt0, Trt1. Again, the only requirement is that they be unique strings.
- Paradigm: this column contains two cells, ROC and crossed. It informs the software that this is an ROC dataset, and the design is crossed, meaning each reader has interpreted each case in each modality (in statistical terminology: modality and reader factors are "crossed").
- There are 5 diseased cases in the dataset (the number of 1's in the LesionID column of the Truth worksheet).
- There are 3 non-diseased cases in the dataset (the number of 0's in the LesionID column).
- There are 5 readers in the dataset (each cell in the ReaderID column contains the string 0, 1, 2, 3, 4).
- There are 2 modalities in the dataset (each cell in the ModalityID column contains the string 0, 1).



CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm
1	0	0	0,1,2,3,4	0,1	ROC
2	0	0	0,1,2,3,4	0,1	crossed
3	0	0	0,1,2,3,4	0,1	
70	1	1	0,1,2,3,4	0,1	
71	1	1	0,1,2,3,4	0,1	
72	1	1	0,1,2,3,4	0,1	
73	1	1	0,1,2,3,4	0,1	
74	1	1	0,1,2,3,4	0,1	

Figure 2.1: Truth worksheet for file rocCr.xlsx

2.6 The structure of an ROC dataset

In the following code chunk the first statement retrieves the name of the data file, located in a hidden directory that one need not be concerned with. The second statement reads the file using the function `DfReadDataFile()` and saves it to object `x`. The third statement shows the structure of the dataset object `x`.

```r
rocCr <- system.file("extdata", "toyFiles/ROC/rocCr.xlsx",
                        package = "RJafroc", mustWork = TRUE)
x <- DfReadDataFile(rocCr, newExcelFileFormat = TRUE)
str(x)
#> List of 12
#>  $ NL           : num [1:2, 1:5, 1:8, 1] 1 3 2 3 2 2 1 2 3 2 ...
#>  $ LL           : num [1:2, 1:5, 1:5, 1] 5 5 5 5 5 5 5 5 5 5 ...
#>  $ lesionVector : int [1:5] 1 1 1 1 1
#>  $ lesionID     : num [1:5, 1] 1 1 1 1 1
#>  $ lesionWeight : num [1:5, 1] 1 1 1 1 1
#>  $ dataType     : chr "ROC"
#>  $ modalityID   : Named chr [1:2] "0" "1"
#>   ..- attr(*, "names")= chr [1:2] "0" "1"
#>  $ readerID     : Named chr [1:5] "0" "1" "2" "3" ...
#>   ..- attr(*, "names")= chr [1:5] "0" "1" "2" "3" ...
#>  $ design       : chr "CROSSED"
#>  $ normalCases  : int [1:3] 1 2 3
#>  $ abnormalCases: int [1:5] 70 71 72 73 74
#>  $ truthTableStr: num [1:2, 1:5, 1:8, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
```

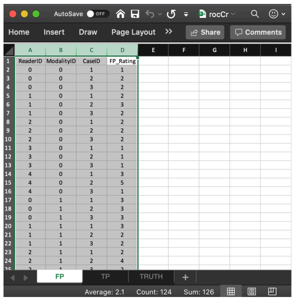
- In the above code chunk flag `newExcelFileFormat` is set to `TRUE` as otherwise columns D - F in the `Truth` worksheet are ignored and the dataset is assumed to be crossed, with `dataType` automatically determined from the contents of the FP and TP worksheets.
- Flag `newExcelFileFormat = FALSE` is for compatibility with older

JAFROC format Excel files, which did not have these columns in the Truth worksheet. Its usage is deprecated.

- The dataset object x is a list variable with 12 members.
- The x$NL member, with dimension [2, 5, 8, 1], contains the ratings of normal cases. The extra values in the third dimension, filled with NAs, are needed for compatibility with FROC datasets, as unlike ROC, false positives are possible on diseased cases.
- The x$LL, with dimension [2, 5, 5, 1], contains the ratings of abnormal cases.
- The x$lesionVector member is a vector with 5 ones representing the 5 diseased cases in the dataset.
- The x$lesionID member is an array with 5 ones.
- The x$lesionWeight member is an array with 5 ones.
- The lesionVector, lesionID and lesionWeight members are not used for ROC datasets. They are there for compatibility with FROC datasets.
- The dataType member indicates that this is an ROC dataset.
- The x$modalityID member is a vector with two elements "0" and "1", naming the two modalities.
- The x$readerID member is a vector with five elements "0", "1", "2", "3" and "4", naming the five readers.
- The x$design member is CROSSED; specifies the dataset design, which is "CROSSED".
- The x$normalCases member lists the integer names of the normal cases, 1, 2, 3.
- The x$abnormalCases member lists the integer names of the abnormal cases, 70, 71, 72, 73, 74.
- The x$truthTableStr member quantifies the structure of the dataset, as explained in **Chapter 00 Vignette #3-#5**.

2.7 The false positive (FP) ratings

These are found in the FP or NL worksheet, see below.



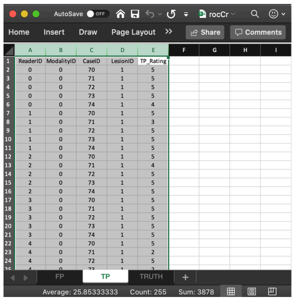
	A	B	C	D
1	ReaderID	ModalityID	CaseID	FP_Rating
2	0	0	1	1
3	0	0	2	2
4	0	0	3	2
5	1	0	1	2
6	1	0	2	3
7	1	0	3	2
8	2	0	1	2
9	2	0	2	2
10	2	0	3	2
11	3	0	1	1
12	3	0	2	1
13	3	0	3	1
14	4	0	1	3
15	4	0	2	5
16	4	0	3	1
17	0	1	1	3
18	0	1	2	3
19	0	1	3	3
20	1	1	1	3
21	1	1	2	2
22	1	1	3	2
23	2	1	1	2
24	2	1	2	4
25	2	1	3	2

Figure 2.2: FP worksheet for file rocCr.xlsx

- It consists of 4 columns, each of length 30 (= # of modalities times number of readers times number of non-diseased cases).
- `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 6 times (= # of modalities times number of non-diseased cases).
- `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 15 times (= # of readers times number of non-diseased cases).
- `CaseID`: the case labels for non-diseased cases: `1`, `2` and `3`. Each label occurs 10 times (= # of modalities times # of readers).
- The label of a diseased case cannot occur in the FP worksheet. If it does the software generates an error.
- `FP_Rating`: the floating point ratings of non-diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID` and `CaseID` for that row.

2.8 The true positive (TP) ratings

These are found in the `TP` or `LL` worksheet, see below.



	A	B	C	D	E
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating
2	0	0	70	1	5
3	0	0	71	1	5
4	0	0	72	1	5
5	0	0	73	1	5
6	0	0	74	1	4
7	1	0	70	1	5
8	1	0	71	1	3
9	1	0	72	1	5
10	1	0	73	1	5
11	1	0	74	1	5
12	2	0	70	1	5
13	2	0	71	1	4
14	2	0	72	1	5
15	2	0	73	1	5
16	2	0	74	1	5
17	3	0	70	1	5
18	3	0	71	1	5
19	3	0	72	1	5
20	3	0	73	1	5
21	3	0	74	1	5
22	4	0	70	1	5
23	4	0	71	1	2
24	4	0	72	1	5
25	4	0	73	1	2

Average: 25.85333333 Count: 255 Sum: 3878

Figure 2.3: TP worksheet for file rocCr.xlsx

- It consists of 5 columns, each of length 50 (= # of modalities times number of readers times number of diseased cases).
- `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 10 times (= # of modalities times number of diseased cases).
- `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 25 times (= # of readers times number of diseased cases).
- `LesionID`: For an ROC dataset this column contains fifty 1's (each diseased case has one lesion).
- `CaseID`: the case labels for non-diseased cases: `70`, `71`, `72`, `73` and `74`. Each label occurs 10 times (= # of modalities times # of readers). The label of a non-diseased case cannot occur in the TP worksheet.
- `TP_Rating`: the floating point ratings of diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID`, `LesionID` and `CaseID` for that row.

2.9 Correspondence between `NL` member of dataset and the `FP` worksheet

- The list member `x$NL` is an array with `dim = c(2,5,8,1)`.
  - The first dimension (2) comes from the number of modalities.
  - The second dimension (5) comes from the number of readers.
  - The third dimension (8) comes from the **total** number of cases.
  - The fourth dimension is alway 1 for an ROC dataset.
- The value of `x$NL[1,5,2,1]`, i.e., 5, corresponds to row 15 of the FP table, i.e., to `ModalityID` = 0, `ReaderID` = 4 and `CaseID` = 2.
- The value of `x$NL[2,3,2,1]`, i.e., 4, corresponds to row 24 of the FP table, i.e., to `ModalityID` 1, `ReaderID` 2 and `CaseID` 2.
- All values for case index > 3 are `-Inf`. For example the value of `x$NL[2,3,4,1]` is `-Inf`. This is because there are only 3 non-diseased cases. The extra length is needed for compatibility with FROC datasets.

2.10 Correspondence between `LL` member of dataset and the `TP` worksheet

- The list member `x$LL` is an array with `dim = c(2,5,5,1)`.
  - The first dimension (2) comes from the number of modalities.
  - The second dimension (5) comes from the number of readers.
  - The third dimension (5) comes from the number of diseased cases.
  - The fourth dimension is alway 1 for an ROC dataset.
- The value of `x$LL[1,1,5,1]`, i.e., 4, corresponds to row 6 of the TP table, i.e., to `ModalityID` = 0, `ReaderID` = 0 and `CaseID` = 74.
- The value of `x$LL[1,2,2,1]`, i.e., 3, corresponds to row 8 of the TP table, i.e., to `ModalityID` = 0, `ReaderID` = 1 and `CaseID` = 71.
- There are no -Inf values in `x$LL`: `any(x$LL == -Inf)` = FALSE.

2.11 Correspondence using the `which` function

- Converting from **names** to **subscripts** (indicating position in an array) can be confusing.
- The following example uses the `which` function to help out.
- The first line says that the `abnormalCase` named 70 corresponds to subscript 1 in the LL array case dimension.
- The second line prints the NL rating for `modalityID` = 0, `readerID` = 1 and `normalCases` = 1.

- The third line prints the LL rating for `modalityID` = 0, `readerID` = 1 and `abnormalCases` = 70.
- The last line shows what happens if one enters an invalid value for name; the result is a `numeric(0)`.
- Note that in each of these examples, the last dimension is 1 because we are dealing with an ROC dataset.
- The reader is encouraged to examine the correspondence between the NL and LL ratings and the Excel file using this method.

```
which(x$abnormalCases == 70)
#> [1] 1
x$NL[which(x$modalityID == "0"),which(x$readerID == "1"),which(x$normalCases == 1),1]
#> [1] 2
x$LL[which(x$modalityID == "0"),which(x$readerID == "1"),which(x$abnormalCases == 70),1
#> [1] 5
x$LL[which(x$modalityID == "a"),which(x$readerID == "1"),which(x$abnormalCases == 70),1
#> numeric(0)
```

2.12 References

Chapter 3

BACKGROUND ON THE F-DISTRIBUTION

3.1 Introduction

Since it plays an important role in sample size estimation, it is helpful to examine the behavior of the F-distribution. In the following `ndf` = numerator degrees of freedom, `ddf` = denominator degrees of freedom and `ncp` = non-centrality parameter (i.e., the Δ appearing in Eqn. (11.6) of (Chakraborty, 2017)).

The use of three `R` functions is demonstrated.

- `qf(p,ndf,ddf)` is the *quantile* function of the F-distribution for specified values of `p`, `ndf` and `ddf`, i.e., the value `x` such that fraction `p` of the area under the F-distribution lies to the right of `x`. Since `ncp` is not included as a parameter, the default value, i.e., zero, is used. This is called the *central* F-distribution.

- `df(x,ndf,ddf,ncp)` is the probability density function (*pdf*) of the F-distribution, as a function of `x`, for specified values of `ndf`, `ddf` and `ncp`.

- `pf(x,ndf,ddf,ncp)` is the probability (or cumulative) distribution function of the F-distribution for specified values of `ndf`, `ddf` and `ncp`.

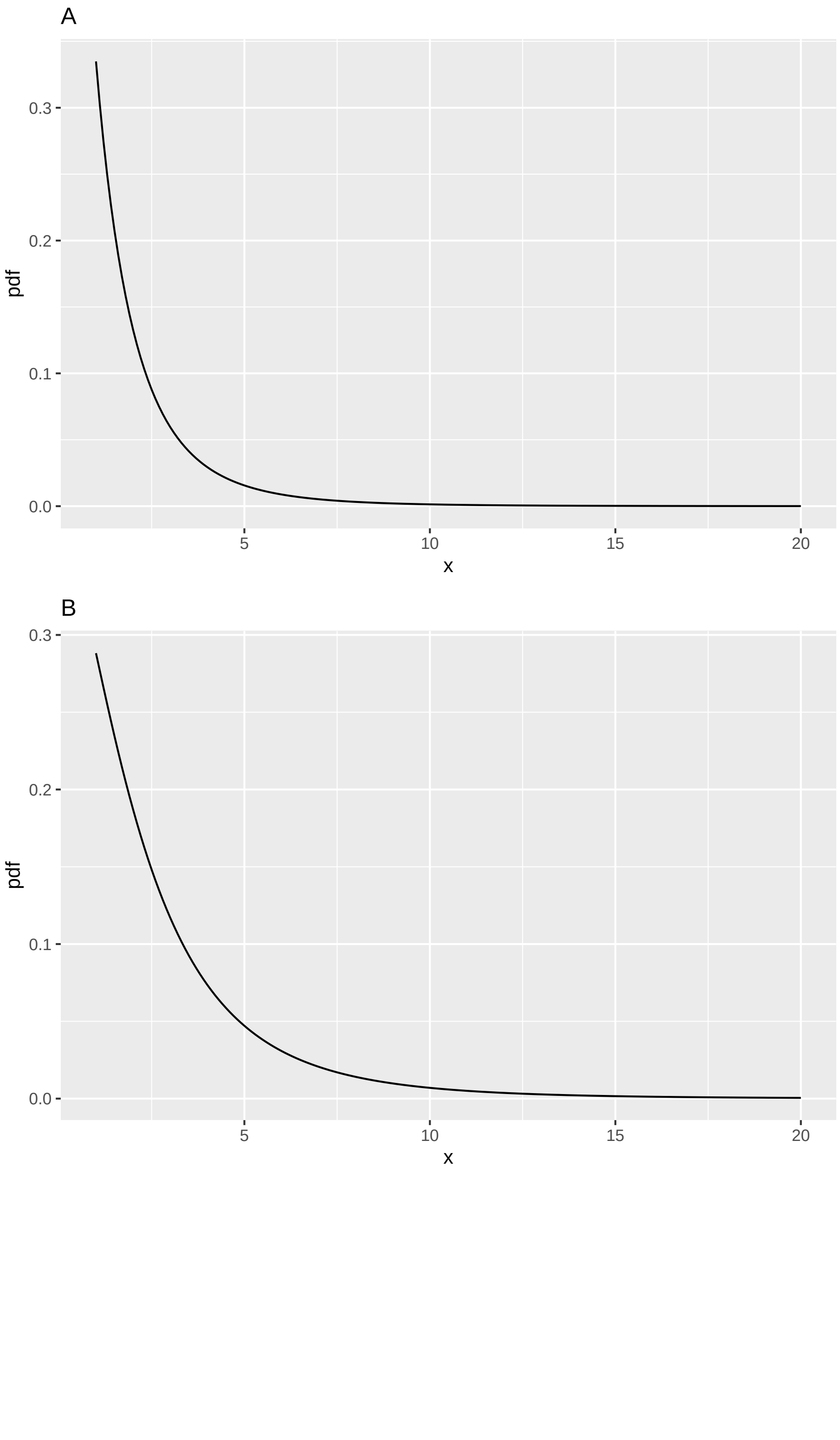
3.2 Effect of `ncp` for `ndf` = 2 and `ddf` = 10

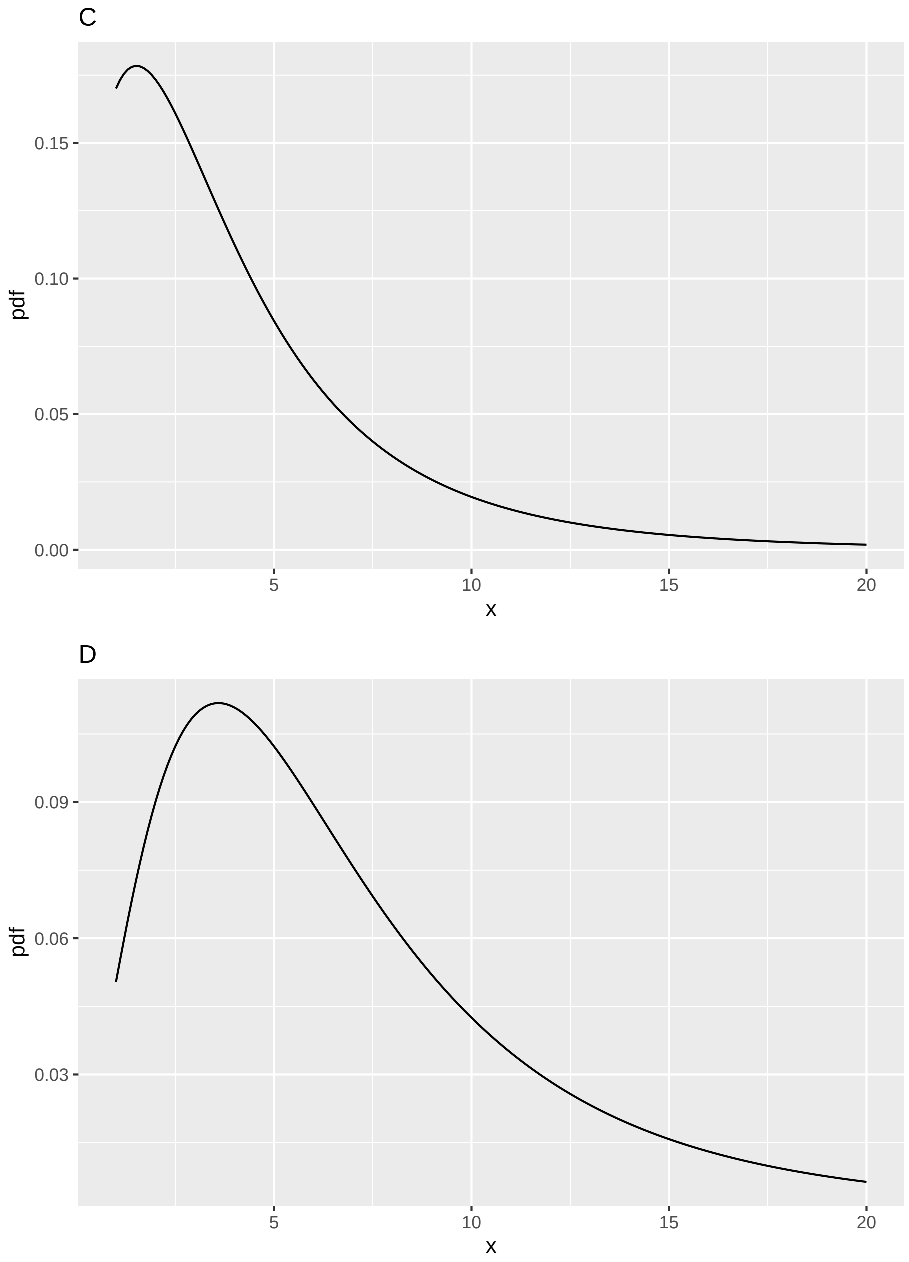
- Four values of `ncp` are considered (0, 2, 5, 10) for `ddf` = 10.

- fCrit is the critical value of the F distribution, i.e., that value such that fraction α of the area is to the right of the critical value, i.e., fCrit is identical to:

$$F_{1-\alpha,ndf,ddf}$$

```
ndf <- 2;ddf <- 10;ncp <- c(0,2,5,10)
alpha <- 0.05
fCrit <- qf(1-alpha, ndf,ddf)
x <- seq(1, 20, 0.1)
myLabel <- c("A", "B", "C", "D")
myLabelIndx <- 1
pFgtFCrit <- NULL
for (i in 1:length(ncp))
{
  y <- df(x,ndf,ddf,ncp=ncp[i])
  pFgtFCrit <- c(pFgtFCrit, 1-pf(fCrit, ndf, ddf, ncp = ncp[i]))
}
for (i in 1:length(ncp))
{
  y <- df(x,ndf,ddf,ncp=ncp[i])
  curveData <- data.frame(x = x, pdf = y)
  curvePlot <- ggplot(data = curveData, mapping = aes(x = x, y = pdf)) +
    geom_line() +
    ggtitle(myLabel[myLabelIndx]);myLabelIndx <- myLabelIndx + 1
  print(curvePlot)
}
fCrit_2_10 <- fCrit # convention fCrit_ndf_ddf
```


A
0.3
0.2
0.1
0.0
pdf
5
10
15
20
x
B
0.3
0.2
0.1
0.0
pdf
5
10
15
20
x



	ndf	ddf	fCrit	ncp	pFgtFCrit
A	2	10	4.102821	0	0.0500000
B	2	10	4.102821	2	0.1775840
C	2	10	4.102821	5	0.3876841
D	2	10	4.102821	10	0.6769776

3.3 Comments

3.3.1 Fig. A

- This corresponds to `ncp = 0`, i.e., the *central* F-distribution.
- The integral under this distribution is unity (this is also true for all plots in this vignette).
- The critical value, `fCrit` in the above code block, is the value of `x` such that the probability of exceeding `x` is α. The corresponding parameter `alpha` is defined above as 0.05.
- In the current example `fCrit` = 4.102821. Notice the use of the quantile function `qf()` to determine this value, and the default value of `ncp`, namely zero, is used; specifically, one does not pass a 4th argument to `qf()`.
- **The decision rule for rejecting the NH uses the NH distribution of the F-statistic**, i.e., reject the NH if F >= `fCrit`. As expected, `prob > fCrit` = 0.05 because this is how `fCrit` was defined.

3.3.2 Fig. B

- This corresponds to `ncp = 2`, `ndf` = 2 and `ddf` = 10.
- The distribution is slightly shifted to the right as compared to Fig. A, thereby making it more likely that the observed value of the F-statistic will exceed the critical value determined for the NH distribution.
- In fact, `prob > fCrit` = 0.177584, i.e., the *statistical power* (compare this to Fig. A where `prob > fCrit` was 0.05).

3.3.3 Fig. C

- This corresponds to `ncp = 5`, `ndf` = 2 and `ddf` = 10.
- Now `prob > fCrit` = 0.3876841.
- Power has increased compared to Fig. B.

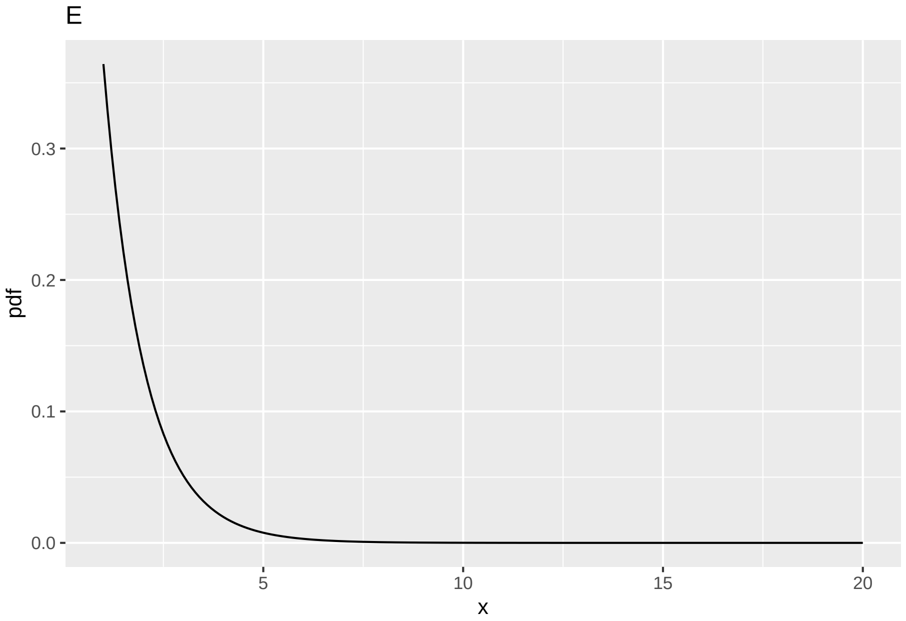
3.3.4 Fig. D

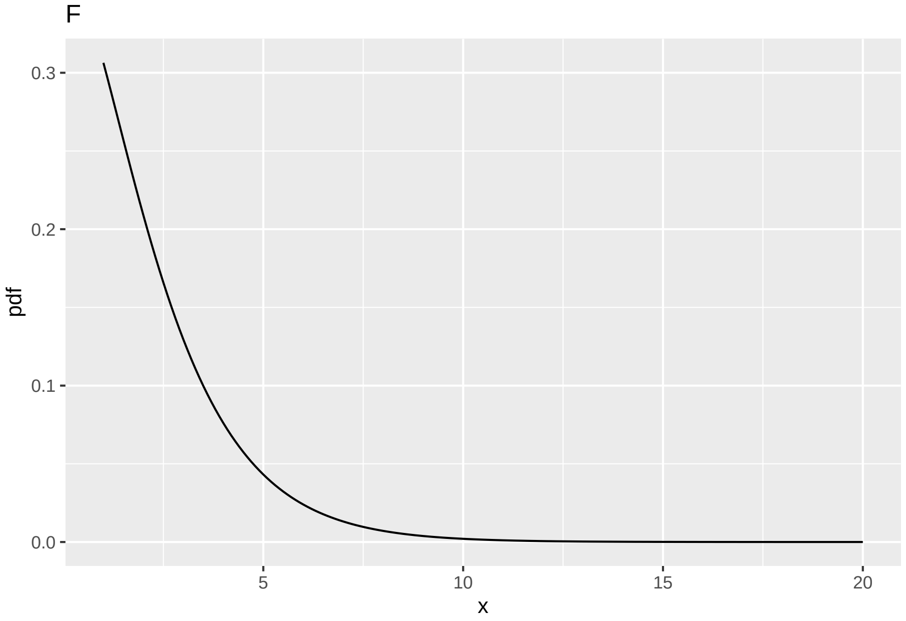
- This corresponds to `ncp = 10`, `ndf` = 2 and `ddf` = 10.
- Now `prob > fCrit` is 0.6769776.
- Power has increased compared to Fig. C.
- The effect of the shift is most obvious in Fig. C and Fig. D.
- Considering a vertical line at `x` = 4.102821, fraction 0.6769776 of the probability distribution in Fig. D lies to the right of this line
- Therefore the NH is likely to be rejected with probability 0.6769776.

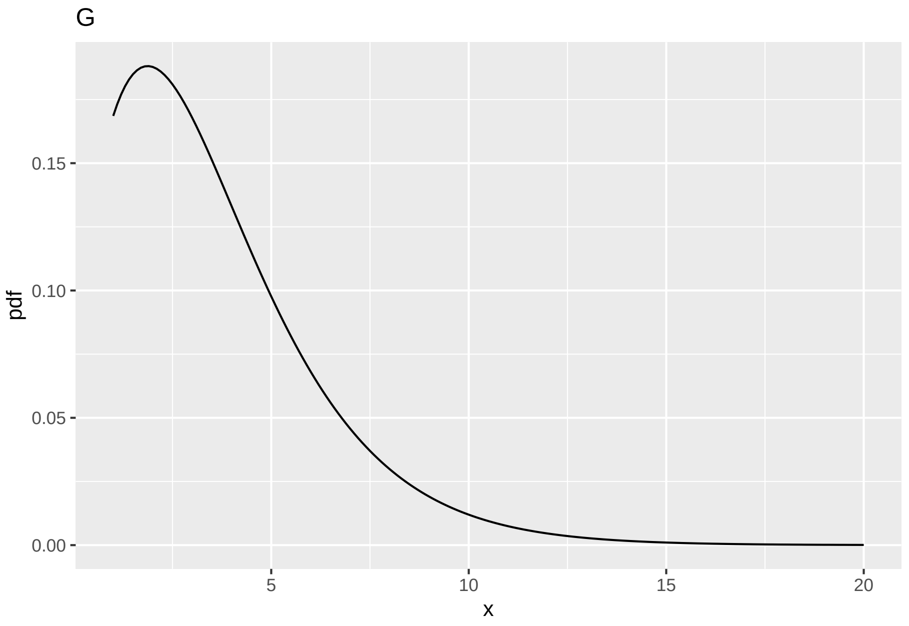
3.3.5 Summary

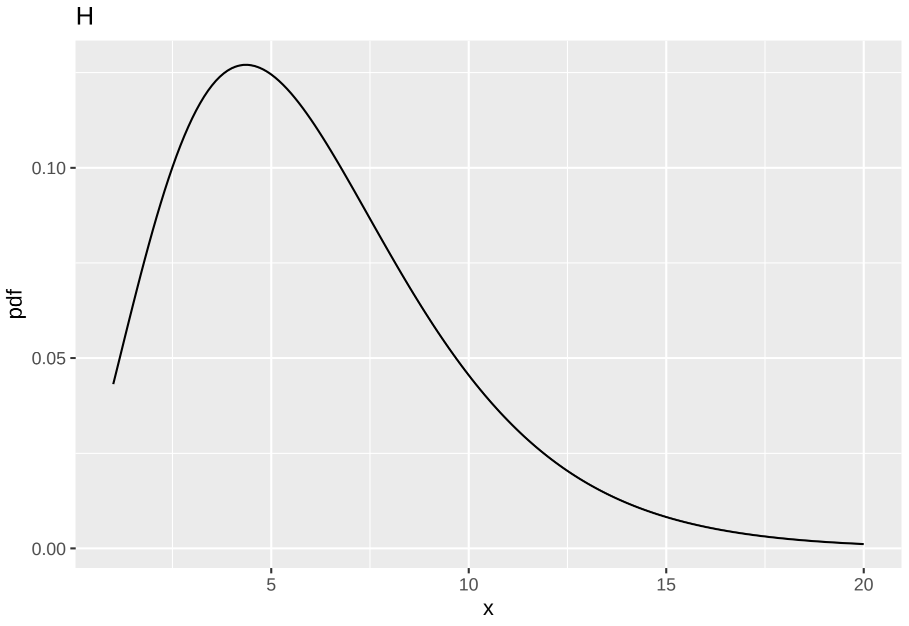
The larger that non-centrality parameter, the greater the shift to the right of the F-distribution, and the greater the statistical power.

3.4 Effect of `ncp` for `ndf` = 2 and `ddf` = 100






G
pdf
0.15
0.10
0.05
0.00
5
10
15
20
x


H
pdf
0.10
0.05
0.00
5
10
15
20
x

	ndf	ddf	fCrit	ncp	pFgtFCrit
A	2	10	4.102821	0	0.0500000
B	2	10	4.102821	2	0.1775840
C	2	10	4.102821	5	0.3876841
D	2	10	4.102821	10	0.6769776
E	2	100	3.087296	0	0.0500000
F	2	100	3.087296	2	0.2199264
G	2	100	3.087296	5	0.4910802
H	2	100	3.087296	10	0.8029764

3.5 Comments

- All comparisons in this sections are at the same values of `ncp` defined above.
- And between `ddf` = 100 and `ddf` = 10.

3.5.1 Fig. E

- This corresponds to `ncp` = 0, `ndf` = 2 and `ddf` = 100.
- The critical value is `fCrit_2_100` = 3.0872959. Notice the decrease compared to the previous value for `ncp` = 0, i.e., 4.102821, for `ddf` = 10.
- One expects that increasing `ddf` will make it more likely that the NH will be rejected, and this is confirmed below.
- All else equal, statistical power increases with increasing `ddf`.

3.5.2 Fig. F

- This corresponds to `ncp` = 2, `ndf` = 2 and `ddf` = 100.
- The probability of exceeding the critical value is `prob > fCrit_2_100` = 0.2199264, greater than the previous value, i.e., 0.177584 for `ddf` = 10.

3.5.3 Fig. G

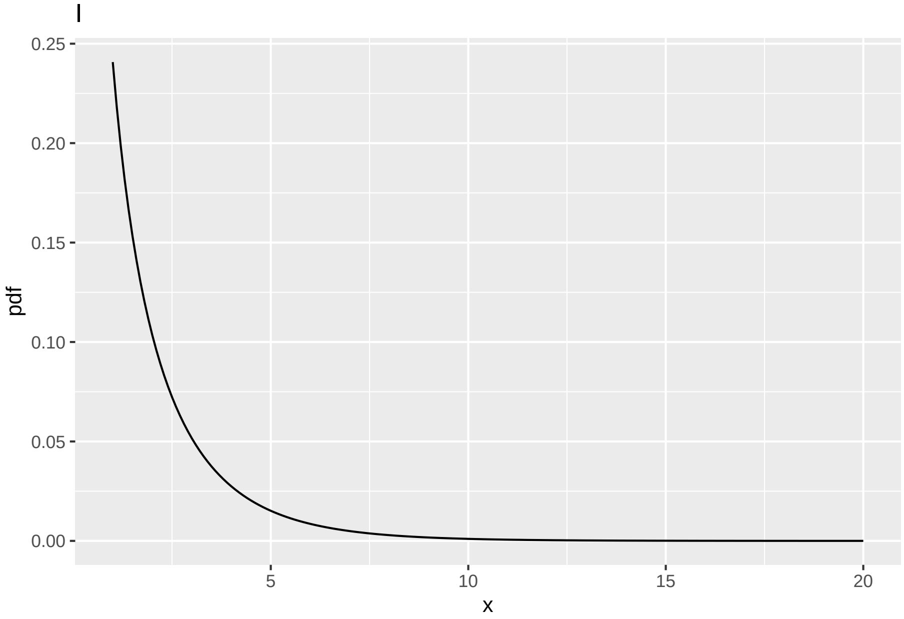
- This corresponds to `ncp = 5`, `ndf` = 2 and `ddf` = 100.
- The probability of exceeding the critical value is `prob > fCrit_2_100` = 0.4910802.
- This is greater than the previous value, i.e., 0.3876841 for `ddf` = 10.

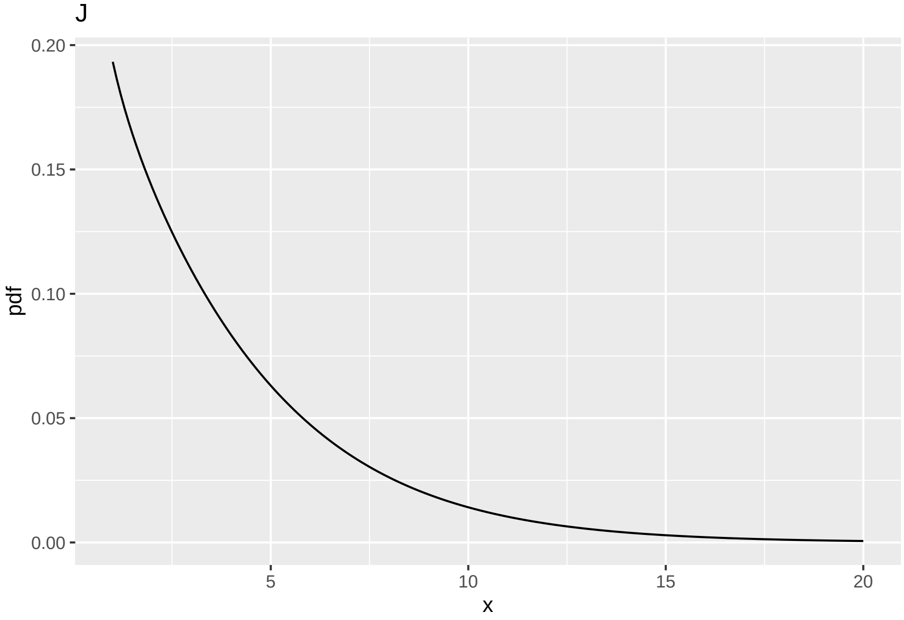
3.5.4 Fig. H

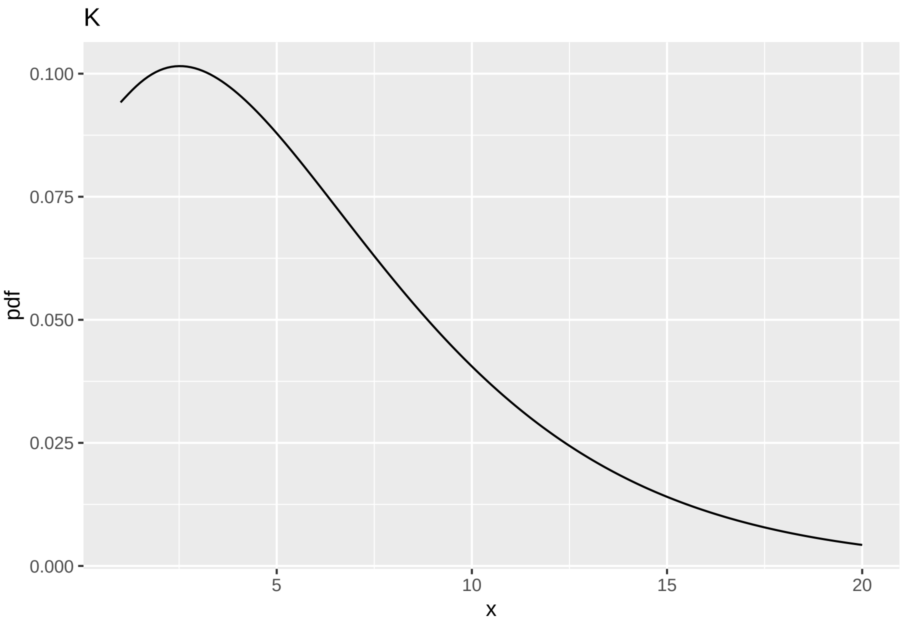
- This corresponds to `ncp = 10`, `ndf` = 2 and `ddf` = 100.

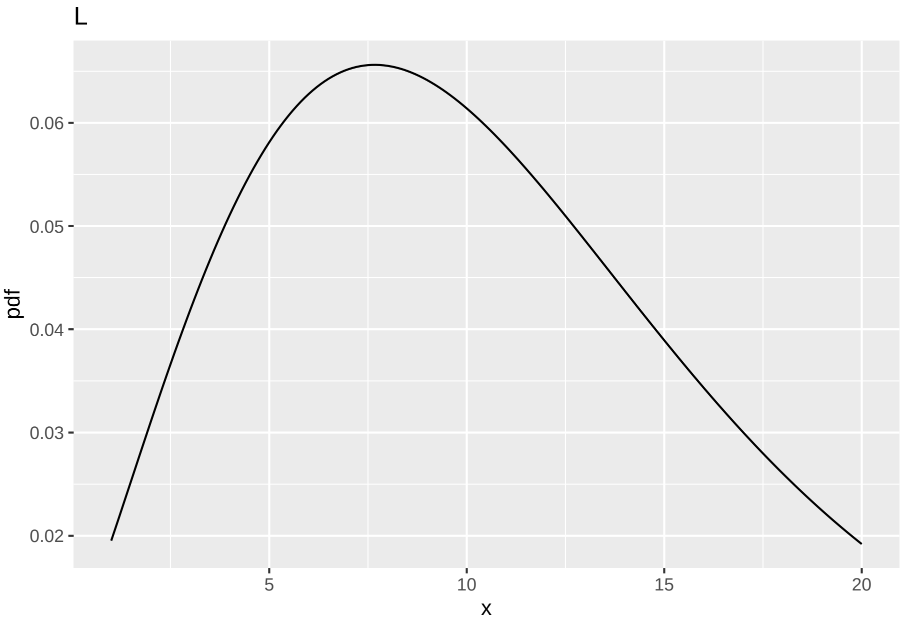
- The probability of exceeding the critical value is `prob > fCrit_2_100` is 0.8029764.
- This is greater than the previous value, i.e., 0.6769776 for `ddf` $= 10$.

3.6 Effect of `ncp` for `ndf` = 1, `ddf` = 100






K
pdf
0.100
0.075
0.050
0.025
0.000
x
5
10
15
20


L
pdf
0.06
0.05
0.04
0.03
0.02
x
5
10
15
20

	ndf	ddf	fCrit	ncp	pFgtFCrit
A	2	10	4.102821	0	0.0500000
B	2	10	4.102821	2	0.1775840
C	2	10	4.102821	5	0.3876841
D	2	10	4.102821	10	0.6769776
E	2	100	3.087296	0	0.0500000
F	2	100	3.087296	2	0.2199264
G	2	100	3.087296	5	0.4910802
H	2	100	3.087296	10	0.8029764
I	1	100	3.936143	0	0.0500000
J	1	100	3.936143	2	0.2883607
K	1	100	3.936143	5	0.6004962
L	1	100	3.936143	10	0.8793619

3.7 Comments

- All comparisons in this sections are at the same values of `ncp` defined above and at `ddf` = 100.
- And between `ndf` = 1 and `ndf` = 2.

3.7.1 Fig. I

- This corresponds to `ncp` = 0, `ndf` = 1 and `ddf` = 100.
- The critical value is `fCrit_1_100` = 3.936143.
- Notice the increase in the critical value as compared to the corresponding value for `ndf = 2`, i.e., 3.0872959.
- One might expect power to decrease, **but see below**.

3.7.2 Fig. J

- This corresponds to `ncp` = 2, `ndf` = 1 and `ddf` = 100.
- Now `prob > fCrit_1_100` = 0.2883607, larger than the previous value 0.2199264.
- The power has actually increased.

3.7.3 Fig. K

- This corresponds to `ncp` = 5, `ndf` = 1 and `ddf` = 100‘’,
- Now `prob > fCrit_1_100` = 0.6004962, larger than the previous value 0.4910802.
- Again, the power has actually increased.

3.7.4 Fig. L

- This corresponds to `ncp` = 10, `ndf` = 1 and `ddf` = 100
- Now `prob > fCrit_1_100` is 0.8793619, larger than the previous value 0.8029764.
- The power has actually increased.

3.8 Summary

- Power increases with increasing `ddf` and `ncp`.
- The effect of increasing `ncp` is quite dramatic. This is because power depends on the square of `ncp`.
- Decreasing `ndf` also **increases** power. At first glance this may seem counterintuitive, as `fCrit` has gone up, but is explained by the differing shapes of the two distributions: the pdf is broader for `ndf` = 1 as compared to `ndf` = 2 (compare Fig. L to H).

3.9 References

Chapter 4

ROC-DBMH sample size from first principles

4.1 Introduction

The starting point is a **pilot** study. The variability in this dataset (specifically the variance components, subsequently converted to mean squares), obtained by running the significance testing function `StSignificanceTesting()`, is used to extrapolate to the necessary numbers of readers and cases, in the **pivotal** study, to achieve the desired power. In this example, the observed effect size in the pilot study is used as the anticipated effect size for the pivotal study – this is generally not a good idea as discussed in **Chapter 11** under "Cautionary notes". Shown below, and the reader should confirm, is a first principles implementation of the relevant formulae in **Chapter 11**.

4.2 Sample size estimation using the DBMH method

The Van Dyke dataset in file `VanDyke.lrc`, in `"MRMC"` format, is regarded as a pilot study. The command `rocData <- DfReadDataFile(fileName, format = "MRMC")` reads the data and saves it to a `dataset` object `rocData`. For more on data formats click here. The next line uses the function `StSignificanceTesting()` to apply `method = "DBMH"` analysis, the default, using the `FOM = "Wilcoxon"` figure of merit. The next line extracts the variance components `varYTR`, `varYTC` and `varYEps` (the Y's denote pseudovalue based values). The next line extracts the effect size.

```
alpha <- 0.05
rocData <- dataset02 ##"VanDyke.lrc"
#fileName <- dataset03 ## "Franken1.lrc"
retDbm <- StSignificanceTesting(dataset = rocData, FOM = "Wilcoxon", method = "DBMH")
varYTR <- retDbm$varComp$varTR;varYTC <- retDbm$varComp$varTC;varYEps <- retDbm$varCo
effectSize <- retDbm$ciDiffTrtRRRC$Estimate
```

The *observed* effect size is `effectSize` = -0.0438003, which, in this example, is used as the *anticipated* effect size, generally not a good idea. **See Chapter 11 for nuances regarding the choice of this all important value.** The following code snippet reveals the names and array indexing of the pseudovalue variance components.

```
retDbm$varComp
#>          varR       varC        varTR     varTC      varRC    varErr
#> 1 0.001534999 0.02724923 0.0002004025 0.0119753 0.01226473 0.0399716
```

For example, the treatment-reader pseudovalue variance component is the third element of `retDbm$varComp`.

4.2.1 Random reader random case (RRRC)

This illustrates random reader random case sample size estimation. Assumed are 10 readers and 163 cases in the pivotal study. The non-centrality parameter is defined by:

TBA

The sampling distribution of the F-statistic under the AH is:

TBA

Also,

TBA

where `d` is the observed effect size, i.e., `effectSize`. The formulae for calculating the mean-squares are in (Hillis and Berbaum, 2004), implemented in `UtilMeanSquares()`.

```
#RRRC
J <- 10;K <- 163
ncp <- (0.5*J*K*(effectSize)^2)/(K*varYTR+max(J*varYTC,0)+varYEps)
MS <- UtilMeanSquares(rocData, FOM = "Wilcoxon", method = "DBMH")
ddf <- (MS$msTR+max(MS$msTC-MS$msTRC,0))^2/(MS$msTR^2)*(J-1)
FCrit <- qf(1 - alpha, 1, ddf)
Power1 <- 1-pf(FCrit, 1, ddf, ncp = ncp)
```

The next line calculates the non centrality parameter, `ncp` = 8.1269825. Note that `effectSize` enters as the **square**. The `UtilMeanSquares()` function returns the mean-squares as a `list` (ignore the last two rows of output for now).

```
str(MS)
#> List of 9
#>  $ msT        : num 0.547
#>  $ msR        : num 0.437
#>  $ msC        : num 0.397
#>  $ msTR       : num 0.0628
#>  $ msTC       : num 0.0521
#>  $ msRC       : num 0.0645
#>  $ msTRC      : num 0.04
#>  $ msCSingleT: num [1:2] 0.336 0.16
#>  $ msCSingleR: num [1:5] 0.1222 0.2127 0.1365 0.0173 0.1661
```

The next line calculates `ddf` = 12.822129. The remaining lines calculate the critical value of the F-distribution, `FCrit` = 4.680382 and statistical power = 0.7494133, which by design is close to 80%, i.e., the numbers of readers and cases were chosen to achieve this value.

4.2.2 Fixed reader random case (FRRC)

This code illustrates fixed reader random case sample size estimation. Assumed are 10 readers and 133 cases in the pivotal study. The formulae are:

TBA

The sampling distribution of the F-statistic under the AH is:

TBA

```
#FRRC
ncp <- (0.5*J*K*(effectSize)^2)/(max(J*varYTC,0)+varYEps)
ddf <- (K-1)
FCrit <- qf(1 - alpha, 1, ddf)
Power2 <- 1-pf(FCrit, 1, ddf, ncp = ncp)
```

This time non centrality parameter, `ncp` = 7.9873835, `ddf` = 132, `FCrit` = 3.912875 and statistical power = 0.8011167. Again, be design, this is close to 80%. Note that when readers are regarded as a fixed effect, fewer cases are needed to achieve the desired power. Freezing out a source of variability results in a more stable measurement and hence fewer cases are needed to achieve the desired power.

4.2.3 Random reader fixed case (RRFC)

This code illustrates random reader random case sample size estimation. Assumed are 10 readers and 53 cases in the pivotal study. The formulae are:

TBA

The sampling distribution of the F-statistic under the AH is:

TBA

```
#RRFC
ncp <- (0.5*J*K*(effectSize)^2)/(K*varYTR+varYEps)
ddf <- (J-1)
FCrit <- qf(1 - alpha, 1, ddf)
Power3 <- 1-pf(FCrit, 1, ddf, ncp = ncp)
```

This time non centrality parameter, `ncp` = 10.0487164, `ddf` = 9, `FCrit` = 5.117355 and statistical power = 0.8049666. Again, be design, this is close to 80%.

4.3 Summary

For 10 readers, the numbers of cases needed for 80% power is largest (163) for RRRC, intermediate (133) for FRRC and least for RRFC (53). For all three analyses, the expectation of 80% power is met.

4.4 References

Chapter 5

Proper ROCs

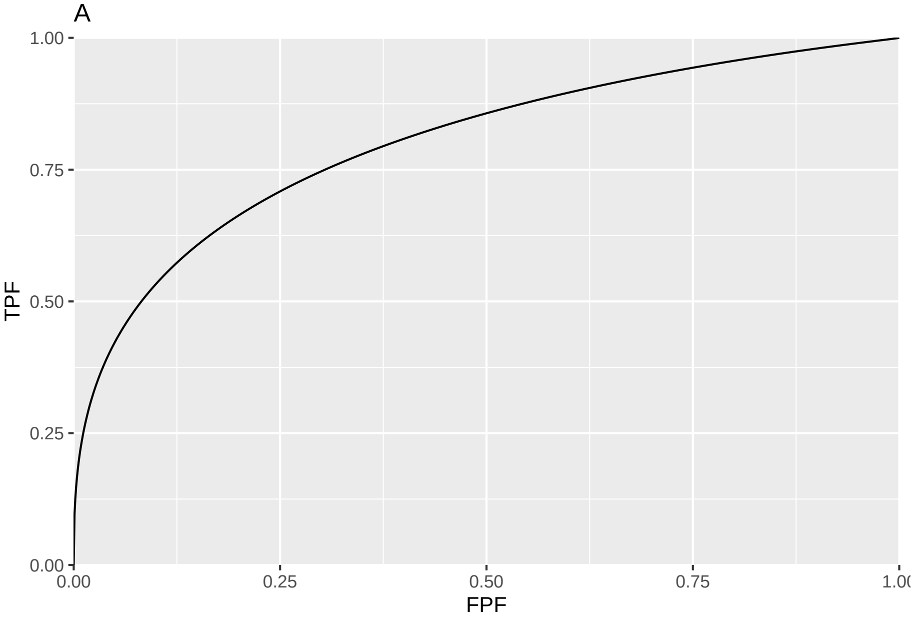
5.1 Helper functions

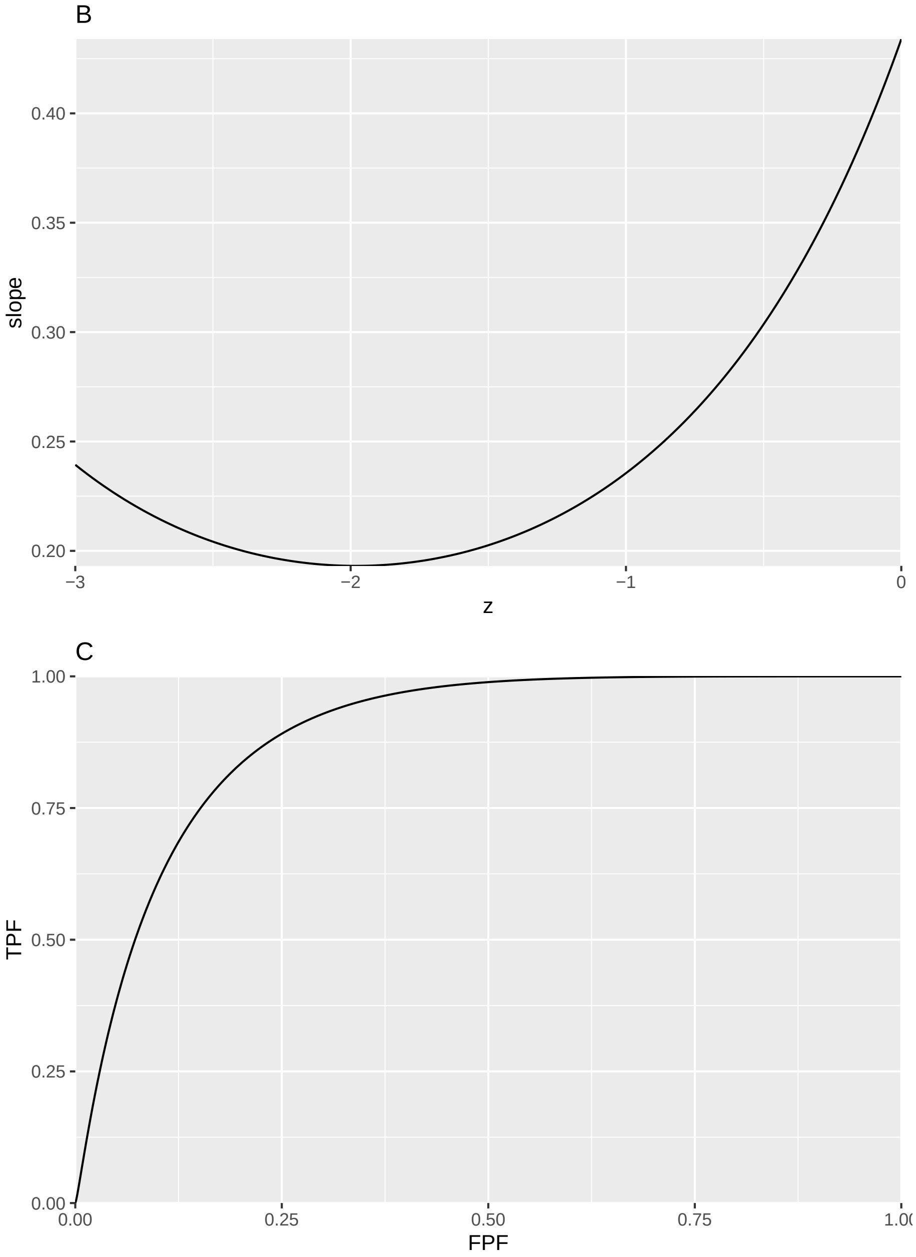
5.2 Definitions of PROPROC parameters in terms of binormal model parameters

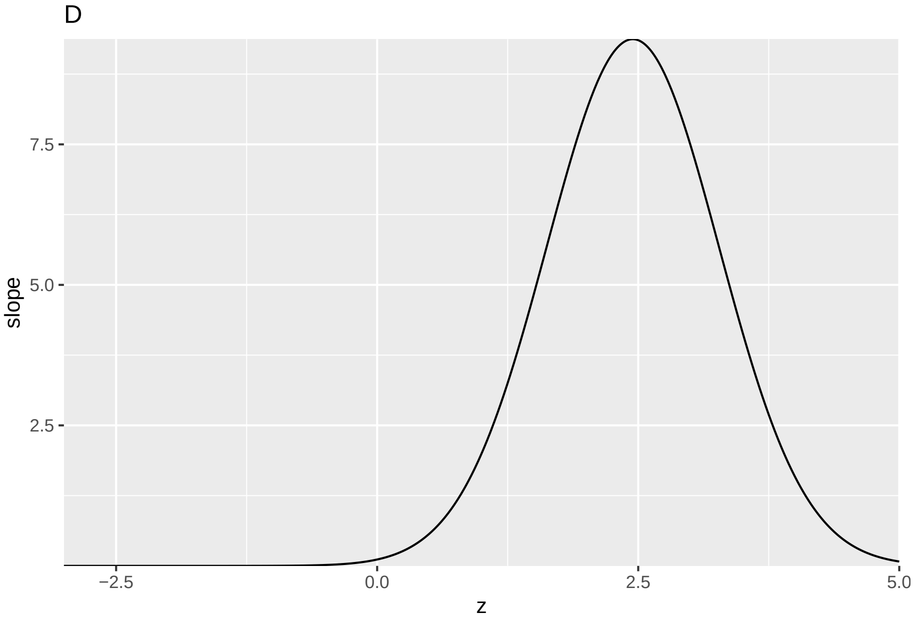
$$
\begin{aligned}
c &= \frac{b-1}{b+1} \\
d_a &= \frac{\sqrt{2}a}{\sqrt{1+b^2}}
\end{aligned}
$$

5.3 Main code and output

```
c1Arr <-   c(-0.1322804, 0.2225588); daArr  <-  c(1.197239, 1.740157)
myLabel <- c("A", "B", "C", "D")
myLabelIndx <- 1
for (i in 1:2)
{
  c1 <- c1Arr[i]
  da <- daArr[i]
  ret <- Transform2ab(da, c1)
  a <- ret$a;b <- ret$b
  if (i == 1) z <- seq(-3, 0, by = 0.01) # may need to adjust limits to view detail of slope plo
  if (i == 2) z <- seq(-3, 5, by = 0.01) # may need to adjust limits to view detail of slope plo
  FPF <- seq(0.0, 1, 0.001)
  TPF <- rocY(FPF, a, b)

  rocPlot <- data.frame(FPF = FPF, TPF = TPF)
  plotRoc <- ggplot(rocPlot, aes(x = FPF, y = TPF)) +
    geom_line()  +
   scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0))   +
    ggtitle(myLabel[myLabelIndx]);myLabelIndx <- myLabelIndx + 1

  slope <-b*dnorm(a-b*z)/dnorm(-z) # same as likelihood ratio

  slopePlot <- data.frame(z = z, slope = slope)
  p <- ggplot(slopePlot, aes(x = z, y = slope)) +
    geom_line()  +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0))  +
    ggtitle(myLabel[myLabelIndx]);myLabelIndx <- myLabelIndx + 1
  print(plotRoc);print(p)
}
```




B
slope
0.40
0.35
0.30
0.25
0.20
−3
−2
−1
0
z
C
TPF
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
FPF



5.4 Discussion

Plot A is for `c1` = -0.1322804, `da` = 1.197239 while plot C is for `c1` = 0.2225588, `da` = 1.740157. Plots B and D are the corresponding slope plots as functions of the binormal model z-sample. In plot A, the slope is infinite near the origin and the curve approaches the upper-right corner with finite slope. The situation is reversed in plot C where the slope is finite near the origin and the curve approaches the upper-right corner with zero slope.

These two readers are from a clinical dataset, `dataset01`. Highest rating inferred ROC data from original FROC data, were analyzed by PROPROC and the resulting parameter values are coded here. They were chosen as they demonstrate key differences in the shapes of proper ROC plots. Plot A corresponds to a negative value of `c1`, which implies `b < 1`. The slope of the proper ROC is infinite near the origin and approaches a positive constant near the upper right corner of the ROC. Plot C is for a positive value of `c1`, i.e., for `b > 1`. Now the slope of the proper ROC is finite near the origin and approaches zero near the upper right corner.

Considering plot D, as one "cuts" the slope axis horizontally with a sliding threshold, starting with very high values and moving downwards, the slope of the ROC curve starts at the origin with a large but finite value. This corresponds to the peak in plot D. Above the peak, there are no solutions for `z`. The slope decreases monotonically to zero, corresponding to the flattening out of the slope at zero for `z ~   -2`.

The two values of `z` corresponding to each cut implies, of course, that the binormal model based proper algorithm has to do a lot of bookkeeping, since each horizontal cut splits the decision axis into 3 regions. One can think of shrinking each of plots B & D horizontally to zero width, and all that remains is the slope axis with a thick vertical line superimposed on it, corresponding to the horizontally collapsed curves. In plot B the vertical line extends from positive infinity down to about 0.1, and represents the range of decision variable samples encountered by the observer on the likelihood ratio scale. In plot D the vertical line extends from a finite value (~9.4) to zero. For the stated binormal model parameters values outside of these ranges are not possible.

Chapter 6

Metz Eqn36 numerical check

6.1 Helper functions

6.2 Main code and output

```
npts <-  10000
for (i in 1:2) {
  for (j in 1:5) {
    C  <-  c1[i,j]
    da  <-  d_a1[i,j]
    ret <- GetLimits(da,C)
    LL <- ret$LL;UL <- ret$UL
    vc  <-  seq (LL, UL, length.out = npts)
    TPF  <-  TruePositiveFraction (vc, da, C)
    FPF <- FalsePositiveFraction (vc, da, C)
    FPF <- rev(FPF);TPF <- rev(TPF)
    df2 <- data.frame(FPF = FPF, TPF = TPF)
    # do integral numerically
    numAuc <- trapz(FPF, TPF)
    # Implement Eqn. 36 from Metz-Pan paper
    rho <- -(1-C^2)/(1+C^2);sigma <- rbind(c(1, rho), c(rho, 1))
    lower <- rep(-Inf,2);upper <- c(-da/sqrt(2),0)
    aucProproc <- pnorm(da/sqrt(2)) + 2 * pmvnorm(lower, upper, sigma = sigma)
    aucProproc <-  as.numeric(aucProproc)
    cat("i = ", i,"j = ", j,"C = ", C, ", da = ", da, "aucProproc =", aucProproc, "Norm. Diff. =
  }
}
#> i =  1 j =  1 C =  -0.1322804 , da =  1.197239 aucProproc = 0.8014164 Norm. Diff. =
#> i =  1 j =  2 C =  -0.08696513 , da =  1.771176 aucProproc = 0.8947898 Norm. Diff.
#> i =  1 j =  3 C =  -0.1444419 , da =  1.481935 aucProproc = 0.8526605 Norm. Diff. =
#> i =  1 j =  4 C =  0.08046016 , da =  1.513757 aucProproc = 0.8577776 Norm. Diff. =
#> i =  1 j =  5 C =  0.2225588 , da =  1.740157 aucProproc = 0.8909392 Norm. Diff. =
#> i =  2 j =  1 C =  -0.08174248 , da =  0.6281251 aucProproc = 0.6716574 Norm. Diff.
#> i =  2 j =  2 C =  0.04976448 , da =  0.9738786 aucProproc = 0.7544739 Norm. Diff.
#> i =  2 j =  3 C =  -0.1326126 , da =  1.155871 aucProproc = 0.7931787 Norm. Diff. =
#> i =  2 j =  4 C =  0.1182226 , da =  1.620176 aucProproc = 0.8740274 Norm. Diff. =
#> i =  2 j =  5 C =  0.0781033 , da =  0.8928816 aucProproc = 0.7360989 Norm. Diff. =
```

6.3 Discussion

Note the close correspondence between the formula, Eqn. 36 in the Metz-Pan paper and the numerical estimate. As a historical note, Eqn. 31 and Eqn. 36 (they differ only in parameterizations) in the referenced publication are provided without proof – it was probably obvious to Prof Metz or he wanted to leave it to us “mere mortals” to figure it out, as a final parting gesture of his legacy. The author once put a significant effort into proving it and even had a bright graduate student from the biostatistics department work on it to no avail. The author has observed that these equations always yield very close to the numerical estimates, to within numerical precisions, so the theorem is correct empirically, but he has been unable to prove it analytically. It is left as an exercise for a gifted reader to prove/disprove these equations.

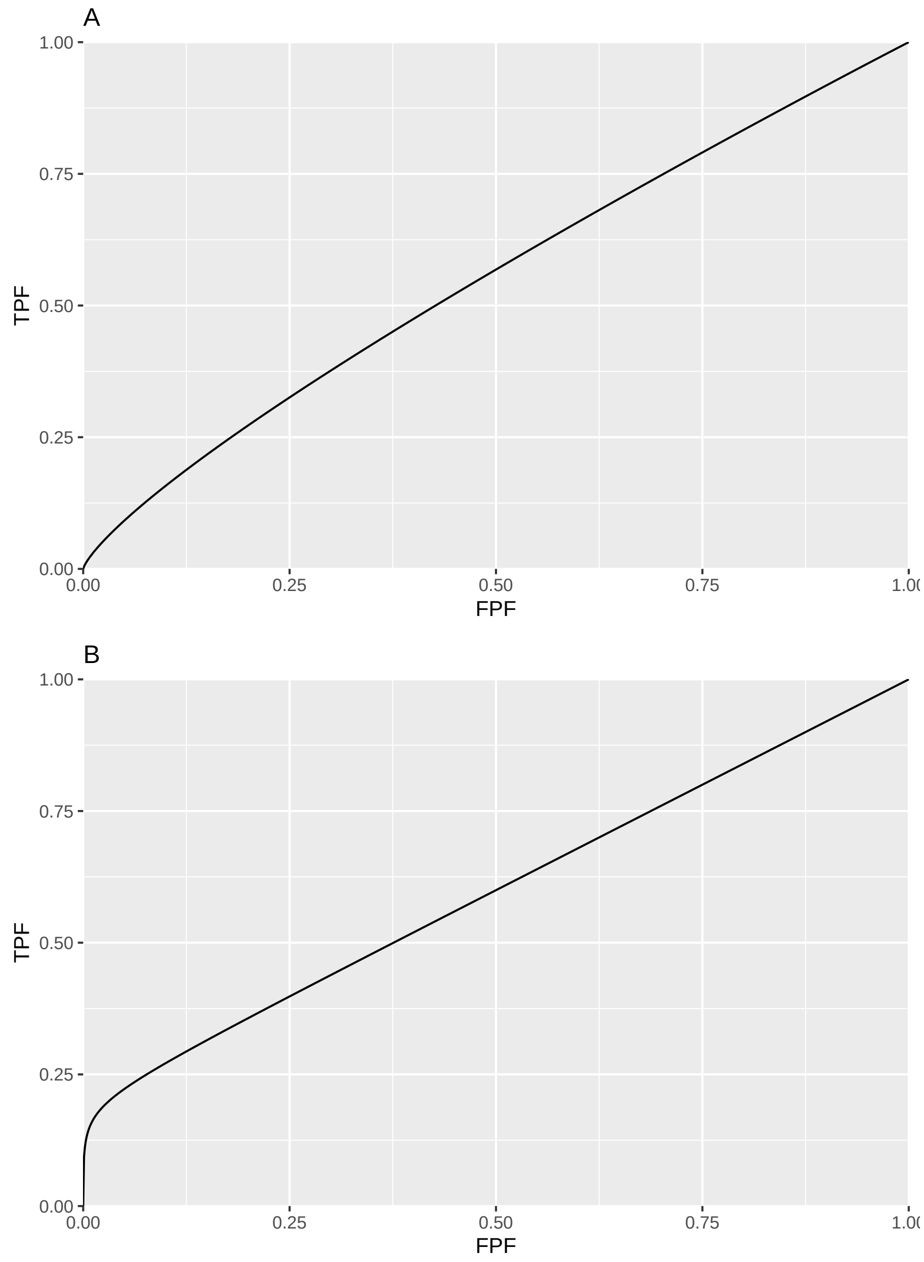
Chapter 7

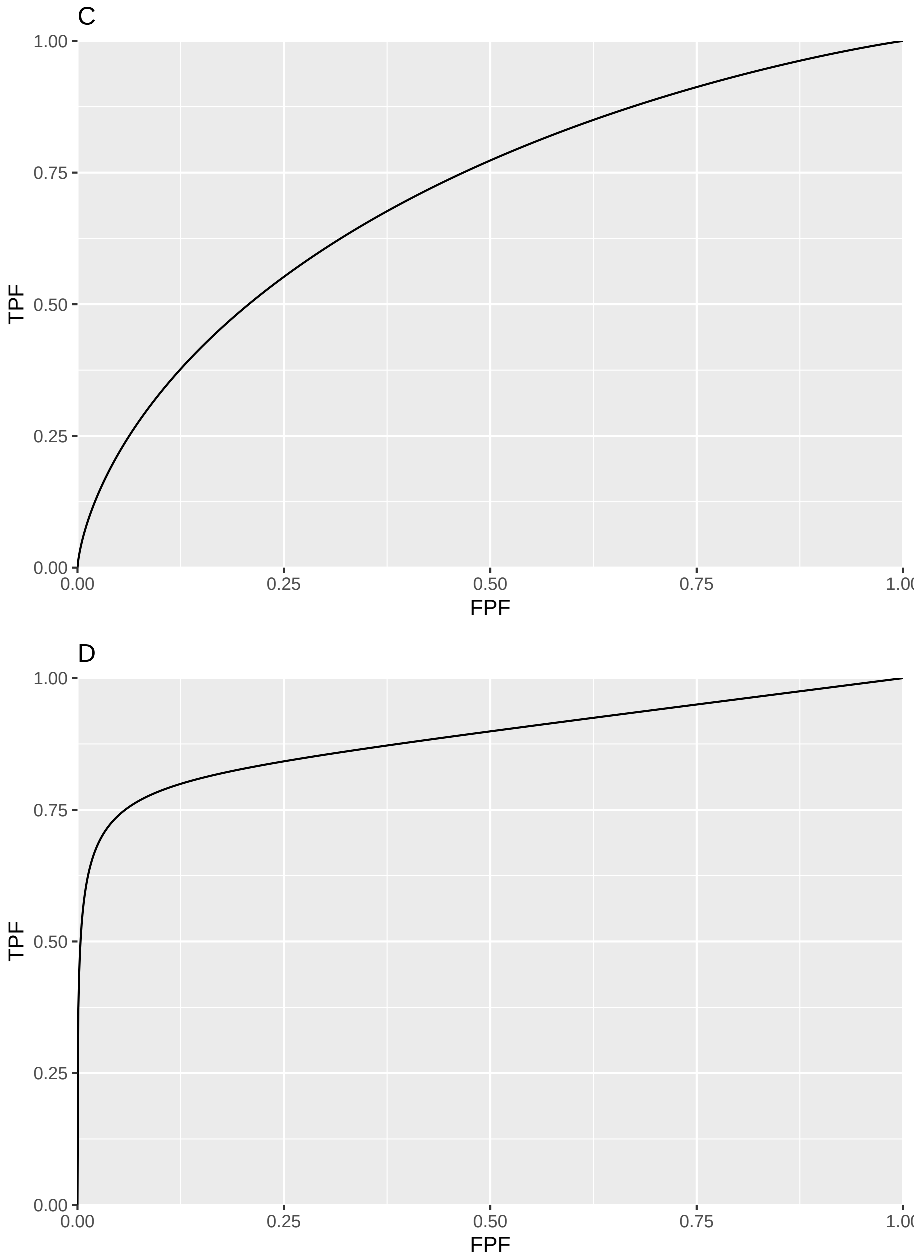
CBM Plots

7.1 Helper functions

7.2 Main code and output

```
#> Fig. A : mu =  1 , alpha =  0.2
#> Fig. B : mu =  3 , alpha =  0.2
#> Fig. C : mu =  1 , alpha =  0.8
#> Fig. D : mu =  3 , alpha =  0.8
```


A
1.00
0.75
0.50
0.25
0.00
TPF
0.00
0.25
0.50
0.75
1.00
FPF
B
1.00
0.75
0.50
0.25
0.00
TPF
0.00
0.25
0.50
0.75
1.00
FPF

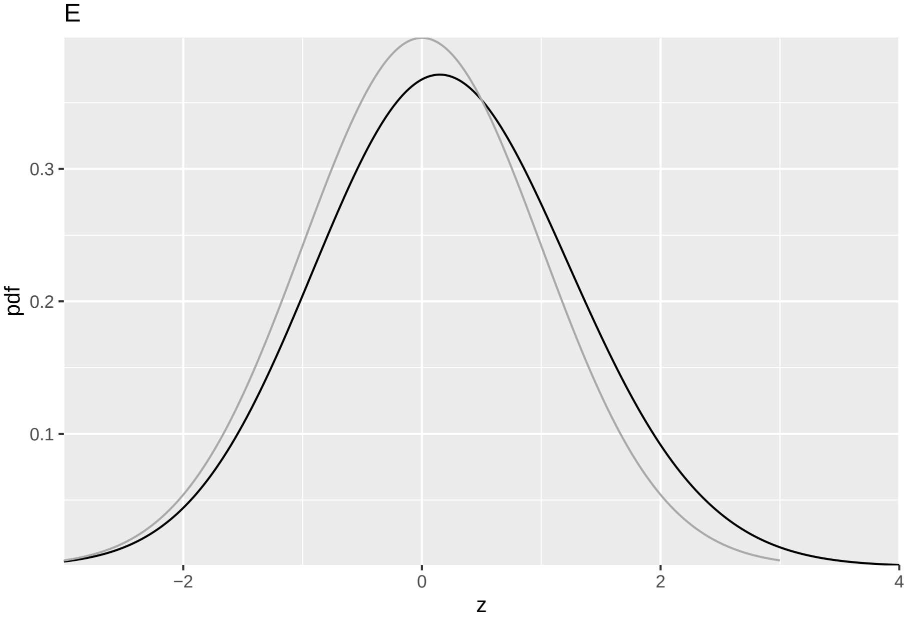

C
1.00
0.75
0.50
0.25
0.00
TPF
0.00
0.25
0.50
0.75
1.0
FPF
D
1.00
0.75
0.50
0.25
0.00
TPF
0.00
0.25
0.50
0.75
1.0
FPF

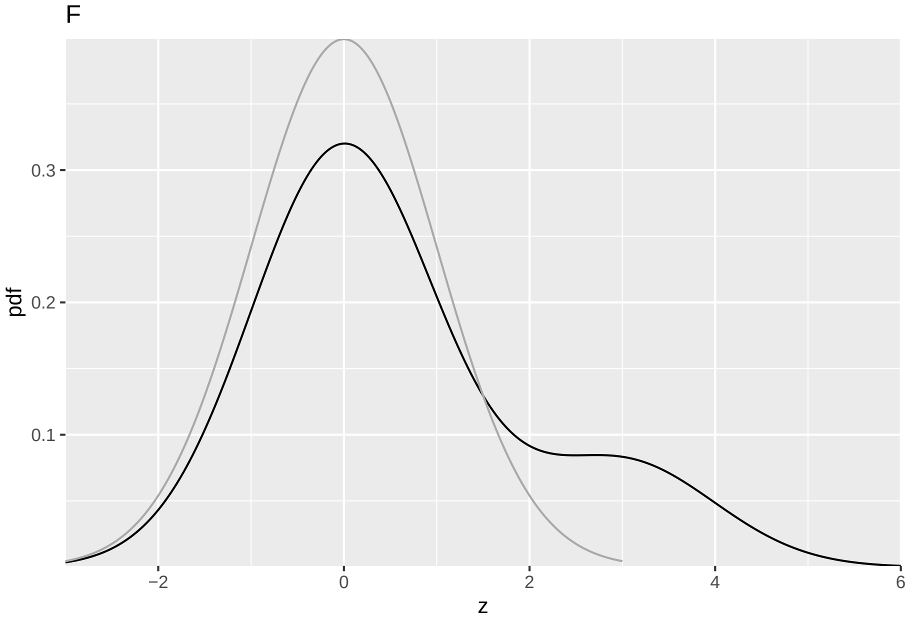
7.3 Comments

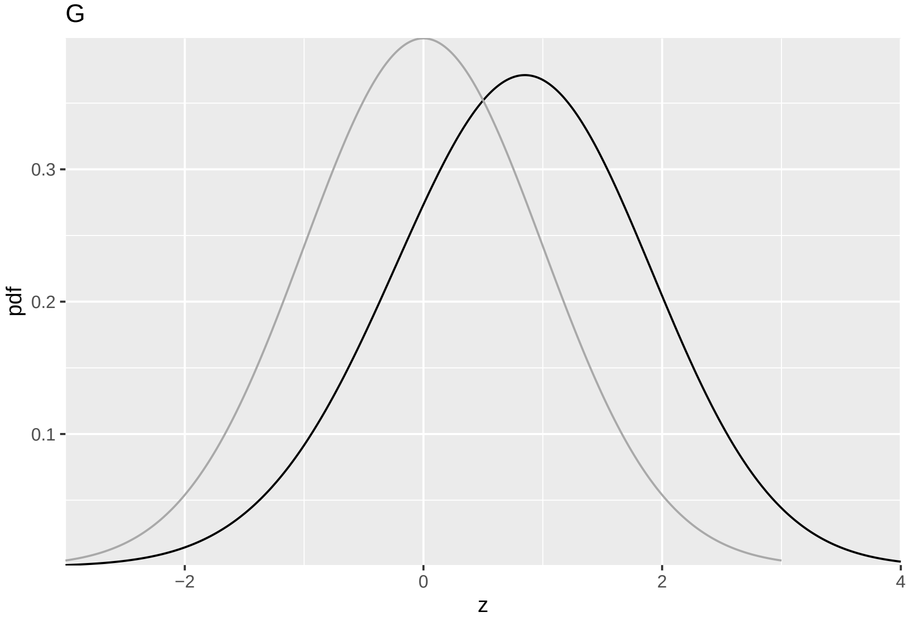
Plots A - D show ROC curves predicted by the CBM model; the corresponding values of the *mu* and *alpha* parameters are indicated above the plots. For small *mu* and/or *alpha* the curve approaches the chance diagonal, consistent with the notion that if the lesion is not visible, performance can be no better than chance level.

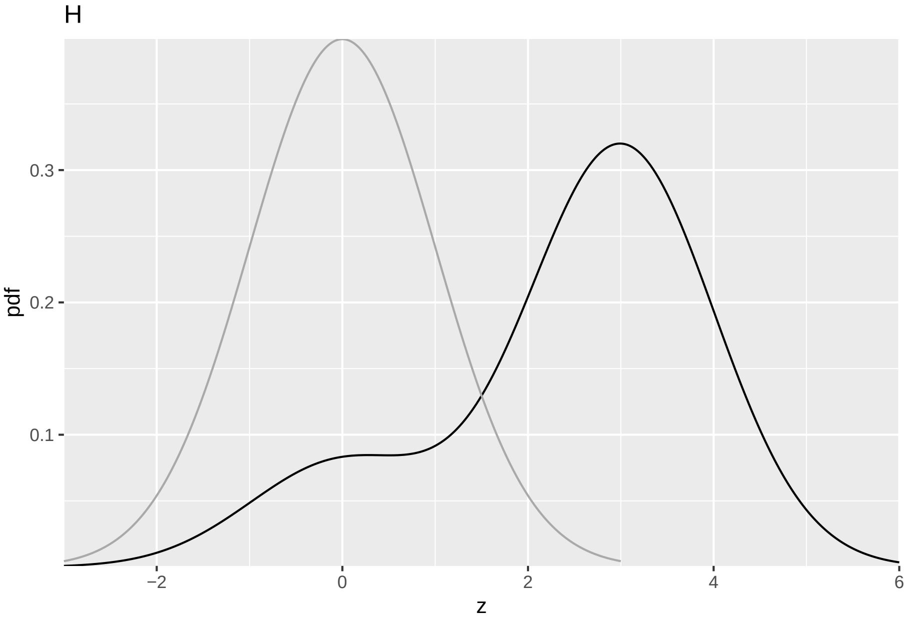
7.4 pdf plots

```
#> Fig. E : mu =  1 , alpha =  0.2
#> Fig. F : mu =  3 , alpha =  0.2
#> Fig. G : mu =  1 , alpha =  0.8
#> Fig. H : mu =  3 , alpha =  0.8
```




F
pdf
0.3
0.2
0.1
z
−2
0
2
4
6


G
pdf
0.3
0.2
0.1
z
−2
0
2
4

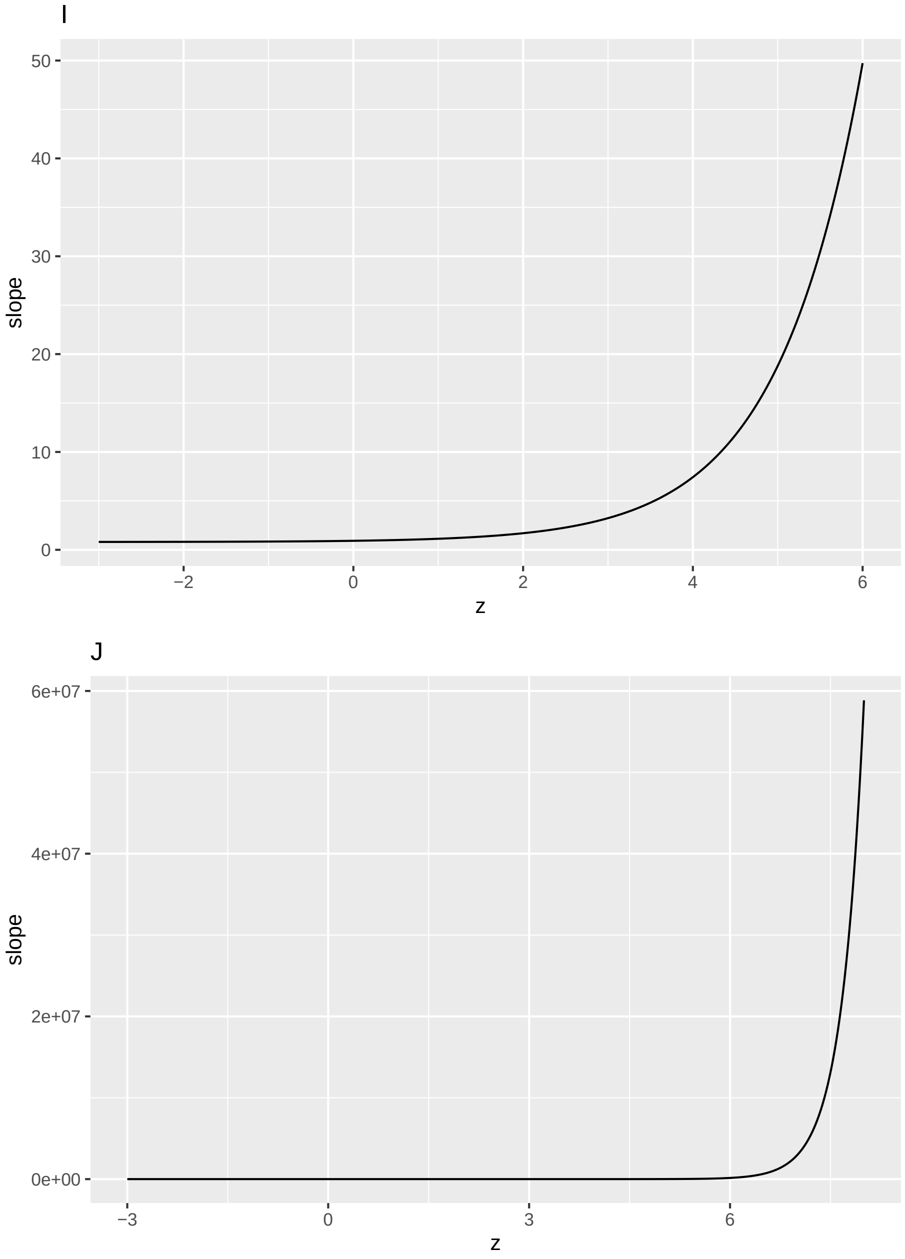


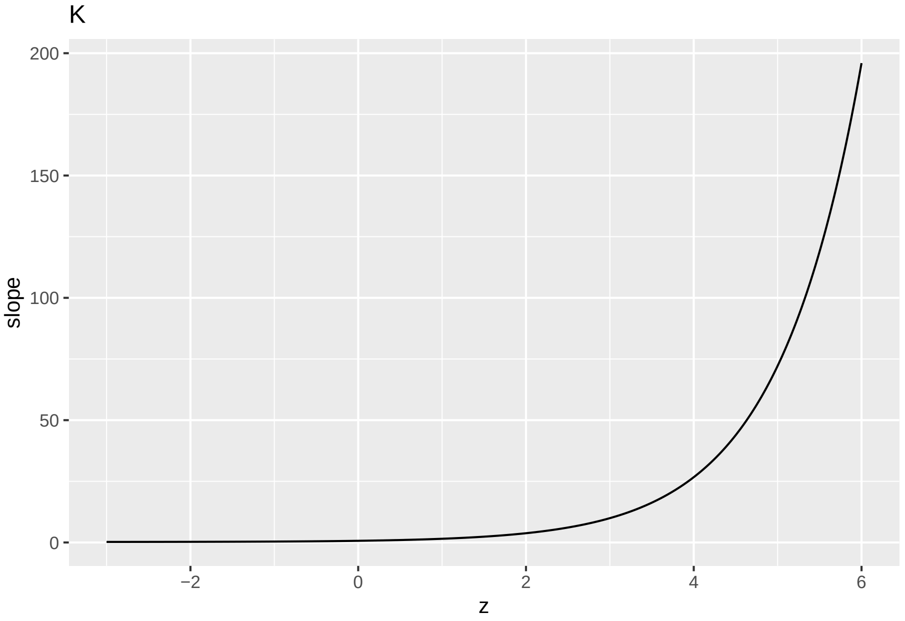
7.5 Comments

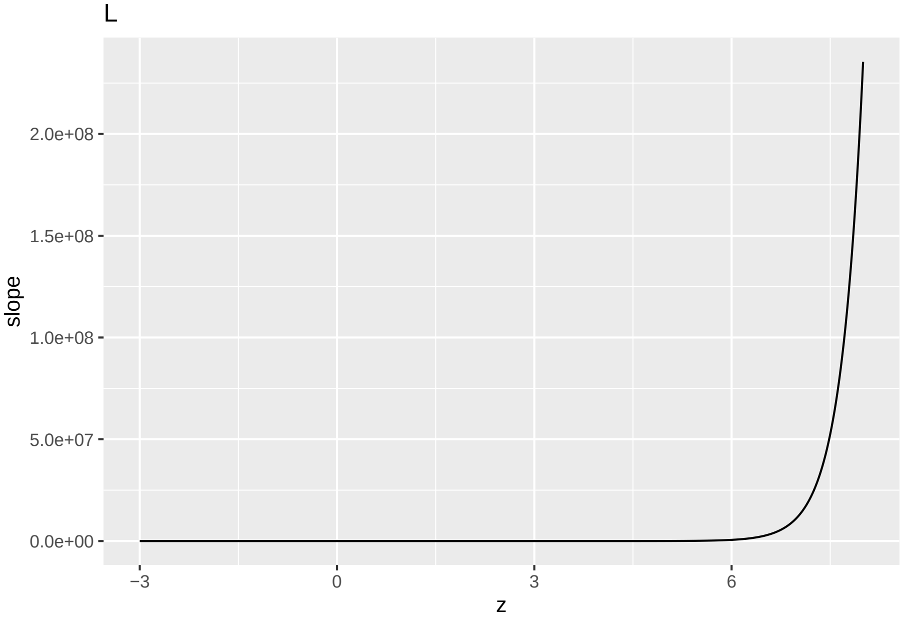
The dark line is the diseased distribution. The grey line is the non-diseased distribution. The bimodal diseased distribution is clearly evident in plots F and H.

7.6 likelihood ratio plots

```
#> Fig. I : mu =  1 , alpha =  0.2
#> Fig. J : mu =  3 , alpha =  0.2
#> Fig. K : mu =  1 , alpha =  0.8
#> Fig. L : mu =  3 , alpha =  0.8
```


I
slope
50
40
30
20
10
0
−2
0
2
4
6
z
J
slope
6e+07
4e+07
2e+07
0e+00
−3
0
3
6
z


K
slope
200
150
100
50
0
−2
0
2
4
6
z


L
slope
2.0e+08
1.5e+08
1.0e+08
5.0e+07
0.0e+00
−3
0
3
6
z

7.7 Comments

Close examination of the region near the flat part shows it does not plateau at zero; rather the minimum is at 1 - *alpha*, explaining the non-zero limiting slope of the predicted curve near (1, 1).

Chapter 8

ROI paradigm data

8.1 Introduction; this vignette is under construction!

- In the region-of-interest (ROI) paradigm (Obuchowski, 1997, Obuchowski et al. (2000)) each case is regarded as consisting of Q_k ($Q_k \geq 1$) "quadrants" or "regions-of-interest" or ROIs, where k is the case index ($k = 1, 2, ..., K$) and K is the total number of cases (i.e., case-level non-diseased plus case-level diseased cases). Each ROI needs to be classified, by the investigator, as either ROI-level-non-diseased (i.e., it has no lesions) or ROI-level-diseased (i.e., it has at least one lesion). **Note the distinction between case-level and ROI-level truth states.** One can have ROI-level non-diseased regions in a case-level diseased case. A case-level diseased case must contain at least one ROI-level diseased region and a case-level non-diseased case cannot have any ROI-level diseased regions.
- The observer gives a single rating (in fact an ordered label) to each ROI, denoted R_{kr} (r = 1, 2, ..., Q_k). Here r is the ROI index and k is the case index. The rating can be an integer or quasi- continuous (e.g., 0 – 100), or a floating point value, as long as higher numbers represent greater confidence in presence of one or more lesions in the ROI.
- The ROI paradigm is not restricted to 4 or even a constant number of ROIs per case. That is the reason for the k subscript in Q_k.
- The ROI data structure is a special case of the FROC data structure, the essential difference being that the number of ratings per case is an a-priori known value, equal to Q_k.
- ROI-level non-diseased region ratings are stored in the `NL` field and ROI-level diseased region ratings are stored in the `LL` field.

- One can think of the ROI paradigm as similar to the FROC paradigm, but with localization accuracy restricted to belonging to a region (one cannot distinguish multiple lesions within a region). Unlike the FROC paradigm, a rating *is required* for every ROI.

8.2 An example ROI dataset

An example simulated ROI dataset is included as `datasetROI`.

```
str(datasetROI)
#> List of 8
#>  $ NL          : num [1:2, 1:5, 1:90, 1:4] 0.95 0.927 0.556 0.805 1.421 ...
#>  $ LL          : num [1:2, 1:5, 1:40, 1:4] 1.57 2.31 2.3 2.34 2.34 ...
#>  $ lesionVector: int [1:40] 2 3 2 2 3 3 1 2 3 3 ...
#>  $ lesionID    : num [1:40, 1:4] 2 1 1 1 1 2 4 1 1 1 ...
#>  $ lesionWeight: num [1:40, 1:4] 0.5 0.333 0.5 0.5 0.333 ...
#>  $ dataType    : chr "ROI"
#>  $ modalityID  : Named chr [1:2] "1" "2"
#>   ..- attr(*, "names")= chr [1:2] "1" "2"
#>  $ readerID    : Named chr [1:5] "1" "2" "3" "4" ...
#>   ..- attr(*, "names")= chr [1:5] "1" "2" "3" "4" ...
datasetROI$NL[1,1,1,]
#> [1]  0.9498680 -0.0582497 -0.7763780  0.0120730
mean(datasetROI$NL[,,1:50,])
#> [1] 0.1014348
datasetROI$NL[1,1,51,]
#> [1] 1.01867 0.34710    -Inf    -Inf
datasetROI$lesionVector[1]
#> [1] 2
datasetROI$LL[1,1,1,]
#> [1] 1.56928 2.05945    -Inf    -Inf
x <- datasetROI$LL;mean(x[is.finite(x)])
#> [1] 1.815513
```

Examination of the output reveals that:

- This is a 2-treatment 5-reader dataset, with 50 non-diseased cases and 40 diseased cases, and $Q_k = 4$ for all k.

- For treatment 1, reader 1, case 1 (the 1st non-diseased case) the 4 ratings are 0.949868, -0.0582497, -0.776378, 0.012073. The mean of all ratings on non-diseased cases is 0.1014348.

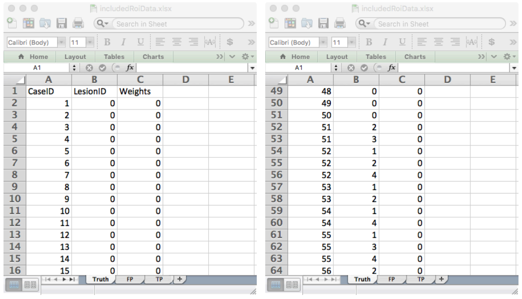
- For treatment 1, reader 1, case 51 (the 1st diseased case) the NL ratings are 1.01867, 0.3471. There are only two finite values because this case has two ROI-level-diseased regions, and 2 plus 2 makes for the assumed 4-regions per case. The corresponding `$lesionVector` field is 2.

- The ratings of the 2 ROI-level-diseased ROIs on this case are 1.56928, 2.05945. The mean rating over all ROI-level-diseased ROIs is 1.8155127.

8.3 The ROI Excel data file

- An Excel file in JAFROC format containing simulated ROI data corresponding to `datasetROI`, is included with the distribution. The first command (below) finds the location of the file and the second command reads it and saves it to a dataset object `ds`. !!!DPC!!!
- The `DfReadDataFile` function automatically recognizes that this is an *ROI* dataset. Its structure is similar to the JAFROC format Excel file, with some important differences, noted below. It contains three worksheets:

```
## fileName <- system.file(
##     "extdata", "RoiData.xlsx", package = "RJafroc", mustWork = TRUE)
## ds <- DfReadDataFile(fileName)
## ds$dataType
```



	A	B	C	D	E
1	CaseID	LesionID	Weights		
2	1	0	0		
3	2	0	0		
4	3	0	0		
5	4	0	0		
6	5	0	0		
7	6	0	0		
8	7	0	0		
9	8	0	0		
10	9	0	0		
11	10	0	0		
12	11	0	0		
13	12	0	0		
14	13	0	0		
15	14	0	0		
16	15	0	0		

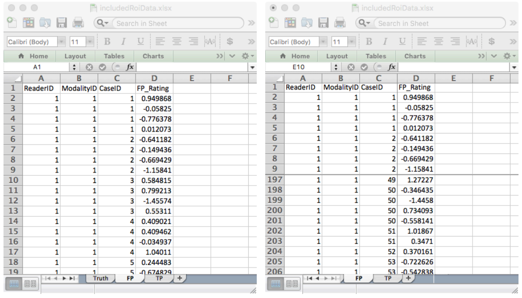
	A	B	C	D	E
49	48	0	0		
50	49	0	0		
51	50	0	0		
52	51	2	0		
53	51	3	0		
54	52	1	0		
55	52	2	0		
56	52	4	0		
57	53	1	0		
58	53	2	0		
59	54	1	0		
60	54	4	0		
61	55	1	0		
62	55	3	0		
63	55	4	0		
64	56	2	0		

Figure 8.1: Fig. 1 two views of Truth worksheet

- The `Truth` worksheet, Fig. 1, indicates which cases are diseased and which are non-diseased and the number of ROI-level-diseased region on each case.
  - There are 50 non-diseased cases (labeled 1-50) under column `CaseID` and 40 diseased cases (labeled 51-90).
  - The `LesionID` field for each non-diseased case (e.g., `CaseID` = 1) is zero and there is one row per case. For diseased cases, this field has a variable number of entries, ranging from 1 to 4. As an example,

there are two rows for `CaseID` = 51 in the Excel file: one with `LesionID` = 2 and one with `LesionID` = 3.

- The `Weights` field is always zero (this field is not used in ROI analysis).



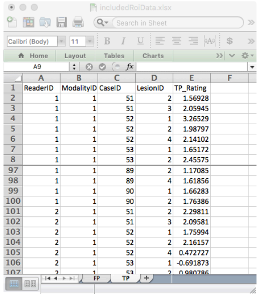
	A	B	C	D	E	F
1	ReaderID	ModalityID	CaseID	FP_Rating		
2	1	1	1	0.949868		
3	1	1	1	-0.05825		
4	1	1	1	-0.776378		
5	1	1	1	0.012073		
6	1	1	2	-0.641182		
7	1	1	2	-0.149436		
8	1	1	2	-0.669429		
9	1	1	2	-1.15841		
10	1	1	3	0.584815		
11	1	1	3	0.799213		
12	1	1	3	-1.45574		
13	1	1	3	0.55311		
14	1	1	4	0.409021		
15	1	1	4	0.409462		
16	1	1	4	-0.034937		
17	1	1	4	1.04011		
18	1	1	5	0.244483		
19	1	1	5	-0.674829		

	A	B	C	D	E	F
1	ReaderID	ModalityID	CaseID	FP_Rating		
2	1	1	1	0.949868		
3	1	1	1	-0.05825		
4	1	1	1	-0.776378		
5	1	1	1	0.012073		
6	1	1	2	-0.641182		
7	1	1	2	-0.149436		
8	1	1	2	-0.669429		
9	1	1	2	-1.15841		
197	1	1	49	1.27227		
198	1	1	50	-0.346435		
199	1	1	50	-1.4458		
200	1	1	50	0.734093		
201	1	1	50	-0.558141		
202	1	1	51	1.01867		
203	1	1	51	0.3471		
204	1	1	52	0.370161		
205	1	1	53	-0.722626		
206	1	1	53	-0.542838		

Figure 8.2: Fig. 2 two views of FP worksheet

- The `FP` (or `NL`) worksheet - this lists the ratings of ROI-level-non-diseased regions.
  - For `ReaderID` = 1, `ModalityID` = 1 and `CaseID` = 1 there are 4 rows, corresponding to the 4 ROI-level-non-diseased regions in this case. The corresponding ratings are 0.949868, -0.0582497, -0.776378, 0.012073. The pattern repeats for other treatments and readers, but the rating are, of course, different.

  - Each `CaseID` is represented in the `FP` worksheet (a rare exception could occur if a case-level diseased case has 4 diseased regions).



	A	B	C	D	E	F
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating	
2	1	1	51	2	1.56928	
3	1	1	51	3	2.05945	
4	1	1	52	1	3.26529	
5	1	1	52	2	1.98797	
6	1	1	52	4	2.14102	
7	1	1	53	1	1.65172	
8	1	1	53	2	2.45575	
97	1	1	89	2	1.17085	
98	1	1	89	4	1.61856	
99	1	1	90	1	1.66283	
100	1	1	90	2	1.76386	
101	2	1	51	2	2.29811	
102	2	1	51	3	2.09581	
103	2	1	52	1	1.75994	
104	2	1	52	2	2.16157	
105	2	1	52	4	0.472727	
106	2	1	53	1	-0.691873	
107	2	1	53	2	0.980786	

Figure 8.3: Fig. 2 TP worksheet

- The `TP` (or `LL`) worksheet - this lists the ratings of ROI-level-diseased regions.

- Because non-diseased cases generate TPs, one does not find any entry with `CaseID` = 1-50 in the `TP` worksheet.

- The lowest `CaseID` in the `TP` worksheet is 51, which corresponds to the first diseased case.

- There are two entries for this case, corresponding to the two ROI-level-diseased regions present in this case. Recall that corresponding to this `CaseID` in the `Truth` worksheet there were two entries with `LesionID` = 2 and 3. These must match the `LesionID`'s listed for this case in the `TP` worksheet. Complementing these two entries, in the `FP` worksheet for `CaseID` = 51, there are 2 entries corresponding to the two ROI-level-non-diseased regions in this case.

- One should confirm that for each diseased case the sum of the number of entries in the `TP` and `FP` worksheets is always 4.

8.4 Next, TBA

The next vignette illustrates significance testing for this paradigm.

8.5 References

Chapter 9

Analyzing data acquired according to the ROI paradigm

9.1 Introduction; this vignette is under construction!

9.2 Note to self (10/29/19) !!!DPC!!!

The FOM and DeLong method implementations need checking with a toy dataset.

9.3 Introduction

- For an ROI dataset `StSignificanceTesting()` automatically defaults to `method = "ORH"`, `covEstMethod = "DeLong"` and `FOM = "ROI"`.
- The covariance estimation method is based on the original DeLong method (DeLong et al., 1988), which is valid only for the trapezoidal AUC, i.e. ROC data, as extended by (Obuchowski, 1997) to ROI data, see formula below.
- The essential differences from conventional ROC analyses are in the definition of the ROI figure of merit, see below, and the procedure developed by (Obuchowski, 1997) for estimating the covariance matrix. Once the

covariances are known, `method = "ORH"` can be applied to perform significance testing, as described in (Obuchowski and Rockette, 1995) and (Chakraborty, 2017, Chapter 10).

9.4 The ROI figure of merit

Let X_{kr} denote the rating for the r$^{\text{th}}$ **lesion-containing** ROI in the k$^{\text{th}}$ case and let n_k^L be the total number of lesion-containing ROIs in the k$^{\text{th}}$ case. Similarly, let Y_{kr} denote the rating for the r$^{\text{th}}$ **lesion-free** ROI in the k$^{\text{th}}$ case and n_k^N denote the total number of lesion-free ROIs in the k$^{\text{th}}$ case. Let N_L denote the total number of lesion-containing ROIs in the image set and N_N denote the total number of lesion-free ROIs. These are given by:

$$N_L = \sum\nolimits_k n_k^L$$

and

$$N_N = \sum\nolimits_k n_k^N$$

The ROI figure of merit θ is defined by:

$$\theta = \frac{1}{N_L N_N} \sum\nolimits_k \sum\nolimits_{k'} \sum_{r=1}^{n_k^L} \sum_{r'=1}^{n_{k'}^N} \psi(X_{kr}, Y_{k'r'})$$

The kernel function $\Psi(X, Y)$ is defined by:

$$\psi\left(X, Y\right) = \begin{bmatrix} 1 & \text{if} & X < Y \\ 0.5 & \text{if} & X = Y \\ 0 & \text{if} & X > Y \end{bmatrix}$$

The ROIs are *effectively regarded as mini-cases* and one calculates the FOM as the Wilcoxon statistic considering the mini-cases as actual cases. The correlations between the ratings of ROIs on the same case are accounted for in the analysis.

9.5 Calculation of the ROI figure of merit.

```
UtilFigureOfMerit(datasetROI, FOM = "ROI")
#>           Rdr1      Rdr2      Rdr3      Rdr4      Rdr5
#> Trt1 0.9057239 0.8842834 0.8579279 0.9350207 0.8352103
#> Trt2 0.9297186 0.9546035 0.8937652 0.9531716 0.8770076
fom <- UtilFigureOfMerit(datasetROI, FOM = "ROI")
```

- If the correct `FOM` is not supplied, it defaults to `FOM = ROI`.
- This is a 2-treatment 5-reader dataset.
- For treatment 1, reader 1 the figure of merit is 0.9057239.
- For treatment 2, reader 5 the figure of merit is 0.8770076.
- Etc.

9.6 Significance testing

When `dataset$dataType == "ROI"` the FOM defaults to “ROI” (meaning the above formula) and the covariance estimation method defaults to `covEstMethod = "DeLong"`.

```
ret <- StSignificanceTesting(datasetROI, FOM = "Wilcoxon")
#> ROI dataset: forcing method = `ORH`, covEstMethod = `DeLong` and FOM = `ROI`.
str(ret)
#> List of 14
#>  $ fomArray             : num [1:2, 1:5] 0.906 0.93 0.884 0.955 0.858 ...
#>   ..- attr(*, "dimnames")=List of 2
#>   .. ..$ : chr [1:2] "Trt1" "Trt2"
#>   .. ..$ : chr [1:5] "Rdr1" "Rdr2" "Rdr3" "Rdr4" ...
#>  $ meanSquares          :'data.frame':    1 obs. of  3 variables:
#>   ..$ msT : num 0.00361
#>   ..$ msR : num 0.00256
#>   ..$ msTR: num 0.000207
#>  $ varComp              :'data.frame':    1 obs. of  6 variables:
#>   ..$ varR : num 0.00108
#>   ..$ varTR: num 0.000153
#>   ..$ cov1 : num 0.000247
#>   ..$ cov2 : num 0.000187
#>   ..$ cov3 : num 0.000154
#>   ..$ var  : num 0.000333
#>  $ FTestStatsRRRC       :'data.frame':    1 obs. of  4 variables:
#>   ..$ fRRRC  : num 9.76
#>   ..$ ndfRRRC: num 1
#>   ..$ ddfRRRC: num 12.8
#>   ..$ pRRRC  : num 0.00817
#>  $ ciDiffTrtRRRC        :'data.frame':    1 obs. of  8 variables:
#>   ..$ Treatment: chr "Trt1-Trt2"
#>   ..$ Estimate : num -0.038
#>   ..$ StdErr   : num 0.0122
#>   ..$ DF       : num 12.8
#>   ..$ t        : num -3.12
#>   ..$ PrGTt    : num 0.00817
#>   ..$ CILower  : num -0.0643
#>   ..$ CIUpper  : num -0.0117
#>  $ ciAvgRdrEachTrtRRRC :'data.frame':    2 obs. of  6 variables:
#>   ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>   ..$ Area     : num [1:2] 0.884 0.922
#>   ..$ StdErr   : num [1:2] 0.0232 0.0197
#>   ..$ DF       : num [1:2] 12.2 10.1
#>   ..$ CILower  : num [1:2] 0.833 0.878
#>   ..$ CIUpper  : num [1:2] 0.934 0.966
#>  $ FTestStatsFRRC      :'data.frame':    1 obs. of  4 variables:
#>   ..$ fFRRC  : num 16.6
#>   ..$ ndfFRRC: num 1
#>   ..$ ddfFRRC: num Inf
#>   ..$ pFRRC  : num 4.58e-05
#>  $ ciDiffTrtFRRC       :'data.frame':    1 obs. of  8 variables:
#>   ..$ Treatment: chr "Trt1-Trt2"
#>   ..$ Estimate : num -0.038
#>   ..$ StdErr   : num 0.00933
#>   ..$ DF       : num Inf
#>   ..$ t        : num -4.08
#>   ..$ PrGTt    : num 4.58e-05
#>   ..$ CILower  : num -0.0563
#>   ..$ CIUpper  : num -0.0197
#>  $ ciAvgRdrEachTrtFRRC :'data.frame':    2 obs. of  6 variables:
#>   ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>   ..$ Area     : num [1:2] 0.884 0.922
#>   ..$ StdErr   : num [1:2] 0.0163 0.0129
#>   ..$ DF       : num [1:2] Inf Inf
#>   ..$ CILower  : num [1:2] 0.852 0.896
#>   ..$ CIUpper  : num [1:2] 0.916 0.947
#>  $ ciDiffTrtEachRdrFRRC:'data.frame':    5 obs. of  9 variables:
#>   ..$ Reader   : Factor w/ 5 levels "Rdr1","Rdr2",..: 1 2 3 4 5
#>   ..$ Treatment: Factor w/ 1 level "Trt1-Trt2": 1 1 1 1 1
#>   ..$ Estimate : num [1:5] -0.024 -0.0703 -0.0358 -0.0182 -0.0418
#>   ..$ StdErr   : num [1:5] 0.01025 0.01448 0.01648 0.00928 0.01398
#>   ..$ DF       : num [1:5] Inf Inf Inf Inf Inf
#>   ..$ t        : num [1:5] -2.34 -4.86 -2.17 -1.96 -2.99
#>   ..$ PrGTt    : num [1:5] 1.93e-02 1.20e-06 2.97e-02 5.05e-02 2.79e-03
#>   ..$ CILower  : num [1:5] -0.0441 -0.0987 -0.0681 -0.0363 -0.0692
#>   ..$ CIUpper  : num [1:5] -3.90e-03 -4.19e-02 -3.53e-03 3.88e-05 -1.44e-02
#>  $ varCovEachRdr       :'data.frame':    5 obs. of  3 variables:
#>   ..$ Reader: Factor w/ 5 levels "Rdr1","Rdr2",..: 1 2 3 4 5
#>   ..$ Var   : num [1:5] 0.000269 0.000227 0.000481 0.000168 0.000522
#>   ..$ Cov1  : num [1:5] 0.000216 0.000122 0.000345 0.000125 0.000424
#>  $ FTestStatsRRFC      :'data.frame':    1 obs. of  4 variables:
#>   ..$ fRRFC  : num 17.5
#>   ..$ ndfRRFC: num 1
#>   ..$ ddfRRFC: num 4
#>   ..$ pRRFC  : num 0.0139
#>  $ ciDiffTrtRRFC       :'data.frame':    1 obs. of  8 variables:
#>   ..$ Treatment: chr "Trt1-Trt2"
#>   ..$ Estimate : num -0.038
#>   ..$ StdErr   : num 0.00909
#>   ..$ DF       : num 4
#>   ..$ t        : num -4.18
#>   ..$ PrGTt    : num 0.0139
#>   ..$ CILower  : num -0.0633
#>   ..$ CIUpper  : num -0.0128
#>  $ ciAvgRdrEachTrtRRFC :'data.frame':    2 obs. of  6 variables:
#>   ..$ Treatment: Factor w/ 2 levels "Trt1","Trt2": 1 2
#>   ..$ Area     : num [1:2] 0.884 0.922
#>   ..$ StdErr   : num [1:2] 0.0175 0.0157
#>   ..$ DF       : num [1:2] 4 4
#>   ..$ CILower  : num [1:2] 0.835 0.878
#>   ..$ CIUpper  : num [1:2] 0.932 0.965
```

- While `ret` is a list with many (22) members, their meanings should be clear from the notation. As an example:
- The variance components are given by:

```
ret$varComp
#>          varR        varTR         cov1         cov2         cov3          var
#> 1 0.001082359 0.0001526084 0.0002465125 0.0001870571 0.0001543764 0.0003333119
```

9.6.1 RRRC analysis

```
ret$FTestStatsRRRC$fRRRC
#> [1] 9.763602
ret$FTestStatsRRRC$ndfRRRC
#> [1] 1
ret$FTestStatsRRRC$ddfRRRC
#> [1] 12.82259
ret$FTestStatsRRRC$pRRRC
#> [1] 0.008173042
```

- The F-statistic is , with `ndf = 1` and `ddf` = , which yields a p-value of .
- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtRRRC
#>   Treatment    Estimate     StdErr       DF         t       PrGTt     CILower
#> 1 Trt1-Trt2 -0.03802005 0.01216768 12.82259 -3.124676 0.008173042 -0.06434373
#>       CIUpper
#> 1 -0.01169636
```

- The FOM difference (treatment 1 minus 2) is -0.03802, which is significant, p-value = 0.008173, F-statistic = 9.7636016, ddf = 12.8225898. The confidence interval is (-0.0643437, -0.0116964).

9.6.2 FRRC analysis

```
ret$FTestStatsFRRC$fFRRC
#> [1] 16.6135
ret$FTestStatsFRRC$ndfFRRC
#> [1] 1
ret$FTestStatsFRRC$ddfFRRC
#> [1] Inf
ret$FTestStatsFRRC$pFRRC
#> [1] 4.582365e-05
```

- The F-statistic is 16.6135014, with `ndf = 1` and `ddf` = `Inf`, which yields a p-value of 4.5823651×10^{-5}.
- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtFRRC
#>   Treatment    Estimate      StdErr  DF         t        PrGTt     CILower
#> 1 Trt1-Trt2 -0.03802005 0.009327861 Inf -4.075966 4.582365e-05 -0.05630232
#>       CIUpper
#> 1 -0.01973778
```

9.6.3 RRFC analysis

```
ret$FTestStatsRRFC$fRRFC
#> [1] 17.48107
ret$FTestStatsRRFC$ndfRRFC
#> [1] 1
ret$FTestStatsRRFC$ddfRRFC
#> [1] 4
ret$FTestStatsRRFC$pRRFC
#> [1] 0.01390667
```

- The F-statistic is 17.4810666, with `ndf = 1` and `ddf` = 4, which yields a p-value of 0.0139067.
- The confidence interval for the reader averaged difference between the two treatments is given by:

```
ret$ciDiffTrtRRFC
#>   Treatment    Estimate     StdErr DF         t      PrGTt     CILower
#> 1 Trt1-Trt2 -0.03802005 0.00909345  4 -4.181037 0.01390667 -0.06326751
#>       CIUpper
#> 1 -0.01277258
```

9.7 Summary

TBA

9.8 References

Bibliography

Chakraborty, D. P. (2017). *Observer Performance Methods for Diagnostic Imaging - Foundations, Modeling, and Applications with R-Based Examples*. CRC Press, Boca Raton, FL.

DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44:837–845.

Hillis, S. L. and Berbaum, K. S. (2004). Power estimation for the dorfman-berbaum-metz method. *Acad. Radiol.*, 11(11):1260–1273.

Metz, C. (1978). Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.

Obuchowski, N. A. (1997). Nonparametric analysis of clustered roc curve data. *Biometrics*, 53:567–578.

Obuchowski, N. A., Lieber, M. L., and Powell, K. A. (2000). Data analysis for detection and localization of multiple abnormalities with application to mammography. *Acad. Radiol.*, 7(7):516–525.

Obuchowski, N. A. and Rockette, H. (1995). Hypothesis testing of the diagnostic accuracy for multiple diagnostic tests: An anova approach with dependent observations. *Communications in Statistics: Simulation and Computation*, 24:285–308.