

RJafroc Documentation

Dev P. Chakraborty, PhD

2019-08-07

Contents

1	Preface	5
2	Introduction	7
3	ROC data format	9
3.1	Introduction	9
3.2	An actual MRMC ROC dataset	10
3.3	The ROC Excel data file	13
3.4	Summary	16
4	FROC data format	19
4.1	Introduction	19
4.2	An actual MRMC FROC dataset	20
4.3	The FROC Excel data input file	29
4.4	Summary	35
5	Split Plot Analysis	37
5.1	Introduction	37
6	ROI data format	39
6.1	ROI paradigm	39

Chapter 1

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 2

Introduction

- This is the book describing the **RJafroc** packages.
- 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

Chapter 3

ROC data format

3.1 Introduction

- In the receiver operating characteristic (**ROC**) paradigm (Metz, 1978) the observer’s task is to **rate** (i.e., assign an ordered label representing the degree of suspicion) each case for confidence in presence of disease. The rating is frequently called a *confidence level*.
- 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 case ¹.
- For human observer studies a 6-point rating scale is recommended, collected via two questions (Chakraborty, 2017):
 - Is the case diseased?
 - * Binary response: *Yes* or *No*.
 - What is your confidence in the preceding decisions?
 - * Three level response: *Low*, *Medium* or *High*.
- With algorithmic readers, e.g., computer aided detection (CAD) algorithms, a floating point rating, if possible, should be retained.
- In the most common study design, termed multiple-reader multiple-case (**MRMC**), the rating collection procedure is repeated for all cases, treatments and readers.

¹The directionality of the rating is not a limitation. If lower values correspond to increased confidence level, it is only necessary to transform the observed rating by subtracting it from a constant value. The constant value can be chosen arbitrarily, typically as the maximum of all observed ratings, thereby ensuring that the transformed value is always non-negative.

3.2 An actual MRMC ROC dataset

An actual MRMC ROC dataset (Van Dyke et al., 1993) is included as `dataset02`. It has the following structure:

```
str(dataset02)
#> List of 8
#> $ NL      : num [1:2, 1:5, 1:114, 1] 1 3 2 3 2 2 1 2 3 2 ...
#> $ LL      : num [1:2, 1:5, 1:45, 1] 5 5 5 5 5 5 5 5 5 5 ...
#> $ lesionNum : int [1:45] 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionID  : num [1:45, 1] 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionWeight: num [1:45, 1] 1 1 1 1 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" ...
```

3.2.1 Overview of the data structure

- The `dataset` structure is a `list` variable with 8 members ².
 - Ratings of non-diseased cases are stored in the `NL` list member.
 - Ratings of diseased cases are stored in the `LL` list member.
 - The `lesionNum` list member is an array of length 45, filled with ones. It lists the number of lesions per case, which for ROC data, is always unity. The length of this array equals the number of diseased cases K2, see below.
 - The `lesionID` list member is a [45 x 1] array, also filled with ones.³
 - The `LesionWeight` list member is also a [45 x 1] array filled with ones.
 - The `dataType` list member equals the string "ROC", identifying it as an ROC dataset.
 - The `modalityID` list member is a string array identifying the names of the treatments (see below).
 - The `readerID` list member is a string array, identifying the names of the readers (see below).

3.2.2 Details of the `modalityID` and `readerID` list members

- The names of the treatments are in the `modalityID` list member:

²This is true for ROC, FROC and ROI datasets. LROC datasets have 9 `list` members.

³The second "unnecessary" dimension is necessary for compatibility with FROC datasets.

```
attributes(dataset02$modalityID)
#> $names
#> [1] "0" "1"
```

- For example, the name of the first treatment is "0". The names can be longer strings, but use of very long string names may mess up the output formats of the analysis report. As per the **KISS** principle ⁴, keep the names short.
- The names of the readers are in the `readerID` array:

```
attributes(dataset02$readerID)
#> $names
#> [1] "0" "1" "2" "3" "4"
```

For example, the name of the second reader is "1". A similar caveat regarding long reader names applies.

3.2.3 Details of the NL and LL list members

- For either NL or LL list members, the fourth dimension has unit length. This dimension, which is strictly speaking unnecessary for ROC data, is retained for ease of generalizability to the FROC and ROC paradigms, where more than one rating per case is possible.
- `dataset02` is a 2-treatment 5-reader dataset (the lengths of the first and second dimensions, respectively, of the NL and LL list members).

3.2.3.1 Numbers of non-diseased and diseased cases

```
K <- length(dataset02$NL[1,1,,1])
K2 <- length(dataset02$LL[1,1,,1])
K1 <- K - K2
```

- K1 is the number of non-diseased cases, while K2 is the number of diseased cases.
- The third dimension of the NL array is the total number of **all** cases, i.e., $K = 114$, and the third dimension of the LL array, i.e., $K2 = 45$, is the total number of diseased cases.
- Subtracting the number of diseased cases from the number of all cases yields the number of non-diseased cases.

⁴For those not familiar with it, KISS is American for **Keep It Simple, Stupid**.

- Therefore, in this dataset, there are **45** diseased cases and **69** non-diseased cases.

3.2.3.2 Why dimension the NL array for the total number of cases?

- Again, this is for ease of generalizability to the FROC and ROI paradigms.

3.2.3.3 Ratings on a non-diseased case

- For ROC data a non-diseased case can have only one, and exactly one, NL rating.
- For treatment 1, reader 1 and case 1 (the first non-diseased case), the NL rating is "1":

```
dataset02$NL[1,1,1,1]
#> [1] 1
mean(dataset02$NL[,1:K1,1])
#> [1] 1.784058
```

- This study utilized a 5-point rating scale, 1 thru 5, so non-diseased cases are expected to have low ratings; in this case the lowest rating was observed.
- The mean rating over all non-diseased cases, treatments and readers, is 1.784058.

3.2.3.4 Ratings on a diseased case

- For ROC data a diseased case can have only one, and exactly one, LL rating.
- For treatment 1, reader 1, case 1 (the first diseased case) the LL rating is:

```
dataset02$LL[1,1,1,1]
#> [1] 5
mean(dataset02$LL)
#> [1] 4.297778
```

- As noted previously, this study utilized a 5-point rating scale, 1 thru 5, so diseased cases are expected to have high ratings; in this case the highest rating was observed.
- The mean rating over all diseased cases, treatments and readers, is 4.297778.

3.3 The ROC Excel data file

- An Excel file in JAFROC format containing ROC data corresponding to `dataset02` is included with the **RJafroc** package. The first command (below) finds the location of the file and the second command reads it and saves it to a dataset object `ds`.

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

- On my Mac, the file name is huge:
- `/Library/Frameworks/R.framework/Versions/3.6/Resources/library/RJafroc/extdata/includedRocData.xlsx`.
- Use the following code to make a duplicate of the Excel file in a more convenient location.

```
DfSaveDataFile(dataset02, "MyDataset02.xlsx", format = "JAFROC")
#> Note: zip::zip() is deprecated, please use zip::zipr() instead
```

- `DfReadDataFile` is short for *Data File Function to Read a Data File*.
- All data file related functions start with `Df`, and a similar organization applies to other functions. This makes it easier, in my opinion, to find a function in the R help system.
- To see the online help files, use the following command:

```
help("RJafroc-package")
```

- Click the "Show in new window" button to see it full screen (i.e., not constrained to a single quarter panel in **RStudio**).
- This advice applies, of course, to all help files.

3.3.1 The ROC Excel data input file

- It contains three worksheets, **Truth**, **TP** and **FP**.
- The **Truth** worksheet defines the ground-truth of each case. It indicates which cases are diseased and which are non-diseased.
- The **FP** worksheet lists the ratings of non-diseased cases.
- The **TP** worksheet lists the ratings of diseased cases ⁵.

⁵OK, I am being inconsistent. I am using NL, LL for the ratings and FP, TPj for the worksheets. The worksheet format will accept NL and LL instead of FP and TP. However, for ease of generalization to the FROC paradigm it is necessary to use NL and LL for the list members of the dataset object.

3.3.1.1 The Truth worksheet organization

- The **CaseID** column lists the numeric labels identifying each case. Again, string names are possible, but keep them short.
- A 1 in the **LesionID** column denotes a diseased case.
- A 0 in the **LesionID** column denotes a non-diseased case.
- The **Weight** column is irrelevant for ROC data ⁶.
- The contents of the **Truth** worksheet corresponding to **dataset02** are displayed next:

CaseID	LesionID	Weight
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0
41	0	0
42	0	0
43	0	0
44	0	0
45	0	0
46	0	0
47	0	0
48	0	0
49	0	0
50	0	0
51	0	0
52	0	0
53	0	0
54	0	0
55	0	0
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0	0
62	0	0
63	0	0
64	0	0
65	0	0
66	0	0
67	0	0
68	0	0
69	0	0
70	1	1
71	1	1
72	1	1
73	1	1
74	1	1
75	1	1
76	1	1
77	1	1
78	1	1
79	1	1
80	1	1
81	1	1
82	1	1
83	1	1
84	1	1
85	1	1
86	1	1
87	1	1
88	1	1
89	1	1
90	1	1
91	1	1
92	1	1
93	1	1
94	1	1
95	1	1
96	1	1
97	1	1
98	1	1
99	1	1
100	1	1
101	1	1
102	1	1
103	1	1
104	1	1
105	1	1
106	1	1
107	1	1
108	1	1
109	1	1
110	1	1
111	1	1
112	1	1
113	1	1
114	1	1

- There are 69 non-diseased cases (labeled 1-69) under column **CaseID**.
- There are 45 diseased cases (labeled 70-114) under column **CaseID**.
- The **LesionID** field for each non-diseased case (e.g., **CaseID** = 1) is zero. A zero in this field defines a non-diseased case.
- The **LesionID** field for each diseased case (e.g., **CaseID** = 70) is unity. A unit value in this field defines a diseased case.
- The **Weights** field is irrelevant for ROC datasets. For convenience it is filled with zeroes.

⁶It is only needed for FROC data.

3.3.1.2 The FP/NL worksheet organization

The following screen-shots show different parts of the FP worksheet for dataset02.

The screenshots show the FP worksheet for dataset02. The data is organized in a grid with 69 rows and 2 columns of ratings, totaling 138 data points. The columns are labeled A through H, and the rows are numbered 1 through 69. The data is organized in a grid with 69 rows and 2 columns of ratings, totaling 138 data points.

- The FP (or NL) worksheet lists the ratings of non-diseased cases.
- The **ModalityID** values range from 0 to 1, corresponding to two treatments.
- The **ReaderID** values range from 0 to 4, corresponding to five readers.
- The **CaseID** values range from 1 to 69, corresponding to non-diseased cases **only**.
- For each reader and treatment, each non-diseased case gets one rating; therefore the length of the column labeled **FP-Rating** is **69 x 2 x 5 = 690**.
- The FP ratings tend to be low, there are a lot of ones, fewer twos, even fewer threes, and an occasional four and a five rating may be found.

3.3.1.3 The TP/LL worksheet organization

The following screen-shots show different parts of the FP worksheet for dataset02.

* The TP (or LL) worksheet lists the ratings of diseased cases. * The **ModalityID** values range from 0 to 1, corresponding to two treatments. * The **ReaderID** values range from 0 to 4, corresponding to five readers. * The **CaseID** values range from 70 to 114, corresponding to diseased cases **only**. * For each reader and treatment, each non-diseased case gets one rating; therefore the length of the column labeled **FP-Rating** is $45 \times 2 \times 5 = 450$. * The TP ratings tend to be high, there are a lot of fives, fewer fours, even fewer threes, and an occasional two and a one rating may be found.

3.4 Summary

- Since each case gets one rating, the ROC data structure is relatively easy to visualize. For a single treatment and single reader, all of the information in the dataset can be summarize by a two-row five-column table, with one row listing the number of non-diseased cases rated 1, the number rated two, etc., ending with the number rated five, and a corresponding row for diseased cases. These 10 values contain all of the information contained in the Excel file for the specified treatment and reader. The example below is for treatment 0 and reader 0:

```
n1 <- dataset02$NL
binnedFpCounts <- array(5)
for (b in 1:5) binnedFpCounts[b] <- sum(n1[1,1,1:K1,1] == b)
```



```
ll <- dataset02$LL
binnedTpCounts <- array(5)
for (b in 1:5) binnedTpCounts[b] <- sum(ll[1,1,1:K2,1] == b)
binnedFpCounts
#> [1] 47 9 10 2 1
binnedTpCounts
#> [1] 4 1 2 10 28
sum(binnedFpCounts)
#> [1] 69
sum(binnedTpCounts)
#> [1] 45
```

- The values in `binnedFpCounts` sum to 69.
- The values in `binnedTpCounts` sum to 45.
- A similar table is needed for each treatment-reader combination.
- The real value of the Excel format is that it allows generalization to other paradigms where the number of ratings per case is variable.

Chapter 4

FROC data format

4.1 Introduction

- In the free-response ROC (**FROC**) paradigm (Bunch et al., 1978) the observer’s task is to:
 - **mark** (i.e., indicate the location of) and
 - **rate** (i.e., assign an ordered label representing the degree of suspicion) regions in the image that are perceived as suspicious for presence of disease.
 - Accordingly, FROC data consists of **mark-rating pairs**, where each mark indicates a region ¹ that was considered suspicious for presence of a localized lesion and the rating is the corresponding confidence level.
 - The number of mark-rating pairs on any particular case is a-priori unpredictable. It is a non-negative random integer (i.e., 0, 1, 2, ...) that depends on the case, the reader and the modality. The relatively unstructured nature of FROC data makes FROC paradigm data seemingly more difficult to analyze than ROC paradigm data ².
- By adopting a proximity criterion, each mark is classified by the investigator as a lesion localization (LL) - if it is close to a real lesion - or a non-lesion localization (NL) otherwise.
- 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 ³.

¹In order to avoid confusion with the region-of-interest or ROI-paradigm, I do not like to use the term ROI to describe the marks made by the observer.

²I say “seemingly”, because the only real difference between ROC and FROC analyses is in the selection of the figure of merit.

³As with the ROC paradigm, the directionality of the rating is not a limitation.

- For human observer studies a 4 or 5-point rating scale is recommended:
 - 1: Very low, but finite possibility that the region is diseased.
 - 2: Low possibility that the region is diseased.
 - 3: Moderate possibility that the region is diseased.
 - 4: High possibility that the region is diseased.
 - 5: Very high possibility that the region is diseased.
- The actual adjectives used to describe the labels are unimportant. What is important is the ordering of the labels and that the observer holds them relatively constant for the duration of the study. More allowed ratings, provided the observer can work with them, leads to greater definition of the relevant empirical operating curves (to be introduced later).
- With algorithmic readers, e.g., computer aided detection (CAD) algorithms, a floating point rating, if possible, should be retained.
- In the most common study design, termed multiple-reader multiple-case (MRMC), the rating collection procedure is repeated for all cases, treatments and readers.

4.2 An actual MRMC FROC dataset

An actual MRMC FROC dataset is included as `dataset04` (Zanca et al., 2009). It has the following structure:

```
str(dataset04)
#> List of 8
#> $ NL          : num [1:5, 1:4, 1:200, 1:7] -Inf -Inf 1 -Inf -Inf ...
#> $ LL          : num [1:5, 1:4, 1:100, 1:3] 4 5 4 5 4 3 5 4 4 3 ...
#> $ lesionNum   : int [1:100] 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionID    : num [1:100, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionWeight: num [1:100, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
#> $ dataType    : chr "FROC"
#> $ modalityID  : Named chr [1:5] "1" "2" "3" "4" ...
#> ..- attr(*, "names")= chr [1:5] "1" "2" "3" "4" ...
#> $ readerID    : Named chr [1:4] "1" "3" "4" "5"
#> ..- attr(*, "names")= chr [1:4] "1" "3" "4" "5"
```

The `dataset` structure is a `list` variable with 8 members.

4.2.1 The NL aand LL list members

- Ratings of actually non-diseased regions are stored in the NL list member.
- Ratings of actually diseased regions are stored in the LL list member.
- `dataset04` corresponds to 5-treatments and 4-readers (the lengths of the first and second dimensions, respectively, of the NL and LL arrays).

4.2.2 The `lesionNum` list member

- The `lesionNum` list member is an array of length 100 filled with integers ranging from 1 to 3, the latter being the maximum number of actual lesions per case in `dataset04`. The length of this array equals the number of diseased cases, 100 in the current example. The contents of `lesionNum` are shown below:

```
dataset04$lesionNum[1:20]
#> [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
dataset04$lesionNum[21:40]
#> [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
dataset04$lesionNum[41:60]
#> [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2
dataset04$lesionNum[61:80]
#> [1] 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 2 2 2 2 1
dataset04$lesionNum[81:100]
#> [1] 2 2 2 2 3 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
```

- The above entries tell us that while most cases contain only one lesion each, some contain 2 or even 3 lesions per case.

4.2.3 The `lesionID` list member

- The `lesionID` list member is a [100 x 3] array.
- Essentially it establishes a way of distinguishing between different lesions on a case by naming them, or what amounts to the same thing, by labeling them.
- The problem of distinguishing between different lesions on a case is peculiar to the FROC paradigm. With only one conceptual lesion per diseased case, the ROC paradigm does not face this problem.
- The second dimension of this array indicates that there is at least one diseased case with three lesions.

```
dataset04$lesionID[1:10,]
#>      [,1] [,2] [,3]
#> [1,]    1 -Inf -Inf
#> [2,]    1 -Inf -Inf
#> [3,]    1 -Inf -Inf
#> [4,]    1 -Inf -Inf
#> [5,]    1 -Inf -Inf
#> [6,]    1 -Inf -Inf
#> [7,]    1 -Inf -Inf
#> [8,]    1 -Inf -Inf
```

```
#> [9,]      1 -Inf -Inf
#> [10,]     1 -Inf -Inf
```

- This indicates that the first ten diseased cases contain one lesion each.
- The lesion on case 1 is **labeled** by the value 1. The `-Inf` denote missing values. Since there is only one lesion, the placeholders for the second or third lesion (not present on this case, but needed to hold lesion labels in other cases) need to be filled with negative infinities.
- The following example may help clarify this point.

```
dataset04$lesionID[90:100,]
#>      [,1] [,2] [,3]
#> [1,]     1     2 -Inf
#> [2,]     1     2     3
#> [3,]     1     2     3
#> [4,]     1     2     3
#> [5,]     1     2     3
#> [6,]     1     2     3
#> [7,]     1     2     3
#> [8,]     1     2     3
#> [9,]     1     2     3
#> [10,]    1     2     3
#> [11,]    1     2     3
```

- Diseased case 90 has two lesions, labeled 1 and 2 respectively.
- The key point is this: each lesion on a case has a distinct *name*. Just as each case has to have a distinct name (i.e., label), each lesion within a (diseased) case has to have a distinct name.
- When an observer assigns a rating to a particular lesion on a case, the experimenter needs to record this information correctly.
- For example, if the lesion with `lesionID = 1` is marked and rated a particular value, this value needs to be entered in the spread-sheet as belonging to the lesion named `lesionID = 1`.
- The TP/LL worksheet has a `lesionID` column.⁴
- The distinction implied by different `lesionID` values is important if the lesion weights are unequal.⁵

⁴Since the ROC paradigm does not allow multiple lesions per case, each diseased case conceptually containing only one lesion, the distinction between different `lesionID` values on the same diseased case does not arise.

⁵For equally weighted lesions, the name distinction implied by `lesionID` is not important, but then the analysis would only be valid for equally weighted lesions.

4.2.4 The `LesionWeight` list member

- The `LesionWeight` list member is also a [100 x 3] array filled with values that add up to unity for each case. The meaning of lesion weights is discussed in here: (Chakraborty and Berbaum, 2004; Chakraborty and Yoon, 2009; Chakraborty, 2017)
- Briefly, the lesion weights are the clinical importance of detecting the lesion.
- As an example, a highly visible lesion might have low clinical significance if it is likely to be benign, and it would be characterized by a lower weight than a less visible but more deadly lesion on the same case.
- In order to give each case equal importance, the weights must sum to one.
- In the current example, see below, the lesions are equally weighted.
- The choice of weights is a clinical consideration, determined by the costs and benefits of missing or finding the lesion. Lacking this information, which is the most common scenario, it makes sense to weight the lesions equally.
- On a case with one lesion, `lesionWeight = 1`, and the other `lesionWeights` are assigned negative infinity values.
- On a case with two lesions, `lesionWeight = 0.5` for the first two lesions, and the third weight is assigned the negative infinity value.
- On a case with three lesions, `lesionWeight = 0.3333` for the lesions.
- Rather than assign these values manually, set the `Weight` column of the `Truth` worksheet to zeroes. Then the software automatically assigns equal weights when the Excel sheet is read.

```
dataset04$lesionWeight[1:20,]
#>      [,1] [,2] [,3]
#> [1,]    1 -Inf -Inf
#> [2,]    1 -Inf -Inf
#> [3,]    1 -Inf -Inf
#> [4,]    1 -Inf -Inf
#> [5,]    1 -Inf -Inf
#> [6,]    1 -Inf -Inf
#> [7,]    1 -Inf -Inf
#> [8,]    1 -Inf -Inf
#> [9,]    1 -Inf -Inf
#> [10,]   1 -Inf -Inf
#> [11,]   1 -Inf -Inf
#> [12,]   1 -Inf -Inf
#> [13,]   1 -Inf -Inf
#> [14,]   1 -Inf -Inf
#> [15,]   1 -Inf -Inf
#> [16,]   1 -Inf -Inf
#> [17,]   1 -Inf -Inf
```

```

#> [18,] 1 -Inf -Inf
#> [19,] 1 -Inf -Inf
#> [20,] 1 -Inf -Inf
dataset04$lesionWeight[21:40,]
#>      [,1] [,2] [,3]
#> [1,] 1 -Inf -Inf
#> [2,] 1 -Inf -Inf
#> [3,] 1 -Inf -Inf
#> [4,] 1 -Inf -Inf
#> [5,] 1 -Inf -Inf
#> [6,] 1 -Inf -Inf
#> [7,] 1 -Inf -Inf
#> [8,] 1 -Inf -Inf
#> [9,] 1 -Inf -Inf
#> [10,] 1 -Inf -Inf
#> [11,] 1 -Inf -Inf
#> [12,] 1 -Inf -Inf
#> [13,] 1 -Inf -Inf
#> [14,] 1 -Inf -Inf
#> [15,] 1 -Inf -Inf
#> [16,] 1 -Inf -Inf
#> [17,] 1 -Inf -Inf
#> [18,] 1 -Inf -Inf
#> [19,] 1 -Inf -Inf
#> [20,] 1 -Inf -Inf
dataset04$lesionWeight[41:60,]
#>      [,1] [,2] [,3]
#> [1,] 1.0 -Inf -Inf
#> [2,] 1.0 -Inf -Inf
#> [3,] 1.0 -Inf -Inf
#> [4,] 1.0 -Inf -Inf
#> [5,] 1.0 -Inf -Inf
#> [6,] 1.0 -Inf -Inf
#> [7,] 1.0 -Inf -Inf
#> [8,] 1.0 -Inf -Inf
#> [9,] 1.0 -Inf -Inf
#> [10,] 1.0 -Inf -Inf
#> [11,] 1.0 -Inf -Inf
#> [12,] 1.0 -Inf -Inf
#> [13,] 1.0 -Inf -Inf
#> [14,] 1.0 -Inf -Inf
#> [15,] 1.0 -Inf -Inf
#> [16,] 0.5 0.5 -Inf
#> [17,] 0.5 0.5 -Inf
#> [18,] 1.0 -Inf -Inf

```



```

#> [19,] 1.0 -Inf -Inf
#> [20,] 0.5 0.5 -Inf
dataset04$lesionWeight[61:80,]
#>      [,1] [,2] [,3]
#> [1,] 1.0 -Inf -Inf
#> [2,] 1.0 -Inf -Inf
#> [3,] 1.0 -Inf -Inf
#> [4,] 1.0 -Inf -Inf
#> [5,] 1.0 -Inf -Inf
#> [6,] 1.0 -Inf -Inf
#> [7,] 1.0 -Inf -Inf
#> [8,] 1.0 -Inf -Inf
#> [9,] 0.5 0.5 -Inf
#> [10,] 0.5 0.5 -Inf
#> [11,] 1.0 -Inf -Inf
#> [12,] 1.0 -Inf -Inf
#> [13,] 1.0 -Inf -Inf
#> [14,] 0.5 0.5 -Inf
#> [15,] 0.5 0.5 -Inf
#> [16,] 0.5 0.5 -Inf
#> [17,] 0.5 0.5 -Inf
#> [18,] 0.5 0.5 -Inf
#> [19,] 0.5 0.5 -Inf
#> [20,] 1.0 -Inf -Inf
dataset04$lesionWeight[81:100,]
#>      [,1]      [,2]      [,3]
#> [1,] 0.5000000 0.5000000 -Inf
#> [2,] 0.5000000 0.5000000 -Inf
#> [3,] 0.5000000 0.5000000 -Inf
#> [4,] 0.5000000 0.5000000 -Inf
#> [5,] 0.3333333 0.3333333 0.3333333
#> [6,] 0.5000000 0.5000000 -Inf
#> [7,] 0.5000000 0.5000000 -Inf
#> [8,] 0.5000000 0.5000000 -Inf
#> [9,] 0.5000000 0.5000000 -Inf
#> [10,] 0.5000000 0.5000000 -Inf
#> [11,] 0.3333333 0.3333333 0.3333333
#> [12,] 0.3333333 0.3333333 0.3333333
#> [13,] 0.3333333 0.3333333 0.3333333
#> [14,] 0.3333333 0.3333333 0.3333333
#> [15,] 0.3333333 0.3333333 0.3333333
#> [16,] 0.3333333 0.3333333 0.3333333
#> [17,] 0.3333333 0.3333333 0.3333333
#> [18,] 0.3333333 0.3333333 0.3333333
#> [19,] 0.3333333 0.3333333 0.3333333

```

```
#> [20,] 0.3333333 0.3333333 0.3333333
```

4.2.5 The `dataType` list member

- The `dataType` list member equals the string "FROC", identifying `dataset04` as an FROC dataset.

Examination of the output reveals that:

- Location-level ratings of non-diseased regions are stored in the NL list member.
- Location-level ratings of diseased regions are stored in the LL list member.

4.2.6 Details of the `modalityID` and `readerID` list members

- The names of the treatments are in the `modalityID` list member:

```
attributes(dataset04$modalityID)
#> $names
#> [1] "1" "2" "3" "4" "5"
```

- For example, the name of the second treatment is "2". The names can be longer strings, but use of very long string names may mess up the output formats of the analysis report.
- The names of the readers are in the `readerID` array:

```
attributes(dataset04$readerID)
#> $names
#> [1] "1" "3" "4" "5"
```

For example, the name of the second reader is "3". Apparently reader "2" "dropped out" of the study. A similar caveat regarding long reader names applies.

4.2.7 Details of the NL and LL list members

- For either NL or LL list members, the fourth dimension can have length greater than unity.
- For the NL list member **this length is determined by the treatment-reader-case combination yielding the most NL marks per case.**

- For the LL list member **this length is determined by the case with the most true lesions.**
- **dataset02** is a 2-treatment 5-reader dataset (the lengths of the first and second dimensions, respectively, of the NL and LL list members).

4.2.8 Numbers of non-diseased and diseased cases

```
length(dataset04$NL[1,1,,1])
#> [1] 200
length(dataset04$LL[1,1,,1])
#> [1] 100
```

- The third dimension of the NL array is the total number of **all** cases, i.e., 200, and the third dimension of the LL array, i.e., 100, is the total number of diseased cases.
- Subtracting the number of diseased cases from the number of all cases yields the number of non-diseased cases.
- Therefore, in this dataset, there are 100 diseased cases and 100 non-diseased cases.

4.2.9 Why dimension the NL array for the total number of cases?

- Because, in addition to LLs, NLs are possible on diseased cases.
- Only LLs are possible on diseased cases.
- Only NLs are possible on non-diseased cases.
- The missing values are filled in with `-Inf`.

4.2.10 Ratings on a non-diseased case

- For treatment 1, reader 1 and case 1 (the first non-diseased case), the NL ratings are:

```
dataset04$NL[1,1,1,]
#> [1] -Inf -Inf -Inf -Inf -Inf -Inf -Inf
```

4.2.10.1 The meaning of a negative infinity rating

- Obviously, a real rating cannot be negative infinity⁶. This value is reserved for **missing ratings**, and more generally, **missing marks**⁷. For example, since all values in the above code chunk are negative infinities, this means this treatment-reader-case combination did not yield any mark-rating pairs. This possibility, alluded to above, is only possible with FROC data. All other paradigms (ROC, LROC and ROI) yield at least one rating per case.
- The length of the fourth dimension of the NL array is determined by that treatment-reader-case combination yielding the maximum number of NLs. Consider the following chunk:

```
for (i in 1:5)
  for (j in 1:4)
    for (k in 1:200)
      if (all(dataset04$NL[i,j,k,] != -Inf))
        cat(i, j, k, all(dataset04$NL[i,j,k,] != -Inf), "\n")
#> 5 4 192 TRUE
```

- This shows that the fourth dimension of the NL array has to be of length 7 because *one, and only reader*, specifically reader “4”, made 7 NL marks on a diseased case in treatment “5”!

4.2.10.2 Ratings on a diseased case

Unlike non-diseased cases, diseased cases can have both NL and LL ratings.

- For treatment 1, reader 1, case 51 (the 1st diseased case) the NL ratings are:

```
dataset04$NL[1,1,51,]
#> [1] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
dataset04$lesionNum[1]
#> [1] 1
dataset04$LL[1,1,1,]
#> [1] 4 -Inf -Inf
mean(is.finite(dataset04$LL))
#> [1] 0.3043333
```

⁶If an observer is so highly confident in the *absence* of a localized lesion, he will simply *not mark* the location in question; if he did, then, logically, he should mark *all* areas in the image that are definitely not lesions; in the FROC paradigm only regions with a reasonable degree of suspicion are marked. The radiologist only wishes to draw attention to regions that are reasonably suspicious; the definition of “reasonable” is determined by clinical considerations.

⁷Since there is a one-to-one correspondence between marks and ratings.

. There are only two finite values because this case has two ROI-level-abnormal regions, and 2 plus 2 makes for the assumed 4-regions per case. The corresponding `$lesionNum` field is 1.

```
mean(is.finite(dataset04$NL[,1:50,]))
#> [1] 0.05942857
dataset04$NL[1,1,51,]
#> [1] -Inf -Inf -Inf -Inf -Inf -Inf -Inf
dataset04$lesionNum[1]
#> [1] 1
dataset04$LL[1,1,1,]
#> [1] 4 -Inf -Inf
mean(is.finite(dataset04$LL))
#> [1] 0.3043333
```

```
mean(is.finite(dataset04$NL[,1:50,]))
#> [1] 0.05942857
dataset04$NL[1,1,51,]
#> [1] -Inf -Inf -Inf -Inf -Inf -Inf -Inf
dataset04$lesionNum[1]
#> [1] 1
dataset04$LL[1,1,1,]
#> [1] 4 -Inf -Inf
mean(is.finite(dataset04$LL))
#> [1] 0.3043333
```

- The ratings of the 2 LLs on this case are 4. The mean rating over all LLs (in all treatments and readers) is 3.6785323.

```
mean(is.finite(dataset04$NL[,1:50,]))
#> [1] 0.05942857
dataset04$NL[1,1,51,]
#> [1] -Inf -Inf -Inf -Inf -Inf -Inf -Inf
dataset04$lesionNum[1]
#> [1] 1
dataset04$LL[1,1,1,]
#> [1] 4 -Inf -Inf
mean(is.finite(dataset04$LL))
#> [1] 0.3043333
```

4.3 The FROC Excel data input file

- An Excel file in JAFROC format containing FROC data corresponding to `dataset04` is included with the **RJafroc** package (Zanca et al., 2009).

```

fileName <- system.file(
  "extdata", "includedFrocData.xlsx", package = "RJafroc", mustWork = TRUE)
ds <- DfReadDataFile(fileName)
ds$dataType
#> [1] "FROC"

```

- On my Mac, the file name is huge:
- /Library/Frameworks/R.framework/Versions/3.6/Resources/library/RJafroc/extdata/includedFrocData.xlsx
- Use the following code to make a duplicate of the Excel file in a more convenient location.

```

DfSaveDataFile(dataset04, "MyDataset04.xlsx", format = "JAFROC")
#> Note: zip::zip() is deprecated, please use zip::zipr() instead

```

- The structure of the Excel file is superficially similar to the ROC Excel file considered in the previous chapter. However, there are several important differences noted below **in bold**.
 - The **Truth** worksheet defines the ground-truth of each case. It indicates which cases are diseased and which are non-diseased. **For diseased cases, it additionally indicates the number of lesions, and the weight to be assigned to each lesion.**
 - The **FP** worksheet lists the **number (zero or more)** and ratings of NL marks on non-diseased **and diseased cases**.
 - The **TP** worksheet lists the **number (zero or more)** and ratings of LL marks, i.e., **marked lesions**, on diseased cases. Obviously, LL marks cannot occur on non-diseased cases.

4.3.1 The Truth worksheet

- The **CaseID** column lists the numeric labels identifying each case. Again, string names are possible, but keep them short.
- A 1, or more, in the **LesionID** column denotes a diseased case.
- A 0 in the **LesionID** column denotes a non-diseased case.
- The **Weight** column is relevant for FROC data.
- The contents of the **Truth** worksheet corresponding to **dataset04** are displayed next:

CaseID	LesionID	Weight
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1
36	1	1
37	1	1
38	1	1
39	1	1
40	1	1
41	1	1
42	1	1
43	1	1
44	1	1
45	1	1
46	1	1
47	1	1
48	1	1
49	1	1
50	1	1
51	1	1
52	1	1
53	1	1
54	1	1
55	2	0.5
56	2	0.5
57	2	0.5
58	2	0.5
59	2	0.5
60	2	0.5
61	2	0.5
62	2	0.5
63	2	0.5
64	2	0.5
65	2	0.5
66	2	0.5
67	2	0.5
68	3	0.33
69	3	0.33
70	3	0.33
71	3	0.33
72	3	0.33
73	3	0.33
74	3	0.33
75	3	0.33
76	3	0.33
77	3	0.33
78	0	0
79	0	0
80	0	0
81	0	0
82	0	0
83	0	0
84	0	0
85	0	0
86	0	0
87	0	0
88	0	0
89	0	0
90	0	0
91	0	0
92	0	0
93	0	0
94	0	0
95	0	0
96	0	0
97	0	0
98	0	0
99	0	0

- There are 100 diseased cases (labeled 0-99) under column **CaseID** and 100 non-diseased cases (labeled 100-199).⁸
- The **LesionID** field for each non-diseased case is zero and there is one row per case for such cases. **For diseased cases, this field has one or more entries, ranging from 1 to 3 for this particular dataset.** In other words, for each diseased case, the number of rows equals the number of lesion on the case.
- As an example, cases labeled 0 - 54 (and other cases like 60 - 67, etc.) have single lesions each, with **LesionID** = 1 and **Weight** = 1 have one row per case in the worksheet.
- As another example, there are two rows for **CaseID** = 77: one with **LesionID** = 1 (labeling the first lesion on this case) and one with **LesionID** = 2 (labeling the second and last lesion on this case). The weights of these lesions are explicitly specified as 0.5 each.

⁸The non-diseased cases numbered 128 - 199 are not shown above. They are similar to the ones that are shown - one row per case with a zero under the lesionID column and a zero under the Weights column.

- As a final example, there are three rows for `CaseID = 95`: one with `LesionID = 1`, one with `LesionID = 2` and the last with `LesionID = 3`. The weights of these lesions are explicitly specified to be 0.33 each.⁹
- Alternatively, the `Weights` field can be set to zeroes (for all cases) to more conveniently ensure equal weighting to much higher precision.
- Important: every case must have at least one row describing it in the **Truth** worksheet.
- The Excel files should be “robust” with respect to sorting on different columns. By “robust” I mean the resulting R dataset object, resulting from `DfReadDataFile`, should be unchanged. I have not tested this on all files, but if someone brings deviations from this statement to my attention, I will look into it. As a simple example of “robustness”, notice that in the **Truth** worksheet being currently explained, the diseased cases come before the non-diseased ones.

4.3.2 The FP/NL worksheet

The following screen-shots show different parts of the FP worksheet for `dataset04`.

⁹The software performs a check to ensure that the weight sum to unity, in this case 1% error is considered close enough for “government work”!

ReaderID	ModalityID	CaseID	FP_Rating
1	4	3	2
2	4	8	2
3	4	14	2
4	4	15	1
5	4	29	3
6	4	30	1
7	4	44	3
8	4	46	2
9	4	59	3
10	4	63	2
11	4	65	1
12	4	77	3
13	4	91	3
14	4	96	4
15	4	107	3
16	4	109	1
17	4	110	2
18	4	111	1
19	4	113	2
20	4	114	1
21	4	119	2
22	4	122	3
23	4	122	3
24	4	122	3
25	4	122	3
26	4	124	1
27	4	129	1
28	4	133	2
29	4	137	1
30	4	140	2
31	4	140	1
32	4	140	2
33	4	140	1
34	4	142	1
35	4	142	2
36	4	148	4
37	4	148	4
38	4	149	3
39	4	149	2
40	4	149	2
41	4	150	1
42	4	151	1
43	4	154	1
44	4	156	2
45	4	157	3

- The FP worksheet lists the ratings of NLs on both non-diseased and diseased cases. **Unlike the ROC paradigm, NLs can occur on diseased cases. Additionally, for a given treatment and reader, the number of rows per case cannot be predicted apriori. It could be 0, 1, 2, etc.** While there is in principle no upper limit to the number of NLs per case, radiologists seldom make more than one or two NLs on any case.
- Note the *absence* of the `LesionID` field. The NL marks do not correspond to real lesions, which by definition, can only occur on diseased cases.
- It is possible (in principle) that the FP worksheet is blank. The observer simply does not mark any NLs. See (Chakraborty, 2017) for how the FROC paradigm correctly interprets this situation as indicative of good performance.

4.3.3 The TP/LL worksheet

The following screen-shots show different parts of the TP worksheet for dataset04.

The image displays three screenshots of an Excel spreadsheet titled "includedFrocData.xlsx". The first screenshot shows the "TP" worksheet with columns A through E: ReaderID, ModalityID, CaseID, LesionID, and TP_Rating. The second screenshot shows the "FP" worksheet with columns A through E: CaseID, LesionID, TP_Rating, and two empty columns. The third screenshot shows the "TP" worksheet with columns A through E: CaseID, LesionID, TP_Rating, and two empty columns. The data is organized into rows, with some rows highlighted in blue.

- The TP worksheet lists the ratings of LLs on diseased cases. **Only diseased cases can appear on this worksheet. Additionally, for a given treatment and reader, the number of rows per case cannot be predicted apriori. It could be 0, 1, 2, 3** (the upper limit comes from the maximum number of lesions per case in this dataset).
- Note the *presence* of the LesionID field, which essentially names the different lesions on each diseased case.
- As an example, for reader 1, treatment 4, case 99, all three lesions (labeled 1, 2 and 3) are marked and rated 4, 5 and 5, respectively.

```
dataset04$lesionID[100]
#> [1] 1
dataset04$LL[4, 1, 100, ]
#> [1] 4 5 5
```

- The diseased case labeled 99 in the Excel Truth sheet is actually diseased case number 100 in the dataset04\$LL array.

- Remember that the diseased cases are labeled “0” thru “99”, but the array index in the dataset object runs from 1 to 100.
- As another example, for reader 1, treatment 4, case 3, there are no LL marks. The one lesion on this case went unmarked.

```
dataset04$lesionID[4]
#> [1] 1
dataset04$LL[4, 1, 4, ]
#> [1] -Inf -Inf -Inf
```

- Unmarked lesions are indicated by `-Inf`.
- It is possible (in principle) that the TP worksheet is blank. See (Chakraborty, 2017) for how the FROC paradigm correctly interprets this situation as indicative of poor performance.

4.4 Summary

TBA

Chapter 5

Split Plot Analysis

5.1 Introduction

Jason: Thanks for inviting me to join your profile. Could you describe what you were planning on doing? Please use a simple example that illustrates the basic data collection. The key point I want to be sure about is that strict pairing of readers across different treatments is preserved. I once had a study that had different readers interpreting in different treatments, which makes it impossible to separate the treatment effect (which is our primary interest) from a treatment-reader interaction effect. I am going to post that problem on Research Gate (the regulatory agency was telling them to do the study in a non-scientific way). Dev

Dr. Chakraborty, there will be 8 readers in my study. Around 800 CT images will be collected. Each reader reads 100 CT images unaided and aided by the CAde in two independent reading sessions separated by a washout period of 4 weeks. Each CT image will only be interpreted by one reader. I think it is a split-plot design with cases nested within reader so the strict pairing of readers across different treatments is preserved. Thanks for your help!

Hi Jason, This problem has been solved in the ROC context (localization information not used) by Hillis and others; they may even have software. To analyze it using localization, one needs to use a location-sensitive figure of merit, like the weighted AFROC. That would be my recommendation. So, for each reader j , one has a figure of merit θ_{ij} , where i is the modality index. One can average over j , yielding $\theta_{i\cdot}$. Significance testing can be performed in the usual manner - e.g., DBMH or ORH. A custom program would need to be written or one can construct one out of R-scripts using the existing functions in RJafroc (the Windows program is obsolete). I can help in this regard. I would ask you to read the relevant papers and explain to me if my approach, summarized above, is basically correct. In any case, any method for ROC analysis can

be adapted to FROC analysis if one uses the appropriate figure of merit. Good luck. Dev

Jason: Attached is the Hillis paper; I am not a statistician and find the term “nested” confusing; this paper claims to analyze split plot design using the OR approach as applied to ROC split plot data; if you think this is the correct approach for your study, apart from the limitation to ROC, I can easily extend the software to analyze FROC split plot data; please review and advise; Dev

Chapter 6

ROI data format

6.1 ROI paradigm

- 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). The ROIs are defined prior to the study and made known to all observers participating in the study. Unlike the FROC paradigm, a **rating is required for every ROI**.

Bibliography

- Bunch, P. C., Hamilton, J. F., Sanderson, G. K., and Simmons, A. H. (1978). A free-response approach to the measurement and characterization of radiographic-observer performance. *J of Appl Photogr. Eng.*, 4:166–171.
- Chakraborty, D. P. (2017). *Observer Performance Methods for Diagnostic Imaging - Foundations, Modeling, and Applications with R-Based Examples*. CRC Press, Boca Raton, FL.
- Chakraborty, D. P. and Berbaum, K. S. (2004). Observer studies involving detection and localization: Modeling, analysis and validation. *Med Phys*, 31(8):2313–2330.
- Chakraborty, D. P. and Yoon, H. J. (2009). Jafroc analysis revisited: figure-of-merit considerations for human observer studies. *Proc. SPIE Medical Imaging: Image Perception, Observer Performance, and Technology Assessment*, 7263:72630T.
- Metz, C. (1978). Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.
- Van Dyke, C., White, R., Obuchowski, N., Geisinger, M., Lorig, R., and Meziane, M. (1993). Cine mri in the diagnosis of thoracic aortic dissection. *79th RSNA Meetings*.
- Zanca, F., Jacobs, J., Van Ongeval, C., Claus, F., Celis, V., Geniets, C., Provost, V., Pauwels, H., Marchal, G., and Bosmans, H. (2009). Evaluation of clinical image processing algorithms used in digital mammography. *Medical Physics*, 36(3):765–775.