

RJafroc documentation

Dev P. Chakraborty

2019-08-05

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 file	29
5	ROI data format	33
5.1	ROI paradigm	33

Chapter 1

Preface

- This book, an extended documentation of the **RJafron** 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/RJafronBook>

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)
```

- `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.
- This bit of experiential advice applies, of course, to all help files.

3.3.1 The ROC Excel file organization

- 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.

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.

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.

⁶It is only needed for FROC data.

- 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
115	0	0
116	0	0
117	0	0
118	0	0
119	0	0
120	0	0
121	0	0
122	0	0
123	0	0
124	0	0
125	0	0
126	0	0
127	0	0
128	0	0
129	0	0
130	0	0
131	0	0
132	0	0
133	0	0
134	0	0
135	0	0
136	0	0
137	0	0
138	0	0
139	0	0
140	0	0
141	0	0
142	0	0
143	0	0
144	0	0
145	0	0
146	0	0

- 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.

3.3.1.2 The FP/NL worksheet organization

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

ReaderID	ModalityID	CaseID	FP_Rating
0	0	1	1
0	0	2	2
0	0	3	2
0	0	4	1
0	0	5	5
0	0	6	1
0	0	7	1
0	0	8	1
0	0	9	3
0	0	10	1
0	0	11	1
0	0	12	1
0	0	13	4
0	0	14	1
0	0	15	1
0	0	16	1
0	0	17	1
0	0	18	1
0	0	19	1
0	0	20	1
0	0	21	1
0	0	22	1
0	0	23	3
0	0	24	1
0	0	25	3
0	0	26	1
0	0	27	1
0	0	28	2
0	0	29	1
0	0	30	4
0	0	31	1
0	0	32	1
0	0	33	1
0	0	34	1
0	0	35	1
0	0	36	1
0	0	37	1
0	0	38	1
0	0	39	1
0	0	40	1
0	0	41	1
0	0	42	1
0	0	43	1
0	0	44	1
0	0	45	1
0	0	46	1
0	0	47	2
0	0	48	2
0	0	49	3
0	0	50	3
0	0	51	2
0	0	52	1
0	0	53	2
0	0	54	2
0	0	55	2
0	0	56	2
0	0	57	1
0	0	58	2
0	0	59	2
0	0	60	2
0	0	61	2
0	0	62	3
0	0	63	1
0	0	64	2
0	0	65	3
0	0	66	2
0	0	67	1
0	0	68	1
0	0	69	1

- 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 $69 \times 2 \times 5 = 690$.
- 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:
 - indicate (i.e., **mark** 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 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 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.
- 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

Such a dataset (Zanca et al., 2009) is included as `dataset04`. 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"
```

4.2.1 Overview of the FROC data structure

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

4.2.1.1 The `lesionNum` list member

- The `lesionNum` list member is an array of length 100, filled with integers ranging from 1 to the maximum number of actual lesions (per case) in the dataset. The length of this array equals the number of diseased cases `K2`, 100 in the current example. For this dataset, 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.

4.2.1.2 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.
- 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 that has 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 each contain one lesion.
- 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 possible 2 or 3 lesions (not present on this case, but needed to hold lesion labels in other cases) need to be filled with negative infinities.
- The previous paragraph may appear confusing. The following example may help clarify it.

```
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
```

- Now we see more interesting values.
- Diseased case 90 has two lesions.
- These are labeled 1 and 2 respectively.
- The key point is this: each lesion on a case has a *name*. The names are used in the data file creation process.
- 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 named “1” is marked and rated a particular value, this value needs to be entered in the spread-sheet as belonging to the lesion named “1”.
- As we saw for ROC data, the TP/LL worksheet has a `lesionID` column. Since the ROC paradigm does not allow multiple lesions per case, each diseased case conceptually containing one lesion, so the question of distinct lesion names of the same diseased case does not arise.
- At least one investigator made the mistake of assuming the lesion name, i.e., the `lesionID` field had no meaning, and randomly assigned the rating to the lesions.

4.2.1.3 The `LesionWeight` list member

- The `LesionWeight` list member is also a $[100 \times 3]$ array filled with values that add up to unity for each case.

```
dataset04$LesionWeight
#>           [,1]      [,2]      [,3]
#>  [1,] 1.0000000 -Inf -Inf
#>  [2,] 1.0000000 -Inf -Inf
#>  [3,] 1.0000000 -Inf -Inf
#>  [4,] 1.0000000 -Inf -Inf
#>  [5,] 1.0000000 -Inf -Inf
#>  [6,] 1.0000000 -Inf -Inf
#>  [7,] 1.0000000 -Inf -Inf
#>  [8,] 1.0000000 -Inf -Inf
#>  [9,] 1.0000000 -Inf -Inf
#> [10,] 1.0000000 -Inf -Inf
#> [11,] 1.0000000 -Inf -Inf
#> [12,] 1.0000000 -Inf -Inf
#> [13,] 1.0000000 -Inf -Inf
#> [14,] 1.0000000 -Inf -Inf
#> [15,] 1.0000000 -Inf -Inf
#> [16,] 1.0000000 -Inf -Inf
#> [17,] 1.0000000 -Inf -Inf
#> [18,] 1.0000000 -Inf -Inf
#> [19,] 1.0000000 -Inf -Inf
#> [20,] 1.0000000 -Inf -Inf
#> [21,] 1.0000000 -Inf -Inf
#> [22,] 1.0000000 -Inf -Inf
#> [23,] 1.0000000 -Inf -Inf
#> [24,] 1.0000000 -Inf -Inf
#> [25,] 1.0000000 -Inf -Inf
#> [26,] 1.0000000 -Inf -Inf
#> [27,] 1.0000000 -Inf -Inf
#> [28,] 1.0000000 -Inf -Inf
#> [29,] 1.0000000 -Inf -Inf
#> [30,] 1.0000000 -Inf -Inf
#> [31,] 1.0000000 -Inf -Inf
#> [32,] 1.0000000 -Inf -Inf
#> [33,] 1.0000000 -Inf -Inf
#> [34,] 1.0000000 -Inf -Inf
#> [35,] 1.0000000 -Inf -Inf
#> [36,] 1.0000000 -Inf -Inf
#> [37,] 1.0000000 -Inf -Inf
#> [38,] 1.0000000 -Inf -Inf
```

```

#> [39,] 1.0000000 -Inf -Inf
#> [40,] 1.0000000 -Inf -Inf
#> [41,] 1.0000000 -Inf -Inf
#> [42,] 1.0000000 -Inf -Inf
#> [43,] 1.0000000 -Inf -Inf
#> [44,] 1.0000000 -Inf -Inf
#> [45,] 1.0000000 -Inf -Inf
#> [46,] 1.0000000 -Inf -Inf
#> [47,] 1.0000000 -Inf -Inf
#> [48,] 1.0000000 -Inf -Inf
#> [49,] 1.0000000 -Inf -Inf
#> [50,] 1.0000000 -Inf -Inf
#> [51,] 1.0000000 -Inf -Inf
#> [52,] 1.0000000 -Inf -Inf
#> [53,] 1.0000000 -Inf -Inf
#> [54,] 1.0000000 -Inf -Inf
#> [55,] 1.0000000 -Inf -Inf
#> [56,] 0.5000000 0.5000000 -Inf
#> [57,] 0.5000000 0.5000000 -Inf
#> [58,] 1.0000000 -Inf -Inf
#> [59,] 1.0000000 -Inf -Inf
#> [60,] 0.5000000 0.5000000 -Inf
#> [61,] 1.0000000 -Inf -Inf
#> [62,] 1.0000000 -Inf -Inf
#> [63,] 1.0000000 -Inf -Inf
#> [64,] 1.0000000 -Inf -Inf
#> [65,] 1.0000000 -Inf -Inf
#> [66,] 1.0000000 -Inf -Inf
#> [67,] 1.0000000 -Inf -Inf
#> [68,] 1.0000000 -Inf -Inf
#> [69,] 0.5000000 0.5000000 -Inf
#> [70,] 0.5000000 0.5000000 -Inf
#> [71,] 1.0000000 -Inf -Inf
#> [72,] 1.0000000 -Inf -Inf
#> [73,] 1.0000000 -Inf -Inf
#> [74,] 0.5000000 0.5000000 -Inf
#> [75,] 0.5000000 0.5000000 -Inf
#> [76,] 0.5000000 0.5000000 -Inf
#> [77,] 0.5000000 0.5000000 -Inf
#> [78,] 0.5000000 0.5000000 -Inf
#> [79,] 0.5000000 0.5000000 -Inf
#> [80,] 1.0000000 -Inf -Inf
#> [81,] 0.5000000 0.5000000 -Inf
#> [82,] 0.5000000 0.5000000 -Inf
#> [83,] 0.5000000 0.5000000 -Inf

```



```
#> [84,] 0.5000000 0.5000000 -Inf
#> [85,] 0.3333333 0.3333333 0.3333333
#> [86,] 0.5000000 0.5000000 -Inf
#> [87,] 0.5000000 0.5000000 -Inf
#> [88,] 0.5000000 0.5000000 -Inf
#> [89,] 0.5000000 0.5000000 -Inf
#> [90,] 0.5000000 0.5000000 -Inf
#> [91,] 0.3333333 0.3333333 0.3333333
#> [92,] 0.3333333 0.3333333 0.3333333
#> [93,] 0.3333333 0.3333333 0.3333333
#> [94,] 0.3333333 0.3333333 0.3333333
#> [95,] 0.3333333 0.3333333 0.3333333
#> [96,] 0.3333333 0.3333333 0.3333333
#> [97,] 0.3333333 0.3333333 0.3333333
#> [98,] 0.3333333 0.3333333 0.3333333
#> [99,] 0.3333333 0.3333333 0.3333333
#> [100,] 0.3333333 0.3333333 0.3333333
```

- + The ``dataType`` list member equals the string ``"FROC"``, identifying it as an FROC 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).

Examination of the output reveals that:

- The `dataset` structure is a list with 8 members.
- This is a 5-treatment 4-reader dataset (the lengths of the first and second dimensions, respectively, of the NL and LL arrays). The names of the treatments are in the `modalityID` array:
- Location-level ratings of non-diseased regions are stored in the NL list member of the dataset.
- Location-level ratings of diseased regions are stored in the LL list member of the dataset.

4.2.2 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.

4.2.3 Details of the NL and LL list members

TBA * 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).

4.2.3.1 Numbers of non-diseased and diseased cases

- TBA

```
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.

4.2.4 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`.

4.2.4.1 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
```

4.2.4.2 The meaning of a negative infinity rating

- Obviously, a real rating cannot be negative infinity ⁴. This value is reserved for **missing ratings**, and more generally, **missing marks** ⁵. 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”!

4.2.5 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:

⁴If 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.

⁵Since there is a one-to-one correspondence between marks and ratings.

```
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
```

. 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 ROI-level-abnormal ROIs on this case are 4. The mean rating over all ROI-level-abnormal ROIs 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 file

An Excel file in JAFROC format containing simulated ROI data corresponding to `dataset04`, 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`.

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

4.3.1 The FROC Excel file organization

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:

4.3.1.1 The Truth worksheet organization

4.3.1.2 The FP/NL worksheet organization

4.3.1.3 The TP/LL worksheet organization

The top two screenshots show the 'Truth' worksheet. The first screenshot displays rows 1 to 27, and the second screenshot displays rows 28 to 50. Both show columns A through F. Column A is 'CaseID', column B is 'LesionID', column C is 'Weight', column D is 'Truth', and column E is 'ROI'. The data shows a mix of normal and abnormal cases with varying lesion counts and weights.

The bottom screenshot shows the 'FP' worksheet, displaying rows 224 to 250. It has the same column structure (A-F) as the 'Truth' worksheet. The data in this section shows a series of normal cases (CaseID 180-199) where all 'Truth' values are 0.

- The **Truth** worksheet - this indicates which cases are diseased and which are non-diseased and the number of ROI-level-abnormal region on each case.
 - There are 50 normal cases (labeled 1-50) under column **CaseID** and 40 abnormal cases (labeled 51-90).
 - The **LesionID** field for each normal case (e.g., **CaseID** = 1) is zero and there is one row per case. For abnormal 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).

The left screenshot shows the 'FP' worksheet with columns A (ReaderID), B (ModalityID), C (CaseID), and D (FP_Rating). The data rows show ratings for various combinations of ReaderID, ModalityID, and CaseID. The right screenshot shows the 'TP' worksheet with columns A (ReaderID), B (ModalityID), C (CaseID), and D (TP_Rating). The data rows show ratings for various combinations of ReaderID, ModalityID, and CaseID.

- The FP (or NL) worksheet - this lists the ratings of ROI-level-normal regions.
 - For **ReaderID** = 1, **ModalityID** = 1 and **CaseID** = 1 there are 4 rows, corresponding to the 4 ROI-level-normal regions in this case. The corresponding ratings are . 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 abnormal case has 4 abnormal regions).

The screenshot shows the 'TP' worksheet with columns A (ReaderID), B (ModalityID), C (CaseID), D (LesionID), and E (TP_Rating). The data rows show ratings for various combinations of ReaderID, ModalityID, CaseID, and LesionID.

- The TP (or LL) worksheet - this lists the ratings of ROI-level-abnormal regions.
 - Because normal 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 abnormal case.
 - There are two entries for this case, corresponding to the two ROI-level-abnormal 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-normal regions in this case.
 - One should be satisfied that for each abnormal case the sum of the number of entries in the TP and FP worksheets is always 4.

Chapter 5

ROI data format

5.1 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.
- 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.