

RJafroc Documentation

Dev P. Chakraborty, PhD

2020-03-10

Contents

Preface	7
1 Introduction	9
1.1 References	9
2 ROC data format	11
2.1 Introduction	11
2.2 Note to existing users	11
2.3 The Excel data format	12
2.4 Illustrative toy file	12
2.5 The Truth worksheet	12
2.6 The structure of an ROC dataset	13
2.7 The false positive (FP) ratings	15
2.8 The true positive (TP) ratings	16
2.9 Correspondence between NL member of dataset and the FP work- sheet	16
2.10 Correspondence between LL member of dataset and the TP work- sheet	17
2.11 Correspondence using the which function	17
2.12 References	18
3 FROC data format	19
3.1 Purpose	19
3.2 Introduction	19

3.3	The Excel data format	20
3.4	The Truth worksheet	20
3.5	The structure of an FROC dataset	21
3.6	The false positive (FP) ratings	22
3.7	The true positive (TP) ratings	23
3.8	On the distribution of numbers of lesions in abnormal cases . . .	24
3.9	Definition of lesWghtDistr array	27
3.10	Summary	29
3.11	References	30
4	ROC split plot data format	31
4.1	Introduction	31
4.2	The Excel data format	31
4.3	The Truth worksheet	31
4.4	The structure of the ROC split plot dataset	33
4.5	The truthTableStr member	34
4.6	The false positive (FP) ratings	36
4.7	The true positive (TP) ratings	37
4.8	Summary	38
4.9	References	39
5	FROC split plot data format	41
5.1	Introduction	41
5.2	The Excel data format	41
5.3	The Truth worksheet	41
5.4	The structure of the FROC split plot dataset	43
5.5	The false positive (FP) ratings	45
5.6	The true positive (TP) ratings	46
5.7	Summary	47
5.8	References	48

<i>CONTENTS</i>	5
-----------------	---

6 Simulate an FROC split plot dataset	49
6.1 This vignette is under construction!!	49
6.2 The starting point is an actual crossed FROC dataset	49
6.3 Understanding <code>truthTableStr</code> object <code>t1</code>	50
6.4 Modify a crossed FROC workbook to simulate a split-plot FROC design	51
6.5 Example of deletion of interpretations	53
6.6 Understanding <code>truthTableStr</code> object <code>t2</code>	54
6.7 References	54

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 describing 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

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.

CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm
1	0	0	0.1,3,5,4	0.1	ROC
2	0	0	0.1,3,5,4	0.1	ROC
3	0	0	0.1,3,5,4	0.1	crossed
70	1	1	0.1,3,5,4	0.1	
71	1	1	0.1,3,5,4	0.1	
72	1	1	0.1,3,5,4	0.1	
73	1	1	0.1,3,5,4	0.1	
74	1	1	0.1,3,5,4	0.1	

Figure 2.1: Truth worksheet for file rocCr.xlsx

```
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$NL` member, with dimension `[2, 5, 8, 1]`, contains the ratings of normal cases. The extra values in the third dimension, filled with `NA`s, are needed for compatibility with FROC datasets, as unlike ROC, false positives are possible on diseased cases.
- The `x$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.

ReaderID	ModalityID	CaseID	FP_Rating
0	0	1	0
0	0	2	0
0	0	3	0
0	1	1	0
0	1	2	0
0	1	3	0
1	0	1	0
1	0	2	0
1	0	3	0
1	1	1	0
1	1	2	0
1	1	3	0
2	0	1	0
2	0	2	0
2	0	3	0
2	1	1	0
2	1	2	0
2	1	3	0
3	0	1	0
3	0	2	0
3	0	3	0
3	1	1	0
3	1	2	0
3	1	3	0
4	0	1	0
4	0	2	0
4	0	3	0
4	1	1	0
4	1	2	0
4	1	3	0
0	1	1	1
0	1	2	1
0	1	3	1
1	0	1	1
1	0	2	1
1	0	3	1
1	1	1	1
1	1	2	1
1	1	3	1
2	0	1	1
2	0	2	1
2	0	3	1
2	1	1	1
2	1	2	1
2	1	3	1
3	0	1	1
3	0	2	1
3	0	3	1
3	1	1	1
3	1	2	1
3	1	3	1
4	0	1	1
4	0	2	1
4	0	3	1
4	1	1	1
4	1	2	1
4	1	3	1

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).

- ## 2.8 The true positive (TP) ratings

The screenshot shows the Microsoft Excel application window. The title bar includes icons for Windows, AutoSave, and various application functions. The ribbon is set to 'PivotTable Tools' with the 'Design' tab selected. The PivotTable is located in the 'Data' section of the ribbon. The PivotTable structure is as follows:

ReaderID	ModelID	Count
1	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	1
1	56	1
1	57	1
1	58	1
1	59	1
1	60	1
1	61	1
1	62	1
1	63	1
1	64	1
1	65	1
1	66	1
1	67	1
1	68	1
1	69	1
1	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

The status bar at the bottom of the window displays the following information:

- Average: 76.65333333
- Count: 265
- Sum: 3878

- 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.

- The list member `x$NL` is an array with `dim = c(2,5,8,1)`.

2.10. CORRESPONDENCE BETWEEN LL MEMBER OF DATASET AND THE TP WORKSHEET17

- 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 always 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$LL` is an array with `dim = c(2,5,5,1)`.
 - The first dimension (2) comes from the number of modalities.
 - The second dimension (5) comes from the number of readers.
 - The third dimension (5) comes from the number of diseased cases.
 - The fourth dimension is always 1 for an ROC dataset.
- The value of `x$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

FROC data format

3.1 Purpose

- Explain the data format of the input Excel file for FROC datasets.
- Explain the format of the FROC dataset.
- Explain the lesion distribution array returned by `UtilLesionDistr()`.
- Explain the lesion weights array returned by `UtilLesionWeightsDistr()`.
- Details on the FROC paradigm are in my book.

3.2 Introduction

- See my book Chakraborty (2017) for full details.
- In the Free-response Receiver Operating Characteristic (FROC) paradigm (Chakraborty, 1989) the observer searches each case for signs of **localized disease** and marks and rates localized regions that are sufficiently suspicious for disease presence.
- FROC data consists of **mark-rating pairs**, where each mark is a localized-region that was considered sufficiently suspicious for presence of a localized lesion and the rating is the corresponding confidence level.
- 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 observer assigns a rating to each region. The rating, as in the ROC paradigm, 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 a lesion at the indicated region.

3.3 The Excel data format

The Excel file has three worksheets. These are named **Truth**, **NL** or **FP** and **LL** or **TP**.

3.4 The Truth worksheet

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

- Since a diseased case may have more than one lesion, the first five columns contain **at least** 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, 1, 2, etc., with each 0 representing a non-diseased case, 1 representing the *first* lesion on a diseased case, 2 representing the second lesion on a diseased case, if present, and so on.
- The non-diseased cases are labeled 1, 2 and 3, while the diseased cases are labeled 70, 71, 72, 73 and 74.
- There are 3 non-diseased cases in the dataset (the number of 0's in the **LesionID** column).
- There are 5 diseased cases in the dataset (the number of 1's in the **LesionID** column of the **Truth** worksheet).
- There are 3 readers in the dataset (each cell in the **ReaderID** column contains 0, 1, 2).
- There are 2 modalities in the dataset (each cell in the **ModalityID** column contains 0, 1).
- **Weight**: floating point; 0, for each non-diseased case, or values for each diseased case that add up to unity.
- Diseased case 70 has two lesions, with **LesionIDs** 1 and 2, and weights 0.3 and 0.7. Diseased case 71 has one lesion, with **LesionID** = 1, and **Weight** = 1. Diseased case 72 has three lesions, with **LesionIDs** 1, 2 and 3 and weights 1/3 each. Diseased case 73 has two lesions, with **LesionIDs** 1, and 2 and weights 0.1 and 0.9. Diseased case 74 has one lesion, with **LesionID** = 1 and **Weight** = 1.
- **ReaderID**: a comma-separated listing of readers, each represented by a unique **integer**, that have interpreted the case. In the example shown below each cell has the value 0, 1, 2. **Each cell has to be text formatted. Otherwise Excel will not accept it.**
- **ModalityID**: a comma-separated listing of modalities (or treatments), each represented by a unique **integer**, that apply to each case. In the example each cell has the value 0, 1. **Each cell has to be text formatted.**

- **Paradigm:** In the example shown below, the contents are **FROC** and **crossed**. It informs the software that this is an FROC dataset and the design is “crossed”, as in **Vignette #1**.

	A	B	C	D	E	F	G	H
	CaseID	LesionID	Weights	ReaderID	ModalityID	Paradigm		
1	1	0	0	0.12	0.1	FROC		
2	2	0	0	0.12	0.1	crossed		
3	3	0	0	0.12	0.1			
4	3	0	0	0.12	0.1			
5	70	1	0.3	0.12	0.1			
6	70	2	0.7	0.12	0.1			
7	71	1	1	0.12	0.1			
8	72	1	0.333	0.12	0.1			
9	72	2	0.333	0.12	0.1			
10	72	3	0.333	0.12	0.1			
11	73	1	0.3	0.12	0.1			
12	73	2	0.9	0.12	0.1			
13	78	1	1	0.12	0.1			

Figure 3.1: Truth worksheet for file inst/extdata/toyFiles/FROC/frocCr.xlsx

3.5 The structure of an FROC dataset

The example shown above corresponds to Excel file inst/extdata/toyFiles/FROC/frocCr.xlsx in the project directory.

```
frocCr <- system.file("extdata", "toyFiles/FROC/frocCr.xlsx",
                      package = "RJaFROC", mustWork = TRUE)
x <- DfReadDataFile(frocCr, newExcelFileFormat = TRUE)
str(x)
#> List of 12
#> $ NL          : num [1:2, 1:3, 1:8, 1:2] 1.02 2.89 2.21 3.01 2.14 ...
#> $ LL          : num [1:2, 1:3, 1:5, 1:3] 5.28 5.2 5.14 4.77 4.66 4.87 3.01 3.27 3.31 3.19 ...
#> $ lesionVector : int [1:5] 2 1 3 2 1
#> $ lesionID     : num [1:5, 1:3] 1 1 1 1 1 ...
#> $ lesionWeight : num [1:5, 1:3] 0.3 1 0.333 0.1 1 ...
#> $ dataType     : chr "FROC"
#> $ modalityID   : Named chr [1:2] "0" "1"
#> ..- attr(*, "names")= chr [1:2] "0" "1"
#> $ readerID     : Named chr [1:3] "0" "1" "2"
#> ..- attr(*, "names")= chr [1:3] "0" "1" "2"
#> $ design       : chr "CROSSED"
#> $ normalCases  : int [1:3] 1 2 3
#> $ abnormalCases: int [1:5] 70 71 72 73 74
#> $ truthTableStr: num [1:2, 1:3, 1:8, 1:4] 1 1 1 1 1 1 1 1 1 1 ...
```

- This follows the general description in **Vignette #1**. The differences are described below.

- The `x$dataType` member indicates that this is an FROC dataset.
- The `x$lesionVector` member is a vector whose contents reflect the number of lesions in each diseased case, i.e., 2, 1, 3, 2, 1 in the current example.
- The `x$lesionID` member indicates the labeling of the lesions in each diseased case.

```
x$lesionID
#>      [,1] [,2] [,3]
#> [1,]    1    2 -Inf
#> [2,]    1 -Inf -Inf
#> [3,]    1    2    3
#> [4,]    1    2 -Inf
#> [5,]    1 -Inf -Inf
```

- This shows that the lesions on the first diseased case are labeled 1 and 2. The `-Inf` is a filler used to denote a missing value. The second diseased case has one lesion labeled 1. The third diseased case has three lesions labeled 1, 2 and 3, etc.
- The `lesionWeight` member is the clinical importance of each lesion. Lacking specific clinical reasons, the lesions should be equally weighted; this is *not* true for this toy dataset.

```
x$lesionWeight
#>      [,1]      [,2]      [,3]
#> [1,] 0.3000000 0.7000000 -Inf
#> [2,] 1.0000000 -Inf -Inf
#> [3,] 0.3333333 0.3333333 0.3333333
#> [4,] 0.1000000 0.9000000 -Inf
#> [5,] 1.0000000 -Inf -Inf
```

- The first diseased case has two lesions, the first has weight 0.3 and the second has weight 0.7. The second diseased case has one lesion with weight 1. The third diseased case has three equally weighted lesions, each with weight 1/3. Etc.

3.6 The false positive (FP) ratings

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

- It consists of 4 columns, of equal length. **The common length is unpredictable.** It could be zero if the dataset has no NL marks (a distinct possibility if the lesions are very easy to find and the modality and/or observer has high performance). All one knows is that the common length is an integer greater than or equal to zero.

ReaderID	ModalityID	CaseID	FP_Rating
0	0	1	1.02
0	0	1	2.17
0	0	2	2.22
0	0	3	1.9
1	0	1	2.23
1	0	2	3.1
1	0	2	2.23
1	0	3	2.07
2	0	1	2.14
2	0	2	1.86
2	0	3	1.95
0	1	1	2.89
0	1	2	2.89
0	1	74	0.84
0	1	72	1.85
0	1	5	3.22
1	1	1	0.01
1	1	2	1.96
1	1	3	2.08
2	1	71	2.24
2	1	71	4.01
2	1	72	1.86

Figure 3.2: Fig. 2: FP/NL worksheet for file inst/extdata/toyFiles/FROC/frocCr.xlsx

- In the example dataset, the common length is 22.
- **ReaderID**: the reader labels: these must be 0, 1, or 2, as declared in the **Truth** worksheet.
- **ModalityID**: the modality labels: must be 0 or 1, as declared in the **Truth** worksheet.
- **CaseID**: the labels of cases with NL marks. In the FROC paradigm, NL events can occur on non-diseased **and** diseased cases.
- **FP_Rating**: the floating point ratings of NL marks. Each row of this worksheet yields a rating corresponding to the values of **ReaderID**, **ModalityID** and **CaseID** for that row.
- For **ModalityID** 0, **ReaderID** 0 and **CaseID** 1 (the first non-diseased case declared in the **Truth** worksheet), there is a single NL mark that was rated 1.02, corresponding to row 2 of the FP worksheet.
- Diseased cases with NL marks are also declared in the FP worksheet. Some examples are seen at rows 15, 16 and 21-23 of the FP worksheet.
- Rows 21 and 22 show that **caseID** = 71 got two NL marks, rated 2.24, 4.01.
- That this is the *only* case with two marks determines the length of the fourth dimension of the **x\$NL** list member, 2 in the current example. Absent this case, the length would have been one.
- In general, the case with the most NL marks determines the length of the fourth dimension of the **x\$NL** list member.
- The reader should convince oneself that the ratings in **x\$NL** reflect the contents of the FP worksheet.

3.7 The true positive (TP) ratings

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

- This worksheet can only have diseased cases. The presence of a non-

Figure 3.3: Fig. 3: TP/LL worksheet for file inst/extdata/toyFiles/FROC/frocCr.xlsx

diseased case in this worksheet will generate an error.

- The common vertical length, 31 in this example, is a-priori unpredictable. Given the structure of the **Truth** worksheet for this dataset, the maximum length would be 9 times 2 times 3, assuming every lesion is marked for each modality, reader and diseased case. The 9 comes from the total number of non-zero entries in the **LesionID** column of the **Truth** worksheet.
- The fact that the length is smaller than the maximum length means that there are combinations of modality, reader and diseased cases on which some lesions were not marked.
- As an example, the first lesion in **CaseID** equal to 70 was marked (and rated 5.28) in **ModalityID** 0 and **ReaderID** 0.
- The length of the fourth dimension of the **x\$LL** list member, 3 in the present example, is determined by the diseased case with the most lesions in the **Truth** worksheet.
- The reader should convince oneself that the ratings in **x\$LL** reflect the contents of the TP worksheet.

3.8 On the distribution of numbers of lesions in abnormal cases

- Consider a much larger dataset, **dataset11**, with structure as shown below:

```
x <- dataset11
str(x)
#> List of 12
#> $ NL      : num [1:4, 1:5, 1:158, 1:4] -Inf -Inf -Inf -Inf -Inf ...
#> $ LL      : num [1:4, 1:5, 1:115, 1:20] -Inf -Inf -Inf -Inf -Inf ...
#> $ lesionVector : int [1:115] 6 4 7 1 3 3 3 8 11 2 ...
#> $ lesionID    : num [1:115, 1:20] 1 1 1 1 1 1 1 1 1 1 ...
```


3.8. ON THE DISTRIBUTION OF NUMBERS OF LESIONS IN ABNORMAL CASES 25

```
#> $ lesionWeight : num [1:115, 1:20] 0.167 0.25 0.143 1 0.333 ...
#> $ dataType      : chr "FROC"
#> $ modalityID    : Named chr [1:4] "1" "2" "3" "4"
#> ..- attr(*, "names")= chr [1:4] "1" "2" "3" "4"
#> $ readerID      : Named chr [1:5] "1" "2" "3" "4" ...
#> ..- attr(*, "names")= chr [1:5] "1" "2" "3" "4" ...
#> $ design        : chr "CROSSED"
#> $ normalCases   : int [1:43] 6 9 14 27 62 66 70 71 83 91 ...
#> $ abnormalCases: int [1:115] 1 2 3 5 7 8 10 11 13 17 ...
#> $ truthTableStr: num [1:4, 1:5, 1:158, 1:21] 1 1 1 1 1 1 1 1 1 1 ...
```

- Focus for now in the 115 abnormal cases.
- The numbers of lesions in these cases is contained in `x$lesionVector`.

```
x$lesionVector
#> [1] 6 4 7 1 3 3 3 8 11 2 4 6 2 16 5 2 8 3 4 7 11 1 4 3 4
#> [26] 4 7 3 2 5 2 2 7 6 6 4 10 20 12 6 4 7 12 5 1 1 5 1 2 8
#> [51] 3 1 2 2 3 2 8 16 10 1 2 2 6 3 2 2 4 6 10 11 1 2 6 2 4
#> [76] 5 2 9 6 6 8 3 8 7 1 1 6 3 2 1 9 8 8 2 2 12 1 1 1 1
#> [101] 1 3 1 2 2 1 1 1 1 3 1 1 1 2 1
```

- For example, the first abnormal case contains 6 lesions, the second contains 4 lesions, the third contains 7 lesions, etc. and the last abnormal case contains 1 lesion.
- To get an idea of the distribution of the numbers of lesions per abnormal cases, one could interrogate this vector as shown below using the `which()` function:

```
for (el in 1:max(x$lesionVector)) cat(
  "abnormal cases with", el, "lesions = ",
  length(which(x$lesionVector == el)), "\n")
#> abnormal cases with 1 lesions = 25
#> abnormal cases with 2 lesions = 23
#> abnormal cases with 3 lesions = 13
#> abnormal cases with 4 lesions = 10
#> abnormal cases with 5 lesions = 5
#> abnormal cases with 6 lesions = 11
#> abnormal cases with 7 lesions = 6
#> abnormal cases with 8 lesions = 8
#> abnormal cases with 9 lesions = 2
#> abnormal cases with 10 lesions = 3
#> abnormal cases with 11 lesions = 3
#> abnormal cases with 12 lesions = 3
#> abnormal cases with 13 lesions = 0
```

```
#> abnormal cases with 14 lesions = 0
#> abnormal cases with 15 lesions = 0
#> abnormal cases with 16 lesions = 2
#> abnormal cases with 17 lesions = 0
#> abnormal cases with 18 lesions = 0
#> abnormal cases with 19 lesions = 0
#> abnormal cases with 20 lesions = 1
```

- This tells us that 25 cases contain 1 lesion
- Likewise, 23 cases contain 2 lesions
- Etc.

3.8.1 Definition of lesDistr array

- Let us ask what is the fraction of (abnormal) cases with 1 lesion, 2 lesions etc.

```
for (el in 1:max(x$lesionVector)) cat("fraction of abnormal cases with", el, "lesions = ",
                                     length(which(x$lesionVector == el))/length(x$lesionVector), "\n")
#> fraction of abnormal cases with 1 lesions = 0.2173913
#> fraction of abnormal cases with 2 lesions = 0.2
#> fraction of abnormal cases with 3 lesions = 0.1130435
#> fraction of abnormal cases with 4 lesions = 0.08695652
#> fraction of abnormal cases with 5 lesions = 0.04347826
#> fraction of abnormal cases with 6 lesions = 0.09565217
#> fraction of abnormal cases with 7 lesions = 0.05217391
#> fraction of abnormal cases with 8 lesions = 0.06956522
#> fraction of abnormal cases with 9 lesions = 0.0173913
#> fraction of abnormal cases with 10 lesions = 0.02608696
#> fraction of abnormal cases with 11 lesions = 0.02608696
#> fraction of abnormal cases with 12 lesions = 0.02608696
#> fraction of abnormal cases with 13 lesions = 0
#> fraction of abnormal cases with 14 lesions = 0
#> fraction of abnormal cases with 15 lesions = 0
#> fraction of abnormal cases with 16 lesions = 0.0173913
#> fraction of abnormal cases with 17 lesions = 0
#> fraction of abnormal cases with 18 lesions = 0
#> fraction of abnormal cases with 19 lesions = 0
#> fraction of abnormal cases with 20 lesions = 0.008695652
```

- This tells us that fraction 0.217 of (abnormal) cases contain 1 lesion
- And fraction 0.2 of (abnormal) cases contain 2 lesions
- Etc.

- This information is contained the the `lesDistr` array
- It is coded in the Utility function `UtilLesionDistr()`

```
lesDistr <- UtilLesionDistr(x)
lesDistr
#>      [,1]      [,2]
#> [1,]    1 0.217391304
#> [2,]    2 0.200000000
#> [3,]    3 0.113043478
#> [4,]    4 0.086956522
#> [5,]    5 0.043478261
#> [6,]    6 0.095652174
#> [7,]    7 0.052173913
#> [8,]    8 0.069565217
#> [9,]    9 0.017391304
#> [10,]   10 0.026086957
#> [11,]   11 0.026086957
#> [12,]   12 0.026086957
#> [13,]   16 0.017391304
#> [14,]   20 0.008695652
```

- The `UtilLesionDistr()` function returns an array with two columns and number of rows equal to the number of distinct values of lesions per case.
- The first column contains the number of distinct values of lesions per case, 14 in the current example.
- The second column contains the fraction of diseased cases with the number of lesions indicated in the first column.
- The second column must sum to unity

```
sum(UtilLesionDistr(x)[,2])
#> [1] 1
```

- The lesion distribution array will come in handy when it comes to predicting the operating characteristics from using the Radiological Search Model (RSM), as detailed in Chapter 17 of my book.

3.9 Definition of `lesWghtDistr` array

- This is returned by `UtilLesionWeightsDistr()`.
- This contains the same number of rows as `lesDistr`.
- The number of columns is one plus the number of rows as `lesDistr`.
- The first column contains the number of distinct values of lesions per case, 14 in the current example.

- The second column contains the weights of cases with number of lesions per case corresponding to row 1.
- The third column contains the weights of cases with number of lesions per case corresponding to row 2.
- Etc.
- Missing values are filled with `-Inf`.

```
lesWghtDistr <- UtilLesionWeightsDistr(x)
cat("dim(lesDistr) =", dim(lesDistr), "\n")
#> dim(lesDistr) = 14 2
cat("dim(lesWghtDistr) =", dim(lesWghtDistr), "\n")
#> dim(lesWghtDistr) = 14 21
cat("lesWghtDistr = \n\n")
#> lesWghtDistr =
lesWghtDistr
#>      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
#> [1,]  1 1.00000000      -Inf      -Inf      -Inf      -Inf      -Inf
#> [2,]  2 0.50000000 0.50000000      -Inf      -Inf      -Inf      -Inf
#> [3,]  3 0.33333333 0.33333333 0.33333333      -Inf      -Inf      -Inf
#> [4,]  4 0.25000000 0.25000000 0.25000000 0.25000000      -Inf      -Inf
#> [5,]  5 0.20000000 0.20000000 0.20000000 0.20000000 0.20000000      -Inf
#> [6,]  6 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667
#> [7,]  7 0.14285714 0.14285714 0.14285714 0.14285714 0.14285714 0.14285714
#> [8,]  8 0.12500000 0.12500000 0.12500000 0.12500000 0.12500000 0.12500000
#> [9,]  9 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111 0.11111111
#> [10,] 10 0.10000000 0.10000000 0.10000000 0.10000000 0.10000000 0.10000000
#> [11,] 11 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909
#> [12,] 12 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333
#> [13,] 16 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000
#> [14,] 20 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000
#>      [,8]      [,9]      [,10]      [,11]      [,12]      [,13]      [,14]
#> [1,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [2,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [3,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [4,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [5,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [6,]      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [7,] 0.14285714      -Inf      -Inf      -Inf      -Inf      -Inf      -Inf
#> [8,] 0.12500000 0.12500000      -Inf      -Inf      -Inf      -Inf      -Inf
#> [9,] 0.11111111 0.11111111 0.11111111      -Inf      -Inf      -Inf      -Inf
#> [10,] 0.10000000 0.10000000 0.10000000 0.10000000      -Inf      -Inf      -Inf
#> [11,] 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909      -Inf      -Inf
#> [12,] 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333      -Inf
#> [13,] 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000 0.0625
#> [14,] 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.0500
```

```

#>      [,15] [,16] [,17] [,18] [,19] [,20] [,21]
#> [1,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [2,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [3,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [4,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [5,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [6,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [7,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [8,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [9,]  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [10,] -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [11,] -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [12,] -Inf  -Inf  -Inf  -Inf  -Inf  -Inf  -Inf
#> [13,] 0.0625 0.0625 0.0625 -Inf  -Inf  -Inf  -Inf
#> [14,] 0.0500 0.0500 0.0500 0.05 0.05 0.05 0.05

```

- Row 3 corresponds to 3 lesions per case and the weights are 1/3, 1/3 and 1/3.
- Row 13 corresponds to 16 lesions per case and the weights are 0.06250000, 0.06250000, ..., repeated 13 times.
- Note that the number of rows is less than the maximum number of lesions per case (20).
- This is because some configurations of lesions per case (e.g., cases with 13 lesions per case) do not occur in this dataset.

3.10 Summary

- The FROC dataset has far less regularity in structure as compared to an ROC dataset.
- The length of the first dimension of either `x$NL` or `x$LL` list members is the total number of modalities, 2 in the current example.
- The length of the second dimension of either `x$NL` or `x$LL` list members is the total number of readers, 3 in the current example.
- The length of the third dimension of `x$NL` is the total number of cases, 8 in the current example. The first three positions account for NL marks on non-diseased cases and the remaining 5 positions account for NL marks on diseased cases.
- The length of the third dimension of `x$LL` is the total number of diseased cases, 5 in the current example.
- The length of the fourth dimension of `x$NL` is determined by the case (diseased or non-diseased) with the most NL marks, 2 in the current example.
- The length of the fourth dimension of `x$LL` is determined by the diseased case with the most lesions, 3 in the current example.

3.11 References

Chapter 4

ROC split plot data format

4.1 Introduction

- The purpose of this vignette is to explain the data format of the input Excel file for an ROC *split-plot* dataset.
- In a split-plot dataset each reader interprets a *different* sub-set of cases in all modalities, i.e., the cases interpreted by different readers have no overlap.
- Each sub-set of cases can have different numbers of non-diseased and diseased cases.
- The example below assumes the same numbers of non-diseased and diseased cases.
- The data format has been extended to **NewFormat** to allow such datasets.

4.2 The Excel data format

As before, the Excel file has three worksheets named **Truth**, **NL** or **FP** and **LL** or **TP**. The Excel file corresponding to the example that follows is `inst/extdata/toyFiles/ROC/rocSp.xlsx`.

4.3 The Truth worksheet

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

- The first five columns contain as many rows as there are cases in the dataset.

- **CaseID**: unique **integers**, one per case, representing the cases in the dataset.
- **LesionID**: integers 0, representing non-diseased cases and 1 representing the diseased cases.
- The **ReaderID** column is a listing of readers each represented by a **unique string**. Note that, unlike the crossed design, the **ReaderID** column has *single values*. **Each cell has to be text formatted.**
- The non-diseased cases interpreted by reader with **ReaderID** value 1 are labeled 6, 7, 8, 9 and 10, each with **LesionID** value 0.
- The diseased cases interpreted by this reader are labeled 16, 17, 18, 19 and 20, each with **LesionID** value 1.
- The second reader, with **ReaderID** value 4, interprets five non-diseased cases labeled 21, 22, 23, 24 and 25, each with **LesionID** value 0, and five diseased cases labeled 36, 37, 38, 39 and 40, each with **LesionID** value 1.
- The third reader, with **ReaderID** value 5, interprets five non-diseased cases labeled 46, 47, 48, 49 and 50, each with **LesionID** value 0 and five diseased cases labeled 51, 52, 53, 54 and 55, each with **LesionID** value 1.
- **Weight**: floating point value 0 - this is not used for ROC data.
- **ModalityID**: a comma-separated listing of modalities, each represented by a **unique string**. In the example shown below each cell has the value 1, 2. **Each cell has to be text formatted.**
- **Paradigm**: In the example shown in this vignette, the contents are ROC and split-plot.

CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm
6	0	0	1	1,2	ROC
7	0	0	1	1,2	ROC
8	0	0	1	1,2	ROC
9	0	0	1	1,2	ROC
10	0	0	1	1,2	ROC
16	1	0	1	1,2	split-plot
17	1	0	1	1,2	split-plot
18	1	0	1	1,2	split-plot
19	1	0	1	1,2	split-plot
20	1	0	1	1,2	split-plot
21	0	0	4	1,2	ROC
22	0	0	4	1,2	ROC
23	0	0	4	1,2	ROC
24	0	0	4	1,2	ROC
25	0	0	4	1,2	ROC
36	1	0	4	1,2	split-plot
37	1	0	4	1,2	split-plot
38	1	0	4	1,2	split-plot
39	1	0	4	1,2	split-plot
40	1	0	4	1,2	split-plot
46	0	0	5	1,2	ROC
47	0	0	5	1,2	ROC
48	0	0	5	1,2	ROC
49	0	0	5	1,2	ROC
50	0	0	5	1,2	ROC
51	1	0	5	1,2	split-plot
52	1	0	5	1,2	split-plot
53	1	0	5	1,2	split-plot
54	1	0	5	1,2	split-plot
55	1	0	5	1,2	split-plot

Figure 4.1: Fig. 1: Truth worksheet for file inst/extdata/toyFiles/ROC/rocSp.xlsx

4.4 The structure of the ROC split plot dataset

- The example shown in Fig. 1 corresponds to Excel file `inst/extdata/toyFiles/ROC/rocSp.xlsx` in the project directory.

```
rocSp <- system.file("extdata", "toyFiles/ROC/rocSp.xlsx",
                     package = "RJafroc", mustWork = TRUE)
x <- DfReadDataFile(rocSp, newExcelFileFormat = TRUE)
str(x)
#> List of 12
#> $ NL          : num [1:2, 1:3, 1:30, 1] 1 1 -Inf -Inf -Inf ...
#> $ LL          : num [1:2, 1:3, 1:15, 1] 5 2.3 -Inf -Inf -Inf ...
#> $ lesionVector : int [1:15] 1 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionID     : num [1:15, 1] 1 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionWeight : num [1:15, 1] 1 1 1 1 1 1 1 1 1 1 1 ...
#> $ dataType     : chr "ROC"
#> $ modalityID   : Named chr [1:2] "1" "2"
#> ..- attr(*, "names")= chr [1:2] "1" "2"
#> $ readerID     : Named chr [1:3] "1" "4" "5"
#> ..- attr(*, "names")= chr [1:3] "1" "4" "5"
#> $ design       : chr "SPLIT-PLOT"
#> $ normalCases  : int [1:15] 6 7 8 9 10 21 22 23 24 25 ...
#> $ abnormalCases: int [1:15] 16 17 18 19 20 36 37 38 39 40 ...
#> $ truthTableStr: num [1:2, 1:3, 1:30, 1:2] 1 1 NA NA NA NA 1 1 NA NA ...
```

- `DfReadDataFile()` flag `newExcelFileFormat` **must** be set to `TRUE` for split plot data.
- The dataset object `x` is a `list` variable with 12 members.
- There are 15 diseased cases in the dataset (the number of 1's in the `LesionID` column of the Truth worksheet) and 15 non-diseased cases (the number of 0's in the `LesionID` column).
- `x$NL`, with dimension `[2, 3, 30, 1]`, contains the ratings of normal cases. The extra values in the third dimension, filled with `NA`s, are needed for compatibility with FROC datasets.
- `x$LL`, with dimension `[2, 3, 15, 1]`, contains the ratings of abnormal cases.
- The `x$lesionVector` member is a vector with 15 ones representing the 15 diseased cases in the dataset.
- The `x$lesionID` member is an array with 15 ones (this member is needed for compatibility with FROC datasets).
- The `x$lesionWeight` member is an array with 15 ones (this member is needed for compatibility with FROC datasets).
- The `dataType` member is `ROC` which specifies the data collection method ("ROC", "FROC", "LROC" or "ROI").
- The `x$modalityID` member is a vector with two elements "1" and "2", naming the two modalities.

- #### 4.5 The truthTableStr member

- ```
x$truthTableStr[,1,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] 1 1 1 1 1 NA NA NA NA NA NA NA NA NA
#> [2,] 1 1 1 1 1 NA NA NA NA NA NA NA NA NA
#> [,15]
#> [1,] NA
#> [2,] NA
```

- [illegible]

```
#> [1,] NA
#> [2,] NA
```

- The following shows that the second reader interprets the next group of five normal cases, indexed 6 through 10, in both modalities.

```
x$truthTableStr[,2,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] NA NA NA NA NA 1 1 1 1 1 NA NA NA NA
#> [2,] NA NA NA NA NA 1 1 1 1 1 NA NA NA NA
#> [,15]
#> [1,] NA
#> [2,] NA
```

- The following shows that the third reader interprets the next group of five normal cases, indexed 11 through 15, in both modalities.

```
x$truthTableStr[,3,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] NA NA NA NA NA NA NA NA NA NA 1 1 1 1
#> [2,] NA NA NA NA NA NA NA NA NA NA 1 1 1 1
#> [,15]
#> [1,] 1
#> [2,] 1
```

- The following shows that the first reader interprets the first group of five abnormal cases, indexed 16 through 20, in both modalities.

```
x$truthTableStr[,1,16:30,2]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] 1 1 1 1 1 NA NA NA NA NA NA NA NA NA
#> [2,] 1 1 1 1 1 NA NA NA NA NA NA NA NA NA
#> [,15]
#> [1,] NA
#> [2,] NA
```

- In the following all elements are **NA** because abnormal cases correspond to `lesionID = 2`.

```
x$truthTableStr[,1,16:30,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
#> [2,] NA NA NA NA NA NA NA NA NA NA NA NA NA NA
#> [,15]
#> [1,] NA
#> [2,] NA
```

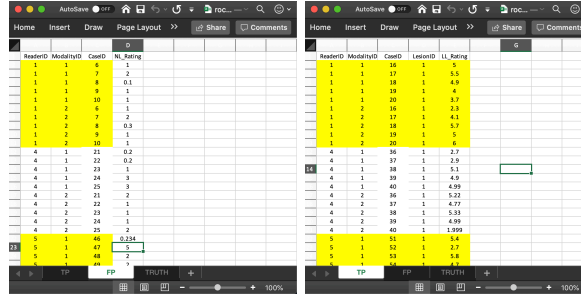


Figure 4.2: Fig. 2 FP/TP worksheets; LEFT=FP, (b) RIGHT=TP

## 4.6 The false positive (FP) ratings

- These are found in the FP or NL worksheet, see Fig. 2, left panel.
- This worksheet has the ratings of non-diseased cases.
- The common vertical length is 30 in this example (2 modalities times 3 readers times 5 non-diseased cases per reader).
- **ReaderID**: the reader labels: these must be from 1, 4 or 5, as declared in the **Truth** worksheet.
- **ModalityID**: the modality labels: 1 or 2, as declared in the **Truth** worksheet.
- **CaseID**: the labels of non-diseased cases. Each **CaseID** - **ReaderID** combination must be consistent with that declared in the **Truth** worksheet.
- **NL\_Rating**: the floating point ratings of non-diseased cases. Each row of this worksheet yields a rating corresponding to the values of **ReaderID**, **ModalityID** and **CaseID** for that row.

```
x$NL[,1,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] 1 2 0.1 1 1 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [2,] 1 2 0.3 1 1 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$NL[,2,1:15,1]
```

```

#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf 0.2 0.2 1 3 3 -Inf -Inf -Inf -Inf
#> [2,] -Inf -Inf -Inf -Inf -Inf 2.0 1.0 1 1 2 -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$NL[,3,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 0.234 5 2 2
#> [2,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 3.000 2 2 2
#> [,15]
#> [1,] 2.00
#> [2,] 0.33

```

- The first line of the above code shows the ratings, in both modalities, of the first five non-diseased cases with **CaseIDs** 6,7,8,9,10 (indexed 1, 2, 3, 4, 5 and appearing in the first five columns) interpreted by the first reader (**ReaderID** 1).
- The second line shows the ratings, in both modalities, of the next five non-diseased cases with **CaseIDs** 21,22,23,24,25 (indexed 6, 7, 8, 9, 10 and appearing in the next five columns) interpreted by the second reader (**ReaderID** 4).
- The third line shows the ratings, in both modalities, of the final five non-diseased cases with **CaseIDs** 46,47,48,49,50 (indexed 11, 12, 13, 14, 15 and appearing in the final five columns) interpreted by the third reader (**ReaderID** 5).
- Values such as `x$NL[, ,16:30,1]`, which are there for compatibility with FROC data, are all filled with `-Inf`.

## 4.7 The true positive (TP) ratings

- These are found in the TP or LL worksheet, see Fig. 2, right panel.
- This worksheet has the ratings of diseased cases.
- The common vertical length is 30 in this example (2 modalities times 3 readers times 5 diseased cases per reader).
- **ReaderID**: the reader labels: these must be from 1, 4 or 5, as declared in the **Truth** worksheet.
- **ModalityID**: the modality labels: 1 or 2, as declared in the **Truth** worksheet.
- **CaseID**: the labels of diseased cases. Each **CaseID** - **ReaderID** combination must be consistent with that declared in the **Truth** worksheet.

- **LL\_Rating**: the floating point ratings of diseased cases. Each row of this worksheet yields a rating corresponding to the values of **ReaderID**, **ModalityID** and **CaseID** for that row.

```
x$LL[,1,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] 5.0 5.5 4.9 4 3.7 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [2,] 2.3 4.1 5.7 5 6.0 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$LL[,2,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf 2.70 2.90 5.10 4.90 4.990 -Inf -Inf -Inf -Inf
#> [2,] -Inf -Inf -Inf -Inf -Inf 5.22 4.77 5.33 4.99 1.999 -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$LL[,3,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 5.4 2.7 5.8 4.7
#> [2,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 5.4 2.7 5.8 4.7
#> [,15]
#> [1,] 5
#> [2,] 5
```

- The first line of code shows the ratings, in both modalities, of the first five diseased cases with **CaseIDs** 16,17,18,19,20 (indexed 1, 2, 3, 4, 5 and appearing in the first five columns) interpreted by the first reader (**ReaderID** 1).
- The second line shows the ratings, in both modalities, of the next five diseased cases with **CaseIDs** 36,37,38,39,40 (indexed 6, 7, 8, 9, 10 and appearing in the next five columns) interpreted by the second reader (**ReaderID** 4).
- The third line shows the ratings, in both modalities, of the final five non-diseased cases with **CaseIDs** 51,52,53,54,55 (indexed 11, 12, 13, 14, 15 and appearing in the final five columns) interpreted by the third reader (**ReaderID** 5).

## 4.8 Summary

- The FROC dataset has far less regularity in structure as compared to an ROC dataset.

- The length of the first dimension of either  $\mathbf{x}_{NL}$  or  $\mathbf{x}_{LL}$  list members is the total number of modalities, 2 in the current example.
- The length of the second dimension of either  $\mathbf{x}_{NL}$  or  $\mathbf{x}_{LL}$  list members is the total number of readers, 3 in the current example.
- The length of the third dimension of  $\mathbf{x}_{NL}$  is the total number of cases, 8 in the current example. The first three positions account for NL marks on non-diseased cases and the remaining 5 positions account for NL marks on diseased cases.
- The length of the third dimension of  $\mathbf{x}_{LL}$  is the total number of diseased cases, 5 in the current example.
- The length of the fourth dimension of  $\mathbf{x}_{NL}$  is determined by the case (diseased or non-diseased) with the most NL marks, 2 in the current example.
- The length of the fourth dimension of  $\mathbf{x}_{LL}$  is determined by the diseased case with the most lesions, 3 in the current example.

## 4.9 References





## Chapter 5

# FROC split plot data format

### 5.1 Introduction

- The purpose of this vignette is to explain the data format of the input Excel file for an FROC *split-plot* dataset.
- In a split-plot dataset each reader interprets a sub-set of cases in all modalities.
- The cases interpreted by different readers have no overlap.
- It is assumed, for now, that each sub-set of cases has the same numbers of non-diseased and diseased cases.

### 5.2 The Excel data format

The Excel file has three worksheets named **Truth**, **NL or FP** and **LL or TP**.

### 5.3 The Truth worksheet

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

- The first five columns contain as many rows as there are non-diseased cases (9) plus total number of lesions (27) in the dataset (each row with a non-zero **LesionID** corresponds to a lesion).

- **CaseID**: unique **integers**, one per case, representing the cases in the dataset.
- **LesionID**: integers 0, 1, 2, etc., with each 0 representing a non-diseased case, 1 representing the *first* lesion on a diseased case, 2 representing the second lesion on a diseased case, if present, and so on.
- The three non-diseased cases interpreted by reader with **ReaderID** value 0 are labeled 1, 2, 3, while the diseased cases interpreted by this reader are labeled 70, 71, 72, 73 and 74, with **LesionID** values ranging from 1 to 3.
- The second reader, with **ReaderID** value 1, interprets three non-diseased cases labeled 4, 5 and 6, each with **LesionID** value 0, and five diseased cases labeled 80, 81, 82, 83 and 84, with **LesionID** values ranging from 1 to 3.
- The third reader, with **ReaderID** value 2, interprets three non-diseased cases labeled 7, 8 and 9, each with **LesionID** value 0 and five diseased cases labeled 90, 91, 92, 93 and 94, with **LesionID** values ranging from 1 to 3.
- **Weight**: floating point value adding upto unity for diseased cases as required for FROC data.
- **ModalityID**: a comma-separated listing of modalities, each represented by a unique **integer**. In the example shown below each cell has the value 0, 1. **Each cell has to be text formatted.**
- **Paradigm**: In the example shown below, the contents are FROC and split-plot.

| CaseID | LesionID | Weight | ReaderID | ModalityID | Paradigm   |
|--------|----------|--------|----------|------------|------------|
| 1      | 0        | 0      | 0        | 0,1        | froc       |
| 2      | 0        | 0      | 0        | 0,1        | froc       |
| 3      | 0        | 0      | 0        | 0,1        | froc       |
| 4      | 0        | 0      | 1        | 0,1        | split plot |
| 5      | 0        | 0      | 1        | 0,1        | split plot |
| 6      | 0        | 0      | 1        | 0,1        | split plot |
| 7      | 0        | 0      | 2        | 0,1        | split plot |
| 8      | 0        | 0      | 2        | 0,1        | split plot |
| 9      | 0        | 0      | 2        | 0,1        | split plot |
| 70     | 1        | 0.333  | 0        | 0,1        | froc       |
| 71     | 1        | 0.333  | 0        | 0,1        | froc       |
| 72     | 1        | 0.333  | 0        | 0,1        | froc       |
| 73     | 1        | 0.333  | 0        | 0,1        | froc       |
| 74     | 1        | 0.333  | 0        | 0,1        | froc       |
| 80     | 1        | 0.333  | 1        | 0,1        | split plot |
| 81     | 1        | 0.333  | 1        | 0,1        | split plot |
| 82     | 1        | 0.333  | 1        | 0,1        | split plot |
| 83     | 1        | 0.333  | 1        | 0,1        | split plot |
| 84     | 1        | 0.333  | 1        | 0,1        | split plot |
| 90     | 1        | 0.333  | 2        | 0,1        | split plot |
| 91     | 1        | 0.333  | 2        | 0,1        | split plot |
| 92     | 1        | 0.333  | 2        | 0,1        | split plot |
| 93     | 1        | 0.333  | 2        | 0,1        | split plot |
| 94     | 1        | 0.333  | 2        | 0,1        | split plot |

Figure 5.1: Two views of Truth worksheet for file frocSp.xlsx

## 5.4 The structure of the FROC split plot dataset

The example shown in Fig. 1 corresponds to Excel file `inst/extdata/toyFiles/FROC/frocSp.xlsx` in the project directory.

```
frocSp <- system.file("extdata", "toyFiles/FROC/frocSp.xlsx",
 package = "RJafroc", mustWork = TRUE)
x <- DfReadDataFile(frocSp, newExcelFileFormat = TRUE)
str(x)
#> List of 12
#> $ NL : num [1:2, 1:3, 1:24, 1:3] 1.02 2.89 -Inf -Inf -Inf ...
#> $ LL : num [1:2, 1:3, 1:15, 1:3] 5.28 5.2 -Inf -Inf -Inf ...
#> $ lesionVector : int [1:15] 2 1 3 2 1 2 1 3 2 1 ...
#> $ lesionID : num [1:15, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
#> $ lesionWeight : num [1:15, 1:3] 0.3 1 0.333 0.1 1 ...
#> $ dataType : chr "FROC"
#> $ modalityID : Named chr [1:2] "0" "1"
#> ..- attr(*, "names")= chr [1:2] "0" "1"
#> $ readerID : Named chr [1:3] "0" "1" "2"
#> ..- attr(*, "names")= chr [1:3] "0" "1" "2"
#> $ design : chr "SPLIT-PLOT"
#> $ normalCases : int [1:9] 1 2 3 4 5 6 7 8 9
#> $ abnormalCases: int [1:15] 70 71 72 73 74 80 81 82 83 84 ...
#> $ truthTableStr: num [1:2, 1:3, 1:24, 1:4] 1 1 NA NA NA NA 1 1 NA NA ...
```

- Flag `newExcelFileFormat` **must** be set to `TRUE` for split plot data.
- The dataset object `x` is a `list` variable with 12 members.
- Note that the `dataType` member is `FROC` and the `design` member is `SPLIT-PLOT`.
- There are 15 diseased cases in the dataset (the number of 1's in the `LesionID` column of the `Truth` worksheet) and 9 non-diseased cases (the number of 0's in the `LesionID` column).
- The `x$lesionVector` member is a vector with 15 ones representing the 15 diseased cases in the dataset.
- The `x$lesionID` member is a 15 x 3 array labeling the lesions in the dataset.
- The `x$lesionWeight` member is a 15 x 3 array.

```
x$lesionVector
#> [1] 2 1 3 2 1 2 1 3 2 1 2 1 3 2 1
x$lesionID
#> [,1] [,2] [,3]
#> [1,] 1 2 -Inf
#> [2,] 1 -Inf -Inf
#> [3,] 1 2 3
```

```

#> [4,] 1 2 -Inf
#> [5,] 1 -Inf -Inf
#> [6,] 1 2 -Inf
#> [7,] 1 -Inf -Inf
#> [8,] 1 2 3
#> [9,] 1 2 -Inf
#> [10,] 1 -Inf -Inf
#> [11,] 1 2 -Inf
#> [12,] 1 -Inf -Inf
#> [13,] 1 2 3
#> [14,] 1 2 -Inf
#> [15,] 1 -Inf -Inf
x$lesionWeight
#> [,1] [,2] [,3]
#> [1,] 0.3000000 0.7000000 -Inf
#> [2,] 1.0000000 -Inf -Inf
#> [3,] 0.3333333 0.3333333 0.3333333
#> [4,] 0.1000000 0.9000000 -Inf
#> [5,] 1.0000000 -Inf -Inf
#> [6,] 0.3000000 0.7000000 -Inf
#> [7,] 1.0000000 -Inf -Inf
#> [8,] 0.3333333 0.3333333 0.3333333
#> [9,] 0.1000000 0.9000000 -Inf
#> [10,] 1.0000000 -Inf -Inf
#> [11,] 0.3000000 0.7000000 -Inf
#> [12,] 1.0000000 -Inf -Inf
#> [13,] 0.3333333 0.3333333 0.3333333
#> [14,] 0.1000000 0.9000000 -Inf
#> [15,] 1.0000000 -Inf -Inf

```

- The `x$truthTableStr` member is a 2 x 3 x 24 x 4 array, i.e., I x J x K x (maximum number of lesions per case plus 1). The `plus 1` is needed to accommodate normal cases with `lesionID = 0`.
- Each entry in this array is either 1, meaning the corresponding interpretation exists, or NA, meaning the corresponding interpretation does not exist.
- For example, `x$truthTableStr[1,1,1,1]` is 1. This means that an interpretation exists for the first treatment (`modalityID = 0`), first reader (`readerID = 0`) and first (normal) case `caseID = 1` and `lesionID = 0`. This example corresponds to row 2 in the TRUTH worksheet.
- `x$truthTableStr[1,1,4,1]` is NA, which means an interpretation does not exist for the first treatment, first reader and fourth (normal) case.
- However, `x$truthTableStr[1,2,4,1]` is 1, which means an interpretation does exist for the first treatment, second reader and fourth (normal) case. This example corresponds to row 5 in the TRUTH worksheet.

- Likewise, `x$truthTableStr[1,1,10,3]` is 1, which means an interpretation does exist for the first treatment, first reader, tenth (abnormal) case and `lesionID = 2`. This example corresponds to row 12 in the TRUTH worksheet.
- As an aside, in the FROC paradigm an interpretation need not yield a mark-rating pair. An interpretation means the reader was “exposed to” and had the opportunity to mark the corresponding treatment-reader-case-lesion combination.
- The reader should confirm that the contents of `x$truthTableStr` summarizes the structure of the data in the TRUTH worksheet.

## 5.5 The false positive (FP) ratings

These are found in the FP or NL worksheet, see Fig. 2.

The figure displays two side-by-side Excel spreadsheets. The left spreadsheet, titled 'NL/FP', has columns: ReaderID, ModalityID, CaseID, and FP\_Rating. It contains 30 rows of data. The right spreadsheet, titled 'LL/TP', has columns: ReaderID, ModalityID, CaseID, LesionID, and TP\_Rating. It also contains 30 rows of data. Both spreadsheets show a summary bar at the bottom with statistics like Average, Count, and Sum.

Figure 5.2: NL/FP worksheet, left, and LL/TP worksheet, right, for file `frocSp.xlsx`

- This worksheet has the ratings of non-diseased cases.
- The common vertical length is 30 in this example (2 modalities times 3 readers times 5 non-diseased cases per reader).
- **ReaderID**: the reader labels: these must be from 0, 1 or 2, as declared in the Truth worksheet.
- **ModalityID**: the modality labels: 0 or 1, as declared in the Truth worksheet.
- **CaseID**: the labels of non-diseased cases. Each **CaseID**, **ModalityID**, **ReaderID** combination must be consistent with that declared in the Truth worksheet.
- **FP\_Rating**: the floating point ratings of non-diseased cases. Each row of this worksheet yields a rating corresponding to the values of **ReaderID**, **ModalityID** and **CaseID** for that row. Each **CaseID**, **ModalityID**, **ReaderID** combination must be consistent with that declared in the Truth worksheet.

```

x$NL[,1,1:9,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
#> [1,] 1.02 2.22 1.90 -Inf -Inf -Inf -Inf -Inf -Inf
#> [2,] 2.89 0.84 1.85 -Inf -Inf -Inf -Inf -Inf -Inf
x$NL[,2,1:9,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
#> [1,] -Inf -Inf -Inf 2.21 3.10 2.21 -Inf -Inf -Inf
#> [2,] -Inf -Inf -Inf 3.22 3.01 1.96 -Inf -Inf -Inf
x$NL[,3,1:9,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
#> [1,] -Inf -Inf -Inf -Inf -Inf -Inf 2.14 1.98 1.95
#> [2,] -Inf -Inf -Inf -Inf -Inf -Inf 2.24 4.01 1.65

```

- The first line of the above code shows the ratings, in both modalities, of the first three non-diseased cases with **CaseIDs** 1,3,3 (indexed 1, 2, 3 and appearing in the first three columns) interpreted by the first reader (**ReaderID** 0).
- The second line shows the ratings, in both modalities, of the next three non-diseased cases with **CaseIDs** 4,5,6 (indexed 4, 5, 6 and appearing in the next three columns) interpreted by the second reader (**ReaderID** 1).
- The third line shows the ratings, in both modalities, of the final three non-diseased cases with **CaseIDs** 7,8,9 (indexed 7, 8, 9 and appearing in the final three columns) interpreted by the third reader (**ReaderID** 2).
- Values such as `x$NL[, ,16:30,1]`, which are there for compatibility with FROC data, are all filled with `-Inf`.

## 5.6 The true positive (TP) ratings

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

- This worksheet has the ratings of diseased cases.
- The common vertical length is 30 in this example (2 modalities times 3 readers times 5 diseased cases per reader).
- **ReaderID**: the reader labels: these must be from 0, 1 or 2, as declared in the **Truth** worksheet.
- **ModalityID**: the modality labels: 0 or 1, as declared in the **Truth** worksheet.
- **CaseID**: the labels of diseased cases. Each **CaseID**, **ModalityID**, **ReaderID** combination must be consistent with that declared in the **Truth** worksheet.
- **TP\_Rating**: the floating point ratings of diseased cases. Each row of this worksheet yields a rating corresponding to the values of **ReaderID**, **ModalityID** and **CaseID** for that row. Each **CaseID**, **ModalityID**,

ReaderID combination must be consistent with that declared in the Truth worksheet.

```
x$LL[,1,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] 5.28 3.01 5.98 5.00 4.26 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [2,] 5.20 3.27 4.61 5.18 4.72 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$LL[,2,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf -Inf 5.14 3.31 4.92 4.95 5.30 -Inf -Inf -Inf -Inf
#> [2,] -Inf -Inf -Inf -Inf -Inf -Inf 4.77 3.19 5.20 5.39 5.01 -Inf -Inf -Inf -Inf
#> [,15]
#> [1,] -Inf
#> [2,] -Inf
x$LL[,3,1:15,1]
#> [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
#> [1,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 4.66 4.03 5.22 4.94
#> [2,] -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf 4.87 1.94 -Inf -Inf
#> [,15]
#> [1,] 5.27
#> [2,] 4.78
```

- The first line of code shows the ratings, in both modalities, of the first five diseased cases with **CaseIDs** 70,71,72,73,74 (indexed 1, 2, 3, 4, 5 and appearing in the first five columns) interpreted by the first reader (**ReaderID** 0).
- The second line shows the ratings, in both modalities, of the next five diseased cases with **CaseIDs** 80,81,82,83,84 (indexed 6, 7, 8, 9, 10 and appearing in the next five columns) interpreted by the second reader (**ReaderID** 1).
- The third line shows the ratings, in both modalities, of the final five non-diseased cases with **CaseIDs** 90,91,92,93,94 (indexed 11, 12, 13, 14, 15 and appearing in the final five columns) interpreted by the third reader (**ReaderID** 2).

## 5.7 Summary

- TBA

## 5.8 References



## Chapter 6

# Simulate an FROC split plot dataset

### 6.1 This vignette is under construction!!

- This is a follow-up on the recently added (v1.3.1) capability to read a split-plot dataset.
- Lacking an actual split-plot dataset to test the routines, I decided to simulate one.
- The simulated dataset is of dataType FROC and the number of cases interpreted by each reader is reader-dependent.
- This makes it *really* exercise the validity of the `DfReadDataFile` function.
- In my experience, the `dataset$truthTableStr` member is invaluable in catching data entry errors so much of this vignette focuses on it.

### 6.2 The starting point is an actual crossed FROC dataset

The example shown below begins with the Excel file `inst/extdata/FrocData.xlsx` in the project directory (this corresponds to the 5-modality FED dataset `dataseet04` (Zanca et al., 2009) with modalities 1, 2 and 3 removed). 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 `DfReadDataFile()` and saves it to object `x1`. The next statement extracts the `truthTableStr` list member, saves it to `t1` and shows its structure.

```

fed <- system.file("extdata", "FrocData.xlsx",
 package = "RJafroc", mustWork = TRUE)
x1 <- DfReadDataFile(fed, newExcelFileFormat = TRUE)
t1 <- x1$truthTableStr
str(t1)
#> num [1:2, 1:4, 1:200, 1:4] 1 1 1 1 1 1 1 1 1 1 ...

```

- There are 100 normal and 100 abnormal cases in this two-modality four-reader crossed dataset.
- Note that `t1` is the original crossed dataset `truthTableStr`.
- Recall from earlier vignette that for the fourth subscript of `t1` the value 1 applies to cases with no lesions (normals), value 2 applies to cases with one lesion, value 3 applies to cases with two lesions and 4 applies to cases with three lesions.
- The value for any allowed interpretation is 1 and otherwise it is NA.

### 6.3 Understanding `truthTableStr` object `t1`

- The following line yields 200 ( $=2 \times 100$ ) as reader 1 (second subscript) provides interpretations in both modalities (first subscript is blank meaning both modalities) for all 100 normal cases (third subscript is 1:100 and fourth subscript is 1) and therefore each of these interpretations yields a TRUE (i.e., 1).

```

sum(!is.na(t1[,1,1:100,1]))
#> [1] 200

```

- The following line yields 0 as the third subscript is 1:100, implying normal cases, but the fourth subscript is 2:4, implying abnormal cases and therefore each of these interpretations yields an NA and `!is.na(NA)` equals FALSE (i.e., zero).

```

sum(!is.na(t1[,1,1:100,2:4]))
#> [1] 0

```

- The following line also yields 800 ( $=2 \times 4 \times 100$ ) as readers 1:4 provide interpretations in both modalities for all normal cases and each interpretation yields a 1.

```

sum(!is.na(t1[, ,1:100,1]))
#> [1] 800

```

- The following line yields 200 ( $=2 \times 100$ ) because the fourth subscript (2) applies to abnormal cases with at least one lesion, and each abnormal case is guaranteed to have at least one lesion (i.e., a 1 entry in the **LesionID** column of the Excel Truth worksheet) and each of these interpretations yields a 1.

```
sum(!is.na(t1[,1,101:200,2]))
#> [1] 200
```

- The following line yields 62 ( $=2 \times 31$ ) because the fourth subscript (3) applies to abnormal cases with at least two lesions, and inspection of the **LesionID** column in the original Excel file reveals that 31 abnormal cases have two lesions.

```
sum(!is.na(t1[,1,101:200,3]))
#> [1] 62
```

- The following line yields 22 ( $=2 \times 11$ ) because the fourth subscript (4) applies to abnormal cases with three lesions. Inspection of the **LesionID** column reveals that 11 abnormal cases have three lesions.

```
sum(!is.na(t1[,1,101:200,4]))
#> [1] 22
```

- The following line yields 242 ( $=200+31+11$ ), the number of rows in the Truth worksheet.

```
sum(!is.na(t1[1,1,,]))
#> [1] 242
```

## 6.4 Modify a crossed FROC workbook to simulate a split-plot FROC design

- We create a simulated split-plot FROC dataset starting from a crossed FROC dataset.
- The basic idea is to modify interpretations that do not belong to a specified split-plot design.
- This was done (one could say the “hard way”) by manually making appropriate changes to `inst/extdata/FrocData.xlsx` and saving the results to `inst/extdata/toyFiles/FROC/FrocDataSpVaryK1K2.xlsx`. The file-name is intended to emphasize that the numbers of normal and abnormal cases can be reader-dependent, as long as they individually add up to 100.

- We divided the 100 normal and 100 abnormal cases into 4 groups of normal and abnormal cases, where each group is interpreted by one reader only.
- The first groups of cases, interpreted by reader 1 (label “1”), consists of 23 normal (case labels “100:122”) and 24 abnormal (case labels “0:23”) cases.
- The second groups of cases, interpreted by reader 2 (label “3”), consists of 27 normal (case labels “123:149”) and 26 abnormal (case labels “24:49”) cases.
- The third groups of cases, interpreted by reader 3 (label “4”), consists of 22 normal (case labels “150:171”) and 23 abnormal (case labels “51:73”) cases.
- The fourth groups of cases, interpreted by reader 4 (label “5”), consists of 28 normal (case labels “172:199”) and 27 abnormal (case labels c(“50,74:99”)) cases.

Figure 6.1: Truth worksheets; (a) LEFT=FrocData.xlsx, original crossed dataset; (b) RIGHT=FrocDataSpVaryK1K2.xlsx, modified split-plot dataset

- The above figure shows that the **ReaderID** column for cases 0:23 has been replaced with a 1, meaning that only reader 1 interprets these cases in both modalities. This yields 24 abnormal cases for reader 1 in each modality. Normal cases for this reader are 100:122.
- Not shown above: **all interpretations by reader 1 occurring for cases outside of 0:23 and 100:122 in the other two worksheets (TP and FP) are deleted.**
- The **ReaderID** column for cases 24:49 are replaced by 3, corresponding to the second reader. All interpretations by this reader for cases outside of 24:49 in the other two worksheets are deleted. Normal cases for this reader are 123:149 and observations outside of this range in the TP and FP worksheets are deleted.
- The **ReaderID** column for cases 51:73 are replaced by 4, corresponding to the third reader. All interpretations by this reader for cases outside of 51:73 in the other two worksheets are deleted. Normal cases for this reader are 150:171 and observations outside of this range in the TP and FP worksheets are deleted.

- The `ReaderID` column for cases 50 and 74:79 are replaced by 5, corresponding to the fourth reader. All interpretations by this reader for cases outside of the specified range in the other two worksheets are deleted. Normal cases for this reader are 172:199 and observations outside of this range in the TP and FP worksheets are deleted.
- The modified file is read by the code chunk below. The `read` function explicitly tests that observations outside of the specified ranges in the Truth sheet are not present in the other two worksheets.

## 6.5 Example of deletion of interpretations

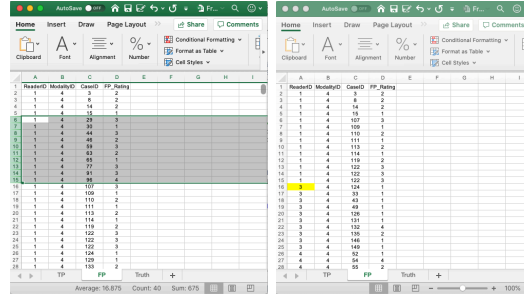


Figure 6.2: FP worksheets; (a) LEFT=FrocDataFP.xlsx, original crossed dataset; (b) RIGHT=FrocDataSpVaryK1K2FP.xlsx, modified split-plot dataset

- The two figures above illustrate deletion of interpretations.
- The left panel shows the FP worksheet for the original crossed data.
- For reader 1 and modality 4 it contains cases 29, 30, 44, ..., 91, 96 that do not belong to the split-plot dataset for this reader.
- Specifically, they are outside of 0:23 and 100:122, the allowed ranges for this reader.
- These are deleted, see right panel of above figure. The next allowed cases for this reader are 107, 109, ..., 122.
- The procedure is repeated for all readers and both TP and FP sheets.

```
fedsp <- system.file("extdata", "toyFiles/FROC/FrocDataSpVaryK1K2.xlsx",
 package = "RJaFROC", mustWork = TRUE)
x2 <- DfReadDataFile(fedsp, newExcelFileFormat = TRUE)
t2 <- x2$truthTableStr
```

## 6.6 Understanding `truthTableStr` object `t2`

- The following line below yields 46 ( $= 2 \times 23$ ) as reader 1 (second subscript) provides interpretations in both modalities (first subscript is blank) for all normal cases (third subscript is 1:100 and fourth subscript is 1) and there are 23 normal cases interpreted by reader 1.

```
sum(!is.na(t2[,1,1:100,1]))
#> [1] 46
```

- The following line confirms the first line, with a 1 contribution coming from each case in range 1:23.

```
sum(!is.na(t2[,1,1:23,1]))
#> [1] 46
```

- The following line yields 48 ( $= 2 \times 24$ ) because the fourth subscript (2) applies to abnormal cases with at least one lesion, and we know that this reader has interpreted 24 abnormal cases.

```
sum(!is.na(t2[,1,101:124,2]))
#> [1] 48
```

- The following line yields 54 ( $= 2 \times 27$ ) because the fourth subscript (1) applies to normal cases and we know that reader 2 has interpreted 27 normal cases.

```
sum(!is.na(t2[,2,,1]))
#> [1] 54
```

- The following line yields 52 ( $= 2 \times 26$ ) because the fourth subscript (2:4) applies to abnormal cases with at least one lesion, and we know that reader 2 has interpreted 26 abnormal cases.

```
sum(!is.na(t2[,2,,2:4]))
#> [1] 52
```

## 6.7 References

# Bibliography

- Chakraborty, D. P. (1989). Maximum likelihood analysis of free-response receiver operating characteristic (froc) data. *Medical Physics*, 16(4):561–568.
- 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.
- 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.