The RJafroc Quick Start Book

Dev P. Chakraborty, PhD

2023-12-25

Contents

Chapter 1

Preface

TBA

1.1 Rationale and Organization

- See here for an overview of my AI/FROC research websites.
- All references in this book to `RJafroc` refer to the `R` package with that name (case sensitive) (Chakraborty and Zhai, 2022).

1.2 Getting help on the software

- If you have installed `RJafroc` from `GitHub`:
  - Type ?`RJafroc-package` (RStudio will auto complete …) followed by `Enter`.
  - Scroll down and click on `Index`
- Regardless of where you installed from you can use the `RJafroc` website to access help.
  - Look under the `Reference` tab.
  - For example, for help on the function `PlotEmpiricalOperatingCharacteristics` look here

1.3 Acknowledgements

TBA

1.3.1 Persons who have stimulated my thinking:

Harold Kundel, MD

Claudia Mello-Thoms, PhD

Dr. Xuetong Zhai (contributed significantly to the significance testing sections and other chapters of my book).

1.4 Contributors to the software

Dr. Xuetong Zhai (he developed the first version of `RJafroc`)

Dr. Peter Phillips

Online Latex Editor at this website. I found this very useful in learning and using Latex to write math equations.

1.5 Dataset contributors

TBA

1.6 Accessing files and code

You would not normally need to access the files used to create the book. But if you are adventurous, …

To access files/code one needs to `fork` the `GitHub` repository. This will create, on your computer, a copy of all files used to create this document. To compile the files try `Build Book` and select `gitbook`. You will probably get errors corresponding to missing packages that are not loaded on your machine. All required packages are listed in the DESCRIPTION file. Install those packages and try again …

Quick Start

Chapter 2

ROC data format

2.1 How much finished 50%

(remove duplication)

2.2 Introduction

The JAFROC Excel data format was adopted circa. 2006. The purpose of this chapter is to explain the format of this file and how to read this file into a dataset object suitable for analysis using the `RJafroc` package.

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 for presence of disease. Location information associated with the disease, if applicable, is not collected.

2.3 Note to existing users

- The Excel file format has recently undergone changes involving three additional columns in the `Truth` worksheet. These are needed for generalization to other data collection paradigms and for better data entry error control.

- `RJafroc` will work with original format Excel files provided the `NewExcelFileFormat` flag in `DfReadDataFile` is set to `FALSE`, which is the default (see help page below).

- Going forward, one should use the new format, described below, and use `NewExcelFileFormat = TRUE` to read the file.

```
knitr::include_graphics("images/roc-data-format/DfReadDataFile.png")
```

DfReadDataFile {RJafroc}

Read a data file

Description

Read a disk file and create a ROC, FROC or LROC dataset object from it.

Usage

```
DfReadDataFile(
  fileName,
  format = "JAFROC",
  newExcelFileFormat = FALSE,
  lrocForcedMark = NA,
  delimiter = ",",
  sequentialNames = FALSE
)
```

Arguments

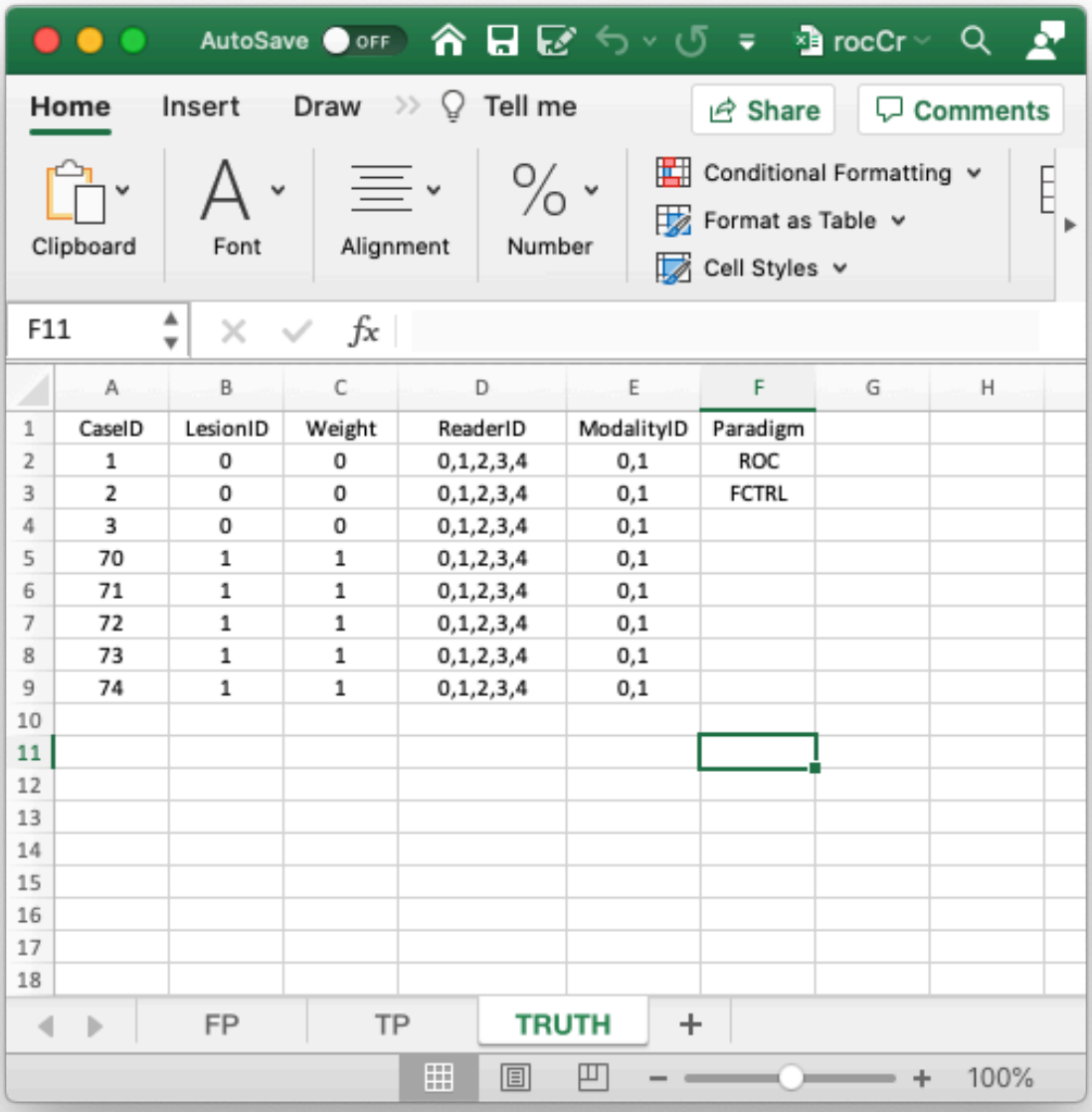
`fileName`	A string specifying the name of the file. The file-extension must match the fo
`format`	A string specifying the format of the data file. It can be `"JAFROC"`, the defau Excel file (**not `.xls`**), `"MRMC"` or `"iMRMC"`. For `"MRMC"` the format is deter extension (`.csv` or `.txt` or `.lrc`) as specified in https://perception.lab.uio extension is `.imrmc` and the format is described in https://code.google.com **following note for important information about deprecation of the `"MR`
`newExcelFileFormat`	Logical. Must be true to read LROC data. This argument only applies to the default is `FALSE`. If `TRUE` the function accommodates 3 additional columns `FALSE`, the original function (as in version 1.2.0) is used and the three extra an error.
`lrocForcedMark`	Logical: For LROC dataset only: is a forced mark required on every image? not required, set it to `FALSE` otherwise to `TRUE`.
`delimiter`	The string delimiter to be used for the `"MRMC"` format ("," is the default), see https://perception.lab.uiowa.edu/. This parameter is not used when reading files.
`sequentialNames`	A logical variable: if `TRUE`, consecutive integers (starting from 1) will be use IDs (i.e., names). Otherwise, treatment and reader IDs in the original data fi

2.4 Three worksheets in the Excel data file

- The illustrations in this chapter are for Excel file `R/quick-start/rocCr.xlsx` in the project directory. I assume the reader has forked the `RJafrocQuickStart` repository. See Section @ref(#quick-start-index-how-to-access-files) for how to get this file, and all other files and code in this `bookdown` book, on your computer.

- This is a *toy file*, i.e., a small made-up dataset used to illustrate essential features of the data format.

- The Excel file has three worksheets: `Truth`, `NL` (or `FP`) and `LL` (or `TP`). The worksheet names are case insensitive.



	A	B	C	D	E	F	G	H
1	CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm		
2	1	0	0	0,1,2,3,4	0,1	ROC		
3	2	0	0	0,1,2,3,4	0,1	FCTRL		
4	3	0	0	0,1,2,3,4	0,1			
5	70	1	1	0,1,2,3,4	0,1			
6	71	1	1	0,1,2,3,4	0,1			
7	72	1	1	0,1,2,3,4	0,1			
8	73	1	1	0,1,2,3,4	0,1			
9	74	1	1	0,1,2,3,4	0,1			
10								
11								
12								
13								
14								
15								
16								
17								
18								

2.4.1 The Truth worksheet

The Truth worksheet shown above contains 6 columns: CaseID, LesionID, Weight, ReaderID, ModalityID and Paradigm. These names are case sensitive.

1. CaseID: **unique integers**, one per case, representing the cases in the dataset. In the current 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 integers.

2. LesionID: integers 0 or 1, with each 0 representing a non-diseased case and each 1 representing a diseased case.

3. Weight: this field is not used for ROC data.

4. ReaderID: a **comma-separated** string containing the reader (i.e., radiologist or observer) labels, each represented by a **unique integer**, that have interpreted the case. In the example shown below each cell has the value 0, 1, 2, 3, 4 meaning each of these readers has interpreted all cases. With multiple readers each cell in this column has to be text formatted as otherwise Excel will not accept it. Select the worksheet, then Format - Cells - Number - Text - OK.

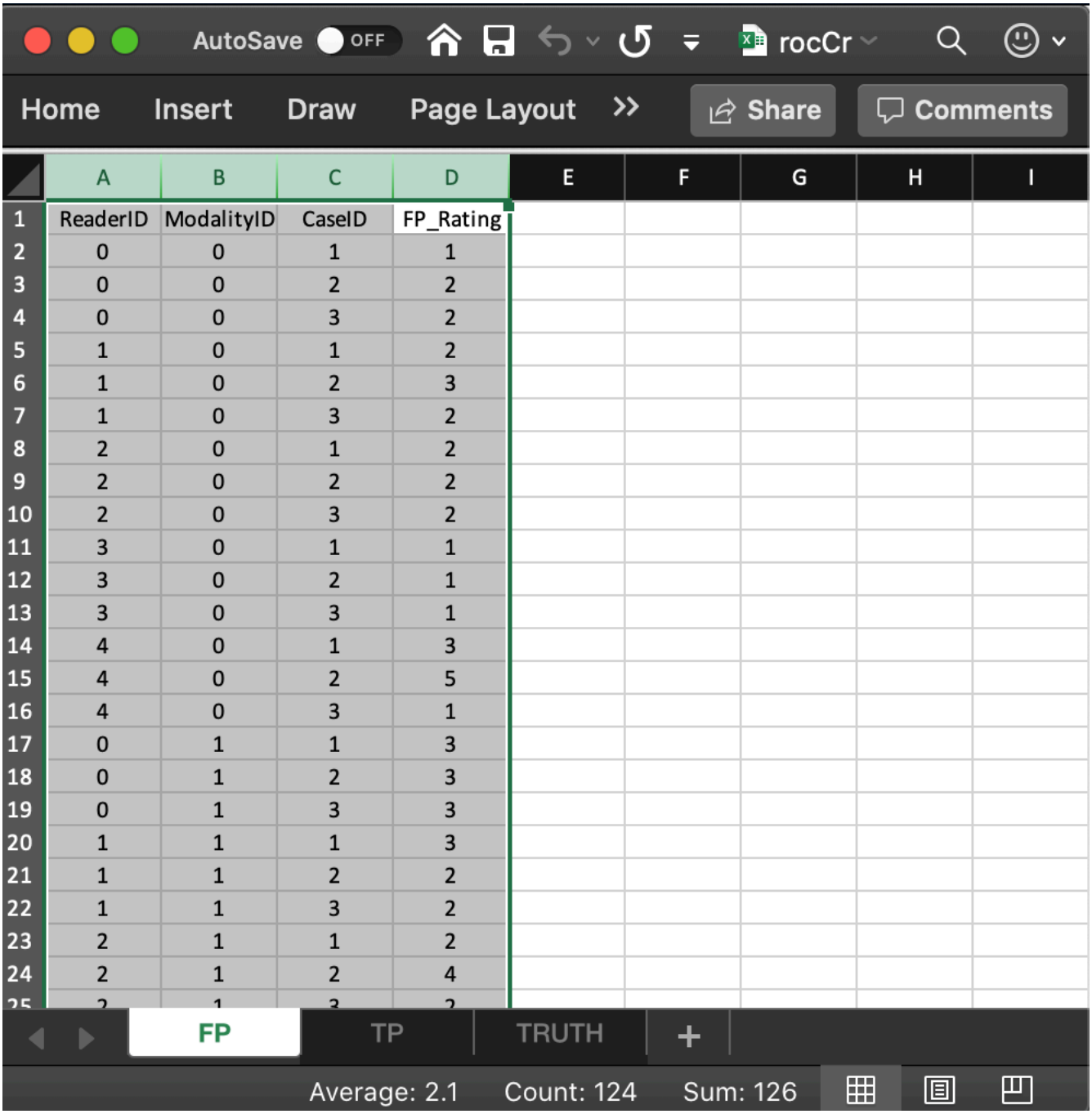
5. ModalityID: a comma-separated string containing the modality labels, each represented by a **unique integer**. In the example each cell has the value 0, 1 meaning this is a two-modality study. As above, with multiple modalities each cell has to be text formatted as otherwise Excel will not accept it.

6. Paradigm: this column contains two cells, ROC and FCTRL. It means that this is an ROC dataset and the study design is factorial (or fully-crossed), i.e., each reader interprets each case in each modality.

2.4.2 Comments on the Truth worksheet

There are 5 diseased cases in the dataset (the number of 1's in the LesionID column of the Truth worksheet). There are 3 non-diseased cases in the dataset (the number of 0's in the LesionID column). There are 5 readers in the dataset (each cell in the ReaderID column contains the string 0, 1, 2, 3, 4). There are 2 modalities in the dataset (each cell in the ModalityID column contains the string 0, 1).

2.4.3 The FP/NL worksheet



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

It consists of 4 columns, each of length 30 (i.e., # of modalities x number of readers x number of non-diseased cases). The (case sensitive) column names and meanings are as follows:

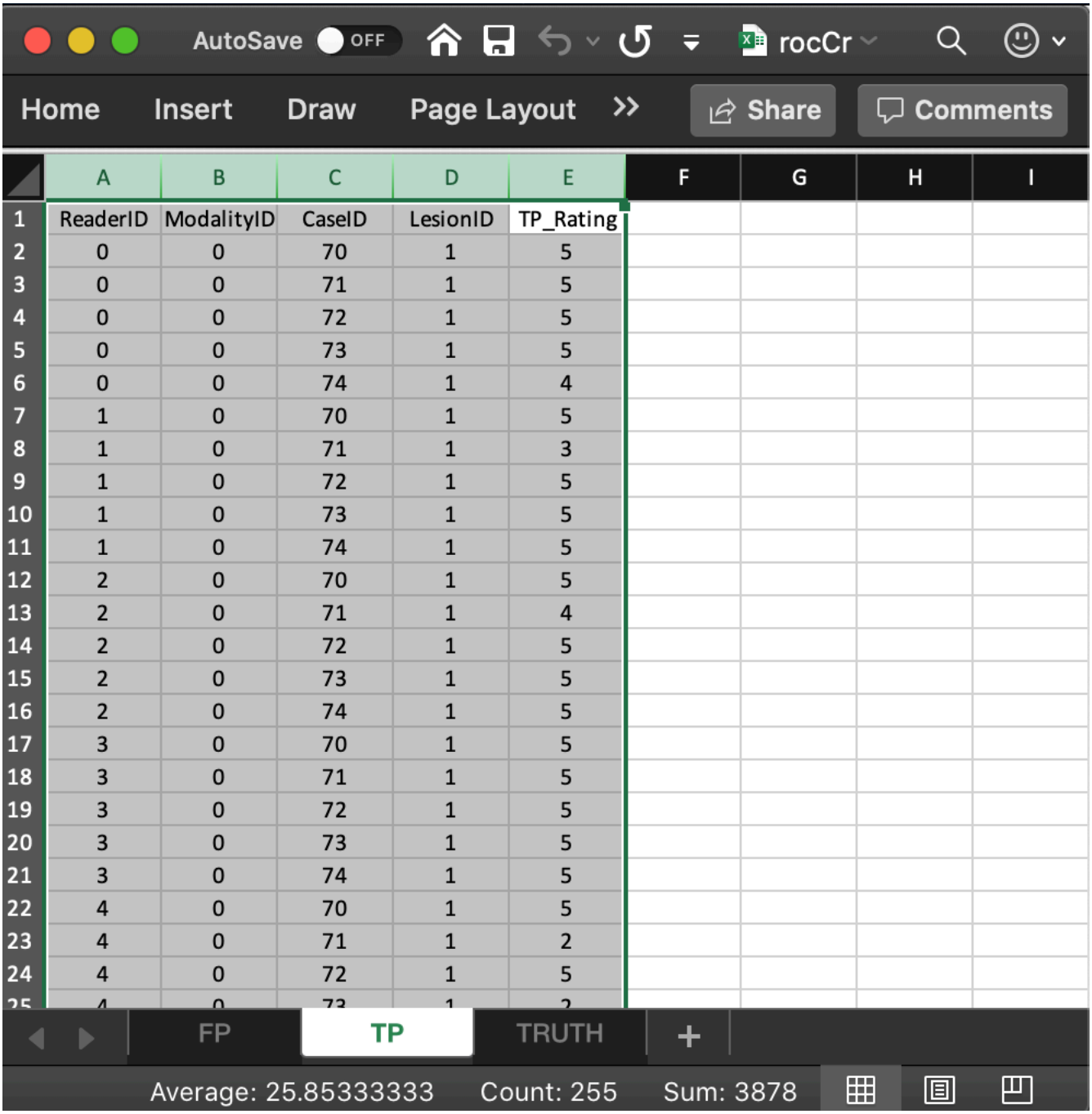
1. `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 6 times (i.e., # of modalities x number of non-diseased cases).

2. `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 15 times (i.e., # of readers x number of non-diseased cases).

3. `CaseID`: the case labels for non-diseased cases: `1`, `2` and `3`. Each label occurs 10 times (i.e., # of modalities x # of readers). The label of a diseased case cannot occur in the FP worksheet. If it does the software generates an error.

4. `FP_Rating`: the (floating point) ratings of non-diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID` and `CaseID` for that row.

2.4.4 The TP/LL worksheet



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

It consists of 5 columns, each of length 50 (i.e., # of modalities x number of readers x number of diseased cases). The (case sensitive) column names and meanings are as follows:

1. `ReaderID`: the reader labels: `0`, `1`, `2`, `3` and `4`. Each reader label occurs 10 times (i.e., # of modalities x number of diseased cases).

2. `ModalityID`: the modality or treatment labels: `0` and `1`. Each label occurs 25 times (i.e., # of readers x number of diseased cases).

3. `LesionID`: For an ROC dataset this column contains fifty 1's (each diseased case has one lesion).
4. `CaseID`: the case labels for non-diseased cases: `70`, `71`, `72`, `73` and `74`. Each label occurs 10 times (i.e., # of modalities x # of readers). For an ROC dataset the label of a non-diseased case cannot occur in the TP worksheet. If it does the software generates an error.
5. `TP_Rating`: the (floating point) ratings of diseased cases. Each row of this worksheet contains a rating corresponding to the values of `ReaderID`, `ModalityID`, `LesionID` and `CaseID` for that row.

2.5 Reading the Excel file

The following code uses the function `DfReadDataFile` to read the Excel file and save it to object `x`.

```
x <- DfReadDataFile("R/quick-start/rocCr.xlsx", newExcelFileFormat = TRUE)
```

- `newExcelFileFormat` is set to `TRUE` as otherwise columns D - F in the `Truth` worksheet are ignored and the dataset is assumed to be factorial, with `dataType` "automatically" determined from the contents of the FP and TP worksheets. [1]
- Flag `newExcelFileFormat = FALSE`, the default, is for compatibility with the original JAFROC format Excel format, which did not have columns D - F in the `Truth` worksheet. Its usage is deprecated.

2.6 Structure of dataset object

Most users will not need to be concerned with the internal structure of the dataset object `x`. For those interested in it, for my reference, and for ease of future maintenance of the software, this is deferred to Section 11.2.1.

[1] The assumptions underlying the "automatic" determination could be defeated by data entry errors.

Chapter 3

FROC data format

3.1 How much finished 90%

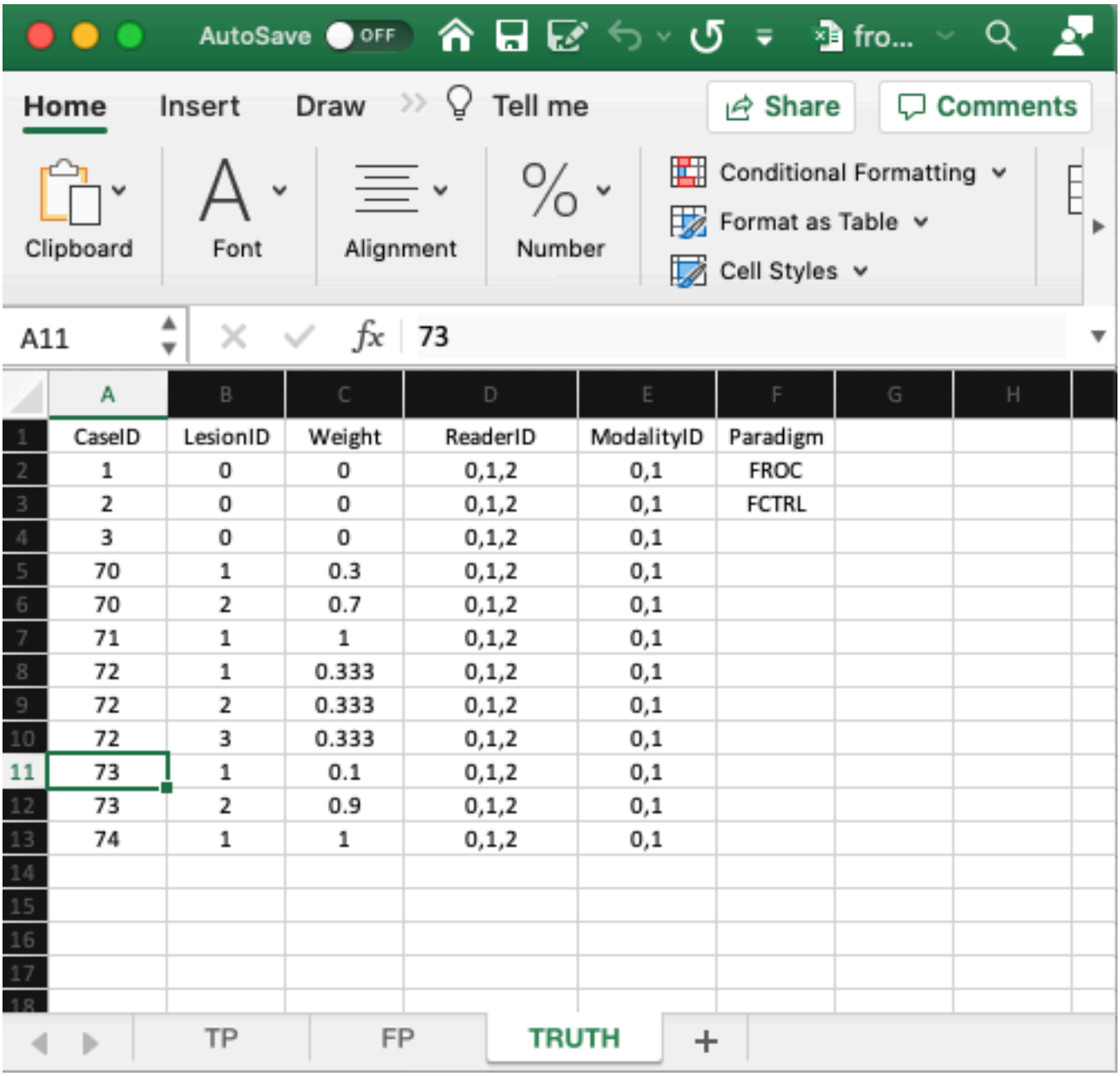
3.2 Introduction

The purpose of this chapter is to explain the format of the FROC Excel file and how to read this file into a dataset object suitable for analysis using the `RJafroc` package.

In the FROC paradigm the observer assigns a rating and a location to suspicious regions in images that exceed the reporting threshold. As an example a CAD algorithm may find tens of suspicious regions in each image but the algorithm designer only shows those regions (typically one or two) whose confidence levels exceed the chosen threshold.

The chapter is illustrated with a toy data file, `R/quick-start/frocCr.xlsx` in which readers '0', '1' and '2' interpret 8 cases in two modalities, '0' and '1'. The design is 'factorial', abbreviated to `FCTRL` in the software; this is also termed a 'fully-crossed' design. The Excel file has three worksheets named `Truth`, `NL` (or `FP`) and `LL` (or `TP`). These names are case-insensitive.

3.3 The `Truth` worksheet



	A	B	C	D	E	F	G	H
1	CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm		
2	1	0	0	0,1,2	0,1	FROC		
3	2	0	0	0,1,2	0,1	FCTRL		
4	3	0	0	0,1,2	0,1			
5	70	1	0.3	0,1,2	0,1			
6	70	2	0.7	0,1,2	0,1			
7	71	1	1	0,1,2	0,1			
8	72	1	0.333	0,1,2	0,1			
9	72	2	0.333	0,1,2	0,1			
10	72	3	0.333	0,1,2	0,1			
11	73	1	0.1	0,1,2	0,1			
12	73	2	0.9	0,1,2	0,1			
13	74	1	1	0,1,2	0,1			
14								
15								
16								
17								
18								

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 in the dataset. There are 8 cases ('1','2','3','70','71','72','73' and '74') in the dataset and 12 rows in the `Truth` worksheet, because some of the diseased cases contain more than one lesion.

1. `CaseID`: unique **integers** representing the individual cases in the dataset: e.g., '1', '2', '3', the 3 non-diseased cases and '70', '71', '72', '73', '74', the 5 diseased cases. The ordering of the numbers is inconsequential. [1]

2. `LesionID`: non-negative integers 0, 1, 2, ..., where:

   - Each 0 represents a non-diseased case, e.g., this field is zero for non-diseased cases '1', '2' and '3'.
   - Each 1 represents the *first* lesion in a diseased case, 2 represents the *second* lesion, if present, and so on.

[1] `CaseID` should not be so large that it cannot be represented in Excel by an integer; to be safe use unsigned short 8-bit integers.

3. `Weight` or clinical importance associated with lesion:

    - It is 0 for each non-diseased case,
    - For each diseased case the values must sum to unity.
    - A shortcut to assigning equal weights to all lesions in a case is to fill the `Weight` column with zeroes.

4. `ReaderID`: see Section 2.4.1.

5. `ModalityID`: see Section 2.4.1.

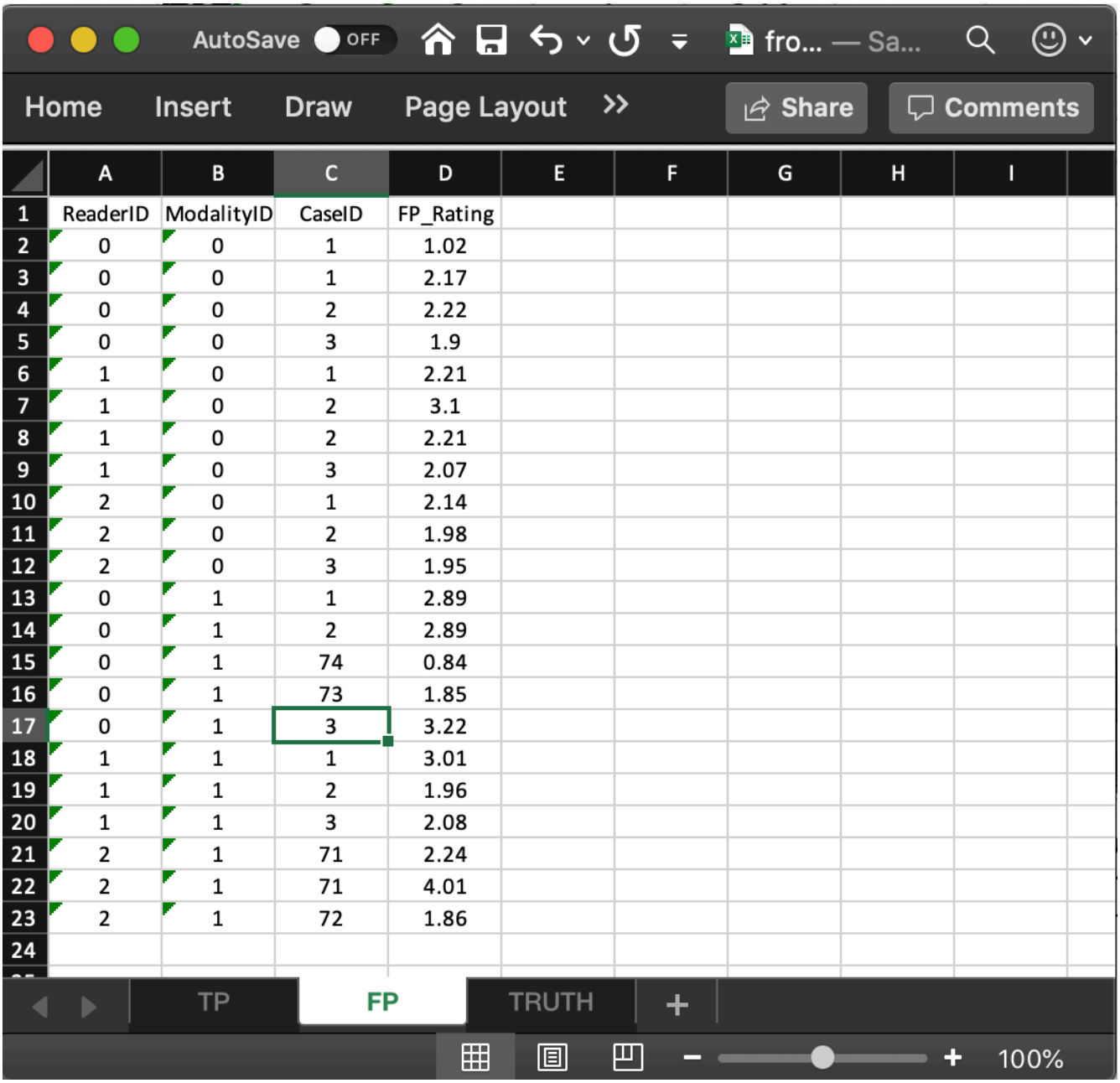
6. `Paradigm`: see Section 2.4.1.

3.3.1 Comments on the `Truth` worksheet

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). There are 3 readers in the dataset labeled '0, 1, 2'. There are 2 modalities in the dataset labeled '0, 1'. Diseased case `70` has two lesions, with `LesionID`s '1' and '2' and weights 0.3 and 0.7, respectively. Diseased case `71` has one lesion with `LesionID` = 1 and `Weight` = 1. Diseased case `72` has three lesions with `LesionID`s 1, 2 and 3 and weights 1/3 each. Diseased case `73` has two lesions, with `LesionID`s 1, and 2 and weights 0.1 and 0.9, respectively. Diseased case `74` has one lesion, with `LesionID` = 1 and `Weight` = 1. Note that `LesionID`s *identify* the lesions - for example, a lesion with high morbidity may be labeled `LesionID` = 1 and assigned weight 0.9 while a second lower morbidity lesion on the same case may be assigned `LesionID` = 2 and weight 0.1. In this example reversing the lesion IDs would lead to incorrect weight assignments.

3.4 The FP ratings

These are found in the `FP` or `NL` worksheet.



	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	FP_Rating					
2	0	0	1	1.02					
3	0	0	1	2.17					
4	0	0	2	2.22					
5	0	0	3	1.9					
6	1	0	1	2.21					
7	1	0	2	3.1					
8	1	0	2	2.21					
9	1	0	3	2.07					
10	2	0	1	2.14					
11	2	0	2	1.98					
12	2	0	3	1.95					
13	0	1	1	2.89					
14	0	1	2	2.89					
15	0	1	74	0.84					
16	0	1	73	1.85					
17	0	1	3	3.22					
18	1	1	1	3.01					
19	1	1	2	1.96					
20	1	1	3	2.08					
21	2	1	71	2.24					
22	2	1	71	4.01					
23	2	1	72	1.86					
24									

It consists of 4 columns of equal length. The common length is an integer random variable ≥ 0. It could be zero if the dataset has no NL marks (a possibility if the lesions are easy to find or the observer has perfect performance). In this example the common length is 22, which is a-priori unpredictable: for example, if the dataset has many FPs it could be large.

1. `ReaderID`: the reader labels: these must be one of `0`, `1`, or `2` as declared in the `Truth` worksheet.

2. `ModalityID`: the modality labels: must be one of `0` or `1` as declared in the `Truth` worksheet.

3. `CaseID`: the labels of cases with `NL` marks. These must be one of `1`, `2`, `3`, `70`, `71`, `72`, `73`, `74` as declared in the `Truth` worksheet. In the FROC paradigm `NL` events can occur on non-diseased **and** diseased cases.

4. `FP_Rating`: the floating point ratings of `NL` marks. Each cell contains the rating corresponding to the values of `ReaderID`, `ModalityID` and `CaseID` for that row.

3.4.1 Comments on the FP worksheet

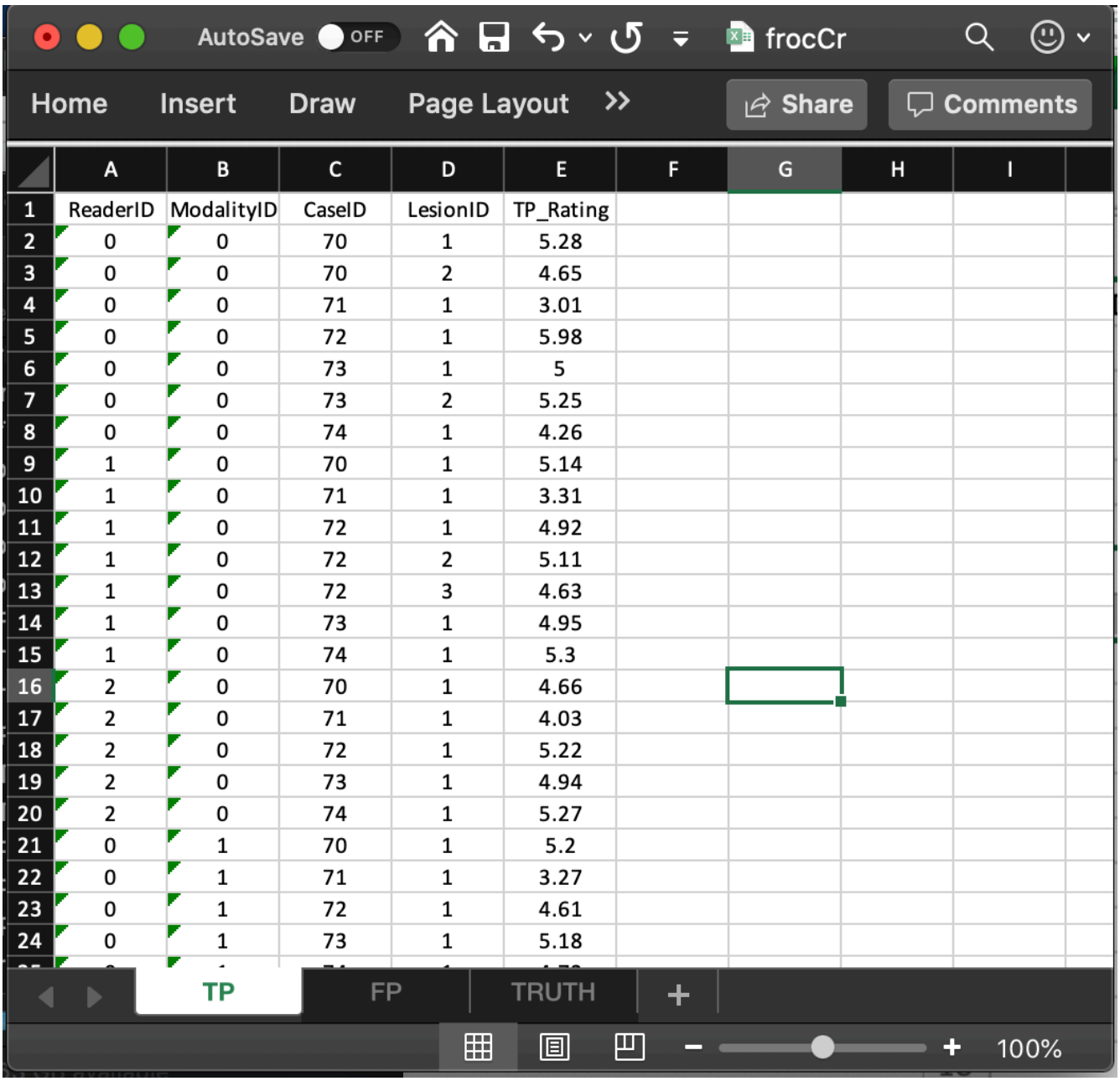
- For ModalityID 0, ReaderID 0 and CaseID 1 (the first non-diseased case declared in the Truth worksheet), there is a single NL mark that was rated 1.02, corresponding to row 2 of the FP worksheet.

- Diseased cases with NL marks are also recorded in the FP worksheet. Some examples are seen at rows 15, 16 and 21, 22, 23. Rows 21 and 22 show that caseID = 71 got two NL marks, rated 2.24, 4.01.

- Since this is the *only* case with two NL marks, it determines the length of the fourth dimension of the ds$ratings$NL, which is 2 in this example. Absent this case, the length would have been one. The case with the most NL marks determines the length of the fourth dimension of ds$ratings$NL. The reader should confirm that the ratings in ds$ratings$NL reflect the contents of the FP worksheet.

3.5 The TP ratings

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



	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating				
2	0	0	70	1	5.28				
3	0	0	70	2	4.65				
4	0	0	71	1	3.01				
5	0	0	72	1	5.98				
6	0	0	73	1	5				
7	0	0	73	2	5.25				
8	0	0	74	1	4.26				
9	1	0	70	1	5.14				
10	1	0	71	1	3.31				
11	1	0	72	1	4.92				
12	1	0	72	2	5.11				
13	1	0	72	3	4.63				
14	1	0	73	1	4.95				
15	1	0	74	1	5.3				
16	2	0	70	1	4.66				
17	2	0	71	1	4.03				
18	2	0	72	1	5.22				
19	2	0	73	1	4.94				
20	2	0	74	1	5.27				
21	0	1	70	1	5.2				
22	0	1	71	1	3.27				
23	0	1	72	1	4.61				
24	0	1	73	1	5.18				

This worksheet can only have diseased cases. The presence of a non-diseased case will generate an error. The common vertical length, 31 in this example, is a-priori unpredictable (as some lesions may not be marked). The maximum possible length, assuming every lesion is marked for each modality, reader and diseased case, is 9 X 2 X 3 = 54. The 9 comes from the total number of non-zero entries in the `LesionID` column of the `Truth` worksheet, the 2 from the number of modalities and 3 from the number of readers.

The fact that the actual length (31) is smaller than the maximum length (54) means that there are combinations of modality, reader and diseased cases on which some lesions were not marked.

As examples, line 2 in the worksheet, the first lesion in `CaseID` equal to `70` was marked (and rated 5.28) in `ModalityID 0` and `ReaderID 0`. Line 3 in the worksheet, the second lesion in `CaseID` equal to `70` was also marked (and rated 4.65) in `ModalityID 0` and `ReaderID 0`. However, lesions 2 and 3 in `CaseID` = 72 were not marked (line 5 in the worksheet indicates that for this modality-reader-case combination only the first lesion was marked). The reader should confirm that the ratings in `ds$ratings$LL` reflect the contents of the `TP` worksheet.

3.6 Reading the FROC dataset

The example shown above corresponds to file `R/quick-start/frocCr.xlsx` in the project directory. The next code reads this file into an `R` object `ds`.

```
frocCr <- "R/quick-start/frocCr.xlsx"
ds <- DfReadDataFile(frocCr, newExcelFileFormat = TRUE)
str(ds)
#> List of 3
#>  $ ratings     :List of 3
#>   ..$ 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 ...
#>   ..$ LL_IL: logi NA
#>  $ lesions     :List of 3
#>   ..$ perCase: int [1:5] 2 1 3 2 1
#>   ..$ IDs    : num [1:5, 1:3] 1 1 1 1 1 ...
#>   ..$ weights: num [1:5, 1:3] 0.3 1 0.333 0.1 1 ...
#>  $ descriptions:List of 7
#>   ..$ fileName     : chr "frocCr"
#>   ..$ type         : chr "FROC"
#>   ..$ name         : logi NA
#>   ..$ truthTableStr: num [1:2, 1:3, 1:8, 1:4] 1 1 1 1 1 1 1 1 1 1 ...
#>   ..$ design       : chr "FCTRL"
#>   ..$ 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"
```

This follows the general description in Chapter 2. The differences are described below.

- The `ds$descriptions$type` member indicates that this is an `FROC` dataset.
- The `ds$lesions$perCase` member is a vector containing the number of lesions in each diseased case, i.e., 2, 1, 3, 2, 1 in the current example.
- The `ds$lesions$IDs` member indicates the labeling of the lesions in each diseased case.

```
ds$lesions$IDs
#>      [,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 denoting 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 (except for the third diseased case).

```
ds$lesions$weights
#>           [,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.

3.7 The distribution of lesions in diseased cases

Consider a much larger real dataset, `dataset11`, with structure as shown below (for descriptions of all embedded datasets see Chapter 12):

```
ds <- dataset11
str(ds)
#> List of 3
#>  $ ratings     :List of 3
#>   ..$ 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 ...
#>   ..$ LL_IL: logi NA
#>  $ lesions     :List of 3
#>   ..$ perCase: int [1:115] 6 4 7 1 3 3 3 8 11 2 ...
#>   ..$ IDs    : num [1:115, 1:20] 1 1 1 1 1 1 1 1 1 1 ...
#>   ..$ weights: num [1:115, 1:20] 0.167 0.25 0.143 1 0.333 ...
#>  $ descriptions:List of 7
#>   ..$ fileName     : chr "dataset11"
#>   ..$ type         : chr "FROC"
#>   ..$ name         : chr "DOBBINS-1"
#>   ..$ truthTableStr: num [1:4, 1:5, 1:158, 1:21] 1 1 1 1 1 1 1 1 1 1 ...
#>   ..$ design       : chr "FCTRL"
#>   ..$ 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" ...
```

The large number of lesions is explained by the fact that this is a volumetric CT image for lung nodule detection (each nodule was verified by 3 radiologists).

Focus on the 115 diseased cases: the numbers of lesions in individual cases is contained in `ds$lesions$perCase`.

```
ds$lesions$perCase
#>   [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 diseased case contains 6 lesions, the second contains 4 lesions, the third contains 7 lesions, etc., and the last diseased case contains 1 lesion. To get the distribution of the numbers of lesions per diseased cases one could use the `which()` function:

```
for (el in 1:max(ds$lesions$perCase)) cat(
  "number of diseased cases with", el, "lesions = ",
  length(which(ds$lesions$perCase == el)), "\n")
#> number of diseased cases with 1 lesions =  25
#> number of diseased cases with 2 lesions =  23
#> number of diseased cases with 3 lesions =  13
#> number of diseased cases with 4 lesions =  10
#> number of diseased cases with 5 lesions =  5
#> number of diseased cases with 6 lesions =  11
#> number of diseased cases with 7 lesions =  6
#> number of diseased cases with 8 lesions =  8
#> number of diseased cases with 9 lesions =  2
#> number of diseased cases with 10 lesions =  3
#> number of diseased cases with 11 lesions =  3
#> number of diseased cases with 12 lesions =  3
#> number of diseased cases with 13 lesions =  0
#> number of diseased cases with 14 lesions =  0
#> number of diseased cases with 15 lesions =  0
#> number of diseased cases with 16 lesions =  2
#> number of diseased cases with 17 lesions =  0
#> number of diseased cases with 18 lesions =  0
#> number of diseased cases with 19 lesions =  0
#> number of diseased cases with 20 lesions =  1
```

This tells us that 25 cases contain 1 lesion. Likewise, 23 cases contain 2 lesions, etc. Note that there are no cases with 13, 14, 15, 17, 18, and 19 lesions.

3.7.1 Definition of `lesID` array

The fraction of diseased cases with 1 lesion, 2 lesions etc, can be calculated as follows:

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

Fraction 0.217 of diseased cases contain 1 lesion, fraction 0.2 of (diseased) cases contain 2 lesions, etc.

This information is more readily obtained using the `RJafroc` function `UtilLesDistr()` as shown next (be sure to view both screens):

```
UtilLesDistr(ds)
#>    lesID        Freq
#> 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    13 0.000000000
#> 14    14 0.000000000
#> 15    15 0.000000000
#> 16    16 0.017391304
#> 17    17 0.000000000
#> 18    18 0.000000000
#> 19    19 0.000000000
#> 20    20 0.008695652
```

- The `UtilLesDistr()` function returns a dataframe with two columns.
- The first column (`lesID`) contains the number of lesions per case.
- The second column (`Freq`) contains the fraction of diseased cases with the number of lesions indicated in the first column.
- The second column sums to unity:

```
sum(UtilLesDistr(ds)$Freq)
#> [1] 1
```

3.8 Lesion weights

- This `dataframe` is returned by `UtilLesWghtsDS()` or `UtilLesWghtsLD()`.
- This contains the same number of rows as `lesID`.
- The number of columns is one plus the number of rows.
- The first column contains the number of lesions per case.
- The second through the last column contain the weights of cases with number of lesions per case in column 1.
- Missing values are filled with `-Inf`.

```
UtilLesWghtsDS(ds, relWeights = 0)
#>       [,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,]   13 0.07692308 0.07692308 0.07692308 0.07692308 0.07692308 0.07692308
#> [14,]   14 0.07142857 0.07142857 0.07142857 0.07142857 0.07142857 0.07142857
#> [15,]   15 0.06666667 0.06666667 0.06666667 0.06666667 0.06666667 0.06666667
#> [16,]   16 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000
#> [17,]   17 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353
#> [18,]   18 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
#> [19,]   19 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
#> [20,]   20 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000
#>              [,8]       [,9]      [,10]      [,11]      [,12]      [,13]
#>  [1,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [2,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [3,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [4,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [5,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [6,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [7,] 0.14285714       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [8,] 0.12500000 0.12500000       -Inf       -Inf       -Inf       -Inf
#>  [9,] 0.11111111 0.11111111 0.11111111       -Inf       -Inf       -Inf
#> [10,] 0.10000000 0.10000000 0.10000000 0.10000000       -Inf       -Inf
#> [11,] 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909       -Inf
#> [12,] 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333 0.08333333
#> [13,] 0.07692308 0.07692308 0.07692308 0.07692308 0.07692308 0.07692308
#> [14,] 0.07142857 0.07142857 0.07142857 0.07142857 0.07142857 0.07142857
#> [15,] 0.06666667 0.06666667 0.06666667 0.06666667 0.06666667 0.06666667
#> [16,] 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000 0.06250000
#> [17,] 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353
#> [18,] 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
#> [19,] 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
#> [20,] 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000
#>             [,14]      [,15]      [,16]      [,17]      [,18]      [,19]
#>  [1,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [2,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [3,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [4,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [5,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [6,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [7,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [8,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#>  [9,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#> [10,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#> [11,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#> [12,]        -Inf       -Inf       -Inf       -Inf       -Inf       -Inf
#> [13,] 0.07692308       -Inf       -Inf       -Inf       -Inf       -Inf
#> [14,] 0.07142857 0.07142857       -Inf       -Inf       -Inf       -Inf
#> [15,] 0.06666667 0.06666667 0.06666667       -Inf       -Inf       -Inf
#> [16,] 0.06250000 0.06250000 0.06250000 0.06250000       -Inf       -Inf
#> [17,] 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353       -Inf
#> [18,] 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
#> [19,] 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158 0.05263158
#> [20,] 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000 0.05000000
```

```
#>            [,20] [,21]
#>  [1,]       -Inf  -Inf
#>  [2,]       -Inf  -Inf
#>  [3,]       -Inf  -Inf
#>  [4,]       -Inf  -Inf
#>  [5,]       -Inf  -Inf
#>  [6,]       -Inf  -Inf
#>  [7,]       -Inf  -Inf
#>  [8,]       -Inf  -Inf
#>  [9,]       -Inf  -Inf
#> [10,]       -Inf  -Inf
#> [11,]       -Inf  -Inf
#> [12,]       -Inf  -Inf
#> [13,]       -Inf  -Inf
#> [14,]       -Inf  -Inf
#> [15,]       -Inf  -Inf
#> [16,]       -Inf  -Inf
#> [17,]       -Inf  -Inf
#> [18,]       -Inf  -Inf
#> [19,] 0.05263158  -Inf
#> [20,] 0.05000000  0.05
## or
## UtilLesWghtsLD(UtilLesDistr(ds), relWeights = 0)
##
```

- Row 3 corresponds to 3 lesions per case and the weights are 1/3, 1/3 and 1/3.
- Row 13 corresponds to 13 lesions per case and the weights are 0.06250000, 0.06250000, ..., repeated 13 times.
- Note that the number of rows equals the maximum number of lesions per case (20).

Chapter 4

LROC data format

4.1 How much finished 75%

4.2 Introduction

In the Localization Receiver Operating Characteristic (LROC) paradigm (Starr et al., 1977, 1975; Swensson, 1996) the observer assigns an overall ROC-rating to each case and marks the most suspicious region in each case. Additionally, each diseased case has *exactly* one lesion. On a diseased case and if the mark is close to the real lesion, the mark is scored as a correct-localization (`CL`) and otherwise it is scored as an incorrect-localization (`IL`). On a non-diseased case the mark is always classified as a false-positive (`FP`).

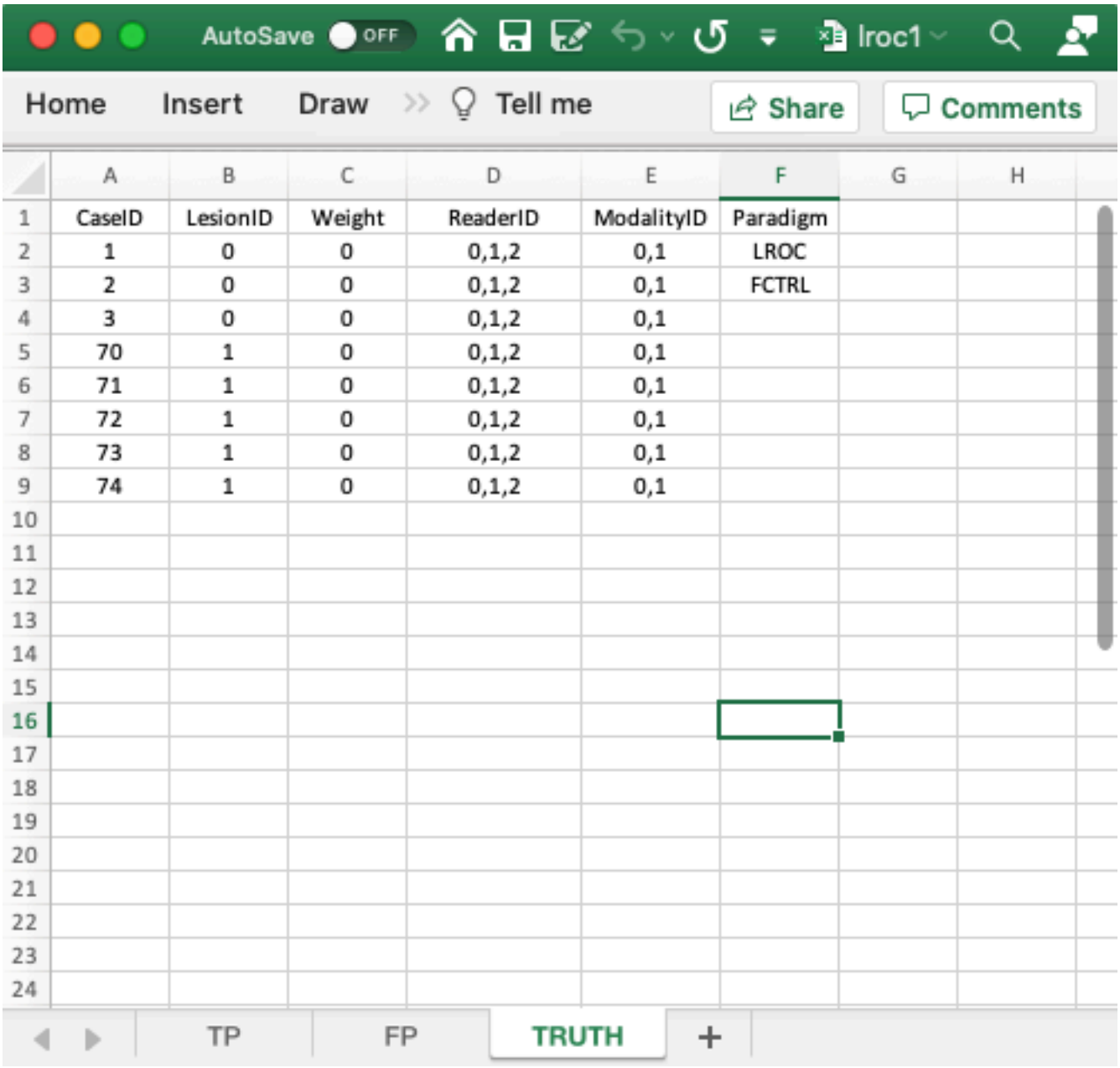
4.3 Forced vs. not-forced marks

The paradigm is illustrated with two toy data files, `R/quick-start/lroc1.xlsx` and `R/quick-start/lroc2.xlsx`. These files illustrate two-modality three-reader LROC datasets with 3 non-diseased and 5 diseased cases.

- The `Truth` worksheet is common to both files.
- File `R/quick-start/lroc1.xlsx` illustrates the classic (i.e., as originally introduced) LROC paradigm where *one mark per case is forced/required*.
- File `R/quick-start/lroc2.xlsx` illustrates the paradigm when one mark-rating pair per case is not forced. There is some history behind this: the basic issue was what was the observer supposed to do when there there was nothing to report. Swensson initially thought that even if there was nothing to report, there must be a region, selected from the set of very low confidence regions, which was most likely to be a lesion (like the maximum of the set of minimums). Most radiologists had difficulty with the forced localization requirement - if they see nothing suspicious, why should they be forced to mark a most suspicious location. The paradigm was subsequently altered so that if the confidence level was below a certain value, say 12 percent on a 0 to 100 scale, the radiologist did not have to report a location. LROCFIT software was modified accordingly, and internal to the software the mark was assigned a random location - which ended up being classified as an incorrect-localization in most cases.

4.4 `Truth` worksheet

- The `Truth` worksheet is similar to that described previously for the ROC and FROC paradigms. The only difference is the first entry in the `Paradigm` column, which is `LROC`.



	A	B	C	D	E	F	G	H
1	CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm		
2	1	0	0	0,1,2	0,1	LROC		
3	2	0	0	0,1,2	0,1	FCTRL		
4	3	0	0	0,1,2	0,1			
5	70	1	0	0,1,2	0,1			
6	71	1	0	0,1,2	0,1			
7	72	1	0	0,1,2	0,1			
8	73	1	0	0,1,2	0,1			
9	74	1	0	0,1,2	0,1			
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

Figure 4.1: The common Truth worksheet for LROC Excel files 'R/quick-start/lroc1.xlsx' and 'R/quick-start/lroc2.xlsx'.

- Since each diseased case has one lesion, the first five columns contain as many rows as there are cases in the dataset. There being 8 cases in the dataset, there are 8 rows of data.
- `CaseID`: unique **integers** representing the cases in the dataset: '1', '2', '3', the 3 non-diseased cases, and '70', '71', '72', '73', '74', the 5 diseased cases.

- `LesionID`: integers 0 or 1.
  - Each 0 represents a non-diseased case,
  - Each 1 represents the sole lesion in the diseased case.
- 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).
- `Weight`: this column is filled with zeroes. As with the ROC paradigm, with one lesion per case the weights are irrelevant.
- `ReaderID`: In the example shown each cell has the value '0, 1, 2'. There are 3 readers in the dataset, labeled `0`, `1` and `2`.
- `ModalityID`: In the example each cell has the value `0, 1`. There are 2 modalities in the dataset, labeled `0` and `1`.
- `Paradigm`: The contents are `LROC` and `FCTRL`: this is an `LROC` dataset and the design is "factorial".

4.5 TP worksheet, forced localization true

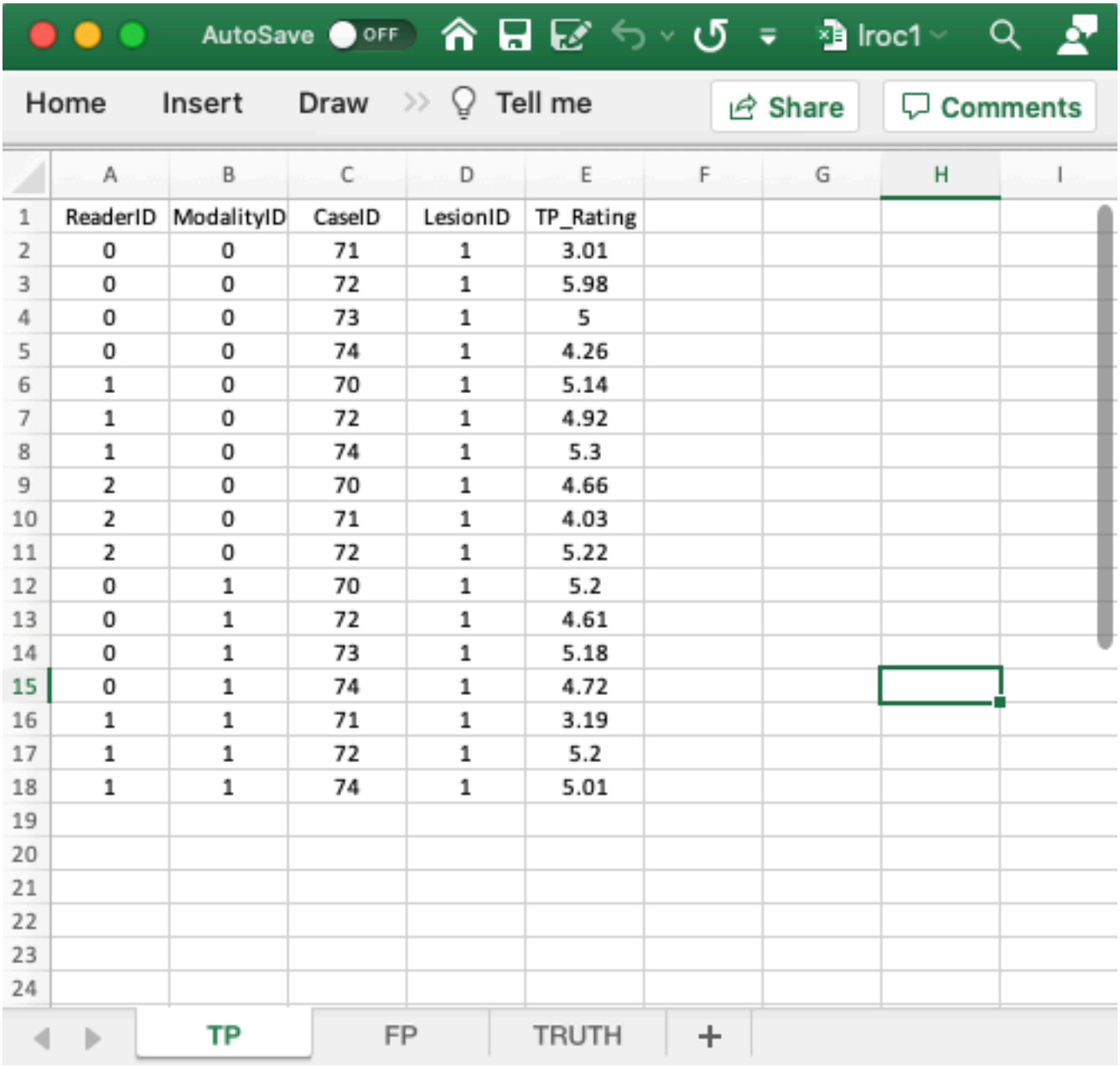
- The `TP` worksheet is similar to that described previously for the ROC and FROC paradigms.
- However, in the LROC paradigm this worksheet records correct localizations only.
- This worksheet can only have diseased cases. The presence of a non-diseased case in this worksheet will generate an error.
- The key difference is that for each modality-reader-diseased-case there can be at most one entry. Also, if a particular combination is missing in the TP worksheet then it must appear in the FP worksheet. This is because this is a forced-mark-per-case dataset.
- There can be at most 30 rows of data in this worksheet: 2 modalities times 3 readers times 5 diseased cases. Since there in fact only 17 rows of data, the missing 13 rows must occur in the FP worksheet.
- Recall that each entry in the TP worksheet represents a correct localization while each missing entry represents an incorrect localization. The incorrect localizations are recorded in the FP worksheet.

4.6 FP worksheet, forced localization true

- The `FP` worksheet is similar to that described previously for the ROC and FROC paradigms.
- Because of the forced mark requirement, there are 18 rows of data corresponding to non-diseased cases: 2 modalities times 3 readers times 3 non-diseased cases. The missing 13 rows from the TP worksheet are listed next; these correspond to the incorrect localizations on diseased cases. Therefore, the total number of rows in this worksheet is 18 + 13 = 31.
- As an example, it is seen that for `modalityID` = 0 and `readerID` = 0, `caseID` = 70 does not appear in the TP worksheet. The lesion on this case was not correctly localized; therefore it appears in the FP worksheet as an incorrect localization.
- As another example, for `modalityID` = 0 and `readerID` = 1, `caseID` = 71 does not appear in the TP worksheet; instead it appears in the FP worksheet.
- As a final example, for `modalityID` = 1 and `readerID` = 2, none of the diseased cases appears in the TP worksheet; instead they all appear in the FP worksheet.

4.7 Reading forced localization true LROC dataset

The images shown above correspond to file `R/quick-start/lroc1.xlsx`. The next code reads this file into an `R` object `ds1`. Note the usage of the `lrocForcedMark` flag, which is set to `TRUE`, because this is a forced localization LROC dataset.



	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating				
2	0	0	71	1	3.01				
3	0	0	72	1	5.98				
4	0	0	73	1	5				
5	0	0	74	1	4.26				
6	1	0	70	1	5.14				
7	1	0	72	1	4.92				
8	1	0	74	1	5.3				
9	2	0	70	1	4.66				
10	2	0	71	1	4.03				
11	2	0	72	1	5.22				
12	0	1	70	1	5.2				
13	0	1	72	1	4.61				
14	0	1	73	1	5.18				
15	0	1	74	1	4.72				
16	1	1	71	1	3.19				
17	1	1	72	1	5.2				
18	1	1	74	1	5.01				
19									
20									
21									
22									
23									
24									

Figure 4.2: The TP worksheet for forced localization true LROC Excel file 'R/quick-start/lroc1.xlsx'.

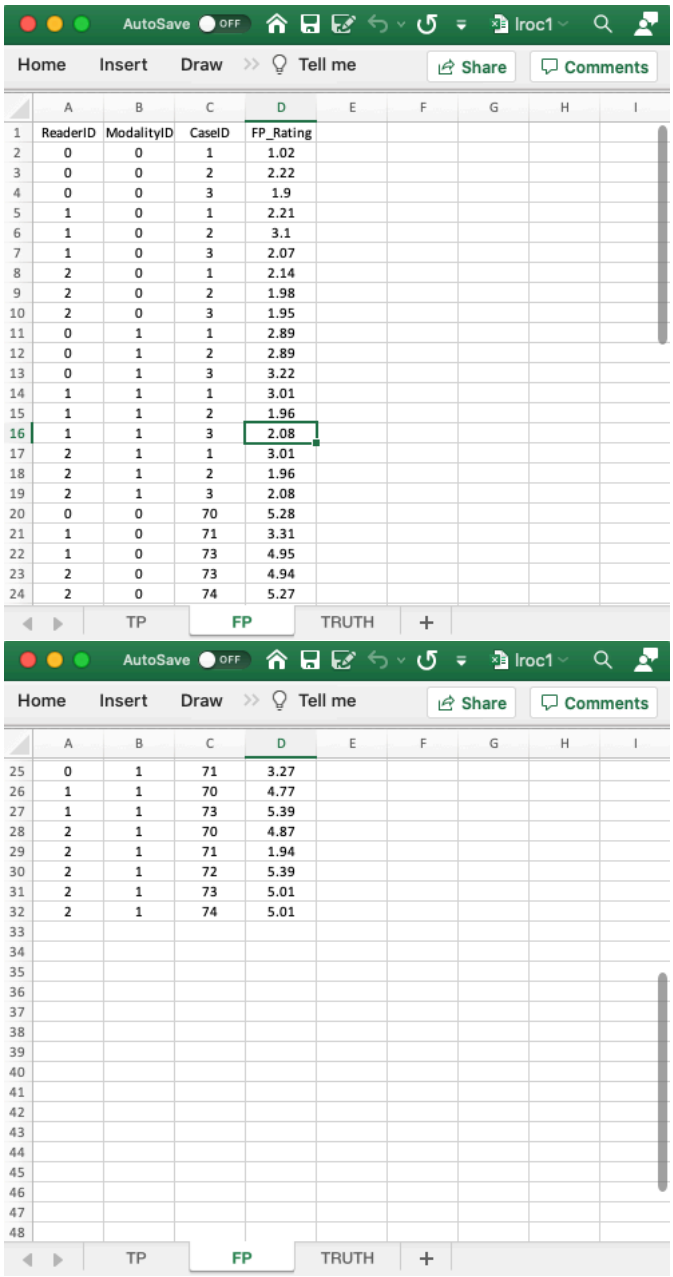


Figure 4.3: The FP worksheet (continued from left panel to right panel) for forced localization LROC Excel file 'R/quick-start/lroc1.xlsx'.

```
lroc1 <- "R/quick-start/lroc1.xlsx"
ds1 <- DfReadDataFile(lroc1, newExcelFileFormat = TRUE, lrocForcedMark = T)
str(ds1)
#> List of 3
#>  $ ratings     :List of 3
#>   ..$ NL   : num [1:2, 1:3, 1:8, 1] 1.02 2.89 2.21 3.01 2.14 3.01 2.22 2.89 3.1 1.96 ...
#>   ..$ LL   : num [1:2, 1:3, 1:5, 1] -Inf 5.2 5.14 -Inf 4.66 ...
#>   ..$ LL_IL: num [1:2, 1:3, 1:5, 1] 5.28 -Inf -Inf 4.77 -Inf ...
#>  $ lesions     :List of 3
#>   ..$ perCase: int [1:5] 1 1 1 1 1
#>   ..$ IDs    : num [1:5, 1] 1 1 1 1 1
#>   ..$ weights: num [1:5, 1] 1 1 1 1 1
#>  $ descriptions:List of 7
#>   ..$ fileName     : chr "lroc1"
#>   ..$ type         : chr "LROC"
#>   ..$ name         : logi NA
#>   ..$ truthTableStr: num [1:2, 1:3, 1:8, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
#>   ..$ design       : chr "FCTRL"
#>   ..$ 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"
```

This follows the general description in Chapter 2. The differences are described below.

- `ds1$ratings$NL` is a [2,3,8,1] dimension vector. For each modality and reader, only the first three elements, corresponding to the three non-diseased cases, are finite, the rest are `-Inf`.

For example:

```
ds1$ratings$NL[1,1,,1]
#> [1] 1.02 2.22 1.90 -Inf -Inf -Inf -Inf -Inf
```

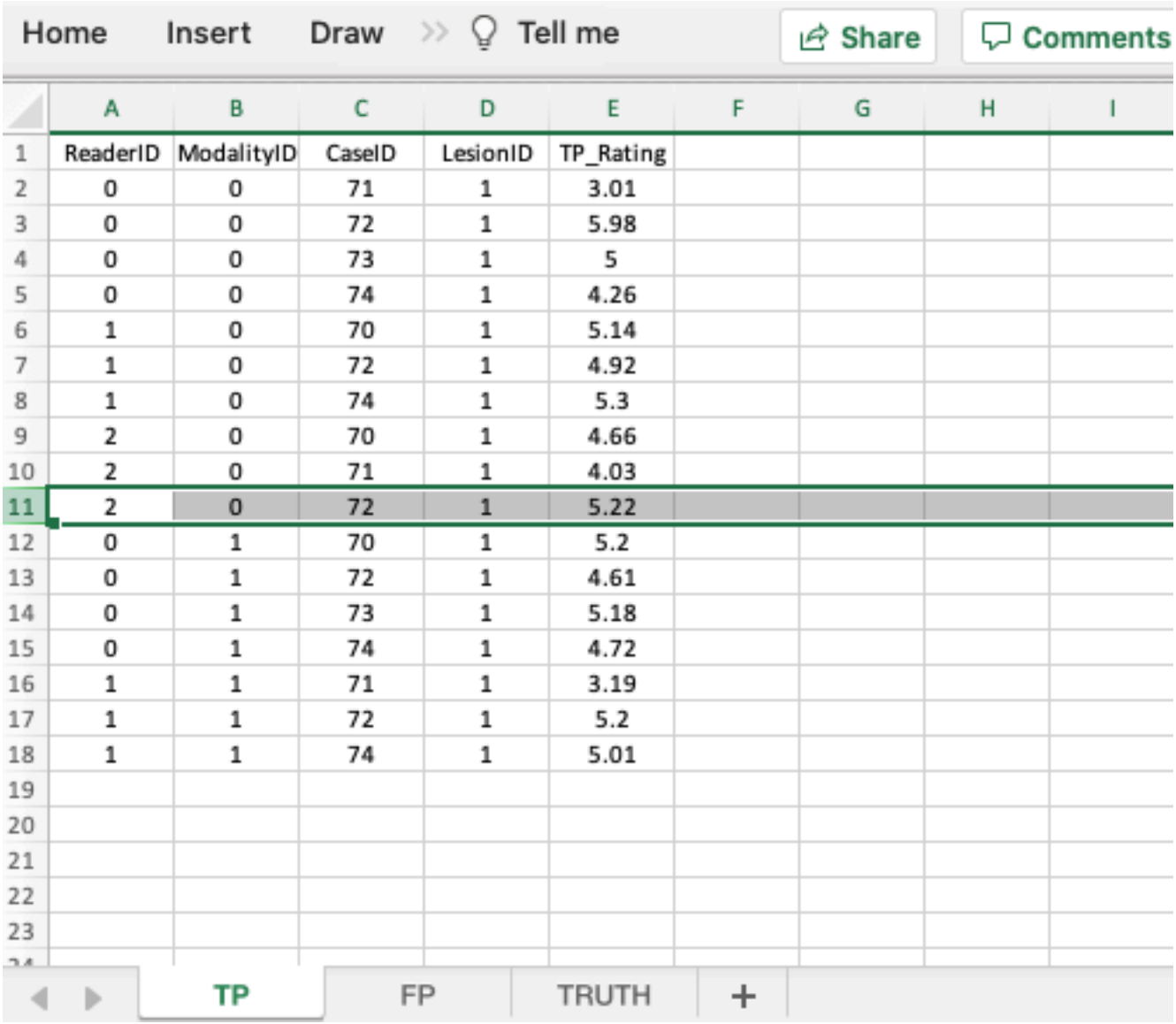
- `ds1$ratings$LL` is a [2,3,5,1] dimension vector. For each modality and reader, only the first three elements, corresponding to the three non-diseased cases, are finite, the rest are `-Inf`.

For example, since none of the lesions are localized for `modalityID` = 1 (second modality) and `readerID` = 2 (third reader), the following code yields a vector consisting of five `-Inf` values:

```
ds1$ratings$LL[2,3,,1]
#> [1] -Inf -Inf -Inf -Inf -Inf
```

- `ds1$ratings$LL_IL` is a [2,3,5,1] dimension vector. These contain the ratings of incorrect localizations on diseased cases. For the just preceding modality-reader combination, this yields a vector with 5 finite values, the ratings of incorrect localizations for `modalityID` = 1 and `readerID` = 2.

```
ds1$ratings$LL_IL[2,3,,1]
#> [1] 4.87 1.94 5.39 5.01 5.01
```

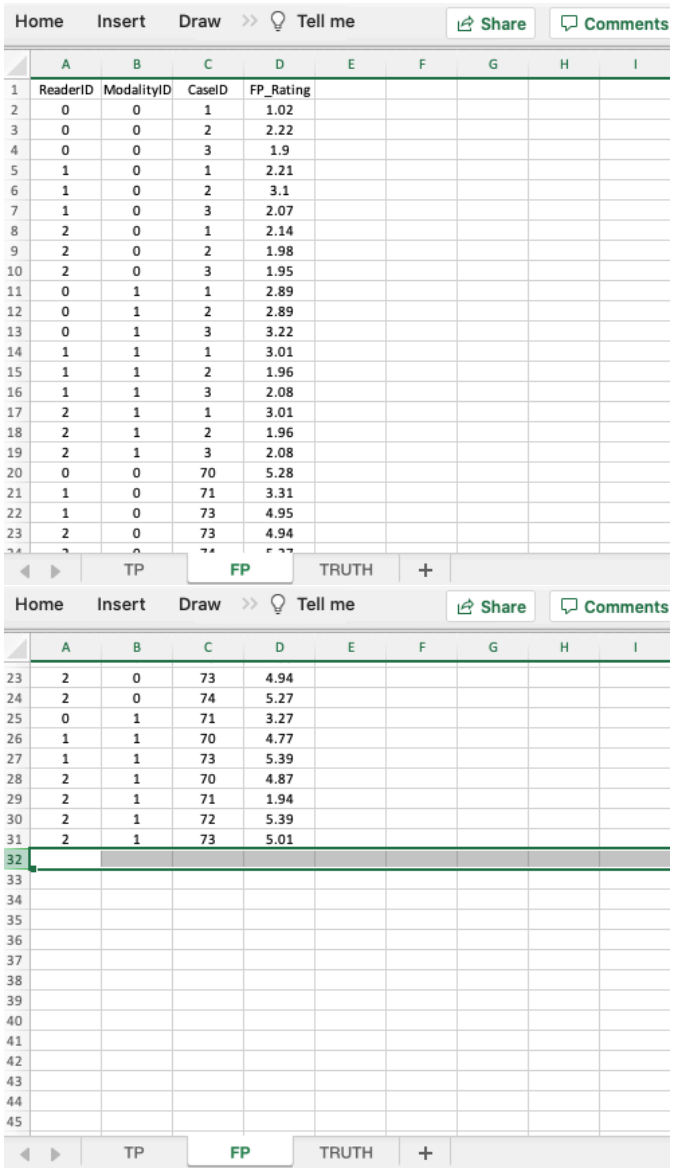


Home Insert Draw Tell me Share Comments

	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating				
2	0	0	71	1	3.01				
3	0	0	72	1	5.98				
4	0	0	73	1	5				
5	0	0	74	1	4.26				
6	1	0	70	1	5.14				
7	1	0	72	1	4.92				
8	1	0	74	1	5.3				
9	2	0	70	1	4.66				
10	2	0	71	1	4.03				
11	2	0	72	1	5.22				
12	0	1	70	1	5.2				
13	0	1	72	1	4.61				
14	0	1	73	1	5.18				
15	0	1	74	1	4.72				
16	1	1	71	1	3.19				
17	1	1	72	1	5.2				
18	1	1	74	1	5.01				
19									
20									
21									
22									
23									

TP FP TRUTH +

Figure 4.4: The TP worksheet for forced localization false LROC Excel file 'R/quick-start/lroc2.xlsx'.



	A	B	C	D
1	ReaderID	ModalityID	CaseID	FP_Rating
2	0	0	1	1.02
3	0	0	2	2.22
4	0	0	3	1.9
5	1	0	1	2.21
6	1	0	2	3.1
7	1	0	3	2.07
8	2	0	1	2.14
9	2	0	2	1.98
10	2	0	3	1.95
11	0	1	1	2.89
12	0	1	2	2.89
13	0	1	3	3.22
14	1	1	1	3.01
15	1	1	2	1.96
16	1	1	3	2.08
17	2	1	1	3.01
18	2	1	2	1.96
19	2	1	3	2.08
20	0	0	70	5.28
21	1	0	71	3.31
22	1	0	73	4.95
23	2	0	73	4.94

	A	B	C	D
23	2	0	73	4.94
24	2	0	74	5.27
25	0	1	71	3.27
26	1	1	70	4.77
27	1	1	73	5.39
28	2	1	70	4.87
29	2	1	71	1.94
30	2	1	72	5.39
31	2	1	73	5.01
32				

Figure 4.5: The FP worksheet (continued from left panel to right panel) for forced localization false LROC Excel file 'R/quick-start/lroc2.xlsx'.

4.8 TP worksheet, forced localization false

4.9 FP worksheet, forced localization false

- If a particular modality-reader-case combination is missing in the TP worksheet then it need not appear in the FP worksheet. This is because this is not a forced-mark-per-case dataset.
- As an example, `modalityID` = 1, `readerID` = 2 and `caseID` = 74 does not appear in either TP or FP worksheets.

4.10 Reading forced localization false LROC dataset

The next example is for file `R/quick-start/lroc2.xlsx`. The following code reads this file into an `R` object `x2`. Note that for this dataset one must set the `lrocForcedMark` flag to `FALSE`, because this is *not* a forced localization LROC dataset. Setting `lrocForcedMark` flag to `TRUE` will generate an error.

```
lroc2 <- "R/quick-start/lroc2.xlsx"
x2 <- DfReadDataFile(lroc2, newExcelFileFormat = TRUE, lrocForcedMark = F)
str(x2)
#> List of 3
#>  $ ratings     :List of 3
#>   ..$ NL   : num [1:2, 1:3, 1:8, 1] 1.02 2.89 2.21 3.01 2.14 3.01 2.22 2.89 3.1 1.96 ...
#>   ..$ LL   : num [1:2, 1:3, 1:5, 1] -Inf 5.2 5.14 -Inf 4.66 ...
#>   ..$ LL_IL: num [1:2, 1:3, 1:5, 1] 5.28 -Inf -Inf 4.77 -Inf ...
#>  $ lesions     :List of 3
#>   ..$ perCase: int [1:5] 1 1 1 1 1
#>   ..$ IDs    : num [1:5, 1] 1 1 1 1 1
#>   ..$ weights: num [1:5, 1] 1 1 1 1 1
#>  $ descriptions:List of 7
#>   ..$ fileName     : chr "lroc2"
#>   ..$ type         : chr "LROC"
#>   ..$ name         : logi NA
#>   ..$ truthTableStr: num [1:2, 1:3, 1:8, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
#>   ..$ design       : chr "FCTRL"
#>   ..$ 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"
```

- The `x2$ratings$LL` array is a [2,3,5,1] dimension vector. For each modality and reader, only the first three elements, corresponding to the three non-diseased cases, are finite, the rest are `-Inf`.

For example, since none of the lesions are localized for `modalityID` = 1 (second modality) and `readerID` = 2 (third reader), the following code yields a vector consisting of five `-Inf` values:

```
x2$ratings$LL[2,3,,1]
#> [1] -Inf -Inf -Inf -Inf -Inf
```

- The `x2$ratings$LL_IL` is a [2,3,5,1] dimension vector. These contain the ratings of incorrect localizations on diseased cases. For the just preceding modality-reader combination, this yields a vector with 4 finite values, the ratings of incorrect localizations for `modalityID` = 1 and `readerID` = 2.

```
x2$ratings$LL_IL[2,3,,1]
#> [1] 4.87 1.94 5.39 5.01 -Inf
```

For this modality-reader combination case 74 (i.e., the fifth diseased case) was unmarked. It does not appear in either the TP or the FP worksheet.

4.11 Summary

The difference from the previous data structures is the existence of `LL_IL` in the `ratings` list, which contains the ratings of incorrect localizations. Recall that for ROC and FROC paradigms this member was `NA`. When the data obeys forced localization, the corresponding flag should be set to `TRUE`, otherwise it should be set to `FALSE`. The default value of this flag is `NA`, which will work for ROC or FROC datasets. For LROC datasets it should be set to `T/F`.

Choosing an appropriate figure of merit

4.12 How much finished 0 percent

WARNING: Usage of FOM = HrSe or FOM = HrSp is strongly discouraged. Consider comparing two readers or two treatments using either of these FOMs. The rating is a *subjective ordered label*. It need not be used consistently between readers and treatments. A reader using a strict reporting criteria, who only marks a lesion when he is very confident, will have smaller HrSe and larger HrSp than a reader who adopts a laxer criteria, even though true performance, as measured by ROC AUC or percentage correct in 2AFC task, are identical. This is ROC-101: ROC AUC was recommended by Metz, ca. 1978 instead of sensitivity or specificity.

4.13 Introduction

Assuming the study has been properly conducted, e.g., ROC or FROC, probably the most important step before beginning to analyze the dataset is to choose an apprpriate figure of merit (i.e., performance metric).

4.14 ROC dataset

In the ROC paradigm every modality-reader-case combination yields a single rating. The appropriate FOM is the Wilcoxon statistic, which is identical to the AUC under the empirical ROC curve.

4.15 FROC dataset

In the FROC paradigm every modality-reader-case combination yields a random number (zero or more) of mark-rating pairs.

4.15.1 FOM = wAFROC

For most FROC datasets the appropriate FOM is the AUC under the weighted AFROC plot, as illustrated next for `daset05` which has two modalities and 9 readers.

```
fom_wAFROC <- UtilFigureOfMerit(dataset = dataset05, FOM = "wAFROC")
as.data.frame(lapply(fom_wAFROC, format, decimal.mark = ".", digits = 4))
```

```
##   X.0.7245. X.0.8024. X.0.881. X.0.9686. X.0.8096. X.0.846. X.0.6133. X.0.7514.
## 1    0.7245    0.8024    0.881    0.9686    0.8096    0.846    0.6133    0.7514
##   X.0.5773. X.0.7209. X.0.8493. X.0.8719. X.0.8928. X.0.937. X.0.8026.
## 1    0.5773    0.7209    0.8493    0.8719    0.8928    0.937    0.8026
##   X.0.8995. X.0.764. X.0.819.
## 1    0.8995    0.764    0.819
```

4.15.2 FOM = HrSe

Recall that the concepts of sensitivity and specificity are reserved for ROC data - i.e., one rating per case. To compute these from FROC data one needs a method for inferring a single rating from the possibly multiple (zero or more) ratings ocurring on each case (if the case has no marks one assings a rating that is smaller than any the ratings of explicitly marked locations, e.g., minus infinity). The recommended procedure is to assign the rating of the highest rated mark on each case, of $-\infty$ if the case has no marks, as its inferred ROC rating. This has the has the effect of converting the FROC dataset to an inferred ROC dataset. The function `DfFroc2Roc` does exactly this:

```
dataset05$descriptions$type
```

```
## [1] "FROC"
```

```
ds <- DfFroc2Roc(dataset05)
ds$descriptions$type
```

```
## [1] "ROC"
```

`HrSe` is the abbreviation for "highest rating sensitivity", sensitivity derived from the rating of the highest rated mark on each case. Replacing the possibly multiple ratings occurring on each case with the highest rating amounts to an assumption, a very good one in my opinion. Since the ratings are ordered labels (i.e., non-numeric vaues) any numerical computation, such as the average, would be invalid. It is also common sense: if a case has 3 marks rated 80, 30 and 15, why would the ROC rating be anything but 80. Finally, there is historical precedence for this assumption: (Bunch et al., 1977; Swensson, 1996).

Usage of `FOM = HrSe` is illustrated next for `dataset05`.

```
fom_HrSe <- UtilFigureOfMerit(dataset = dataset05, FOM = "HrSe")
as.data.frame(lapply(fom_HrSe, format, decimal.mark = ".", digits = 4))
```

```
##   X.0.9362. X.1. X.0.8298. X.0.9574. X.0.8936. X.0.9574..1 X.0.7021. X.0.8511.
## 1    0.9362    1    0.8298    0.9574    0.8936      0.9574    0.7021    0.8511
##   X.0.8298..1 X.0.8511..1 X.0.9574..2 X.1..1 X.0.8723. X.0.9362..1 X.0.8936..1
## 1      0.8298      0.8511      0.9574      1    0.8723      0.9362      0.8936
##   X.0.9149. X.0.8936..2 X.0.9787.
## 1    0.9149      0.8936    0.9787
```

Notice that each listed value is greater TBA?? than the corresponding value when using `FOM = "wAFROC"`. This should not come as a surprise as

```
for (i in 1:2)
    for (j in 1:9) {
        cat("i = ", i, ", j = ", j, "\n")
        if (fom_HrSe[i,j] > fom_wAFROC[i,j]) cat ("TRUE \n") else cat("FALSE \n")
}
```

```
## i =  1 , j =  1
## TRUE
## i =  1 , j =  2
## FALSE
## i =  1 , j =  3
## TRUE
## i =  1 , j =  4
## TRUE
```

```
## i =  1 , j =  5
## TRUE
## i =  1 , j =  6
## TRUE
## i =  1 , j =  7
## FALSE
## i =  1 , j =  8
## TRUE
## i =  1 , j =  9
## TRUE
## i =  2 , j =  1
## TRUE
## i =  2 , j =  2
## FALSE
## i =  2 , j =  3
## TRUE
## i =  2 , j =  4
## TRUE
## i =  2 , j =  5
## TRUE
## i =  2 , j =  6
## TRUE
## i =  2 , j =  7
## FALSE
## i =  2 , j =  8
## TRUE
## i =  2 , j =  9
## TRUE
```

Chapter 5

DBM analysis text output

5.1 TBA How much finished

50%

5.2 Introduction

This chapter illustrates significance testing using the DBM method.

5.3 Analyzing the ROC dataset

This illustrates the `St()` function. The significance testing method is specified as `"DBM"` and the figure of merit `FOM` is specified as "Wilcoxon". The embedded dataset `dataset03` is used.

```
ret <- St(dataset03, FOM = "Wilcoxon", method = "DBM")
```

5.4 Explanation of the output

The function returns a list with 5 members:

- `FOMs`: figures of merit.
- `ANOVA`: ANOVA tables.
- `RRRC`: random-reader random-case analyses results.
- `FRRC`: fixed-reader random-case analyses results.
- `RRFC`" random-reader fixed-case analyses results.

Let us consider them individually.

```
str(ret$FOMs)
#> List of 3
#>  $ foms        : num [1:2, 1:4] 0.853 0.85 0.865 0.844 0.857 ...
#>   ..- attr(*, "dimnames")=List of 2
#>   .. ..$ : chr [1:2] "trtTREAT1" "trtTREAT2"
#>   .. ..$ : chr [1:4] "rdrREADER_1" "rdrREADER_2" "rdrREADER_3" "rdrREADER_4"
#>  $ trtMeans    :'data.frame':    2 obs. of  1 variable:
```

```
#>   ..$ Estimate: num [1:2] 0.848 0.837
#>  $ trtMeanDiffs:'data.frame':    1 obs. of  1 variable:
#>   ..$ Estimate: num 0.0109
```

- `FOMs` is a list of 3
  - `foms` is a [2x4] dataframe: the figure of merit for each of of the four observers in the two treatments.
  - `trtMeans` is a [2x1] dataframe: the average figure of merit over all readers for each treatment.
  - `trtMeanDiffs` a [1x1] dataframe: the difference(s) of the reader-averaged figures of merit for all different-treatment pairings. In this example, with only two treatments, there is only one different-treatment pairing.

```
ret$FOMs$foms
#>           rdrREADER_1 rdrREADER_2 rdrREADER_3 rdrREADER_4
#> trtTREAT1   0.8534600   0.8649932   0.8573044   0.8152420
#> trtTREAT2   0.8496156   0.8435097   0.8401176   0.8143374
ret$FOMs$trtMeans
#>            Estimate
#> trtTREAT1 0.8477499
#> trtTREAT2 0.8368951
ret$FOMs$trtMeanDiffs
#>                       Estimate
#> trtTREAT1-trtTREAT2 0.01085482
```

```
str(ret$ANOVA)
#> List of 4
#>  $ TRCanova     :'data.frame':   8 obs. of  3 variables:
#>   ..$ SS: num [1:8] 0.0236 0.2052 52.5284 0.0151 6.41 ...
#>   ..$ DF: num [1:8] 1 3 99 3 99 297 297 799
#>   ..$ MS: num [1:8] 0.02357 0.06841 0.53059 0.00502 0.06475 ...
#>  $ VarCom       :'data.frame':   6 obs. of  1 variable:
#>   ..$ Estimates: num [1:6] 3.78e-05 5.13e-02 -7.13e-04 -2.89e-03 2.79e-02 ...
#>  $ IndividualTrt:'data.frame':   3 obs. of  3 variables:
#>   ..$ DF       : num [1:3] 3 99 297
#>   ..$ trtTREAT1: num [1:3] 0.0493 0.294 0.105
#>   ..$ trtTREAT2: num [1:3] 0.0242 0.3014 0.1034
#>  $ IndividualRdr:'data.frame':   3 obs. of  5 variables:
#>   ..$ DF         : num [1:3] 1 99 99
#>   ..$ rdrREADER_1: num [1:3] 0.000739 0.203875 0.091559
#>   ..$ rdrREADER_2: num [1:3] 0.0231 0.2234 0.0803
#>   ..$ rdrREADER_3: num [1:3] 0.0148 0.2142 0.0612
#>   ..$ rdrREADER_4: num [1:3] 4.09e-05 2.85e-01 6.06e-02
```

- ANOVA is a list of 4
  - `TRCanova` is a [8x3] dataframe: the treatment-reader-case ANOVA table, see below, where SS is the sum of squares, DF is the denominator degrees of freedom and MS is the mean squares, and T = treatment, R = reader, C = case, TR = treatment-reader, TC = treatment-case, RC = reader-case, TRC = treatment-reader-case.
  - `VarCom` is a [6x1] dataframe: the variance components, see below, where `varR` is the reader variance, `varC` is the case variance, `varTR` is the treatment-reader variance, `varTC` is the treatment-case variance, `varRC` is the reader-case variance, and `varTRC` is the treatment-reader-case variance.
  - `IndividualTrt` is a [3x3] dataframe: the individual treatment variance components averaged over all readers, see below, where `msR` is the mean square reader, `msC` is the mean square case and `msRC` is the mean square reader-case.

  - `IndividualRdr` is a [3x5] dataframe: the individual reader variance components averaged over treatments, see below, where `msT` is the mean square treatment, `msC` is the mean square case and `msTC` is the mean square treatment-case.

```
ret$ANOVA$TRCanova
#>                  SS  DF          MS
#> T       0.02356541   1 0.023565410
#> R       0.20521800   3 0.068406000
#> C      52.52839868  99 0.530589886
#> TR      0.01506079   3 0.005020264
#> TC      6.41004881  99 0.064747968
#> RC     39.24295381 297 0.132131158
#> TRC    22.66007764 297 0.076296558
#> Total 121.08532315 799          NA
ret$ANOVA$VarCom
#>            Estimates
#> VarR    3.775568e-05
#> VarC    5.125091e-02
#> VarTR  -7.127629e-04
#> VarTC  -2.887147e-03
#> VarRC   2.791730e-02
#> VarErr  7.629656e-02
ret$ANOVA$IndividualTrt
#>       DF  trtTREAT1  trtTREAT2
#> msR    3 0.04926635 0.02415991
#> msC   99 0.29396753 0.30137032
#> msRC 297 0.10504787 0.10337984
ret$ANOVA$IndividualRdr
#>      DF  rdrREADER_1 rdrREADER_2 rdrREADER_3  rdrREADER_4
#> msT   1 0.0007389761  0.02307702  0.01476929 4.091217e-05
#> msC  99 0.2038747746  0.22344191  0.21424677 2.854199e-01
#> msTC 99 0.0915587344  0.08027926  0.06122898 6.057067e-02
```

```
str(ret$RRRC)
#> List of 3
#>  $ FTests         :'data.frame': 2 obs. of  4 variables:
#>   ..$ DF   : num [1:2] 1 3
#>   ..$ MS   : num [1:2] 0.02357 0.00502
#>   ..$ FStat: num [1:2] 4.69 NA
#>   ..$ p    : num [1:2] 0.119 NA
#>  $ ciDiffTrt      :'data.frame': 1 obs. of  7 variables:
#>   ..$ Estimate: num 0.0109
#>   ..$ StdErr  : num 0.00501
#>   ..$ DF      : num 3
#>   ..$ t       : num 2.17
#>   ..$ PrGTt   : num 0.119
#>   ..$ CILower : num -0.00509
#>   ..$ CIUpper : num 0.0268
#>  $ ciAvgRdrEachTrt:'data.frame': 2 obs. of  5 variables:
#>   ..$ Estimate: num [1:2] 0.848 0.837
#>   ..$ StdErr  : num [1:2] 0.0244 0.0236
#>   ..$ DF      : num [1:2] 70.1 253.6
#>   ..$ CILower : num [1:2] 0.799 0.79
#>   ..$ CIUpper : num [1:2] 0.896 0.883
```

- RRRC, a list of 3 containing results of random-reader random-case analyses

  - `FTtests`: is a [2x4] dataframe: results of the F-tests, see below, where `FStat` is the F-statistic and `p` is the p-value. The first row is the treatment effect and the second is the error term.
  - `ciDiffTrt`: is a [1x7] dataframe: the confidence intervals between different-treatments, see below, where `StdErr` is the standard error of the estimate, `t` is the t-statistic and `PrGTt` is the p-value.
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for each treatment, averaged over all readers in the treatment, see below, where `CILower` is the lower 95% confidence interval and `CIUpper` is the upper 95% confidence interval.

```
ret$RRRC$FTests
#>           DF          MS    FStat         p
#> Treatment  1 0.023565410 4.694058 0.1188379
#> Error      3 0.005020264       NA        NA
ret$RRRC$ciDiffTrt
#>                         Estimate      StdErr DF        t     PrGTt      CILower
#> trtTREAT1-trtTREAT2 0.01085482 0.005010122  3 2.166577 0.1188379 -0.005089627
#>                          CIUpper
#> trtTREAT1-trtTREAT2 0.02679926
ret$RRRC$ciAvgRdrEachTrt
#>            Estimate     StdErr        DF   CILower   CIUpper
#> trtTREAT1 0.8477499 0.02440215  70.12179 0.7990828 0.8964170
#> trtTREAT2 0.8368951 0.02356642 253.64403 0.7904843 0.8833058
```

```
str(ret$FRRC)
#>  NULL
```

- FRRC, a list of 4 containing results of fixed-reader random-case analyses
  - `FTtests`: is a [2x4] dataframe: results of the F-tests, see below.
  - `ciDiffTrt`: is a [1x7] dataframe: the confidence intervals between different-treatments, see below.
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for the average reader over each treatment
  - `ciDiffTrtEachRdr`: is a [4x7] dataframe: the confidence intervals for each different-treatment pairing for each reader.

```
ret$FRRC$FTests
#> NULL
ret$FRRC$ciDiffTrt
#> NULL
ret$FRRC$ciAvgRdrEachTrt
#> NULL
ret$FRRC$ciDiffTrtEachRdr
#> NULL
```

```
str(ret$RRFC)
#>  NULL
```

- RRFC, a list of 3 containing results of random-reader fixed-case analyses
  - `FTtests`: is a [2x4] dataframe: results of the F-tests, see below.
  - `ciDiffTrt`: is a [1x7] dataframe: the confidence intervals between different-treatments, see below.
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for the average reader over each over each treatment.

```
ret$RRFC$FTests
#> NULL
```

```
ret$RRFC$ciDiffTrt
#> NULL
ret$RRFC$ciAvgRdrEachTrt
#> NULL
```

Chapter 6

OR analysis text output

6.1 TBA How much finished

90%

6.2 Introduction

This chapter illustrates significance testing using the DBM and OR methods.

6.3 Analyzing the ROC dataset

The only change is to specify `method = "OR"` in the significance testing function. The same dataset is used as was used in the previous chapter.

```
ret <- St(dataset03, FOM = "Wilcoxon", method = "OR")
```

6.4 Explanation of the output

The function returns a list with 5 members.

- `FOMs`: figures of merit, identical to that in the DBM method.
- `ANOVA`: ANOVA tables.
- `RRRC`: random-reader random-case analyses results.
- `FRRC`: fixed-reader random-case analyses results.
- `RRFC`" random-reader fixed-case analyses results.

Let us consider the ones that are different from the DBM method.

- ANOVA is a list of 4
  - `TRanova` is a [3x3] dataframe: the treatment-reader ANOVA table, see below, where SS is the sum of squares, DF is the denominator degrees of freedom and MS is the mean squares, and T = treatment, R = reader, TR = treatment-reader.

  - `VarCom` is a [6x2] dataframe: the variance components, see below, where `varR` is the reader variance, `varTR` is the treatment-reader variance, `Cov1`, `Cov2`,`Cov3` and `Var` are as defined in the OR model. The second column lists the correlations defined in the OR model.
  - `IndividualTrt` is a [2x4] dataframe: the individual treatment mean-squares, variances and Cov_2, averaged over all readers, see below, where `msREachTrt` is the mean square reader, `varEachTrt` is the variance and `cov2EachTrt` is `Cov2EachTrt` in each treatment.
  - `IndividualRdr` is a [2x4] dataframe: the individual reader variance components averaged over treatments, see below, where `msTEachRdr` is the mean square treatment, `varEachRdr` is the variance and `cov1EachRdr` is Cov_1 for each reader.

```
ret$ANOVA$TRanova
#>              SS DF           MS
#> T  0.0002356541  1 2.356541e-04
#> R  0.0020521800  3 6.840600e-04
#> TR 0.0001506079  3 5.020264e-05
ret$ANOVA$VarCom
#>           Estimates      Rhos
#> VarR   2.331994e-05        NA
#> VarTR -6.838915e-04        NA
#> Cov1   7.916821e-04 0.5188717
#> Cov2   4.836377e-04 0.3169781
#> Cov3   5.125091e-04 0.3359006
#> Var    1.525776e-03        NA
ret$ANOVA$IndividualTrt
#>           DF   msREachTrt  varEachTrt  cov2EachTrt
#> trtTREAT1  3 0.0004926635 0.001522778 0.0004722991
#> trtTREAT2  3 0.0002415991 0.001528775 0.0004949762
ret$ANOVA$IndividualRdr
#>             DF   msTEachRdr  varEachRdr  cov1EachRdr
#> rdrREADER_1  1 7.389761e-06 0.001477168 0.0005615802
#> rdrREADER_2  1 2.307702e-04 0.001518606 0.0007158133
#> rdrREADER_3  1 1.476929e-04 0.001377379 0.0007650890
#> rdrREADER_4  1 4.091217e-07 0.001729953 0.0011242462
```

- RRRC, a list of 3 containing results of random-reader random-case analyses
  - `FTtests`: is a [2x4] dataframe: results of the F-tests, see below, where `FStat` is the F-statistic and `p` is the p-value. The first row is the treatment effect and the second is the error term.
  - `ciDiffTrt`: is a [1x7] dataframe: the confidence intervals between different treatments, see below, where `StdErr` is the standard error of the estimate, `t` is the t-statistic and `PrGTt` is the p-value.
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for the average reader over each treatment, see below, where `CILower` is the lower 95% confidence interval and `CIUpper` is the upper 95% confidence interval.

```
ret$RRRC$FTests
#>           DF           MS    FStat         p
#> Treatment  1 2.356541e-04 4.694058 0.1188379
#> Error      3 5.020264e-05       NA        NA
ret$RRRC$ciDiffTrt
#>                        Estimate      StdErr DF        t     PrGTt      CILower
#> trtTREAT1-trtTREAT2 0.01085482 0.005010122  3 2.166577 0.1188379 -0.005089627
#>                       CIUpper
#> trtTREAT1-trtTREAT2 0.02679926
ret$RRRC$ciAvgRdrEachTrt
#>            Estimate     StdErr        DF   CILower   CIUpper         Cov2
#> trtTREAT1 0.8477499 0.02440215  70.12179 0.7990828 0.8964170 0.0004722991
#> trtTREAT2 0.8368951 0.02356642 253.64403 0.7904843 0.8833058 0.0004949762
```

- FRRC, a list of 5 containing results of fixed-reader random-case analyses
  - `FTtests`: is a [2x4] dataframe: results of the chisquare-tests, see below. Here is a difference from DBM: in the OR method for FRRC the denominator degrees of freedom of the F-statistic is infinite, and the test becomes equivalent to a chisquare test with the degrees of freedom equal to $I - 1$, where I is the number of treatments.
  - `ciDiffTrt`: is a [1x6] dataframe: the confidence intervals between different treatments, see below. An additional column lists
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for the average reader over each treatment
  - `ciDiffTrtEachRdr`: is a [4x6] dataframe: the confidence intervals for each different-treatment pairing for each reader.
  - `IndividualRdrVarCov1`: is a [4x2] dataframe: Var and Cov_1 for individual readers.

```
ret$FRRC$FTests
#> NULL
ret$FRRC$ciDiffTrt
#> NULL
ret$FRRC$ciAvgRdrEachTrt
#> NULL
ret$FRRC$ciDiffTrtEachRdr
#> NULL
ret$FRRC$IndividualRdrVarCov1
#> NULL
```

- RRFC, a list of 3 containing results of random-reader fixed-case analyses
  - `FTtests`: is a [2x4] dataframe: results of the F-tests, see below.
  - `ciDiffTrt`: is a [1x7] dataframe: the confidence intervals between different treatments, see below.
  - `ciAvgRdrEachTrt`: is a [2x5] dataframe: the confidence intervals for the average reader over each over each treatment.

```
ret$RRFC$FTests
#> NULL
ret$RRFC$ciDiffTrt
#> NULL
ret$RRFC$ciAvgRdrEachTrt
#> NULL
```

Chapter 7

OR analysis Excel output

7.1 TBA How much finished

90%

7.2 Introduction

This chapter illustrates significance testing using the OR method. But, instead of the perhaps unwieldy output in Chapter 6, it generates an Excel output file containing the following worksheets:

- `Summary`
- `FOMs`
- `ANOVA`
- `RRRC`
- `FRRC`
- `RRFC`

7.3 Generating the Excel output file

This illustrates the `UtilOutputReport()` function. The arguments are the embedded dataset, `dataset03`, the same dataset as in the previous two chapters, the report file base name `ReportFileBaseName` is set to `R/quick-start/MyResults`, the report file extension `ReportFileExt` is set to `xlsx`, the `FOM` is set to "Wilcoxon", the `method` of analysis is set to "OR", and the flag `overWrite = TRUE` overwrites any existing file with the same name, as otherwise the program will pause for user input.

ROC sample size

Chapter 8

ROC-DBM sample size

8.1 TBA How much finished 10%

8.2 Introduction

The question addressed here is "how many readers and cases", usually abbreviated to *sample-size*, should one employ to conduct a "well-planned" ROC study. The reasons for the quotes will shortly become clear. If cost were no concern, the reply would be: "as many readers and cases as one can get". While there are other considerations affecting sample-size, e.g., the data collection paradigm and the analysis method, this chapter is restricted to the MRMC ROC data collection paradigm with data analyzed by the DBM method described in a previous chapter.

It turns out that provided one can specify conceptually valid effect-sizes between different paradigms (i.e., in the same "units"), the methods described in this chapter are extensible to other paradigms; see TBA Chapter 19 for sample size estimation for FROC studies. *For this reason it is important to understand the concepts of sample-size estimation in the simpler ROC context.*

For simplicity and practicality this chapter is restricted to analysis of two-treatment data ($I = 2$). The purpose of most imaging system assessment studies is to determine, for a given diagnostic task, whether radiologists perform better using a new treatment over the conventional treatment, and whether the difference is statistically significant. Therefore, the two-treatment case is the most common one encountered. While it is possible to extend the methods to more than two treatments, the extensions are not clinically interesting.

Assume the figure of merit (FOM) θ is chosen to be the area AUC under the ROC curve (empirical or fitted is immaterial as far as the methodology and formulae are concerned; however, the choice will affect statistical power). The statistical analysis determines the significance level of the study, i.e., the probability or p-value for incorrectly rejecting the null hypothesis (NH) that the two θs are equal: $NH : \theta_{1\bullet} = \theta_{2\bullet}$, where the subscripts refer to the two treatments and the bullet represents the average over the reader index. If the p-value is smaller than a pre-specified α, typically set at 5%, one rejects the NH and declares the treatments different at the α significance level. Statistical power is the probability of rejecting the null hypothesis when the alternative hypothesis $AH : \theta_1 \neq \theta_2$ is true.

The true value of the difference between the treatments, known as the *true effect-size* is, of course, unknown (if it were known there would be no need to conduct the ROC study: one simply adopts the treatment with the higher θ). Sample-size estimation involves making an educated guess at the true effect-size, called the *anticipated effect size*, denoted d. Increasing the anticipated effect size will increase statistical power but may represent an unrealistic expectation of the true difference between the treatments. Conversely, an unduly small d might be clinically insignificant, besides requiring a very large sample-size to achieve sufficient statistical power.

Statistical power depends on the magnitude of d divided its standard deviation $\sigma(d)$, i.e. it depends on $D = \frac{|d|}{\sigma(d)}$. The sign of d is relevant as it determines whether the project is worth pursuing at all (see TBA §11.8.4). The ratio is termed (Cohen, 1988) Cohen's D. When this signal-to-noise-ratio-like quantity is large, statistical power approaches 100%. Reader and case variability and data correlations determine $\sigma(d)$. No matter how small the anticipated d, as long as it is finite, then, using sufficiently large numbers of readers and cases $\sigma(d)$ can be made

sufficiently small to achieve near 100% statistical power. Of course, a very small effect-size may not be clinically significant. There is a difference between *statistical significance* and *clinical significance*. An effect-size in AUC units could be so small as to be clinically insignificant, but by employing a sufficiently large sample size one could achieve statistical significance.

What determines clinical significance? A small effect-size could be clinically significant if it applies to a large population, where the small improvement is amplified by the number of patients benefiting from the new treatment. In contrast, for an "orphan" disease, i.e., one with very low prevalence, an effect-size of 0.05 might not be enough to justify the additional cost. The improvement might have to be 0.1 before it is worth it for a new treatment to be brought to market. One hates to monetize life and death issues, but there is no getting away from it, as cost/benefit issues determine clinical significance. The arbiters of clinical significance are engineers, imaging scientists, clinicians, epidemiologists, insurance companies and those who set government health care policies. The engineers and imaging scientists determine whether the effect-size the clinicians would like is feasible from technical and scientific viewpoints. The clinician determines, based on incidence of disease and other considerations, e.g., altruistic, malpractice, cost of the new device and insurance reimbursement, what effect-size is justifiable. Cohen has suggested that d values of 0.2, 0.5, and 0.8 be considered small, medium, and large, respectively, but he has also argued against their indiscriminate usage. However, after a study is completed, clinicians often find that an effect-size that biostatisticians label as small may, in certain circumstances, be clinically significant and an effect-size that they label as large may in other circumstances be clinically insignificant. Clearly, this is a complex issue. Some suggestions on choosing a clinically significant effect size are made in (TBA §11.12).

Having developed a new imaging modality the R&D team wishes to compare it to the existing standard with the short-term goal of making a submission to the FDA to allow them to perform pre-market testing of the device. The long-term goal is to commercialize the device. Assume the R&D team has optimized the device based on physical measurements, (TBA Chapter 01), perhaps supplemented with anecdotal feedback from clinicians based on a few images. Needed at this point is a pilot study. A pilot study, conducted with a relatively small and practical sample size, is intended to provide estimates of different sources of variability and correlations. It also provides an initial estimate of the effect-size, termed the *observed effect-size*, d. Based on results from the pilot the sample-size tools described in this chapter permit estimation of the numbers of readers and cases that will reduce $\sigma(d)$ sufficiently to achieve the desired power for the larger "pivotal" study. [A distinction could be made in the notation between observed and anticipated effect sizes, but it will be clear from the context. Later, it will be shown how one can make an educated guess about the anticipated effect size from an observed effect size.]

This chapter is concerned with multiple-reader MRMC studies that follow the fully crossed factorial design meaning that each reader interprets a common case-set in all treatments. Since the resulting pairings (i.e., correlations) tend to decrease $\sigma(d)$ (since the variations occur in tandem, they tend to cancel out in the difference, see (TBA Chapter 09, Introduction), for Dr. Robert Wagner's sailboat analogy) it yields more statistical power compared to an unpaired design, and consequently this design is frequently used. Two sample-size estimation procedures for MRMC are the Hillis-Berbaum method (Hillis and Berbaum, 2004) and the Obuchowski-Rockette (Obuchowski, 1998) method. With recent work by Hillis, the two methods have been shown to be substantially equivalent.

This chapter will focus on the DBM approach. Since it is based on a standard ANOVA model, it is easier to extend the NH testing procedure described in Chapter 09 to the alternative hypothesis, which is relevant for sample size estimation. [TBA Online Appendix 11.A shows how to translate the DBM formulae to the OR method (Hillis et al., 2011).]

Given an effect-size, and choosing this wisely is the most difficult part of the process, the method described in this chapter uses pseudovalue variance components estimated by the DBM method to predict sample-sizes (i.e., different combinations of numbers of readers and cases) necessary to achieve a desired power.

8.3 Statistical Power

The concept of statistical power was introduced in [TBA Chapter 08] but is worth repeating. There are two possible decisions following a test of a null hypothesis (NH): reject or fail to reject the NH. Each decision is associated with a probability on an erroneous conclusion. If the NH is true and one rejects it, the probability of the ensuing Type-I error is denoted α. If the NH is false and one fails to reject it, the probability of the ensuing Type II- error is denoted β. Statistical power is the complement of β, i.e.,

$$Power = 1 - \beta \tag{8.1}$$

Typically, one aims for $\beta = 0.2$ or less, i.e., a statistical power of 80% or more. Like $\alpha = 0.05$, this is a *convention* and more nuanced cost-benefit considerations may cause the researcher to adopt a different value.

8.3.1 Observed vs. anticipated effect-size

Assuming no other similar studies have already been conducted with the treatments in question, the observed effect-size, although "merely an estimate", is the best information available at the end of the pilot study regarding the value of the true effect-size. From the two previous chapters one knows that the significance testing software will report not only the observed effect-size, but also a 95% confidence interval associate with it. It will be shown later how one can use this information to make an educated guess regarding the value of the anticipated effect-size.

8.3.2 Dependence of statistical power on estimates of model parameters

Examination of the expression for , Eqn. (11.5), shows that statistical power increases if:

- The numerator is large. This occurs if: (a) the anticipated effect-size d is large. Since effect-size enters as the *square*, TBA Eqn. (11.8), it is has a particularly strong effect; (b) If $J \times K$ is large. Both of these results should be obvious, as a large effect size and a large sample size should result in increased probability of rejecting the NH.
- The denominator is small. The first term in the denominator is $(\sigma_\epsilon^2 + \sigma_{\tau RC}^2)$. These two terms cannot be separated. This is the residual variability of the jackknife pseudovalues. It should make sense that the smaller the variability, the larger is the non-centrality parameter and the statistical power.
- The next term in the denominator is $K\sigma_{\tau R}^2$, the treatment-reader variance component multiplied by the total number of cases. The reader variance σ_R^2 has no effect on statistical power, because it has an equal effect on both treatments and cancels out in the difference. Instead, it is the treatment-reader variance σ_R^2 that contributes "noise" tending to confound the estimate of the effect-size.
- The variance components estimated by the ANOVA procedure are realizations of random variables and as such subject to noise (there actually exists a beast such as variance of a variance). The presence of the K term, usually large, can amplify the effect of noise in the estimate of σ_R^2, making the sample size estimation procedure less accurate.
- The final term in the denominator is $J\sigma_{\tau C}^2$. The variance σ_C^2 has no impact on statistical power, as it cancels out in the difference. The treatment-case variance component introduces "noise" into the estimate of the effect size, thereby decreasing power. Since it is multiplied by J, the number of readers, and typically $J << K$, the error amplification effect on accuracy of the sample size estimate is not as bad as with the treatment-reader variance component.
- Accuracy of sample size estimation, essentially estimating confidence intervals for statistical power, is addressed in (Chakraborty, 2010).

8.3.3 Formulae for random-reader random-case (RRRC) sample size estimation

8.3.4 Significance testing

8.3.5 p-value and confidence interval

8.3.6 Comparing DBM to Obuchowski and Rockette for single-reader multiple-treatments

Having performed a pilot study and planning to perform a pivotal study, sample size estimation follows the following procedure, which assumes that both reader and case are treated as random factors. Different formulae, described later, apply when either reader or case is treated as a fixed factor.

- Perform DBM analysis on the pilot data. This yields the observed effect size as well as estimates of all relevant variance components and mean squares appearing in TBA Eqn. (11.5) and Eqn. (11.7).
- This is the difficult but critical part: make an educated guess regarding the effect-size, d, that one is interested in "detecting" (i.e., hoping to reject the NH with probability $1-\beta$). The author prefers the term "anticipated" effect-size to "true" effect-size (the latter implies knowledge of the true difference between the modalities which, as noted earlier, would obviate the need for a pivotal study).
- Two scenarios are considered below. In the first scenario, the effect-size is assumed equal to that observed in the pilot study, i.e., $d = d_{obs}$.
- In the second, so-called "best-case" scenario, one assumes that the anticipate value of d is the observed value plus two-sigma of the confidence interval, in the correct direction, of course, i.e., $d = |d_{obs}| + 2\sigma$. Here σ is one-fourth the width of the 95% confidence interval for d_{obs}. Anticipating more than 2σ greater than the observed effect-size would be overly optimistic. The width of the CI implies that chances are less than 2.5% that the anticipated value is at or beyond the overly optimistic value. These points will become clearer when example datasets are analyzed below.
- Calculate statistical power using the distribution implied by Eqn. (11.4), to calculate the probability that a random value of the relevant F-statistic will exceed the critical value, as in §11.3.2.
- If power is below the desired or "target" power, one tries successively larger value of J and / or K until the target power is reached.

8.4 Formulae for fixed-reader random-case (FRRC) sample size estimation

It was shown in TBA §9.8.2 that for fixed-reader analysis the non-centrality parameter is defined by:

$$\Delta = \frac{JK\sigma_\tau^2}{\sigma_\epsilon^2 + \sigma_{\tau RC}^2 + J\sigma_{\tau C}^2} \tag{8.2}$$

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

$$F_{AH|R} \equiv \frac{MST}{MSTC} \sim F_{I-1,(I-1)(K-1),\Delta} \tag{8.3}$$

8.4.1 Formulae for random-reader fixed-case (RRFC) sample size estimation

It is shown in TBA §9.9 that for fixed-case analysis the non-centrality parameter is defined by:

$$\Delta = \frac{JK\sigma_\tau^2}{\sigma_\epsilon^2 + \sigma_{\tau RC}^2 + K\sigma_{\tau R}^2} \tag{8.4}$$

Under the AH, the test statistic is distributed as a non-central F-distribution as follows:

$$F_{AH|C} \equiv \frac{MST}{MSTR} \sim F_{I-1,(I-1)(J-1),\Delta} \tag{8.5}$$

8.4.2 Fixed-reader random-case (FRRC) analysis TBA

It is a realization of a random variable, so one has some leeway in the choice of anticipated effect size - more on this later. Here J^* and K^* refer to the number of readers and cases in the *pilot* study.

8.4.3 Random-reader fixed-case (RRFC) analysis

8.4.4 Single-treatment multiple-reader analysis

8.5 Discussion/Summary/2

FROC analysis

Chapter 9

Analyzing FROC data

9.1 TBA How much finished

10%

9.2 Introduction

Analyzing FROC data is, apart from a single difference, very similar to analyzing ROC data. *The crucial difference is the selection of an appropriate location-sensitive figure of merit.* The reason is that the DBMH and ORH methods are applicable to any scalar figure of merit. Any appropriate FROC figure of merit reduces the mark rating data for a single dataset (i.e., a single treatment, a single reader and a number of cases) to a single scalar figure of merit.

The author recommends usage of the weighted AFROC figure of merit, where the lesions should be equally weighted, the default, unless there are strong clinical reasons for assigning unequal weights.

The chapter starts with analysis of a sample FROC dataset, #4 in Online Chapter 24. Any analysis should start with visualization of the relevant operating characteristic. Extensive examples are given using `RJafroc` implemented functions. Suggestions are made on how to report the results of a study (the suggestions apply equally to ROC studies). A method called *crossed-treatment analysis*, applicable when one has two treatment factors and their levels are crossed and one wishes to draw conclusions regarding the effect of treatments after averaging over all levels of the treatments.

9.3 Example 1

The following is a listing of file "mainAnalyzewAFROC.R". It performs both wAFROC and inferred ROC analyses of the same dataset and the results are saved to tables similar in structure to the Excel output tables shown for DBMH analysis of ROC data in §9.10.2.

Empirical wAFROC-AUC and ROC-AUC for all combinations of treatments and readers, and reader-averaged AUCs for each treatment (Rdr. Avg.). The weighted AFROC results were obtained from worksheet FOMs in file FedwAfroc.xlsx. The highest rating AUC results were obtained from worksheet FOMs in file FedHrAuc.xlsx. The wAFROC-AUCs are smaller than the corresponding ROC-AUCs.

The datasets that come with this book are described in Online Chapter 24. Four of these are ROC datasets, one an LROC dataset and the rest (nine) are FROC datasets. For non-ROC datasets, the highest rating method was used to infer the corresponding ROC data. The datasets are identified in the code by strings contained in the string-array variable fileNames (line 7 - 8). Line 9 selects the dataset to be analyzed. In the example shown the "FED" dataset has been selected. It is a 5 treatment 4 radiologist FROC dataset1 acquired by Dr. Federica Zanca. Line 13 loads the dataset; this is done internal to the function loadDataFile(). Line 11 constructs the name of the wAFROC file

and line 12 does the same for the ROC datafile. Line 15 which "spills over" to line 16 without the need for a special continuation character, generates an output file by performing DBMH significance testing (method = "DBMH") using fom = "wAFROC", i.e., the wAFROC figure of merit – this is the critical change. If one changes this to fom = "HrAuc", lines 19 – 20, then inferred ROC analysis occurs. In either case the default analysis, i.e., option = "ALL" is used, i.e., random-reader random-case (RRRC), fixed-reader random-case (FRRC) and random-reader fixed-case (RRFC). Results are shown below for random-reader random-case only.

The results of wAFROC analysis are saved to FedwAfroc.xlsx and that of inferred ROC analysis are saved to FedHrAuc.xlsx. The output file names need to be explicitly stated as otherwise they would overwrite each other (as a time-saver, checks are made at lines 14 and 18 to determine if the analysis has already been performed, in which case it is skipped).

In the Excel data file the readers are named 1, 3, 4 and 5 – the software treats the reader names as labels. The author's guess is that for some reason complete data for reader 2 could not be obtained. The renumber = TRUE option has the effect of renumbering the readers 1 through 4. Without renumbering, the output would be aesthetically displeasing, but have no effect on the conclusions.

Figures of merit, empirical wAFROC-AUC and empirical ROC-AUC, and the corresponding reader averages for both analyses are summarized in Table 19.1. The weighted AFROC results were obtained by copy and paste operations from worksheet FOMs in file FedwAfroc.xlsx. The highest rating AUC results were obtained by similar operations from worksheet FOMs in Excel file FedHrAuc.xlsx. As expected, each wAFROC-AUC is smaller than the corresponding ROC-AUC.

Table 19.1: Empirical wAFROC-AUC and ROC-AUC for all combinations of treatments and readers, and reader-averaged AUCs for each treatment (Rdr. Avg.). The weighted AFROC results were obtained from worksheet FOMs in file FedwAfroc.xlsx. The highest rating AUC results were obtained from worksheet FOMs in file FedHrAuc.xlsx. The wAFROC-AUCs are smaller than the corresponding ROC-AUCs.

Table 19.2 shows results for RRRC analysis using the wAFROC-AUC FOM. The overall F-test of the null hypothesis that all treatments have the same reader-averaged FOM, rejected the NH: F(4, 36.8) = 7.8, p = 0.00012 . The numerator degree of freedom ndf is I - 1 = 4. Since the null hypothesis is that all treatments have the same FOM, this implies that at least one pairing of treatments yielded a significant FOM difference. The control for multiple testing is in the formulation of the null hypothesis and no further Bonferroni-like2 correction is needed. To determine which specific pairings are significantly different one examines the p-values (listed under Pr>t) in the "95% CI's FOMs, treatment difference" portion of the table. It shows that the following differences are significant at alpha = 0.05, namely "1 – 3", "1 – 5", "2 – 3", "2 – 5", "3 – 4" and "4 – 5"; these are indicated by asterisks. The values listed under the "95% CI's FOMs, each treatment" portion of the table show that treatment 4 yielded the highest FOM (0.769) followed closely by treatments 2 and 1, while treatment 5 had the least FOM (0.714), slightly worse than treatment 3. This explains why the p-value for the difference 4 - 5 is the smallest (0.00007) of all the listed p-values in the "95% CI's FOMs, each treatment" portion of the table. Each instance where the p-value for the individual treatment comparisons yields a significant p-value is accompanied by a 95% confidence interval that does not include zero. The two statements of significance, one in terms of a p-value and one in terms of a CI, are equivalent. When it comes to presenting results for treatment FOM differences, I prefer the 95% CI but some journals insist on a p-value, even when it is not significant. Note that two sequential tests are involved, an overall F-test of the NH that all treatments have the same performance and only if this yields a significant results is one justified in looking at the p-values of individual treatment pairings.

Table 19.2: wAFROC-AUC analysis: results of random-reader random-case (RRRC) analysis, in worksheet "RRRC"". [ddf = denominator degrees of freedom of F-distribution. df = degrees of freedom of t-distribution. Stderr = standard error. CI = confidence interval. * = Significantly different at alpha = 0.05.]

Table 19.3 shows corresponding results for the inferred ROC-AUC FOM. Again the null hypothesis was rejected: F(4, 16.8) = 3.46, p = 0.032. This means at least two treatments have significantly different FOMs. Looking down the table, one sees that the same 6 pairs (as compared to wAFROC analysis) are significantly different, 1 – 3, 1- 5, etc., as indicated by the asterisks. The last five rows of the table show that treatment 4 had the highest performance while treatment 5 had the lowest performance. At the 5% significance level, both methods yielded the same significant differences, but this is not always true. While it is incorrect to conclude from a single dataset that a smaller p-value is indicative of higher statistical power, simulation testing under controlled conditions has consistently shown higher statistical power for the wAFROC-AUC FOM3,4 as compared to the inferred ROC-AUC FOM.

Table 19.3: Inferred ROC-AUC analysis: results of random-reader random-case (RRRC) analysis, in worksheet "RRRC"". ddf = denominator degrees of freedom of F-distribution. df = degrees of freedom of t-distribution. Stderr = standard error. CI = confidence interval; * = Significantly different at alpha = 0.05.].

9.4 TBA Plotting wAFROC and ROC curves

It is important to display empirical wAFROC/ROC curves, not just for publication purposes, but to get a better feel for the data. Since treatments 4 and 5 showed the largest difference, the corresponding /ROC plots for them are displayed. The code is in file mainwAfrocRocPlots.R.

Sourcing this code yields Fig. 19.1. Plot (A), originating from lines 16 – 19, shows individual reader wAFROC plots for treatment 4 (solid lines) and treatment 5 (dashed lines). Running the software on one's computer best shows the color-coding. While difficult to see, examination of this plot shows that all readers performed better in treatment 4 than in treatment 5 (i.e., for each color the solid line is above the dashed line). Plot (B), originating from lines 21 – 25, shows reader-averaged wAFROC plots for treatments 4 (red line, upper curve) and 5 (blue line, lower curve). If one changes, for example, line 19 from print(plot1$wAFROCPlot$)$toprint$($plot1wAFROCPoints) the code will output the coordinates of the points describing the curve, which gives the user the option to copy and paste the operating points into alternative plotting software.

Lines 16 – 19 create plots for all specified treatment-reader combinations. The "trick" to creating reader-averaged curves, such as in (B) is defining two list variables, plotT and plotR, at lines 21 – 22, the first containing the treatments to be plotted, list(4,5), and the second, a list of equal length, containing the arrays of readers to be averaged over, list(c(1:4), c(1:4)). More examples can be found in the help page for PlotEmpiricaOperatingCharacteristics().

Meaningful operating points on the reader average curves cannot be defined. This is because ratings are treatment and reader specific labels, so one cannot for example, average bin counts over all readers to construct a table like ROC Table 4.1 or its AFROC counterpart, Table 13.3.

Instead, the following procedure is used internal to PlotEmpiricaOperatingCharacteristics(). The reader-averaged plot for a specified treatment is obtained by dividing the FPF range from 0 to 1 into finely spaced steps of 0.005. For each FPF value the wLLF values for that treatment are averaged over all readers, yielding the reader-averaged ordinate. Calculating confidence intervals on the reader-averaged curve is possible but cumbersome and unnecessary in my opinion. The relevant information, namely the 95% confidence interval on the difference in reader-averaged AUCs, is already contained in the program output, see Table 19.2, row labeled "4 – 5*". The difference is 0.05488 with a 95% confidence interval (0.03018, 0.07957).

Fig. 19.1: FED dataset; (A): individual reader wAFROC plots for treatments 4 and 5. While difficult to see, all readers performed better in treatment 4 as indicated by each colored solid line being above the corresponding dashed lines. (B): reader-averaged wAFROC plots for treatments 4 and 5. The performance superiority of treatment 4 is fairly obvious in this curve. The difference is significant, p = 0.00012.

Inferred ROC plots corresponding to Fig. 19.1 were generated by lines 20-24, i.e., by changing opChType = "wAFROC" to opChType = "ROC", and print(plot2$wAFROCPlot$)$toprint$($plot2ROCPlot), resulting in Fig. 19.2. From Table 19.3 it is seen that the difference in reader-averaged AUCs is 0.04219 with a 95% confidence interval (0.00727, 0.07711). The observed wAFROC effect-size, 0.05488, is larger than the corresponding inferred ROC effect-size, 0.04219. This is a common observation, but sampling variability compounded with small differences, could give different results.

Fig. 19.2: FED dataset; (A): individual reader ROC plots for treatments 4 and 5. While difficult to see, all readers performed better in treatment 4. (B): reader-averaged ROC plots for treatments 4 and 5. The performance superiority of treatment 4 is fairly obvious in this curve. The difference is significant, p = 0.03054.

9.5 Reporting an FROC study

The methods section should make it clear exactly how the study was conducted. The information should be enough to allow some one else to replicate the study. How many readers, how many cases, how many treatments were used. How was ground truth determined and if the FROC paradigm was used, how were true lesion locations determined?

The instructions to the readers should be clearly stated in writing. Precautions to minimize reading order effects should be stated – usually this is accomplished by interleaving cases from different treatments so that the chances that cases from a particular treatment is always seen first by every reader are minimized. Additionally, images from the same case, but in different treatments, should not be viewed in the same reading session. Reading sessions are usually an hour, and the different sessions should ideally be separated by at least one day. Users generally pay minimal attention to training sessions. It is recommended that at least 25% of the total number of interpretations be training cases and cases used for training should not be used in the main study. Feedback should be provided during training session to allow the reader to become familiar with the range of difficulty levels regarding diseased and non-diseased cases in the dataset. Deception, e.g., stating a higher prevalence than is actually used, is usually not a good idea. The user-interface should be explained carefully. The best user interface is intuitive, minimizes keystrokes and requires the least explanation.

In publications, the paradigm used to collect the data (ROC, FROC, etc.) and the figure of merit used for analysis should be stated. If FROC, the proximity criterion should be stated. The analysis should state the NH and the alpha of the test, and the desired generalization. The software used and appropriate references should be cited. The results of the overall F-test, the p-value, the observed F-statistic and its degrees of freedom should be stated. If the NH is not rejected, one should cite the observed inter-treatment FOM differences, confidence intervals and p-values and ideally provide preliminary sample size estimates. This information could be useful to other researchers attempting to conduct a larger study. If the NH is rejected, a table of inter-treatment FOM differences such as Table 19.3 should be summarized. Reader averaged plots of the relevant operating characteristics for each treatment should be provided. In FROC studies it is recommended to vary the proximity criterion, perhaps increasing it by a factor of 2, to test if the final conclusions (is NH rejected and if so which treatment is highest) are unaffected.

Assuming the study has been done properly and with sufficiently large number of cases, the results should be published in some form, even if the NH is not rejected. The dearth of datasets to allow reasonable sample size calculations is a real problem in this field. The dataset set should be made available, perhaps on Research Gate, or if communicated to me, they will be included in the Online Appendix material. Datasets acquired via NIH or other government funding must be made available upon request, in an easily decipherable format. Subsequent users of these datasets must cite the original source of the data. Given the high cost of publishing excess pages in some journals, an expanded version, if appropriate for clarity, should be made available using online posting avenues.

9.6 Crossed-treatment analysis

This analysis was developed for a particular application6 in which nodule detection in an anthropomorphic chest phantom in computed tomography (CT) was evaluated as a function of tube charge and reconstruction method. The phantom was scanned at 4 values of mAs and images were reconstructed with adaptive iterative dose reduction 3D (AIDR3D) and filtered back projection (FBP). Thus there are two treatment factors and the factors are crossed since for each value of the mAs factor there were two values of the reconstruction algorithm factor. Interest was in determining if whether performance depends on mAs and/or reconstruction method.

In a typical analysis of MRMC ROC or FROC study, treatment is considered as a single factor with I levels, where I is usually small. The figure of merit for treatment i (i =1,2,...,I) and reader j (j =1,2,...,J) is denoted ; the case set index is suppressed. MRMC analysis compares the observed magnitude of the difference in reader-averaged figures of merit between treatments i and i', , to the estimated standard deviation of the difference. For example, the reader-averaged difference in figures of merit is , where the dot symbol represents the average over the corresponding (reader) index. The standard deviation of the difference is estimated using the DBMH or the ORH method, using for example jackknifing to determine the variance components and/or covariances. With I levels, the number of distinct i vs. i' comparisons is I (I −1)/2. If the current study were analyzed in this manner, where I =8 (4 levels of mAs and two image reconstruction methods), then this would imply 28 comparisons. The large number of comparisons leads to loss of statistical power in detecting the effect of a specific pair of treatments, and, more importantly, does not inform one of the main points of interest: whether performance depends on mAs and/or reconstruction method. For example, in standard analysis the two reconstruction algorithms might be compared at different mAs levels, and one is in the dark as to which factor (algorithm or mAs) caused the observed significant difference.

Unlike conventional ROC type studies, the images in this study are defined by two factors. The first factor, tube charge, had four levels: 20, 40, 60 and 80 mAs. The second factor, reconstruction method, had two levels: FBP and

AIDR3D. The figure of merit is represented by , where represents the levels of the first factor (mAs), and represents the levels of the second factor (reconstruction method), . Two sequential analyses were performed: (i) mAs analysis, where the figure of merit was averaged over (the reconstruction index); and (ii) reconstruction analysis, where the figure of merit was averaged over (the mAs index). For example, the mAs analysis figure of merit is , where the dot represents the average over the reconstruction index, and the corresponding reconstruction analysis figure of merit is , where the dot represents the average over the mAs index. Thus in either analysis, the figure of merit is dependent on a single treatment factor, and therefore standard DBMH or ORH methods apply.

The mAs analysis determines whether tube charge is a significant factor and in this analysis the number of possible comparisons is only six. The reconstruction analysis determines whether AIDR3D offers any advantage over FBP and in this analysis the number of possible comparisons is only one. Multiple testing on the same dataset increases the probability of Type I error, therefore a Bonferroni correction is applied by setting the threshold for declaring significance at 0.025; this is expected to conservatively maintain the overall probability of a Type I error at $=$ 0.05. Crossed-treatment analysis is used to describe this type of analysis of ROC/FROC data, which yields clearer answers on which of the two factors effects performance. The averaging over the other treatment has the effect of increasing the power of the study in detecting differences in each of the two factors.

Since the phantom is unique, and conclusions are only possible that are specific to this one phantom, the case (or image) factor was regarded as fixed. For this reason only results of random-reader fixed-case analyses are reported.

9.7 Discussion / Summary

An IDL (Interactive Data Language, currently marketed by Exelis Visual Information Solutions, www.exelisvis.com) version of JAFROC was first posted to a now obsolete website on 4/16/2004. This software required a license for IDL, which most users did not have. Subsequently, (9/27/2005) a version was posted which allowed analysis using the freely downloadable IDL Virtual Machine software (a method for freely distributing compiled IDL code). On 1/11/2011 the standalone Windows-compatible version was posted (4.0) and the current version is 4.2. JAFROC is windows compatible (XP, Vista and Windows 7, 8 and10).

To our knowledge JAFROC is the only easily accessible software currently available that can analyze FROC data. Workstation software for acquiring ROC and FROC data is available from several sources7-9. The Windows version is no longer actively supported (bugs, if pointed out, will be corrected). Current effort to conduct research and distribute software uses the R platform10. There are several advantages to this. R is an open-source platform - we have already benefited from a bug pointed out by a user . R runs on practically any platform (Windows, OSX, Linux, etc.). Also, developing an R package benefits from other contributed R-packages, which allow easy computation of probability integrals, random number generation, and parallel computing to speed up computations, to name just a few. The drawback with R, and this has to with its open source philosophy, is that one cannot readily integrate existing ROC code, developed on other platforms and other programming languages (specifically, DLLs are not allowed in R). So useful programs like CORROC2 and CBM were coded in C++, since R allows C++ programs to be compiled and included in a package.

Due to the random number of marks per image, data entry in the FROC paradigm is inherently more complicated and error-prone than in ROC analysis, and consequently, and in response to feedback from users, much effort has gone into error checking. The users have especially liked the feature where the program indicates the Excel sheet name and line-number where an error is detected. User-feedback has also been very important in detecting program bugs and inconsistencies in the documentation and developing additional features (e.g., ROI analysis).

Interest in the FROC paradigm is evidenced by the fact that Ref. 3 describing the JAFROC method has been cited over 273 times. Over 25,000 unique visitors have viewed my website, at least 73 have downloaded the software and over 107 publications using JAFROC have appeared. The list is available on my website. JAFROC has been applied to magnetic resonance imaging, virtual computerized tomography colonoscopy, digital tomosynthesis (chest and breast), mammography dose and image processing optimization, computer aided detection (CAD), computerized tomography, and other applications.

Since confusion still appears to exist, especially among statisticians, regarding perceived neglect of intra-image correlations of ratings and how true negatives are handled in FROC analysis11, we close with a quote from respected sources 12 “(Chakraborty and Berbaum) have presented a solution to the FROC problem using a jackknife resampling approach that respects the correlation structure in the images … their paradigm successfully passes a rigorous

statistical validation test". Since 2005 the National Institutes for Health (NIH) has been generous with supporting the research and users of JAFROC have been equally generous with providing their datasets, which have resulted in several collaborations.

FROC sample size

Chapter 10

FROC sample size estimation

10.1 How much finished 99 percent

10.2 Overview

This chapter is split into two parts:

- Part 1 is a step-by-step (or first-principles) approach to FROC paradigm sample size estimation.
- Part 2 encapsulates some of the details in function `SsFrocNhRsmModel()` which makes it easier to use the sample size estimation method.

10.3 Part 1

10.3.1 Introduction

FROC sample size estimation is not fundamentally different from ROC sample size estimation detailed in Chapter 8 and summarized next.

Summary of ROC sample size estimation

Based on analysis of a pilot ROC dataset and using a specified figure of merit (FOM), e.g., `FOM = Wilcoxon`, and either `method = "DBM"` or `method = "OR"` for significance testing, one estimates the intrinsic variability of the data expressed in terms of FOM variance components. For the DBM method these are the pseudovalue-based variance components while for OR method these are the FOM-based variance and covariances. **In this chapter the OR method will be used.** The second step is to specify a clinically realistic effect-size, e.g., the anticipated AUC difference between the two modalities.

Given the variance components and the anticipated AUC difference the sample size functions (`RJafroc` function names beginning with `Ss`) allow one to estimate the number of readers and cases necessary to detect (i.e., reject the null hypothesis) the modality AUC difference with probability β, typically chosen to be 20 percent (corresponding to 80 percent statistical power) while maintaining the NH (zero AUC difference) rejection rate probability at α, typically chosen to be 5 percent.

Summary of FROC sample size estimation

In FROC analysis the only difference, indeed the critical difference, is the choice of FOM; e.g., `FOM = "wAFROC"` instead of the ROC-AUC, `FOM = "Wilcoxon"`. The FROC dataset is analyzed using the OR method. This yields the covariance matrix corresponding to wAFROC-AUC FOM. Next one specifies the effect-size **in wAFROC-AUC units** and this step requires care. The ROC-AUC has a historically well-known interpretation, namely it is the classification ability at separating diseased patients from non-diseased patients, while the wAFROC-AUC does

not. Needed is a way of relating the effect-size in the easily understood ROC-AUC unit to one in wAFROC-AUC unit. This requires a physical model, e.g., the RSM, that predicts both ROC and wAFROC curves and their corresponding AUCs.

1. One chooses an ROC-AUC effect-size that is realistic and one that clinicians understand and can therefore participate in the effect-size postulation process. Lacking such information I recommend, based on past ROC studies, 0.03 as typical of a small effect size and 0.05 as typical of a moderate effect size.
2. One converts the ROC effect-size to a wAFROC-AUC effect-size using the method described in the next section.
3. One uses the sample size tools in `RJafroc` to determine sample size for a desired statistical power.

   It is important to recognize is that all quantities have to be in the same units. When performing ROC analysis, everything (variance components and effect-size) has to be in units of the selected FOM, e.g., `FOM = "Wilcoxon"`. When performing wAFROC analysis, everything has to be in units of the wAFROC-AUC, i.e., `FOM = "wAFROC"`. The variance components and effect-size in wAFROC-AUC units will be different from their corresponding ROC counterparts. In particular, as shown next, a given ROC-AUC effect-size generally corresponds to a larger effect-size in wAFROC-AUC units. The reason for this is that the range over which wAFROC-AUC can vary, namely 0 to 1, which is twice the corresponding ROC-AUC range (0.5 to 1). For the same reason the wAFROC variance components also tend to be larger than the ROC variance components.

The next section explains the steps used to implement #2 above.

10.3.2 Relating ROC and wAFROC effect-sizes

The steps are illustrated using `dataset04`, a 5 treatment, 4 radiologist and 200 case FROC dataset (Zanca et al., 2009) acquired on a 5-point scale.

10.3.2.1 Extract NH treatments

If there are more than two treatments in the pilot dataset, as in `dataset04`, one extracts those treatments that represent "almost" null hypothesis data (in the sense of similar AUCs):

```
frocDataNH <- DfExtractDataset(dataset04, trts = c(1,2))
```

The preceding code extracts treatments 1 and 2 which were found (Zanca et al., 2009) to be "almost" equivalent (i.e., the NH could not be rejected for the wAFROC-AUC difference between these treatments). More than two almost NH treatments can be used if they have similar AUCs, as this will improve the stability of the procedure. However, the final sample size predictions are restricted to two treatments in the pivotal study.

The next two steps are needed since **the RSM fits binned ROC data**.

10.3.2.2 Convert the FROC NH data to ROC

If the original data is FROC one converts it to ROC:

```
rocDataNH <- DfFroc2Roc(frocDataNH)
```

10.3.2.3 Bin the data

If the NH dataset uses continuous ratings one bins the ratings:

```
# For dataset04 this is unnecessary as it is already binned, but it can't hurt
rocDataBinNH <- DfBinDataset(rocDataNH, opChType = "ROC")
```

The default number of bins should be used. Unlike binning using arbitrarily set thresholds the thresholds found by `DfBinDataset()` are unique as they are chosen to maximize the empirical ROC-AUC.

10.3.2.4 Determine the lesion distribution and weights of the FROC dataset

`lesDistr` is the lesion distribution, see Section 3.7.1 and line 1 of the following code. The RSM fitting algorithm needs to know how lesion-rich the dataset is as the predicted ROC-AUC depends on it. For this dataset fraction 0.69 of diseased cases have one lesion, fraction 0.2 have two lesions and fraction 0.11 have three lesions. One also needs the lesion weights matrix, **W**, see Section 3.8. The call at line 2 to `UtilLesWghtsDS` uses the default argument `relWeights = 0` which assigns equal weights to all lesions.

```
1 lesDistr <- UtilLesDistr(frocDataNH)
2 W <- UtilLesWghtsDS(frocDataNH)
```

Note that `lesDistr` and **W** are determined from the **FROC** NH dataset as this information is lost upon conversion to an ROC dataset.

10.3.2.5 Fit the RSM to the ROC data

For each treatment and reader the fitting algorithm `FitRsmRoc()` is applied (see below, lines 4 - 11) to the binned NH ROC dataset. The returned values are `mu`, `lambda` and `nu`, corresponding to the physical RSM parameters μ, λ and ν.

```
1  I <- dim(rocDataBinNH$ratings$NL)[1] # number of levels of treatment factor
2  J <- dim(rocDataBinNH$ratings$NL)[2] # number of levels of reader factor
3  RsmParmsNH <- array(dim = c(I,J,3)) # 3 corresponds to the three RSM parameters
4  for (i in 1:I) {
5    for (j in 1:J)  {
6      fit <- FitRsmRoc(rocDataBinNH, trt = i, rdr = j, lesDistr$Freq)
7      RsmParmsNH[i,j,1] <- fit[[1]] # mu
8      RsmParmsNH[i,j,2] <- fit[[2]] # lambda
9      RsmParmsNH[i,j,3] <- fit[[3]] # nu
10   }
11 }
```

10.3.2.6 Compute the median values of the RSM parameters

I recommend taking the median (not the mean) of the $I \times J$ values of each parameter as representing the average NH dataset (he median is less sensitive to outliers than the mean).

```
muNH <- median(RsmParmsNH[,,1])
lambdaNH <- median(RsmParmsNH[,,2])
nuNH <- median(RsmParmsNH[,,3])
```

The defining values of the RSM-based NH fitting model are `muNH` = 3.3121519, `lambdaNH` = 1.714368 and `nuNH` = 0.7036564.

10.3.2.7 Compute ROC and wAFROC NH AUCs

The next step is the compute the analytical (i.e., RSM-based) AUCs under the respective ROC and the wAFROC curves:

```
aucRocNH <- UtilAnalyticalAucsRSM(muNH, lambdaNH, nuNH, lesDistr = lesDistr$Freq)$aucROC
aucwAfrocNH <- UtilAnalyticalAucsRSM(muNH, lambdaNH, nuNH, lesDistr = lesDistr$Freq)$aucwAFROC
```

The AUCs are: `aucRocNH = 0.8791542` and `aucwAfrocNH = 0.7198615`. Note that the wAFROC-FOM is smaller than the ROC-FOM as it includes both detection and localization performance (the ROC-AUC only measures detection performance).

10.3.2.8 Compute ROC and wAFROC Alternative Hypotheses AUCs for a range of ROC-AUC effect sizes

To create the alternative hypothesis (AH) condition, one increments μ_{NH} by $\Delta\mu$. Tempting as it may be, it is not enough to simply increase the μ parameter as increasing μ will simultaneously decrease λ and increase ν (see the Astronomical Analogy in the RJafrocFrocBook). To account for this tandem effect one extracts the **intrinsic** parameters λ_i, ν_i, at line 7 of the following code, and then converts back to the physical parameters at line 9 using the incremented μ. Note the usage of the functions `Util2Intrinsic` (convert physical to intrinsic) and `Util2Physical` (convert intrinsic to physical) . The resulting ROC-AUC and wAFROC-AUC are then calculated. This yields the effect size (AH value minus NH value) using ROC and wAFROC FOMs for a series of specified $\Delta\mu$ values. These are used to relate the wAFROC effect size for a specified ROC effect size.

```
1  deltaMu <- seq(0.01, 0.2, 0.01)
2  esROC <- array(dim = length(deltaMu))
3  eswAFROC <- array(dim = length(deltaMu))
4
5  # get intrinsic parameters
6  par_i <- Util2Intrinsic(muNH, lambdaNH, nuNH) # convert physical to intrinsic
7
8  for (i in 1:length(deltaMu)) {
9
10   # find physical parameters for the increased muNH (accounting for the tandem effects)
11   par_p <- Util2Physical(muNH + deltaMu[i], par_i$lambda_i, par_i$nu_i)  # convert intrinsic to physical
12
13   # AH ROC value minus NH ROC value
14   esROC[i] <- UtilAnalyticalAucsRSM(
15     muNH + deltaMu[i], par_p$lambda, par_p$nu, lesDistr = lesDistr$Freq)$aucROC - aucRocNH
16
17   # AH wAFROC value minus NH wAFROC value
18   eswAFROC[i] <- UtilAnalyticalAucsRSM(
19     muNH + deltaMu[i], par_p$lambda, par_p$nu, lesDistr = lesDistr$Freq)$aucwAFROC - aucwAfrocNH
20
21 }
```

Here is a plot of wAFROC effect size (y-axis) vs. ROC effect size.

The plot is linear and the intercept is close to zero. This makes it easy to implement an interpolation function. In the following code line 1 fits `eswAFROC` vs. `esROC` using a linear model `lm()` function constrained to pass through the origin (the "-1"). One expects this constraint since for `deltaMu = 0` the effect size must be zero no matter how it is measured.

```
1  lmFit <- lm(eswAFROC ~ -1 + esROC) # the "-1" fits to straight line through the origin
2  scaleFactor <- lmFit$coefficients
3  effectSizeROC <- seq(0.01, 0.0525, 0.0025)
4  effectSizewAFROC <- effectSizeROC*scaleFactor
```

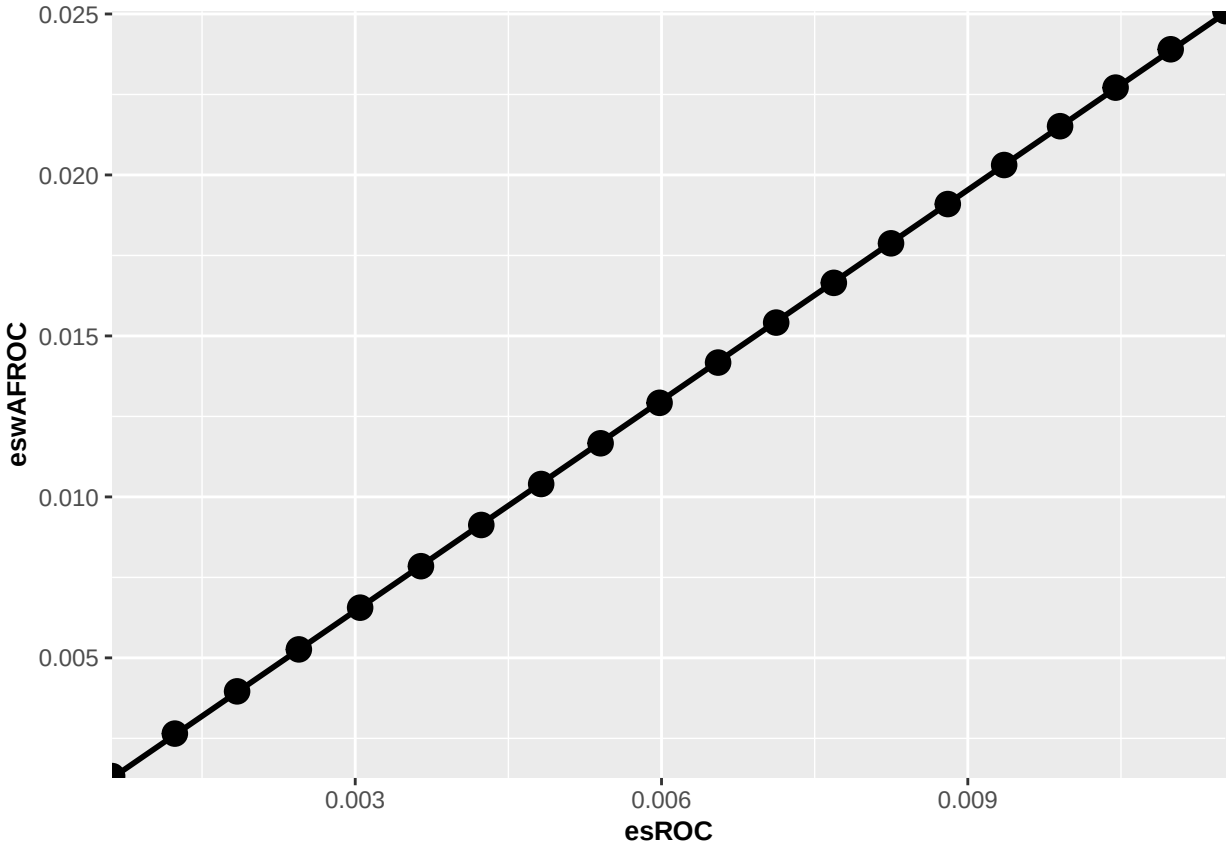


Figure 10.1: Plot of wAFROC effect size vs. ROC effect size. The straight line fit through the origin has slope 2.169.

The slope of the zero-intercept constrained straight line fit is `scaleFactor` = 2.169 and the squared correlation coefficient is `R2` = 0.9999904 (the fit is very good). Therefore, the conversion from ROC to wAFROC effect size is:

```
effectSizewAFROC = scaleFactor * effectSizeROC
```

For this dataset the wAFROC effect size is 2.169 times the ROC effect size. The wAFROC effect size is expected to be larger than the ROC effect size because the range of wAFROC-AUC, $1 - 0 = 1$, is twice that of ROC-AUC, $1 - 0.5 = 0.5$.

10.3.3 ROC and wAFROC variance components

The following skeleton code shows the arguments of the function `UtilORVarComp` used to calculate the OR variance components (other arguments are left at their default values).

```
UtilORVarComp(
  dataset,
  FOM
)
```

`UtilORVarComp()` is applied to `rocDataNH` and `frocDataNH` (using "Wilcoxon" and "wAFROC" FOMs as appropriate) followed by the extraction of the ROC and wAFROC variance components.

```
varComp_roc <- UtilORVarComp(
  rocDataNH,
  FOM = "Wilcoxon")$VarCom[-2]

varComp_wafroc <- UtilORVarComp(
  frocDataNH,
  FOM = "wAFROC")$VarCom[-2]
```

VarCom[-2] removes the second column of each dataframe containing the correlations. The ROC and wAFROC variance components are:

```
##                ROC        wAFROC
## VarR  0.0008277380  0.0018542289
## VarTR 0.0001526507 -0.0004439279
## Cov1  0.0002083377  0.0003736844
## Cov2  0.0002388384  0.0003567162
## Cov3  0.0001906167  0.0003058902
## Var   0.0007307912  0.0009081383
```

10.3.4 ROC and wAFROC power for equivalent effect-sizes

The following code compares ROC and wAFROC random-reader random-case (RRRC) powers for equivalent effect sizes.

First, one needs the numbers of readers `JStar` and cases `KStar` in the pilot dataset (lines 1 - 2 in following code) and those in the pivotal study, `JPivot` and `KPivot` (line 3). The values for the pivotal study have been arbitrarily set at 5 readers and 100 cases.

```
JStar <- length(dataset04$ratings$NL[1,,1,1])
KStar <- length(dataset04$ratings$NL[1,1,,1])
JPivot <- 5;KPivot <- 100
```

Next one extracts the OR ROC variance components from the previously computed list `varComp_roc`.

```
# these are OR variance components assuming FOM = "Wilcoxon"
varR_roc <- varComp_roc["VarR","Estimates"]
varTR_roc <- varComp_roc["VarTR","Estimates"]
Cov1_roc <- varComp_roc["Cov1","Estimates"]
Cov2_roc <- varComp_roc["Cov2","Estimates"]
Cov3_roc <- varComp_roc["Cov3","Estimates"]
Var_roc <- varComp_roc["Var","Estimates"]
```

The procedure is repeated for the OR wAFROC variance components using the previously computed list `varComp_wafroc`.

```
# these are OR variance components assuming FOM = "wAFROC"
varR_wafroc <- varComp_wafroc["VarR","Estimates"]
varTR_wafroc <- varComp_wafroc["VarTR","Estimates"]
Cov1_wafroc <- varComp_wafroc["Cov1","Estimates"]
Cov2_wafroc <- varComp_wafroc["Cov2","Estimates"]
Cov3_wafroc <- varComp_wafroc["Cov3","Estimates"]
Var_wafroc <- varComp_wafroc["Var","Estimates"]
```

We are now ready for the power calculations. The needed function is `SsPowerGivenJK` ("sample size for given number of readers J and cases K"):

```
SsPowerGivenJK(
  dataset,
  FOM,
  J,
  K,
  effectSize,
  ...
)
```

In the following code the OR ROC variance components are passed to `SsPowerGivenJK` at lines 12-18. The OR wAFROC variance components are passed to `SsPowerGivenJK` at lines 29-34. Setting `dataset = NULL` means that the function does not need a dataset as the variance components are supplied instead using the `...` argument.

The for-loop (lines 3 - 36) calculates ROC power (line 19) and wAFROC power (line 35) for a number of ROC (line 11) and corresponding wAFROC (line 28) effect sizes.

```
power_roc <- array(dim = length(effectSizeROC))
power_wafroc <- array(dim = length(effectSizeROC))
for (i in 1:length(effectSizeROC)) {
  # compute ROC power
  # dataset = NULL means use the supplied variance components instead of dataset
  ret <- SsPowerGivenJK(
    dataset = NULL,
    FOM = "Wilcoxon",
    J = JPivot,
    K = KPivot,
    effectSize = effectSizeROC[i],
    list(JStar = JStar,
         KStar = KStar,
         VarTR = varTR_roc,
         Cov1 = Cov1_roc,
         Cov2 = Cov2_roc,
```

```
        Cov3 = Cov3_roc,
        Var = Var_roc))
  power_roc[i] <- ret$powerRRRC

  # compute wAFROC power
  # dataset = NULL means use the supplied variance components instead of dataset
  ret <- SsPowerGivenJK(
    dataset = NULL,
    FOM = "wAFROC",
    J = JPivot,
    K = KPivot,
    effectSize = effectSizewAFROC[i],
    list(JStar = JStar,
        KStar = KStar,
        VarTR = varTR_wafroc,
        Cov1 = Cov1_wafroc,
        Cov2 = Cov2_wafroc,
        Cov3 = Cov3_wafroc,
        Var = Var_wafroc))
  power_wafroc[i] <- ret$powerRRRC
}
```

Since the wAFROC effect size is 2.1693379 times the ROC effect size, wAFROC power is larger than ROC power. For example, for ROC effect size = 0.035 the wAFROC effect size is 0.076, the ROC power is 0.234 while the wAFROC power is 0.797. The influence of the increased wAFROC effect size is magnified as it enters as the square in the formula for statistical power: this overwhelms the increase, noted previously, in variability of wAFROC-AUC relative to ROC-AUC

The following is a plot of wAFROC power vs. ROC power for the specified effect sizes.

10.4 Part 2

10.4.1 Introduction

This example uses the FED dataset as a pilot FROC study and function `SsFrocNhRsmModel()` (RSM-based FROC NH model) to construct the NH model (thereby encapsulating some of the code in the first part).

10.4.2 Constructing the NH model

The first two treatments are extracted from `dataset04` thereby yielding the NH dataset (line 1). The lesion distribution is specified in line 2. `lesDistr` can be specified independent of that in the pilot dataset. This allows some control over selection of the diseased cases in the pivotal study. However, in this example it is simply extracted from the pilot dataset. Line 3 constructs the NH model using function `SsFrocNhRsmModel` to calculate the NH RSM parameters (lines 4 - 6) and the scale factor (line 7).

```
frocNhData <- DfExtractDataset(dataset04, trts = c(1,2))
lesDistr <- UtilLesDistr(frocNhData) # this can be replaced by the anticipated lesion distribution
ret <- SsFrocNhRsmModel(frocNhData, lesDistr = lesDistr$Freq)
muNH <- ret$mu
lambdaNH <- ret$lambda
nuNH <- ret$nu
scaleFactor <- ret$scaleFactor
```

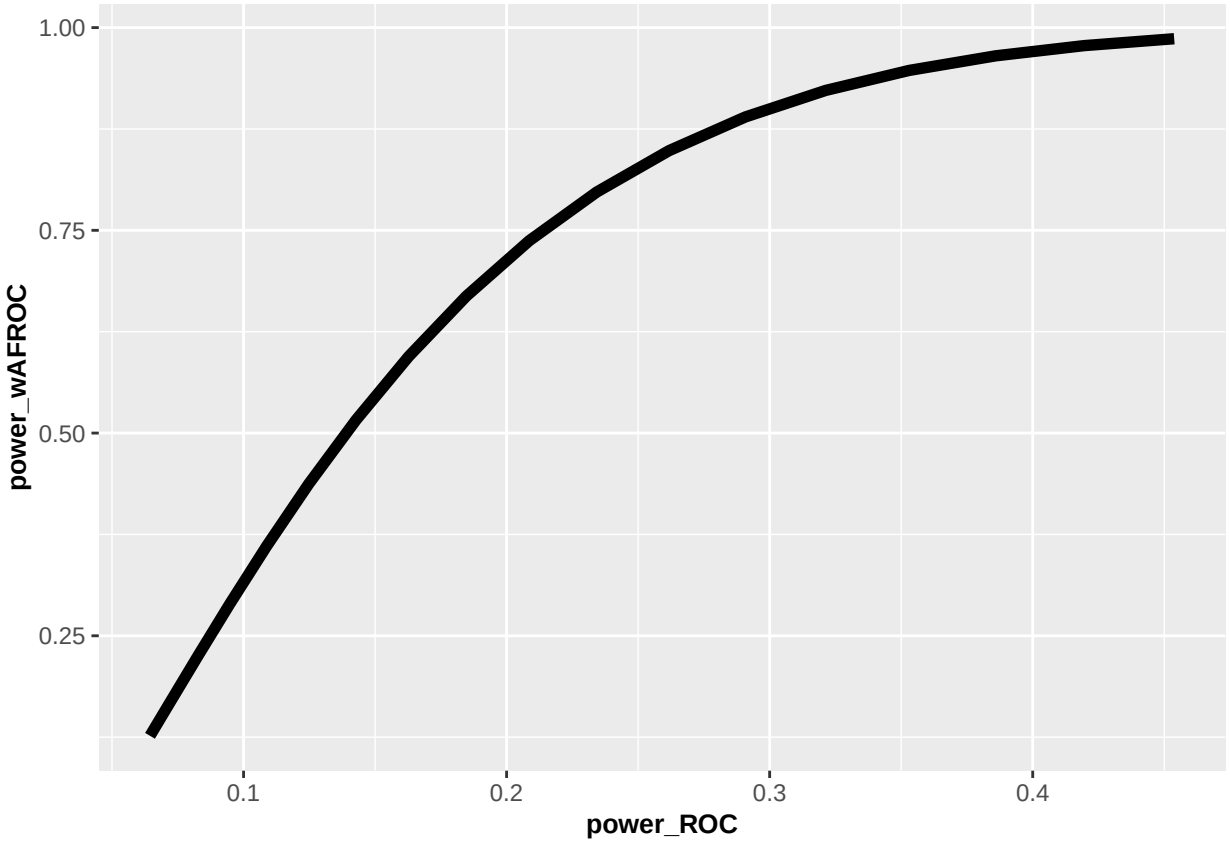


Figure 10.2: Plot of wAFROC power vs. ROC power. For ROC effect size = 0.035 the wAFROC effect size is 0.0759, the ROC power is 0.234 while the wAFROC power is 0.796.

The fitting model is defined by `muNH` = 3.3121519, `lambdaNH` = 1.714368 and `nuNH` = 0.7036564 and `lesDistr$Freq` = 0.69, 0.2, 0.11. The effect size scale factor is `scaleFactor` = 2.1693379. All of these are identical to the Part I values.

10.4.3 Extract the wAFROC variance components

The code applies the significance testing function `St()` to `frocNhData`, using `FOM = "wAFROC"` and extracts the variance components.

```
varComp_wafroc  <- St(
  frocNhData,
  FOM = "wAFROC",
  method = "OR",
  analysisOption = "RRRC")$ANOVA$VarCom
```

10.4.4 wAFROC power for specified ROC effect size, number of readers and number of cases

The following example is for ROC effect size = 0.035 (line 1), 5 readers and 100 cases (line 4) in the **pivotal study**. The function `SsPowerGivenJK` returns the power for the specified number of readers J (line 9), cases K (line 10) and wAFROC effect size (line 11). Since `dataset` is set to `NULL` JStar and KStar (corresponding to the pilot study) and the variance components are supplied as a `list` variable, lines 12 - 18. If dataset is specified then these are calculated from the pilot study dataset.

```
1  effectSizeROC <- 0.035
2  effectSizewAFROC <- scaleFactor * effectSizeROC
3
4  J <- 5;K <- 100 # define pivotal study sample size
5
6  ret <- SsPowerGivenJK(
7    dataset = NULL, # must set
8    FOM = "wAFROC",
9    J = J,
10   K = K,
11   effectSize = effectSizewAFROC,
12   list(JStar = JStar,
13        KStar = KStar,
14        VarTR = varTR_wafroc,
15        Cov1 = Cov1_wafroc,
16        Cov2 = Cov2_wafroc,
17        Cov3 = Cov3_wafroc,
18        Var = Var_wafroc))
19 power_wafroc <- ret$powerRRRC
```

```
## ROC-ES =  0.035 , wAFROC-ES =  0.07592683 , Power-wAFROC =  0.7972542
```

10.4.5 Number of cases for 80 percent power for a given number of readers

Function `SsSampleSizeKGivenJ` (number of cases K for desired power for given number of readers J) is shown below. If `dataset` is set to `NULL` then JStar and KStar and the variance components must be specified as a `list` variable, otherwise these are computed from `dataset`.

```
SsSampleSizeKGivenJ(
  dataset,
  ...,
  J,
  FOM,
  effectSize = NULL,
  alpha = 0.05,
  desiredPower = 0.8,
)
```

The following code returns the number of cases needed for 80 percent power for 6 readers (line 3), wAFROC effect size (line 4) = 0.076 and JStar and KStar and wAFROC variance components (lines 5 -11).

```
1  ret2 <- SsSampleSizeKGivenJ(
2    dataset = NULL,
3    J = 6,
4    effectSize = effectSizewAFROC,
5    list(JStar = JStar,
6         KStar = KStar,
7         VarTR = varTR_wafroc,
8         Cov1 = Cov1_wafroc,
9         Cov2 = Cov2_wafroc,
10        Cov3 = Cov3_wafroc,
11        Var = Var_wafroc))
```

```
## ROC-ES =  0.035 , wAFROC-ES =  0.07592683 , K80RRRC =  84 , Power-wAFROC =  0.8023882
```

Here `K80RRRC` is the number of cases needed for 80 percent power when using RRRC analysis.

Software details

Chapter 11

Excel file and dataset details

11.1 Introduction

This chapter is included to document recent Excel file format changes and the new dataset structure.

11.2 ROC dataset

```
x <- DfReadDataFile("R/quick-start/rocCr.xlsx", newExcelFileFormat = TRUE)
```

11.2.1 The structure of a factorial ROC dataset object

x is a `list` with 3 members: `ratings`, `lesions` and `descriptions`.

```
str(x, max.level = 1)
#> List of 3
#>  $ ratings     :List of 3
#>  $ lesions     :List of 3
#>  $ descriptions:List of 7
```

The `x$ratings` member contains 3 sub-lists.

```
str(x$ratings)
#> List of 3
#>  $ 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 ...
#>  $ LL_IL: logi NA
```

- `x$ratings$NL`, with dimension [2, 5, 8, 1], contains the ratings of normal cases. The first dimension (2) is the number of treatments, the second (5) is the number of readers and the third (8) is the total number of cases. For ROC datasets the fourth dimension is always unity. The five extra values [1] in the third dimension, of `x$ratings$NL` which are filled with `NAs`, are needed for compatibility with FROC datasets.
- `x$ratings$LL`, with dimension [2, 5, 5, 1], contains the ratings of abnormal cases. The third dimension (5) corresponds to the 5 diseased cases.

[1] With only 3 non-diseased cases why does one need 8 values?

- `x$ratings$LL_IL`, equal to NA', is there for compatibility with LROC data, `IL` denotes incorrect-localizations.

The `x$lesions` member contains 3 sub-lists.

```
str(x$lesions)
#> List of 3
#>  $ perCase: int [1:5] 1 1 1 1 1
#>  $ IDs    : num [1:5, 1] 1 1 1 1 1
#>  $ weights: num [1:5, 1] 1 1 1 1 1
```

- The `x$lesions$perCase` member is a vector with 5 ones representing the 5 diseased cases in the dataset.

- The `x$lesions$IDs` member is an array with 5 ones.

```
x$lesions$weights
#>      [,1]
#> [1,]    1
#> [2,]    1
#> [3,]    1
#> [4,]    1
#> [5,]    1
```

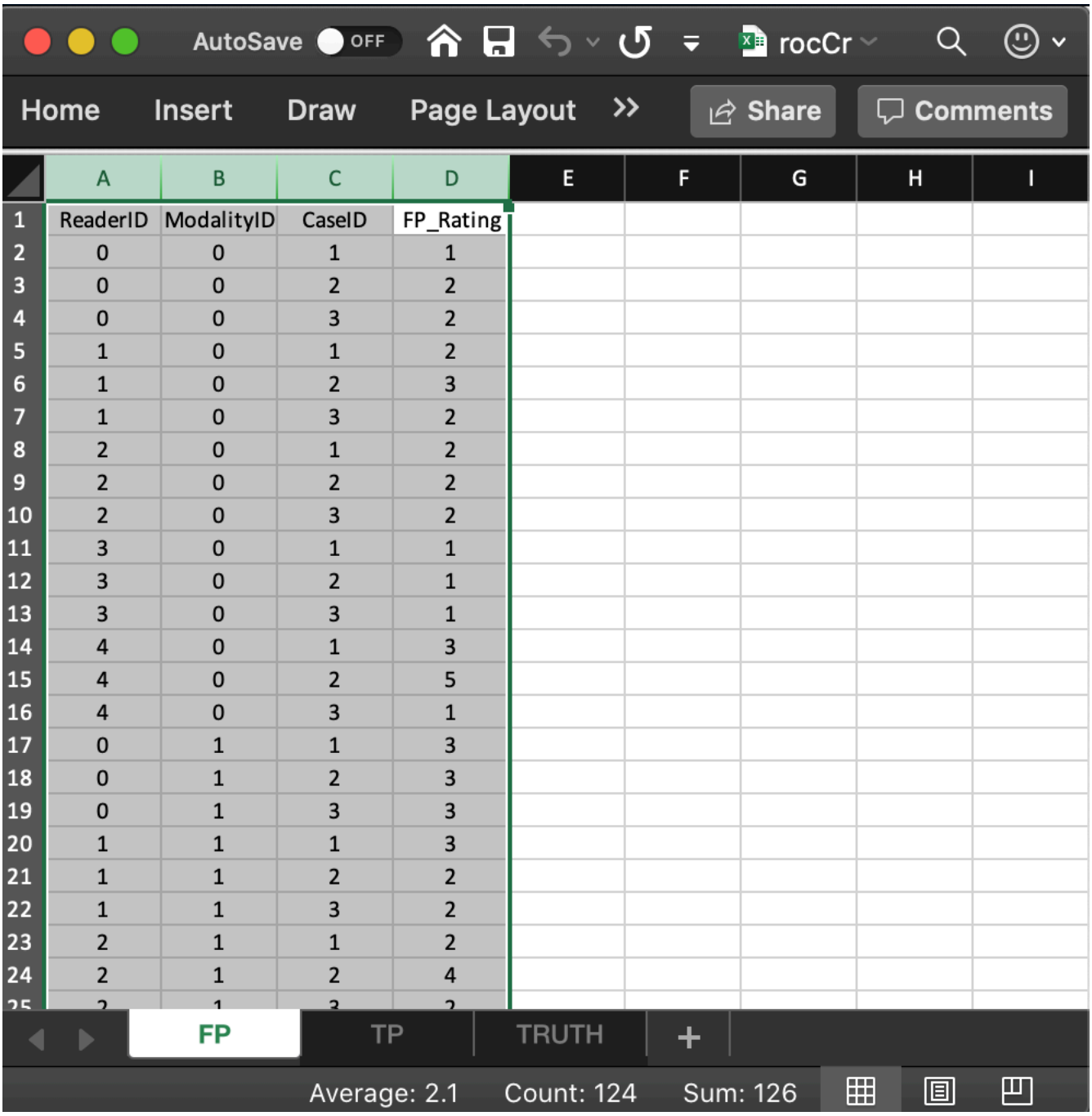
`x$lesions$weights` member is an array with 5 ones. These are irrelevant for ROC datasets. They are there for compatibility with FROC datasets.

`x$descriptions` contains 7 sub-lists.

```
str(x$descriptions)
#> List of 7
#>  $ fileName     : chr "rocCr"
#>  $ type         : chr "ROC"
#>  $ name         : logi NA
#>  $ truthTableStr: num [1:2, 1:5, 1:8, 1:2] 1 1 1 1 1 1 1 1 1 1 ...
#>  $ design       : chr "FCTRL"
#>  $ 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" ...
```

- `x$descriptions$fileName` is intended for internal use.
- `x$descriptions$type` indicates that this is an `ROC` dataset.
- `x$descriptions$name` is intended for internal use.
- `x$descriptions$truthTableStr` is intended for internal use, see Section 11.3.2.
- `x$descriptions$design` specifies the dataset design, which is "FCTRL" in the present example ("FCTRL" = a factorial dataset).
- `x$descriptions$modalityID` is a vector with two elements `"0"` and `"1"`, the names of the two modalities.
- `x$readerID` is a vector with five elements `"0"`, `"1"`, `"2"`, `"3"` and `"4"`, the names of the five readers.

11.2.2 The FP worksheet

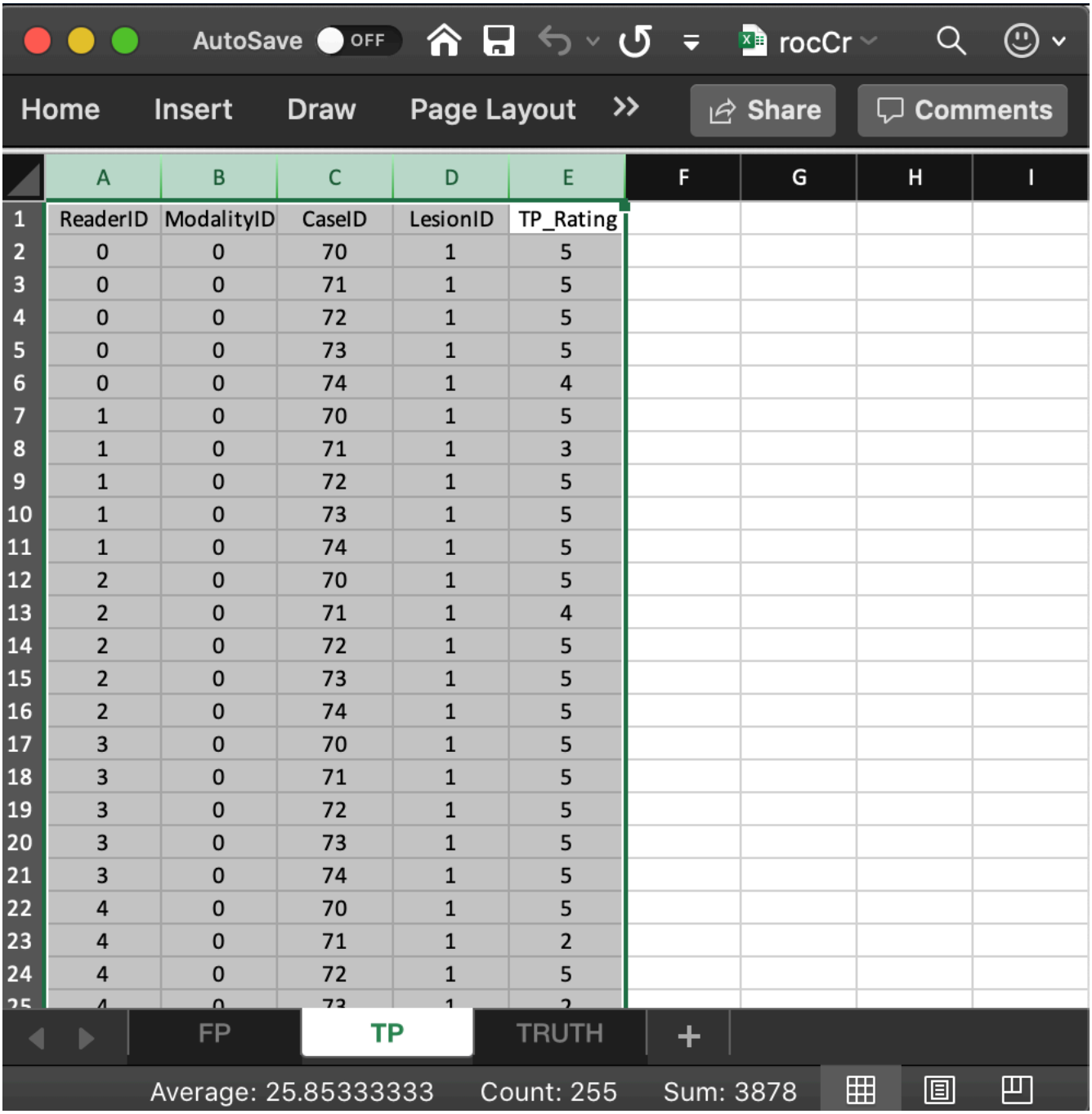


	A	B	C	D
1	ReaderID	ModalityID	CaseID	FP_Rating
2	0	0	1	1
3	0	0	2	2
4	0	0	3	2
5	1	0	1	2
6	1	0	2	3
7	1	0	3	2
8	2	0	1	2
9	2	0	2	2
10	2	0	3	2
11	3	0	1	1
12	3	0	2	1
13	3	0	3	1
14	4	0	1	3
15	4	0	2	5
16	4	0	3	1
17	0	1	1	3
18	0	1	2	3
19	0	1	3	3
20	1	1	1	3
21	1	1	2	2
22	1	1	3	2
23	2	1	1	2
24	2	1	2	4

- The list member `x$ratings$NL` is an array with `dim = c(2,5,8,1)`.
  - The first dimension (2) comes from the number of modalities.
  - The second dimension (5) comes from the number of readers.
  - The third dimension (8) comes from the **total** number of cases.
  - The fourth dimension is always 1 for an ROC dataset.
- The value of `x$ratings$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$ratings$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 and case index <= 8 are `-Inf`. For example the value of `x$ratings$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.

11.2.3 The `TP` worksheet



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

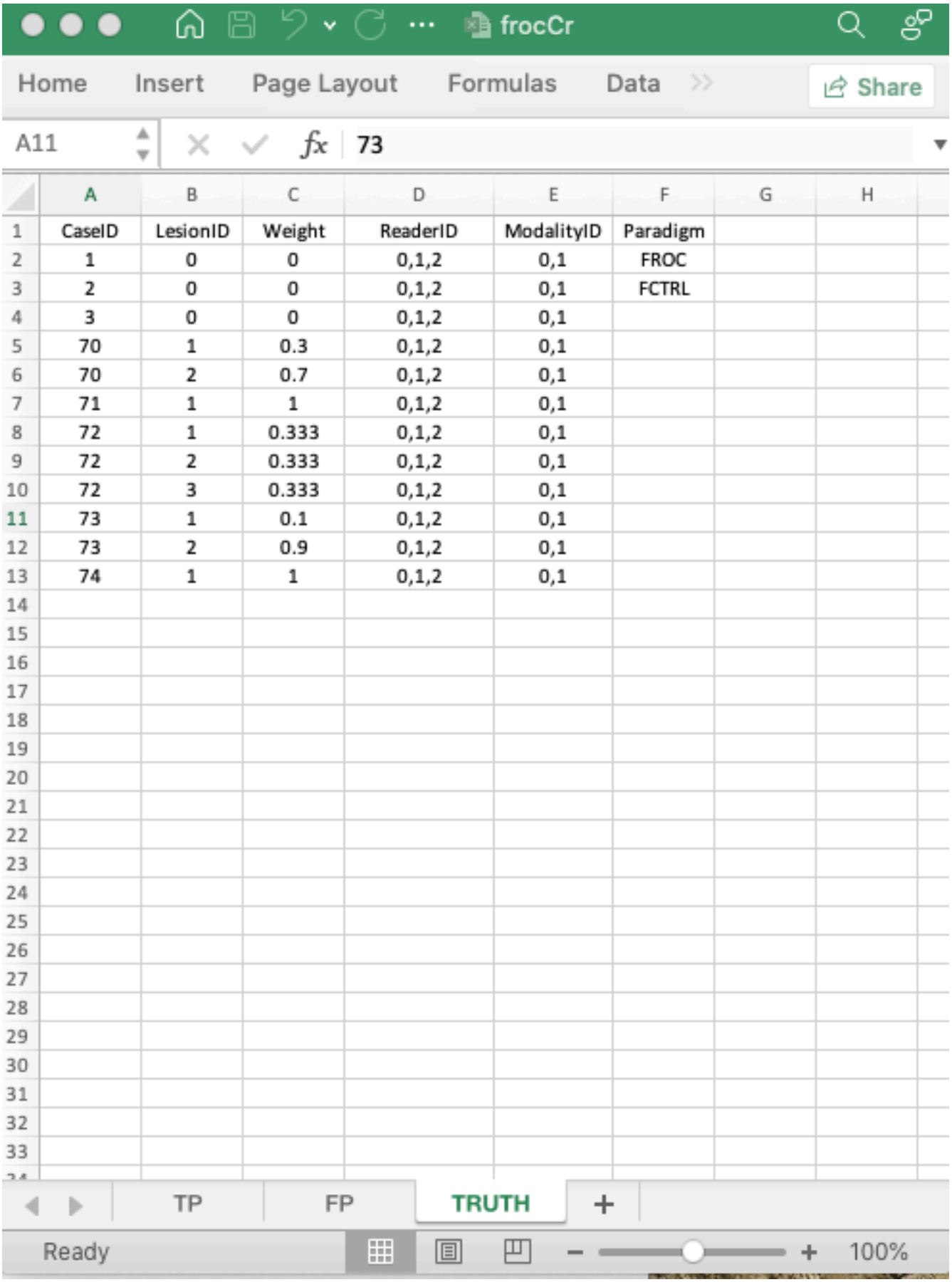
- The list member `x$ratings$LL` is an array with `dim = c(2,5,5,1)`.
  - The first dimension (2) comes from the number of modalities.
  - The second dimension (5) comes from the number of readers.
  - The third dimension (5) comes from the number of diseased cases.
  - The fourth dimension is always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

- The value of `x$ratings$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$ratings$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.
- The value of `x$ratings$LL[1,4,4,1]`, i.e., 5, corresponds to row 21 of the TP table, i.e., to `ModalityID` = 0, `ReaderID` = 3 and `CaseID` = 74.
- The value of `x$ratings$LL[1,5,2,1]`, i.e., 2, corresponds to row 23 of the TP table, i.e., to `ModalityID` = 0, `ReaderID` = 4 and `CaseID` = 71.
- There are no `-Inf` values in `x$ratings$LL`: `any(x$ratings$LL == -Inf)` = FALSE. This is true for any ROC dataset.

11.2.4 caseIndex vs. caseID

- The `caseIndex` is the array index used to access elements in the NL and LL arrays. The case-index is always an integer in the range 1, 2, ..., up to the array length. Remember that unlike C++, R indexing starts from 1.
- The `caseID` is any integer value, including zero, used to uniquely label the cases.
- Regardless of what order they occur in the worksheet, the non-diseased cases are always ordered first. In the current example the case indices are 1, 2 and 3, corresponding to the three non-diseased cases with `caseIDs` equal to 1, 2 and 3.
- Regardless of what order they occur in the worksheet, in the NL array the diseased cases are always ordered *after* the last non-diseased case. In the current example the case indices in the `NL` array are 4, 5, 6, 7 and 8, corresponding to the five diseased cases with `caseIDs` equal to 70, 71, 72, 73, and 74. In the `LL` array they are indexed 1, 2, 3, 4 and 5. Some examples follow:
- `x$ratings$NL[1,3,2,1]`, a FP rating, refers to `ModalityID` 0, `ReaderID` 2 and `CaseID` 2 (since the modality and reader IDs start with 0).
- `x$ratings$NL[2,5,4,1]`, a FP rating, refers to `ModalityID` 1, `ReaderID` 4 and `CaseID` 70, the first diseased case; this is `-Inf`.
- `x$ratings$NL[1,4,8,1]`, a FP rating, refers to `ModalityID` 0, `ReaderID` 3 and `CaseID` 74, the last diseased case; this is `-Inf`.
- `x$ratings$NL[1,3,9,1]`, a FP rating, is an illegal value, as the third index cannot exceed 8.
- `x$ratings$NL[1,3,8,2]`, a FP rating, is an illegal value, as the fourth index cannot exceed 1 for an ROC dataset.
- `x$ratings$LL[1,3,1,1]`, a TP rating, refers to `ModalityID` 0, `ReaderID` 2 and `CaseID` 70, the first diseased case.
- `x$ratings$LL[2,5,4,1]`, a TP rating, refers to `ModalityID` 1, `ReaderID` 4 and `CaseID` 73, the fourth diseased case.

11.3 FROC dataset



	A	B	C	D	E	F	G	H
1	CaseID	LesionID	Weight	ReaderID	ModalityID	Paradigm		
2	1	0	0	0,1,2	0,1	FROC		
3	2	0	0	0,1,2	0,1	FCTRL		
4	3	0	0	0,1,2	0,1			
5	70	1	0.3	0,1,2	0,1			
6	70	2	0.7	0,1,2	0,1			
7	71	1	1	0,1,2	0,1			
8	72	1	0.333	0,1,2	0,1			
9	72	2	0.333	0,1,2	0,1			
10	72	3	0.333	0,1,2	0,1			
11	73	1	0.1	0,1,2	0,1			
12	73	2	0.9	0,1,2	0,1			
13	74	1	1	0,1,2	0,1			
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								

11.3.1 The structure of a factorial FROC dataset

```
x <- DfReadDataFile("images/software-details/frocCr.xlsx", newExcelFileFormat = TRUE)
```

The dataset `x` is a `list` variable with 3 members: `x$ratings`, `x$lesions` and `x$descriptions`.

```
str(x, max.level = 1)
#> List of 3
#>  $ ratings     :List of 3
#>  $ lesions     :List of 3
#>  $ descriptions:List of 7
```

The `x$ratings` member contains 3 sub-lists.

```
str(x$ratings)
#> List of 3
#>  $ 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 ...
#>  $ LL_IL: logi NA
```

- There are `K2 = 5` diseased cases (the length of the third dimension of `x$ratings$LL`) and `K1 = 3` non-diseased cases (the length of the third dimension of `x$ratings$NL` minus `K2`).
- `x$ratings$NL`, a [2, 3, 8, 2] array, contains the NL ratings on non-diseased and diseased cases.
- `x$ratings$LL`, a [2, 3, 5, 3] array, contains the ratings of LLs on diseased cases.
- `x$ratings$LL_IL` is `NA`, this field applies to an LROC dataset (contains incorrect localizations on diseased cases).

The `x$lesions` member contains 3 sub-lists.

```
str(x$lesions)
#> List of 3
#>  $ perCase: int [1:5] 2 1 3 2 1
#>  $ IDs    : num [1:5, 1:3] 1 1 1 1 1 ...
#>  $ weights: num [1:5, 1:3] 0.3 1 0.333 0.1 1 ...
```

- `x$lesions$perCase` is the number of lesions per diseased case vector, i.e., 2, 1, 3, 2, 1.
- `max(x$lesions$perCase)` is the maximum number of lesions per case, i.e., `r`max(x*lesions*perCase)‘.
- `x$lesions$weights` is the weights of lesions.

```
x$lesions$weights
#>           [,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 weights for the first diseased case are 0.3 and 0.7. The weight for the second diseased case is 1. For the third diseased case the three weights are 1/3 each, etc. For each diseased case the finite weights sum to unity.

`x$descriptions` contains 7 sub-lists.

```
str(x$descriptions)
#> List of 7
#>  $ fileName     : chr "frocCr"
#>  $ type         : chr "FROC"
#>  $ name         : logi NA
#>  $ truthTableStr: num [1:2, 1:3, 1:8, 1:4] 1 1 1 1 1 1 1 1 1 1 ...
#>  $ design       : chr "FCTRL"
#>  $ 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"
```

- `x$descriptions$filename` is for internal use.
- `x$descriptions$type` is FROC, which specifies the data collection method.
- `x$descriptions$name` is for internal use.
- `x$descriptions$truthTableStr` is for internal use; it quantifies the structure of the dataset; it is explained in the next section.
- `x$descriptions$design` is FCTRL; it specifies the study design.
- `x$descriptions$modalityID` is a vector with two elements 0, 1 naming the two modalities.
- `x$readerID` is a vector with three elements 0, 1, 2 naming the three readers.

11.3.2 truthTableStr

- For this dataset `I` = 2, `J` = 3 and `K` = 8.
- `truthTableStr` is a `2 x 3 x 8 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 non-diseased cases).
- Each entry in this array is either `1`, meaning the corresponding interpretation happened, or `NA`, meaning the corresponding interpretation did not happen.

11.3.2.1 Explanation for non-diseased cases

Since the fourth index is set to 1, in the following code only non-diseased cases yield ones and all diseased cases yield `NA`.

```
all(x$descriptions$truthTableStr[,,1:3,1] ==1)
#> [1] TRUE
all(is.na(x$descriptions$truthTableStr[,,4:8,1]))
#> [1] TRUE
```

11.3.2.2 Explanation for diseased cases with one lesion

Since the fourth index is set to 2, in the following code all non-diseased cases yield `NA` and all diseased cases yield 1 as all diseased cases have at least one lesion.

```
all(is.na(x$descriptions$truthTableStr[,,1:3,2]))
#> [1] TRUE
all(x$descriptions$truthTableStr[,,4:8,2] == 1)
#> [1] TRUE
```

11.3.2.3 Explanation for diseased cases with two lesions

Since the fourth index is set to 3, in the following code all non-diseased cases yield `NA`; the first diseased case `70` yields 1 (this case contains two lesions); the second disease case `71` yields `NA` (this case contains only one lesion);

the third disease case 72 yields NA (this case contains only two lesions); the fourth disease case 73 yields 1 (this case contains two lesions); the fifth disease case 74 yields NA (this case contains one lesion).

```
# all non diseased cases
all(is.na(x$descriptions$truthTableStr[,,1:3,3]))
#> [1] TRUE
# first diseased case
all(x$descriptions$truthTableStr[,,4,3] == 1)
#> [1] TRUE
# second diseased case
all(is.na(x$descriptions$truthTableStr[,,5,3]))
#> [1] TRUE
# third diseased case
all(x$descriptions$truthTableStr[,,6,3] == 1)
#> [1] TRUE
# fourth diseased case
all(x$descriptions$truthTableStr[,,7,3] == 1)
#> [1] TRUE
# fifth diseased case
all(is.na(x$descriptions$truthTableStr[,,8,3]))
#> [1] TRUE
```

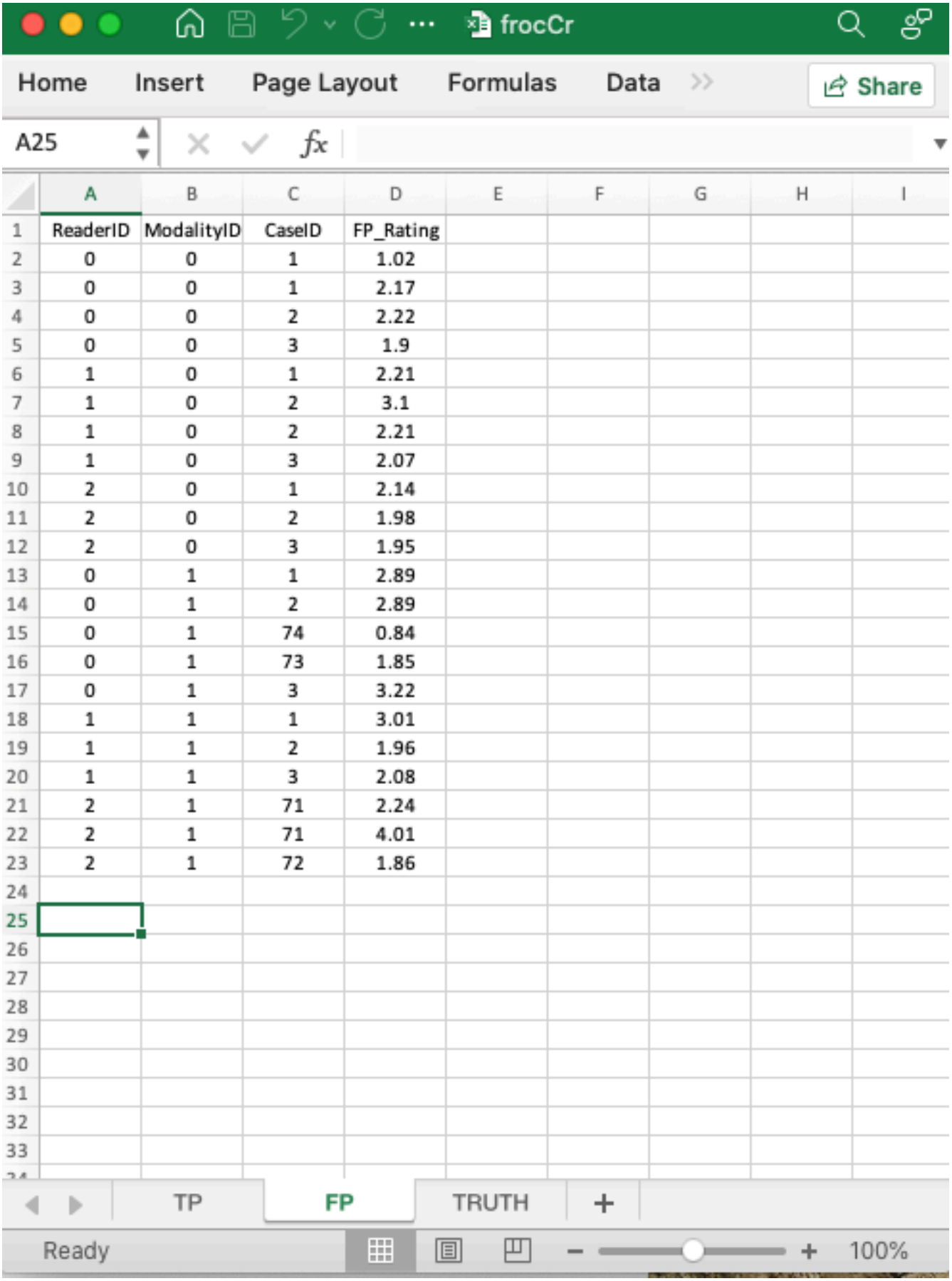
11.3.2.4 Explanation for diseased cases with three lesions

Since the fourth index is set to 4, in the following code all non-diseased cases yield NA; the first diseased case 70 yields NA (this case contains two lesions); the second disease case 71 yields NA (this case contains one lesion); the third disease case 72 yields NA (this case contains two lesions); the fourth disease case 73 yields 1 (this case contains three lesions); the fifth disease case 74 yields NA (this case contains one lesion).

```
# all non diseased cases
all(is.na(x$descriptions$truthTableStr[,,1:3,4]))
#> [1] TRUE
# first diseased case
all(is.na(x$descriptions$truthTableStr[,,4,4]))
#> [1] TRUE
# second diseased case
all(is.na(x$descriptions$truthTableStr[,,5,4]))
#> [1] TRUE
# third diseased case
all(x$descriptions$truthTableStr[,,6,4] == 1)
#> [1] TRUE
# fourth diseased case
all(is.na(x$descriptions$truthTableStr[,,7,4]))
#> [1] TRUE
# fifth diseased case
all(is.na(x$descriptions$truthTableStr[,,8,4]))
#> [1] TRUE
```

11.3.3 The FP worksheet

These are found in the FP or NL worksheet:



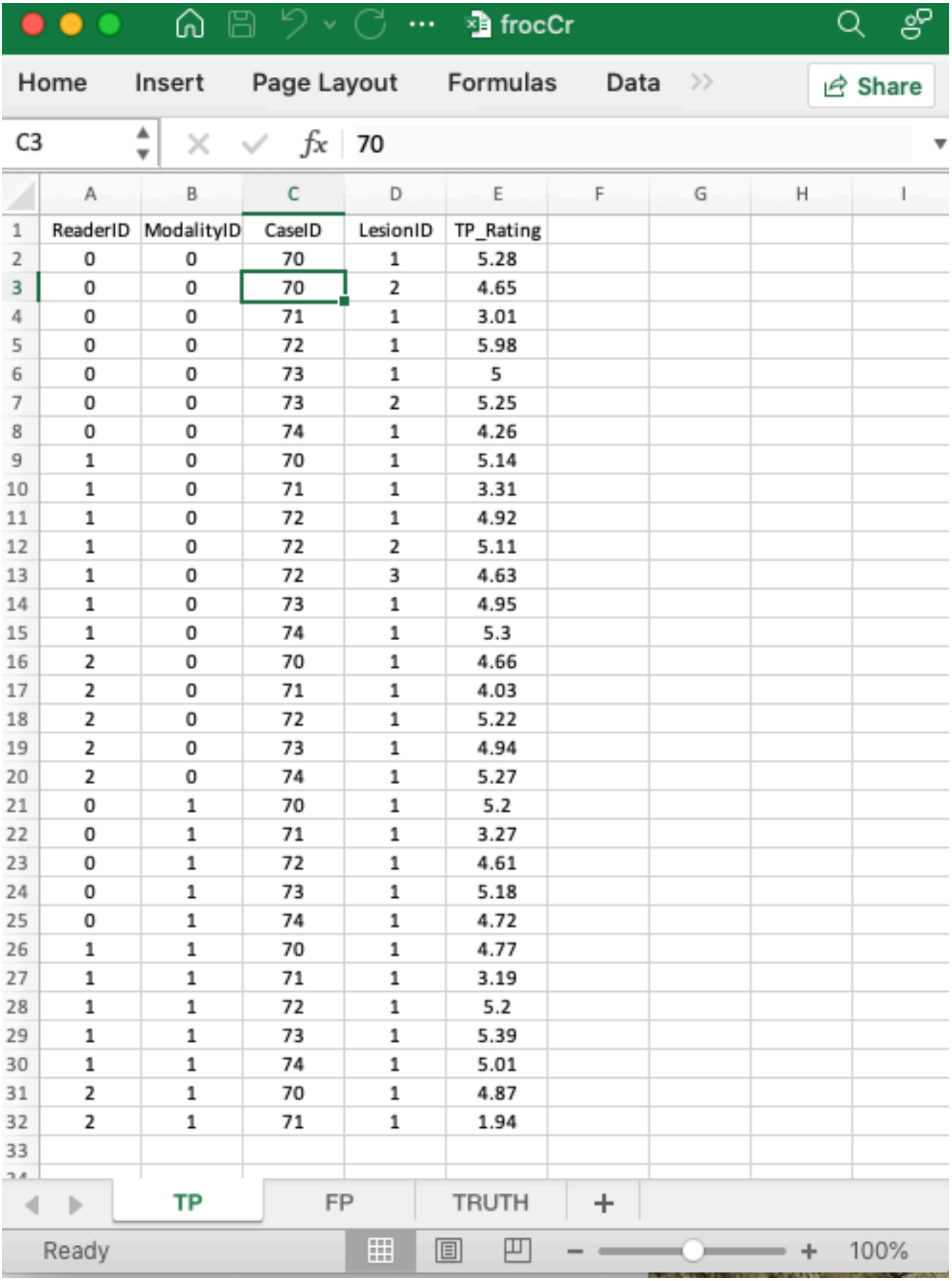
	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	FP_Rating					
2	0	0	1	1.02					
3	0	0	1	2.17					
4	0	0	2	2.22					
5	0	0	3	1.9					
6	1	0	1	2.21					
7	1	0	2	3.1					
8	1	0	2	2.21					
9	1	0	3	2.07					
10	2	0	1	2.14					
11	2	0	2	1.98					
12	2	0	3	1.95					
13	0	1	1	2.89					
14	0	1	2	2.89					
15	0	1	74	0.84					
16	0	1	73	1.85					
17	0	1	3	3.22					
18	1	1	1	3.01					
19	1	1	2	1.96					
20	1	1	3	2.08					
21	2	1	71	2.24					
22	2	1	71	4.01					
23	2	1	72	1.86					
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									

- The common vertical length is 22 in this example.
- `ReaderID`: the reader labels: `0`, `1,2, as declared in the`Truth‘ worksheet.
- `ModalityID`: the modality labels: `0` or `1`, as declared in the `Truth` worksheet.
- `CaseID`: `1`, `2`, `3`, `71`, `72`, `73`, `74`, as declared in the `Truth` worksheet; note that not all cases have NL marks on them.

- `NL_Rating`: the ratings of non-diseased cases.

11.3.4 The TP worksheet

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



frocCr

Home Insert Page Layout Formulas Data Share

C3 fx 70

	A	B	C	D	E	F	G	H	I
1	ReaderID	ModalityID	CaseID	LesionID	TP_Rating				
2	0	0	70	1	5.28				
3	0	0	70	2	4.65				
4	0	0	71	1	3.01				
5	0	0	72	1	5.98				
6	0	0	73	1	5				
7	0	0	73	2	5.25				
8	0	0	74	1	4.26				
9	1	0	70	1	5.14				
10	1	0	71	1	3.31				
11	1	0	72	1	4.92				
12	1	0	72	2	5.11				
13	1	0	72	3	4.63				
14	1	0	73	1	4.95				
15	1	0	74	1	5.3				
16	2	0	70	1	4.66				
17	2	0	71	1	4.03				
18	2	0	72	1	5.22				
19	2	0	73	1	4.94				
20	2	0	74	1	5.27				
21	0	1	70	1	5.2				
22	0	1	71	1	3.27				
23	0	1	72	1	4.61				
24	0	1	73	1	5.18				
25	0	1	74	1	4.72				
26	1	1	70	1	4.77				
27	1	1	71	1	3.19				
28	1	1	72	1	5.2				
29	1	1	73	1	5.39				
30	1	1	74	1	5.01				
31	2	1	70	1	4.87				
32	2	1	71	1	1.94				
33									

TP FP TRUTH +

Ready 100%

- This worksheet has the ratings of diseased cases.
- `ReaderID`: the reader labels: these must be from `0`, `1`, `2`, as declared in the `Truth` worksheet.
- `ModalityID`: `0` or `1`, as declared in the `Truth` worksheet.
- `CaseID`: these must be from `70`, `71`, `72`, `73`, `74`, as declared in the `Truth` worksheet; not all diseased cases have LL marks.

- `LL_Rating`: the ratings of diseased cases.

DATASETS

Chapter 12

Datasets

12.1 Datasets embedded in `RJafroc`

They are identified in the code by dataset`dd` (where `dd` is an integer in the range `01` to `14`). As an example, `dataset01` can be viewed here.

12.1.1 Dataset01

`dataset01` "TONY" FROC dataset (Chakraborty and Svahn, 2011)

```
## List of 3
##  $ NL   : num [1:2, 1:5, 1:185, 1:3] 3 -Inf 3 -Inf 4 ...
##  $ LL   : num [1:2, 1:5, 1:89, 1:2] 4 4 3 -Inf 3.5 ...
##  $ LL_IL: logi NA
```

12.1.2 Dataset02

`dataset02` "VAN-DYKE" (Van Dyke) ROC dataset (Van Dyke et al., 1993)

```
## List of 3
##  $ 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 ...
##  $ LL_IL: logi NA
```

12.1.3 Dataset03

`dataset03` "FRANKEN" (Franken) ROC dataset (Franken et al., 1992)

```
## List of 3
##  $ NL   : num [1:2, 1:4, 1:100, 1] 3 3 4 3 3 3 4 1 1 3 ...
##  $ LL   : num [1:2, 1:4, 1:67, 1] 5 5 4 4 5 4 4 5 2 2 ...
##  $ LL_IL: logi NA
```

12.1.4 Dataset04

`dataset04` "FEDERICA" (Federica Zanca) FROC dataset (Zanca et al., 2009)

```
## List of 3
##  $ 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 ...
##  $ LL_IL: logi NA
```

12.1.5 Dataset05

`dataset05` "THOMPSON" (John Thompson) FROC dataset (Thompson et al., 2014)

```
## List of 3
##  $ NL   : num [1:2, 1:9, 1:92, 1:7] 4 5 -Inf -Inf 8 ...
##  $ LL   : num [1:2, 1:9, 1:47, 1:3] 5 9 -Inf 10 8 ...
##  $ LL_IL: logi NA
```

12.1.6 Dataset06

- `dataset06` "MAGNUS" (Magnus Bath) FROC dataset (Vikgren et al., 2008)

```
## List of 3
##  $ NL   : num [1:2, 1:4, 1:89, 1:17] 1 -Inf -Inf -Inf 1 ...
##  $ LL   : num [1:2, 1:4, 1:42, 1:15] -Inf -Inf -Inf -Inf -Inf ...
##  $ LL_IL: logi NA
```

12.1.7 Dataset07

`dataset07` "LUCY-WARREN" (Lucy Warren) FROC dataset (Warren et al., 2014)

```
## List of 3
##  $ NL   : num [1:5, 1:7, 1:162, 1:4] 1 2 1 2 -Inf ...
##  $ LL   : num [1:5, 1:7, 1:81, 1:3] 2 -Inf 2 -Inf 1 ...
##  $ LL_IL: logi NA
```

12.1.8 Dataset08

`dataset08` "PENEDO" (Monica Penedo) FROC dataset (Penedo et al., 2005)

```
## List of 3
##  $ NL   : num [1:5, 1:5, 1:112, 1] 3 2 3 2 3 0 0 4 0 2 ...
##  $ LL   : num [1:5, 1:5, 1:64, 1] 3 2 4 3 3 3 3 4 4 3 ...
##  $ LL_IL: logi NA
```

12.1.9 Dataset09

`dataset09` "NICO-CAD-ROC" (Nico Karssemeijer) ROC dataset (Hupse et al., 2013)

```
## List of 3
##  $ NL   : num [1, 1:10, 1:200, 1] 28 0 14 0 16 0 31 0 0 0 ...
##  $ LL   : num [1, 1:10, 1:80, 1] 29 12 13 10 41 67 61 51 67 0 ...
##  $ LL_IL: logi NA
```

12.1.10 Dataset10

`dataset10` "RUSCHIN" (Mark Ruschin) ROC dataset (Ruschin et al., 2007)

```
## List of 3
##  $ NL   : num [1:3, 1:8, 1:90, 1] 1 0 0 0 0 0 1 0 0 0 ...
##  $ LL   : num [1:3, 1:8, 1:40, 1] 2 1 1 2 0 0 0 0 0 3 ...
##  $ LL_IL: logi NA
```

12.1.11 Dataset11

`dataset11` "DOBBINS-1" (James Dobbins) FROC dataset (Dobbins III et al., 2016)

```
## List of 3
##  $ 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 ...
##  $ LL_IL: logi NA
```

12.1.12 Dataset12

`dataset12` "DOBBINS-2" (James Dobbins) ROC dataset (Dobbins III et al., 2016)

```
## List of 3
##  $ NL   : num [1:4, 1:5, 1:152, 1] -Inf -Inf -Inf -Inf -Inf ...
##  $ LL   : num [1:4, 1:5, 1:88, 1] 3 4 4 -Inf -Inf ...
##  $ LL_IL: logi NA
```

12.1.13 Dataset13

`dataset13` "DOBBINS-3" (James Dobbins) FROC dataset (Dobbins III et al., 2016)

```
## List of 3
##  $ NL   : num [1:4, 1:5, 1:158, 1:4] -Inf 3 -Inf 4 5 ...
##  $ LL   : num [1:4, 1:5, 1:106, 1:15] -Inf -Inf -Inf -Inf -Inf ...
##  $ LL_IL: logi NA
```

12.1.14 Dataset14

`dataset14` "FEDERICA-REAL-ROC" (Federica Zanca) *real* ROC dataset (Zanca et al., 2012)

```
## List of 3
##  $ NL   : num [1:2, 1:4, 1:200, 1] 2 2 2 2 1 3 2 2 3 1 ...
##  $ LL   : num [1:2, 1:4, 1:100, 1] 6 5 6 4 5 5 5 5 5 4 ...
##  $ LL_IL: logi NA
```

12.2 Other datasets

12.2.1 DatasetCadLroc

`datasetCadLroc` "NICO-CAD-LROC" (Nico Karssemeijer) standalone CAD LROC dataset

```
## List of 3
##  $ NL   : num [1, 1:10, 1:200, 1] 28 0 14 0 16 0 31 0 0 0 ...
##  $ LL   : num [1, 1:10, 1:80, 1] 0 0 0 0 0 0 0 0 67 0 ...
##  $ LL_IL: num [1, 1:10, 1:80, 1] 29 12 13 10 41 67 61 51 0 0 ...
```

12.2.2 datasetCadSimuFroc

`datasetCadSimuFroc` "SIM-CAD-FROC" (Nico Karssemeijer) simulated standalone CAD FROC dataset

```
## List of 3
##  $ NL   : num [1, 1:10, 1:200, 1] 28 -Inf 14 -Inf 16 ...
##  $ LL   : num [1, 1:10, 1:80, 1] -Inf -Inf -Inf -Inf -Inf ...
##  $ LL_IL: logi NA
```

Bibliography

Bunch, P. C., Hamilton, J. F., Sanderson, G. K., and Simmons, A. H. (1977). A free response approach to the measurement and characterization of radiographic observer performance. In *Application of Optical Instrumentation in Medicine VI*, volume 127, pages 124–135. International Society for Optics and Photonics.

Chakraborty, D. and Svahn, T. (2011). Estimating the parameters of a model of visual search from roc data: an alternate method for fitting proper roc curves. In *Medical Imaging 2011: Image Perception, Observer Performance, and Technology Assessment*, volume 7966, pages 189–197. SPIE.

Chakraborty, D. and Zhai, X. (2022). *RJafroc: Artificial Intelligence Systems and Observer Performance*. R package version 2.1.1.9000.

Chakraborty, D. P. (2010). Prediction accuracy of a sample-size estimation method for ROC studies. *Academic radiology*, 17:628–638.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, 2 edition.

Dobbins III, J. T., McAdams, H. P., Sabol, J. M., Chakraborty, D. P., Kazerooni, E. A., Reddy, G. P., Vikgren, J., and Båth, M. (2016). Multi-institutional evaluation of digital tomosynthesis, dual-energy radiography, and conventional chest radiography for the detection and management of pulmonary nodules. *Radiology*, 282(1):236–250.

Franken, Edmund A., J., Berbaum, K. S., Marley, S. M., Smith, W. L., Sato, Y., Kao, S. C. S., and Milam, S. G. (1992). Evaluation of a digital workstation for interpreting neonatal examinations: A receiver operating characteristic study. *Investigative Radiology*, 27(9):732–737.

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

Hillis, S. L., Obuchowski, N. A., and Berbaum, K. S. (2011). Power estimation for multireader ROC methods: An updated and unified approach. *Academic Radiology*, 18(2):129–142.

Hupse, R., Samulski, M., Lobbes, M., Heeten, A., Imhof-Tas, M., Beijerinck, D., Pijnappel, R., Boetes, C., and Karssemeijer, N. (2013). Standalone computer-aided detection compared to radiologists' performance for the detection of mammographic masses. *European Radiology*, 23(1):93–100.

Obuchowski, N. A. (1998). Sample size calculations in studies of test accuracy. *Statistical Methods in Medical Research*, 7(4):371–392.

Penedo, M., Souto, M., Tahoces, P. G., Carreira, J. M., Villalon, J., Porto, G., Seoane, C., Vidal, J. J., Berbaum, K. S., Chakraborty, D. P., and Fajardo, L. L. (2005). Free-response receiver operating characteristic evaluation of lossy jpeg2000 and object-based set partitioning in hierarchical trees compression of digitized mammograms. *Radiology*, 237(2):450–457.

Ruschin, M., Timberg, P., Bath, M., Hemdal, B., Svahn, T., Saunders, R., Samei, E., Andersson, I., Mattsson, S., Chakraborty, D. P., and Tingberg, A. (2007). Dose dependence of mass and microcalcification detection in digital mammography: free response human observer studies. *Medical Physics*, 34:400 – 407.

Starr, S., Metz, C., Lusted, L., Sharp, P., and Herath, K. (1977). Comments on the generalization of receiver operating characteristic analysis to detection and localization tasks. *Physics in Medicine & Biology*, 22(2):376.

Starr, S. J., Metz, C. E., Lusted, L. B., and Goodenough, D. J. (1975). Visual detection and localization of radiographic images. *Radiology*, 116(3):533–538.

Swensson, R. G. (1996). Unified measurement of observer performance in detecting and localizing target objects on images. *Medical physics*, 23(10):1709–1725.

Thompson, J. D., Hogg, P., Manning, D. J., Szczepura, K., and Chakraborty, D. P. (2014). A free-response evaluation determining value in the computed tomography attenuation correction image for revealing pulmonary incidental findings: a phantom study. *Academic radiology*, 21(4):538–545.

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*.

Vikgren, J., Zachrisson, S., Svalkvist, A., Johnsson, A. A., Boijsen, M., Flinck, A., Kheddache, S., and Bath, M. (2008). Comparison of chest tomosynthesis and chest radiography for detection of pulmonary nodules: Human observer study of clinical cases. *Radiology*, 249(3):1034–1041.

Warren, L. M., Given-Wilson, R. M., Wallis, M. G., Cooke, J., Halling-Brown, M. D., Mackenzie, A., Chakraborty, D. P., Bosmans, H., Dance, D. R., and Young, K. C. (2014). The effect of image processing on the detection of cancers in digital mammography. *American Journal of Roentgenology*, 203(2):387–393.

Zanca, F., Hillis, S. L., Claus, F., Van Ongeval, C., Celis, V., Provoost, V., Yoon, H.-J., and Bosmans, H. (2012). Correlation of free-response and receiver-operating-characteristic area-under-the-curve estimates: Results from independently conducted froc/roc studies in mammography. *Medical physics*, 39(10):5917–5929.

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.