

# Analyse d'un tableau 2 x 2

Eléments de statistiques

*JcB*

*30/01/2016*

## 1 Calcul des valeurs caractéristiques

### 1.1 Création d'une table 2x2

A partir de 2 vecteurs de variables nominales dichotomiques. Chez 10 sujets on mesure 2 variables:

- age > 50 ans: oui = 1, non = 0
- présence de métastases: oui = 1, non = 0

```
v1 <- c(1,1,1,0,0,1,0,1,0,1)
v2 <- c(1,1,0,0,0,1,1,1,0,1)
```

Création d'un tableau de contingence 2x2 avec la routine `twotwo` qui est 4 à 5 fois plus rapide que la fonction `table`.

```
twotab <- twoXtwo(v1,v2, c("Age > 50", "Présence de métas"))
twotab
```

```
##           Présence de métas
## Age > 50 FALSE TRUE
##   FALSE      3     1
##   TRUE       1     5
```

#### 1.1.1 Tableau exprimé en pourcentages

```
ptab = prop.table(twotab)
round(ptab, 2)
```

```
##           Présence de métas
## Age > 50 FALSE TRUE
##   FALSE    0.3  0.1
##   TRUE     0.1  0.5
```

Toutes les cases sont divisées par l'effectif total.

#### 1.1.2 Tableau exprimé en pourcentage des lignes

```
rowtab = prop.table(twotab, margin=1)
round(rowtab, 2)
```

```
##           Présence de métas
## Age > 50 FALSE TRUE
##    FALSE  0.75 0.25
##    TRUE   0.17 0.83
```

Toutes les cases sont divisées par l'effectif de la ligne

### 1.1.3 Tableau exprimé en pourcentage des colonnes

```
coltab = prop.table(twotab, margin=2)
round(coltab, 2)
```

```
##           Présence de métas
## Age > 50 FALSE TRUE
##    FALSE  0.75 0.17
##    TRUE   0.25 0.83
```

Toutes les cases sont divisées par l'effectif de la colonne

### 1.1.4 Précision

```
accur = sum(diag(ptab))
accur
```

```
## [1] 0.8
```

### 1.1.5 Sensibilité

```
sens = rowtab["TRUE", "TRUE"]
sens
```

```
## [1] 0.8333333
```

### 1.1.6 Spécificité

```
spec = rowtab["FALSE", "FALSE"]
spec
```

```
## [1] 0.75
```

### 1.1.7 Valeur prédictive positive

```
ppv = coltab["TRUE", "TRUE"]
ppv
```

```
## [1] 0.8333333
```

### 1.1.8 Valeur prédictive négative

```
npv = coltab["FALSE", "FALSE"]
npv
```

```
## [1] 0.75
```

### 1.1.9 Chi2

```
chisq.test(twotab)
```

```
## Warning in chisq.test(twotab): Chi-squared approximation may be incorrect
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: twotab
## X-squared = 1.4062, df = 1, p-value = 0.2357
```

### 1.1.10 Fisher exact

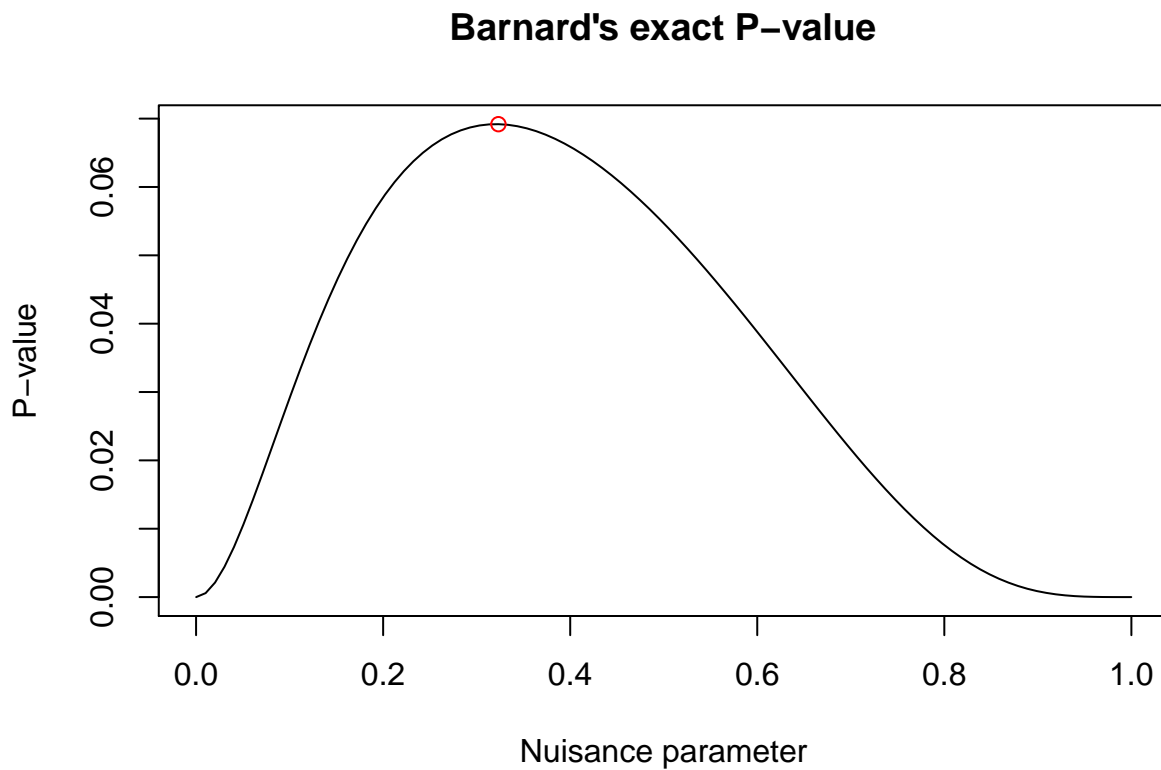
```
fisher.test(twotab)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: twotab
## p-value = 0.1905
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 0.40161 930.24692
## sample estimates:
## odds ratio
## 10.29391
```

### 1.1.11 Test de Barnard

```
Barnard(twotab)
```

```
##  
## 2x2 matrix Barnard's exact test: 100 5x7 tables were evaluated  
## -----  
## Wald statistic = 1.8447  
## Nuisance parameter = 0.32327  
## p-values: 1-tailed = 0.069185 2-tailed = 0.13837  
## -----  
##
```

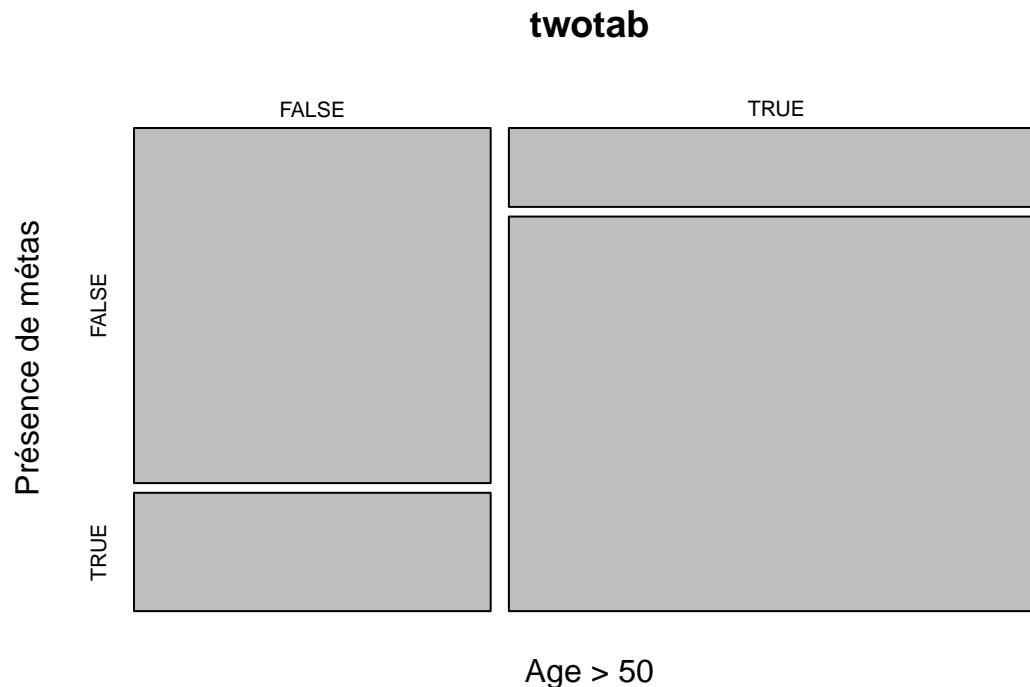


#### 1.1.12 Odds ratio

```
# oddsratioWald.proc(3,1,1,5)  
oddsratioWald.proc(twotab["FALSE", "FALSE"], twotab["FALSE", "TRUE"], twotab["TRUE", "FALSE"], twotab["TRUE", "TRUE"])  
  
##      LowerCI OR  UpperCI alpha  
## 1 0.6626447 15 339.5485 0.05
```

#### 1.1.13 Aspect graphique

```
mosaicplot(twotab)
```



## 2 Compléments

[Plus de 2 modalités](#)

### 2.1 About Barnard's exact test

[Source](#)

About half a year ago, I was studying various statistical methods to employ on contingency tables. I came across a promising method for  $2 \times 2$  contingency tables called “Barnard's exact test“. Barnard's test is a non-parametric alternative to [Fisher's exact test](#) which can be more powerful (for  $2 \times 2$  tables) but is also more time-consuming to compute (References can be found in the [Wikipedia article](#) on the subject).

The test was first published by George [Alfred Barnard](#) (1945) ([link to the original paper in Nature](#)). [Mehta and Senchaudhuri \(2003\)](#) explain why Barnard's test can be more powerful than Fisher's under certain conditions:

*When comparing Fisher's and Barnard's exact tests, the loss of power due to the greater discreteness of the Fisher statistic is somewhat offset by the requirement that Barnard's exact test must maximize over all possible  $p$ -values, by choice of the nuisance parameter, . For  $2 \times 2$  tables the loss of power due to the discreteness dominates over the loss of power due to the maximization, resulting in greater power for Barnard's exact test. But as the number of rows and columns of the observed table increase, the maximizing factor will tend to dominate, and Fisher's exact test will achieve greater power than Barnard's.*

#### 2.1.1 About the R implementation of Barnard's exact test

After finding about Barnard's test I was sad to discover that (at the time) there had been no R implementation of it. But last week, I received a surprising e-mail with good news. The sender, Peter Calhoun, currently a graduate student at the University of Florida, had implemented the algorithm in R. Peter had found my posting on the R mailing list (from almost half a year ago) and was so kind as to share with me (and the

rest of the R community) his R code for computing Barnard's exact test. Here is some of what Peter wrote to me about his code:

On a side note, I believe there are more efficient codes than this one. For example, I've seen codes in Matlab that run faster and display nicer-looking graphs. However, this code will still provide accurate results and a plot that gives the p-value based on the nuisance parameter. I did not come up with the idea of this code, I simply translated Matlab code into R, occasionally using different methods to get the same result. The code was translated from:

Trujillo-Ortiz, A., R. Hernandez-Walls, A. Castro-Perez, L. Rodriguez-Cardozo. Probability Test. A MATLAB file. URL

<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6198>

My goal was to make this test accessible to everyone. Although there are many ways to run this test through Matlab, I hadn't seen any code to implement this test in R. I hope it is useful for you, and if you have any questions or ways to improve this code, please contact me at [calhoun.peter@gmail.com](mailto:calhoun.peter@gmail.com)

Update (21.04.2010): In case you are facing a table with structural zeros (that is, missing values in the table), the package [aylmer](#) might be able to help you (it offers a generalization of Fisher's exact test)

Update (16.06.2011): A more updated (and faster) code is now available on the post (thanks goes, again, to Peter Calhoun). The post is updated with a newer example. The old R code for the function can still be [found here](#).

Update (12.04.2012): I just found out that there is a new R package (from 2011-11-21) that also implements Barnard's test. You can find it [here](#).

## 2.2 Computing Odds Ratios in R

[ExploringDataBlog](#)

[Measuring association using odds ratios](#)