

# RStudio/Testes diagnósticos

De SisLAU

Neste tutorial utilizaremos o RStudio para aplicar a regra de Bayes em testes diagnósticos.

Este roteiro discute sensibilidade, especificidade, valores preditivos e razões de verossimilhança. Este assunto tem aplicação direta na prática médica; você pode levá-lo para a "beira do leito".

## Índice

- 1 Histórico
- 2 Objetivos
- 3 Entrada de dados
  - 3.1 A partir dos dados brutos
    - 3.1.1 O pacote epiR
  - 3.2 A partir dos dados-resumo
  - 3.3 A partir de sensibilidade, especificidade e prevalência
- 4 Saída dos resultados
- 5 Formas para usar TesteDiagnostico.R

Para este tutorial, baixe e descompacte os RScripts create\_table.R, dataframe\_to\_table.R, sespprev\_to\_table.R e TesteDiagnostico.R encontrados em Bayes.zip e os dados em GESTANTES AFolico B12 rubeola.txt.

## Histórico

### Fundamentos da Pesquisa Científica em Medicina



Medicina

Outras Unidades Curriculares

Fonoaudiologia

Departamentos

Aulas

Resumos e Cadernos

E-books

Questões

Agenda e locais  
de aula

Outros materiais e links úteis

R scripts

R data

Classroom Turma A

(<https://classroom.google.com/u/0/c/MjgzODcyMzIzMDda>)

Classroom Turma B

(<https://classroom.google.com/u/0/c/MjgzODcyODE5OTRa>)

Código MSP1290

Júpiter: (<https://uspdigital.usp.br/jupiterweb/obterDisciplina?sgldis=MSP1290>)

Responsáveis: Paulo S. P. Silveira

José de Oliveira Siqueira

Koichi Sameshima

Maria Aparecida A. K. Folgueira

Alicia Matijasevich Manitto

Edimar Bocchi

Paulo Rossi Menezes

Colaboradores:

José Jukemura

Gilberto de Castro Júnior

Luisa Lina Villa

Lúcia da Conceição Andrade

Ester Sabino

Marcelo Tatit Sapienza

Ulysses Ribeiro Júnior

Alexandre Faisal Cury

Heráclito C. Barbosa

José Eluf Neto

Alexandre Granjeiro

Este roteiro está baseado no que foi inicialmente elaborado pelo Prof. Koichi Sameshima, em maio de 2012 e transcrita para o formato wiki por Helio Junji Shimozako e pelo Prof. Paulo S. P. Silveira em abril de 2013, versão ainda disponível clicando-se aqui.

Renata Levy

Maria Fernanda Peres

Situação desta Unidade Curricular (UC29)

Relatório editorial

Editar este quadro ([http://sislau.fm.usp.br/index.php?title=Predefini%C3%A7%C3%A3o:Unidade\\_curricular/UC29&action=edit](http://sislau.fm.usp.br/index.php?title=Predefini%C3%A7%C3%A3o:Unidade_curricular/UC29&action=edit))

Com a adoção do R / R Commander para esta disciplina, a versão aqui apresentada partiu da transferência do roteiro antigo por Tiana Carla Lopes Moreira em dezembro de 2013, adaptada e ampliada pelo Prof. Paulo Silveira, que adicionou os equivalentes feitos com o uso do Rcommander em abril de 2014.

Em 2019 deixamos o Rcommander e passamos a apresentar a programação em R (facilitada pelo uso do RStudio). Os exemplos desenvolvidos em planilha (Excel, Gnumeric e similares) não são mais necessários. Os professores Paulo Silveira e José Siqueira desenvolveram *scripts* em R, aqui apresentados, que você pode facilmente adaptar às suas necessidades.

## Objetivos

Ao final deste roteiro, você deve ser capaz, utilizando R, de:

- Abrir o ambiente do RStudio e executar *scripts* em R;
- Escolher e alterar os scripts fornecidos para cada necessidade;
- Conceituar e aplicar os conceitos de padrão-ouro em relação a um teste diagnóstico;
- Calcular sensibilidade, especificidade, valores de verossimilhança e valores preditivos;
- Obter e interpretar os resultados providos pelas funções do R (particularmente *epiR::epi.tests()*).

## Entrada de dados

Aqui necessitamos de tabelas de contingência 2x2, tipicamente montadas com contagens sobre o padrão-ouro (*gold-standard*) nas colunas, o teste, exame, sinal ou sintoma nas linhas:

	Doença+	Doença-	
Teste +	a	b	a+b
Teste -	c	d	c+d
	a+c	b+d	n=a+b+c+d

Dados podem ser fornecidos em diversos formatos e, portanto, você tem que lidar com isso.

## A partir dos dados brutos

Vamos, para exemplificar, retomar os dados utilizados na aula de regressão linear. Os dados de 65 pacientes estão em **GESTANTES AFolico B12 rubeola.txt**. Veja como funciona o *script* `dataframe_to_table.R` que começa com seguintes comandos:

```
filename <- "GESTANTES_AFolico_B12_rubeola.txt"
gestantes <- read.table(filename,header=TRUE,sep="\t",dec=".")
```

Funcionará se o arquivo estiver na mesma pasta que o *script* (o que é recomendável porque facilita); caso contrário, ajuste. A função *read.table()* traz os dados do arquivo texto prevendo que a primeira linha tem o nome das variáveis (header=TRUE), colunas separadas por tab (*sep*="\t") e números usando o ponto decimal (*dec*="."). A tabela ficará montada na variável *gestantes*

Vamos construir uma tabela de contingência presumindo que a medida de hemoglobina sérica (HB) é o padrão-ouro, e testaremos o hematócrito (HT) como candidato mais simples e barato a substituí-la.

O comando R é *table*, que tem o formato.

```
tabela <- table (gestantes$HT, gestantes$HB)
```

Este comando faz as contagens, mas observe seu conteúdo, com grande quantidade de zeros. A tabela na qual podemos operacionalizar a regra de Bayes deve ter duas colunas (as categorias com o diagnóstico confiável de Anemia ou não-Anemia) e duas linhas (o mesmo diagnóstico dado pelo Hematócrito). Temos, portanto, que classificar as pacientes em duas categorias de acordo com a Hemoglobina (anêmicas quando  $HB < 12g/cl$ ) e duas categorias de acordo com o Hematócrito (anêmicas quando  $HT < 37\%$ ). Então é necessário criar colunas adicionais (HBcat e HTcat, *cat* para lhe lembrar que a nova variável é categórica) no dataframe *tabela*:

```
gestantes$HBcat[gestantes$HB < 12] <- "Gold+"
gestantes$HBcat[gestantes$HB >= 12] <- "Gold-"
gestantes$HTcat[gestantes$HT < 37] <- "Anemia+"
gestantes$HTcat[gestantes$HT >= 37] <- "Anemia-"
```

e, então construir a tabela:

```
tabela <- table (gestantes$HTcat, gestantes$HBcat)
```

Veja o conteúdo armazenado na tabela com

```
tabela
```

Caso tudo tenha corrido bem, deverá encontrar:

	Gold-	Gold+
Anemia-	44	1
Anemia+	18	2

Convenientemente, o R usa como nomes das colunas e das linhas os textos que você colocou nas novas colunas de *gestantes*. Use, portanto, nomes convenientes.

## O pacote epiR

Em R, o melhor pacote que encontramos para lidar com as implicações diagnósticas da regra de Bayes, é o *epiR*. Como sempre, é necessário instalar e colocar na memória com:

```
install.packages("epiR")
library("epiR")
```

O código mínimo para obter todas as medidas que observamos na aula teórica é

```
out <- epi.tests(tabela, conf.level = 0.95)
print(out)
sumario <- summary(out)
print(sumario)
```



Estas rotinas fornecem as estimativas pontuais e, também, os intervalos de confiança. O valor de  $\alpha=5\%$  é fornecido em *conf.level=0.95*.

A primeira função, *epi.tests* produz:

	Outcome +	Outcome -	Total
Test +	44	1	45
Test -	18	2	20
Total	62	3	65

Point estimates and 95 % CIs:

Apparent prevalence	0.69 (0.57, 0.80)
True prevalence	0.95 (0.87, 0.99)
Sensitivity	0.71 (0.58, 0.82)
Specificity	0.67 (0.09, 0.99)
Positive predictive value	0.98 (0.88, 1.00)
Negative predictive value	0.10 (0.01, 0.32)
Positive likelihood ratio	2.13 (0.43, 10.63)
Negative likelihood ratio	0.44 (0.18, 1.06)

enquanto *summary* gera:

	est	lower	upper
aprev	0.6923077	0.56552711	0.8008931
tprev	0.9538462	0.87098872	0.9903789
se	0.7096774	0.58050980	0.8180060
sp	0.6666667	0.09429932	0.9915962
diag.acc	0.7076923	0.58174132	0.8139710
diag.or	4.8888889	0.41672350	57.3551396
nnd	2.6571429	-3.07511702	1.2351745
youden	0.3763441	-0.32519088	0.8096022
ppv	0.9777778	0.88229567	0.9994375
npv	0.1000000	0.01234853	0.3169827
plr	2.1290323	0.42633228	10.6320319
nlr	0.4354839	0.17887430	1.0602205



O *script* `dataframe_to_table.R` faz todas as operações definidas aqui. É o código mínimo para operar a partir de um conjunto de dados conhecido.

A partir dos dados-resumo

Imagine que, em vez de lhe fornecer os dados brutos, só tenhamos acesso à tabela. É uma situação comum em trabalhos científicos publicados. Retomemos o exemplo da aula teórica.

<b>Ferritina sérica</b>	<b>Anemia</b>		<b>Total</b>
	Sim	Não	
$FS < 15 \mu g/l \Rightarrow +$	731	270	1001
$FS \geq 15 \mu g/l \Rightarrow -$	78	1500	1578
<b>Total</b>	<b>809</b>	<b>1770</b>	<b>2579</b>

Estes dados, modificados a partir de um exemplo (<https://ebm-tools.knowledgetranslation.net/card>), confronta o exame de ferritina sérica com hemoglobina. Os dados originais, porém, não estão disponíveis.

Em R, facilmente, podemos construir a tabela com:

```
tabela <- as.table(matrix(c("a","b",
                           "c","d"),
                          nrow = 2, byrow = TRUE))
colnames(tabela) <- c("D+", "D-")
rownames(tabela) <- c("T+", "T-")
```

No caso, substitua os valores adequados:

```
tabela <- as.table(matrix(c(731,270,
                           78,1500),
                          nrow = 2, byrow = TRUE))
colnames(tabela) <- c("Anemia+", "Anemia-")
rownames(tabela) <- c("Ferritina+", "Ferritina-")
```

Confira com o comando:

```
print (tabela)
```

obtendo:

```
      Anemia+ Anemia-
Ferritina+   731    270
Ferritina-    78   1500
```

## A partir de sensibilidade, especificidade e prevalência

Em muitos trabalhos, ou quando quiser simular situações, só estão disponíveis os valores de sensibilidade e especificidade, e a estimativa populacional da prevalência.

Você pode recriar a tabela para uma população dando os dados. A partir do exemplo acima, experimente:

```
sensibilidade <- 0.9035847
especificidade <- 0.8474576
prevalencia <- 0.3136875
n <- 2579
```

```
tabela <- as.table(matrix(n*c(prevalencia*sensibilidade, (1 - prevalencia)*(1 - especificidade),
                             prevalencia*(1 - sensibilidade), (1 - prevalencia)*especificidade),
                          nrow = 2, byrow = TRUE))
colnames(tabela) <- c("D+", "D-")
rownames(tabela) <- c("T+", "T-")
```

```
print (tabela)
```

Para obter:

```
      D+      D-
T+  731.00008 270.00004
T-   77.99998 1499.99990
```

Repare que recriamos a tabela a partir da sensibilidade, especificidade e prevalência. Novamente, com a tabela formada, é só usar as funções do *epiR*.

Há pequenos erros por causa de arredondamentos dos valores de sensibilidade e especificidade. Caso queira processar a tabela com números inteiros, use

```
tabela <- round(tabela, 0)
```

e confira com

```
print (tabela)
```

para obter

```
      D+      D-
T+   731    270
T-    78   1500
```



Caso não queira simular uma população, use  $n=1$ . Os intervalos de confiança não terão significado, mas as estimativas pontuais, para um único paciente, serão as mesmas.

## Saída dos resultados

O *script* TesteDiagnostico.R, disponibilizado e usado para a aula teórica, fornece a saída do *epiR* e vários gráficos. As saídas das funções do *epiR* aparecem na tela, no RStudio na área da **Console** (como foi feito nos exemplos anteriores). Os gráficos aparecem em janela à parte, no RStudio na área de **Plots**.

Alternativamente, pode-se guardar as saídas em arquivos texto ou pdf.



Este *script* é bem mais longo que os anteriores. Sugerimos que localize os trechos do código aqui apresentados e os adapte às suas necessidades.

O jeito mais simples, adotado aqui, é com um nome de arquivo de saída fixado no início do *script*:

```
filename <- "Teste_diagnostico"
```

Caso prefira a saída em tela, use

```
filename <- NA
```

(NA, para R, não é uma variável; significa *not available*, não disponível)

Quando a saída de dados vai ser iniciada, aparece:

```
if (!is.na(filename))  
{  
  filetxt <- paste(filename, ".txt", sep="")  
  filepdf <- paste(filename, ".pdf", sep="")  
  sink(filetxt)  
  pdf(filepdf)  
}
```

Este procedimento verifica se existe nome de arquivo em *filename*. Caso exista, cria dois outros nomes de arquivo, adicionando as extensões *.txt* e *.pdf*. Os dois comandos seguintes instruem o R a desviar todas as saídas textuais e todos os gráficos para os dois arquivos, respectivamente.

No final do *script*, para voltar a usar a console e ver os gráficos na área de [Plots], aparece:

```
sink()  
dev.off()
```

O *sink()* vazio interrompe a saída de dados para o arquivo *txt*. O comando *dev.off()* descarrega toda a saída gráfica para o arquivo *pdf* e o fecha.



Quando o processamento de um script for interrompido por qualquer motivo com *sink* dirigindo a saída para um arquivo, digite manualmente

```
sink()
```

para poder voltar a usar a *Console*.

Da mesma forma, pode haver problema com o *pdf* preso. Então use manualmente

```
dev.off()
```

## Formas para usar TesteDiagnostico.R

Procuramos fazer este *script* o mais flexível possível. A tabela de contingências é montada a partir de prevalência, sensibilidade e especificidade, como em `sespprev_to_table.R` apresentado acima. Portanto, não é difícil caso queira fazer versões que recebam dados brutos ou dados resumidos.

Outra flexibilidade é que você pode escolher o valor de *n*: caso escolha 1, calcula para um paciente; pode escolher outro valor positivo, para simular uma população, ou pode deixar em branco para deixar o programa escolher o tamanho da população para você. Neste último caso, o programa pergunta o *delta* - aumentará o tamanho da população até que a variação do intervalo de confiança de LR+ e LR- estabilize (seja menor que este *delta*).

Este *script*, então, lhe dá duas saídas: a primeira é o resultado das funções do *epiR*. O restante são resultados da simulação, na forma de gráficos e algumas estatísticas, incluindo medidas de tamanho de efeito.

Disponível em "[http://sislau.fm.usp.br/index.php?title=RStudio/Testes\\_diagnósticos&oldid=29397](http://sislau.fm.usp.br/index.php?title=RStudio/Testes_diagnósticos&oldid=29397)"

- 
- Esta página foi modificada pela última vez à(s) 18h30min de 8 de março de 2020.