

Universidade do Minho

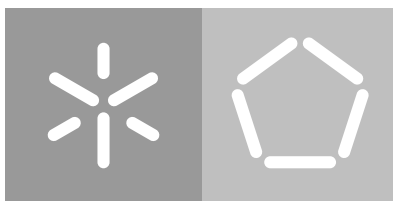
Escola de Engenharia

Departamento de Informática

José Pedro dos Santos Quintas

**Análise através da curva ROC:
que ferramentas utilizar?**

May 2020



Universidade do Minho

Escola de Engenharia

Departamento de Informática

José Pedro dos Santos Quintas

Análise através da curva ROC: que ferramentas utilizar?

Dissertação de mestrado

Mestrado em Bioinformática

Dissertação supervisionada por

Ana Cristina Braga

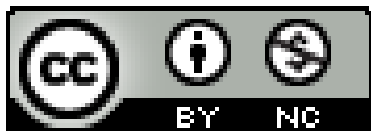
May 2020

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

AGRADECIMENTOS

Para esta dissertação foi relevante o apoio, a colaboração e o incentivo de diversas pessoas, às quais quero deixar um agradecimento especial. Em primeiro lugar quero agradecer à minha orientadora, Professora Ana Cristina Braga, pelo conhecimento partilhado, pela aprendizagem, pela oportunidade de crescimento e sobretudo pela inteira disponibilidade demonstrada. Meus sinceros agradecimentos pela consideração e pela confiança depositada em mim ao longo deste percurso. Quero novamente agradecer pela orientação a qual permitiu um grande crescimento pessoal e académico no decorrer da elaboração desta dissertação. Agradecer ainda ao Departamento de Produção de Sistemas (DPS) da Universidade do Minho pela disponibilização de *software* capaz de trabalhar na área exigida para esta dissertação.

Quero agora agradecer à Sarah, a minha companheira, que sem dúvida foi o grande pilar nas fases complicadas deste trabalho. Sempre prestável, compreensiva e a pessoa que sempre acreditou em mim. Por todo o amor e amizade a quem só tenho a agradecer por ter estado sempre lá para me ouvir e corrigir de forma ponderada e acertada. Agradecer ainda aos “Maiores” por todos os momentos passados fora da Universidade, que entre cafés e convívios foram um grande ponto de distração e desabafos. Agradecer ao Hugo e ao Gil que sempre se demonstraram prontos a ajudar e a ouvir os meus problemas. Também ao Bruno que mesmo estando longe foi dos mais interessados pelo meu percurso académico, um bom ouvinte e um grande amigo. Por fim, o mais importante, quero agradecer à minha família, aos meus pais pelo apoio prestado nos mais diversos níveis e também ao meu irmão e cunhada que estiveram sempre disponíveis a ajudar, quero ainda agradecer aos quatro pelo incentivo, pela força, pela coragem e pelo carinho transmitidos.

Quero novamente agradecer a toda a gente que me apoiou neste percurso, com pequenos gestos que fazem toda a diferença, o meu muito obrigado!

“A coisa mais valiosa que podes fazer é cometer um erro, não conseguirás aprender seja o que for ao ser perfeito.” – Adam Osborne

ABSTRACT

ROC (Receiver Operating Characteristic) curve is a statistic tool that allows the evaluation of the accuracy of a classification system. These curves are drawn on a two-dimensional graph, with the ordinate representing the true positive fraction or sensitivity and the abscissa representing the false positive fraction or 1 -specificity. The index that evaluates the accuracy of these graphs is represented by the area under the curve (AUC) where the larger that area is the bigger the test performance is.

Its first appearance dates to the year of 1950. Nevertheless, computationally, the first software only appeared around 1993 and since then several tools have been made available for its analysis. Regarding the theoretical part of the subject, there is a vast bibliography existing which introduces all the necessary concepts to analyze a ROC curve visually and statistically. However, only a few of those documents discuss the evaluation and the comparison of software that attain these same curves, consisting of old works in which the vast majority corresponds to software that when compared to the current scenario are outdated or fell out of use.

The R software environment with a programming language mainly for statistical use is currently one of the best tools to perform the ROC analysis. The variety of packages in this work environment make it an interesting study product, which allows us to take advantage of the different features in different the packages or enjoy the same features but by different means and formats. Like R there are several tools that can perform this same analysis, as is the case of STATA software, which receives regular updates that have been improving this tool recurrently. With the versatility of allowing us to work from a command line or through menus predefined by the software itself, it makes it a very accessible and convenient tool to explore.

The R language is also related to the package called shiny, which can create browser applications through its own commands, making it possible to transpose the different commands of packages R into a single application. Due to the wide variety of ROC packages in R, it is interesting to link them to shiny. Therefore, a library in the application format was designed to group the different packages on the same browser page. The result of this is ROSY application available on <https://pquintasbcl.shinyapps.io/ROSY/>.

Due to the increasing use of ROC analysis in different systems, it is essential to explore the best computational methods to process it in a correct way. Therefore, in this work the research and selection of different software/tools to perform this type of analysis is done, based on the different existing bibliographic documents in order to compare them and create

a checklist, which will allow us to visualize the fundamental characteristics present in each software analyzed.

Keywords: ROC curves, statistic, software, R, shiny, package, checklist

RESUMO

A curva ROC (Receiver Operating Characteristic) é uma ferramenta estatística que permite avaliar o desempenho de um sistema de classificação. Estas curvas são representadas num gráfico bidimensional, com a ordenada a representar a fração de verdadeiros positivos ou sensibilidade e a abcissa a representar a fração de falsos positivos ou 1-especificidade. O índice que avalia a medida de exatidão destes gráficos é a área abaixo da curva (AUC) e quanto maior for a área maior é o desempenho do sistema em causa.

A sua primeira aparição remete para o ano de 1950, contudo, computacionalmente o primeiro *software* terá surgido por volta de 1993 e, desde então, que têm sido disponibilizadas diversas ferramentas para a sua análise. Relativamente à parte teórica do tema abordado, é disponibilizada uma vasta bibliografia capaz de introduzir todos os conceitos necessários para se conseguir analisar visualmente e estatisticamente uma curva ROC. Contudo, há poucos registos relativos à avaliação e comparação de *software* que produzem estas mesmas curvas, sendo trabalhos demasiado antigos, ou seja, na sua grande maioria foi utilizado *software* que comparativamente ao cenário real está desatualizado ou caiu em desuso.

O R é um ambiente de trabalho com uma linguagem de programação destinada essencialmente à estatística, sendo por isso atualmente, uma das melhores fontes para realizar análise ROC. A variedade de *packages* existentes neste ambiente de trabalho torna-o num interessante produto de estudo, sendo possível tirar partido de diferentes funcionalidades em diferentes *packages* ou então usufruir das mesmas funcionalidades mas por meios e formatos distintos. Em paralelo com o R existem diversas ferramentas capazes de realizar esta mesma análise, como é o caso do *software* STATA, que sofre recorrentemente atualizações que têm aprimorado esta ferramenta. Com a versatilidade de se poder trabalhar através de uma linha de comandos ou através de menus pré-definidos pelo próprio *software* torna-o bastante acessível e prático de explorar.

O ambiente de trabalho R está ainda relacionado com um *package* denominado *shiny*, que possui a capacidade de criar aplicações *browser* através de comandos próprios, sendo assim possível transpor os diferentes comandos de *packages* R numa aplicação. Devido à grande variedade de *packages* ROC existentes neste ambiente de trabalho, torna-se interessante criar uma ligação dos mesmos com o *shiny*. Sendo por isso, idealizada uma *library* no formato de aplicação para agrupar os diferentes *packages* numa mesma página *web*. O resultado é a aplicação ROSY disponível em <https://pquintasbcl.shinyapps.io/R0SY/>.

Devido à crescente utilização deste tipo de análise nos diferentes sistemas, torna-se fundamental explorar os melhores métodos computacionais para processar uma correta

análise ROC, pelo que, neste trabalho é efetuada a pesquisa e seleção de diferentes *software*/ferramentas capazes de realizar este tipo de análise, tendo como base os diferentes registos bibliográficos já existentes e posteriormente a sua comparação e criação de uma *checklist*, que permitirá visualizar as características fundamentais presentes em cada *software* analisado.

Keywords: Curvas ROC, estatística, *software*, R, *shiny*, *package*, *checklist*

ÍNDICE

1	INTRODUÇÃO	1
1.1	Enquadramento	1
1.2	Motivação	2
1.3	Objetivos	3
1.4	Metodologia	3
1.5	Estrutura da dissertação	4
2	ESTADO DA ARTE	5
2.1	Análise ROC e a sua história	5
2.2	Metodologia da curva ROC	7
2.3	Indicadores de desempenho da curva ROC	8
2.4	Indicadores de mortalidade neonatal	9
2.5	Outros estudos que abordam a metodologia	10
2.6	<i>Software</i> curva ROC	11
2.7	Proposta de <i>ckecklist</i> de funções	14
2.8	Base de dados utilizada	15
2.9	Aplicações <i>shiny</i>	16
3	SOFTWARE/FERRAMENTAS/ <i>packages</i> PARA ANÁLISE ROC	19
3.1	<i>Packages</i> R	19
3.1.1	<i>pROC</i>	19
3.1.2	<i>ROCR</i>	24
3.1.3	<i>ROSE</i>	26
3.1.4	<i>ROC</i> do <i>Bioconductor</i>	27
3.1.5	<i>verification</i>	29
3.1.6	<i>Epi</i>	30
3.1.7	<i>caTools</i>	32
3.1.8	<i>Comp2ROC</i>	34
3.2	<i>Software</i> STATA	36
3.2.1	<i>roctab</i>	37
3.2.2	<i>roccomp</i>	41
3.2.3	<i>rocreg</i> e <i>rocregplot</i>	45
3.2.4	<i>rocgold</i>	48
3.2.5	<i>rocfi</i>	48
4	<i>shiny</i>	50

4.1	Como obter e utilizar <i>shiny</i>	51
4.2	Como construir uma aplicação <i>shiny</i>	51
4.3	Criação da aplicação	51
4.3.1	<i>ui.R</i>	52
4.3.2	<i>server.R</i>	54
5	RESULTADOS	57
5.1	Elaboração da <i>checklist</i>	57
5.1.1	Critérios de cotação	58
5.1.2	Análise crítica	59
5.2	Construção do <i>score</i> de avaliação das ferramentas analisadas	61
5.2.1	Critérios de construção	61
5.2.2	Avaliação das ferramentas	61
5.3	Aplicação <i>shiny</i> para seleção de <i>packages</i> R	63
5.3.1	Fluxograma da aplicação <i>shiny</i>	63
5.3.2	<i>Checklist</i> e <i>Userguide</i>	65
5.3.3	<i>Upload</i> de ficheiros	66
5.3.4	<i>Output</i> para uma curva	68
5.3.5	<i>Output pROC</i> - Comparar duas curvas	70
5.3.6	<i>Output Comp2ROC</i> - Amostras dependentes e independentes	70
6	CONCLUSÃO	73
6.1	Conclusões	73
6.2	Perspectivas para trabalhos Futuros	75

LISTA DE FIGURAS

Figura 1	Gráficos bidimensionais de análise ROC.	9
Figura 2	Criação da ckecklist.	14
Figura 3	Amostra dependente no editor <i>Excel</i> .	15
Figura 4	Amostra independente no editor <i>Excel</i> .	15
Figura 5	Gráfico <i>pROC</i> .	20
Figura 6	Gráfico <i>pROC</i> com <i>ggroc</i> .	21
Figura 7	<i>pROC direction</i> comandos.	22
Figura 8	Gráficos <i>direction</i> do <i>pROC</i> .	22
Figura 9	Exemplo para a função <i>roc.test</i> do <i>pROC</i> .	23
Figura 10	Comandos <i>ROCR</i> .	24
Figura 11	<i>Output ROCR</i> .	25
Figura 12	<i>Output ROSE</i> .	27
Figura 13	Comandos <i>Bioconductor</i> .	28
Figura 14	<i>Output Bioconductor</i> .	28
Figura 15	Comandos <i>verification</i> .	30
Figura 16	<i>Output verification</i> .	31
Figura 17	<i>Output Epi</i> .	32
Figura 18	Comandos <i>Epi</i> .	32
Figura 19	Comandos <i>caTools</i> .	33
Figura 20	<i>Output caTools</i> .	33
Figura 21	Gráfico ROC <i>Comp2ROC</i> .	34
Figura 22	Gráfico áreas e histograma do <i>Comp2ROC</i> .	35
Figura 23	<i>Output</i> no terminal do <i>Comp2ROC</i> .	35
Figura 24	Comandos <i>Comp2ROC</i> .	35
Figura 25	Aceder menu <i>roctab</i> .	37
Figura 26	Menu <i>roctab</i> .	39
Figura 27	<i>Output</i> menu <i>roctab</i> exemplo 1.	39
Figura 28	<i>Output</i> menu <i>roctab</i> exemplo 2.	39
Figura 29	Gráfico ROC gerado pelo menu <i>roctab</i> .	40
Figura 30	Comandos <i>roctab</i> .	40
Figura 31	<i>Output</i> linha de comandos <i>roctab</i> .	41
Figura 32	Gráfico ROC gerado por comandos <i>roctab</i> .	42
Figura 33	Aceder menu <i>roccomp</i> .	42

Figura 34	<i>Output roccomp.</i>	43
Figura 35	Curvas num mesmo gráfico <i>roccomp.</i>	43
Figura 36	Curvas ROC em gráficos separados <i>roccomp.</i>	44
Figura 37	<i>Dataset</i> amostras independentes <i>roccomp.</i>	44
Figura 38	Exemplo menu <i>roccomp</i> para amostras independentes.	45
Figura 39	Gráfico independentes <i>roccomp.</i>	45
Figura 40	Aceder menu <i>rocreg.</i>	46
Figura 41	Menu <i>rocreg.</i>	46
Figura 42	Aceder menu <i>rocregplot.</i>	47
Figura 43	Menu <i>rocregplot.</i>	47
Figura 44	Aceder menu <i>rocgold.</i>	48
Figura 45	Aceder menu <i>rocfit.</i>	49
Figura 46	Aceder menu <i>rocfit</i> para processar gráfico.	49
Figura 47	Diagrama componentes <i>shiny.</i>	52
Figura 48	Código separadores da aplicação.	53
Figura 49	Código do corpo do separador.	53
Figura 50	<i>Dataset</i> reativo aplicação.	54
Figura 51	<i>Dataset</i> reativo aplicação <i>Comp2ROC.</i>	55
Figura 52	Comandos <i>renderPlot</i> e <i>renderPrint.</i>	56
Figura 53	Fluxograma <i>shiny.</i>	64
Figura 54	<i>Checklist - ROSY.</i>	65
Figura 55	<i>User Guide - ROSY.</i>	66
Figura 56	<i>Upload</i> geral - <i>ROSY.</i>	66
Figura 57	<i>Comp2ROC</i> File Depedents - <i>ROSY.</i>	67
Figura 58	<i>Comp2ROC</i> File Independents - <i>ROSY.</i>	67
Figura 59	Caixas de texto de <i>input</i> para uma curva - <i>ROSY.</i>	69
Figura 60	<i>Output</i> uma curva - <i>ROSY.</i>	69
Figura 61	<i>Comparasion 2 curves pROC - ROSY.</i>	70
Figura 62	<i>Output Comp2ROC</i> dependentes exemplo 1 - <i>ROSY.</i>	71
Figura 63	<i>Output Comp2ROC</i> dependentes exemplo 2 - <i>ROSY.</i>	71
Figura 64	<i>Output Comp2ROC</i> independentes exemplo 1 - <i>ROSY.</i>	72
Figura 65	<i>Output Comp2ROC</i> independentes exemplo 2 - <i>ROSY.</i>	72

LISTA DE TABELAS

Tabela 1	Matriz de confusão de classificadores ROC.	8
Tabela 2	Descrição <i>CRAN</i> para funções <i>pROC</i> .	23
Tabela 3	Descrição opções <i>roctab</i> .	41
Tabela 4	<i>Checklist</i> de todas as ferramentas.	57
Tabela 5	Métricas de avaliação das ferramentas ROC.	62
Tabela 6	Tabela da avaliação detalhada das ferramentas ROC.	62

ACRÓNIMOS

A

ARN Ácido ribonucleico.

AUC Area Under the Curve.

C

CLASSVAR classification variable.

CRAN Comprehensive R Archive Network.

CRIB Clinical Risk Index for Babies.

D

D-VALUE decimal reduction time.

F

FFN Fração de Falsos Negativos.

FFP Fração de Falsos Positivos.

FN Falsos Negativos.

FP Falsos Positivos.

FVN Fração de Verdadeiros Negativos.

FVP Fração de Verdadeiros Positivos.

I

IC Intervalo de Confiança.

N

NA Not Available.

NTISS Neonatal Therapeutic Intervention Scoring System.

P

P-VALUE probability value.

PAUC partial Area Under the Curve.

PCA Principal Component Analysis.

R

REFVAR reference variable.

ROC Receiver Operating Characteristic.

S

SNAP Score for Neonatal Acute Physiology.

SNAP-PE Score for Neonatal Acute Physiology - Perinatal Extension.

U

UI user interface.

V

VN Verdadeiros Negativos.

VP Verdadeiros Positivos.

VUS Volume Under Surface.

INTRODUÇÃO

1.1 ENQUADRAMENTO

A teoria associada à curva **Receiver Operating Characteristic (ROC)** foi desenvolvida na segunda guerra mundial por engenheiros operadores de radar, visando a detecção de aviões inimigos pelos radares. Esta teoria foi também implementada por psicólogos na área da psicomетria, para a detecção e percepção de estímulos. A metodologia de suporte a esta teoria foi então desenvolvida tendo dois pilares fundamentais: a teoria de detecção de sinal e a teoria de decisão estatística (Metz, 2008). As suas características e o poder de análise inegável, permitiram a sua utilização em diferentes áreas como a psicologia experimental, economia, ciências da terra, sociologia, ciências atmosféricas, *machine learning*, mineração de dados e diagnóstico médico (Gonçalves et al., 2014; Krzanowski and Hand, 2009). Tratando-se de um dissertação na área da Bioinformática, procurar-se-á dar maior enfoque à aplicação das últimas três áreas.

Na área do *machine learning*, Spackman foi um dos primeiros utilizadores de gráficos **ROC** para avaliar o desempenho de diferentes máquinas de aprendizagem (Spackman, 1989). Relativamente à mineração de dados, é recorrente a aplicação das curvas **ROC** nos trabalhos realizados na área, contudo o tema não é devidamente enquadrado nos seus estudos, tornando necessário ao leitor procurar uma melhor explicação para o que é a análise **ROC** noutros trabalhos. Embora os gráficos sejam relativamente simples, existem pequenas questões que não são facilmente perceptíveis para pessoas com menos conhecimento na área (Metz, 1978). Nos diagnósticos médicos estas curvas têm o poder de avaliar a sua qualidade, classificando os pacientes como positivos ou negativos em relação a determinada doença (Metz, 2008). A curva **ROC** é atualmente a ferramenta estatística mais desenvolvida para descrever o desempenho destes testes (Pepe, 2011).

Estas curvas são então, uma ferramenta muito útil para avaliar a exatidão de um diagnóstico médico. A sua utilização permite visualizar e organizar classificadores tendo em conta o seu desempenho. As curvas **ROC** são representadas em gráficos bidimensionais onde x , abcissa, representa a *Fração de Falsos Positivos (FFP)* e y , ordenada, representa a *Fração de Verdadeiros Positivos (FVP)*. Às **FFP** e **FVP**, na avaliação de um teste de diagnóstico, estão

associados outros dois conceitos, a especificidade ($1 - FFP$) e a sensibilidade (FVP). Estes valores são representados num plano unitário como medidas de probabilidades que variam entre 0 e 1.

Num diagnóstico médico, a FVP poderá significar que o indivíduo doente é bem classificado, enquanto *Fração de Verdadeiros Negativos* (FVN) significa um indivíduo saudável que é corretamente classificado no diagnóstico (Braga, 2000).

O índice que permite avaliar a precisão de um classificador no teste de diagnóstico, é a *Area Under the Curve* (AUC), calculada através dos gráficos ROC . Assim, verifica-se que quanto maior for a área, maior é o desempenho do teste em que foi utilizado como classificador.

As curvas ROC são então consideradas um mecanismo de análise de excelência na área de diagnóstico, pela sua capacidade em descrever com precisão um teste contínuo ou de valor ordinal.

Através da análise ROC as informações de diferentes testes podem ainda ser combinadas e comparadas Pepe (2011).

Em 1989 era estimado que na área da análise ROC existisse uma taxa de publicações de cerca de 30 artigos por ano Hanley (1989). Já em 2009, Krzanowski e Hand mostram que esse número aumentou bastante, passando de 854 artigos (1992-1995) para 4463 (2004-2007) (Krzanowski and Hand, 2009). Esta crescente popularidade cria a necessidade de se utilizar *software* específico como ferramenta de auxílio, como por exemplo a utilização de *packages* R para a realização destas análises (Sing et al., 2005). Assim, nesta dissertação será compilada e analisada uma série de ferramentas capazes de realizar este tipo de análise com curvas ROC . Para suportar o desenvolvimento desta dissertação, levantaram-se as seguintes questões que se pretendem responder com esta investigação:

- Qual o *software* a usar para análise ROC face a um determinado objetivo? Quais as estatísticas que utilizam?
- Existe algum *software* que compare sistemas através das curvas ROC empíricas?

1.2 MOTIVAÇÃO

O crescente aumento de popularidade na área das curvas ROC levou também ao seu desenvolvimento informático, sendo possível, com alguns comandos básicos ou simplesmente acedendo a diferentes menus, a criação de curvas ROC que avaliam diferentes tipos de testes nas mais variadas áreas. Tal como abordado no subcapítulo 1.1, esta área tem uma taxa de publicações em crescendo, sendo que, os 4463 artigos publicados entre 2004-2007 provam um aumento do interesse da mesma. Associados a estes artigos surgem também novas ferramentas que realizam esta análise, contudo na sua grande maioria não existem

trabalhos que definam corretamente a capacidade de cada uma destas ferramentas, surgindo a necessidade de realizar uma nova e melhor avaliação qualitativa de algumas ferramentas de interesse na área, servindo como base de apoio aos utilizadores.

1.3 OBJETIVOS

Tendo em conta as questões levantada para esta investigação, foram então posteriormente delineados os seus objetivos. Considerando a área em que se insere, irá ser dado relevo à abordagem das curvas *ROC* na área da Bioinformática.

Assim, traçou-se como objetivo geral desta dissertação:

- efetuar um levantamento de *software* existentes para análise através de curvas *ROC* e avaliar as suas funcionalidades.

Com base neste objetivo geral delinearam-se os seguintes objetivos específicos:

- perceber a utilidade da análise *ROC* para analisar sistemas;
- realizar análises com exemplos a aplicar nas diferentes áreas do conhecimento;
- testar os exemplos em cada um dos *software* utilizados;
- fazer uma *checklist* com os pontos comuns e as especificidades de cada um dos *software*;
- perceber de que forma o *package shiny* poderá ser utilizado na análise *ROC* e nesta dissertação;
- criação de uma aplicação *shiny* enquadrada com a temática da análise *ROC*.

1.4 METODOLOGIA

Tendo em conta o problema proposto e os objetivos delineados, o tipo de estudo será essencialmente descritivo com características exploratórias. Serão utilizadas diversas ferramentas tendo em conta cada uma das suas especificidades e analisadas as suas funções para os mesmos conjuntos de dados. Será criada uma *checklist* que visa uma revisão mais simplificada dos resultados deste estudo bem como criada uma aplicação relativa ao ambiente de trabalho R e aos seus *packages*.

Inicialmente será dada prioridade a *software* que já tenha alguma recomendação para realizar este tipo de análise e posteriormente o estudo passará para *software* menos reconhecido. O objetivo principal é então reunir o maior número de ferramentas para tornar a *checklist* o mais completa possível. Será tido em conta parâmetros como a morfologia

das curvas *ROC*, a simplicidade em realizar os testes, a sua eficácia estatística, os métodos matemáticos utilizados, os tutoriais disponibilizados, a forma de leitura dos dados, os dados que são fornecidos como *output*, o conforto do *software*, os seus aspetos funcionais e outras características de relevo.

1.5 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está dividida em seis capítulos. No primeiro capítulo da introdução é apresentado o enquadramento do tema em estudo, a motivação para este trabalho assim como a formulação das questões de investigação face ao problema proposto, o delineamento dos objetivos assim como a metodologia de investigação proposta para a resolução do problema. No segundo capítulo irá ser apresentado o estado da arte recorrendo à pesquisa bibliográfica sobre o tema e utilizando as palavras chave: análise *ROC*, *AUC*, curva *ROC*, *software ROC*, *R*, *package*, *shiny*, aplicação e *STATA*.

Utilizaram-se para o efeito fontes de referência bibliográfica primárias, como artigos, teses, relatórios técnicos e livros.

Relativamente ao terceiro capítulo, é apresentada uma breve descrição teórica dos diferentes *packages R*, bem como do *software STATA*. Posteriormente as diferentes funções disponíveis para cada uma das ferramentas serão testadas, utilizando para o efeito sempre o mesmo *dataset*, permitindo assim consistência nos resultados obtidos para posterior comparação. Um dos resultados obtidos, a (*checklist*), terá como base as características estudadas neste capítulo.

No quarto capítulo, é introduzida a temática da aplicação *shiny* que será explorada dentro do ambiente de trabalho *R*. Para o efeito são apresentadas algumas características deste *package R* que servirá de construção para uma aplicação *shiny*.

Para o quinto capítulo serão apresentados os resultados obtidos no final da dissertação. Para o efeito será demonstrada uma *checklist* onde estão demonstradas as características *ROC* das diferentes ferramentas e o formato em que se encontra a aplicação projetada para o *shiny*.

A dissertação terá ainda um capítulo final referente a uma conclusão e perspetivas futuras.

ESTADO DA ARTE

O capítulo do estado da arte é referente à pesquisa de trabalhos já elaborados dentro da temática desta dissertação. Para o efeito, serão recolhidos diferentes tipos de documentos de referência bibliográfica primária, como artigos, teses, relatórios técnicos e livros. Como o trabalho proposto nesta dissertação acaba por interagir com diferentes áreas, existe a necessidade de fazer uma busca mais alargada do que simplesmente a temática das curvas *ROC*. Sendo assim, para além da temática *ROC*, assuntos como, *software*, áreas de investigação, indicadores do *dataset*, aplicações *shiny* e outros assuntos inerentes serão também eles aqui abordados. Os conceitos teóricos sobre análise *ROC* apresentados neste capítulo têm como base o trabalho desenvolvido na tese de doutoramento de Braga (Braga, 2000).

2.1 ANÁLISE ROC E A SUA HISTÓRIA

Green e Swets em 1966, surgem na história da análise *ROC* relacionados com a teoria de deteção de sinal, sendo que os seus trabalhos remetem-se à área da psicologia. Nesta teoria os observadores decidem qual a regra de otimização para determinado sinal, o que deverá corresponder a uma maximização de respostas corretas para o caso em estudo (Green and Swets, 1966).

Posteriormente Swets e Pickett em 1982,, remetem o caso de estudo para a área de sistemas de diagnóstico, procurando abordar os muitos problemas que surgem na sua avaliação de desempenho (Swets and Pickett, 1982).

Na área de diagnóstico médico, surge o nome de Metz como um dos pioneiros na aplicação da análise *ROC* a este tipo de sistema. Inicialmente com um trabalho realizado em 1978 (Metz, 1978), onde refere que a curva *ROC* é uma descrição empírica simples mas completa de um diagnóstico médico. Nesse trabalho, são ainda descritos e introduzidos diversos conceitos de elevada relevância, como é o caso de “sensibilidade” e “especificidade”, bem como os índices relacionados “fração de verdadeiros positivos” e “fração de falsos positivos”.

Posteriormente, Metz publicou um trabalho no ano de 1986, na área da radiologia. Pretendeu assim fornecer aos radiologistas uma visão do acerto do teste de diagnóstico

usado na classificação dos pacientes como doentes ou saudáveis. Os conceitos da análise *ROC* voltaram a ser refinados para se tornarem mais intuitivos e é dada uma breve revisão aos requisitos necessários num estudo *ROC*, tornando-se uma prova da versatilidade que esta análise tem nas diversas áreas (Metz, 1986).

Em 1989, Metz apresenta um outro estudo na área do diagnóstico médico onde são enumerados os cuidados a ter pelos responsáveis médicos quando utilizam este tipo de metodologia (Metz, 1989). O mesmo autor continuou a realizar trabalhos de elevada relevância com diversos progressos na área da análise *ROC*, entre os quais trabalhos relacionados com o modelo binomial (Metz and Pan, 1999).

Em 2008, este autor apresenta uma revisão da literatura relativa à análise *ROC* aplicada à medicina, onde são abordados diversos tópicos de elevada relevância, como os níveis de eficácia num diagnóstico, a quantificação da precisão de um diagnóstico, os dados históricos da análise *ROC*, a aplicabilidade da análise *ROC* na medicina, as condicionantes do tipo de amostragem, ajuste da curva *ROC*, estimativa de máxima verosimilhança e como avaliar estatisticamente curvas *ROC* (Metz, 2008).

Para o conceito de *AUC*, este foi explorado por diversos autores entre os quais se destacam os estudos de Hanley e McNeil em 1982, onde se exploraram as diferentes definições atribuídos a este índice (Hanley and Mcneil, 1982). São referidos por estes autores trabalhos de Green e Swets (Green and Swets, 1966) que definiram a *AUC* como sendo a probabilidade de identificar corretamente qual estímulo era um “ruído” e qual era um “sinal mais ruído”, e também o trabalho de Green e Moses (Green and Moses, 1966) onde foi verificado empiricamente este conceito num teste de memória de reconhecimento.

Nesse trabalho de Hanley e McNeil os autores chegaram à mesma conclusão que Bamber mas aplicado a um diagnóstico médico, sendo que, em 1975 Bamber (Bamber, 1975) relacionou o conceito de *AUC* com a estatística de Wilcoxon e Mann-Whitney definindo-a como a probabilidade de classificar corretamente um par em normal ou anormal (Hanley and Mcneil, 1982).

Já em 1983, estes autores exploram a relação entre *AUC* com o teste estatístico de Wilcoxon como forma de determinar o erro padrão (SE), qual o tamanho da amostra necessário para medir com precisão a *AUC* e ainda para obter um poder estatístico desejado, ou seja, baixa taxa de erro (Hanley and Mcneil, 1983). Neste trabalho são ainda referidas algumas formas de cálculo da *AUC*, entre elas a regra do trapézio, cálculo a partir da razão de máxima verosimilhança de Dorfman e Alf e ainda a partir da representação da curva *ROC* no plano designado por binomial. O método recomendado pelos autores é o de Dorman e Alf por ser mais robusto (Dorfman and Alf, 1969).

2.2 METODOLOGIA DA CURVA ROC

Como referido no Capítulo 1.1 desta dissertação, as curvas *ROC* empíricas podem ser representadas em gráficos bidimensionais, num plano unitário com medidas de probabilidade de valores a variar de 0 a 1. Na coordenada das abcissas é representada a *FFP* e na coordenada das ordenadas a *FVP*, dadas por:

$$FFP = \frac{FP}{VN + FP}$$

e

$$FVP = \frac{VP}{VP + FN}$$

Usualmente na área do diagnóstico médico estas variáveis são também definidas como sensibilidade (ordenada) e 1-especificidade (abscissa).

Existem outras duas frações, relativas aos verdadeiros negativos (*FVN*) e aos falsos negativos (*Fração de Falsos Negativos (FFN)*):

$$FVN = \frac{VN}{VN + FP}$$

e

$$FFN = \frac{FN}{VP + FN}$$

No processo de classificação em diagnóstico médico, a *FFN* e a *FFP* são muito importantes devido a serem erros e consequentemente trazerem graves implicações para o indivíduo. Sob esta perspetiva, um falso negativo terá sempre mais impacto que um falso positivo, pois um “doente” ao ser inferido como um falso negativo não será sujeito a qualquer tipo de tratamento o que pode em casos extremos, levar à sua morte. Já um falso positivo não expressa consequências tão graves, embora o paciente seja sujeito a um grande nível de stress e transtornos pessoais, a longo prazo o paciente terá menos sequelas que um falso negativo, por este necessitar de uma intervenção médica que não lhe foi fornecida.

Na análise *ROC*, caso os verdadeiros positivos e os verdadeiros negativos sejam igual a 1 (100%) diz-se que se está perante um teste perfeito onde não há existência de qualquer tipo de erro (Pepe, 2011).

Segundo Fawcett, no campo da informática, o gráfico da curva *ROC*, tem como principal utilidade o facto de possibilitar organizar e visualizar o desempenho de classificadores (Fawcett, 2003).

Num processo de classificação, tem-se quatro tipos de iterações distintas, caso a ocorrência seja positiva e esteja classificada como positiva estamos perante um *Verdadeiros Positivos (VP)*,

caso seja classificada como negativa aí tem-se um *Falsos Negativos (FN)*. Se a ocorrência é negativa e é classificada como negativa, tem-se um *Verdadeiros Negativos (VN)* e no sentido contrário, se a amostra for classificada como positiva tem-se um *Falsos Positivos (FP)*. Estas quatro classificações formam então a designada matriz de confusão, como apresentado na Tabela 1 (Fawcett, 2003).

Tabela 1: Matriz de confusão 2x2 de classificadores ROC.

		Classe Real	
		Positivo (D)	Negativo (\bar{D})
Classe Prevista	Positivo (T^+)	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Negativo (T^-)	Falso Negativo (FN)	Verdadeiro Negativo (VN)

2.3 INDICADORES DE DESEMPENHO DA CURVA ROC

Para avaliar o desempenho da curva *ROC* é utilizada essencialmente uma medida de precisão, a *AUC* (Hanley and Mcneil, 1982). O valor desta medida está compreendido entre 0 e 1, contudo e visto que a curva *ROC* deverá estar sempre acima da linha diagonal que divide o espaço unitário, que define um quadrado com coordenadas, (0,0) a (1,1), a sua área terá de ser sempre superior a 0,5. Esta medida de avaliação de precisão e desempenho de um diagnóstico médico está diretamente relacionada com o teste de Wilcoxon e Mann-Whitney e também com o coeficiente de Gini. Segundo Hand e Till, o coeficiente de Gini relaciona-se com a *AUC* de acordo com a seguinte formula (Hand and Till, 2001):

$$1 + \text{Gini} = 2 * \text{AUC}$$

Para melhor visualizar e interpretar esta medida de precisão foram gerados 3 gráficos com diferentes valores de *AUC* no *software* R, tal como pode ser verificado na Figura 1, quanto maior o valor de *AUC* mais eficaz será o teste, isto é melhor desempenho será apresentado na classificação.

Estas curvas são atualmente consideradas uma das melhores ferramentas para descrever o desempenho de testes dentro das áreas referidas. A sua utilização desde a segunda guerra mundial tem sido melhorada e impulsionada recentemente pelo uso de *software* (Pepe, 2011).

Devido à grande diversidade de ferramentas disponíveis é de extrema importância perceber a eficácia de cada uma destas, para cada uma das tarefas a realizar na análise da curva *ROC*. Por haver a necessidade de se processar rapidamente dados e avaliá-los, o *software* apresenta-se como uma ferramenta imprescindível, para cumprir as necessidades das unidades responsáveis no processamento de dados e na sua respetiva análise em intervalos de tempo curtos.

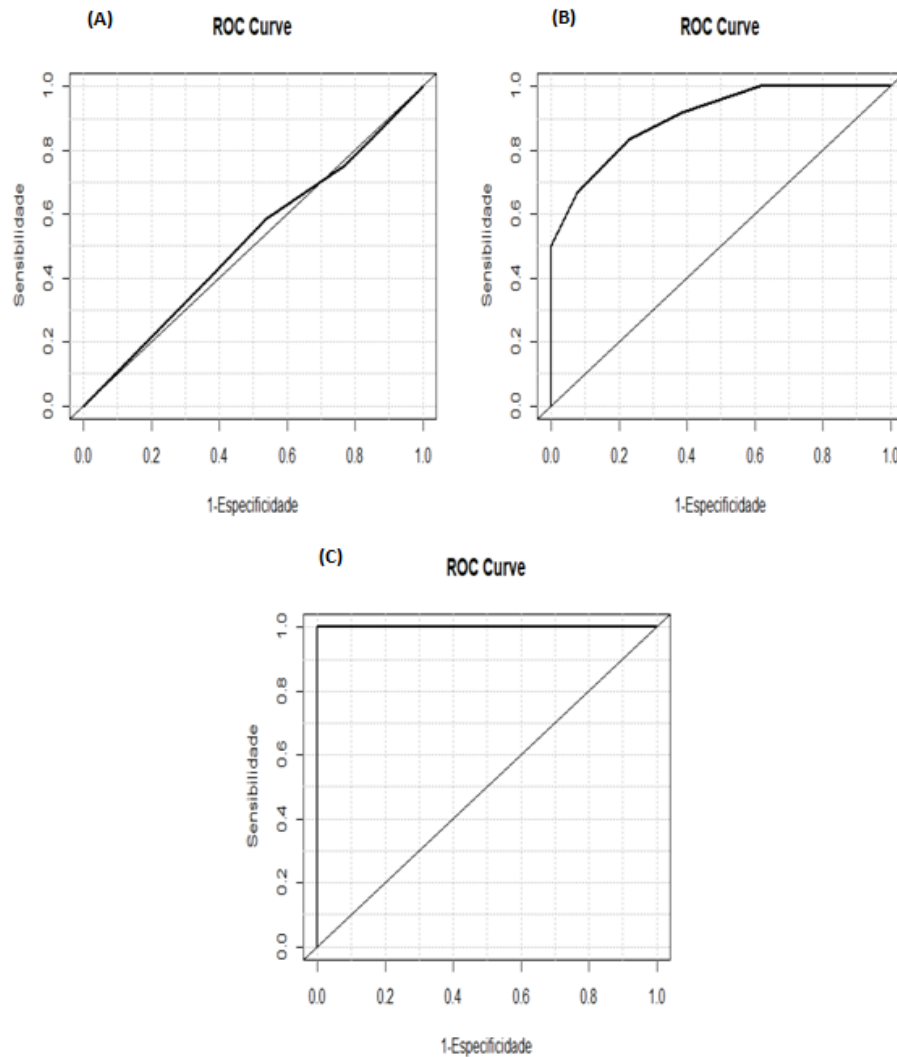


Figura 1: Gráficos bidimensionais de análise ROC com valor de AUC para diferentes níveis de desempenho (A) baixo, (B) médio e (C) alto.

2.4 INDICADORES DE MORTALIDADE NEONATAL

Foram também analisados artigos relativos aos indicadores neonatais que serão utilizados neste trabalho para efetuar a análise através de curvas *ROC*, tanto para variáveis dependentes (relacionadas) como independentes (não relacionadas), de forma a interpretar o significado de cada um deles.

Inicialmente foi explorado um artigo da revista *The Lancet* pertencente à organização *The International Neonatal Network* de 1993. Neste trabalho, é dada ênfase ao indicador *Clinical Risk Index for Babies (CRIB)*, um indicador que se apresentou como um potencial substituto ao peso ao nascer que seria o mais utilizado à altura. Sendo assim, o *CRIB* procurava ser um indicador mais abrangente utilizando para o efeito diversos fatores tais como: peso ao

nascer, idade gestacional, fração máxima e mínima de oxigénio inspirado e presença de deformações congénitas. Para o efeito, neste estudo foi realizado um sistema de pontuação que procurava validar o facto de o *CRIB* ser um indicador viável comparativamente ao peso ao nascer. A conclusão final para *CRIB* é que este era um índice robusto de risco neonatal inicial mais preciso do que o peso ao nascer e simples o suficiente para se utilizar rotineiramente (*The-International-Neonatal-Network, 1993*).

Também neste trabalho é utilizado no sistema de pontuação para comparação de indicadores, o índice *Score for Neonatal Acute Physiology (SNAP)* e *Score for Neonatal Acute Physiology - Perinatal Extension (SNAP-PE)*, ambos são mencionados como contendo um grande número de variáveis que condicionam o seu valor final, sendo por isso, considerados neste estudo indicadores mais úteis para pesquisa do que para utilização de rotina. Por fim são apresentados valores de *AUC* para análise *ROC* que pretendem comparar a precisão dos diversos indicadores, sendo que, o *CRIB* juntamente com o *SNAP* e o *SNAP-PE* apresentam valores mais satisfatórios que aqueles fornecidos pelo peso à nascença, mas o *CRIB*, apresenta a vantagem de ser de mais fácil utilização por apresentar menos variáveis condicionantes para chegar ao seu valor, facilitando o seu uso frequente.

No artigo de Rautonen J. et al. os autores tinham como objetivo comparar o desempenho de três diferentes indicadores, *CRIB*, *SNAP* e *SNAP-PE* para avaliar o risco de mortalidade neonatal. Neste artigo, bem como no anterior para avaliar o desempenho dos diferentes indicadores foram utilizadas curvas *ROC* empíricas que deram uma *AUC* superior para *CRIB* e inferior para os outros dois indicadores, sendo assim, as conclusões apontam para que neste conjunto de indicadores *CRIB* seja o mais apto a ser utilizado, até porque, como referido anteriormente, utiliza um menor número de parâmetros na prática e fornece resultados mais satisfatórios (*Rautonen et al., 1994*).

Num artigo de Courcy-Wheeler et al. de 1995, procura-se justificar o facto do indicador *CRIB* ser potencialmente melhor para a classificação de mortalidade neonatal do que o peso ao nascer. Os resultados não foram conclusivos e embora o *CRIB* tenha sido melhor para a predição de mortalidade bem como a predição de morbilidade, não foi tão bom para a previsão do tempo de vida. Embora os estudos apontassem para que o *CRIB* fosse um melhor indicador que o peso à nascença os resultados deste trabalho não foram suficientes para o definir como um indicador fiável a ser utilizado nas unidades hospitalares (*Courcy-Wheeler et al., 1995*).

2.5 OUTROS ESTUDOS QUE ABORDAM A METODOLOGIA

Os trabalhos de Fawcett explorados neste capítulo remetem a datas mais recentes. Em 2003 e 2006, o autor refere a utilidade das curvas *ROC* em duas áreas particulares e de especial interesse, sendo elas, a mineração de dados e o *machine learning*. Nesses mesmos trabalhos

procurou explicar a complexidade das curvas *ROC*, sugerindo que embora este método aparente ser simples e prático existem algumas questões que não são facilmente perceptíveis, tentando assim esclarecer e guiar os leitores da forma acertada de se processar uma curva *ROC* (Fawcett, 2003, 2006).

O livro de Pepe revelou-se como sendo um trabalho bastante completo e esclarecedor de toda a temática das curvas *ROC* na área médica. Nesta referência são abordados outros assuntos estatísticos como os testes de classificadores binários, análise de regressão, covariáveis e muitas outras questões de interesse relacionadas com a curva *ROC*. Relativamente ao foco do livro, são abordadas temáticas que permitiram consolidar as bases de interesse tanto teóricas como práticas para a análise *ROC*. É dada uma contextualização geral mas específica, são exploradas as suas propriedades matemáticas, abordado o assunto da curva *ROC* binormal, explora-se o índice *AUC*, a análise *ROC* para testes ordinais, análise *ROC* empírica bem como a sua modelação (Pepe, 2011).

O livro de Krzanowski e Hand em 2009 define os conceitos principais ligados à curva *ROC* e apresenta-os de uma forma simples para que possam ser compreendidos por pessoas das mais diferentes áreas. Sendo este livro relativamente recente, apresenta-se bastante atualizado relativamente a toda a temática da curva *ROC* (Krzanowski and Hand, 2009).

A revista *Revstat* (edição especial do INE sobre curvas *ROC*) apresenta artigos do ano de 2014 com uma revisão geral de diferentes conceitos. Nesta revista, são apresentados trabalhos bastantes atualizados e que exploram diferentes questões das curvas *ROC*. No artigo de Gonçalves et al., é feita uma descrição geral dos métodos de estimação das curvas *ROC* que estão constantemente a ser desenvolvidos, é dado um ênfase especial aos métodos frequentistas e ao método Bayesiano (Gonçalves et al., 2014). Num segundo artigo de Pardo-Hernandez et al., é abordada a presença de covariáveis nas curvas *ROC* e a sua respetiva estimação (Pardo-Fernandez et al., 2014). Já num terceiro artigo da autoria de Nakas, são introduzidos alguns novos conceitos, como o *Volume Under Surface (VUS)*, e é explorada a técnica de análise de superfície *ROC*, que corresponde à representação 3D do sistema de classificação (Nakas, 2014). Num último artigo de Alonzo, é abordada a medida "*verification bias*" como forma de avaliar e estimar a precisão de um teste de diagnóstico médico. Em todos eles é feito também um enquadramento dos conceitos básicos relativos às curvas *ROC*, como descrito anteriormente (Todd, 2014).

2.6 software CURVA ROC

Não existem muitos trabalhos relacionados com a comparação de *software* no tema das curvas *ROC*, daí o grande contributo que pode ser dado por esta dissertação.

Entre os existentes, é de realçar o trabalho de Stephan et al., onde foram analisados e comparados oito programas de computador para análise *ROC* (Stephan et al., 2003).

Este artigo remete ao ano de 2003, o que pode justificar o facto de alguns programas se encontrarem inativos ou desatualizados, contudo, é bastante interessante perceber a metodologia adotada pelos autores para realizar este trabalho. Os autores definem três objetivos principais, sendo eles: selecionar, comparar e avaliar diferente *software*. Estes são avaliados consoante cinco critérios delineados pelos próprios autores, nomeadamente, o *input* e o *output* de dados, o conforto, a qualidade do manual e a qualidade dos resultados devolvidos. O *input* refere-se à forma que os dados são inseridos, o *output* à forma que os dados são apresentados e processados, o conforto está relacionado com a compatibilidade com outro *software*, com a presença de tutoriais e se o *software* se encontra devidamente atualizado, já o critério da qualidade dos resultados reflete a integridade e a correção dos mesmos e por último o critério do manual do utilizador onde os autores avaliam a sua qualidade. Posteriormente, a comparação foi efetuada tendo em conta os cinco critérios definidos pelos autores. A cada critério foram atribuídos diferentes *scores* consoante o desempenho das ferramentas e a soma total de *scores* serviu como o termo de comparação. Com este método de análise é possível verificar que o diferente *software* utilizado estava ainda numa fase muito inicial o que levou os autores a concluir que não é possível fazer uma boa análise ROC utilizando apenas uma das ferramentas, embora alguns apresentem um bom desempenho nenhum deles se apresenta como 100% eficaz para as necessidades dos utilizadores.

Existem ainda outras características avaliadas para as oito ferramentas comparadas, entre elas, o número máximo de curvas ROC num gráfico, se é possível fazer a comparação de curvas, se há processamento de dados, se há processamento dos gráficos, se é dada a sensibilidade/especificidade com *Intervalo de Confiança (IC)*'s e o valor médio da *AUC* por *software*. Embora nenhum dos programas se tenha mostrado capaz de efetuar uma adequada análise ROC sem ajuda suplementar de outro *software*, os autores conseguiram fazer uma apreciação global razoável para três das oito ferramentas analisadas. O facto deste trabalho possuir dados bastantes desatualizados fortifica a necessidade de realizar uma nova avaliação para o diferente *software* disponibilizado e utilizado hoje em dia.

Nestes últimos anos existiu uma melhoria substancial da qualidade de todas as ferramentas capazes de realizar a análise ROC e embora em nenhum deles seja obtida uma avaliação perfeita do que é pretendido, têm um desempenho bem mais satisfatório daquele que era alcançado antigamente.

A linguagem de programação R e o seu editor *Rstudio* permitem realizar a análise ROC através dos mais variados *packages*. Inicialmente foram pesquisados e selecionados alguns desses *packages* para perceber a dinâmica da análise ROC dentro deste ambiente de trabalho, nomeadamente o *pROC*, *verification*, *Comp2ROC*, *ROCR*, *ROC (Bioconductor)*, *caTools*, *Epi* e *ROSE*. Em auxílio foram usados os documentos explicativos da plataforma [Comprehensive R Archive Network \(CRAN\)](#) para cada um dos *packages*. Esta é uma ferramenta particularmente

interessante por se apresentar no formato *open source* tornando o seu acesso aberto a toda a comunidade sem qualquer tipo de restrição. Também será explorado outro *software*, o *STATA*, contudo este apresenta desde logo um aspeto negativo por ser necessário adquirir licença para a sua utilização.

Para fortificar o conhecimento da análise *ROC* pelos diferentes *packages* do *R* foram analisados dois artigos.

O trabalho de Cunha e Braga (Cunha and Braga, 2017) realiza uma boa abordagem geral dos aspetos mais importantes para alguns *packages* do *R*, sendo eles: *pROC*, *ROCR*, *verification*, *caTools*, *Comp2ROC*, *ROC (Bioconductor)* e *Epi*. Na secção dos materiais e métodos deste artigo são especificados os comandos gerais para cada tarefa a ser realizada numa análise *ROC* e a forma de se processar os dados. Utilizando os comandos disponibilizados por cada *package* e com um *dataset* comum, realizou-se a respetiva análise obtendo-se valores de *AUC* relativamente semelhantes e tendo todos a capacidade de processar graficamente a curva *ROC*. Posteriormente é apresentada uma tabela que fornece outras características para os diferentes *packages*, entre eles temos: a disponibilidade no *CRAN*, se processa a curva *ROC*, se realiza o processo de *smoothing*, se apresenta o valor de *AUC*, o valor de *AUC* com o *plot*, se fornece os *IC's* e se compara curvas.

Já num outro artigo analisado de Robin et al., é explorado essencialmente o *package pROC*, contudo, são apresentadas algumas características em comparação com outros *packages* como o *smoothing*, *pauc* (partial area under the curve), *IC's*, *plotting* com *IC's*, testes estatísticos e se está disponível no *CRAN*. Os *packages* comparados com *pROC* nestes parâmetros são o *ROCR*, *verification*, *ROC (Bioconductor)* e o *pcvsuite*. Contudo a base do trabalho consiste em abordar especificamente cada função do *pROC*, sendo que na sua globalidade os autores cotaram este *package* com grande eficácia (Robin et al., 2011).

Em relação ao *STATA*, foi analisado o artigo de Pepe et al. de modo a introduzir este *software*. Este trabalho foi publicado em 2009 e são explicados comandos do *software* para executar a análise *ROC*, bem como, qual a capacidade do *software* para realizar determinadas funções. São então mencionadas algumas características desta ferramenta como: a comparação de curvas ser realizada pela *AUC* ou pela zona parcial sob a curva *ROC*, a análise poder ser ajustada para covariáveis, é possível realizar regressão, podem ser realizadas comparações pontuais entre curvas *ROC* e é possível calcular estimadores paramétricos, não paramétricos e semi-paramétricos. Na íntegra, é explicada toda a metodologia dos comandos deste *software*, que ajustando os seus parâmetros possibilitam a realização de uma análise *ROC* adequada bem como a comparação de diferentes curvas (Pepe et al., 2009).

2.7 PROPOSTA DE *checklist* DE FUNÇÕES

Tendo em conta o trabalho referido anteriormente de Stephan et al. (Stephan et al., 2003), será proposta a criação de uma *checklist* que visa facilitar a comparação dos *software* utilizados nesta dissertação, relativamente a cada uma das funções na análise *ROC*. Os passos a serem cumpridos para se chegar a uma *checklist* final são representados na Figura 2.

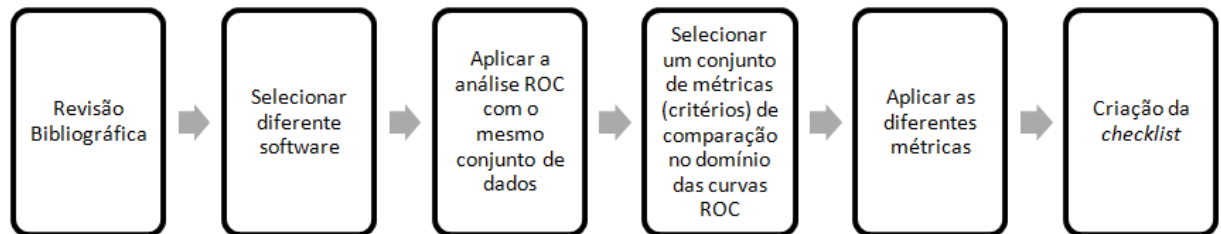


Figura 2: Etapas a serem percorridas até à criação da *checklist* final.

Como se pode verificar inicialmente é necessária uma revisão bibliográfica, nesta etapa é então pretendido não só uma contextualização da temática das curvas *ROC* bem como a familiarização com o *software* e funções disponibilizadas.

De seguida, será selecionado o *software* com base na informação recolhida pela bibliografia e com pequenos testes práticos visando o conhecimento da ferramenta em causa. Ao longo do trabalho novo *software* pode ser acrescentado para fortalecer a componente prática ou retirados caso a sua aplicabilidade não esteja adequada ao que foi inicialmente projetado. Para que a comparação seja feita de forma credível, o *dataset* utilizado ao longo do trabalho terá de ser o mesmo, de forma a que os resultados obtidos sejam o mais próximos possíveis e âmbito de comparação.

A quarta etapa acaba por se sobressair pela importância que não tem sido reconhecida de forma adequada em trabalhos anteriores. Levando em conta o artigo já referido foram utilizados cinco diferentes critérios, sendo eles, o *input* e *output* de dados, o conforto do *software*, a qualidade do manual e a qualidade dos resultados devolvidos. Para esta dissertação a metodologia a ser utilizada será semelhante e os mesmos critérios serão tidos em conta, contudo é necessário para se fazer uma correta comparação acrescentar novos critérios que serão cruciais para se realizar todo o processo de forma adequada. Existem ainda outros aspetos importantes numa análise e esses mesmos deverão estar presentes na *checklist*, entre eles tem-se por exemplo, a possibilidade de se colocar mais do que uma curva numa análise *ROC*, a forma que a *AUC* é devolvida e se é devolvida, a morfologia da curva, se é possível realizar comparação de curvas, os métodos matemáticos utilizados e a acessibilidade ao *software*.

Como é possível verificar a caracterização do *software* deve ser o mais abrangente possível para que se descubra efetivamente em que é que cada um é mais útil e em que situações deve

ser utilizado. Na generalidade o artigo alvo de comparação é limitado não só pelo número reduzido de critérios utilizados, bem como o facto do *software* avaliado estar atualmente desatualizado, inutilizável ou então ser *software* pouco referenciado por outros autores na área das curvas *ROC*.

Esta dissertação pretende então uma renovação de dados relativamente a trabalhos passados de comparação de *software* da análise *ROC* e ser o mais completa possível (utilizando o maior número de *software* explorável), servindo como uma referência para quem utiliza estas ferramentas.

2.8 BASE DE DADOS UTILIZADA

Para realizar a componente prática do trabalho será necessário utilizar um mesmo *dataset* em todas as ferramentas e para o efeito serão utilizados dois conjuntos de dados distintos disponibilizados no formato *.xls* e *.csv*. Estes dois conjuntos de dados diferem por um ser relativo a amostras dependentes e o outro a amostras independentes. Nas Figuras 3 e 4 encontram-se duas ilustrações do aspeto parcial do *dataset* no editor *Excel*.

	A	B	C	D	E	F
1	PESOAG	CRIB	NTISS	SNAP	SNAPPE	result
2	8	6	19	15	15	0
3	9	1	13	22	62	0
4	5	5	22	19	29	0

Figura 3: Formato no editor *Excel* das variáveis dependentes.

	A	B	C	D	E	F	G	H
1	Hospital1	Res1	Hospital2	Res2	Hospital3	Res3	Hospital4	Res4
2	2	0	2	0	1	0	2	0
3	4	0	12	0	8	0	8	0
4	1	0	2	0	0	0	0	0

Figura 4: Formato no editor *Excel* da amostra independente.

Introduzindo ambos os termos de amostras dependentes e independentes existem diferenças de interesse relevante ao realizarmos uma análise *ROC* quanto à sua estrutura de dados. Quando se refere a amostras independentes está-se a falar de um conjunto de dados que não tem qualquer interdependência entre si, ou seja, são dados não correlacionados em que cada indicador/variável tem um resultado independente e as amostras para cada um podem ser de tamanhos diferentes. Relativamente a um plano de amostragem que conduza a um conjunto de dados como o desenvolvido na Figura 3, designado por amostras dependentes, relacionados ou emparelhados, os dados estão correlacionados e têm interdependência entre si no que se refere à obtenção de um resultado final. Tal como se pode verificar na Figura 3 o resultado final é apenas um para cada indicador/variável em estudo. Sendo assim, os cinco indicadores (PESOAG, CRIB, Neonatal Therapeutic Intervention Scoring System (NTISS),

SNAP e *SNAP-PE*) possuem um resultado final (*result*) comum a todos os indicadores para cada linha do *dataset*, isto é, o indivíduo é classificado em cada um dos indicadores para o resultado. No *dataset* com as amostras independentes foram recolhidos valores em quatro hospitais distintos, sendo que, para cada hospital está associado um resultado final que não está relacionado com os restantes.

No exemplo de amostras independentes (Figura 4) são listados os resultados para um classificador *CRIB* em hospitais (*Hospital1*, *Hospital2*, *Hospital3* e *Hospital4*), com valores de 0 ou 1 que correspondem, respetivamente, a vivo e falecido. Sendo que são listados neste *dataset* quatro hospitais, tem-se quatro amostras independentes.

Para o exemplo ilustrado na Figura 4 de amostras dependentes, estão listados valores para diferentes indicadores de mortalidade neonatal. Estes indicadores são: *PESOAG*, *CRIB*, *NTISS*, *SNAP* e *SNAP-PE*. Numa linha do *dataset* todos os valores dos indicadores estão associados a um mesmo resultado, sendo que todos eles funcionam de forma a que os valores superiores correspondam ao estado positivo, à exceção do *PESOAG* que funciona de forma inversa, uma vez que a um peso inferior está associado uma maior taxa de falecimento neonatal.

2.9 APLICAÇÕES *shiny*

De forma a sistematizar todo o trabalho realizado com a criação da lista de atributos (*checklist*) para o uso de *software* foi proposta a criação de uma aplicação *shiny*, esta mesma gerada no próprio ambiente de trabalho R. As aplicações *shiny* são geradas utilizando para o efeito o *package shiny* do *software* R. Introduzindo este mesmo *package* ao nosso trabalho, o produto final projetado é então uma aplicação capaz de fornecer ao utilizador uma simplificação na escolha de um dos *packages* R para o processamento de curvas *ROC*. Antes de começar a criação desta aplicação foi necessário proceder a uma revisão bibliográfica capaz de introduzir esta nova temática.

Num trabalho direcionado ao *Big Data* de Wonhee Cho et al. são explorados diferentes *packages* da linguagem R que possuem a capacidade de gerar funções interativas e de visualização. São descritos neste trabalho alguns aspetos relacionados com o *shiny*, que foi um dos *packages* R comparados relativamente à sua capacidade de gerar aplicações dinâmicas através de funções interativas. É mencionada como sendo uma ferramenta essencialmente utilizada por estatísticos e analistas que tencionam partilhar o seu trabalho *online*, tornando posteriormente as aplicações acessíveis a diferentes entidades como gerentes, clientes, cientistas e qualquer pessoa interessada na aplicação gerada e acessível via *web*. São ainda mencionados alguns aspetos do *shiny* como os seus pontos fortes, fraquezas, eficácia, recurso de visualização e eficiência. Nos pontos fortes é considerado como um *package* capacitado de gerar aplicações bonitas, poderosas e de fácil construção, através das suas

ligações reativas entre *inputs* e *outputs*. O principal ponto fraco será a dificuldade em alterar o modelo da interface. Como recurso de visualização é descrito como tendo a capacidade de combinar o poder computacional do R com a interatividade moderna *web*. No que diz respeito à eficiência é mencionada o facto do conteúdo poder ser visualizado dinamicamente no momento em que o usuário inicia sessão e a sua eficácia é referente à facilidade de criar aplicativos da *web* interativos com o R (Cho et al., 2014).

Tal com descrito no trabalho de Jimmy Doi et al. para criação de uma aplicação *shiny* é apenas necessário o conhecimento de linguagem R, sendo o seu produto final um aplicativo *web* escrito exclusivamente nesta linguagem. Este mesmo autor aborda a temática da educação e refere que por vezes estas aplicações são criadas exclusivamente para simplificar metodologias de ensino, uma vez que se apresenta como uma ferramenta interativa, dinâmica, *user-friendly* e visualmente atraente, sendo também bastante semelhante aos *applets* da linguagem *Java*. São representadas algumas aplicações *shiny* no seu trabalho que permitem perceber a versatilidade deste *package* e é dado também um pequeno tutorial de como começar a trabalhar com esta funcionalidade (Doi, 2016).

No trabalho de Wojciechowski, Hopkins e Upton é abordada a criação de modelos farmacêuticos através das aplicações *shiny*. Os autores definem esta aplicação como tendo a capacidade de revolucionar a partilha e comunicação da simulação de modelos farmacêuticos. Esta aplicação foi selecionada essencialmente por permitir a programadores em R mostrarem interativamente a saída de dados em *web browsers*, ou seja, permite ao utilizador visualizar e trabalhar com a aplicação em qualquer computador com acesso à internet. Neste trabalho são mencionados outros trabalhos criados com o *package shiny* como é o caso de uma ferramenta de educação para dosagem destinada a estudantes do ensino médio e uma ferramenta de simulação completa para o medicamento Ibuprofeno, de forma a validar a utilização deste *package* na área da farmacêutica. Ao longo deste artigo são ainda fornecidas diversas instruções de como aceder ao *shiny* e de como o utilizar (Wojciechowski et al., 2015).

Num trabalho de Jonathan W. Nelson et al., dentro da área da Bioinformática, temos a aplicação START criada para realizar sequenciamentos de *Ácido ribonucleico (ARN)*. Esta aplicação foi escrita em linguagem R e devolvida pelo *package shiny* no formato de uma aplicação. À semelhança das restantes aplicações abordadas o facto de ser possível aceder à mesma através de qualquer computador com acesso à internet é apontado como um fator bastante positivo pelos autores. Com o START é possível aceder a diferente informação no sequenciamento de *ARN*, como por exemplo, o *clustering* através de *Principal Component Analysis (PCA) plots* e *heatmaps*, comparação de diferentes grupos de sequenciamento por *scatterplots* ou *volcanoplots* e ainda a criação de *boxplots* relativas ao nível de expressão dos genes. A aplicação está disponível *online* na plataforma do *shiny* desde 2016, embora, inicialmente seja mencionado no próprio artigo que estaria apenas dois anos acessível por este meio (Nelson et al., 2017).

Foi ainda analisado um artigo de Ebrahim Jahanshiri e Abdul Shariffna na área da agricultura de precisão, onde a aplicação *shiny* do R apareceu como um substituto à linguagem python (Jahanshiri and Shariff, 2014). Estes autores referem que o python não apresenta um sistema dedicado inteiramente à análise estatística de dados geográficos e o facto do R estar em formato *open-source*, com uma grande popularidade e num momento em que está a ser introduzido o *package shiny* à comunidade, levou a que os autores aprofundassem o seu trabalho de processamento geográfico na *web*, utilizado para o efeito este mesmo *package*. Sendo assim é demonstrado ao longo deste trabalho a possibilidade de ser criada uma aplicação *shiny* para mapeamento e análise estatística relativa a dados agrícolas. Embora este *package* seja apontado como uma ótima solução para satisfazer as necessidades de um programador, pelo facto de estar escrito em R onde estão disponibilizadas uma enorme quantidade de métodos para mapeamento e análise estatística de dados espaciais são também apontadas algumas fragilidades/desafios a este *package* por este mesmo ser relativamente recente. Como desafios são mencionados aspetos como as transformações de sistemas de coordenadas, dados imprecisos ou ausentes, a complexidade de alguns algoritmos e a existência de diferentes formatos de dados.

No artigo de Michaela Spitzer et al. aborda-se a criação de uma aplicação para geração rápida de *boxplots*, através da importação de uma matriz de dados definida pelo utilizador ou colada diretamente no aplicativo para que seja assim devolvida uma *boxplot* básica com algumas opções de recursos adicionais. No final pode ser atribuído um título aos gráficos, personalizar a sua cor, dimensões e orientação para que esteja apto a ser extraído num dos formatos disponibilizados (.eps, .pdf ou .svg). Uma aplicação bastante simples e prática para qualquer usuário que necessite de gerar um *boxplot* do seu conjunto de dados de forma rápida e eficaz (Spitzer et al., 2014).

O trabalho de Wang, Hallow e James apresenta o *package RxODE* que através do *shiny* é capaz de demonstrar grande utilidade na área da farmacologia. Este *package* facilita a simulação rápida de equações diferenciais completamente dentro do R e para facilitar a avaliação e interatividade, estes modelos são traduzidos diretamente em *shiny*. São ainda mencionados aspetos positivos relativos ao ambiente de trabalho R e ao *package shiny*. Para o R o facto de apresentar diversos meios de analisar e visualizar resultados de simulações (p.e *package ggplot*) e para *shiny* a capacidade de os expor *online* via *web*, tornando este o meio ideal para realizar simulações farmacológicas em tempo real (Wang et al., 2015).

SOFTWARE / FERRAMENTAS / PACKAGES PARA ANÁLISE ROC

Neste capítulo será abordada a componente prática relativa à seleção do diferente *software*, ferramentas e *packages* utilizados para processar curvas *ROC*. Os testes práticos serão realizados utilizando para o efeito o mesmo *dataset*, possibilitando assim que seja realizada uma comparação coerente e correta. Serão utilizadas as métricas referidas no Capítulo 2 que possibilitarão a elaboração de uma lista de atributos (*checklist*) para ajudar na tomada de decisão sobre o *software*/ferramenta mais adaptado para cada especificidade e qual o melhor ou pior na generalidade das suas funcionalidades.

3.1 *Packages* R

Inicialmente foi utilizado o ambiente de trabalho R no editor *Rstudio*. Foram recolhidos oito *packages* já com alguma cotação e presente em diferente bibliografia.

O ambiente de trabalho R é utilizada essencialmente na área da estatística e permite processar gráficos. A sua linguagem de programação é derivada da linguagem S e a sua sintaxe é semelhante à linguagem C (Team, 2019). A sua semântica na realidade tem uma estrutura da linguagem de programação funcional. Apresenta-se de uma forma simples uma vez que é possível na sua grande maioria a partir da linha de comando com expressões simples chegar ao que o utilizador procura, contudo, também permite gerar funções de raiz caso o utilizador necessite de implementar novas funcionalidades.

Uma vez que já existe diverso trabalho na área das curvas *ROC* neste ambiente trabalho, a sua utilização resume-se essencialmente à utilização de comandos pré-definidos em diferentes *packages* que possuem diferentes funcionalidades.

3.1.1 *pROC*

O *pROC* (Robin et al., 2011) é um *package* desenvolvido em R que se encontra na versão 1.16.1 e cuja data da última atualização remonta a 14 de janeiro de 2020 estando disponível toda a informação do *package* na plataforma *CRAN*. Com cerca de um autor, atualmente, Xavier

Robin é o único responsável pela manutenção do *package* atualizado. Apresenta algumas dependências de outros *packages* como é o caso do *methods*, *plyr* e *Rcpp*.

A função principal deste *package* é o comando `roc()`. Está direcionado a realizar automaticamente uma análise *ROC*, sendo que, caso solicitado poderá realizar *smoothing* (`smooth=TRUE`), calcular *AUC* (`auc= TRUE`), calcular *IC's* (`ci=TRUE`) e processar o gráfico com a curva (`plot=TRUE`), para tal, como indicado anteriormente entre parênteses será necessário incluir os parâmetros lógicos (*TRUE/FALSE*) no comando principal. Existe ainda a possibilidade de processar separadamente estes parâmetros através de comandos individuais, como é o caso do comando `plot.roc()` que traça a curva *ROC*. Este *package* permite ao utilizador processar um gráfico com mais do que uma curva *ROC*, para o efeito, simplesmente será necessário colocar o parâmetro `add` com o valor lógico *TRUE* no comando da curva *ROC* que deseja adicionar ao gráfico previamente processado. Para diferenciação das curvas, o utilizador poderá adicionar o parâmetro `col` para definir uma cor específica para cada uma das curvas. Visualmente apresenta alguns defeitos, a abcissa tem os seus valores apresentados em um espaço limitado tornando a curva *ROC* visualmente pouco atrativa e apresenta os valores de abcissa de $[1,0]$ em vez de $[0,1]$, utilizando assim a variável especificidade em vez de 1-especificidade que é o normal em análises *ROC*. Na Figura 5 está ilustrado o aspeto que tomam os gráficos gerados por este *package*.

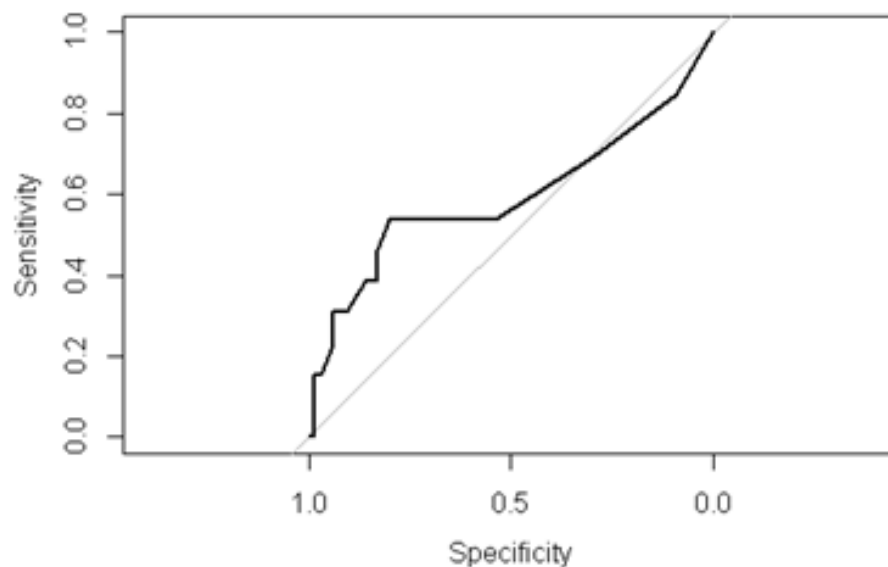


Figura 5: Curva ROC apresentada utilizando o *package pROC*.

Este *package* apresenta ainda uma outra função (`ggroc()`) capaz de produzir gráficos *ROC*, através do *package ggplot2*. Os gráficos processados por esta função apresentam-se mais apelativos uma vez que os valores da abcissa estão colocados normalmente no seu espaço, já o problema da utilização dos valores $[1,0]$ em vez de $[0,1]$ na abcissa mantém-se, utilizando

assim a variável especificidade em vez de 1-especificidade. Na Figura 6 é representado um exemplo da utilização desta função e o aspeto que a mesma toma.

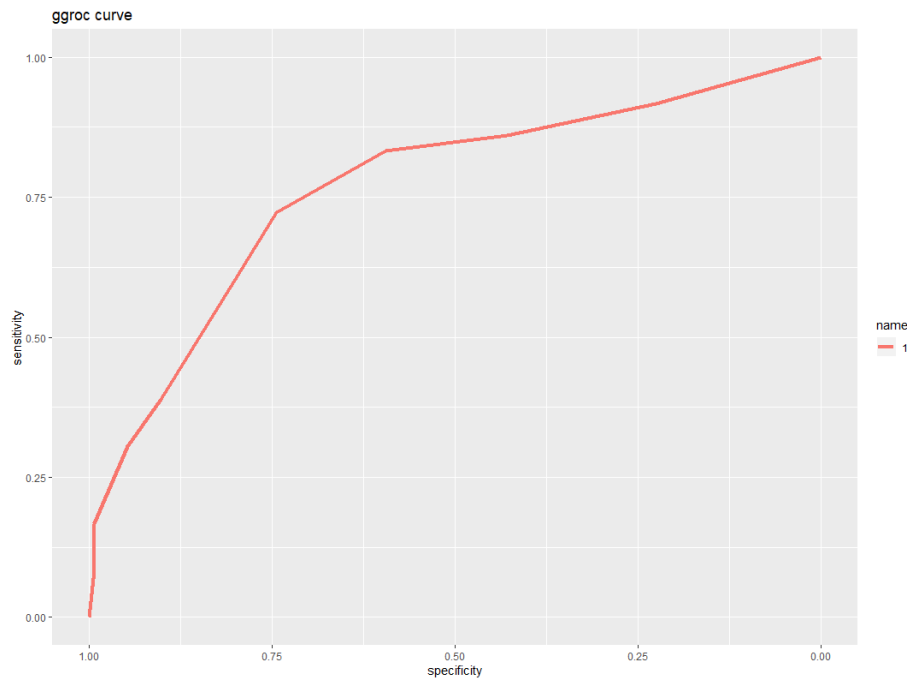


Figura 6: Curva ROC apresentada utilizando a função `ggroc()` do *package* `pROC`.

Posteriormente, será feita uma revisão da utilização deste *package* nos *datasets* deste estudo. Inicialmente foi necessário proceder à instalação do *package* e ao seu carregamento através das seguintes linha de comando:

```
install.packages("pROC") e library(pROC)
```

Relativamente às variáveis dependentes o indicador PESOAG funciona de forma inversa às restantes variáveis, uma vez que, quanto menor o seu valor maior é a probabilidade associada ao valor positivo da escala (1)-morte. Quando traçada a curva ROC a partir deste *package*, para a variável PESOAG, foi possível averiguar que automaticamente a curva surgia exatamente no mesmo formato das restantes variáveis/indicadores, ou seja, com *AUC* superior a 0.5, o que não era expectável, visto que os valores funcionam de forma inversa. Este acontecimento deve-se ao facto do *pROC* possuir um parâmetro capaz de automaticamente definir a direção da variável, utilizando para o efeito a mediana mais alta. Caso o utilizador pretenda alterar o formato da curva pode então modificar o parâmetro *default* de *direction* dentro da função *roc* com duas distintas opções, sendo elas:

```
direction = "<" e direction = ">"
```

De forma a demonstrar mais facilmente o assunto abordado é demonstrada a linha de comandos (Figura 7) e os respetivos gráficos das curvas ROC (Figura 8) traçadas para duas

variáveis que funcionam de forma distinta, neste caso o PESOAG (onde a escala funciona de forma invertida) e o *CRIB* (onde a escala funciona normalmente).

```
17 roc1= roc(DI$result,DI$PESOAG,plot=T,ci=TRUE,direction = "<")
18 roc2= roc(DI$result,DI$PESOAG,plot=T,ci=TRUE,direction = ">")
19 roc3= roc(DI$result,DI$CRIB,plot=T,ci=TRUE,direction = "<")
20 roc4= roc(DI$result,DI$CRIB,plot=T,ci=TRUE,direction = ">")
```

Figura 7: Linha de comandos utilizada para gerar gráficos ROC com diferente *direction*.

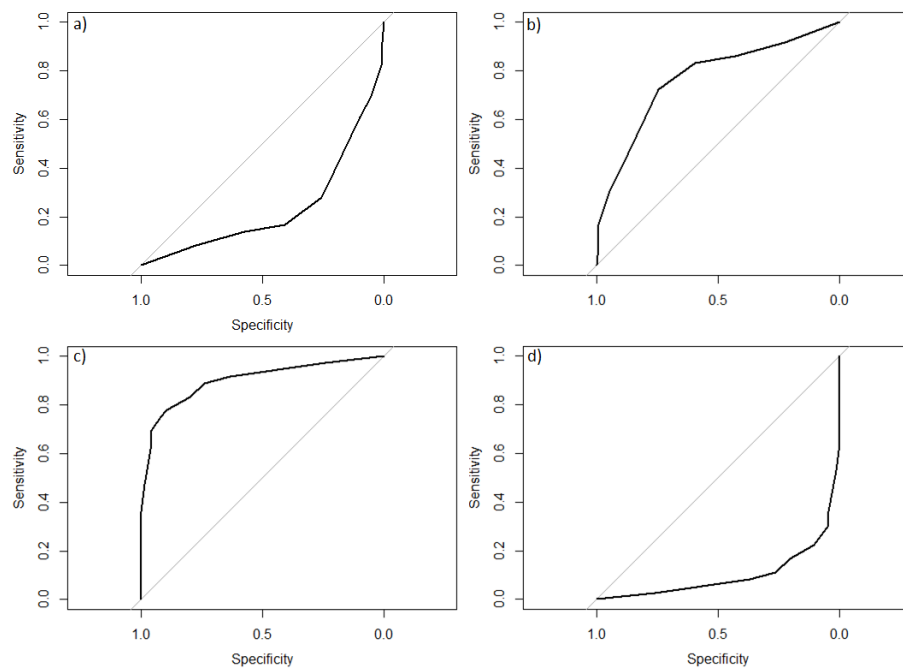


Figura 8: Gráficos gerados a partir da linha de comandos da Figura 7, gerando para o efeito o gráfico a, b, c e d que correspondem às variáveis roc1, roc2, roc3 e roc4, respetivamente.

Contudo, estas curvas só permitem uma correta análise quando estão acima da linha $x=y$ tomando valores de *AUC* superiores a 0.5 e para tal é necessário que o valor de *direction* esteja a tomar os valores *default*. No âmbito geral podemos considerar este *package* de fácil utilização e com ferramentas de apoio suficientes para a sua implementação. Contudo é de assinalar que a curva *ROC* é apresentada num gráfico onde a coordenada das abcissas é apresentada de forma invertida, com valores a variar de 1 a 0 e num curto espaço, tornando a curva esteticamente um pouco desformatada e de mais difícil análise. Nas restantes funcionalidades para análise *ROC* e nas diferentes vertentes apresenta-se como um *package* bastante competente e completo.

Visto que o *package pROC* se encontra unicamente focado na análise *ROC* é apresentada uma tabela final (Tabela 2) com uma breve descrição para cada comando, fornecido pelo documento do *CRAN*.

Comparação de duas curvas ROC:

Tabela 2: Descrição dada pelo CRAN para as funções do *package pROC*.

Comando	Descrição
roc	Construir uma curva ROC
are.paired	Determinar se duas curvas ROC são emparelhadas
auc	Calcular a área abaixo da curva ROC
ci	Calcular intervalos de confiança de uma curva ROC
ci.auc	Calcular o IC da AUC
ci.coords	Calcular o IC de coordenadas arbitrárias
ci.se	Calcular o IC de sensibilidades em determinadas especificidades
ci.sp	Calcular o IC de especificidades em determinadas sensibilidades
ci.thresholds	Calcular o IC de especificidade e sensibilidade dos valores de corte
coords	Coordenadas da curva ROC
cov	Covariância entre duas AUC's
ggroc	Traçar uma curva com <i>ggplot2</i>
has.partial.auc	Determinar se a curva ROC tem uma AUC parcial
lines.roc	Adicionar uma linha ROC a um gráfico ROC
plot.ci	Adicionar intervalos de confiança a um gráfico de curva ROC
plot	Processar uma curva ROC
print	Imprimir um objecto da curva ROC
roc.test	Comparar a AUC de duas curvas ROC
smooth	Ajustar uma curva ROC
var	Variância da AUC

Para realizar a comparação de duas curvas *ROC* é utilizada a função *roc.test()*. Não são colocadas no mesmo gráfico, apenas se comparam parâmetros sendo demonstrado na Figura 9 um exemplo prático para esta função, com a respetiva linha de comandos e *output*.

```
92 roc1= roc(DI$result,DI$PESOAG,plot=T,ci=TRUE,direction = ">")
93 roc2= roc(DI$result,DI$CRIB,plot=T,ci=TRUE,direction = "<")
94 roc.test(roc2,roc3)
```

Output:

```
Bootstrap test for two correlated ROC curves

data: roc2 and roc3
D = -2.1912, boot.n = 2000, boot.stratified = 1, p-value = 0.02844
alternative hypothesis: true difference in AUC is not equal to 0
sample estimates:
AUC of roc1 AUC of roc2
0.7680660 0.8994361
```

Figura 9: Comparação de duas análises ROC através do comando *roc.test()* do *package pROC*.

Caso o utilizador deseje visualizar as duas curvas *ROC* num mesmo gráfico pode utilizar para o efeito o parâmetro *add=T* na função *roc()*.

3.1.2 ROCR

O *ROCR* (Sing et al., 2005) é um *package* desenvolvido em R está atualmente na versão 1.0.7 sendo que a última atualização é referente à data de 26 de março de 2015. Toda a sua informação está disponível no [CRAN](#). Os seus autores são Tobias Sing, Oliver Sander, Niko Beerenwinkel e Thomas Lengauer, sendo que o responsável pela sua manutenção é Tobias Sing. Está dependente de outros dois *packages*, o *methods* e o *gplots*. É um dos *packages* mais antigos na área e apresenta uma metodologia distinta dos restantes apresentados neste trabalho.

Este é então um *package* flexível que apresenta mais de 25 medidas de desempenho que combinadas podem criar curvas de desempenho bidimensionais. De acordo com o manual depositado no [CRAN](#), este *package* consegue reconhecer algumas medidas de desempenho distintas tais como: precisão, taxa de erro, taxa de verdadeiros positivos, taxa de falsos positivos, taxa de verdadeiros negativos, taxa de falsos negativos, sensibilidade, especificidade, *AUC* e ainda muitas outras medidas que não são de relevância para o estudo em causa. Está indicado para gerar gráficos *ROC*, curvas de sensibilidade/especificidade e gráficos de *precision/recall*. Tal como se irá verificar posteriormente, para trabalhar com este *package* na área das curvas *ROC* são necessários três comandos que funcionalmente são totalmente distintos daqueles utilizados por outros *packages* da área dentro do R. Para simplificar a sua utilização é apresentada a Figura 10, que procura facilitar a explicação dos comandos.

```

13 library(ROCR)
14
15 pred = prediction(-DI$PESOAG,DI$result)
16
17 roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
18
19 plot(roc.perf, lwd=3)
20 abline(a=0, b= 1)
21
22 auc.perf = performance(pred, measure = "auc")
23 auc.perf@y.values
24

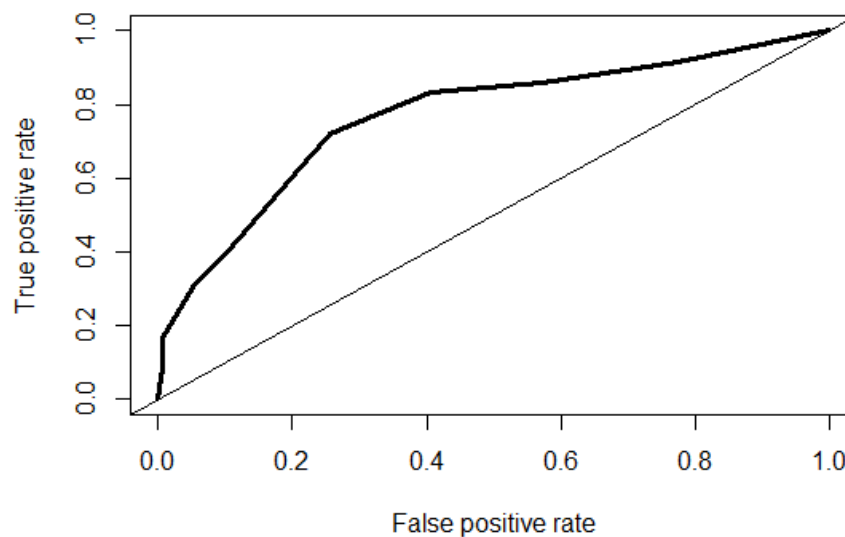
```

Figura 10: Exemplo do código necessário à realização da análise ROC com o *package* *ROCR*.

Relativamente à linha 15 da Figura 10, onde é utilizado o comando *prediction()*, o mesmo serve para transformar o *dataset* anteriormente carregado num formato standarizado, por isso o utilizador deverá passar como parâmetros, em primeiro a sua *prediction* que corresponde ao valor da variável dependente ou independente e em segundo o seu *result* que corresponde a um valor binário associado ao estado de morto ou vivo.

Após a criação da variável *pred* e esta estar num formato standarizado, será passada como parâmetro a uma nova função, a *performance()*. São então fornecidos os valores necessários para a análise *ROC*, como a *FVP* e *FFP*.

Nesta etapa, embora toda a estrutura da análise *ROC* esteja criada é necessário comandos que devolvam o *output* desejado. Para o efeito existem então dois comandos, o *plot()* que irá gerar o gráfico *ROC* e, novamente o *performance()* mas desta vez com o parâmetro *measure = "auc"* para gerar o valor da *AUC* da análise *ROC*. Relativamente ao comando *plot()* é possível adicionar mais que uma curva caso o utilizador deseje, colocando para o efeito o parâmetro *add=T* na função. Já para a *AUC*, após a criação da variável, o seu valor de área estará no objeto *y.values*. Um exemplo de gráfico e o respetivo *output* da *AUC* no terminal são representados na Figura 11.



```
AUC:
> auc.pert@y.values
[[1]]
[1] 0.768066
```

Figura 11: *Output* fornecido pelos comandos do *package ROCR*.

Relativamente à análise dos *datasets* utilizados para realizar a avaliação do *package*, tanto para variáveis dependentes como para variáveis independentes esta ferramenta não apresentou complicações de maior. Ainda assim e visto que este *package* não apresenta qualquer mecanismo que possibilite a utilização de escalas invertidas, foi necessário a conversão dos valores do PESOAG para valores negativos para que, como referido anteriormente, este indicador funciona de forma inversa aos restantes, gerando por consequência curvas *ROC* que estão na parte inferior da linha $x=y$ e com *AUC* inferior a 0.5. Para utilizadores menos familiarizados com o R, poderá haver alguma dificuldade em gerar este tipo de curvas, contudo e com a leitura do manual disponível no *CRAN* a sua utilização torna-se acessível a toda a comunidade.

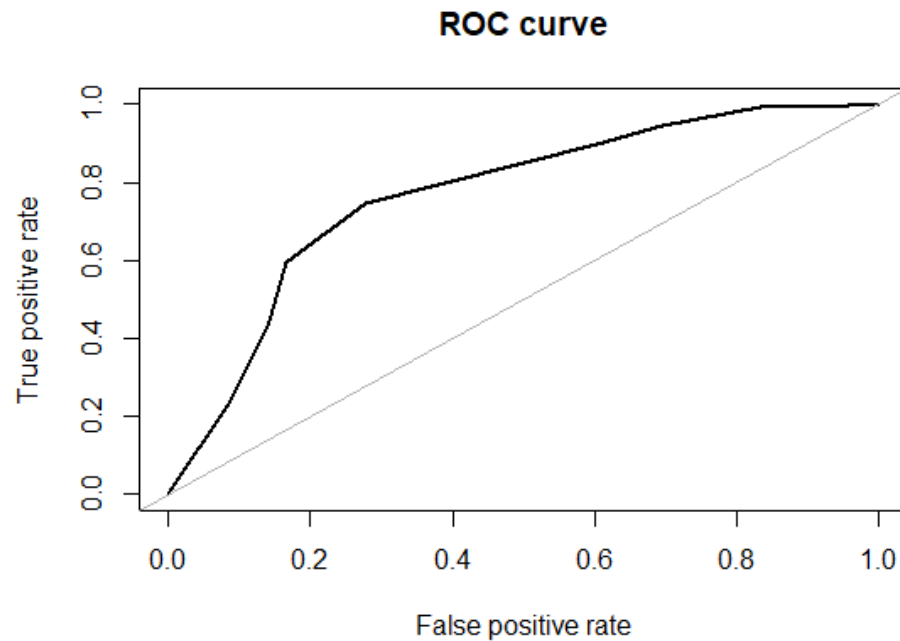
3.1.3 ROSE

O *package* ROSE (Lunardon et al., 2014) também se encontra disponível na plataforma do CRAN e a última atualização está descrita na data de 19 de fevereiro de 2015, estando apenas na versão 0.0.3. Desta informação depreende-se que sofreu poucas atualizações quer pela baixa versão do *package*, bem como por apresentar uma última data de atualização antiga. Os autores são Nicola Lunardon, Giovanna Menardi e Nicola Torelli, sendo o primeiro o responsável pela sua manutenção. À semelhança de outros *packages* apresenta dependências, neste caso dos *packages*: MASS, nnet, rpart e tree. Embora não seja descrita na sua página do CRAN foi encontrada algumas dependências ao *package* pROC num código disponibilizado na plataforma GitHub.

A descrição dada pelo CRAN indica que este *package* não está direcionado essencialmente para a análise ROC. São fornecidas então funções para lidar com problemas de classificação binária na presença de classes não equilibradas. ROSE é uma técnica baseada em *bootstrap* funcionando como um auxílio na classificação binária. O comando de interesse para avaliar as métricas da análise ROC neste *package* é o `roc.curve()`.

Passando à utilização do *package* na análise ROC, para o comando `roc.curve()` é necessário apenas fornecer o vetor das *prediction* e das *response* sendo de imediato devolvido o seu *output*, ou seja, a reprodução do gráfico desejado e o respetivo valor de AUC para avaliar a precisão do teste. Como se pode verificar na Figura 12 visualmente o gráfico apresenta-se bem estruturado e o valor da AUC é devolvido no terminal com três casas decimais.

No que diz respeito à análise do *package*, foram encontrados alguns obstáculos que não apresentam qualquer tipo de justificação bibliográfica e prática para a sua ocorrência, contudo após consultar o código do *package* na plataforma GitHub foi possível verificar que o *package* pROC apresenta-se mencionado em algumas partes, o que poderá significar certas dependências a este mesmo *package*. À semelhança dos *packages* pROC e caTools este processa a curva ROC para o indicador PESOAG de forma já invertida, ou seja, já colocada acima da linha $x=y$ e com AUC superior a 0.5. Como referido anteriormente, visto que o *package* pROC aparece mencionado no seu código, possivelmente, o ROSE utilizará o mesmo método de trabalhar com escaladas invertidas, utilizando para o efeito as medianas mais altas. Embora este *package* seja fácil de utilizar é bastante limitado relativamente a uma análise ROC mais ampla, uma vez que apenas faz o processamento do gráfico ROC e calcula da sua respetiva AUC (com 3 casas decimais), não processando mais dados de interesse para uma análise ROC mais completa.



AUC:

```
> roc.curve(DI$result,DI$PESOAG)
Area under the curve (AUC): 0.768
```

Figura 12: Output do package ROSE, AUC no terminal e respetivo gráfico.

3.1.4 ROC do Bioconductor

O package ROC do Bioconductor (Carey and enhancements, 2019) não está disponível no CRAN uma vez que, a própria plataforma do Bioconductor disponibiliza a informação sobre o mesmo. A sua ultima atualização remete à data de 23 de janeiro de 2020 estando atualmente na versão 1.62.0. Pelos dados adquiridos, é possível verificar que este é um package que tem vindo a ser trabalhado recentemente e continuamente. Os seus autores são Vince Carey e Henning Redestig, sendo Vince Carey o responsável pela sua manutenção. O ROC do Bioconductor apresenta dependências ao *utils* e *methods*.

São disponibilizados pelo Bioconductor dois documentos distintos, o *ROC notes* onde é dada uma breve descrição do package, uma introdução a alguns termos da análise ROC bem como exemplos demonstrativos do package. E o documento *Reference Manual* onde estão descritos os comandos bem como os seus parâmetros, um pouco à semelhança daquilo que é feito pelo CRAN. Também é possível aceder na plataforma ao *script* dos exemplos do ficheiro *ROC notes*, possibilitando aos utilizadores visualizar no seu próprio editor a dinâmica de todo o package.

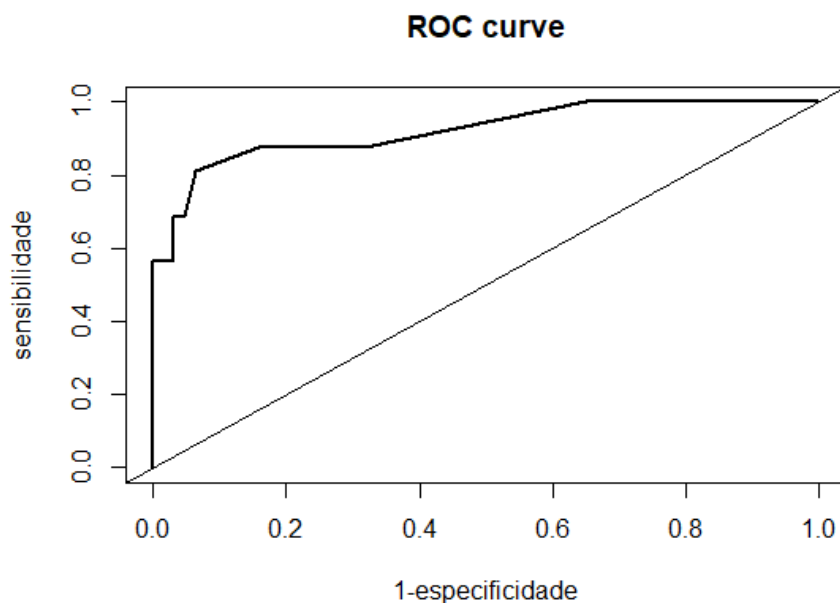
Para se processar a análise *ROC* através do *ROC* do *Bioconductor*, é necessário utilizar o comando principal *rocdemo.sca()*, que irá organizar todos os valores em sensibilidade e especificidade e posteriormente com o comando *AUC()* é possível calcular o valor final da área abaixo da curva. Contudo para se realizar a representação do gráfico dos valores obtidos não existe qualquer comando associado exclusivamente ao *package*, sendo necessário para o efeito utilizar o comando *plot()* do próprio R. É ainda possível com o comando *pAUC()* calcular áreas parciais da curva. Nas Figuras 13 e 14 é apresentado um exemplo da linha de comandos e o seu *output*, respetivamente.

```

46 roc1 = rocdemo.sca(DH$Res4,DH$Hospital4)
47 plot(roc1,lwd=2, main="ROC curve", xlab="1-especificidade", ylab="sensibilidade")
48 abline(0,1)
49
50
51 auc = AUC(roc2)
52 auc
53
54 paucp4 = pAUC(roc1,.5)
55 paucp4

```

Figura 13: Linha de comandos para processar o *output* para o *package* *ROC(Bioconductor)*.



```

AUC e pAUC:
> auc = AUC(roc1)
> auc
[1] 0.921627
> paucp4 = pAUC(roc1,.5)
> paucp4
[1] 0.4196429

```

Figura 14: *Output* do *package* *ROC(Bioconductor)*, AUC e *partial Area Under the Curve* (pAUC) no terminal e respetivo gráfico.

Relativamente à análise dos *datasets*, o *package* apenas apresenta alguma fragilidade em lidar com amostras independentes, uma vez que na sua estrutura as variáveis apresentam dimensões distintas. Sendo assim, serão geradas automaticamente linhas com valores de **Not Available (NA)** que entram em conflito com os comandos do *package*. Não sendo propriamente uma restrição, acaba por tornar a realização da análise *ROC* com este *package* um pouco mais demorada, visto que o utilizador é obrigado a criar vetores próprios com cada uma das variáveis ou a então a fragmentar o *dataset* para conseguir proceder à respetiva análise *ROC*. Relativamente às variáveis invertidas, o *package* não apresenta nenhuma funcionalidade para lidar com estas (PESOAG), sendo que, foi necessário proceder à sua inversão com a substituição por valores negativos, possibilitando assim gerar uma curva com *AUC* superior a 0.5.

Na generalidade este é também um *package* de fácil utilização, sendo que, o comando principal gera apenas os valores da sensibilidade e especificidade havendo a necessidade de posteriormente serem gerados os valores de *AUC* e *pAUC* com comandos do *package*. Com um comando externo ao *package*, o *plot()*, é possível gerar o gráfico para análise. Este é então um *package* um pouco limitado para realizar uma análise *ROC* mais detalhada com diferentes valores de saída associados à análise *ROC*, contudo, apresenta-se de fácil utilização e capaz de gerar um gráfico *ROC* e o respetivo cálculo da *AUC*.

3.1.5 *verification*

O *verification* (NCAR, 2015) encontra-se disponível no *CRAN* e a sua última atualização está datada a 15 de Julho de 2015 com a versão 1.42. Embora tenha sofrido provavelmente diversas atualizações, à semelhança de outros *packages* já não tem sido atualizado há mais de quatro anos. Foi criado por um laboratório de aplicações, o *NCAR - Research Applications Laboratory* e neste momento a sua manutenção é efetuada por Eric Gilleland. Apresenta algumas dependências de *packages* como é o caso do *methods*, *fields*, *boot*, *CircStats*, *MASS* e *dtw*.

Este *package* está focado em verificar previsões para variáveis discretas, contínuas, probabilísticas e expressas como distribuições paramétricas. Para trabalhar diretamente com análise *ROC* são disponibilizados dois comandos, sendo eles o *roc.plot()* e o *roc.area()*

No comando *roc.plot()* é possível calcular também os *IC's*, contudo este *package* para realizar o cálculo dos *IC's* precisa dos valores relativos às *predictions* em percentual o que diverge com o conjunto de dados utilizados nesta dissertação. É recomendado pelo *CRAN* utilizar outros *packages* R que auxiliem para realizar uma análise *ROC* mais abrangente, sendo que o *verification* está cotado como um dos melhores para representação gráfica *ROC*. O facto de este não ser um *package* que detalhe muitas medidas estatísticas ao nível da análise *ROC* pode ser justificado pelo facto de o mesmo não ter sido criado apenas para esta

área estatística. Contudo os dois comandos processam o gráfico *ROC* e calculam a respetiva *AUC*.

Para os *datasets* em estudo, apenas não apresenta qualquer tipo de funcionalidade para inverter a escala do PESOAG, sendo então necessário a conversão deste indicador para valores negativos de forma manual. Para as amostras independentes não foram detetados quaisquer problemas.

Resumindo, os dois comandos associados à análise *ROC* no *verification* processam o gráfico *ROC* e calculam o valor da área sendo também fornecido o *probability value* (*p-value*) e alguns dados como o número de amostras total e quantas correspondem ao estado definido como zero e um. Os valores dos *IC's* não foram calculados para este *dataset* uma vez que as métricas e os parâmetros utilizados neste *package* entram em conflito com o mesmo e o seu documento explicativo não é completamente esclarecedor relativamente a esta funcionalidade. Nas Figuras 15 e 16 são apresentadas a linha de comando para o *package* e o seu respetivo *output* relativo ao processamento do gráfico e aos valores devolvidos no terminal para um dos exemplos testados.

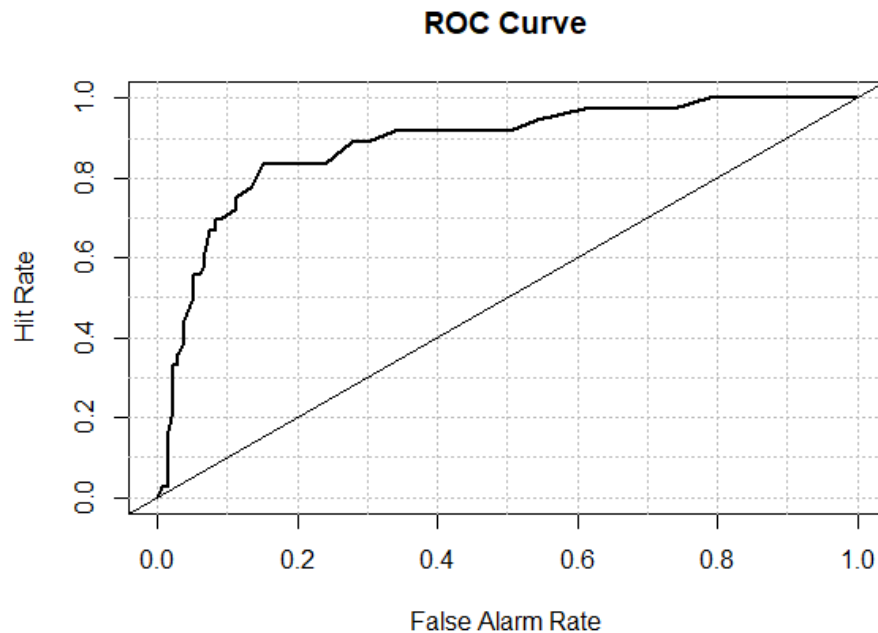
```
25 roc.plot(DI$result, DI$SNAPPE)
26 roc.area(DI$result, DI$SNAPPE)
```

Figura 15: Exemplo para uma linha de comandos para testar o *package verification*.

3.1.6 *Epi*

O *package Epi* (Carstensen et al., 2019) encontra-se atualmente na versão 2.40, sendo que a última atualização decorreu no dia 25 de novembro de 2019. Foi desenvolvido por Bendix Carstensen, Martyn Plummer, Esa Laara e Michael Hills, sendo Bendix Carstensen o responsável pela manutenção do mesmo. Apenas depende do *package utils* segundo referência do *CRAN*. Este *package* possui funções para análise demográfica e epidemiológica com um diagrama Lexis, ou seja, registo e corte de dados de acompanhamento, em particular representação, manipulação e simulação de dados. O comando focado nesta dissertação passa pela função *ROC()* do *package* que permite o calculo de sensibilidades, especificidades e valores preditivos positivos e negativos. O *output* da função é devolvido dentro de um gráfico com uma curva *ROC* traçada e os valores descritos anteriormente bem como a sua *AUC*, sendo assim, não há valores de *output* devolvidos no terminal forçando o utilizador a retirar manualmente os valores do gráfico gerado.

À semelhança do *package verification*, este não está exclusivamente indicado para análise *ROC*, tornando-o pouco específico nesta área. Contudo, apresenta um *output* simples e eficaz no gráfico *ROC*. Para verificar o formato que o *output* é devolvido é apresentada a Figura 17.



Output terminal:

```
> roc.area(DI$result, DI$SNAPPE)
$A
[1] 0.8827277

$N.total
[1] 169

$N.events
[1] 36

$N.noevents
[1] 133

$P.value
[1] 9.682833e-13
```

Figura 16: Exemplo do *output* gerado pelo *package verification*.

De salientar que os valores de sensibilidade, especificidade e *AUC* poderiam ser devolvidos também no terminal para que o utilizador conseguisse aceder mais facilmente a estes valores.

Na utilização do *dataset* desta dissertação o *package* não apresentou problemas e a sua utilização foi bastante acessível. Contudo, a comparação de diferentes curvas *ROC* não é disponibilizada pelo *package* e não trabalha com escalas invertidas, forçando o próprio utilizador a inverter a escala manualmente, convertendo a variável/indicador em causa. No caso de estudo, foram convertidos os valores da escala PESOAG para valores negativos possibilitando assim a criação de uma curva *ROC* com *AUC* superior a 0.5. Ainda assim pode-se considerar este *package* de fácil implementação por ser apenas necessário utilizar um comando com as informações de *response* e *prediction*, tornando executável para qualquer utilizador. Para demonstrar como foi obtido o *output* da Figura 17, encontra-se na Figura 18

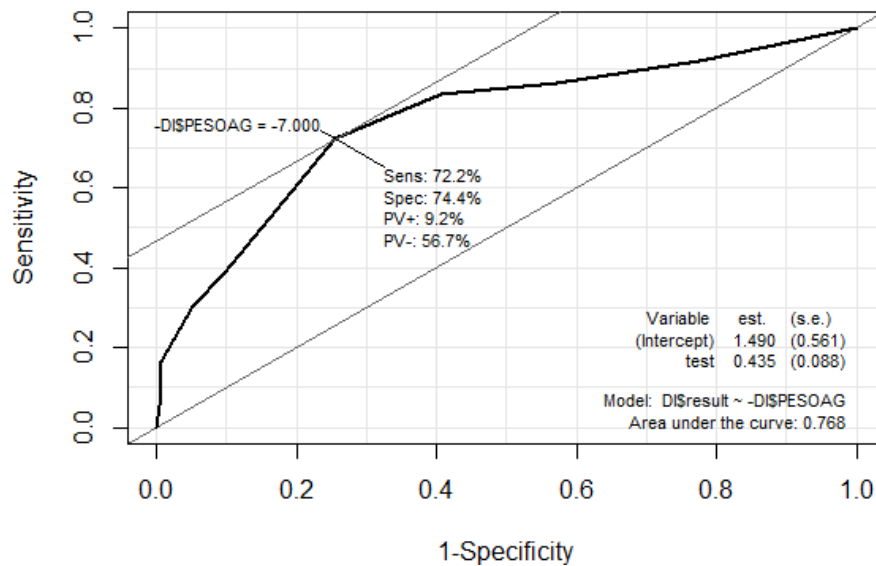


Figura 17: Exemplo de um dos gráficos ROC gerados através do *package Epi* com o *output* implementado no próprio gráfico.

a linha de comandos necessária para obter este mesmo gráfico, comprovando assim a sua simplicidade.

```
11 library(Epi)
12 ROC(-DISPESOAG,DI$result, plot= "ROC")
```

Figura 18: Linha de comandos para o *package Epi*.

3.1.7 *caTools*

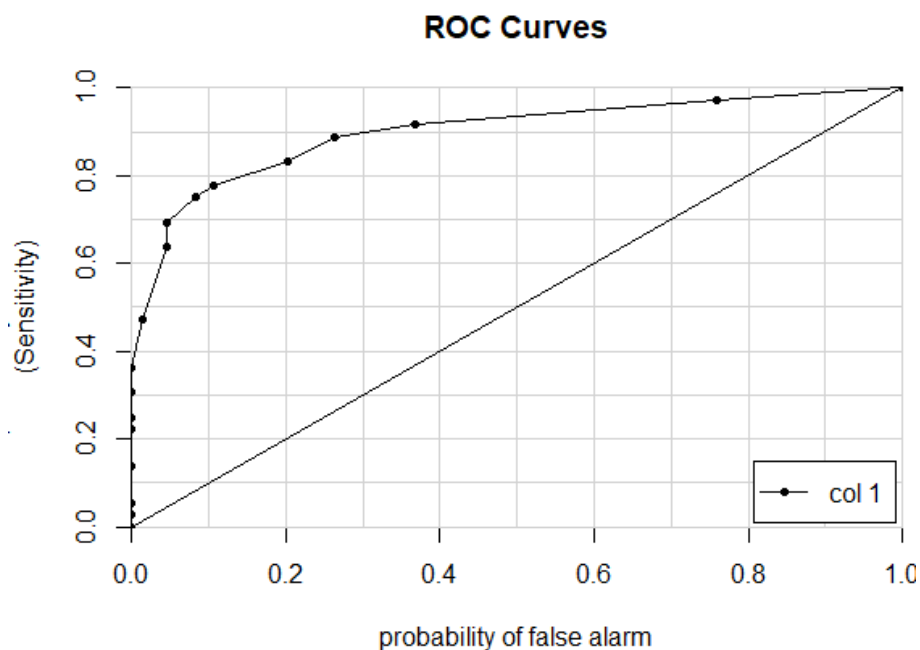
O *caTools* (Tuszynski, 2019) teve a sua última atualização na data de 17 de janeiro de 2020, sendo que, se encontra atualmente na versão 1.18.0. De acordo com o [CRAN](#) não apresenta qualquer tipo de dependências de outros *packages*. O seu autor é Jarek Tuszynski e não há registo de um responsável pela manutenção do *package*, contudo, Harutyun Khachatryan apresenta-se responsável pela manutenção da sua página [CRAN](#). Este *package* está descrito com diversas funções entre elas leitura/escrita para arquivos binários GIF e ENVI, classificador *LogitBoost*, codificador/descodificador *base64*, cálculo rápido de *AUC*, entre outras. Contudo, neste trabalho apenas interessa a utilização deste *package* para o cálculo de *AUC*.

Após explorado o *package* verificou-se que a função responsável por trabalhar na área das curvas *ROC*, é a *colAUC()*, sendo que, como referido anteriormente o seu foco passa pelo cálculo da área abaixo da curva, tendo ainda a capacidade de gerar curvas *ROC*. Para o efeito é necessário passar ao comando o parâmetro *plotROC* com o valor lógico de *TRUE*,

sendo devolvido no final a curva *ROC* num gráfico e o valor da *AUC* no terminal. Não fornece mais nenhum tipo de medida dentro da área das curvas *ROC*. Visualmente o gráfico apresenta algumas falhas, não é apresentada a linha das abcissas e ordenadas bem como a respetiva linha $x=y$ (adicionada apenas através do comando exterior `abline()`), como é demonstrado na Figura 20. A linha de comandos para aceder a este exemplo via R é a representada na Figura 19

```
1 library(caTools)
2
3 colAUC(DI$CRIB,DI$result, plotROC=T, alg=c("wilcoxon","ROC"))
4 abline(0,1)
```

Figura 19: Linha de comandos para aceder ao exemplo do *package caTools*.



AUC:

```
> colAUC(DI$CRIB,DI$result, plotROC=T, alg=c("wilcoxon","ROC"))
      [,1]
0 vs. 1 0.8994361
```

Figura 20: Output do *package caTools*, AUC no terminal e respetivo plot.

Relativamente à utilização deste *package* para análise do *dataset* das variáveis independentes, à semelhança do *package ROC* do *Bioconductor* foram encontrados alguns entraves devido à estrutura de dados. Isto significa que, uma vez que as nossas amostras apresentam dimensões distintas num mesmo *dataframe*, ao efetuar a leitura pelo R irão ser gerados automaticamente valores de *NA* nos espaços em branco. Sendo assim, para limpar estes valores de *NA* o utilizador deverá criar vetores para cada uma das amostras ou então fragmentar o *dataset* em

diversos *sub-datasets*, evitando assim a existência de valores de *NA*. Na amostra de variáveis dependentes este *package* apresenta uma correção dos valores de *prediction*, sendo que, ao ser fornecido este vetor, a própria função realiza uma comparação entre *prediction* e $-prediction$ sendo o *output* aquele que apresentar uma *AUC* superior, evitando assim que sejam gerados gráficos com *AUC* inferior a 0.5. Aplicando esta definição ao nosso *dataset*, aquilo que foi efetuado manualmente para muitos *packages* para a amostra PESOAG ao converter os seus valores para valores negativos, é efetuado automaticamente pelo *package caTools*. Este *package* apenas processa o gráfico e calcula o respetivo valor de *AUC* na análise *ROC*.

3.1.8 Comp2ROC

Relativamente ao *Comp2ROC* (Braga et al., 2016), este *package* teve a sua última atualização a 1 de julho de 2016 e encontra-se na versão 1.1.4. Os seus autores são Ana C. Braga, Hugo Frade, Sara Carvalho e Andre M. Santiago, sendo Ana C. Braga a responsável pela sua manutenção. Apresenta dependências a dois *packages*, o *ROCR* e o *boot*.

Para trabalhar com este *package* existem dois comandos principais, sendo eles, o *roc.curves.boot()* e o *rocboot.summary()*. O principal atributo deste *package* é a realização de uma comparação completa de duas curvas *ROC*. Contudo não processa curvas *ROC* individuais. Devolve o gráfico das curvas *ROC* empíricas para duas escalas a comparar e dois gráficos com informação complementar, relativa à área abaixo da curva e um histograma relativos à metodologia desenvolvida por Braga et al. (Braga et al., 2011), como representado nas figuras 21 e 22.

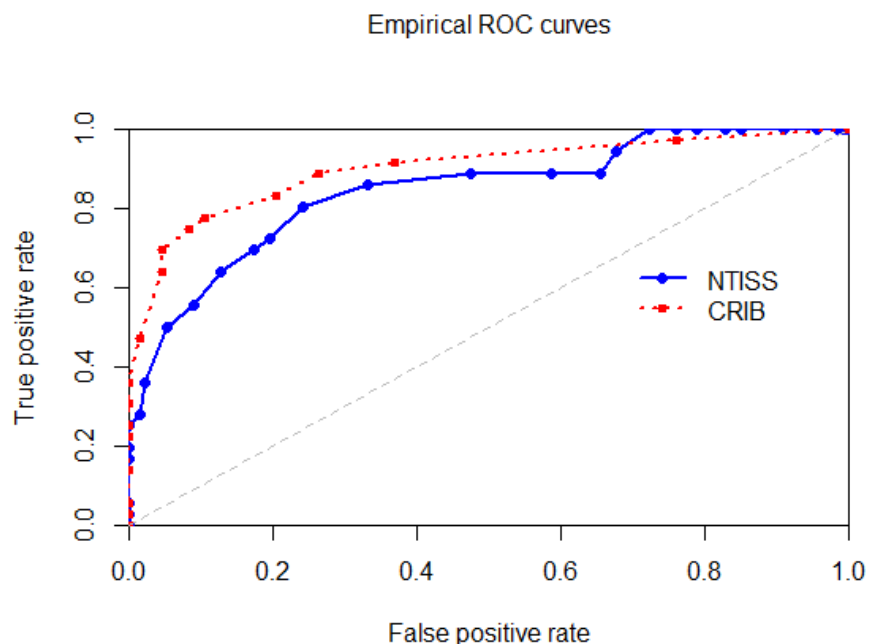


Figura 21: Gráfico da comparação de dois indicadores utilizando o *package Comp2ROC*.

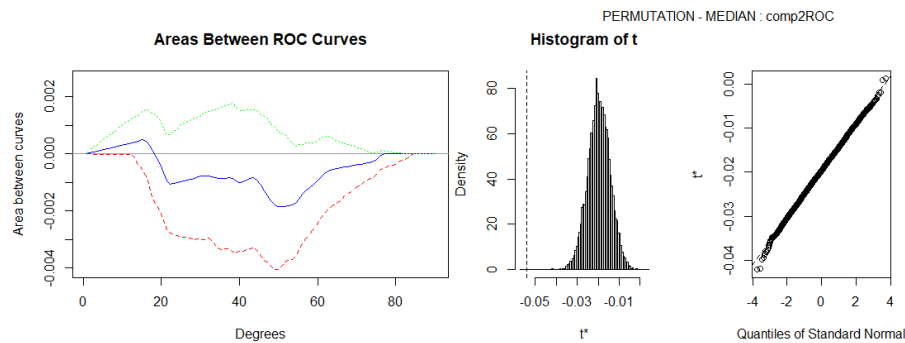


Figura 22: Figura com o gráfico das áreas e histograma devolvido pelo *package*.

Apesar de trabalhar apenas com duas curvas este *package* consegue devolver bastante informação individual para cada uma delas, como por exemplo: área, erro padrão, área através do método do trapézio, limite superior e inferior do IC. Para a comparação de duas curvas devolve novos dados estatísticos importantes como é o caso do coeficiente de correlação entre as áreas, *z-stats*, *p-value*, a soma das diferenças entre áreas, o limite inferior e superior do IC e o número de vezes que as curvas *ROC* se cruzam. Não é tão acessível para utilizadores menos experientes, mas mais eficaz. Devolve diferentes valores importantes tanto para uma análise de uma curva *ROC* como para a comparação de duas curvas. O *output* com estas informações é demonstrado na Figura 23 e pode ser exportado para *ficheiro.txt*.

```
> rocboot.summary(results,NTISS,CRIB)
```

NTISS		CRIB	
Area:	0.845447	Area:	0.8994361
Standard Error:	0.03859918	Standard Error:	0.0386395
Area through Trapezoidal Method:	0.8453683	Area through Trapezoidal Method:	0.8993809
CI Upper bound (Percentil Method):	0.9160479	CI Upper bound (Percentil Method):	0.9561456
CI Lower bound (Percentil Method):	0.7660793	CI Lower bound (Percentil Method):	0.8298742

```
Correlation coefficient between areas: 0.345652
TEST OF DIFFERENCES
Z stats: -1.296885
p-value: 0.1946709
Sum of Global Areas Differences (TS): -0.05398914
CI Upper bound (Percentil Method): 0.0561038
CI Lower bound (Percentil Method): -0.1506057
Number of Crossings: 1
```

Figura 23: Dados fornecidos no terminal pelo *package Comp2ROC*.

Para aceder aos exemplos demonstrados nas Figuras 21, 22 e 23 é necessária a linha de comandos representada na Figura 24.

```
21 nameE = "comp2ROC"
22 NTISS = "NTISS"
23 CRIB = "CRIB"
24 data = read.file("dados_iniciais.csv", TRUE, ";", ".", NTISS, TRUE, CRIB, TRUE, "result", TRUE)
25 results=roc.curves.boot(data,name=nameE,mod1=NTISS,mod2=CRIB)
26 rocboot.summary(results,NTISS,CRIB)
```

Figura 24: Exemplo de uma linha de comandos para processar o *output* para o *Comp2ROC*.

Relativamente à sua utilização direta nos *datasets* da dissertação, este *package* não apresentou qualquer tipo de complicações. Embora a sua utilização seja um pouco mais complexa,

após a organização do código e interpretar cada um dos parâmetros é possível obter um *output* bastante satisfatório e completo. Relativamente à variável PESOAG o *Comp2ROC* já apresenta uma funcionalidade própria que permite ao utilizador indicar ao R se está a trabalhar com uma variável onde o valor positivo corresponde ao valor mais alto ou mais baixo da escala a estudar. Para o efeito é necessário num dos parâmetros da função *read.file()* indicar a seguir à variável um indicador *TRUE* ou *FALSE*, sendo o *TRUE* associados às variáveis onde o valor positivo (1) está associado a valores superiores e o *FALSE* a valores inferiores. De salientar que esta função (*read.file()*) é exclusiva deste *package* e apresenta uma estrutura particular.

3.2 *software stata*

Depois da exploração de alguns *packages* selecionados para o ambiente de trabalho R, foi solicitada uma licença para poder ser explorado o *software STATA*. Logo à partida este *software* apresenta um grande entrave que é o facto de ser necessário a compra de licenças para a sua exploração. Numa primeira abordagem este *software* pode parecer um pouco confuso, contudo após a sua análise existem diversos fatores positivos a ter em conta. Desde logo o facto de ser possível trabalhar num ambiente amigável onde existem funcionalidades para trabalhar em diferentes campos da área da estatística. Seguindo-se para um *input* de dados bastante simples e esclarecedor sem problemas de maior detetados, onde é disponibilizado um menu prático de se seguir ou então executando uma linha de comandos adequada. Embora este *software* não guarde diretamente o *script* para ser processado, o mesmo é guardado num editor do próprio *software*, o que será suficiente para ficar com o código disponível ao iniciar outras sessões de trabalho.

O mesmo *dataset* utilizado para o R foi importado utilizando para o efeito o comando *import*, sendo também possível através da barra de ferramentas do *software* fazer este mesmo *import* de forma manual, selecionando a opção *file*, seguindo-se *import* e posteriormente o tipo de documento onde está o *dataset*. No *STATA* existe a possibilidade de importar os conjuntos de dados em diversos formatos, sendo eles, *excel*, *txt*, *csv*, *SAS xport*, *Haver dataset*, *OBDC data source* e *XML*.

Após o *dataset* estar pronto a ser usado foram utilizados essencialmente os comandos *roctab* e *roccomp*, uma vez que os mesmos correspondem ao estudo para análises *ROC* não paramétricas. Relativamente aos outros comandos como *rocreg*, *rocregplot*, *rocfits* e *rocgold* apenas foi explorado um pouco dos seus documentos explicativos do *STATA* e os menus que disponibiliza no *software*.

3.2.1 *roctab*

Este é o comando utilizado para trabalhar com os *datasets* utilizados, correspondendo a curvas *ROC* não paramétricas sem covariáveis e realizando para o efeito a respetiva análise *ROC*. Para aceder a este comando existem então duas alternativas, ou através do menu ou pela linha de comandos. Uma breve explicação para ambos será dada de forma a introduzir este comando.

Menu

Inicialmente para aceder ao menu correspondente ao comando *roctab* foram percorridas as etapas demonstradas na Figura 25.

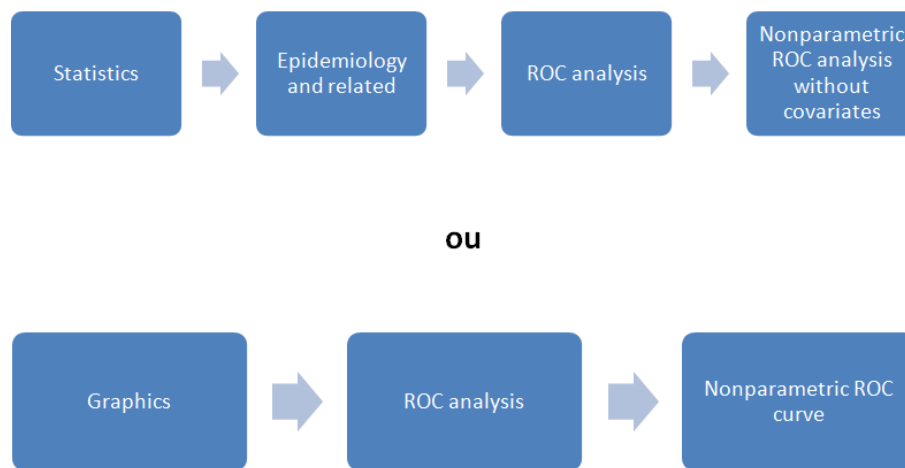


Figura 25: Etapas para aceder ao menu do comando *roctab*.

Ambas têm como destino o mesmo menu e em nada interfere com o tipo de análise realizado. Após entrar no menu do comando, acede-se às funcionalidades que o mesmo disponibiliza dentro do *STATA*. Relativamente ao primeiro valor solicitado, *reference variable (refvar)*, corresponde ao estado do doente e será sempre uma variável binária, com os valores de zero (0) a corresponder à ausência de doença, normal ou negativo, e um (1) à presença de doença, anormal ou positivo. Nos *datasets* utilizados nesta dissertação, esta variável corresponde às colunas *res1*, *res2*, *res3* e *res4* para amostras independentes e para a coluna *result* nas amostras dependentes. Na *classification variable (classvar)* terá de ser atribuído o resultado do teste, em que quanto maior for o valor do resultado maior será a probabilidade de morte ou risco elevado de morte. No *dataset*, para amostras dependentes corresponde aos indicadores de taxa de mortalidade neonatal e para as amostras independentes aos valores dos diferentes hospitais. De mencionar, que as variáveis que funcionem de forma inversa, isto é, valores mais baixos da escala correspondem ao atributo positivo ($result=1$),

devem ser previamente convertidas para valores negativos, aplicando o comando *replace* e multiplicando os valores da variável por -1 .

Posteriormente são apresentadas diferentes funcionalidades neste menu. O utilizador pode então trabalhar com as seguintes funcionalidades:

- apresentar os índices de Gini e Pietra;
- cálculo binomial dos IC's;
- exibir códigos numéricos em vez de etiquetas de valor ;
- exibir sensibilidade/especificidade por cada ponto de corte;
- apresentar uma tabela de contingência para o número de amostras para cada valor das *refvar*;
- seleccionar o método para cálculo dos erros padrão: DeLong, Bamber ou Hanley;
- no final seleccionar o nível de confiança (95% por *default*).

Dentro do gráfico *ROC*:

- reproduzir o gráfico *ROC*;
- apagar a linha de referência de 45 graus do gráfico;
- utilizar especificidade em vez de 1 -especificidade;
- apresentar o valor da *AUC*.

Diferentes ajustes podem ser realizados ao gráfico *ROC* reproduzido, como é o caso de legendas, definir o tipo de curva e as suas propriedades, títulos, dimensões e cores das curvas. Podem ainda ser adicionadas condicionantes à análise *ROC*, acedendo ao separador *if/in* do menu *roctab*.

Foram posteriormente exploradas algumas características deste menu, como é ilustrado na Figura 26.

De seguida ao processar os valores do menu anterior, o *output* é representado tal como surge na Figura 27.

Esta parte do *output* é então fornecida quando o utilizador escolhe apresentar a *AUC* no menu. Os valores devolvidos quando a opção para apresentar os valores das sensibilidades e especificidades é seleccionada, está representada na Figura 28.

Por fim, como foi seleccionado para se apresentar o gráfico, o mesmo surge numa aba separada, tal como podemos verificar na Figura 29.

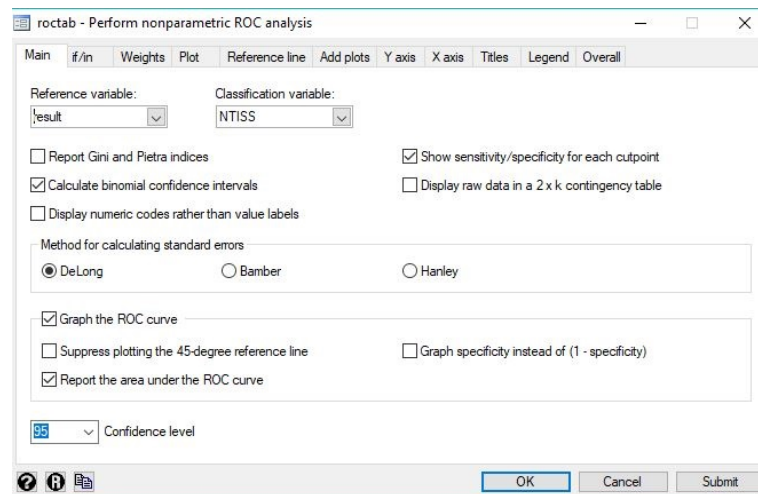


Figura 26: Formato do menu do comando *roctab*.

Obs	ROC Area	Std. Err.	— Binomial Exact — [95% Conf. Interval]	
169	0.8454	0.0386	0.78276	0.89697

Figura 27: Exemplo de *Output* produzido pelo comando *roctab*.

Detailed report of sensitivity and specificity					
Outpoint	Sensitivity	Specificity	Correctly Classified	LR+	LR-
(>= 4)	100.00%	0.00%	21.30%	1.0000	
(>= 5)	100.00%	0.75%	21.89%	1.0076	0.0000
(>= 6)	100.00%	1.50%	22.49%	1.0153	0.0000
(>= 7)	100.00%	4.51%	24.85%	1.0472	0.0000
(>= 8)	100.00%	9.02%	28.40%	1.0992	0.0000
(>= 9)	100.00%	15.04%	33.14%	1.1770	0.0000
(>= 10)	100.00%	17.29%	34.91%	1.2091	0.0000
(>= 11)	100.00%	21.05%	37.87%	1.2667	0.0000
(>= 12)	100.00%	24.06%	40.24%	1.3168	0.0000
(>= 13)	100.00%	27.82%	43.20%	1.3854	0.0000
(>= 14)	94.44%	32.33%	45.56%	1.3957	0.1718
(>= 15)	88.89%	34.59%	46.15%	1.3589	0.3213
(>= 16)	88.89%	41.35%	51.48%	1.5157	0.2687

Figura 28: Exemplo de uma fração do *output* fornecido relativo à opção sensibilidade/especificidade.

Linha de comandos

Utiliza-se então o comando *roctab* seguido do nome da *refvar* (que corresponde aos *results* nos *datasets*) e posteriormente das *classvar* (correspondente aos nomes dos indicadores nas amostras dependentes e dos hospitais nas amostras independentes).

Todas as opções mencionadas anteriormente possuem um parâmetro que pode ser incluído no próprio comando *roctab*. Na Tabela 3 colocam-se as diferentes opções do comando e as respectivas descrições.

Posteriormente, é apresentada a construção de uma análise *ROC* recorrendo ao comando *roctab* e utilizando algumas das opções apresentadas na Tabela 3, através da linha de comandos.

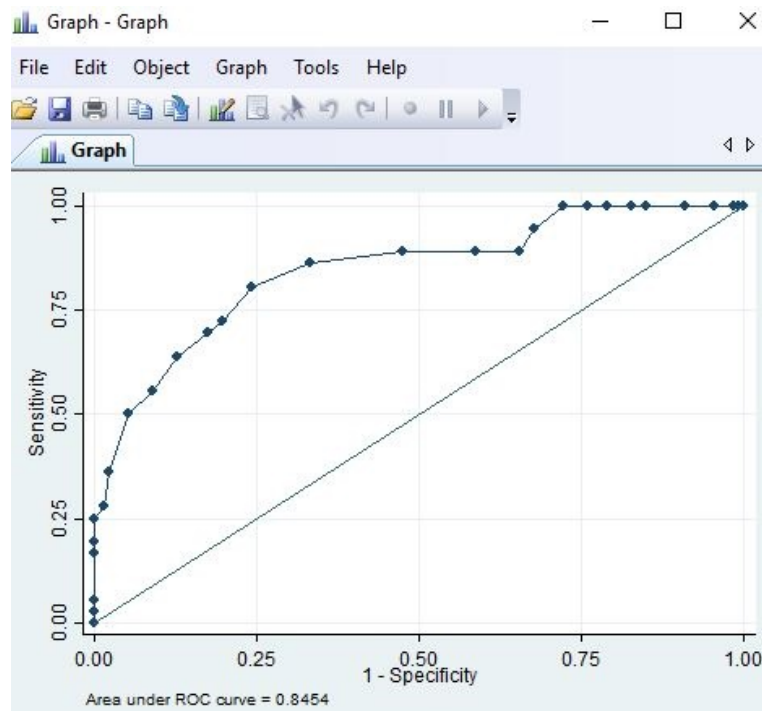


Figura 29: Gráfico ROC gerado pelo menu do *roctab* nas opções definidas.

O comando é introduzido inicialmente seguindo-se das duas variáveis: *refvar* e *classvar*. A partir daqui já será possível gerar a análise ROC com todas as opções em modo *default*. Contudo e de forma a explorar algumas funcionalidades deste *software*, foram incluídos alguns parâmetros que estão representados na Figura 30.

```
Command
roctab Res1 Hospital1, table graph summary
```

Figura 30: Valores fornecidos ao comando *roctab* para gerar análise ROC no STATA via linha de comandos.

Foi pedido então ao comando que gere uma análise ROC utilizando *Res1* como *refvar* e *Hospital1* como *classvar*. Posteriormente são fornecidas algumas opções para devolver determinadas respostas, neste caso foi pedido ao comando que gere uma tabela de contingência, o gráfico ROC e a respetiva AUC que vem acompanhada de outros parâmetros como o erro padrão, número de observações e IC. O *output* foi devolvido e é apresentado na Figura 31.

O gráfico ROC é gerado numa aba distinta, onde é apresentada a AUC por de baixo do gráfico tal como ilustrado na Figura 32.

Para aceder ao documento explicativo do *roctab*:

<https://www.stata.com/manuals13/rroctab.pdf>

Tabela 3: Descrição para as diferentes opções do comando *roctab*.

Opções da função	Descrição
lorenz	Apresentar índices de Gini e Pietra
binomial	Calcular intervalos de confiança binomiais exatos
nolabel	Exibir códigos numéricos em vez de etiquetas de valor
detail	Mostrar detalhes sobre sensibilidade / especificidade para cada ponto de corte
table	Exibir os dados brutos em uma tabela de contingência 2 x k
bamber	Calcular erros padrão usando o método Bamber
hanley	Calcular erros padrão usando o método Hanley
graph	Representar graficamente a curva ROC
noreflin	Retirar a linha de referência de 45 graus
summary	Demonstrar a área sob a curva ROC
specificity	Especificidade em vez de 1-especificidade
level(#)	Definir nível de confiança; o padrão é (95)
plotopts	Personalizar a representação da curva ROC
rlopts	Renderização aleatória da linha de referência
addplot	Adicione outros gráficos ao gráfico gerado

Res1	Hospital1						Total
	6	7	8	9	10	12	
0	0	2	3	1	1	0	65
1	1	0	1	0	0	1	13
Total	1	2	4	1	1	1	78

Res1	Hospital1					Total
	13	14	16	18	19	
0	2	1	0	0	1	65
1	1	0	1	1	0	13
Total	3	1	1	1	1	78

Obs	ROC		—Asymptotic Normal— [95% Conf. Interval]	
	Area	Std. Err.		
78	0.5923	0.1080	0.38056	0.80405

Figura 31: Output fornecido pelo comando realizado na Figura 30.

3.2.2 *roccomp*

A função *roccomp* realiza a comparação de diferentes curvas, sendo possível desenhá-las dentro do mesmo gráfico e comparar diferentes valores devolvidos pelo comando.

Na amostra dependente o comando foi testado com êxito, tanto através do menu como da própria linha de comandos. A sua utilização é bastante simples e visa unicamente a seleção da *refoar* e *classvar*, posteriormente o utilizador deve indicar quais as *classvar* a serem adicionadas ao gráfico.

Para aceder ao menu o utilizador pode percorrer os passos representados na Figura 33.

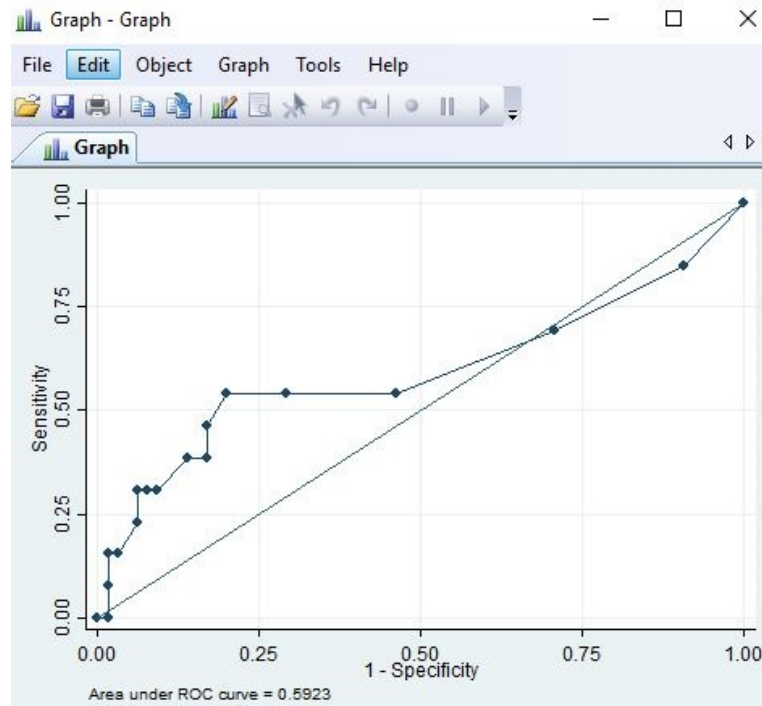


Figura 32: Gráfico ROC gerado pela linha de comandos definida anteriormente.

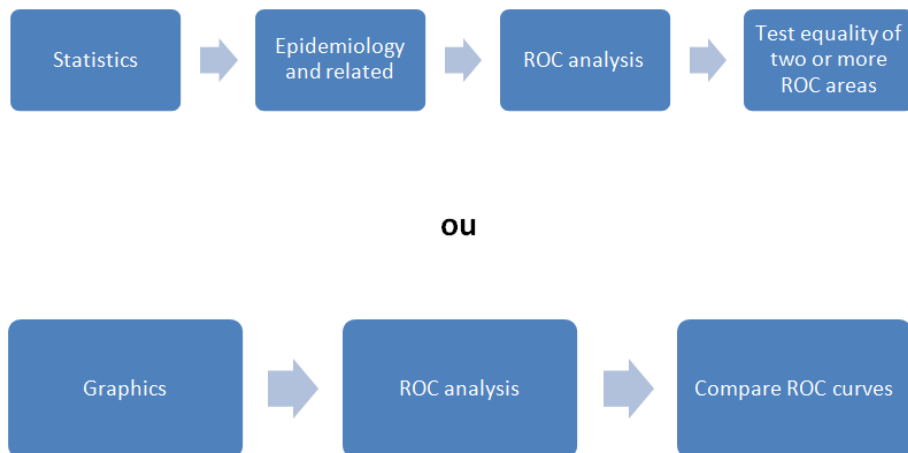


Figura 33: Etapas para aceder ao menu do comando *roccomp*.

O menu apresenta-se de forma simples e após a seleção das diferentes variáveis, o utilizador unicamente terá de selecionar as configurações para o gráfico. Se o utilizador não desejar processar o gráfico *ROC*, serão imediatamente devolvidos no terminal valores relativos à análise *ROC*, com número de observações, *AUC*, erro padrão e *IC*. Caso o utilizador selecione a opção de processar o gráfico *ROC*, pode também aceder a estes valores caso a opção "Report the area under the ROC curve" esteja selecionada. Para contextualizar este *output* é representado na Figura 34 os valores devolvidos.

```
. roccomp result NTISS SNAPPE CRIB PESOAG SNAP, graph summary
```

	Obs	ROC Area	Std. Err.	—Asymptotic Normal— [95% Conf. Interval]	
NTISS	169	0.8454	0.0386	0.76979	0.92110
SNAPPE	169	0.8827	0.0326	0.81874	0.94672
CRIB	169	0.8994	0.0339	0.83306	0.96581
PESOAG	169	0.7681	0.0476	0.67485	0.86128
SNAP	169	0.8818	0.0312	0.82072	0.94285

Ho: area (NTISS) = area (SNAPPE) = area (CRIB) = area (PESOAG) = area (SNAP)
chi2 (4) = 9.43 Prob>chi2 = 0.0511

Figura 34: Algumas métricas devolvidas no *output* do *roccomp*.

À semelhança do *roctab* pode ainda ser suprimida a curva de referência do gráfico (45°). Por fim, uma opção interessante, em que o utilizador pode separar todas as curvas por gráficos diferentes, sendo eles representados em simultâneo num mesmo separador.

Foram testadas duas distintas configurações, uma com as curvas num mesmo gráfico e outra com as curvas separadas por gráficos distintos, é apresentado nas Figuras 35 e 36, respectivamente.

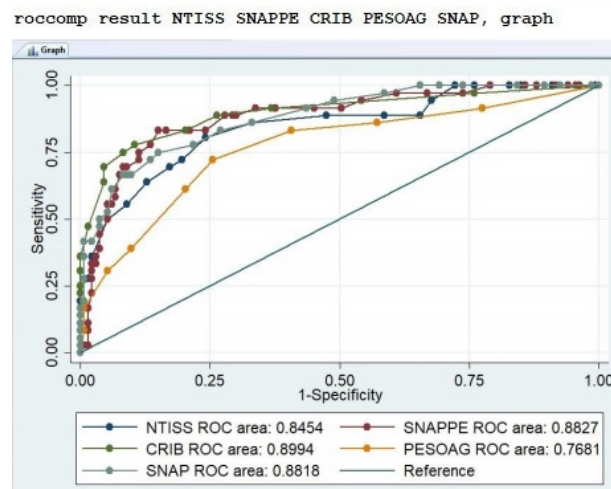


Figura 35: Curvas ROC geradas num mesmo gráfico pelo comando *roccomp*.

Como ilustrado nas Figuras 34, 35 e 36, todos os exemplos foram processados usando a linha de comandos, contudo, facilmente o utilizador conseguirá obter o mesmo resultado pelo menu do *roccomp*.

Relativamente à amostra independente, houve a necessidade de criar um novo *dataset* para esta amostra, uma vez que a organização das variáveis solicitada pelo *software* para este tipo de amostra é diferente da utilizada para as restantes ferramentas. O formato do *dataset* está representado na Figura 37, onde são criadas três variáveis, uma com os valores dos

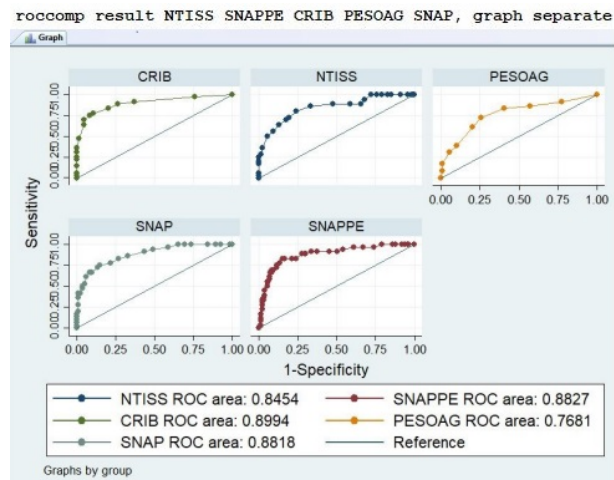


Figura 36: Curvas ROC geradas em gráficos distintos pelo comando *roccomp*.

hospitais, outra com os resultados e por fim uma variável em que a cada linha está associado o número do hospital a que corresponde os valores das outras duas variáveis.

	A	B	C
1	Valores	Res	Hospitais
2	2	0	Hosp1
3	4	0	Hosp1
4	1	0	Hosp1
5	2	0	Hosp1
6	2	0	Hosp1
7	14	0	Hosp1
8	1	0	Hosp1
9	3	0	Hosp1

Figura 37: Formato do *dataset* utilizado no *roccomp* para amostras independentes.

Para criação de um exemplo com este tipo de amostra a Figura 38, ilustra a introdução dos valores dos parâmetros no menu do *roccomp*.

Nos três campos preenchidos com variáveis é então solicitado como anteriormente as variáveis *reference* e *classification*, que neste caso corresponde à variável "Res" e "Valores" do novo *dataframe*. Posteriormente terá de ser definida a variável "split" que corresponde à atribuição de cada linha um hospital, sendo neste caso, a variável denominada como "Hospitais" na Figura 37.

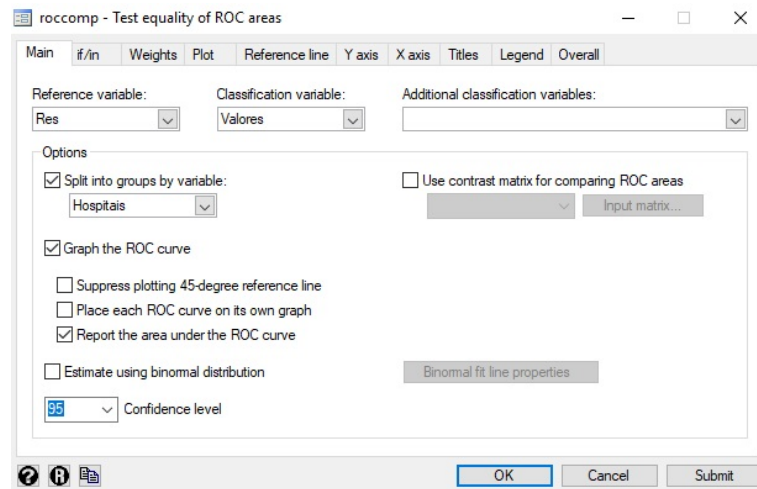


Figura 38: Valores fornecidos ao menu *roccomp* para comparação da análise ROC de amostras independentes.

Na Figura 39 é representado o *output* gráfico devolvido pelo menu criado para comparação de amostras independentes. Relativamente ao restante *output*, são apresentados os mesmos dados estatísticos que os fornecidos para as amostras dependentes neste comando.

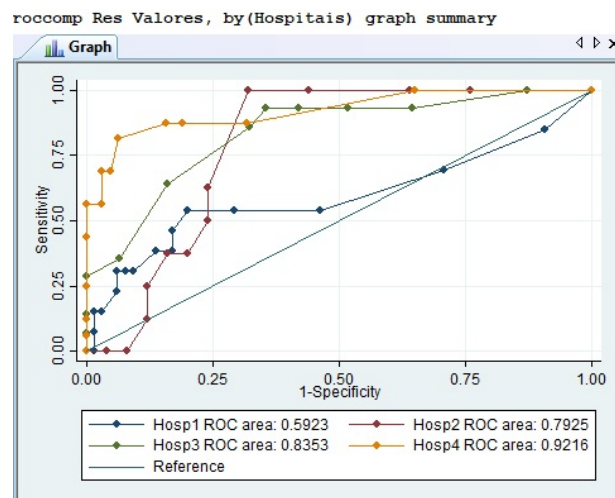


Figura 39: *Output* fornecido pelo comando *roccomp* para amostras independentes.

Para aceder ao documento explicativo do *roccomp*:

<https://www.stata.com/manuals13/rroccomp.pdf>

3.2.3 *roclog* e *roclogplot*

O comando *roclog* realiza modelos de regressão ROC. É utilizado para executar a análise ROC em classificação e permite também, utilizar dados discretos com a presença de cóvariáveis.

À semelhança do comando *roctab*, será sempre necessário fornecer os valores da *refvar* e das *classification variables* para proceder com esta análise. Para aceder ao seu menu, deverá se executar a ordem de operações presente na Figura 40.



Figura 40: Etapas para aceder ao menu do comando *rocreg*.

Já o comando *rocregplot* desenha as curvas *ROC* tal como definidas pela função anterior, por isso mesmo este comando só funciona sendo utilizado logo de imediato à execução do comando *rocreg* para a respetiva análise.

Visto que esta dissertação não está direcionada a esta área das curvas *ROC*, será apenas dada uma abordagem mais superficial às características disponibilizadas por estes comandos.

O *rocreg* tem portanto um menu direcionado à sua utilização, apresentando-se o mesmo como representado na Figura 41.

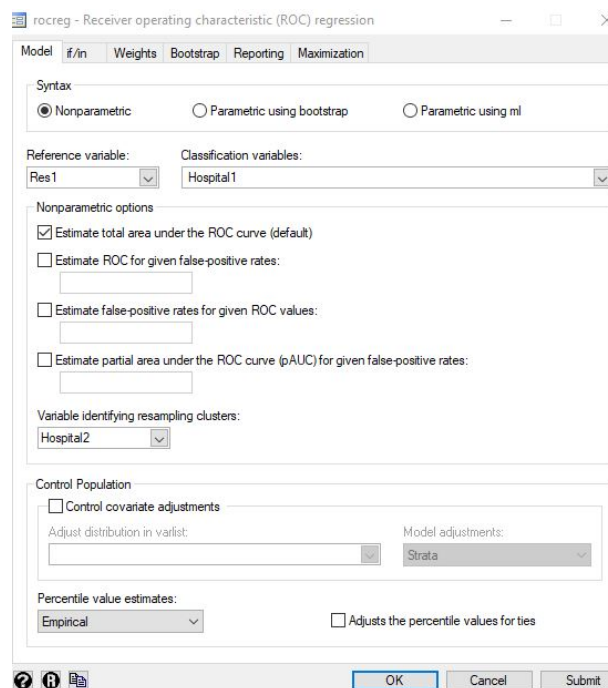


Figura 41: Menu *rocreg* direcionado a criação de modelos *ROC* de regressão.

Inicialmente o utilizador terá de selecionar o tipo de análise *ROC*, não paramétrica, paramétrica com *bootstrap* ou paramétrica usando a máxima verossimilhança. Posteriormente o menu será alterado consoante a opção escolhida anteriormente.

Caso o utilizador deseje processar um gráfico com o modelo *ROC* originado pelo comando *rocreg*, deverá seguir as etapas presentes na Figura 42.

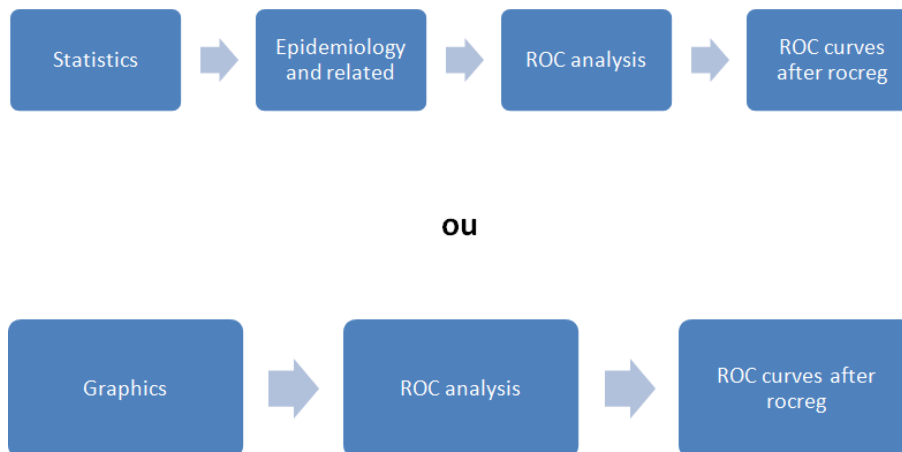


Figura 42: Etapas para acessar ao menu do comando *rocregplot*.

Posteriormente irá acessar ao menu correspondente ao comando *rocregplot* e após configurar a curva, selecionar o botão [OK] e o seu *output* surgirá de imediato. Segue o exemplo ilustrativo na Figura 43.

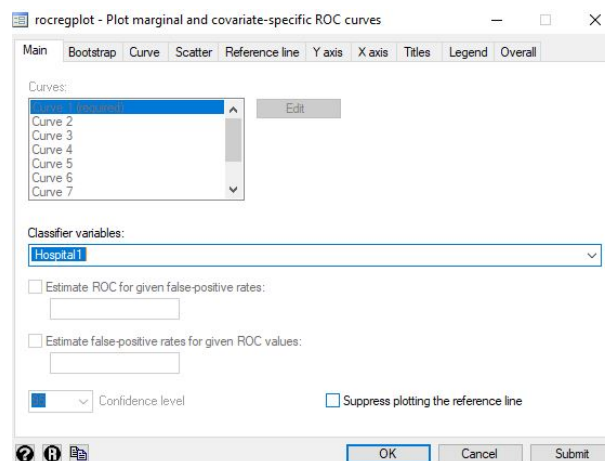


Figura 43: Menu para criação de gráficos através do comando *rocregplot*.

Todos os comandos podem ser utilizados na linha de comandos do próprio *STATA*, com opções específicas que visam realizar o que estiver codificado no seu menu. Para o efeito, o *STATA* disponibiliza documentos onde explica a forma que cada opção do comando interage na análise *ROC*.

Para acessar ao documento explicativo do *rocreg* e *rocregplot*:

<https://www.stata.com/manuals13/rrocreg.pdf>

<https://www.stata.com/manuals13/rrocregplot.pdf>

3.2.4 *rocgold*

Relativamente ao comando *rocgold*, é realizada a análise *ROC* múltipla mas em comparação com uma curva *ROC gold standard*, ou seja, a que estatisticamente apresenta melhores resultados. distingue-se do comando *roccomp* pela entrada referente à variável *Gold standard*, mencionada no menu do comando como *Gold variable*.

Embora processe a *AUC* e o seu erro padrão, são processados também diferentes valores comparativamente ao *roccomp*. Estes valores são referentes à comparação com a variável padrão selecionada pelo utilizador, sendo eles, teste qui-quadrado, graus de liberdade (df) e correção de Bonferroni. No menu é dada ainda a hipótese de se substituir o método Bonferroni pelo método de Sidak's para as comparações múltiplas. A acrescentar estas funcionalidades, tal como no comando *roccomp* o utilizador pode ou não processar o gráfico *ROC* e posteriormente definir se quer suprimir a curva de 45°, se deseja todos os gráficos num mesmo plot e se deseja reportar ou não os valores de *output* em conjunto com o gráfico. Uma vez que este comando não se encontra no âmbito desta dissertação o mesmo não foi testado com o *dataset* utilizado neste trabalho. Para aceder ao seu menu foi necessário percorrer as etapas apresentadas na Figura 44.

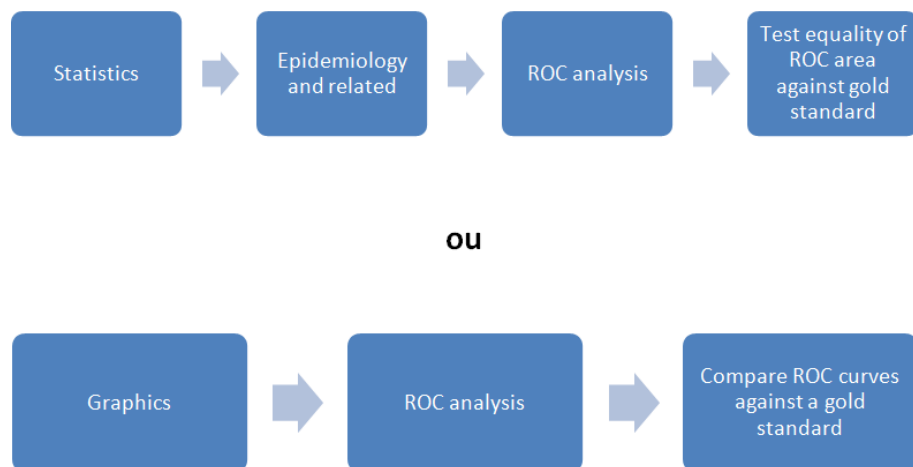


Figura 44: Etapas para aceder ao menu do comando *rocgold*.

O documento explicativo do *rocgold* é o mesmo que o apresentado para o comando *roccomp*.

3.2.5 *rocfitt*

Uma vez que os *datasets* são destinados a curvas *ROC* empíricas, este comando não será testado por se destinar à utilização em amostras cujo objetivo visa o ajuste de curvas *ROC*. Contudo será explorado o seu menu no *software STATA* bem como o seu documento explicativo.

Este comando é referente à utilização em dados que estejam parametrizados e portanto inerentes a uma análise *ROC* paramétrica. Realiza o respetivo ajuste da análise assumindo uma distribuição binomial. Para este comando são devolvidos valores de *output* para os índices da *ROC* Area, $d(m)$, $d(e)$ e $d(a)$, com os seus respetivos valores, erro padrão e IC de 95%.

Para aceder ao menu deste comando é necessário percorrer as etapas representadas da Figura 45. E caso necessite de processar o gráfico deverá seguir a etapa correspondente, apresentada na Figura 46.



Figura 45: Etapas para aceder ao menu do comando *rocfit*.



ou

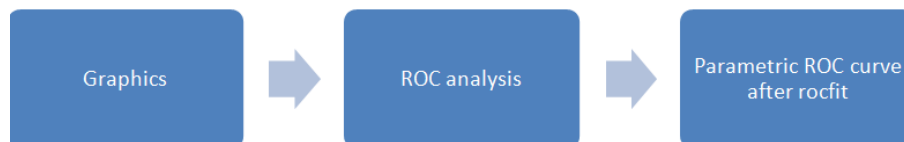


Figura 46: Etapas para aceder ao menu do comando *rocfit* que processa o respetivo gráfico.

No menu apresentado o utilizador apenas terá de fornecer os valores da *classvar* e da *refvar* e caso posteriormente necessite de processar o respetivo gráfico aceder ao menu indicado.

Para aceder ao documento explicativo do *rocfit*:

<https://www.stata.com/manuals13/rrocfit.pdf>

SHINY

Será também o objetivo desta dissertação a utilização do *package shiny* do ambiente de trabalho R. A utilização deste *package* visa a criação de uma aplicação *web* que será posteriormente disponibilizada na plataforma *shinyapps.io*. Esta aplicação aponta à criação de uma espécie de "*library*" com todos os *packages* explorados nesta dissertação e as respetivas funcionalidades disponibilizadas dentro de uma mesma página *web browser*.

Com a utilização deste *package*, numa das ordens de trabalhos desta dissertação, é pretendida a criação de uma aplicação *web*, esta mesma que será disponibilizada posteriormente na plataforma *shinyapps.io*. Esta aplicação aponta à criação de uma espécie de *library* com todos os *packages* abordados neste trabalho e as suas respetivas funcionalidades, disponibilizadas dentro de uma mesma pagina *web browser*.

Esta é uma ferramenta direcionada essencialmente a estatísticos e analistas que desejam partilhar os seus trabalhos, contudo, a sua versatilidade permite a qualquer pessoa dentro de qualquer ramo a utilização deste *package* para criação de aplicações (Cho et al., 2014). Sendo um *package* R inerentemente a linguagem de programação utilizada será a linguagem R, que se apresenta como bastante acessível para qualquer utilizador (Doi, 2016). Este *package* é baseado numa programação reativa visando sempre a que uma alteração na entrada de dados esteja direcionada uma saída de dados distinta. Sendo assim é possível ao utilizador personalizar a interface do aplicativo, acendendo aos *widgets* promovidos pelo *Rstudio* e que são facilmente implementados nas aplicações (Jahanshiri and Shariff, 2014).

Serve como exemplo, quando se alterar variáveis da estrutura de dados (*input*), os gráficos, textos ou tabelas (*output*) serão automaticamente alterados, correspondendo aos novos argumentos passados ao programa. A este fator importantíssimo junta-se o facto de ao utilizar uma aplicação *shiny* não seja necessário haver atualizações na pagina *web* para que a nossa saída de dados seja alterada. Para tornar a programação em *shiny* acessível a qualquer utilizador são disponibilizados diferentes conjuntos de código para *widgets* e para *upload* de ficheiros (por exemplo) pelo próprio *Rstudio*. (Jahanshiri and Shariff, 2014)

4.1 COMO OBTER E UTILIZAR shiny

Este *package* foi criado e disponibilizado pelo *Rstudio*, pelo que a forma mais prática de aceder ao *shiny* para criação de novas aplicações é proceder à instalação do próprio *Rstudio*. Todas as dependências são automaticamente instaladas por este processo o que torna o *package* automaticamente funcional (Wojciechowski et al., 2015).

Existem duas formas de proceder à utilização de aplicações *shiny* já criadas. A primeira será então simplesmente correr o *script* da própria aplicação no *Rstudio* (caso seja disponibilizado), ou então, caso o criador da aplicação disponibilize a mesma *online* por via da plataforma *shinyapps.io*. Neste segundo método a aplicação fica acessível a qualquer pessoa de forma gratuita, pelo que, pode ser utilizada em qualquer computador com acesso a internet.

4.2 COMO CONSTRUIR UMA APLICAÇÃO shiny

Este *package* necessita de duas partes para se apresentar funcional, a *web* interface e o *web* servidor bem como a respetiva atualização do *package*. A *web* interface permite ao utilizador fazer o *upload* de ficheiros e fornecer os parâmetros necessários ao mecanismo R para que sejam produzidos os resultados desejados. Para orientar quem está a começar a trabalhar no *shiny* é fornecido pelo *Rstudio* diferentes exercícios e tutoriais (Wojciechowski et al., 2015)

Para se “desenhar” a estrutura da aplicação que será desenvolvida nesta dissertação, esta mesma é criada dentro de dois *scripts*, *server.R* e *ui.R*, ou então num mesmo *script*, *app.R*, que funcionará exatamente da mesma maneira.

A aplicação criada terá em vista facilitar a implementação de uma análise *ROC* consoante os atributos de cada um dos *packages* estudados neste trabalho. Sendo assim, terá de haver uma ligação entre o conteúdo da *checklist* (Tabela 4) e aquilo que o utilizador desejar.

4.3 CRIAÇÃO DA APLICAÇÃO

Inicialmente é importante referir que a aplicação precisa de estar dividida em duas partes para se tornar funcional, neste caso a *ui.R* e o *server.R*. Sendo assim ambas as componentes serão abordadas em separado por se apresentarem com impactos distintos na aplicação.

Relativamente à componente *user interface (ui)* da aplicação, pode ser definida como o local onde o criador traça o aspeto visual da aplicação. Aqui deverão ser definidos os separadores, os *widgets*, as interações que tomam e a personalização da aplicação.

Já o *server.R* é relativo ao espaço em que o utilizador define as instruções/funções para construção da aplicação, que indispensavelmente entrará em contacto com as partes criadas na componente *ui.R*.

Como referido anteriormente existem duas formas para se criarem as duas componentes, dentro de um mesmo *script* (*app.R*) ou então separados por dois *scripts* (*ui.R* e *server.R*) que estarão ligados por uma função do *shiny*. Para este trabalho foi adotada a opção de criar a aplicação dentro de um mesmo *script*, de realçar que não existe qualquer impacto para as aplicações consoante a escolha, sendo esta apenas uma opção preferencial para o programador.

De forma a simplificar as funções de cada uma das componentes, está representado na Figura 47 um pequeno resumo das mesmas.

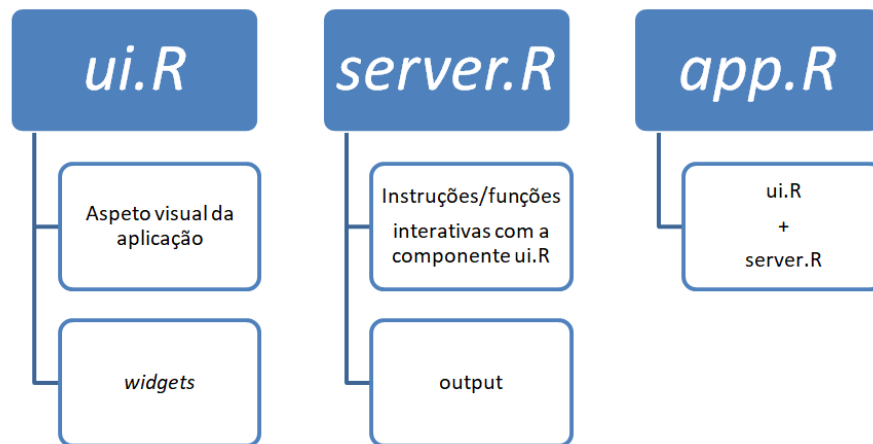


Figura 47: Diagrama resumo das componentes shiny.

Neste capítulo serão apenas demonstrados alguns exemplos para interpretação do código da aplicação, o mesmo encontra-se na íntegra no endereço <https://github.com/pquintasbcl95/R0SY> do repositório online *github*.

4.3.1 *ui.R*

De forma tornar o aspeto da aplicação mais atrativo foi utilizado o *package shinydashboard* como perfil para o painel pretendido.

Posteriormente a estar definida a *dashboard* requerida, foi dado início à estruturação da aplicação. O primeiro parâmetro (*skin*) irá corresponder à cor da *dashboard* e de seguida atribuído o nome da aplicação dentro da função *dashboardHeader*. Por fim, os separadores são criados dentro da função *sidebarmenu* e para a aplicação desenvolvida foram criados os que se apresentam na Figura 48.

Cada um dos separadores estará dentro de uma função *menuItem*, que permitirá a criação do seu próprio espaço dentro da aplicação. No primeiro parâmetro da função é passado o nome com que se apresentará na aplicação, de seguida o nome atribuído à variável para

```

ui <- dashboardPage(skin="red",
  dashboardHeader(title = "ROShiny"),
  dashboardSidebar(sidebarMenu(
    menuItem("Upload File", tabName = "upload", icon = icon("file-o")),
    menuItem("Checklist", tabName = "checklist", icon = icon("th-list")),
    menuItem("User guides", tabName = "userguides", icon = icon("info")),
    menuItem("One Curve", tabName = "onecurve", icon = icon("line-chart")),
    menuItem("Comparison 2 curves - PROC", tabName = "twocurves", icon = icon("line-chart")),
    menuItem("Comp2ROC File - Dependents", tabName = "upload1", icon = icon("file-o")),
    menuItem("Comp2ROC Dependents output", tabName = "output1", icon = icon("file-o")),
    menuItem("Comp2ROC File - Independents", tabName = "upload2", icon = icon("file-o")),
    menuItem("Comp2ROC Independents output", tabName = "output2", icon = icon("file-o"))
  ))

```

Figura 48: Exemplo de código dentro da função *sidebarMenu* para criação dos separadores.

trabalhar dentro da aplicação (*tabName*) e por fim o nome do *icon* que irá processar uma figura ao lado do nome do separador.

Após a criação dos separadores a próxima etapa é referente à criação da estrutura da *dashboard*, utilizando para o efeito a função *dashboardBody*. Aqui é criado o corpo dos separadores, ou seja, o que irá estar dentro dos mesmos. De forma a introduzir esta etapa, será dado um pequeno exemplo, neste caso para o separador do primeiro *upload* de ficheiros com o nome "Upload File" onde o seu interior é criado de acordo com a Figura 49.

```

dashboardBody(
  tabItems(
    # First tab content
    tabItem(tabName = "upload",
      h2("Upload the file you want to use"),
      fluidRow(
        box(fileInput('file1', 'Choose CSV File',
          accept=c('text/csv',
                    'text/comma-separated-values,text/plain',
                    '.csv')),
          tags$br(),
          checkboxInput('header', 'Header', TRUE),
          radioButtons('sep', 'Separator',
            c(Comma=',',
              Semicolon=';',
              Tab='\t'),
            ',')),
        box(tableOutput('contents'))
      )
    )
  )

```

Figura 49: Exemplo de código para criação do corpo do separador *Upload File*.

Para se trabalhar num dos separadores previamente criados é necessária a função *tabItems* e passar-lhe o respetivo *nameID* (*tabName*). A partir do momento que é passada esta função com o respetivo parâmetro já é possível trabalhar/editar, neste caso, o separador *Upload File*. De seguida são passados os respetivos comandos de *widgets* a utilizar, o seu título dentro do separador (*h2*), o *output* que deverá ser colocado (posteriormente criado na componente *server.R*) e neste caso a função para *upload* de ficheiros do próprio *shiny*.

No final, obtém-se então um *tabItem* para cada separador, onde o seu corpo será criado e chamado o *output* que se deseja para cada secção.

4.3.2 *server.R*

Como referido anteriormente, nesta secção é abordada a parte funcional da aplicação. Pelo que as instruções para o que deve ser feito e a produção do *output* é toda processada dentro desta componente.

Inicialmente houve a necessidade de criar uma variável reativa para os *datasets* que serão introduzidos na aplicação, isto quer dizer, que depois da variável ser reativa poderá ser utilizada em qualquer outra função consoante os dados que são introduzidos no menu de *upload* de ficheiros. Na Figura 50 é possível verificar o código utilizado para esta fração da aplicação.

```
server <- shinyserver(function(input, output, session) {

  data <- reactive({
    req(input$file1)

    inFile <- input$file1

  })

  df <- read.csv(inFile$datapath, header = input$header, sep = input$sep)

  updateSelectInput(session, inputId = 'response', label = 'Response:',
                    choices = names(df), selected = names(df))
  updateSelectInput(session, inputId = 'prediction', label = 'Prediction:',
                    choices = names(df), selected = names(df)[2])
  updateSelectInput(session, inputId = 'addresponse', label = 'Add Response:',
                    choices = names(df), selected = names(df))
  updateSelectInput(session, inputId = 'addprediction', label = 'Add Prediction:',
                    choices = names(df), selected = names(df)[2])
  updateSelectInput(session, inputId = 'response1', label = 'Response1:',
                    choices = names(df), selected = names(df))
  updateSelectInput(session, inputId = 'prediction1', label = 'Prediction1:',
                    choices = names(df), selected = names(df)[2])
  updateSelectInput(session, inputId = 'response2', label = 'Response2:',
                    choices = names(df), selected = names(df))
  updateSelectInput(session, inputId = 'prediction2', label = 'Prediction2:',
                    choices = names(df), selected = names(df)[2])

  return(df)
})
```

Figura 50: Criação de *datasets* reativos na aplicação.

A Figura 50 ilustra o código criado para tornar o *dataset* do separador *Upload File* reativo, com a introdução da função *reactive*. A função *updateSelectInput* define as variáveis que o utilizador terá de escolher nos separadores *One curve* e *Comparasion 2 curves - pROC*. No menu da *One curve* podem ser requisitadas até quatro variáveis do *dataset*, neste caso, a *prediction* e a *response* inicial que irão reproduzir um gráfico e os respetivos parâmetros e também uma *prediction* e uma *response* adicional, caso o utilizador deseje adicionar uma curva ao gráfico, sempre dependentes das condicionantes de cada *package*. No menu de *Comparasion 2 curves - pROC* são dadas quatro novas variáveis, neste caso, para comparação

de curvas através do *package* *pROC* exclusivamente, sendo sempre necessária a introdução de *prediction1*, *prediction2*, *response1* e *response2* para tornar a função deste separador viável.

Houve ainda a necessidade da criação de duas novas funções reativas, referente ao *upload* de ficheiros para o *package* *Comp2ROC*. De forma a tornar possível a comparação com o outro carregamento de dados é demonstrado na Figura 51 a variável criada para tornar estas funções reativas.

```
data2 <- reactive({
  req(input$file2)

  inFile2 <- input$file2

  df2 <- Comp2ROC::read.file(inFile2$datapath, header = input$header2 , sep = input$sep2 , ".", input$pred1 , input$direction1, input$pred2
    ,input$direction2, input$res,T)
  return(df2)
})

data3 <- reactive({
  req(input$file3)

  inFile2 <- input$file3

  df3 <- Comp2ROC::read.file(inFile2$datapath, header = input$header3 , sep = input$sep3 , ".", input$pred3 , input$direction3, input$pred4
    ,input$direction4, input$res3,FALSE,input$res4)
  return(df3)
})
```

Figura 51: Criação de *datasets* reativos na aplicação para o *package* *Comp2ROC*.

Como é possível verificar, houve a necessidade de uma introdução de dados completamente distinta, uma vez que o *Comp2ROC* utiliza uma função própria para realizar este carregamento. Sendo assim, foi enquadrada a função *read.file* do *package* *Comp2ROC* numa função reativa. Dentro do *Comp2ROC* os dados de amostras dependentes e independentes embora utilizem a mesma função não são criados com os mesmos parâmetros, daí a necessidade de se criar duas variáveis reativas distintas para este mesmo *package*, com os nomes de *data2* e *data3*.

Relativamente ao código que falta desta componente foram utilizadas funções *Render* para criar o *output* desejado em cada secção. Por exemplo para o separador da *checklist* foi utilizada a função *renderTable* em que foi criada uma tabela por via de uma matriz com o respetivo conteúdo para cada linha. Para imprimir o texto no separador das *User Guides* foi utilizada a função *renderUI*.

Por fim, possivelmente as duas funções mais importantes para quem pretende utilizar esta aplicação, a *renderPlot* e a *renderPrint*. A *renderPlot* é onde são introduzidas as funções de criação dos gráficos da aplicação e a *renderPrint* é referente à devolução de valores de *output* como por exemplo a *AUC*, que é calculada por uma função específica dentro desta mesma função. Para contextualizar estes comandos é apresentado o código utilizado na criação do separador *Comparasion 2 curves - pROC* na Figura 52.

É possível verificar para este exemplo, o código para o *package* *pROC* que compara duas curvas. Foi introduzido nestas duas funções tornando assim o seu *output* visível para o utilizador consoante as respetivas variáveis criadas anteriormente no *dataset* reativo.

```

output$procomp <- renderPrint({
  x <- data()[, c(input$prediction1, input$response1, input$prediction2, input$response2)]
  roc1=pROC::roc(x[[input$response1]], x[[input$prediction1]])
  roc2=pROC::roc(x[[input$response2]], x[[input$prediction2]])
  pROC::roc.test(roc1,roc2)
})

output$proc2curves = renderPlot({
  x <- data()[, c(input$prediction1, input$response1, input$prediction2, input$response2)]
  roc3=roc(x[[input$response1]], x[[input$prediction1]],plot=T)
  roc3=roc(x[[input$response2]], x[[input$prediction2]],plot=T,ci=TRUE,col=2, add=T)
})

```

Figura 52: Código exemplo para criação do *output* por via do *renderPlot* e *renderPrint*.

RESULTADOS

De acordo com a análise exploratória dos diferentes *packages/software* irá ser apresentado neste capítulo uma *checklist* de acordo com métricas criadas para o efeito de avaliação dos mesmos na análise *ROC*.

5.1 ELABORAÇÃO DA *checklist*

Como abordado inicialmente, um dos objetivos finais para a dissertação seria a elaboração de uma *checklist* que servisse como um meio de orientação para a escolha de um *package* R ou *software* consoante as suas especificidades para análise através de curvas *ROC*. Após serem executados os comandos para cada uma das ferramentas com a utilização dos *datasets* já explorados anteriormente, foi possível obter a *checklist* final apresentada na Tabela 4.

Tabela 4: Tabela com a *checklist* realizada para todas as ferramentas.

	pROC	ROSE	ROCR	Comp2ROC	verification	Epi	ROC (Bioconductor)	caTools	STATA
Versão	1.16.1	0.0-3	1.0-7	1.1.4	1.42	2.40	1.62.0	1.18.0	16
Última atualização	2020	2015	2015	2016	2015	2019	2020	2020	2019
Curva empírica	✓	✓	✓	✓	✓	✓	✓	✓	✓
AUC	✓	✓	✓	✓	✓	✓	✓	✓	✓
Colocar mais que uma curva num gráfico	✓	✓	✓	✓	✗	✗	✓	✗	✓
Comparar curvas	✓	✗	✗	✓	✗	✗	✗	✗	✓
IC (Intervalos de Confiança)	✓	✗	✗	✓	✓ ^a	✗	✗	✗	✓
Smooth(ajuste)	✓	✗	✗	✗	✗	✗	✗	✗	✓
Trabalha com escalas invertidas	✓	✓	✗	✓	✗	✗	✗	✓	✗
AUC parcial	✓	✗	✗	✗	✗	✗	✓	✗	✗
Traçar apenas uma curva ROC	✓	✓	✓	✗	✓	✓	✓	✓	✓

^a Trabalha com intervalos de confiança mas o formato dos dados introduzidos não é compatível com os dados utilizados nesta dissertação

Nos subcapítulos seguintes será feita a explicação para os critérios adotados e a análise crítica realizada relativamente aos resultados obtidos.

5.1.1 Critérios de cotação

Como é possível verificar na Tabela 4 apresentada anteriormente, existem onze critérios diferentes, embora dois deles, versão e última atualização não estejam intimamente ligados com a análise *ROC*.

A versão e a última atualização são aqui apresentadas de forma a contextualizar o utilizador num panorama geral sobre qual a frequência com que o *package* é atualizado, se ainda existe alguém a supervisionar possíveis erros e o número de vezes que o *package* já sofreu atualizações. Sendo assim relativamente ao primeiro critério, uma versão baixa pode estar intimamente ligada com um *package* que sofre poucas atualizações ou que foi criado há pouco tempo. Como a maioria dos *packages* utilizados nesta dissertação já foram criados há alguns anos, os que possuem uma versão baixa estão definitivamente a cair em desuso e pouco atualizados. Já o segundo critério, data o ano em que o *package* foi atualizado pela última vez e serve como uma forma de completar a informação adquirida no primeiro critério. Uma data de atualização antiga juntamente com uma versão baixa serve para inferir que o *package* foi abandonado numa fase muito primordial.

Relativamente aos dois critérios seguintes, a curva empírica e a *AUC*, são possivelmente os critérios mais importantes para uma análise *ROC* e estes critérios referem-se simplesmente à capacidade de cada ferramenta de processar a curva e a respetiva área abaixo da mesma.

O critério de colocar mais que uma curva num gráfico foi tido em conta, devido à importância que existe em comparar análises *ROC*. E para que esta comparação tenha um efeito mais alargado e de comparação visual é importante que todas as ferramentas possibilitem a inclusão de mais que uma curva num gráfico, facilitando ao utilizador a comparação de curvas.

Já o critério de comparar curvas, refere-se à comparação de parâmetros estatísticos associados à análise *ROC*. Sendo assim e para não induzir em erro, foi feita a separação entre colocar mais que uma curva *ROC* num gráfico e existir a comparação de diferentes parâmetros realizada pelo próprio *software*. A ferramenta pode permitir a colocação de mais de uma curva num mesmo gráfico mas não exibir parâmetros de comparação, ou vice-versa.

Os *IC's* estão associados à possibilidade do utilizador conseguir aceder a *IC's* para as medidas estatísticas fornecidas pela ferramenta. Ou seja, se são calculados *IC's* para medidas como a *AUC* por exemplo.

O *smooth* ou ajuste da curva é relativo a como o próprio nome indica ao *smoothing* da curva, onde a curva é apresentada alisada de acordo com um ajuste específico.

Para trabalhar com escalas invertidas foi tido em conta a existência de funcionalidades próprias das ferramentas que trabalhem com escalas invertidas de forma automática. É possível de forma manual efetuar a inversão de escalas, contudo é importante distinguir

ferramentas com funcionalidades automatizadas para o efeito, daquelas que necessitam de intervenção no *dataset* por parte do utilizador.

A *pAUC*, tal como o nome indica, é relativo ao cálculo de uma parte da área abaixo da curva. Sendo assim, é escolhido um ponto das abcissas que define uma linha vertical até onde a área será calculada naquela curva.

E por último, o critério de traçar apenas uma curva que serve para diferenciar ferramentas que apenas possuam a capacidade de comparar curvas *ROC* e traçá-las em simultâneo num mesmo gráfico.

5.1.2 *Análise crítica*

Após explicados cada um dos critérios é importante realizar uma interpretação dos dados extraídos da elaboração desta tabela. Os dois primeiros critérios distinguem-se dos restantes por serem valores informativos. Os restantes nove critérios da *checklist* interpretam-se pelos valores de ✓ caso o critério seja satisfeito pela ferramenta e se não for satisfeito o mesmo aparece marcado com um ✗.

Como é perceptível num primeiro impacto, pode-se considerar o *package pROC* do ambiente de trabalho R como a ferramenta mais amplamente adaptada para realizar uma análise *ROC* abrangente. Todos os nove critérios são satisfeitos por esta ferramenta, o que revela um grande índice de competência por este *package*. Também é possível detetar que o *package Epi* é aquele que se apresenta mais limitado relativamente aos critérios em estudo.

No que se refere à análise dos dois primeiros critérios e como explicado anteriormente, os mesmos refletem o grau de atualização da ferramenta. Como é possível verificar pelos valores demonstrados, o *package ROSE* para além de possuir uma versão bastante baixa a sua última atualização é referente ao ano de 2015. Isto leva-nos a inferir que o mesmo está ainda numa versão muito inicial e deixado possivelmente ao "abandono". Existem ainda outros *packages* que a sua última atualização não é propriamente recente, como é o caso do *ROCR*, *Comp2ROC* e *verification*, contudo, é perceptível pelo número da versão que o mesmo já foi sujeito a algumas atualizações e deixado possivelmente no seu ponto "ótimo".

É possível verificar que todo o *software* em estudo satisfaz os dois critérios relativos a traçar a curva empírica e ao cálculo da *AUC*, que são os dois critérios principais para se realizar uma análise *ROC*.

Relativamente ao critério de colocar mais que uma curva num gráfico, mesmo que não sejam devolvidas quaisquer comparações em termos de dados estatísticos, o utilizador fica possibilitado à comparação visual e comportamental de duas ou mais curvas. Este critério apenas não foi satisfeito pelos *packages verification*, *Epi* e *caTools* devido à sua especificidade gráfica, onde não possuem comandos para anexar novas curvas ou a possibilidade de aceder

ao parâmetro $add=T$ do R. Os restantes encontram-se capacitados para colocar mais que uma curva num gráfico por via de comandos próprios ou do parâmetro $add=T$.

Já o parâmetro referente a comparar curvas *ROC*, é apenas satisfeito pelo *Comp2ROC*, *pROC* e pelo *software STATA*. Em todos eles são fornecidos dados distintos na respetiva comparação de curvas, todos eles com relevância a nível estatístico para comparação de curvas *ROC*. No caso do *Comp2ROC* são devolvidos os valores de ambas as amostras para *AUC*, erro padrão, área pelo método trapezoidal e *IC*'s. Posteriormente é devolvido o coeficiente de correlação entre as áreas, os testes de diferenças (*z-stats* e *p-value*) bem como dos *IC*'s e o valor da soma de diferença das áreas, o último parâmetro demonstrado é o número de vezes que há cruzamento entre as duas curvas no gráfico. Já para o *pROC* apenas são devolvidos os valores de χ^2 , graus de liberdade (*df*) e o *p-value*, bem como os valores das *AUC*'s para as duas curvas em análise. Por fim, no *STATA* são devolvidos valores como o número de observações, *AUC*, erro padrão e *IC*'s para as diferentes curvas.

IC's são devolvidos, como já referido anteriormente, pelo *Comp2ROC*, *pROC* e *STATA*, mas também pelo *package verification*, sendo que este último não foi possível testar devido às especificidades de introdução dos dados quando calculado este parâmetro.

O *smoothing* é realizado apenas pelo *pROC* e pelo *STATA*, sendo este um dos critérios que diferencia estas ferramentas das restantes.

Relativamente ao critério de trabalhar com escalas invertidas foi tido em conta apenas as ferramentas que possuam algum tipo de funcionalidade capaz de trabalhar com estas escalas, de ter em conta que é possível trabalhar em todas as ferramentas com escalas invertidas, contudo é necessário proceder a conversão dos valores do *dataset* para valores negativos de forma manual. Os *packages* que não precisam desta conversão e portanto possuem uma funcionalidade própria para trabalhar com escalas invertidas são o *pROC*, *ROSE*, *Comp2ROC* e *caTools*.

O próximo critério é a *pAUC* e é relativa à possibilidade de o utilizador trabalhar com uma zona restrita do gráfico e obter a sua área, esta funcionalidade é apenas possível com *pROC* e *ROC (Bioconductor)*.

Por fim o critério de processar apenas uma curva, embora menos significativo, mas que permite distinguir *Comp2ROC* das restantes ferramentas, uma vez que o *Comp2ROC* se encontra capacitado para trabalhar apenas com duas curvas, ou seja, se um utilizador quiser utilizar uma curva de cada vez não o poderá fazer porque o *Comp2ROC* não possui essa funcionalidade. Todas as restantes ferramentas possuem a capacidade por defeito de gerar apenas uma curva.

5.2 CONSTRUÇÃO DO *score* DE AVALIAÇÃO DAS FERRAMENTAS ANALISADAS

Tendo como base o trabalho de Stephan et al. (Stephan et al., 2003) na avaliação dos programas que efetuam análise *ROC*, também nesta dissertação será realizada uma avaliação qualitativa das ferramentas utilizadas, visando assim completar a informação presente na *checklist* elaborada anteriormente.

5.2.1 Critérios de construção

Serão utilizado seis critérios distintos para realização da avaliação crítica. Neste subcapítulo estes seis critérios serão explicados de forma breve para o seu devido enquadramento.

Relativamente à entrada de dados, embora apenas o *STATA* varie relativamente às demais ferramentas no aspeto do menu e da forma que se introduz na linha de comandos, dentro do R os *packages* também terão avaliações distintas que se referem a forma como os dados são introduzidos nas respetivas funções, sendo este o principal aspeto a ter em conta nas diferentes avaliações das ferramentas.

Na saída de dados são tidas em conta diversas características como o aspeto gráfico, a quantidade e qualidade dos testes estatísticos e ainda a velocidade de processamento dos resultados.

No parâmetro da análise de resultados, é tido em conta a quantidade de valores fornecidos relativamente ao gráfico processado e aos valores estatísticos relacionados à análise *ROC*, como exemplo *AUC*, *pAUC*, *smoothing*, *IC* e outras medidas já abordadas anteriormente.

O manual está relacionado com a disponibilização de um documento explicativo por parte do *package/software*. Será ainda tido em conta se o manual disponibilizou todas as explicações necessárias para o mecanismo da ferramenta.

A facilidade de uso está relacionado como o nome indica, com a facilidade com que a ferramenta foi utilizada. Sendo assim, foram tidos em conta fatores como facilidade de implementação de funções, facilidade de interpretação de resultados, se o manual se encontra disponibilizado de forma concisa e coerente e ainda se é necessário o pagamento de licenças para a sua utilização.

Por fim, no último critério de funcionalidades, está baseado no número de *checks* (✓) que estão presentes na *checklist* para cada uma das ferramentas.

5.2.2 Avaliação das ferramentas

Após explicados os critérios tidos em conta, é importante também criar uma medida de avaliação, essa mesma atribuída consoante apresentado na Tabela 5.

Tabela 5: Métricas de avaliação das ferramentas ROC.

0	muito mau / muito difícil
1	mau / difícil
2	médio
3	bom / fácil
4	muito bom / muito fácil
5	excelente

A métrica criada neste trabalho para a avaliação das ferramentas está cotada de 0 a 5, sendo o valor 5 o representativo da melhor avaliação. No final foi obtida uma tabela com um registo pessoal de avaliação das ferramentas, tendo em conta a experiência obtida na sua utilização no decorrer da dissertação. A Tabela 6, é o resultado final desta componente prática.

Tabela 6: Tabela da avaliação detalhada das ferramentas ROC.

	pROC	ROSE	ROCR	Comp2ROC	verification	Epi	ROC (Bioconductor)	caTools	STATA
Entrada de dados	4	4	3	3	4	4	4	4	4
Saída de dados	4	3	3	4	3	3	3	3	4
Análise de resultados	4	3	3	5	3	3	3	3	4
Manual	5	4	5	5	4	5	5	5	5
Facilidade de uso	5	4	4	4	5	5	5	5	4
Funcionalidades	5	3	2	4	2	1	3	2	4
TOTAL	27	21	20	25	21	21	23	22	25

Relativamente à entrada de dados, os *packages* *ROCR* e *Comp2ROC* acabam por ter uma cotação distinta das restantes ferramentas por apresentarem uma metodologia de introdução de dados nas suas funções um pouco mais complexa, não sendo tão acessível a novos utentes. Contudo, a leitura do *CRAN* será suficiente para tornar toda a análise *ROC* funcional por meio destes *packages*.

Para a saída de dados *pROC*, *Comp2ROC* e *STATA* apresentam-se com o valor mais elevado. Estes distinguem-se claramente das restantes ferramentas devido à maior diversidade de valores apresentados, *pROC* acaba por não obter a cotação máxima por não apresentar um gráfico tão atrativo e com alguns defeitos nas coordenadas e o *Comp2ROC* por não ter a capacidade de gerar apenas uma curva num gráfico e ter algum *delay* no processamento dos seus resultados.

No critério da análise de resultados, o *Comp2ROC* teve uma avaliação máxima por possuir uma análise mais abrangente de resultados para as duas curvas. Já o *pROC* e o *STATA*, embora também realizem esta comparação, os parâmetros disponibilizados são diferentes e mais limitados que os apresentados pelo *Comp2ROC*.

Para a avaliação do manual foi atribuído o valor máximo a todas as ferramentas exceto ao *package* *ROSE* e *verification*. O *ROSE* acaba por ter uma avaliação inferior, uma vez que existem dependências no seu *package* que não são anotadas no *CRAN*, como é o caso do facto de trabalhar com escalas invertidas mas sem existir uma justificação explícita. Contudo e acedendo ao seu código exteriormente ao *CRAN* foram detetadas possíveis dependências ao *package* *pROC* nesta funcionalidade, essas mesmas dependências não estão anotadas no *CRAN*. Já o *verification* apresenta-se limitado na explicação da forma de como devem ser utilizados e formatados os dados para ser possível o cálculo de *IC*'s.

O parâmetro da facilidade de uso vai um pouco de encontro à entrada de dados e à coerência do seu manual, o que faz com que os *packages* *ROSE*, *ROCR* e *Comp2ROC* saiam ligeiramente penalizados. Já o *software* *STATA* acaba por ter um valor mais baixo por ser necessário o pagamento de licenças para usufruir do mesmo.

Para as funcionalidades foram atribuídas cotações associadas ao número de *checks* apresentados na Tabela 4. Sendo assim o *pROC* é o único a possuir a pontuação máxima.

5.3 APLICAÇÃO *shiny* PARA SELEÇÃO DE *packages* R

Tendo em conta os objetivos definidos e o abordado no Capítulo 4, foram iniciados os trabalhos para a criação de uma aplicação *shiny*, que visasse a orientação de utilizadores da metodologia das curvas *ROC* na seleção de *packages* dentro do ambiente de trabalho *R*. Sendo assim, todo o trabalho explorado anteriormente serve para a criação desta aplicação, uma vez que a mesma estará organizada de acordo com aquilo que é demonstrado na *checklist* para o *R*.

O nome escolhido para a aplicação criada foi *ROSY* tendo em conta o gerador de acrónimos *acronymify* disponível online no link www.acronymify.com e utilizando as palavras chaves: *ROC* e *shiny*.

5.3.1 Fluxograma da aplicação *shiny*

Para facilitar a idealização desta aplicação, torna-se fundamental que inicialmente seja projetado um modelo para organizar a forma de como esta poderá ser criada. Para o efeito foi construído um fluxograma com a principal ordem de ideias para esta parte do trabalho. Na Figura 53 está então representada a forma de como a aplicação será criada dentro do ambiente de trabalho *R*.

Como é possível averiguar, inicialmente será dado ao utilizador dois separadores que representarão a *checklist* (exceto a coluna do *STATA* e do *verification*) criada no subcapítulo 5.1 e um apoio ao utilizador relacionado com o que a aplicação disponibiliza. O *package*



Figura 53: Fluxograma para a criação da aplicação *shiny*.

verification não foi introduzido na aplicação devido às suas fragilidades em processar os IC, pelo que se optou por deixar em *stand by* a sua implementação.

No que diz respeito às entradas de dados na aplicação, o próprio utilizador deverá seleccionar consoante o objetivo da análise, para o efeito terá de carregar dados de forma distinta para o *Comp2ROC*, uma vez que os dados para este *package* são introduzidos de forma distinta dentro do R. Ainda dentro do *Comp2ROC* há a necessidade de uma bifurcação desta introdução, relacionada com amostras dependentes e amostras independentes, que também veem os seus dados introduzidos de forma distinta devido ao facto de amostras dependentes necessitarem apenas de uma variável resposta e as independentes de duas. Já nos restantes *packages*, o carregamento dos dados é efetuado da mesma forma para todos. O *package pROC* terá ainda um separador próprio, uma vez que este realiza a comparação de duas curvas *ROC* em alguns parâmetros.

Para obter um *output* de acordo com as necessidades do utilizador é crucial que este tenha conhecimento dos dados que está a introduzir, de forma a que os valores pedidos em cada uma das secções para *prediction* e *results* estejam orientados da forma correta.

5.3.2 Checklist e Userguide

De forma a orientar quem desejar trabalhar dentro desta aplicação, foram criadas duas secções que servem como um guia para os diferentes utilizadores.

A primeira secção é referente à *checklist*, a mesma explorada anteriormente e que permite a um utilizador visualizar num panorama geral as funcionalidades de cada um dos *packages* presentes nesta aplicação. De forma a contextualizar o aspeto que toma esta *checklist* dentro da aplicação este separador é apresentado na Figura 54.

	pROC	ROSE	ROCR	Comp2ROC	Epi	ROC(Bioconductor)	caTools
Version	1.16.1	0.0-3	1.0-7	1.1.4	2.40	1.62.0	1.18.0
Last update	2020	2015	2015	2016	2019	2020	2020
Available in cran	yes	yes	yes	yes	yes	no	yes
Empirical curve	yes	yes	yes	yes	yes	yes	yes
AUC (Area under the curve)	yes	yes	yes	yes	yes	yes	yes
More than a curve in a graph	yes	yes	yes	yes	no	yes	no
Compare curves	yes	no	no	yes	no	no	no
CI (Confidence Intervals)	yes	no	no	yes	no	no	no
Smooth curve	yes	no	no	no	no	no	no
Works with inverted scales	yes	yes	no	yes	no	no	yes
Partial AUC	yes	no	no	no	no	yes	no
Print only one ROC curve	yes	yes	yes	no	yes	yes	yes

Figura 54: Checklist para orientação do utilizador dentro da ROSY.

Como se pode verificar, o *pROC* satisfaz todas as funcionalidades, contudo é importante que o utilizador explore detalhadamente cada um dos *packages* de forma a que o que pretenda seja fornecido de forma mais satisfatória. A importância de existirem diversos *packages* nesta aplicação faz com que o utilizador consiga obter os mesmos resultados através de meios diferentes mas de forma rápida e eficaz.

Relativamente à secção de *User Guides*, são fornecidas hiperligações relativas ao [CRAN](#) e também aos documentos *Bioconductor*, que permitem ao utilizador ser reencaminhado diretamente para o documento explicativo do *package* dentro do *browser*. Ainda dentro desta secção, noutra caixa, é fornecida diversas notas relativas às particularidades dos *packages* dentro da aplicação, permitindo assim guiar o utilizador no manuseamento de cada uma das funcionalidades. Estas especificidades podem ser verificadas na Figura 55.

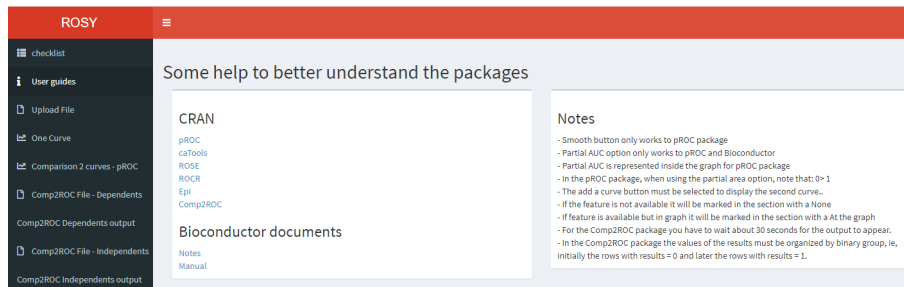


Figura 55: *User Guide* para orientação do utilizador.

O utilizador deverá consultar esta secção ao longo da utilização da aplicação, para que possa mais facilmente seleccionar as funcionalidades adequadas ao fim desejado

5.3.3 Upload de ficheiros

Como explicado no fluxograma, para serem carregados os dados foi necessária a criação de diferentes abas devido às especificidades de carregamento de dados do *package Comp2ROC*. Os documentos para a aplicação são apenas carregados no formato *.csv* e *.txt*. Serão agora explicadas cada uma das abas de *upload* de ficheiros na aplicação.

Inicialmente tratando do carregamento de dados mais geral (todos excepto o *Comp2ROC*), foi extraído o código disponibilizado pelo próprio *Rstudio* para carregamento de dados no *shiny*, sendo posteriormente alterado consoante as necessidades para a aplicação. Na Figura 56 é possível verificar o aspeto que este *upload* de ficheiros toma dentro da aplicação.

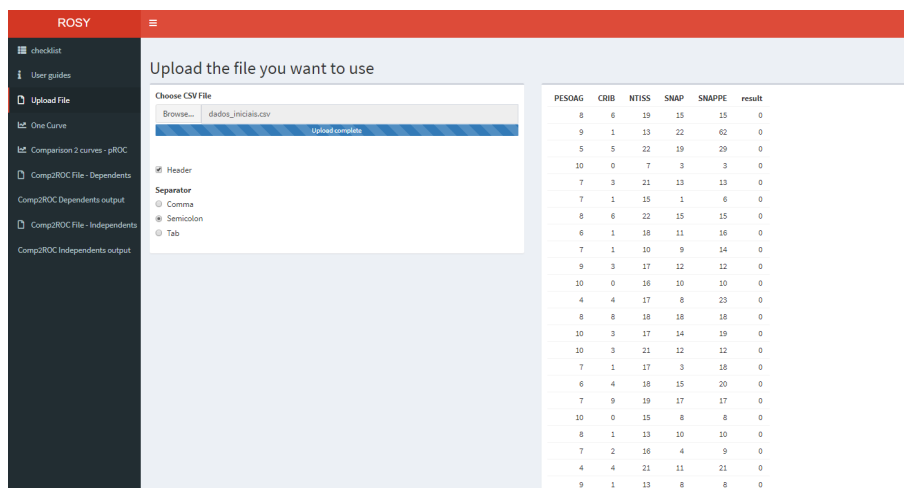


Figura 56: Ficheiro carregado para todos os *packages* exceto *Comp2ROC*.

Neste caso, o método de carregamento para o utilizador é bastante acessível, sendo que, será apenas necessário carregar no botão *"Browse..."* e procurar o documento *.csv* na diretoria em que se encontrar. O utilizador deve ainda referir se a primeira linha do ficheiro é o

cabeçalho ou não e qual o tipo de separador (virgula, ponto e virgula ou contrabarra). De referir então, que este *upload* de ficheiros está apto para os *packages* *pROC*, *ROSE*, *ROCR*, *EpiROC*(*Bioconductor*) e *caTools* ficarem funcionais em toda a aplicação.

Relativamente ao carregamento de dados para as duas componentes do *Comp2ROC* (amostras dependentes e independentes), foram criadas duas diferentes secções para permitir ao utilizador trabalhar em simultâneo com diferentes conjuntos de dados. Sendo assim, os *datasets* serão carregados nas abas "Comp2ROC File - Dependents" e "Comp2ROC File - Independents". Como explicado anteriormente, a entrada de dados para o *Comp2ROC* é distinta dos restantes *packages*, sendo necessário agilizar o código para tornar viável estas duas secções. As Figuras 57 e 58 são representativas do aspeto do carregamento de dados para as duas componentes do *Comp2ROC*.

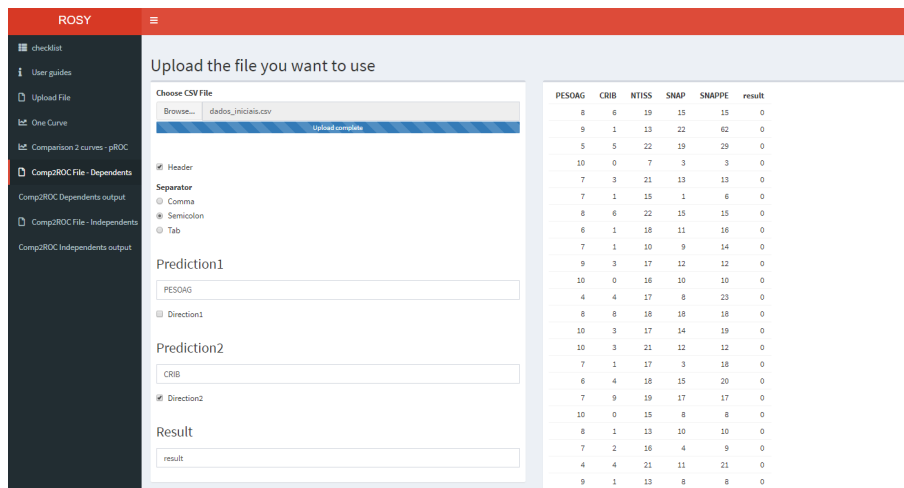


Figura 57: Carregamento de dados para amostras dependentes no *package* *Comp2ROC*.

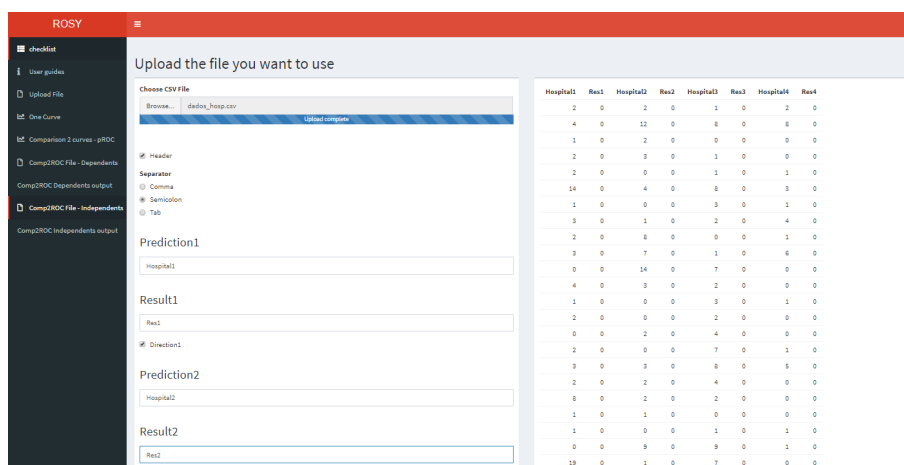


Figura 58: Carregamento de dados para amostras independentes no *package* *Comp2ROC*.

À semelhança do outro carregamento de dados, também para o *Comp2ROC* é necessário fornecer à aplicação algumas informações básicas como: haver cabeçalho e qual o separador utilizado pelo ficheiro. A diferença neste *input* de dados surge então no facto de ser necessário fornecer os valores para as *predictions* e *results* diretamente no carregamento de dados, que irão permitir gerar o *output* desejado. No caso das amostras dependentes terão de ser fornecidos os nomes do cabeçalho para as duas *predictions* e qual o nome do cabeçalho para os resultados dentro do ficheiro. Já as amostras independentes necessitam destes mesmos valores, com a particularidade de ser necessário os nomes para cada *response* separadamente, por estas mesmas serem amostras independentes. Existe ainda um botão designado "*direction*" que tem em vista trabalhar com escalas invertidas dentro do *Comp2ROC*, sendo assim, se nesta caixa estiver um *check* (✓), significa que a escala não está invertida e se não existir um *check* na caixa, estamos a trabalhar com uma escala invertida. Dentro do *dataset* explorado neste trabalho apenas a amostra PESOAG é necessária a remoção deste *check*, tal como se pode verificar na Figura 57.

Os dados carregados em todas as secções serão posteriormente introduzidos numa função reativa, de forma a que se possa trabalhar com os dados noutras secções e com diferentes funções aplicadas aos valores desejados.

5.3.4 *Output para uma curva*

Visto que o *upload* de ficheiros é dado de forma separada, foi decidido também colocar o *output* separadamente de forma a que a aplicação não fique demasiado robusta em determinadas secções. Sendo assim, inicialmente teremos o *output* para uma curva que é dado por todos os *packages* excepto o *Comp2ROC*, sendo que este apenas realiza análise *ROC* com duas curvas.

Para o *output* com uma curva são disponibilizadas diferentes caixas de seleção. Inicialmente as três principais referentes ao *package*, *prediction* e *response*. Na primeira caixa o utilizador seleciona o *package* com que deseja trabalhar e estão disponíveis então as seguintes opções: *pROC*, *ROSE*, *ROCR*, *caTools*, *Epi* e *Bioconductor*. De seguida, o utilizador seleciona a *prediction* e a *response* que estarão disponibilizadas após o carregamento do *dataset* na aba destinada. A partir deste ponto, já deverá começar a ser devolvido algum *output* como o gráfico e *AUC*.

Posteriormente, mais algumas funcionalidades aparecem, temos o *smoothing* (apenas aplicável ao *pROC*) e a *pAUC* (para *pROC* e *Bioconductor*).

No final existe a possibilidade do utilizador colocar mais uma curva no gráfico, embora não seja realizada a comparação de outros parâmetros permite ao utilizador uma visualização conjunta de curvas. Devido às especificidades dos *packages*, esta funcionalidade apenas está apta nesta secção para *ROSE*, *ROCR*, *Bioconductor* e *pROC*. Posteriormente, noutra separador

será possível repetir a mesma funcionalidade para o *pROC*, que será um *package* analisado mais detalhadamente à parte por conseguir realizar comparação de curvas.

Na Figura 59 é possível verificar o aspeto que esta secção toma para introduzir dados.

The screenshot shows the ROSY application interface. On the left is a dark sidebar with a menu containing: checklist, User guides, Upload File, One Curve (highlighted), Comparison 2 curves - pROC, Comp2ROC File - Dependents, Comp2ROC Dependents output, Comp2ROC File - Independents, and Comp2ROC Independents output. The main content area is titled 'Show just one curve' and contains several input fields: 'Package' (dropdown menu with 'Bioconductor' selected), 'Prediction:' (dropdown menu with 'CRIB' selected), 'Response:' (dropdown menu with 'result' selected), a checkbox for 'Smooth (only to pROC)', and a text input for 'Partial AUC value (between 0 and 1)' with '0.5' entered. Below this is a section 'For add a new curve in graph:' with a checked checkbox 'add a curve', an 'Add Prediction:' dropdown menu with 'PESOG' selected, and an 'Add Response:' dropdown menu with 'result' selected.

Figura 59: Caixas de texto de *input* de valores para *packages* que processam apenas uma curva.

De seguida a serem processados os valores pedidos pelo utilizador são fornecidos valores de *output* tal como os representados no exemplo da Figura 60.

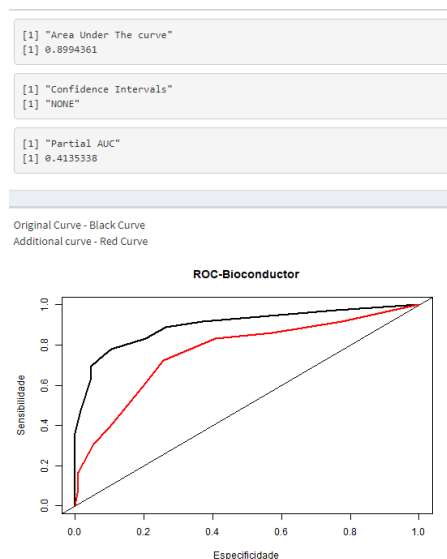


Figura 60: Exemplo de *output* fornecido pela aplicação.

5.3.5 *Output* *pROC* - Comparar duas curvas

Uma vez que o *pROC* possui uma função que realiza a comparação de alguns parâmetros de duas curvas, foi decidido dar especial relevo a este *package* ao criar uma secção própria para esta componente do *pROC*. Sendo assim, nesta aba o utilizador apenas terá de fornecer à aplicação os valores das duas *predictions* e dos dois resultados, para obter o *output* final que contem o *z-value*, o *p-value* e a *AUC* das duas curvas. Ainda como referido anteriormente, o utilizador tem a possibilidade de visualizar com o *package* a forma que as duas curvas tomam num mesmo gráfico.

Para melhor compreender o resultado final, é apresentado na Figura 61 o formato que este separador toma para entrada e saída de dados.

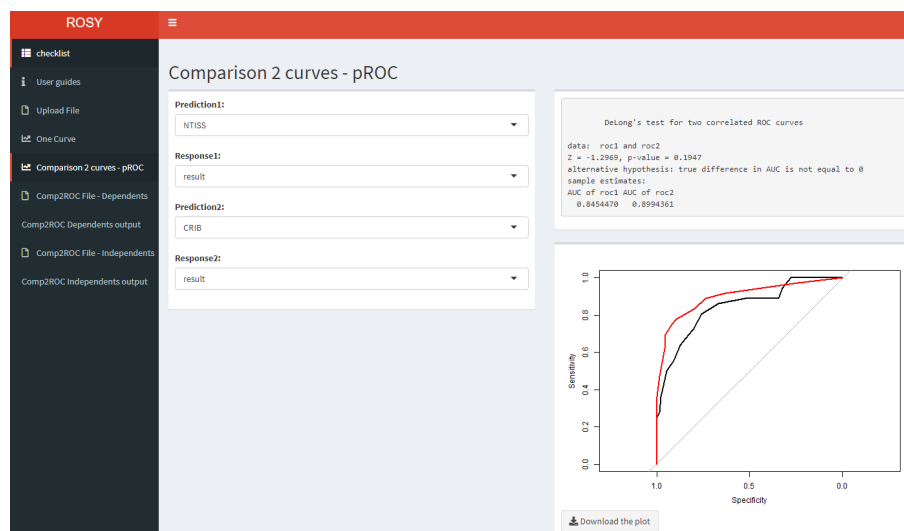


Figura 61: Exemplo do *output* fornecido na aba de comparação de duas curvas com o *package* *pROC*.

5.3.6 *Output* *Comp2ROC* - Amostras dependentes e independentes

Relativamente ao *output* das amostras dependentes para o *package* *Comp2ROC*, este mesmo irá ter um separador destinada ao seu *output*. Todas as informações de interesse serão demonstradas nesta secção após realizar o devido *input* de valores desejados e selecionar o botão "*Run Comp2ROC analysis*" na secção de *output*. O utilizador deverá trocar para o separador que menciona este mesmo *output* assim que desejar processar os valores devolvidos, sendo que, devido ao processamento de dados mais lento do *Comp2ROC* existe um tempo de espera estimado de mais ou menos 30 segundos para que o *output* seja compilado no devido separador. Este *output* deverá ter um formato como o representado nas Figuras 62 e 63.

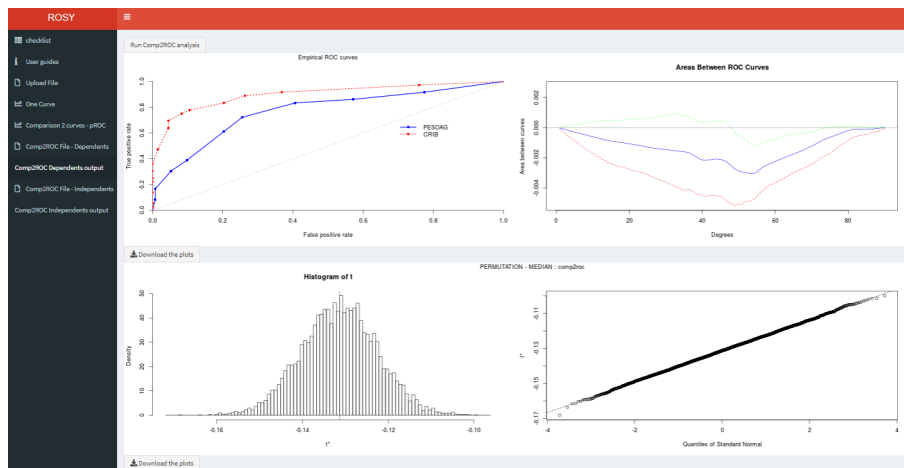


Figura 62: Exemplo do *output* dos gráficos devolvidos pelo *package* *Comp2ROC* na aba para amostras dependentes.

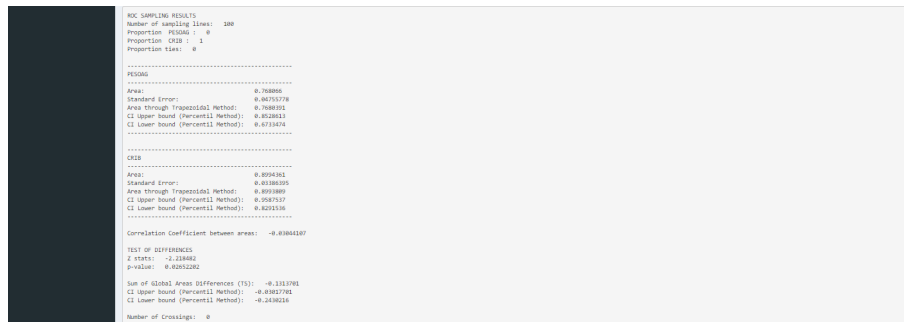


Figura 63: Exemplo do *output* dos diferentes valores estatísticos de comparação de curvas devolvidos pelo *package* *Comp2ROC* na aba para amostras dependentes.

Sendo assim, é possível fornecer diferentes valores de interesse numa análise *ROC* por via do *package* *Comp2ROC*, estando destinado exclusivamente ao estudo por meio de comparação de duas curvas. É apresentado um gráfico com duas curvas *ROC* e ainda três gráficos que visam completar a informação do anterior. Cada um deles com funcionalidades distintas, à direita do gráfico *ROC* apresenta-se uma comparação de valores da área abaixo da curva, seguindo-se um histograma de *t* e um gráfico dos quantis do padrão normal.

Para as amostras independentes o *output* é fornecido exatamente no mesmo formato, sendo que, a única diferença para as amostras dependentes está no *input* e nos valores pedidos pelo mesmo. Serve de demonstração as Figuras ilustrativas 64 e 65 para um exemplo carregado nesta secção.

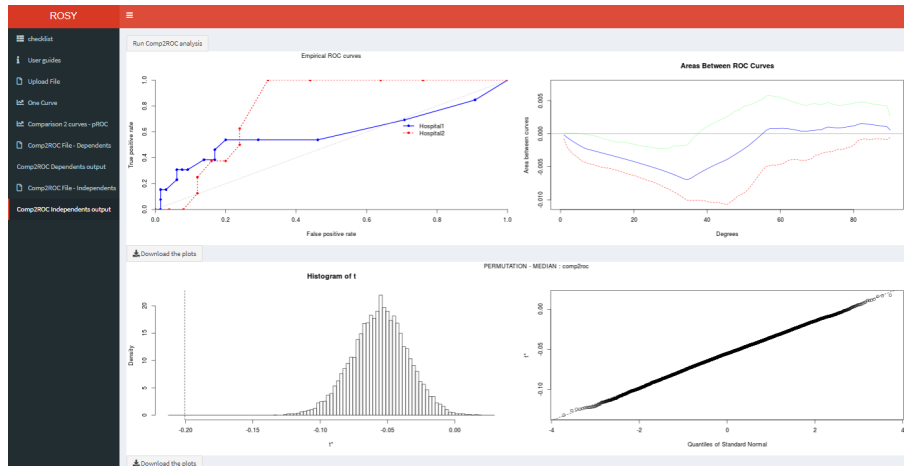


Figura 64: Exemplo do *output* dos gráficos devolvidos pelo *package* *Comp2ROC* na aba para amostras independentes.

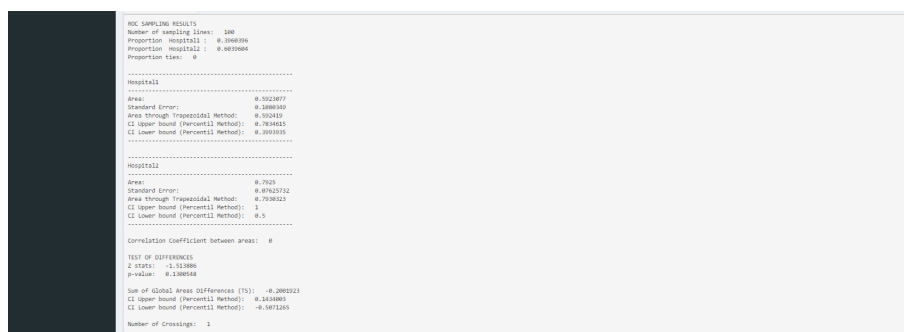


Figura 65: Exemplo do *output* dos diferentes valores estatísticos de comparação de curvas devolvidos pelo *package* *Comp2ROC* na aba para amostras independentes.

CONCLUSÃO

Neste capítulo serão referidas as principais conclusões de todo o trabalho realizado nesta dissertação, tanto na sua componente teórica como na componente prática. Será realizada uma reflexão sobre os temas abordados e ainda projetadas as linhas de orientação para trabalhos futuros que poderão ser realizados de forma a aprimorar o trabalho já efetuado até ao momento. Serão tópicos de interesse assuntos como a *checklist*, a aplicação *shiny*, o R, as curvas *ROC*, o *STATA* e novas ferramentas/*software* para realização de análise *ROC*.

6.1 CONCLUSÕES

Neste trabalho foi proposta a recolha e exploração de diferentes ferramentas/*software* capazes de realizar um estudo que é a análise *ROC*. Para o efeito foram utilizadas diferentes fontes bibliográficas primárias como artigos, teses, relatórios técnicos e livros, onde foi contextualizada a temática *ROC* bem como realizada a respetiva recolha de ferramentas.

Antes de ser processada a primeira etapa prática da dissertação houve a necessidade de testar algumas ferramentas de forma verificar se seriam úteis para comparação relativa à análise *ROC*. Foram por isso testadas as metodologias de diversas ferramentas sendo posteriormente escolhidas nove para serem utilizadas neste trabalho. Em simultâneo, tal como referido anteriormente, a exploração de fontes bibliográficas foi também ela uma forma de recomendação para a seleção das ferramentas.

Nesta primeira etapa da dissertação foi no final gerada uma *checklist* onde as nove ferramentas foram testadas e comparadas em onze critérios distintos. As ferramentas utilizadas foram então os *packages* do R, *pROC*, *ROSE*, *ROCR*, *Comp2ROC*, *verification*, *Epi*, *ROC (Bioconductor)* e *caTools* e ainda o *software STATA*. Estes mesmo foram posteriormente comparados nos onze critérios definidos anteriormente, sendo eles, versão, última atualização, gerar uma curva empírica, cálculo da *AUC*, colocar mais que uma curva num gráfico, comparar curvas, calcular *IC's*, ajuste da curva (*smooth*), trabalhar com escalas invertidas, cálculo da *pAUC* e traçar apenas uma curva *ROC* no gráfico. Após testadas as ferramentas com um *dataset* comum definido para este trabalho, onde foram utilizadas amostras dependentes e

amostras independentes, foi construída uma *checklist* que procurou definir a capacidade de cada uma das ferramentas de uma forma geral. A melhor ferramenta foi então o *package pROC* do ambiente de trabalho R, onde todos os critérios foram realizados com sucesso. Seguiu-se duas ferramentas com boas indicações, cumprindo maioritariamente todas as funcionalidades desejadas, sendo elas o *software STATA* que apenas não trabalha com escalas invertidas e não calcula a *pAUC* e também o *package Comp2ROC* que não realiza ajuste de curva, não calcula *pAUC* e não traça apenas uma curva *ROC* no gráfico. Em sentido contrário a pior ferramenta foi o *package Epi* que apenas realiza o cálculo da *AUC*, traça a curva empírica e consegue traçar apenas uma curva no gráfico, os restantes *packages* embora não estejam tão limitados também não cumprem grande parte das funcionalidades testadas, estando praticamente limitados à realização de uma análise *ROC* com processamento do gráfico e cálculo da *AUC*.

Posteriormente as ferramentas foram avaliadas tendo em conta uma métrica definida para este trabalho, onde são utilizados critérios como a entrada de dados, a saída de dados, a análise de resultados, a facilidade de uso, o manual e as funcionalidades. As ferramentas foram então avaliadas de 0 a 5, onde 0 é a pior classificação e 5 a mais alta, tendo no final uma avaliação detalhada das ferramentas. Tal como expectável esta informação acaba por corroborar a informação presente na *checklist* realizada anteriormente, onde o *pROC* obteve a maior cotação, seguindo-se o *STATA* e o *Comp2ROC*. De notar que nesta avaliação o *package ROCR* acaba por ter a pior classificação, muito em conta à forma que os dados são introduzidos neste *package*, seguindo-se o *Epi* que já na *checklist* tinha demonstrado os piores resultados.

Na última componente prática da dissertação, foi elaborada uma aplicação *shiny* no ambiente de trabalho R, ROSY que está disponível em acesso livre no link <https://pquintasbcl.shinyapps.io/ROSY>. Esta mesma visa então funcionar como uma *library* de seleção de *packages* R, sendo que tudo será processado por via de botões ou caixas de texto em vez de uma linha de comandos. Na aplicação é fornecido aos utilizadores o acesso à *checklist* elaborada neste trabalho (retirados os valores do *STATA* e *verification*), sendo assim, os utilizadores têm uma ideia previamente definida daquilo que poderão fazer com cada um dos *packages* dentro da aplicação. É ainda fornecido uma ajuda ao utilizador dentro de um separador denominado *User guides* com alguma informação útil das funcionalidades dos *packages* dentro e fora da aplicação e ainda hiperligações para os documentos do *CRAN*. Para que fosse possível ter uma melhor organização de ideias relativamente à forma que a aplicação deveria ser criada, foi inicialmente gerado um fluxograma com todas as etapas a percorrer. Foi perceptível desde o início que o *package Comp2ROC*, devido às suas especificidades, teria de ser elaborado de forma distinta dos demais, tendo por isso direito a diferentes separadores dentro da própria aplicação e até uma forma distinta de introduzir os dados do *dataset*. Para realizar o processamento de análise *ROC* de uma curva, o utilizador tem acesso a um

separador no qual define inicialmente qual o *package* a ser utilizado, a partir daí poderá utilizar todas as funcionalidades desta secção. Contudo, será importante o utilizador ter em atenção que determinados *packages* não têm a capacidade de realizar determinadas funções, mesmo elas estando representadas na aplicação, deverão por isso receber um valor nulo ou um erro caso a informação pedida não seja possível de processar. Devido à capacidade do *package pROC* em comparar análises *ROC* em determinados parâmetros foi também criado um separador para esta componente, sendo bastante simples colocar os valores de *input* para obtenção automática da comparação de análises *ROC* e até a inserção das duas curvas num mesmo gráfico. Como referido anteriormente, o *package Comp2ROC*, devido à sua distinta inserção de dados em R teve a sua exploração realizada à parte dos demais, sendo por isso criados separadores diferentes de *input* e *output* para o mesmo. A aplicação permite ainda a exportação dos gráficos processados dentro da aplicação, permitindo ao utilizador o *download* e utilização dos mesmos no próprio computador. O trabalho projetado fica então com os seus objetivos concluídos, sendo que, existem sempre melhorias que podem ser efetuadas tanto à aplicação criada (*ROSY*) bem como no estudo das diferentes ferramentas.

6.2 PERSPECTIVAS PARA TRABALHOS FUTUROS

Relativamente às perspetivas para trabalhos futuros, dentro do trabalho executado existem alguns aspetos que poderão ser objeto de novo estudo. É o caso da utilização de mais e novas ferramentas *ROC* que possam ser de interesse, e compará-las com as demais utilizadas neste trabalho. Funcionalidades que não tenham sido executadas e ou colocadas na aplicação também poderão ser uma melhoria para o trabalho elaborado.

O trabalho presente nesta dissertação pretende ter um alcance a diferentes áreas que necessitem de processar análises *ROC*, relativamente à *checklist* onde foram agrupadas diferentes características de diferentes ferramentas e funcionaria num formato mais informativo e relativamente à aplicação num formato mais interativo onde o utilizador terá a hipótese de processar análises *ROC* com diferentes *packages* R num curto espaço de tempo e sem recorrer a qualquer linha de comando.

Será importante manter uma atualização constante da aplicação *ROSY* de forma a que a mesma tenha relevância para as diferentes comunidades de interesse, para o efeito novos *packages* R podem ser adicionados e deverá ter sido em conta a eventual atualização dos *packages* já implementados e se terão algum impacto nas funcionalidades empregues na aplicação. Esta aplicação devido à sua interatividade característica do *shiny*, poderá ser de interesse para diferentes áreas como da medicina, do ensino, da estatística, da informática e todas aquelas que possam estar relacionadas com avaliação de precisão.

BIBLIOGRAFIA

- Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *Journal of Mathematical Psychology*, 12(4):387 – 415.
- Braga, A. (2000). *Curvas ROC: Aspectos funcionais e aplicações*. PhD thesis.
- Braga, A., Costa, L., and Oliveira, P. (2011). An alternative method for global and partial comparison of two diagnostic systems based on ROC curves. *Journal of Statistical Computation and Simulation*, 83(2):307–325.
- Braga, A., Frade, H., Carvalho, S., and Santiago, A. (2016). *Comp2ROC: Compare Two ROC Curves that Intersect*. R package version 1.1.4.
- Carey, V. and enhancements, H. (2019). *ROC: utilities for ROC, with uarray focus*. R package version 1.60.0.
- Carstensen, B., Plummer, M., Laara, E., and Hills, M. (2019). *Epi: A Package for Statistical Analysis in Epidemiology*. R package version 2.38.
- Cho, W., Lim, Y., Lee, H., Varma, M., Lee, M., and Choi, E. (2014). Big data analysis with interactive visualization using R packages. In *BigDataScience 14*.
- Courcy-Wheeler, R., Wolfe, C., Fitzgerald, A., Spencer, M., Goodman, J., and Gamsu, H. (1995). Use of the CRIB (clinical risk index for babies) score in prediction of neonatal mortality and morbidity. *Archives of disease in childhood. Fetal and neonatal edition*, 73:F32–6.
- Cunha, D. and Braga, A. (2017). Receiver operating characteristic (ROC) packages comparison in R. pages 545–559.
- Doi, J. (2016). Web application teaching tools for statistics using R and shiny. *Technology Innovations in Statistics Education*, 9:1–32.
- Dorfman, D. and Alf, E. (1969). Maximum-likelihood estimation of parameters of signal-detection theory and determination of confidence intervals—rating-method data. *Journal of Mathematical Psychology*, 6(3):487 – 496.
- Fawcett, T. (2003). Roc graphs: Notes and practical considerations for researchers.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition.

- Gonçalves, L., Subtil, A., Oliveira, M., and Bermudez, P. (2014). ROC curve estimation: An overview. *Revstat - Statistical Journal*, 12:1–20.
- Green, D. and Moses, F. (1966). On the equivalence of two recognition measures of short-term memory. *Psychol Bull*, (66):228–234.
- Green, D. and Swets, J. (1966). *Signal detection theory and psychophysics*. Wiley.
- Hand, D. and Till, R. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186.
- Hanley, J. (1989). Receiver operating characteristic (ROC) methodology: the state of the art. *Critical reviews in diagnostic imaging*, 29 3:307–35.
- Hanley, J. and Mcneil, B. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. pages 29–36.
- Hanley, J. and Mcneil, B. (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148 3:839–43.
- Jahanshiri, E. and Shariff, R. (2014). Developing web-based data analysis tools for precision farming using R and shiny. *IOP Conference Series: Earth and Environmental Science*, 20:012014.
- Krzanowski, W. and Hand, D. (2009). *ROC Curves for Continuous Data*. Chapman & Hall/CRC, 1st edition.
- Lunardon, N., Menardi, G., and Torelli, N. (2014). ROSE: a package for binary imbalanced learning. *R Journal*, 6:79–89.
- Metz, C. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283 – 298.
- Metz, C. (1986). ROC methodology in radiologic imaging. *Investigative radiology*, 21:720–33.
- Metz, C. (1989). Some practical issues of experimental design and data analysis in radiological ROC studies. *Investigative radiology*, 24:234–45.
- Metz, C. (2008). ROC analysis in medical imaging: A tutorial review of the literature. *Radiological physics and technology*, 1:2–12.
- Metz, C. and Pan, X. (1999). “proper” binormal ROC curves: Theory and maximum-likelihood estimation. *Journal of Mathematical Psychology*, (1):1 – 33.
- Nakas, C. (2014). Developments in ROC surface analysis and assessment of diagnostic markers in three-class classification problems. *Revstat - Statistical Journal*, 12:43–65.

- NCAR, R. A. L. (2015). *verification: Weather Forecast Verification Utilities*. R package version 1.42.
- Nelson, J., Sklenář, J., Barnes, A., and Minnier, J. (2017). The START app: a web-based rnaseq analysis and visualization resource. *Bioinformatics*, 33:447–449.
- Pardo-Fernandez, J., Rodríguez-Álvarez, M., and Keilegom, I. (2014). A review on ROC curves in the presence of covariates. *Revstat - Statistical Journal*, 12:21–41.
- Pepe, M. (2011). *The Statistical Evaluation Of Medical Tests For Classification And Prediction*. Oxford University Press.
- Pepe, M., Longton, G., and Janes, H. (2009). Estimation and comparison of receiver operating characteristic curves. *The Stata journal*, 9 1:1.
- Rautonen, J., Maekelae, A., Boyd, H., Apajasalo, M., and Pohjavuon, M. (1994). CRIB and SNAP: assessing the risk of death for preterm neonates. *The Lancet*, 343:1272–1273.
- Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J., and Muller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12(1):77.
- Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCr: Visualizing classifier performance in R. *Bioinformatics (Oxford, England)*, 21:3940–1.
- Spackman, K. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160–163, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Spitzer, M., Wildenhain, J., Rappsilber, J., and Tyers, M. (2014). BOXPLOT: A web tool for generation of box plots. *Nature methods*, 11:121–122.
- Stephan, C., Wesseling, S., Schink, T., and Jung, K. (2003). Comparison of eight computer programs for receiver-operating characteristic analysis. *Clinical chemistry*, 49:433–9.
- Swets, J. and Pickett, R. (1982). *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*. Academic Press series in cognition and perception. Elsevier Science & Technology Books.
- Team, R. C. (2019). R language definition. <https://cran.r-project.org/doc/manuals/r-release/R-lang.html>. Accessed: 2019-06-17.
- The-International-Neonatal-Network (1993). The CRIB (clinical risk index for babies) score: A tool for assessing initial neonatal risk and comparing performance of neonatal intensive care units. *Lancet (North American Edition)*, (North American Edition):193–198.

- Todd, A. (2014). Verification bias — impact and methods for correction when assessing accuracy of diagnostic tests.
- Tuszynski, J. (2019). *caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.* R package version 1.17.1.2.
- Wang, W., Hallow, K., and James, D. (2015). A tutorial on RXODE: Simulating differential equation pharmacometric models in R. *CPT: Pharmacometrics and Systems Pharmacology*, 5:n/a–n/a.
- Wojciechowski, J., Hopkins, A., and Upton, R. (2015). Interactive pharmacometric applications using R and the shiny package. *CPT: pharmacometrics and systems pharmacology*, 4:146–159.

