1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

dei25 departamento de engenharia informática 1995 – 2020

Metodologias Experimentais em Informática
2021/22 - 1º Semestre
Meta 3 - Teste das Hipóteses

Cesário Silva [2015230724] - cesariors@student.dei.uc.pt
João Nunes [2017247442] - joaonunes@student.dei.uc.pt
Pedro Carvalho [2017267408] - pccarvalho@student.dei.uc.pt

22 de dezembro de 2021

1 Introdução

No âmbito da disciplina de MEI, para esta terceira meta, foi realizado um estudo que testa as hipóteses apresentadas na meta 2 (com algumas reformulações) com a finalidade de averiguar qual o desempenho dos algoritmos de *backtracking* e quais serão os melhores resultados temporais a escalonar os exames para os alunos de uma universidade.

De modo a aferir o desempenho dos diferentes algoritmos foi, em primeiro lugar, analisadas as diferentes variáveis do sistema e criados ambientes de teste que executava os dois algoritmos para diferentes combinações de variáveis, o qual será explicado mais em detalhe na secção 2 sobre a análise do problema.

Com o estudo e consequente previsão foi feita a formulação das hipóteses e realizado o seu cenário de experimentação. Na secção 3 será apresentado a formulação, o cenário de experimentação e os resultados e discussaõ das diferentes hipóteses realizadas.

Por fim, porcedeu-se à conclusão do trabalho na secção 4.

2 Análise do Problema

De modo a ser possível fazer uma análise mais concisa do trabalho, primeiramente foi identificada uma *research question*, sendo esta a base de toda a exploração dos dados e respetiva análise e discusão dos dados.

Como é que a performance de um algoritmo de escalonamento de exames é afetada pelo número de exames e a probabilidade de dois exames terem pelo menos um aluno em comum?

Com este objetivo em mente, será apresentado de seguida uma síntese dos dois algoritmos de escalonamento de exames utilizados e seguido da identificação das variáveis independentes e dependentes do ambiente.

2.1 Algoritmos de Escalonamento de Exames

Para este trabalho, foram apresentados dois algoritmos de backtracking com uma seed aleatória. Por definição, um algoritmo de backtracking permite construir uma solução recursivamente, removendo os resultados que não vão ao encontro da solução procurada.

O *code1* vai percorrer todos os exames que tem e vai preenchendo o calendário quando encontra um espaço livre para o exame em que se encontra.

O *code2* vai percorrer os espaços no calendário até encontrar um espaço vazio onde pode colocar um exame com o tempo pretendido.

2.2 Variáveis Independentes e Dependentes

Para se proceder à análise exploratória de dados deste trabalho, é importante identificar as variáveis presentes no sistema, assim como verificar se estas são variáveis dependentes ou independentes.

As tabelas seguintes mostram as variáveis do sistema.

Variáveis Independentes	**Descrição**
n	Número de exames
p	Probabilidade de 2 exames terem pelo menos um aluno em comum
s	*Random seed* do sistema
c	Número de *time slots* encontrados pelo algoritmo

Tabela 1: Tabela de variáveis dependentes do sistema

Variáveis Dependetes	**Descrição**
cpu_time_used	Tempo total de execução do algoritmo de escalonamento

Tabela 2: Tabela de variáveis independentes do sistema

Durante os próximos cenários de experimentação iremos observar a dependência do tempo final com o resto das variáveis. Apesar de ser uma variável de saída do sistema, o número de *time slots* encontrados pelos algoritmos não será considerada uma variável dependente (resultado da Análise Exploratória de Dados da meta 1).

3 Teste das Hipóteses

Nesta secção serão apresentadas as hipóteses escolhidas mediante a prévia Análise Exploratória de Dados com a sua respetiva formulação, cenário de experimentação, seguida da apresentação dos resultados e discussão dos mesmos.

3.1 Hipótese 1

3.1.1 Formulação

Para o mesmo ambiente, onde o número de exames e a probabilidade de 2 exames terem pelo menos um aluno em comum é fixo, a performance do algoritmo *code2* é semelhante à do algoritmo *code1*.

3.1.2 Cenário de Experimentação

Para esta hipótese foram criados diferentes cenários de experimentação com base no que foi concluído na meta 1 sobre a análise exploratória de dados. Primeiramente foram escolhidos 2 valores de número de exames e probabilidade que, foram os valores recolhidos da meta 1 onde ainda não havia um aumento significativo do tempo de execução. Os valores escolhidos foram, portanto, 30 exames com uma probabilidade de 0.3 e 40 exames com uma probabilidade de 0.1.

Com cada uma destas combinações foram gerados, através do gerador de Python fornecido, 3 *samples* diferentes idependentes entre si. Com os *samples* gerados, cada um destes foi testado 100 vezes para cada um dos algoritmos de modo a ser registado o tempo de execução de cada teste.

Tanto para a geração em Python como para os 2 códigos C a *random seed* foi sempre diferente.

3.1.3 Resultados e Discussão

Para esta hipótese considera-se um nível de confiança de 95%, ou seja, $\alpha = 0.05$. Com base no cenário de experimentação explicado anteriormente foram obtidos resultados que nos permitem concluir a veracidade da hipótese que foi formulada.

A hipótese nula (H_0) que se pretende testar é: $\mu_{code1} - \mu_{code2} = 0$; sendo a hipótese alternativa (H_1): $\mu_{code1} - \mu_{code2} \neq 0$.

Em primeiro lugar, para cada um dos diferentes samples, foi calculada a diferença entre cada um dos diferentes 100 testes para, de seguida, verificar se a diferença entre os tempos de execução segue uma distribuição normal (através de um QQ-Plot - Figuras 1 e 2).

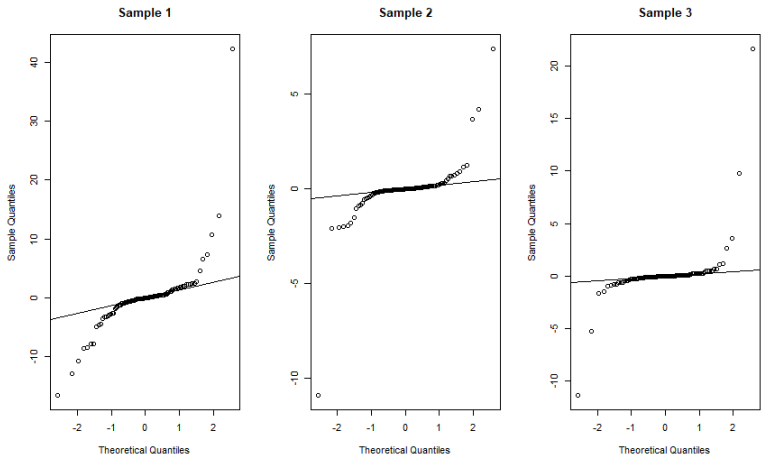

Figura 1: QQ-Plots - 30 Exames e Probabilidade 0.3

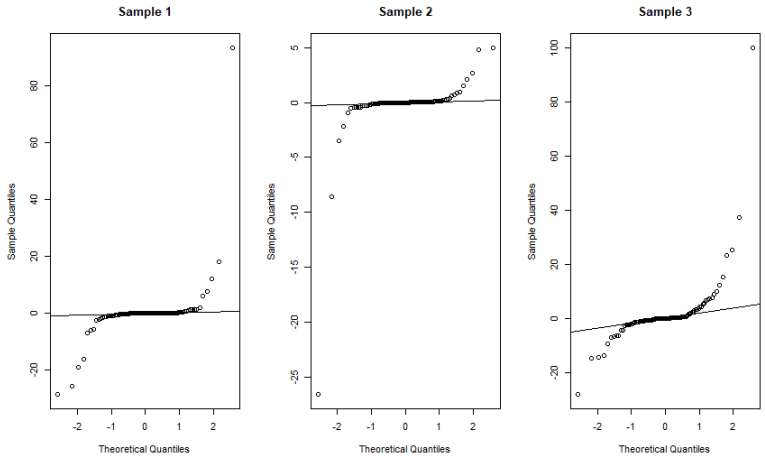

Figura 2: QQ-Plots - 40 Exames e Probabilidade 0.1

Observando os gráficos e através de um teste Shapiro-Wilk dos dados, o p-valor é um valor muito baixo comparado com o α. Assim, os dados não seguem uma distribuição normal logo não será possível fazer um teste paramétrico como *t-test* ou ANOVA e deverá ser utilizado um teste não-paramétrico para a análise da hipótese nula.

Através do teste não-paramétrico Wilcoxon Signed Ranks iremos determinar um valor de W que será comparado com o valor crítico teórico que, através da função *qsignrank* [8] do R para um teste *two-tailed* com um valor de $n = 100$ o valor crítico é de **1956**.

Com o teste Wilcoxon Signed Ranks foi feita a diferença entre os tempos de execução dos 2 códigos e calculado o *rank* de cada uma destas diferenças. Com a ajuda das funções *sign* e *rank* do R foi possível encontrar os valores da soma dos *ranks* positivos e negativos, em que o W será o menor destes dois.

	sample1_30ex	sample2_30ex	sample3_30ex	sample1_40ex	sample2_40ex	sample3_40ex
SumPosRanks	3672	3454	3672	3565	3565	3775
SumNegRanks	1378	1431	1326	1081	820	1225

Tabela 3: Valores das somas dos *ranks* positivos e negativos de cada *sample*

Como se pode observar pela tabela, as somas dos ranks negativos foram aqueles que apresentaram um valor menor, sendo estes declarados como o W. Estes valores de W são agora comparados com o valor crítico de 1956.

Ao comparar-se estes valores de W de cada sample diferente, podemos concluir que o valor crítico é sempre maior que o W. Assim, para um nível de significância de 0.05, H_0 é **rejeitado** e H_1 aceite. Ou seja, a performance dos algoritmos não é necessariamente igual para o mesmo número de exames e probabilidade, apesar de ser não serem muito diferentes como mostra a figura 4.

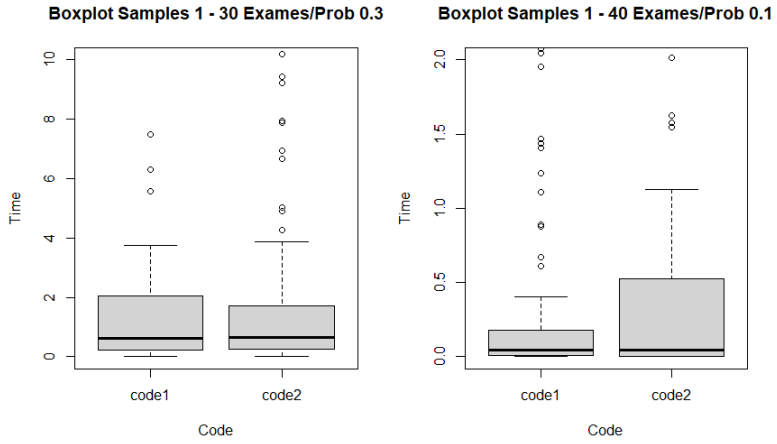

Figura 3: Boxplots de dois dos *samples* testados

Como não há comparação entre diferentes valores de número de exames e probabilidades e apenas são comparados os diferentes algoritmos nesta hipótese, não foi efetuada uma análise *post-hoc*.

3.2 Hipótese 2

3.2.1 Formulação

O tempo máximo de execução é alcançado quando o número de exames é igual, tanto para o *code1* como para o *code2*, independentemente do valor da proba-

bilidade

3.2.2 Cenário de Experimentação

Para esta hipótese inicialmente foram criados cenários de experimentação diferentes com base num número de exames e probabilidade variável. Os valores do número de exames foram 10, 20, 30, 40 e 50, sendo que, pela meta 1, foi concluído que os algoritmos tinham bastante dificuldade em executar testes com um número de exames superior a este. Quanto à probabilidade os valores variavam entre probabilidades baixas, médias e altas, sendo os valores escolhidos 0.2, 0.4, 0.6 e 0.8.

Com cada uma destas combinações de número de exames e probabilidade foram gerados, através do gerador *gen.py* fornecido, um *sample* diferente para cada uma das combinações. Com os *samples* gerados, cada um destes foi testado 20 vezes para cada um dos algoritmos para ser registado se conseguiu terminar em tempo de execução útil ou não.

Para o *gen.py* como para os algoritmos *code1.c* e *code2.c* a *random seed* foi sempre variada para cada um dos diferentes testes.

3.2.3 Resultados e Discussão

Para esta hipótese considera-se um nível de confiança de 95%, ou seja, $\alpha = 0.05$. Com base no cenário de experimentação explicado anteriormente foram obtidos resultados que permitem chegar a uma conclusão de aceitar ou rejeitar a hipótese formulada.

A hipótese nula (H_0) que se pretende testar é: $p_{code1} = p_{code2}$; sendo a hipótese alternativa (H_1): $p_{code1} \neq p_{code2}$.

Para testar esta hipótese fez-se uma proporção das vezes em que o algoritmo terminou em tempo de execução útil, ou seja, menor que 100 segundos que resultou na seguinte tabela.

	code1	*code2*	Total
Executado	236 (232)	228 (232)	464
Não Executado	164 (168)	172 (168)	336
Totals	400	400	800 (Grand Total)

Tabela 4: Valores da contagem dos resultados obtidos nos testes

Depois da recolha dos dados da tabela foi realizado um Teste do Qui-quadrado [9] sobre os resultados.

```
> f <- matrix(c(236,228,164,172),nr=2,byrow=TRUE)
> chisq.test(f)

        Pearson's Chi-squared test with Yates' continuity correction

data:  f
X-squared = 0.25144, df = 1, p-value = 0.6161
```

Figura 4: Procedimento de realização do Teste do Qui-quadrado

Os resultados obtidos permitem concluir que, como o p-valor apresenta maior valor que o nível de significância, então H_0 não é rejeitado.

4 Conclusão

Em conclusão, este trabalho conseguiu dar uma visão mais abrangente sobre métodos estatísticos no que diz respeito à formulação e teste das hipóteses apresentadas. Com esta análise conseguiu-se provar o que já seria esperado da análise exploratória de dados.

No caso da hipótese 1, com o uso de testes não paramétricos, era esperado que a hipótese fosse rejeitada visto que os tempos de execução dos dois algoritmos, apesar de apresentarem alguma semelhança, não são necessariamente iguais.

No caso da hipótese 2, através do uso de proporções e o teste do qui-quadrado, esperava-se que a hipótese nula não fosse rejeitada devido ao facto dos algoritmos não apresentarem grande diferença no que diz respeito ao tempo de execução entre exames e probabilidades, ou seja, na maioria dos casos, ambos os algoritmos não conseguem executar em tempo útil quando apresentam um certo número de exames e probabilidade, por exemplo, quando a probabilidade é mais alta o que, com base nos testes feitos para esta hipótese e sua respetiva análise, provou exatamente o esperado.

Referências

[1] Backtracking Algorithms, https://bit.ly/3mCj6O9

[2] Paquete, L., 2021, 'Slide 7 - Comparing two alternatives', slides 1-41

[3] Paquete, L., 2021, 'Slide 11 - Non-parametric statistical inference', slides 15-26

[4] Checking normality in R, https://bit.ly/3yKAWmz

[5] Understanding Q-Q Plots, https://bit.ly/3snWSTo

[6] Introduction to Statistics for Research: Proportions, chi-squared tests and odds ratios, https://bit.ly/3p9lCN8

[7] ANOVA in R, https://bit.ly/3qgbhhE

[8] SignRank: Distribution of the Wilcoxon Signed Rank Statistic, https://bit.ly/3smAb1X

[9] chisq.test: Pearson's Chi-squared Test for Count Data, https://bit.ly/30PMbh2