Metodologias Experimentais em Informática

Meta 3 - Teste de Hipóteses

Joana Brás - 2021179983
Leandro Pais - 2017251509
Renato Ferreira - 2015228102

Dezembro 2021

1 Introdução

Nesta terceira e última fase do projeto procura-se testar as hipóteses que foram criadas anteriormente. Para contextualizar, importa fazer um apanhado do trabalho realizado até agora.

Inicialmente, numa primeira fase, foi analisado o problema em questão, desenhou-se o método experimental e foram criados os scripts necessários para o efeito. Definiram-se intervalos para as várias variáveis independentes identificadas e partiu-se para a testagem. Com os testes feitos, elaborou-se a análise exploratória de dados com regressão para encontrar o modelo que melhor se adequa aos dados obtidos. Daí concluiu-se que ambos os programas seguem uma complexidade temporal exponencial.

Num segunda fase, com base nos resultados obtidos na primeira meta foram criadas algumas hipóteses para serem testadas numa terceira fase.

2 Método Experimental

A experimentação nesta fase foi em tudo semelhante aquilo que foi feito na fase inicial. Foram utilizados os mesmos scripts, para os mesmos valores de variáveis independentes a correr na mesma máquina.

Assim, geraram-se novos casos de testes todos com seeds diferentes dos casos utilizados anteriormente e procedeu-se à testagem utilizando sempre seeds diferentes para garantir independência entre testes. No entanto, devido ao facto de usarmos os mesmos testes para o código 1 e para o código 2 criou-se emparelhamento dos resultados que foi tido em conta na fase de teste hipóteses.

3 Teste de Hipóteses

Com os resultados obtidos passou-se então à fase de teste das hipóteses estipuladas na meta 2. No entanto, como algumas não se adequavam e seriam complexas de testar com precisão optou-se por testar apenas as seguintes.

3.1 Hipótese 1

Ambos os códigos tem um desempenho semelhante, ou seja, um programa não é mais rápido que o outro.

3.1.1 Formalização da hipótese

$$H_0 : \chi TempoProcessamentoCodigo1 - \chi TempoProcessamentoCodigo2 = 0$$

$$H_1 : \chi TempoProcessamentoCodigo1 - \chi TempoProcessamentoCodigo2 \neq 0$$

3.1.2 Three way anova

Para testar esta hipótese como estavam três variáveis em questão (número de exames, probabilidade de colisão e código 1 ou 2) inicialmente optou-se por utilizar um three way anova implementado da seguinte forma:

```
aov.out = aov(Time~factor(Exams)*factor(Prob)*factor(Program),
data = times_dframe)

summary(aov.out)

plot(aov.out)
```

3.1.2.1 Resultados

Do teste anterior interessa apenas o valor de p-value para o *Time* em função do *Program*. Este valor é $1.3 * e^{-13}$ o que nos leva a **rejeitar a hipótese nula**. Concluindo assim, que existem diferenças significativas entre os programas.

3.1.2.2 Verificação dos pressupostos

Resta agora, fazer a verificação dos pressupostos. Analisando, o *qqplot* abaixo, resultante do anova anterior facilmente se conclui que os dados não seguem uma distribuição normal e como tal, **a conclusão anterior é invalidada**. Isto acontece, pois como foi mencionado anteriormente no design do método experimental foi criado um emparelhamento entre os dados.

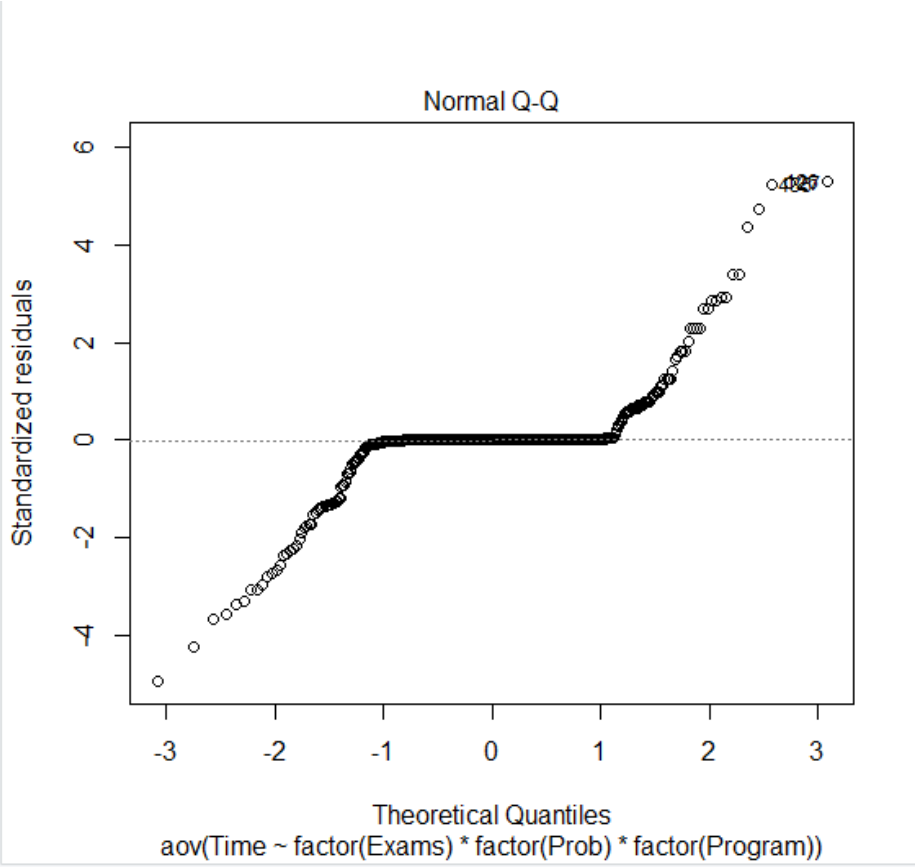

Figure 1: Barplot - Número de Casos Resolvidos: Programa 1 vs Programa 2

3.1.3 Teste de aleatoriedade

Não conseguindo verificar os pressupostos, optou-se então por utilizar testes não paramétricos. Assim a alternativa direta ao three way anova foi o **teste de aleatoriedade**. Este teste não é tão poderoso quanto o anova mas tem a vantagem que não tem grandes pressupostos e que estes estão assegurados pela independência que garantimos através do design experimental. No que toca à implementação, foi a seguinte:

```
#Randomization test
#Get three way anova to adjust p-values
aov.out = aov(Time~factor(Exams)*factor(Prob)*factor(Program),
data = times_dframe)
summary(aov.out)

#F-values
```

```
FExams = summary(aov.out)[[1]]\$F[1]
FProb = summary(aov.out)[[1]]\$F[2]
FProgram = summary(aov.out)[[1]]\$F[3]
FEPP = summary(aov.out)[[1]]\$F[7]

#Initialize p-values
pvalueExams=0
pvalueProb=0
pvalueProgram=0
pvalueFEPP=0

for(i in 1:5000){
  #Randomized sample
  times_dframe\$Time = sample(times_dframe\$Time)

  #Three way anova for randomized sample
  aov.out = aov(Time~factor(Exams)*factor(Prob)*factor(Program),
  data = times_dframe)
  summary(aov.out)

  #Get F-values for aov randomized sample
  pFExams = summary(aov.out)[[1]]\$F[1]
  pFProb = summary(aov.out)[[1]]\$F[2]
  pFProgram = summary(aov.out)[[1]]\$F[3]
  pFEPP = summary(aov.out)[[1]]\$F[7]

  if(pFExams>=FExams){
    pvalueExams=pvalueExams+1
  }

  if(pFProb>=FProb){
    pvalueProb=pvalueProb+1
  }

  if(pFProgram>=FProgram){
    pvalueProgram=pvalueProgram+1
  }

  if(pFEPP>=FEPP){
    pvalueFEPP=pvalueFEPP+1
  }
}

print(pvalueExams/5000)
print(pvalueProb/5000)
print(pvalueProgram/5000)
print(pvalueFEPP/5000)
```

3.1.3.1 Resultados

Do teste anterior, interessa novamente apenas o p-value para a variável *Program* pois é este que está a ser alvo de estudo. Este valor é **0** isto acontece porque os valores são de facto muito diferentes o que nos leva a **rejeitar a hipótese nula**. Concluindo assim, que os códigos produzem, efetivamente, resultados

significativamente diferentes.

3.2 Hipótese 2

Para um número de exames igual a 25, probabilidade de colisão entre 40% e 55% e tempo máximo de execução de 50 segundos, a proporção de instâncias com tempo de execução abaixo de 0.6 segundos é menor ou igual a 50%.

3.2.1 Formalização da hipótese

H_0: Para 25 exames e probabilidade de colisão entre 40% e 55%, a proporção de dados com tempo de execução abaixo de 0.6 segundos é **menor ou igual** a 50%.

H_1: Para 25 exames e probabilidade de colisão entre 40% e 55%, a proporção de dados com tempo de execução abaixo de 0.6 segundos é **superior** a 50%.

3.2.2 Teste de proporções

Para testar esta hipótese, foram geradas 100 instâncias para cada combinação de probabilidades e de número de exames (as probabilidades tomaram os valores de 0.4, 0.45, 0.5, 0.55 e foi fixado o número de exames igual a 25) para o código 1. Esta hipótese foi testada com recurso a um teste de proporções da seguinte forma:

```
#Parametros para alterar:

t_exec=0.6 #tempo de execução até o qual queremos que se encontrem os dados
propor=0.5 #proporção que pretendemos atingir
results_to_use=results_c2 #results_to_use é o código que pretendemos correr
#(codigo 1: results_to_use=results_c1; codigo 2: results_to_use=results_c2)
probs <- c(0.4,0.45,0.5,0.55)
proporcao=c(propor,propor,propor,propor)

for (i in 1:length(probs)) {
  below=0
  for (j in 1:length(results_to_use[,1])) {
    if (results_to_use[j,2]==probs[i] & results_to_use[j,5]<=t_exec) {
      below=below+1
      if (i == 1)
        all_below <- below
    }
  }

  if (i>1)
    all_below <- c(all_below, below)
}

```

```
prop.test(x = all_below,n = c(100,100,100,100),p = proporcao,
alternative = "greater")
```

3.2.2.1 Análise Post-Hoc

Foi feita ainda uma análise post-hoc, que consistiu na verificação da hipótese nula para cada conjunto de número de exames e probabilidades.

```
for (i in 1:length(probs)) {
  below=0
  for (j in 1:length(results_to_use[,1])) {
    if (results_to_use[j,2]==probs[i] & results_to_use[j,5]<=t_exec) {
      below=below+1
    }
  }

  print(prop.test(x = below,n = 100,p = propor, alternative = "greater"))
}
```

3.2.2.2 Resultados

O alvo de estudo neste caso, à semelhança da hipótese anterior, é o valor p-value. Os resultados obtidos a partir da utilização do teste de proporções foram semelhantes para o código 1 e para o código 2. Em ambos os casos se **rejeita a hipótese nula**, ou seja, há mais casos em que o tempo de execução é superior a 0.6 segundos. Como tal, foi feita uma análise post-hoc. Os resultados obtidos de p-value foram os seguintes:

		Probabilidade			
		0.4	0.45	0.5	0.55
Code	Código 1	8.54e-06	2.066e-05	0.9332	1
	Código 2	1.818e-09	0.01786	0.9332	1

Como podemos verificar a partir da tabela, é possível ver que o p-value é inferior ao nível de significância igual a 5% para os casos em que a probabilidade é 0.4 ou 0.45, sendo que se rejeita a hipótese nula para estes casos. No entanto o mesmo não acontece para uma probabilidade igual ou acima de 50%. Nestes dois casos, não se rejeita a hipótese nula.

3.3 Hipótese 3

A terceira e última hipótese consiste em avaliar se o número de casos resolvidos varia significativamente com o aumento do max runtime. Esta hipótese pode ser formalizada da seguinte forma:

3.3.1 Formalização da hipótese

H_0: A proporção de casos resolvidos é a mesma para todos os valores de max runtime.

H_1: As várias proporções são significativamente diferentes.

3.3.2 Teste Qui-Quadrado

Considerando a hipótese em questão optou-se por utilizar o teste do qui-quadrado. Para tal foi construída a seguinte tabela de contingência que inclui o número de casos resolvidos e não resolvidos para os diferentes max runtimes. No total, para cada runtime foram executados 245 testes.

Tabela de Contingência						
	200	400	600	800	1000	**Total**
Solved	165	169	173	174	175	**856**
Unsolved	80	76	72	71	70	**369**
Total	**245**	**245**	**245**	**245**	**245**	**1225**

3.3.2.1 Resultados

Assim, como a tabela anterior é um data frame em R para executar o teste foi apenas chamada a função *chisq.test* com o data frame como input. O que gerou um **p-value** de aproximadamente 0.99 o que para um **nível de significância igual a 5% não nos permite rejeitar a hipótese nula**. Concluindo assim, que as proporções entre os vários max runtimes são, de facto, semelhantes.

4 Conclusão

Foram consideradas muito mais hipóteses do que as enunciadas neste relatório, nomeadamente as presentes no trabalho referente à meta 2. No entanto, por não conhecermos bem a complexidade das mesmas e das respetivas ferramentas de teste uma grande maioria não se adequavam ao contexto do trabalho, forçando-nos a pensar noutras alternativas e hipóteses observáveis a partir dos dados. Um dos exemplos seria a hipótese onde seria fixada a probabilidade de colisão e o número de exames e verificar se a percentagem de casos resolvidos seria superior a 50%. Para estudar esta hipótese, seria aplicado um teste do qui-quadrado, porém, quando foi construída a tabela de contingência, algumas das células tinham uma frequência menor que 5, e nesta situação (*rule of thumb*), não é aconselhado o uso deste teste.

Contudo, as hipóteses apresentadas neste relatório permitiram adquirir mais conhecimentos na área, trabalhar em primeira mão com a linguagem de programação R e perceber como podemos testar certos pressupostos nos dados, assim como que estatística se aplica melhor a cada caso, sendo por isso uma mais-valia para o futuro.

References

[1] *Link para o repositório com scripts, testes e dados de input*
https://github.com/lbpaisDev/MEI_META_3