Entrega 3 - Exploratory Data Analysis and Linear Regression

Metodologias Experimentais em Informática | 2022/2023 | 1.º Semestre

Eva Teixeira	n.º 2019215185	uc2019215185@student.uc.pt	PL1
João Dionísio	n.º 2019217030	uc2019217030@student.uc.pt	PL2
João Oliveira	n.º 2022184283	uc2022184283@student.uc.pt	PL4

1. Introdução

No âmbito da unidade curricular de Metodologias Experimentais em Informática, foi-nos proposto a construção de experiências que avaliem o desempenho de três algoritmos distintos: Dinic, EK (Edmonds–Karp) e MPM (Malhotra, Pramodh-Kumar and Maheshwari). Estes algoritmos visam resolver o problema do Maximum Flow para um determinado grafo.

A experiência pretende avaliar o tempo de execução de cada um dos três algoritmos através da influência das variáveis independentes que caracterizam os grafos: número de vértices (n), probabilidade de arco (p) e capacidade máxima (r).

Para a meta 3 utilizamos técnicas de análise nos dados obtidos anteriormente para obter conclusões sobre o desempenho dos três algoritmos em diferentes circunstâncias.

2. Declaração do problema

Após a análise exploratória dos dados obtidos nos testes iniciais, formulámos três hipóteses:

1. Sendo T_D o tempo de execução do Dinic, T_M o tempo de execução do MPM e T_E o tempo de execução do EK:
   - $H_0 = T_D \geq T_M \geq T_E$
   - $H_1 = T_D < T_M < T_E$

2. Sendo P a probabilidade de arco, $V(T)_{li \leq v \leq ls}$ a variância do tempo de execução para um intervalo$[l_i ; l_s]$ do número de vértices e V o número de vértices, para $P = 50\%$:
   - $H_0 = V(T)_{150 \leq v \leq 350} \leq V(T)_{750 \leq v \leq 950}$
   - $H_1 = V(T)_{150 \leq v \leq 350} > V(T)_{750 \leq v \leq 950}$

3. Sendo $CapMax$, a capacidade máxima e T, o tempo de execução:
   - $H_0 = T \Leftarrow CapMax$ (tempo depende da capacidade máxima)
   - $H_1 = T \;//\; CapMax$ (tempo não depende da capacidade máxima)

3. Análise e discussão

Hipótese I

Hipótese:
Sendo T_D o tempo de execução do Dinic, T_M o tempo de execução do MPM e T_E o tempo de execução do EK:

- $H_0 = T_D \geq T_M \geq T_E$
- $H_1 = T_D < T_M < T_E$

Esta hipótese pretende comparar o desempenho dos três algoritmos. Para melhor aplicar as devidas técnicas de análise, é necessário validar os pressupostos.
Inicialmente, realizou-se o teste de Shapiro-Wilk para validar a distribuição normal dos dados:

```
> shapiro.test(dinic3D$time)

        Shapiro-Wilk normality test

data:  dinic3D$time
W = 0.78017, p-value = 1.314e-15

> shapiro.test(EK3D$time)

        Shapiro-Wilk normality test

data:  EK3D$time
W = 0.5471, p-value < 2.2e-16

> shapiro.test(MPM3D$time)

        Shapiro-Wilk normality test

data:  MPM3D$time
W = 0.76065, p-value = 2.688e-16
```

Figura 1 *Teste de Shapiro para o algoritmo Dinic, EK e MPM*

Os resultados dos testes sugerem que os dados não seguem uma distribuição normal, uma vez que o 'p-value' é muito mais baixo que 0.05. Com isto em consideração, não podemos realizar testes que assumam que os dados estejam normalmente distribuídos, tais como testes T ou Anova. Testes não paramétricos podem ser realizados, visto que este tipo de testes não assumem que os dados seguem uma distribuição particular.

Desta forma, começamos por testar se os tempos de execução do Dinic são inferiores aos do MPM com recurso ao teste de Mann-Whitney:

```
> wilcox.test(dinic3D$time, MPM3D$time, alternative = "less")

        Wilcoxon rank sum test with continuity correction

data:  dinic3D$time and MPM3D$time
W = 10760, p-value = 4.92e-12
alternative hypothesis: true location shift is less than 0
```

Figura 2 *Teste de Mann-Whitney entre os tempos do Dinic e os tempos do MPM*

Sendo o p-value menor que 0.05, concluímos que a hipótese nula pode ser rejeitada e, portanto, o Dinic tem um desempenho superior em relação ao MPM.

De seguida, realizámos um teste de Mann-Whitney outra vez para verificar se os tempos de MPM eram menores que os do EK.

```
> wilcox.test(MPM3D$time, EK3D$time, alternative = "less")

        Wilcoxon rank sum test with continuity correction

data:  MPM3D$time and EK3D$time
W = 8501, p-value < 2.2e-16
alternative hypothesis: true location shift is less than 0
```

Figura 3 *Teste de Mann-Whitney entre os tempos do MPM e os tempos do EK*

Mais uma vez, sendo o p-value menor que o nível de significância, a hipótese nula é rejeitada e consequentemente, concluímos que os tempos do MPM são menores que os do EK.

Em suma, chega-se à conclusão de que o algoritmo Dinic apresenta um melhor desempenho que o MPM, que por sua vez, tem um desempenho superior ao do EK.

Hipótese II

Hipótese:

Sendo P a probabilidade de arco, $V(T)_{li \leq v \leq ls}$ a variância do tempo de execução para um intervalo$[l_i ; l_s]$ do número de vértices e V o número de vértices, para $P = 50\%$:

4. $H_0 = V(T)_{150 \leq v \leq 350} \leq V(T)_{750 \leq v \leq 950}$
5. $H_1 = V(T)_{150 \leq v \leq 350} > V(T)_{750 \leq v \leq 950}$

Para testar esta hipótese usaremos um F-test que compara as variâncias das amostras. Contudo, antes de usar o F-test temos de garantir que as amostras seguem uma distribuição normal. Para isto usamos o teste de Shapiro, a partir do qual dá para ver que, sendo o p-value superior a 0.05, os dados são de facto normalmente distribuídos.

```
> EKtempoP50= EK3D$time[seq(5, 190, by=10)]
> EKverticesP50= EK3D$vertices[seq(5, 190, by=10)]
>
> #shapiro test shows that the data is likely to be normally distributed
> shapiro.test(EKtempoP50[3:7])

        Shapiro-Wilk normality test

data:  EKtempoP50[3:7]
W = 0.89184, p-value = 0.3664

> shapiro.test(EKtempoP50[15:19])

        Shapiro-Wilk normality test

data:  EKtempoP50[15:19]
W = 0.9026, p-value = 0.4244
```

Figura 4 *Testes de Shapiro dos dados do EK para número de vértices entre 150 e 350 e número de vértices entre 750 e 950 e probabilidade de arco de 50%*

De seguida, já podemos realizar o F-test entre estas duas amostras. Realizando o F-test obtivemos um p-value muito menor que o nível de significância (0.05), o que significa que as variâncias são significativamente diferentes. Sendo o F-value positivo concluímos que a variância do primeiro grupo é maior que a do segundo.

```
> result <- var.test(EKtempoP50[3:7] , EKtempoP50[15:19])
> f_value <- result$statistic
> df <- result$parameter
> p_value <- result$p.value
> print(paste("F_value: " ,  f_value))
[1] "F_value:  0.00333282067780266"
> print(paste("DF: " ,  df))
[1] "DF:  4" "DF:  4"
> print(paste("P_value: " ,  p_value))
[1] "P_value:  6.60575245873315e-05"
```

Figura 5 *Resultado do p-value do f-test ao MPM*

Concluindo, podemos confirmar que a hipótese nula deve ser rejeitada, aceitando a hipótese alternativa.

Hipótese III

Hipótese:

Sendo $CapMax$, a capacidade máxima e T, o tempo de execução:

6. $H_0 = T \Leftarrow CapMax$ (tempo depende da capacidade máxima)
7. $H_1 = T \,//\, CapMax$ (tempo não depende da capacidade máxima)

Inicialmente, começamos por verificar a normalidade dos dados usados através do teste Shapiro:

```
> shapiro.test(dinicmaxcap$time)

        Shapiro-Wilk normality test

data:  dinicmaxcap$time
W = 0.7863, p-value = 0.01423

> shapiro.test(EKmaxcap$time)

        Shapiro-Wilk normality test

data:  EKmaxcap$time
W = 0.92592, p-value = 0.4435

> shapiro.test(MPMmaxcap$time)

        Shapiro-Wilk normality test

data:  MPMmaxcap$time
W = 0.888, p-value = 0.1903
```

Figura 6 *Teste Shapiro para os tempos de execução do Dinic, EK, e MPM em função da capacidade máxima*

Como o p-valor do algoritmo Dinic é o único menor que 0.05 (nível de significância), concluímos que os valores do tempo em função da capacidade máxima só não são normalmente distribuídos para o algoritmo de Dinic.

```
> test_results <- kruskal.test(dinicmaxcap$maxcapacity ~ dinicmaxcap$time)
> print(test_results$p.value)
[1] 0.4334701
```

Figura 7 *Teste Kruskal entre a capacidade máxima e o tempo do Dinic*

Assim sendo, para testar se existe correlação entre o tempo e a capacidade máxima no caso do algoritmo Dinic, usámos o teste Kruskal, a partir do qual extraímos um p-valor de 0.4334701, que é maior que 0.05, provando que não existe diferença significativa nos valores do tempo, alterando a capacidade máxima.

```
> model <- aov(EKmaxcap$maxcapacity ~ EKmaxcap$time)
> summary(model)
               Df Sum Sq Mean Sq F value Pr(>F)
EKmaxcap$time   1   9096    9096   0.108  0.752
Residuals      7 590904   84415
```

Figura 8 *Teste ANOVA entre a capacidade máxima e tempo do EK*

```
> model <- aov(MPMmaxcap$maxcapacity ~ MPMmaxcap$time)
> summary(model)
               Df Sum Sq Mean Sq F value Pr(>F)
MPMmaxcap$time  1  92480   92480   1.276  0.296
Residuals       7 507520   72503
```

Figura 9 *Teste ANOVA entre a capacidade máxima e tempo do MPM*

Para o EK e para o MPM, sendo os dados normalmente distribuídos, usamos um teste ANOVA, a partir do qual, da mesma maneira, concluímos que o tempo também não depende da capacidade máxima, nestes algoritmos, sendo o p-value maior que 0.05.

Após a realização dos testes e respetiva observação, podemos afirmar que não existe nenhuma correlação entre a capacidade máxima e os tempos obtidos, pelo que se confirma a nossa hipótese alternativa, de que a capacidade máxima não tem nenhum impacto nos tempos obtidos.

4. Conclusão

Após uma exploração das hipóteses, foram realizados os testes que achamos mais adequados a cada uma das situações, tendo em conta as variáveis envolvidas e o tipo de comparação e relação que existia entre elas.
Os resultados que obtivemos foram positivos e, tendo em conta os testes escolhidos, os valores ajustaram-se aquilo que era pretendido.
De modo geral, todos os nossos resultados foram de encontro às expectativas, confirmando as hipóteses que tínhamos anteriormente formulado. Desse modo, podemos concluir que, as hipóteses que, à primeira vista nos pareciam corretas, são agora validadas, com recurso a testes estatísticos.