Uma imagem com texto

Descrição gerada automaticamente

**Relatório Técnico**

ANADI

2021/2022

**1190827- Luís Manuel Gonçalves Araújo**



# **Resumo**

Este relatório foi elaborado no âmbito da cadeira de Análise de Dados em Informática (ANADI) do segundo semestre do terceiro ano da Licenciatura em Engenharia Informática no Instituto Superior de Engenharia do Porto (ISEP).

Neste relatório é feita uma análise de fiabilidade de um conjunto de dados de acesso à VPN dos servidores do departamento de engenharia informática (DEI) bem como a análise de desempenho de métodos heurísticos na resolução de um problema de escalonamento. Todos os dados apresentados são calculados utilizando a linguagem R, assim como todos os gráficos são gerados usando R e o programa RStudio.

Conteúdo

[**Resumo** 2](#_Toc101652270)

[**1.** **Análise de Fiabilidade** 5](#_Toc101652271)

[**1.1** **Contextualização** 5](#_Toc101652272)

[**1.1.1** **Tipos de testes estatísticos** 5](#_Toc101652273)

[**1.2** **Descrição do Problema** 5](#_Toc101652274)

[**1.3** **Estrutura de Análise** 6](#_Toc101652275)

[**2.** **Análise dos dados da VPN** 6](#_Toc101652276)

[**2.1** **Exercício 1** 6](#_Toc101652277)

[**2.1.1.1** **Número de acessos de cada servidor** 6](#_Toc101652278)

[**2.1.1.2** **Número de falhas de cada servidor** 7](#_Toc101652279)

[**2.1.1.2.1** **Número de vezes que um servidor “x” utiliza um protocolo “y”** 8](#_Toc101652280)

[**2.1.1.3** **Médias, medianas e desvios padrão mensais de acessos de cada servidor (apenas meses completos).** 9](#_Toc101652281)

[**2.2** **Exercício 2** 10](#_Toc101652282)

[**2.2.1 Efetuar um gráfico que nas abcissas representa o tempo, e nas ordenadas o número de falhas simultâneas e o número de acessos simultâneos** 11](#_Toc101652283)

[**2.2.2 Efetuar um diagrama de caixa de bigodes do número diário de falhas simultâneas, para cada servidor (o número diário de falhas simultâneas é o número total de falhas simultâneas que ocorreram nesse dia). Indicar o número de outliers que existem por servidor** 12](#_Toc101652284)

[**3.** **Análise de Desempenho de Algoritmos** 13](#_Toc101652285)

[**3.1** **Contextualização** 13](#_Toc101652286)

[**3.2 Descrição do Problema** 13](#_Toc101652287)

[**3.3** **Estrutura de Análise** 14](#_Toc101652288)

[**4.** **Análise de Desempenho de Métodos Heurísticos na resolução do problema de Escalonamento** 14](#_Toc101652289)

[**4.1 Exercício 1** 14](#_Toc101652290)

[**4.1.1** **Efetuar um gráfico tal que o eixo das abcissas contém os problemas e o eixo das ordenadas contém o valor do makespan das três MH’s para cada instância** 14](#_Toc101652291)

[**4.1.2** **Considerar apenas os dados relativos às 10 instâncias com menor tamanho e verificar se há diferenças significativas entre desempenho das três técnicas** 15](#_Toc101652292)

[**4.1.3** **Identificar a técnica mais eficaz se a resposta à alínea anterior for positiva** 16](#_Toc101652293)

[**4.1.4** **Responder às duas perguntas anteriores usando as 20 instâncias de menor tamanho** 17](#_Toc101652294)

[**4.1.5** **Responder às perguntas anteriores utilizando o makespan normalizado** 19](#_Toc101652295)

[**4.1.6** **Comentar os resultados obtidos usando as duas medidas de desempenho usadas** 22](#_Toc101652296)

[**4.1.7** **Usar os dados relativos às 70 instâncias de menor dimensão e determinar, para cada MH, a reta de regressão linear** 22](#_Toc101652297)

[**4.1.8** **Verificar se os pressupostos sobre os resíduos são verificados (normalidade, homocedasticidade e independência)** 27](#_Toc101652298)

[**4.1.9** **Comentar os resultados obtidos nas duas alíneas anteriores** 32](#_Toc101652299)

[**4.2 Exercício 2** 32](#_Toc101652300)

[**4.2.1 construir um boxplot que contenha os tempos de processamento de cada MH na resolução de cada instância** 33](#_Toc101652301)

[**4.2.2 verificar se existem diferenças significativas nos tempos médios de processamento entre as três técnicas** 34](#_Toc101652302)

[**4.2.3 no caso da resposta da alínea anterior ser positiva, identificar qual a MH mais eficiente** 34](#_Toc101652303)

[**4.2.4 determinar a matriz de correlação entre os tempos de processamento de cada MH e interpretar os resultados** 35](#_Toc101652304)

[**5.** **Referências** 36](#_Toc101652305)

# **Análise de Fiabilidade**

A primeira parte deste trabalho é a análise de fiabilidade do serviço de VPN do Departamento de Engenharia Informática (DEI), através dos dados de início, fim, protocolo, servidor e duração das sessões de 2016 a 2018.

# **Contextualização**

A fiabilidade é a probabilidade de um determinado software operar corretamente, isto é, sem falhas durante um determinado período. É uma forma de avaliar a credibilidade, a qualidade, a segurança e a tolerância a falhas de um sistema ou software específico.

Esta mesma fiabilidade pode ser garantida por testes ao longo de todo o processo de produção, sendo que só assim se podem descobrir erros de funcionamento e falhas para serem corrigidas sem nunca chegarem ao produto final.

# **Tipos de testes estatísticos**

Como dito anteriormente, os testes estatísticos servem para se descobrirem erros de funcionamento de forma a se evitar ter um produto final com falhas.

Apesar de não se terem realizado testes estatísticos nesta parte, é importante mencionar alguns como: teste de hipóteses (permite determinar entre duas hipóteses- hipótese nula (H0) e a hipótese alternativa (H1)- qual a correta podendo ser bilaterais ou unilaterais; t-test (permite aceitar ou não a hipótese nula); teste de Levene (permite comparar médias provenientes de dois conjuntos de dados para determinar se os conjuntos são ou não iguais); etc.

# **Descrição do Problema**

Como dito anteriormente, esta análise de fiabilidade diz respeito ao serviço de VPN dos servidores do DEI, serviço este crucial para os alunos deste mesmo departamento. Estes serviços permitem aos utilizadores localizados em redes exteriores ao DEI criar uma ligação de rede virtual no seu posto de trabalho equivalente a uma ligação física às redes internas do DEI. O seu funcionamento tem de ser consistente e o número de falhas deve ser o menos frequente possível.

De forma a se analisar o funcionamento do serviço é necessário observar os dados fornecidos e retirar as conclusões.

# **Estrutura de Análise**

Esta análise será estruturada de acordo com as alíneas presentes no enunciado. Serão mostrados os resultados de cada alínea bem como um comentário a cada um. A filtragem dos resultados de acordo com os requisitos de cada alínea foi feita com recurso em código R e RStudio.

# **Análise dos dados da VPN**

# **Exercício 1**

Este exercício divide-se em 4 alíneas que pedem vários tipos de filtragem de dados: **2.1.1** número de acessos de cada servidor; **2.1.2** número de falhas de cada servidor; **2.1.3** número de vezes que um servidor “x” utiliza um protocolo “y”; **2.1.4** médias, medianas e desvios padrão mensais de acessos de cada servidor (apenas meses completos).

# **Número de acessos de cada servidor**

Após a importação dos dados fornecidos passou-se à realização dos exercícios começando com o número de acessos de cada servidor. De facto, um acesso é uma sessão com duração superior a um minuto. Assim, após colocar os dados num dataframe, filtraram-se os dados cujo tempo de utilização da VPN em minutos era superior a 1 e agruparam-se (função “group\_by”) por servidor.

Uma imagem com mesa

Descrição gerada automaticamente

Como se verifica, houve um total de 54733 acessos de 66383 valores observados. Ou seja, a probabilidade de ocorrer um acesso foi superior a 82%. Relativamente ao número de acessos:

* Probabilidade de haver um acesso no vsrv10: 3.5%;
* Probabilidade de haver um acesso no vsrv11: 11%;
* Probabilidade de haver um acesso no vsrv16: 29.8%;
* Probabilidade de haver um acesso no vsrv17: 27.5%;
* Probabilidade de haver um acesso no vsrv18: 27.9%.

Verifica-se que o servidor com mais acessos foi o vsrv16 e com menos acessos foi o vsrv10.

# **Número de falhas de cada servidor**

De facto, uma falha é uma sessão com duração igual ou inferior a um minuto. Depois da importação dos dados, colocaram-se estes mesmos num dataframe, filtraram-se os dados cujo tempo de utilização da VPN em minutos era igual ou inferior a 1 minuto e agruparam-se (função “group\_by”) por servidor.

Uma imagem com texto, quadro de resultados, captura de ecrã

Descrição gerada automaticamente

Como se verifica, houve um total de 11650 falhas de 66383 valores observados. Ou seja, a probabilidade de ocorrer uma falha foi superior a 17.5%. Relativamente ao número de falhas:

* Probabilidade de haver um acesso no vsrv10: 2.4%;
* Probabilidade de haver um acesso no vsrv11: 12.8%;
* Probabilidade de haver um acesso no vsrv16: 28.6%;
* Probabilidade de haver um acesso no vsrv17: 27.1%;
* Probabilidade de haver um acesso no vsrv18: 29.2%.

Verifica-se que o servidor com mais falhas foi o vsrv8 e com menos falhas foi o vsrv10.

# **Número de vezes que um servidor “x” utiliza um protocolo “y”**

Neste exercício é pedido que se mostre o número de vezes que um determinado servidor (vsrv8; vsrv10; vsrv11; vsrv16; vsrv17) utiliza um determinado protocolo (PPTP; SSTP; SOFTETHER; OPENVPN L2; OPENVPN L3). Após importação dos dados, criou-se um dataframe só com o server e o protocolo utilizado por ele e contou-se (função “count”) por servidor o número de vezes que utilizou o tal protocolo. Renomeou-se a coluna da frequência utilizando-se a função “rename”.

Uma imagem com texto, armário, mobília, calculadora

Descrição gerada automaticamente

Como se observa, a menor frequência surgiu com a utilização do server vsrv10 e com o protocolo SOFTETHER. Por outro lado, a maior frequência surgiu com a utilização do server vsrv16 e com o protocolo SSTP.

# **Médias, medianas e desvios padrão mensais de acessos de cada servidor (apenas meses completos).**

Neste exercício é pedido que se calcule as médias, medianas e desvios padrões mensais de acessos para cada servidor. Também é necessário ter em conta que os meses têm de ser completos. Para isso agruparam-se inicialmente os servers num array e criou-se uma matriz com o número correspondente ao mês e o seu número de dias para se mais tarde confirmar se um mês está completo (ciclo for que, enquanto que o ano nem o mês avançaram, confirma se o número de dias é igual ao número total de dias do mês em concreto). Após isso colocam-se o número de acessos que houveram em cada mês e utilizam-se as funções “mean”, “median” e “sd” para se calcular a média, mediana e desvio-padrão respetivamente.

Uma imagem com texto, quadro de resultados, armário, mobília

Descrição gerada automaticamente

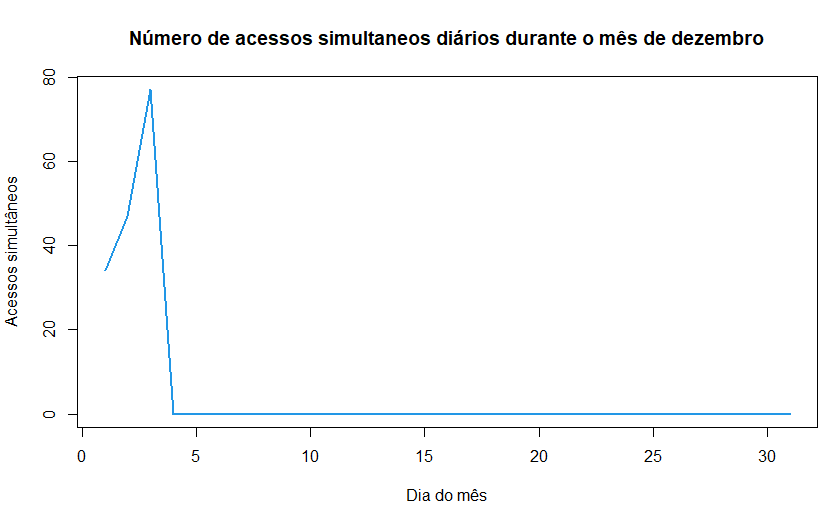
Como se verifica, o servidor vsev16 tem a maior média de acessos, a maior mediana e o maior desvio-padrão. Com um desvio-padrão tão elevado, os valores de acessos deste último servidor encontram-se espalhados por uma alta gama de valores. Por outro lado, é o servidor vsrv10 que tem a menor média e mediana. Quanto ao menor desvio-padrão, este corresponde ao server vsrv11 sendo que os valores de acesso deste servidor estão mais próximos do valor da média.

# **Exercício 2**

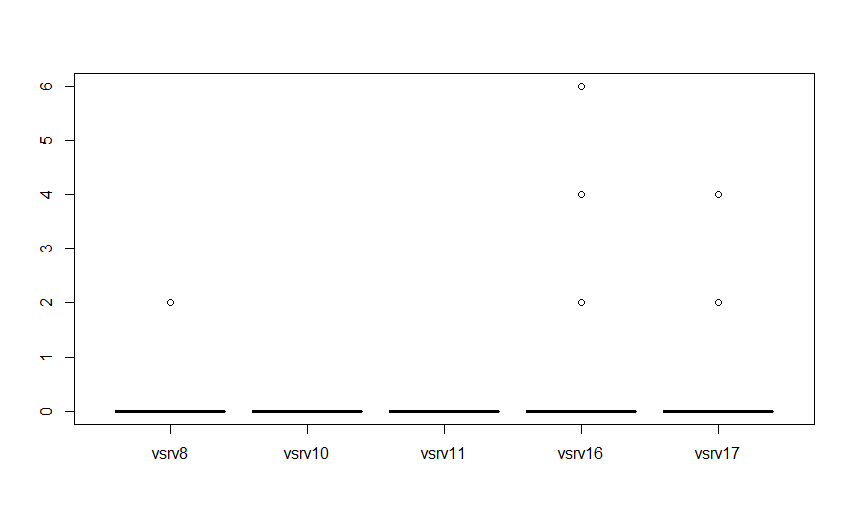
Este exercício divide-se em 3 alíneas que pedem várias filtragens de dados apenas relativamente ao mês de dezembro de 2017 para todos os servidores e protocolos. Estas alíneas são: **2.2.1** efetuar um gráfico que nas abcissas representa o tempo, e nas ordenadas o número de falhas simultâneas e o número de acessos simultâneos; **2.2.2** efetuar um diagrama de caixa de bigodes do número diário de falhas simultâneas, para cada servidor (o número diário de falhas simultâneas é o número total de falhas simultâneas que ocorreram nesse dia). Indicar o número de outliers que existem por servidor; **2.2.3** verificar se há correlação entre o número de falhas simultâneas e o número de acessos simultâneos, no dia 11 de dezembro de 2017 das 12:00 às 14:00.

# **2.2.1 Efetuar um gráfico que nas abcissas representa o tempo, e nas ordenadas o número de falhas simultâneas e o número de acessos simultâneos**

Neste exercício é pedido que se elabore um gráfico em que no eixo das ordenadas se represente o número de falhas e acessos simultâneos e no eixo das abcissas o tempo destes mesmos. Importante referir que o número de falhas simultâneas corresponde ao número de falhas que se inicializaram ou finalizaram num mesmo minuto. Relativamente ao número de acessos simultâneos, este corresponde ao número de acessos que inicializaram antes de um certo minuto e terminaram depois desse minuto.



# **2.2.2 Efetuar um diagrama de caixa de bigodes do número diário de falhas simultâneas, para cada servidor (o número diário de falhas simultâneas é o número total de falhas simultâneas que ocorreram nesse dia). Indicar o número de outliers que existem por servidor**

 Neste exercício é pedido que se elabore um diagrama de caixa de bigodes do número diário de falhas simultâneas para cada servidor. Ainda assim é pedido que se indique o número de outliers.

2.2.3

2.3 Análise de resultados

# **Análise de Desempenho de Algoritmos**

A segunda parte deste trabalho é a análise de desempenho de algoritmos de um problema de escalonamento no qual temos os valores de makespan de 3 métodos heurísticos para os mesmos problemas

# **Contextualização**

De facto, a análise de algoritmos estuda a correção e desempenho de algoritmos. Esta mesma análise de algoritmos procura respostas para perguntas como: “Quanto tempo o algoritmo consome para processar uma entrada de tamanho n?”.

Analisar um algoritmo significa prever a quantidade de recursos que tal algoritmo consome ao ser executado. A análise pode apontar diversos candidatos e exclui diversas soluções não eficientes. Diversas variáveis podem ser objetos de estudo da análise de um algoritmo, contudo, é com maior frequência que se mede o tempo execução.

# **3.2 Descrição do Problema**

Este exercício relaciona-se com um problema de escalonamento sendo este definido como um conjunto de tarefas a executar num conjunto de máquinas. Nesta classe de problemas pretende-se encontrar uma solução que corresponda à melhor sequência de processamento das tarefas do conjunto T, numa ou mais máquinas do conjunto M, de modo que todas as tarefas sejam executadas, respeitando as limitações das máquinas e eventuais restrições adicionais impostas ao problema.

Contudo, a partir de uma certa dimensão torna-se impraticável encontrar a solução ótima de forma eficiente. Uma das métricas de desempenho (funções objetivo) mais usada para se calcular o valor da solução é o makespan.

Este exercício contém um ficheiro com resultados relativos a 80 instâncias do problema de escalonamento. Para cada problema conhece-se o valor do makespan ótimo identificado por OPT e o makespan obtido por cada MH identificadas por MH1, MH2 e MH3.

# **Estrutura de Análise**

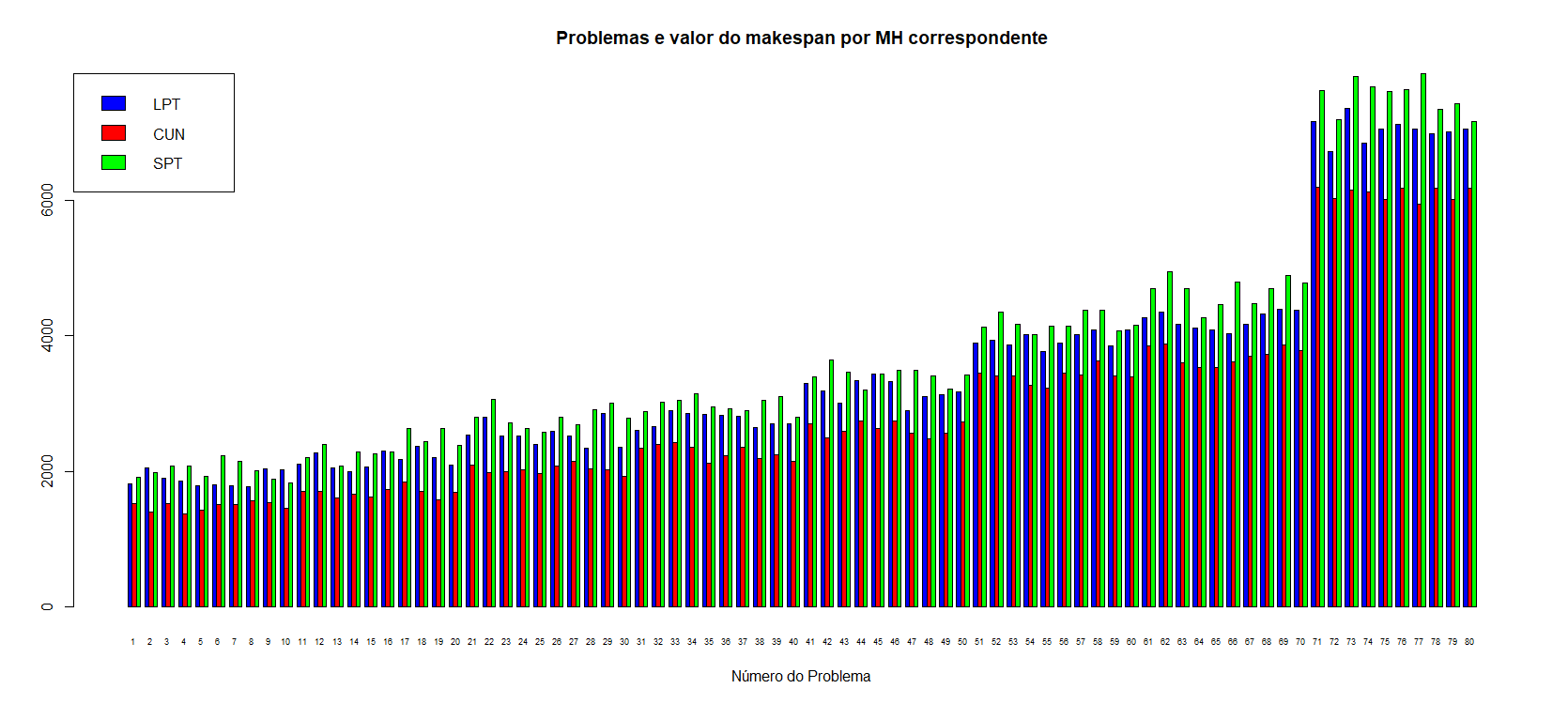
Esta análise será estruturada de acordo com as alíneas presentes no enunciado. Serão mostrados os resultados de cada alínea bem como um comentário a cada um. A filtragem dos resultados de acordo com os requisitos de cada alínea foi feita com recurso em código R e RStudio.

# **Análise de Desempenho de Métodos Heurísticos na resolução do problema de Escalonamento**

# **4.1 Exercício 1**

Este exercício divide-se em 9 alíneas que pedem vários tipos de filtragem de dados: **4.1.1** efetuar um gráfico tal que o eixo das abcissas contém os problemas e o eixo das ordenadas contém o valor do makespan das três MH’s para cada instância; **4.1.2** considerar apenas os dados relativos às 10 instâncias com menor tamanho e verificar se há diferenças significativas entre desempenho das três técnicas; **4.1.3** identificar a técnica mais eficaz se a resposta à alínea anterior for positiva; **4.1.4** responder às duas perguntas anteriores usando as 20 instâncias de menor tamanho; **4.1.5** responder às perguntas anteriores utilizando o makespan normalizado; **4.1.6** comentar os resultados obtidos usando as duas medidas de desempenho usadas; **4.1.7** usar os dados relativos às 70 instâncias de menor dimensão e determinar, para cada MH, a reta de regressão linear; **4.1.8** verificar se os pressupostos sobre os resíduos são verificados (normalidade, homocedasticidade e independência); **4.1.9** comentar os resultados obtidos nas duas alíneas anteriores.

# **Efetuar um gráfico tal que o eixo das abcissas contém os problemas e o eixo das ordenadas contém o valor do makespan das três MH’s para cada instância**

Após a importação dos dados fornecidos passou-se à realização dos exercícios começando com a separação dos dados relativos a cada MH. Colocaram-se os dados numa matriz, renomearam-se as linhas e colunas e realizou-se o gráfico.

Como se observa, estão presentes no gráfico os 80 problemas relacionados com o valor do makespan de cada uma das técnicas. Em cada índice do problema estão contidas as 3 técnicas. Nota-se um aumento gradual do valor do makespan à medida que o problema é maior. No final há um aumento mais abrupto.

# **Considerar apenas os dados relativos às 10 instâncias com menor tamanho e verificar se há diferenças significativas entre desempenho das três técnicas**

Após a importação dos dados, criou-se um dataframe sem a técnica otimizada. Após isso, e através do mesmo dataframe, retiraram-se apenas as 10 menores instâncias. Após isso, realizou-se um teste de Friedman pois este utiliza-se quando há mais de duas amostras emparelhadas. Utilizaram-se também as funções “attach” e “detach” para se aceder às variáveis do dataframe sem ele ser chamado. Assim, temos:

* H0: as três técnicas são iguais VS H1: há pelo menos uma técnica que é diferente das restantes.

Uma imagem com texto

Descrição gerada automaticamente

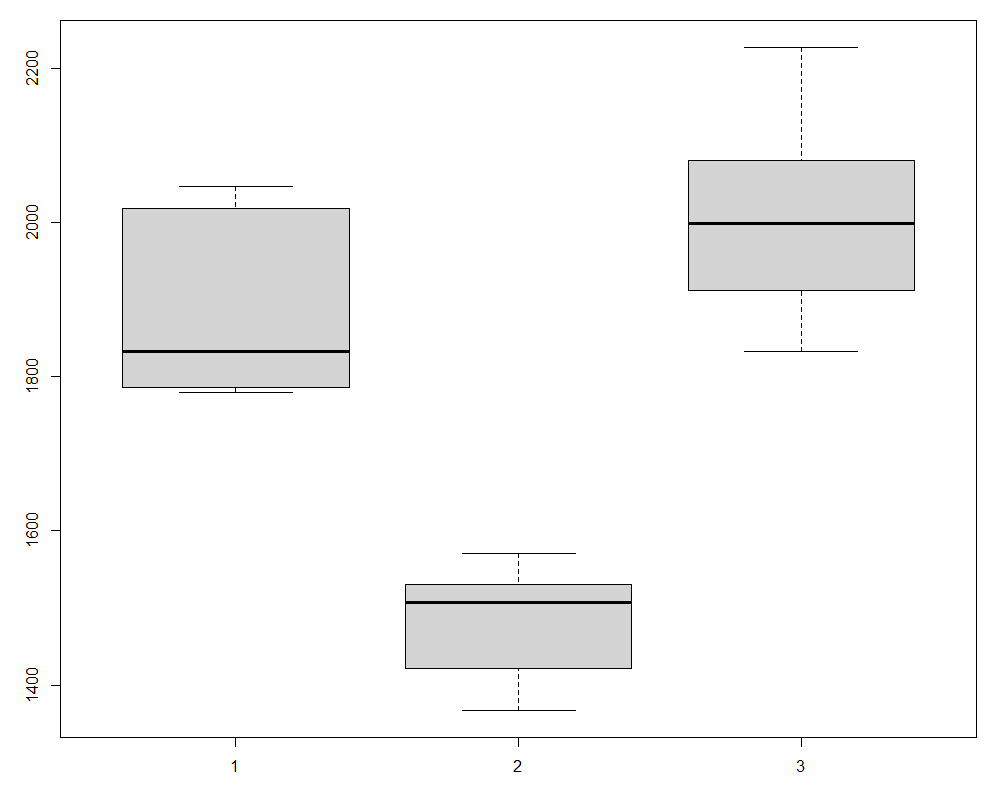
Considerando o alfa=5%, e observando o valor do p-value, podemos rejeitar a hipótese H0. Há assim pelo menos uma técnica que é diferente das restantes.

# **Identificar a técnica mais eficaz se a resposta à alínea anterior for positiva**

Após a importação dos dados passou-se à realização do exercício. Foi utilizado o teste “pairwise.wilcox.test”-post hoc- com a técnica de ajuste “bonferroni” para se verificar as diferenças entre as técnicas.

Uma imagem com texto

Descrição gerada automaticamente

Como se observa no resultado do post-hoc, as técnicas SPT e LPT são mais semelhantes do que a técnica CUN com qualquer uma das restantes. Contudo, só este teste post-hoc não nos mostra qual das técnicas é a mais eficiente. Para isso realizou-se um boxplot com as técnicas LPT, CUN e SPT respetivamente:

Assim conseguimos afirmar que a técnica CUN é a mais eficiente tendo os valores mais baixos de makespan. Por outro lado, a técnica SPT é a menos eficiente. Também se comprova pelo post-hoc que as técnicas LPT e SPT são mais semelhantes que a técnica CUN com as restantes.

# **Responder às duas perguntas anteriores usando as 20 instâncias de menor tamanho**

Após a importação dos dados, criou-se um dataframe sem a técnica otimizada. Após isso, e através do mesmo dataframe, retiraram-se apenas as 20 menores instâncias. Após isso, realizou-se um teste de Friedman pois este utiliza-se quando há mais de duas amostras emparelhadas. Utilizaram-se também as funções “attach” e “detach” para se aceder às variáveis do dataframe sem ele ser chamado. Assim, temos:

* H0: as três técnicas são iguais VS H1: há pelo menos uma técnica que é diferente das restantes

Uma imagem com texto

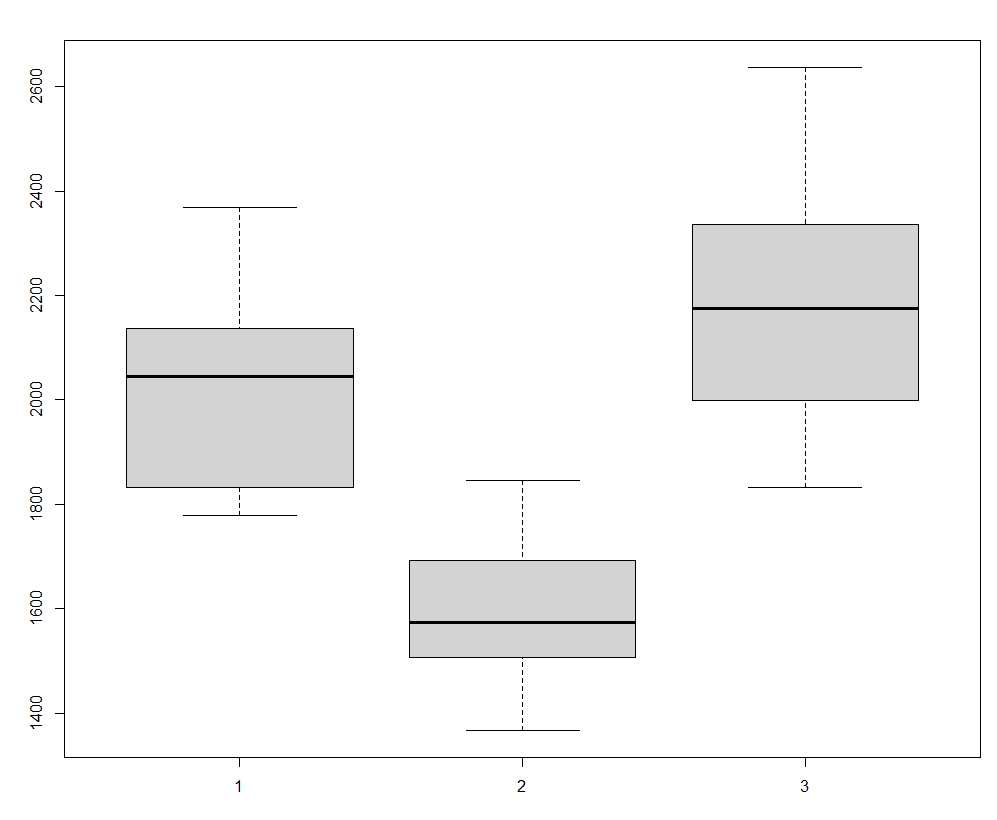
Descrição gerada automaticamente

Considerando o alfa=5%, e observando o valor do p-value, podemos rejeitar a hipótese H0. Há assim pelo menos uma técnica que é diferente das restantes.

Após a importação dos dados passou-se à realização do exercício. Foi utilizado o teste “pairwise.wilcox.test”-post hoc- com a técnica de ajuste “bonferroni” para se verificar as diferenças entre as técnicas.

Uma imagem com texto

Descrição gerada automaticamente

 Como se observa no resultado do post-hoc, as técnicas SPT e LPT são mais semelhantes do que a técnica CUN com qualquer uma das restantes. Contudo, só este teste post-hoc não nos mostra qual das técnicas é a mais eficiente. Para isso realizou-se um boxplot com as técnicas LPT, CUN e SPT respetivamente:

Assim conseguimos afirmar que a técnica CUN é a mais eficiente tendo os valores mais baixos de makespan. Por outro lado, a técnica SPT é a menos eficiente. Também se comprova pelo post-hoc que as técnicas LPT e SPT são mais semelhantes que a técnica CUN com as restantes.

# **Responder às perguntas anteriores utilizando o makespan normalizado**

Após a importação dos dados, criou-se um dataframe só com a técnica otimizada. Além disso, utilizaram-se os dados anteriormente criados e retiraram-se os valores para as 10 menores instâncias e para as 20 menores instâncias. Tanto para calcular o makespan normalizado para as 10 e 20 instâncias criaram-se dataframes para cada técnica e colocaram-se lá o resultado do valor normalizado. Após isso, juntaram-se essas 3 técnicas, tanto para as 10 instâncias como para as 20, num novo dataframe que irá servir para fazer o teste de Friedman. Utilizaram-se também as funções “attach” e “detach” para se aceder às variáveis do dataframe sem ele ser chamado. Assim, para as 10 e 20 instâncias temos:

* H0: as três técnicas são iguais VS H1: há pelo menos uma técnica que é diferente das restantes

Para 10 instâncias:

Uma imagem com texto

Descrição gerada automaticamente

Considerando o alfa=5%, e observando o valor do p-value, podemos rejeitar a hipótese H0. Há assim pelo menos uma técnica que é diferente das restantes.

Uma imagem com texto

Descrição gerada automaticamentePara 20 instâncias:

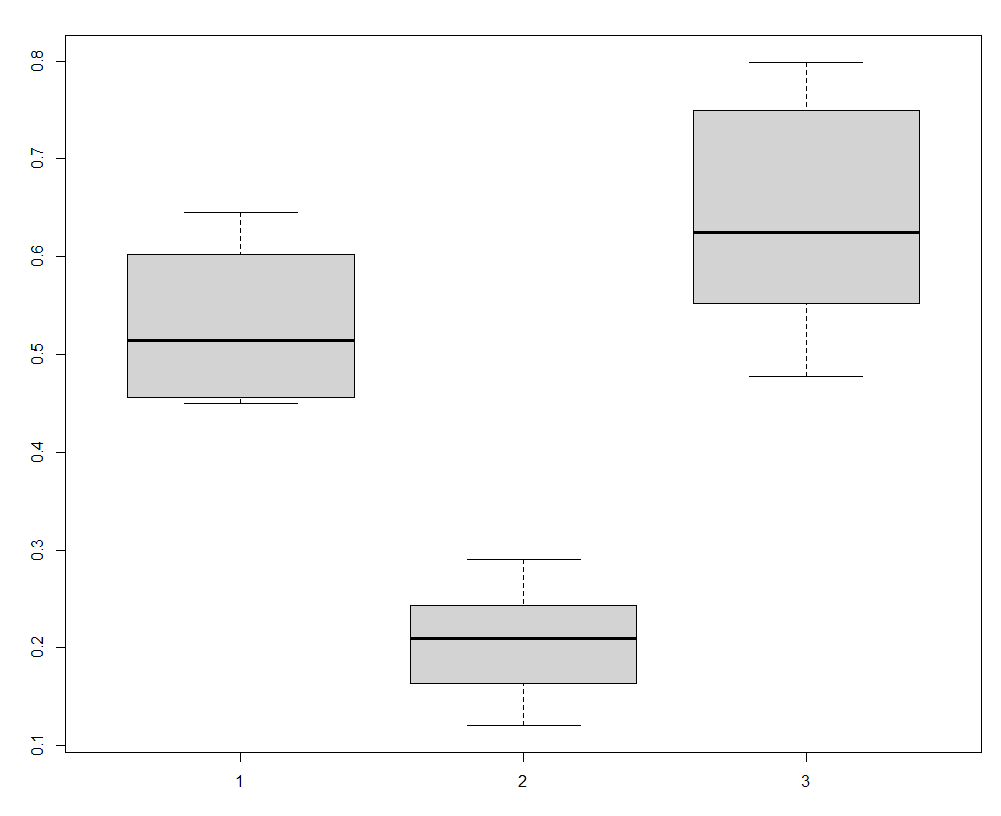
Considerando o alfa=5%, e observando o valor do p-value, podemos rejeitar a hipótese H0. Há assim pelo menos uma técnica que é diferente das restantes.

Após a importação dos dados passou-se à realização do teste “pairwise.wilcox.test”-post hoc- com a técnica de ajuste “bonferroni” para se verificar as diferenças entre as técnicas.

Para 10 instâncias:

Uma imagem com texto

Descrição gerada automaticamente

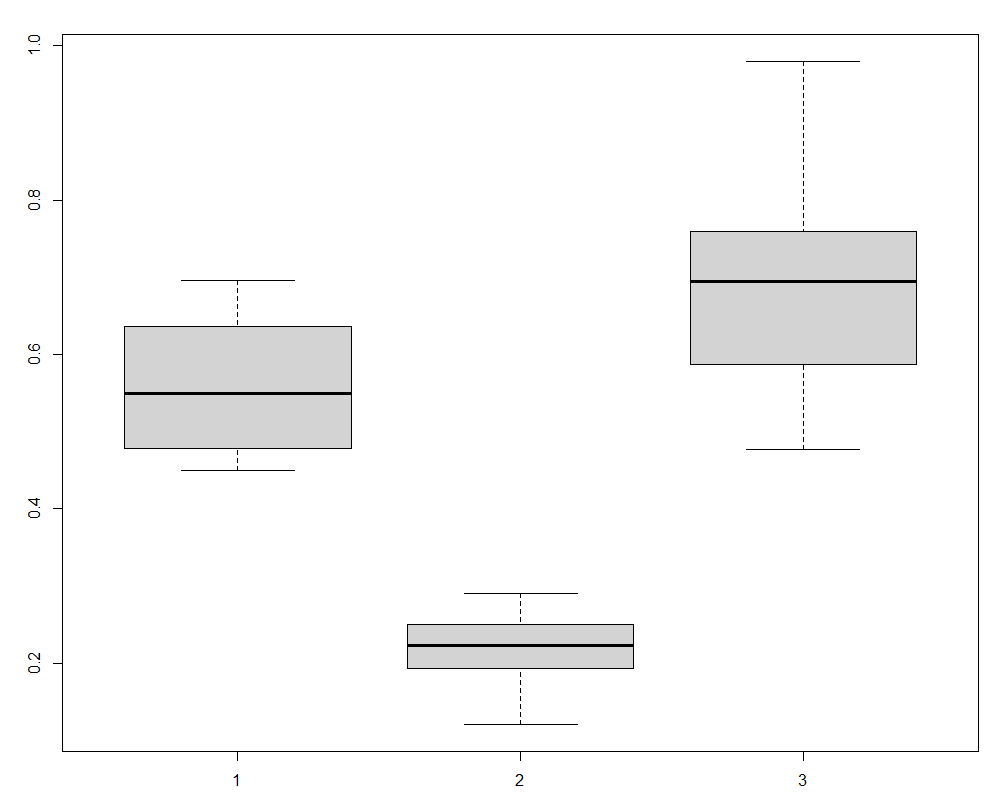
Como se observa no resultado do post-hoc, as técnicas SPT e LPT são mais semelhantes do que a técnica CUN com qualquer uma das restantes. Contudo, só este teste post-hoc não nos mostra qual das técnicas é a mais eficiente. Para isso realizou-se um boxplot com as técnicas LPT, CUN e SPT respetivamente:

Assim conseguimos afirmar que a técnica CUN é a mais eficiente tendo os valores mais baixos de makespan. Por outro lado, a técnica SPT é a menos eficiente. Também se comprova pelo post-hoc que as técnicas LPT e SPT são mais semelhantes que a técnica CUN com as restantes.

Uma imagem com texto

Descrição gerada automaticamente Para 20 instâncias:

Como se observa no resultado do post-hoc, as técnicas SPT e LPT são mais semelhantes do que a técnica CUN com qualquer uma das restantes. Contudo, só este teste post-hoc não nos mostra qual das técnicas é a mais eficiente. Para isso realizou-se um boxplot com as técnicas LPT, CUN e SPT respetivamente:



Assim conseguimos afirmar que a técnica CUN é a mais eficiente tendo os valores mais baixos de makespan. Por outro lado, a técnica SPT é a menos eficiente. Também se comprova pelo post-hoc que as técnicas LPT e SPT são mais semelhantes que a técnica CUN com as restantes.

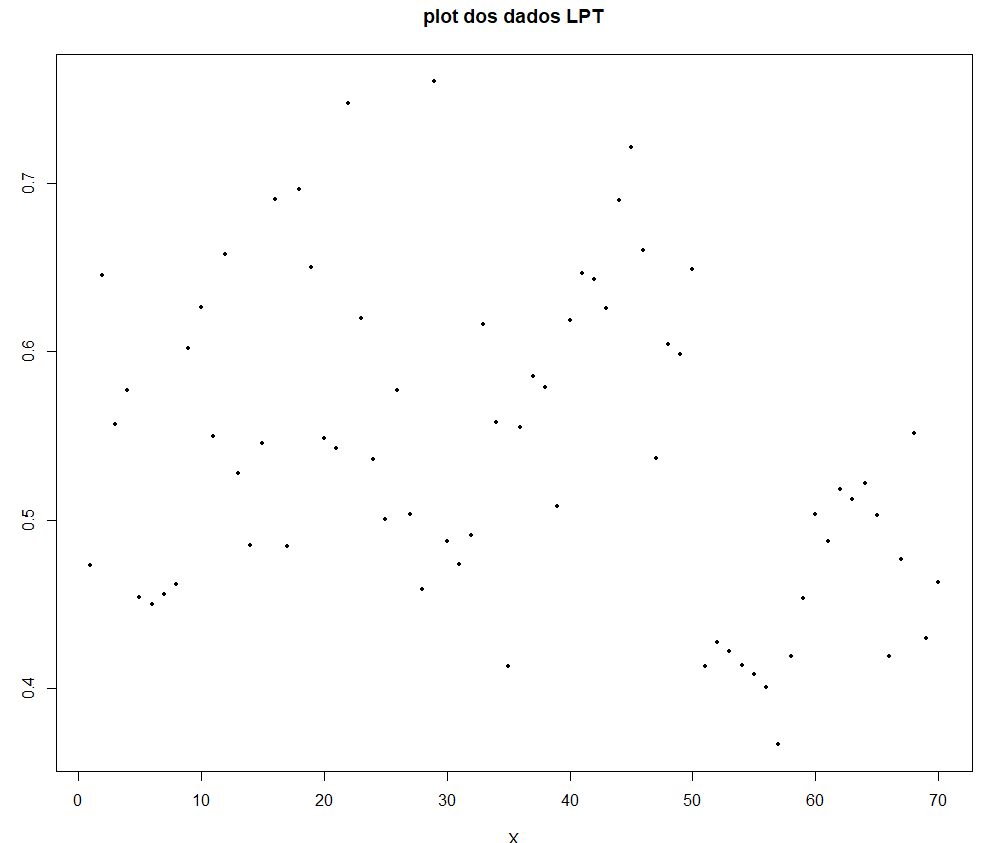
# **Comentar os resultados obtidos usando as duas medidas de desempenho usadas**

Através dos resultados dos testes de Friedman, verificamos que o p-value para as primeiras 10 instâncias é o mesmo para o makespan e para o makespan normalizado. O mesmo acontece com o p-value para as primeiras 20 instâncias.

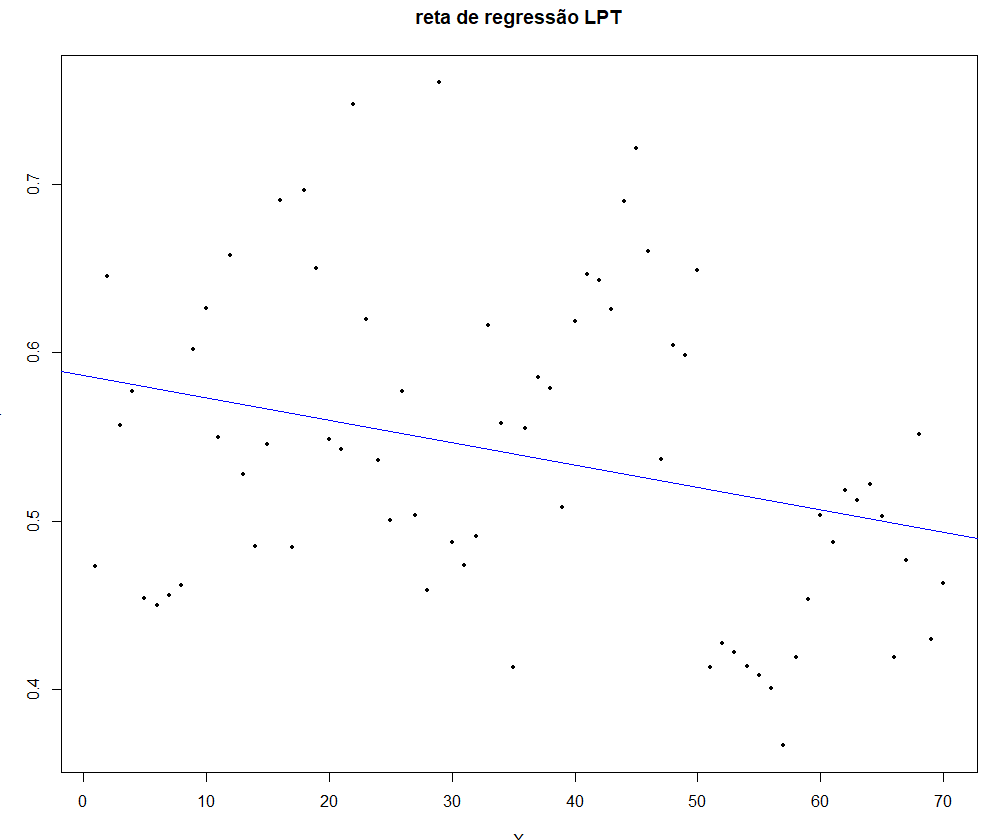
# **Usar os dados relativos às 70 instâncias de menor dimensão e determinar, para cada MH, a reta de regressão linear**

Após a importação dos dados, criou-se um dataframe sem a técnica otimizada. Após isso, e através do mesmo dataframe, retiraram-se apenas as 70 menores instâncias. Utilizou-se também o dataframe só com a técnica otimizada criado anteriormente. Para calcular o makespan normalizado para as 70 instâncias criaram-se dataframes para cada técnica e colocaram-se lá o resultado do valor normalizado. Após isso, começou-se o procedimento para se determinar a reta de regressão linear para cada técnica.

Inicialmente, para cada técnica, foi criado um plot para se poder observar os dados e verificar se é apropriada usar uma regressão linear. De seguida, utilizou-se a função “lm” que fornece o ajuste de um modelo de regressão linear. Após isso fez-se um plot dos dados juntamente com a reta de regressão calculada antes. No final extraiu-se a informação do comando “lm” e retiraram-se os coeficientes da regressão.



Plot para se poder observar os dados da técnica LPT.



Plot dos dados juntamente com a reta de regressão calculada antes da técnica LPT.

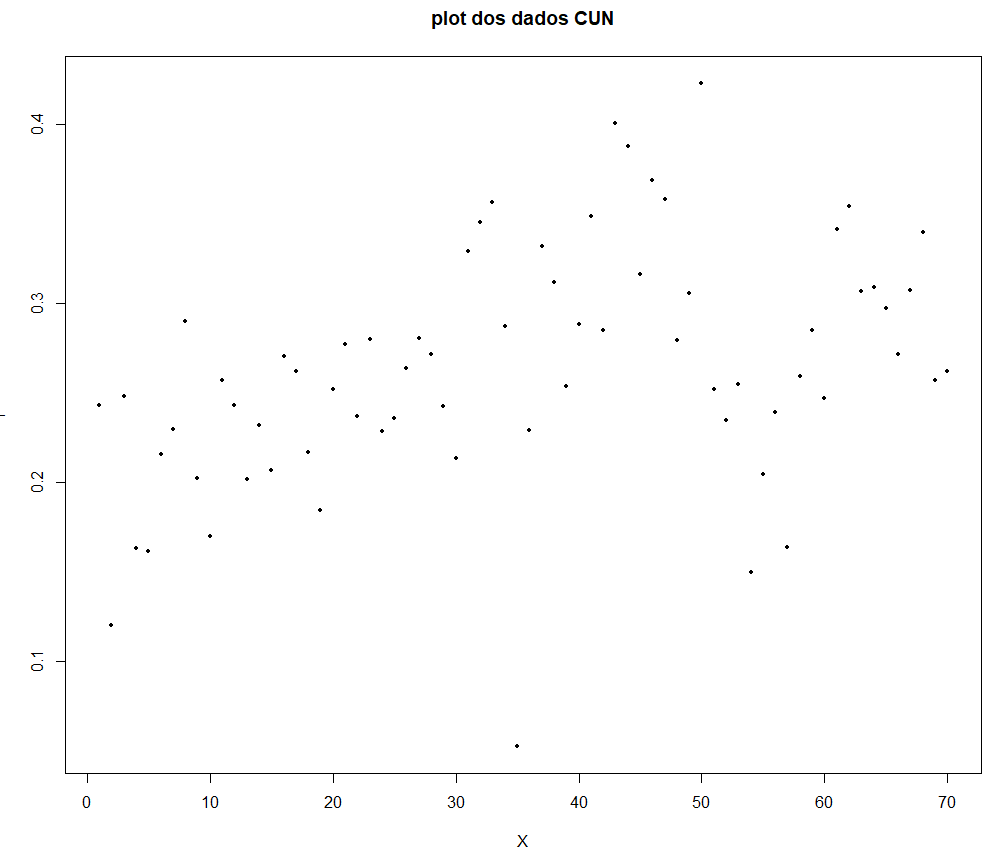
Uma imagem com texto

Descrição gerada automaticamente

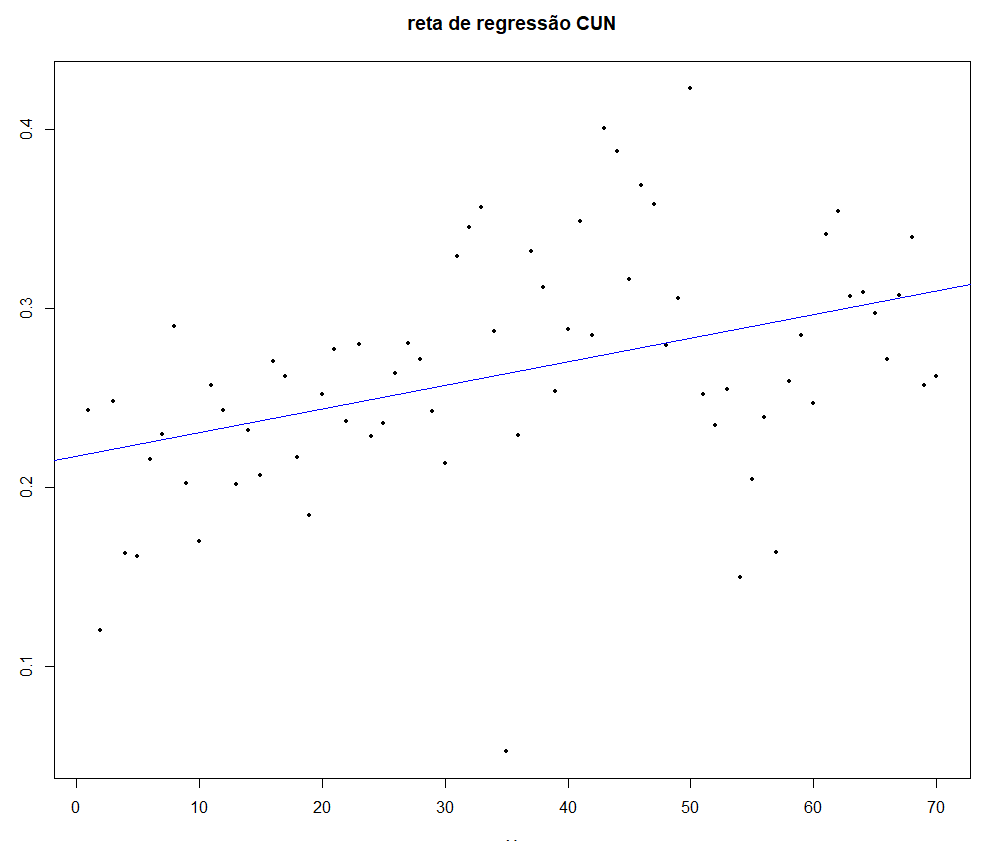
Uma imagem com texto

Descrição gerada automaticamente

Informação do comando “lm” e os coeficientes da regressão da técnica LPT.



Plot para se poder observar os dados da técnica CUN.

Plot dos dados juntamente com a reta de regressão calculada antes da técnica CUN.

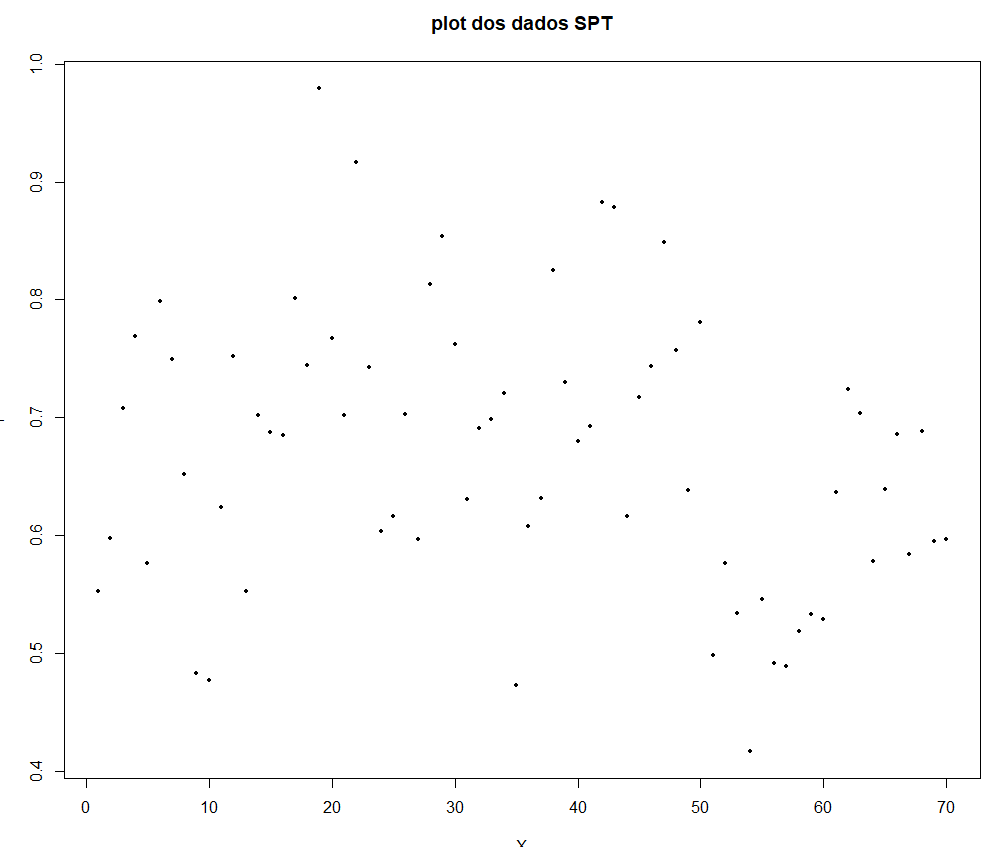
Uma imagem com texto

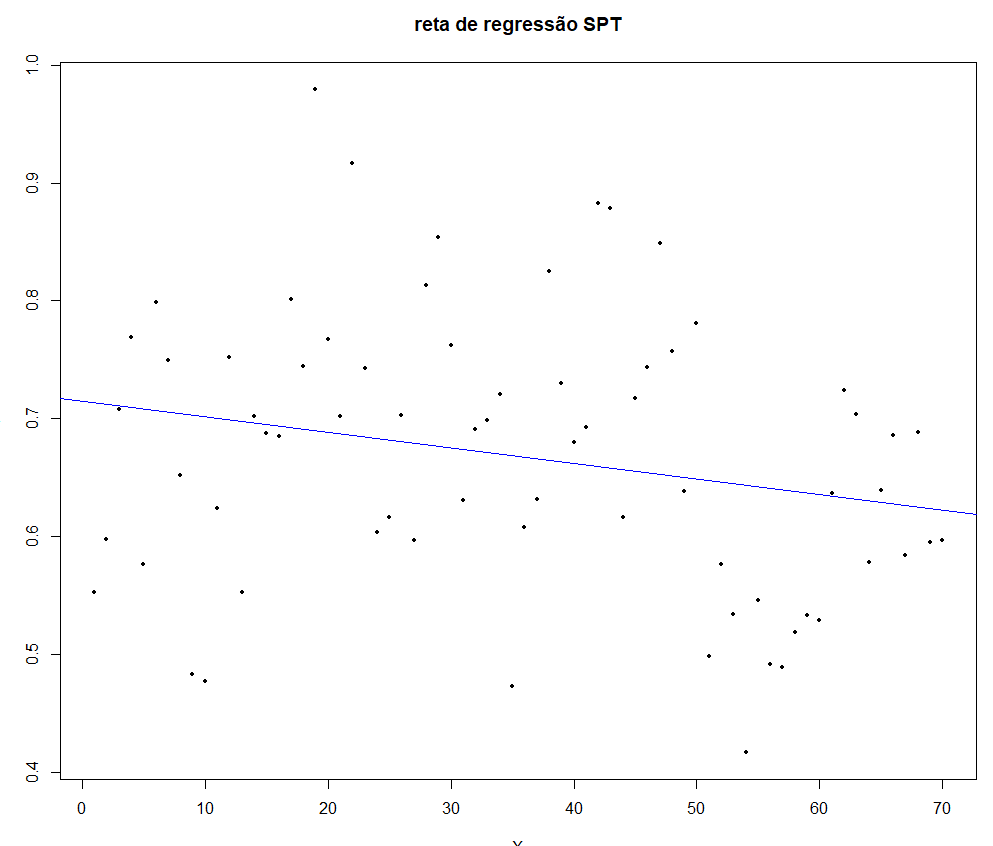
Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Informação do comando “lm” e os coeficientes da regressão da técnica CUN.



Plot para se poder observar os dados da técnica SPT.

Plot dos dados juntamente com a reta de regressão calculada antes da técnica SPT.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Informação do comando “lm” e os coeficientes da regressão da técnica SPT.

Concluindo, verifica-se que as retas de regressão são:

* LPT: Yˆ = 0.5866199 - 0.00133442X
* CUN: Yˆ = 0.2172385 + 0.001322468X
* SPT: Yˆ = 0.7149484 - 0.001317484X

# **Verificar se os pressupostos sobre os resíduos são verificados (normalidade, homocedasticidade e independência)**

Neste exercício, uma vez que se calculou a regressão linear, utilizou-se esta mesma para se verificaram os pressupostos da normalidade, homocedasticidade e independência de cada técnica. Assim, e para cada uma, foi utilizado um teste de Shapiro que verifica se algum dos resíduos segue uma distribuição normal. Para a homocedasticidade foi utilizado um método gráfico com o plot dos resíduos vs os valores ajustados e plot dos resíduos vs os valores da variável X. Depois dividiram-se os dados em dois conjuntos e testou-se se a variância era igual. Por fim, utilizou-se um teste de Durbin-Watson para verificar se os resíduos eram ou não independentes.

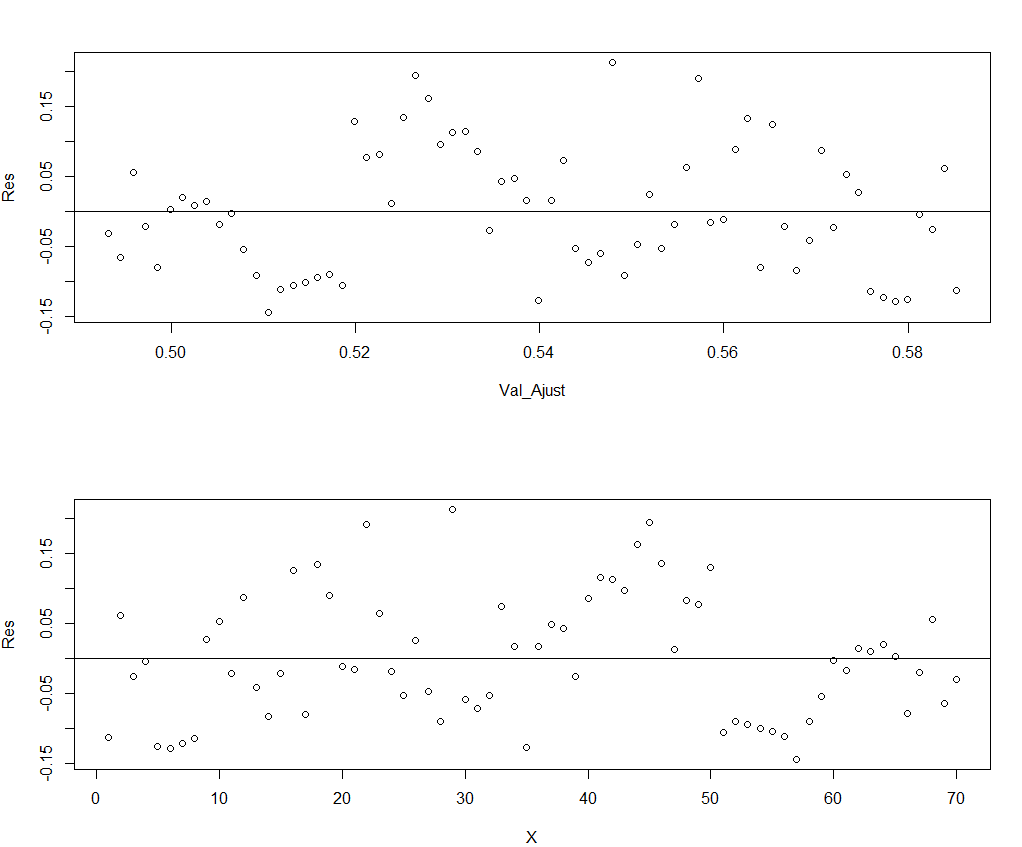
Para a técnica LPT:

Normalidade: H0:Resíduos seguem a normalidade VS H1: Resíduos não seguem a normalidade

Uma imagem com texto

Descrição gerada automaticamente

Considera-se alfa=0.05. Como o p-value é inferior rejeita-se a condição de normalidade. Não podemos efetuar inferência estatística mas iremos testar as outras condições

Homocedasticidade: H0:Resíduos seguem a homocedasticidade VS H1: Resíduos não seguem a homocedasticidade

Uma imagem com texto

Descrição gerada automaticamente

Podemos assumir que a condição de homocedasticidade é verificada pois p-value é superior a alfa (considerando alfa=0.05) logo rejeita-se a hipótese nula.

Independência: H0 : Os resíduos são independentes VS H1 : Os resíduos não são independentes



Podemos assumir que a condição de independência é rejeitada pois o p-value é inferior a alfa.

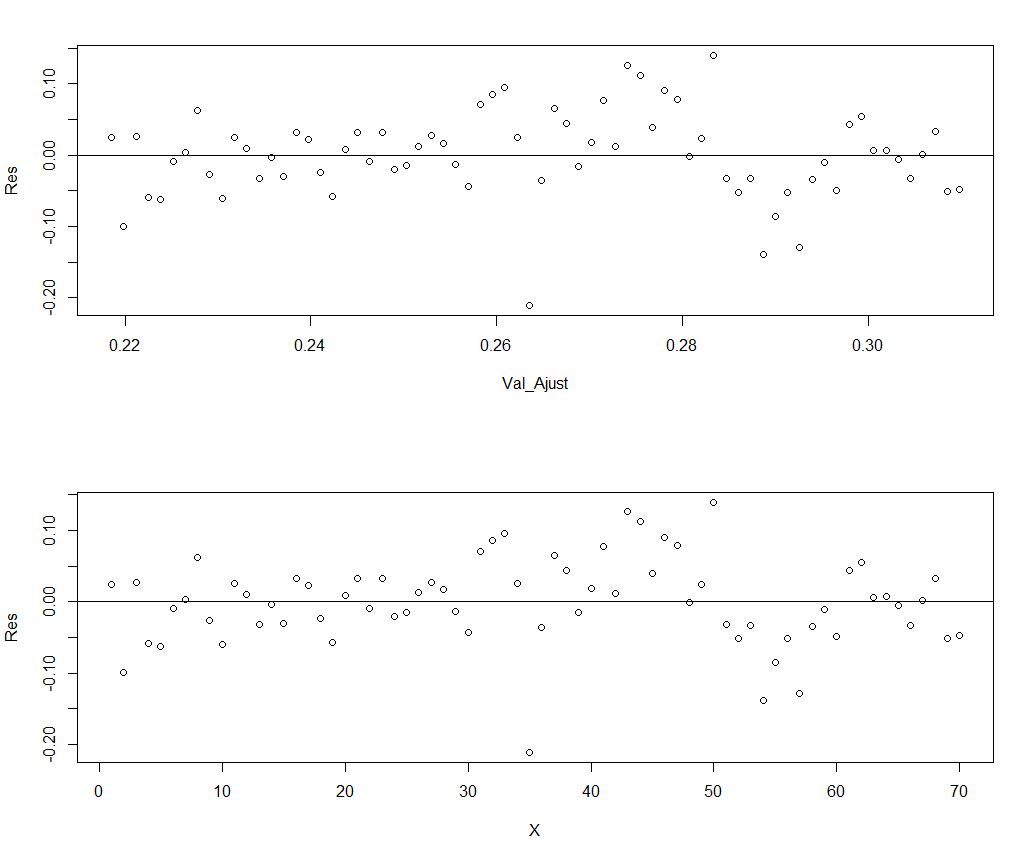
Para a técnica CUN:

Normalidade: H0:Resíduos seguem a normalidade VS H1: Resíduos não seguem a normalidade

Uma imagem com texto

Descrição gerada automaticamente

Considera-se alfa=0.05. Como o p-value é superior não se rejeita a condição de normalidade. Condição de normalidade é verificada.

Homocedasticidade: H0:Resíduos seguem a homocedasticidade VS H1: Resíduos não seguem a homocedasticidade

Uma imagem com texto

Descrição gerada automaticamente

Podemos assumir que a condição de homocedasticidade é verificada pois p-value é superior a alfa (considerando alfa=0.05) logo rejeita-se a hipótese nula.

Independência: H0 : Os resíduos são independentes VS H1 : Os resíduos não são independentes

Uma imagem com texto

Descrição gerada automaticamente

Podemos assumir que a condição de independência é rejeitada pois o p-value é inferior a alfa.

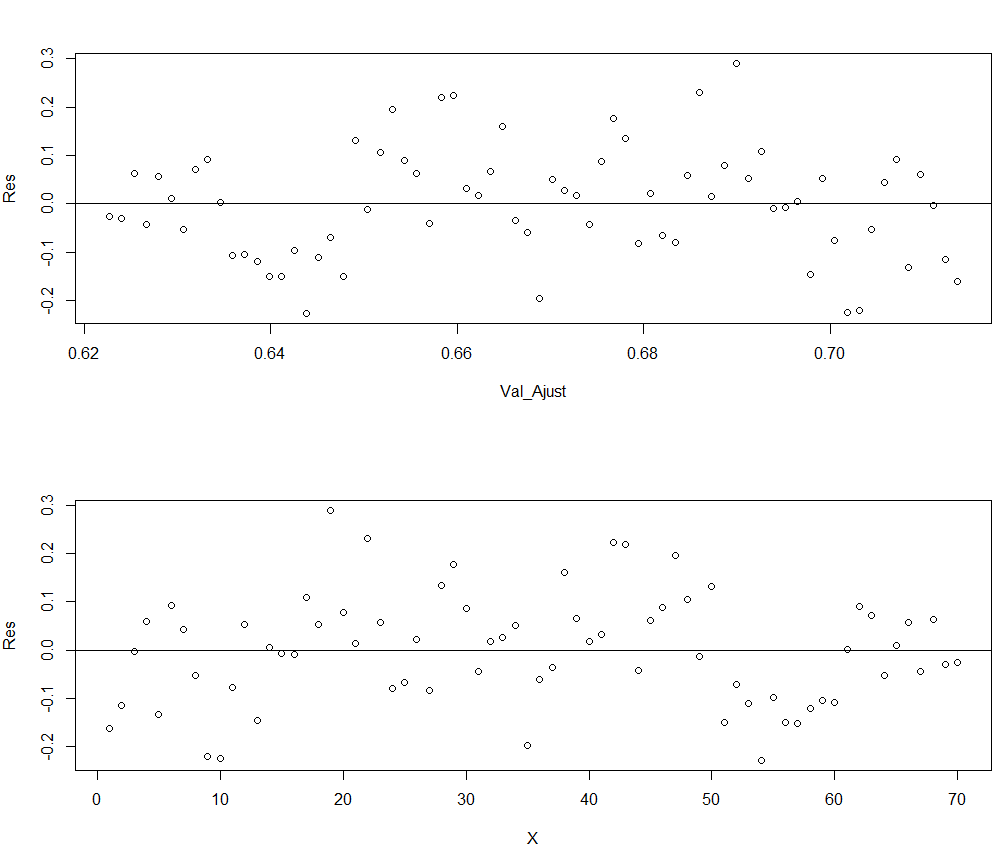
Para a técnica SPT:

Normalidade: H0:Resíduos seguem a normalidade VS H1: Resíduos não seguem a normalidade

Uma imagem com texto

Descrição gerada automaticamente

Considera-se alfa=0.05. Como o p-value é superior não se rejeita a condição de normalidade. Condição de normalidade é verificada.

Homocedasticidade: H0:Resíduos seguem a homocedasticidade VS H1: Resíduos não seguem a homocedasticidade

Uma imagem com texto

Descrição gerada automaticamente

Podemos assumir que a condição de homocedasticidade é verificada pois p-value é superior a alfa (considerando alfa=0.05) logo rejeita-se a hipótese nula.

Independência: H0 : Os resíduos são independentes VS H1 : Os resíduos não são independentes

Uma imagem com texto

Descrição gerada automaticamente

Podemos assumir que a condição de independência é rejeitada pois o p-value é inferior a alfa.

# **Comentar os resultados obtidos nas duas alíneas anteriores**

LPT- condição de normalidade não é verificada, condição de homocedasticidade é verificada e condição de independência não é verificada

CUN- condição de normalidade é verificada, condição de homocedasticidade é verificada e condição de independência não é verificada

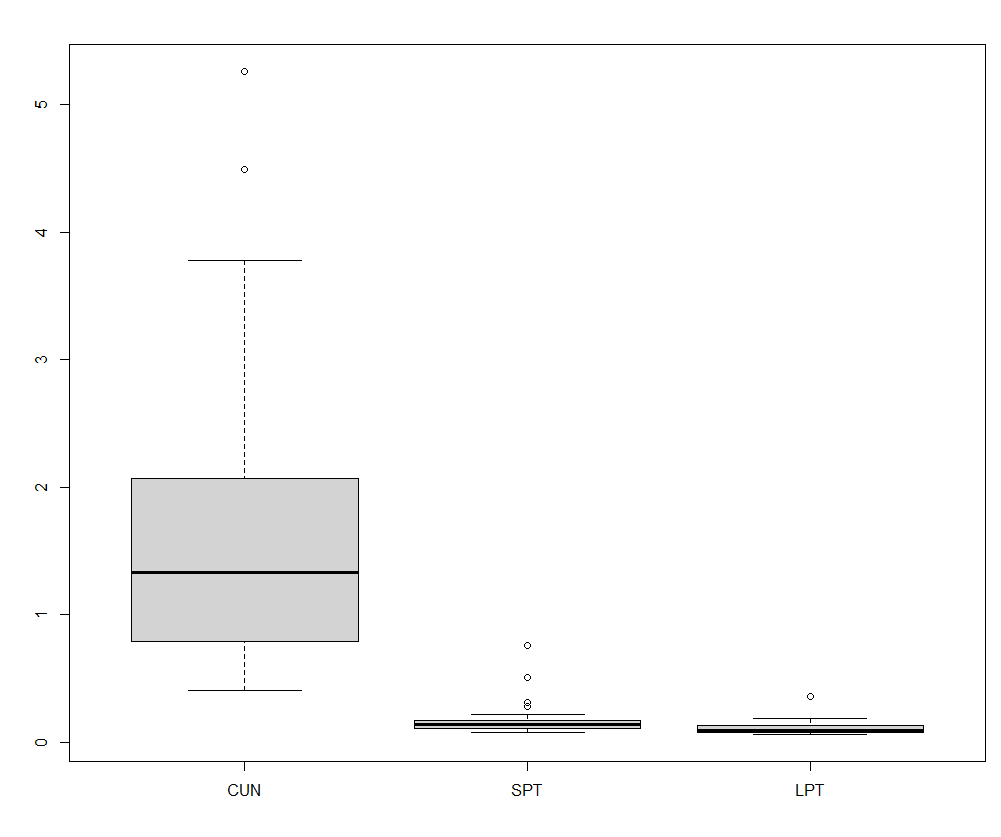
SPT- condição de normalidade é verificada, condição de homocedasticidade é verificada e condição de independência não é verificada

# **4.2 Exercício 2**

Este exercício divide-se em 4 alíneas que pedem vários tipos de filtragem de dados: **4.2.1** construir um boxplot que contenha os tempos de processamento de cada MH na resolução de cada instância; **4.2.2** verificar se existem diferenças significativas nos tempos médios de processamento entre as três técnicas; **4.2.3** no caso da resposta da alínea anterior ser positiva, identificar qual a MH mais eficiente; **4.2.4** determinar a matriz de correlação entre os tempos de processamento de cada MH e interpretar os resultados.

# **4.2.1 construir um boxplot que contenha os tempos de processamento de cada MH na resolução de cada instância**

Após importação dos dados, criaram-se 3 dataframes com os dados relativos a cada técnica. De seguida, juntaram-se todos os dados com a função “cbind” e criou-se o boxplot.



Como se observa no boxplot, a técnica LPT é a mais eficiente, seguida da SPT e por fim a CUN.

# **4.2.2 verificar se existem diferenças significativas nos tempos médios de processamento entre as três técnicas**

Após a importação dos dados, colocou-se num dataframe os dados apenas com a técnica usada, tempo de processamento e o problema em causa. Colocaram-se os valores no tipo numérico e realizou-se um teste de Friedman pois está a haver a comparação entre três amostras emparelhadas. Assim, temos:

* H0: as três técnicas são iguais VS H1: há pelo menos uma técnica que é diferente das restantes

Uma imagem com texto

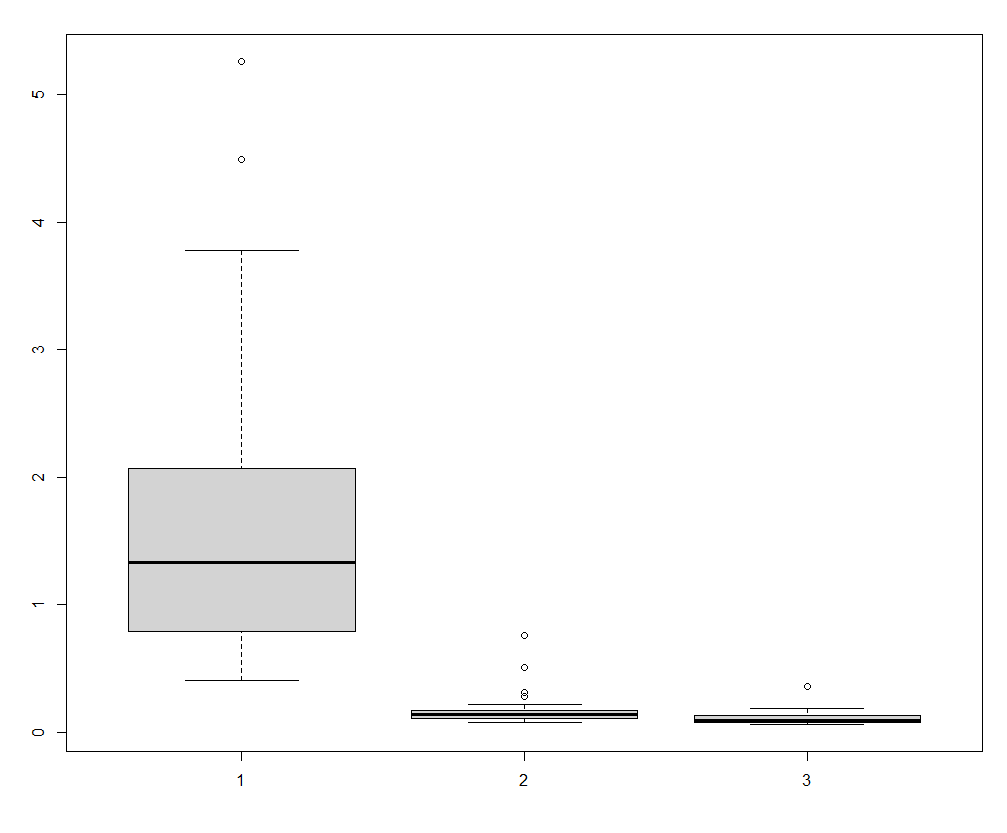
Descrição gerada automaticamente

Considerando o alfa=5%, e observando o valor do p-value, podemos rejeitar a hipótese H0. Há assim pelo menos uma técnica que é diferente das restantes.

# **4.2.3 no caso da resposta da alínea anterior ser positiva, identificar qual a MH mais eficiente**

Uma imagem com texto

Descrição gerada automaticamenteApós a importação dos dados passou-se à realização do exercício. Foi utilizado o teste “pairwise.wilcox.test”-post hoc- com a técnica de ajuste “bonferroni” para se verificar as diferenças entre as técnicas.

 Como se observa no resultado do post-hoc, as técnicas SPT e LPT são mais semelhantes do que a técnica CUN com qualquer uma das restantes. Contudo, só este teste post-hoc não nos mostra qual das técnicas é a mais eficiente. Para isso realizou-se um boxplot com as técnicas CUN, SPT e LPT respetivamente:

Assim conseguimos afirmar que a técnica CUN é a menos eficiente tendo os valores mais elevados de makespan. Por outro lado, a técnica LPT é a mais eficiente. Também se comprova pelo post-hoc que as técnicas LPT e SPT são mais semelhantes que a técnica CUN com as restantes.

# **4.2.4 determinar a matriz de correlação entre os tempos de processamento de cada MH e interpretar os resultados**

Após a importação dos dados, realizou-se a matriz de correlação entre os tempos de processamento de cada MH através da função “rcorr”.

Uma imagem com texto

Descrição gerada automaticamente

Assim tem-se:

* CUN e SPT- -0.10 - pouco correlacionadas negativamente
* CUN e LPT- 0.24- pouco correlacionadas positivamente
* SPT e LPT- 0.37- pouco correlacionadas positivamente mas mais fortemente correlacionadas do que CUN e LPT

# **Referências**

[1]. HEUMANN, C., M. SCHOMAKER and SHALABH, Introduction to statistics and data analysis, Springer International Publishing, 2016.

[2].MADUREIRA, ANA MARIA, Aplicação de Meta-Heurísticas ao Problema de Escalonamento em Ambiente Dinâmico de Produção Discreta, PhD Thesis at Universidade do Minho, 2003.