

QUADRADOS LATINOS COM APLICAÇÕES EM ENGENHARIA DE SOFTWARE

Iván Fernando Herrera Sánchez

Orientador: Prof. Dr. Cristiano Ferraz
Co-orientador: Prof. Dr. Paulo Borba

Área de concentração: Estatística aplicada

Dissertação submetida como requerimento parcial para obtenção do grau de
Mestre em Estatística pela Universidade Federal de Pernambuco

Recife, fevereiro de 2011

Catálogo na fonte
Bibliotecária Joana D'Arc L. Salvador, CRB 4-572

Sánchez, Iván Fernando Herrera.

Quadrados latinos com aplicações em engenharia
de software / Iván Fernando Herrera Sánchez. -

Recife: O Autor, 2011.

ix, 89.f.: fig. tab.

Orientador: Cristiano Ferraz.

Dissertação (Mestrado) - Universidade Federal de
Pernambuco, CCEN, Estatística, 2011.

Inclui bibliografia e apêndice.

1. Planejamento experimental. 2. Linhas de produtos
de software. I. Ferraz, Cristiano (orientador). II. Título.

519.57 (22. ed.)

MEI 2011-019

Universidade Federal de Pernambuco
Pós-Graduação em Estatística

24 de fevereiro de 2011

Nós recomendamos que a dissertação de mestrado de autoria de

Iván Fernando Herrera Sánchez

intitulada

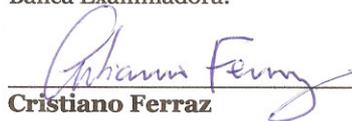
“Quadrados Latinos com Aplicações em Engenharia de Software”

seja aceita como cumprimento parcial dos requerimentos para o grau de Mestre em Estatística.



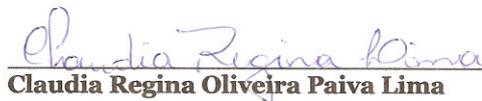
Coordenador da Pós-Graduação em Estatística

Banca Examinadora:



Cristiano Ferraz

orientador



Claudia Regina Oliveira Paiva Lima



Marcel de Toledo Vieira (UFJF)

Este documento será anexado à versão final da dissertação.

Agradecimentos

Quero agradecer profundamente e de todo coração aos meus pais e aos meus irmãos, já que sem o apoio incondicional deles seria impossível haver chegado até este ponto. O amor e respaldo de minha família é a maior motivação para nunca desistir dos retos que implica fazer um mestrado em estatística na UFPE.

É incontável a quantidade de pessoas que me ajudaram ao longo deste tempo, mas ressalto a grandiosa e importante ajuda que recebo de meu amigo Marco Soracipa, um dos melhores estatísticos que conheço pessoalmente. Agradeço infinitamente a meu amigo Marcelo, do qual recebi todo o apoio e ajuda possível nesses primeiros períodos de mestrado. Éramos um bom time para todos os trabalhos que precisamos apresentar. Definitivamente um especial abraço de agradecimento para Luz Milena, Nicolas e Fernando, os colombianos que sempre fizeram sentir-me como em casa. Mas o beijo especial vai ser para minha namorada Mariana, com quem estou e vou a estar eternamente agradecido por todo o apoio oferecido, assim como a paciência, confiança e compreensão em todo momento.

Naqueles instantes de saudade e tristeza foi muito grato ter a companhia de minhas amigas Carolina, Diana, Janete e Alejandra. A colaboração inicial oferecida por Miguel, Alejandro e Piña foi imensa. Sempre é bom saber que você conta com a ajuda de pessoas que estão sempre ai para você, como é o caso de Cristiane, tia Sirlan, senhora Carminha e senhora Rosinha.

Quero comentar que cheguei ao Brasil com uma universidade em meu coração que era a Universidade Nacional de Colômbia e agora regresso com o orgulho de haver pertencido à Universidade Federal de Pernambuco e especialmente ao programa de pós-graduação em estatística. Aqui quero agradecer ao professor Leandro, Sérgio, Raydonal e à professora Audrey, vocês são do melhor, muito, muito obrigado por tudo. Meu anjo Valeria, sem você realmente não tivesse conseguido ficar no Brasil. A ajuda por você oferecida é mais do que você pode dar, mas tua bondade não tem limites.

Por último, e definitivamente o jogador mais importante para poder conseguir este logro, é a CAPES. Seu apoio financeiro é determinante para ter a estabilidade e confiança durante todo o tempo de estudo.

Resumo

Engenharia de software visa o desenvolvimento eficiente e a manutenção sistemática de software, de forma a atender satisfatoriamente as necessidades impostas pelo mercado. Nessa área, a montagem de linhas de produto que minimizem variáveis de custo e de tempo para lançamento de uma família de produtos é estratégica para a garantia de competitividade. Quando se tem disponibilidade de mais de uma abordagem para a montagem de tais linhas, faz-se necessário identificar qual delas seria a mais eficiente. Esta dissertação aborda a utilização de técnicas de planejamento estatístico de experimentos para projetar estudos em engenharia de software que permitam avaliar a eficiência de abordagens de desenvolvimento de software. A motivação do trabalho está no projeto de experimentos para comparação das abordagens PLUSS (modelagem de casos de uso em linhas de produtos para sistemas e software) e MSVCM (modelagem de variações em cenários de casos de uso por mecanismos transversais) de montagem e manutenção de linhas de produtos de software. A dissertação tem como foco o estudo da adequação e desempenho de planos experimentais em quadrados latinos aplicados a engenharia de software. Tais planos possuem um apelo prático importante, pois permitem o controle de duas fontes de variação sistemáticas simultâneas durante a experimentação. Nesta dissertação, o desempenho estatístico de planos em quadrados latinos de ordem 2 é investigado através da teoria de aleatorização e de simulações de Monte Carlo.

Palavras chave: planejamento de experimentos, linhas de produtos de software e teoria de aleatorização.

Abstract

Software engineering aims at developing efficient and systematic maintenance of software in order to satisfactorily meet the needs imposed by the market. In this area, the assembly of product lines that minimize the variables of cost and time to launch a family of products is very important for ensuring competitiveness. When you have availability of more than one approach for the assembly of such lines, it is necessary identify which of them would be more efficient. This dissertation investigates the use of techniques of statistical design of experiments to design studies in software engineering to evaluate the effectiveness of approaches to software development. The motivation of the work is the design of experiments to compare the approaches PLUSS (Product Line Use Case Modeling for Systems and Software Engineering) and MSVCM (Modeling Scenario Variability as Crosscutting Mechanisms) assembly and maintenance software product lines. The dissertation focuses on the study of the adequacy and performance of experimental plans in Latin squares applied to software engineering. Such plans have an important practical appeal because they allow control of two sources of systematic variation simultaneously during the experiment. In this dissertation, the statistical performance of plans in Latin squares of order 2 is investigated through the theory of randomization and Monte Carlo simulations.

Keywords: design of experiment, software product lines and theory of randomization.

Sumário

Lista de Figuras	vii
Lista de Tabelas	ix
1 Introdução	1
2 Experimentos em quadrados latinos	5
2.1 Definição	6
2.2 Modelo linear para um plano em quadrado latino.	8
2.3 Análise de variância	14
2.4 Quadrados latinos replicados	16
2.4.1 Linhas e colunas cruzadas dentro das réplicas.	17
2.4.2 Colunas cruzadas com linhas embutidas nas réplicas.	19
2.4.3 Linhas cruzadas com colunas embutidas nas réplicas.	21
2.4.4 Linhas e colunas embutidas dentro das réplicas.	22
3 Engenharia de software	24
3.1 Definição	25
3.2 Linhas de produtos de software	27
3.2.1 Fundamentos da abordagem de engenharia de linha de produto de software	30
3.2.2 PLUSS e MSVCM	31
4 Quadrados latinos 2x2 e aplicações a engenharia de software	38
4.1 Processo de aleatorização para um plano experimental	38
4.2 Teste de aleatorização em planos experimentais	42

4.3	Análise de um quadrado latino 2×2 com réplicas	47
4.3.1	Estudo de simulação	50
4.3.2	Resultados do estudo de simulação	54
5	Cálculo da função poder do teste F	71
6	Conclusões e recomendações	76
	Referências Bibliográficas	79
	Apêndice	80

Lista de Figuras

2.1	Mesmos analistas e mesmas linhas de produto	19
2.2	Mesmos analistas e diferentes linhas de produto	20
2.3	Mesmas linhas de produto e diferentes analistas	21
2.4	Diferentes analistas e diferentes linhas de produto	23
3.1	Economia em linha de produtos de software. [15]	28
3.2	Diferentes abordagem para o desenvolvimento de uma linha de produtos de software	29
3.3	Diferentes configurações dos cenários compra e procura de pro- dutos. [2]	33
3.4	Cenários Buy Product usando PLUS. [2]	34
3.5	Cenários MSVCM de Buy Product e Search Product. [2] . . .	36
3.6	Exemplo do conhecimento de configurações. [2]	37
3.7	Construção do conhecimento de configurações. [2]	37
4.1	Níveis de significância do teste aleatório frente ao teste apro- ximado F para um PCA	46
4.2	Gráfico de um PABC com oito réplicas	48
4.3	Graus de liberdade da ANOVA para um PCA, PABC e QL [8] . . .	49
4.4	Efeito numérico na perda de graus de liberdade no erro	50
4.5	Saída gerada por SAS no processo de aleatorização	52
4.6	Quadrados latinos 2×2 com 5 réplicas. Rodada 1. Fase 01 . .	55
4.7	Quadrados latinos 2×2 com 5 réplicas. Rodada 1. Fase 02 . .	57
4.8	Quadrados latinos 2×2 com 11 réplicas. Rodada 2. Fase 01 .	59
4.9	Quadrados latinos 2×2 com 11 réplicas. Rodada 2. Fase 01 .	60
4.10	Quadrados latinos 2×2 com 8 réplicas. Rodada 2. Fase 02 . .	62

4.11	Quadrados latinos 2×2 com 8 réplicas. Rodada 2. Fase 02 . .	63
4.12	Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 01 . .	65
4.13	Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 01 . .	66
4.14	Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 02 . .	68
4.15	Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 02 . .	70
5.1	Gráficos da função de poder para diferentes valores de $\hat{\sigma}_e^2$. . .	74
6.1	Relação de custos entre PLUSS e MSVCM	77

Lista de Tabelas

2.1	ANOVA de um quadrado latino	15
2.2	ANOVA para o modelo (2.15)	18
2.3	ANOVA para o modelo (2.16)	20
2.4	ANOVA para o modelo (2.17)	22
2.5	ANOVA para o modelo (2.18)	23
4.1	Cenários de simulação quadrado latino 2×2 com réplicas . . .	54
5.1	Níveis de significância do teste F e do teste exato	72

CAPÍTULO 1

Introdução

Engenharia de software é a área da computação cujo objetivo é desenvolver e manter novos produtos de software no mercado através de atualizações sistemáticas com qualidade, custos e prazos aceitáveis [16]. Nessa área, é estratégica a utilização eficiente de linhas de produtos de software (*Software Product Line-SPL*), desenvolvidas com o objetivo de reduzir custos e tempo em uma família de sistemas de mercado com domínio comum.

Os sistemas pertencentes a uma SPL são gerados por meio de um processo de derivação do produto, que seleciona e configura artefatos reutilizáveis. Esses artefatos incluem os resultados de diferentes disciplinas (ou fases), como requisitos, projeto, implementação e testes e estão disponíveis para todos os membros da SPL. Além disso, reduzem os custos de lançamento de novos produtos e proporcionam melhoras na manutenção do produto.

A gestão de variabilidade, devido à sua natureza transversal inerente, é um desafio comum na adoção da linha de produção de software. Sempre que características não triviais são requeridas os pontos de variação são espalhados ou ficam dispersos através de artefatos na SPL. Além disso, algumas abordagens incluem variantes do produto, e assim, entrelaçam a configuração

conhecida com artefatos de engenharia de software.

Vários mecanismos de orientação a aspectos têm sido propostos para modularizar a especificação de interesses transversais. Embora essas técnicas minimizem problema dos pontos de variação, dado que podem ser usadas para modularizar a especificação de certos recursos, elas não suportam as diferentes fontes de variabilidade por requerimentos que se apresentam na SPL. Com relação ao segundo problema, as abordagens existentes para especificar a variabilidade em cenários de caso de uso não oferecem uma clara separação entre a gestão da variabilidade e a especificação de cenário. Como consequência, no caso onde detalhes sobre as variantes do produto são entrelaçadas com cenários de caso de uso, a remoção de uma variante da linha de produto requer mudanças em todos os cenários relacionados. Em resumo, é difícil evoluir de forma independente a gestão da variabilidade e especificação de cenário.

Diversas abordagens para representar pontos de variação em cenários textuais são propostas, uma delas é modelagem de casos de uso em linhas de produtos para sistemas e software, conhecido como PLUS. No entanto, apesar dos benefícios da representação da variabilidade, essa abordagem não apresenta uma separação clara entre a gestão da variabilidade e especificação de cenário, o que compromete a evolução de uma SPL. Além disso, PLUS pode ser classificada como uma técnica de anotação, que não é adequada para modularizar a natureza transversal de certas características, tem legibilidade pobre e conduz a uma redução da manutenção.

A tese *Modeling Software Product Line Variability in Use Case Scenarios* [2] propõe uma nova abordagem, denominada modelagem de variações em cenários de casos de uso por mecanismos transversais (MSVCM), para representar a variabilidade em cenários de caso de uso. Essa abordagem, visa aumentar a separação de interesses entre o comportamento comum e variantes, bem como aumentar a separação de interesses entre o cenário de especificações e a gestão da variabilidade.

Assim, o interesse principal é saber se existe diferença entre as duas abor-

dagens mencionadas anteriormente, PLUSS e MSVCM. Segundo Bonifácio [2], espera-se que na fase de desenvolvimento a estimativa do tempo para MSVCM seja maior na abordagem de modelagem de casos de uso em linhas de produtos para sistemas e software PLUSS, mas o ganho da abordagem que se propõe está na parte da manutenção. Para testar tais hipóteses realizou-se uma série de experimentos.

O plano experimental usado para a realização desses experimentos foi um plano em quadrados latinos, pois quando dois tipos de fontes de variação são identificados devido à natureza ou ao arranjo das unidades experimentais, o quadrado latino é um plano efetivo para afastar e diferenciar estas fontes de variação do erro experimental. Empregaram-se especificamente um plano aleatorizado em quadrados latinos de ordem 2 devido a que são testados dois tratamentos, neste caso, as abordagens PLUSS e MSVCM para o desenvolvimento e manutenção de linhas de produtos de software. Como também se faz tecnicamente inviável testar mais de dois produtos de software por cada analista, pois resulta exaustivo, já que o tempo usado por parte do analista para desenvolver uma abordagem para um produto de software é extenso e cansativo, implicando análises tendenciosas.

O fato de fazer replicação num quadrado latino 2×2 é importante para poder incrementar os graus de liberdade do erro experimental. O número de réplicas depende, geralmente, da situação experimental em questão. Hinkelmann & Kempthorne [11] comentam que para quadrados pequenos, geralmente de ordem inferior a 4, a realização de um número de réplicas baixo pode dificultar a obtenção de níveis de precisão desejáveis. O teste exato é executável para um número pequeno de quadrados, por exemplo, 6 ou menos, pelo qual é aconselhável optar por este teste.

Neste sentido, os objetivos desta dissertação são:

- estudar a aproximação do teste F ao teste exato em quadrados latinos 2×2 usando as possíveis formas de replicar quadrados latinos;
- determinar o número de réplicas apropriado em relação à função poder

do teste aproximado;

- calcular o teste exato para dados obtidos no experimento realizado na tese [2] e analisar os resultados.

Este trabalho está dividido em seis capítulos. No primeiro capítulo é desenvolvida uma breve introdução sobre a motivação que levou à escolha do assunto a tratar e são apresentados os principais tópicos discutidos nesta dissertação. No segundo capítulo é realizada uma revisão de literatura na área de planejamento de experimentos, focada nos planos aleatorizados em quadrados latinos 2×2 . Para o terceiro capítulo apresenta-se uma revisão dos conceitos fundamentais da área de engenharia de software, principalmente, em linhas de produtos de software. No quarto capítulo é justificado o uso e importância dos quadrados latinos 2×2 em engenharia de software e posteriormente é descrito e apresentado os resultados das simulações do estudo de Monte Carlo para analisar a aproximação do teste F ao teste exato. Para o capítulo cinco são calculados os valores do teste exato para dados reais e para finalizar o capítulo é apresentado o cálculo da função poder do teste F , para diferentes parâmetros. No sexto capítulo se efetua as conclusões e recomendações do experimento realizado para analisar duas abordagens em linhas de produtos de software, comentando alguns aspectos da tese de doutorado [2].

CAPÍTULO 2

Experimentos em quadrados latinos

Suponha que deseja-se comparar e pesquisar diferentes abordagens de linhas de produtos de software, mas fontes de variabilidade sistemática na resposta das unidades experimentais são detectadas e identificadas, por exemplo:

- i) diferentes produtos de software com funcionalidades e objetivos distintos;
- ii) diferentes analistas com formações acadêmicas distintas, habilidades e destrezas desiguais.

Os produtos de software são obtidos de diferentes linhas de produtos e possivelmente com funções e objetivos distintos. Também existem analistas formados em áreas diversas, com anos de experiência desiguais e virtudes dificilmente semelhantes. Se todos os produtos de software tivessem funcionalidades uniformes, poderia ser usado o plano aleatorizado em blocos completos, definindo os analistas como fator de bloqueio. Por outro lado, se os analistas apresentassem habilidades e execuções semelhantes no desenvolvimento das diferentes abordagens para um produto de software, os produtos

de software seriam tomados como fator de bloqueio num plano aleatorizado em blocos completos. No mais, pode-se determinar os dois fatores como bloqueio para eliminar a variabilidade sistemática, e assim, reduzir o erro experimental.

Suponha a existência de r produtos de software $\beta_1, \beta_2, \dots, \beta_r$ e c analistas especializados em linhas de produtos de software $\alpha_1, \alpha_2, \dots, \alpha_c$. Uma forma de proceder é obter de cada linha de produtos de software c produtos e fornecer blocos do tipo $(\beta_i\alpha_j)$, ou seja, combinar cada produto de software com cada analista. Quando se tem t tratamentos e se pode desenvolver t abordagem de linhas de produtos de software, ou seja, t tratamentos por cada bloco $(\beta_i\alpha_j)$, então é possível fazer desenvolvimento de cada produto de software por cada analista, com uma abordagem como tratamento. Assim, para esse experimento temos rc combinações e as observações podem ser analisadas de acordo com um plano aleatorizado em blocos completos com rc blocos e t tratamentos.

Esse experimento pode ser viável para alguns casos, mas impossível para outros, pois precisa de muitas observações e unidades experimentais. Em algumas situações tais planos são fisicamente impossíveis. De acordo com [11], tem-se outros planos experimentais para eliminação de fontes de variação, ou planos de linhas e colunas. Estes planos são chamados de planos do tipo quadrado latino. Um plano aleatorizado em quadrado latino é uma forma especial dos planos linha-coluna.

2.1 Definição

Um quadrado latino representa a forma mais simples de um plano linha-coluna e é usado para comparar t tratamentos em t linhas e t colunas, onde as linhas e colunas são dois fatores de bloqueio. Um quadrado latino de ordem t é um arranjo de t letras latinas em um quadrado de t linhas e t colunas, onde cada letra latina aparece uma vez só em cada linha e uma única vez em cada

coluna. Em geral, é um arranjo de t símbolos em uma ordem tal que cada símbolo acontece exatamente uma vez em cada linha e coluna. No contexto de planejamento de experimentos, as letras latinas são os tratamentos.

Um quadrado latino reduzido ou quadrado latino em forma padrão é um quadrado em que a primeira linha e a primeira coluna são arranjadas em ordem alfabética, por exemplo, para $t = 3$ tem-se

$$\begin{array}{ccc} A & B & C \\ B & C & A \\ C & A & B \end{array}$$

que é o único quadrado latino reduzido, assim como o caso em que $t = 2$, tal como se apresenta a seguir.

$$\begin{array}{cc} A & B \\ B & A \end{array}$$

Permutando linhas, colunas e letras o número de quadrados que pode ser gerado de um quadrado latino reduzido é $(t!)^3$, não necessariamente todos diferentes. Se todas as linhas e todas as colunas são permutadas, pode-se gerar $t!(t-1)!$ quadrados diferentes.

Em geral, o número de quadrados reduzidos de ordem t é denotado por T_t e o número total de quadrados latinos de ordem t por U_t , portanto,

$$U_t = t!(t-1)!T_t$$

No seguinte quadro, obtido de [11], tem-se a lista de T_t para $t = 2, \dots, 8$.

t	T_t
2	1
3	1
4	4
5	56
6	9.408
7	16.942.080
8	535.281.401.856

Neste ponto seria conveniente apresentar o processo de aleatorização para executar um plano aleatorizado em quadrado latino, mas por questões de

conveniência, este tópico é desenvolvido na seção 4.3.1, pois é o algoritmo-base nos processos de simulação efetuados para este estudo. Assim, assuma que a partir da execução de um processo de aleatorização deriva-se o modelo linear apresentado na seguinte seção.

2.2 Modelo linear para um plano em quadrado latino.

Seguindo a teoria apresentada em Hinkelmann & Kempthorne [11], o quadrado latino é analisado como um plano de erro e controle. Suponha que (i, j, k) faz referência à linha, coluna e tratamento, respectivamente, de uma unidade experimental. Assim, tem-se t^3 respostas possíveis de cada tratamento aplicado a cada unidade experimental (UE), e dessa população de resultados, toma-se uma amostra baseada em um quadrado latino $t \times t$, ou seja, um quadrado obtido através de um processo de aleatorização, como é explicado na seção 4.3.1. Essas amostras têm propriedades de balanceamento, ou seja, o mesmo número de observações por tratamento dentro de cada quadrado latino. Essa propriedade é de grande importância, especialmente quando os tratamentos são comparados.

Suponha aditividade unidade-tratamento no sentido exato. Desse modo, a resposta do tratamento k na linha i e coluna j é escrito como

$$T_{ijk} = U_{ij} + T_k \quad (2.1)$$

onde U_{ij} é a contribuição da unidade experimental na i -ésima linha e j -ésima coluna e T_k é a contribuição do tratamento k .

Reescrevendo (2.1) em termos dos efeitos médios das contribuições da unidade experimental na linha \bar{U}_i , coluna \bar{U}_j e linha-coluna $\bar{U}_{..}$, como também do efeito médio do tratamento \bar{T} , tem-se

$$\begin{aligned}
T_{ijk} &= \bar{U}_{..} + (\bar{U}_{i.} - \bar{U}_{..}) + (\bar{U}_{.j} - \bar{U}_{..}) \\
&\quad + (U_{ij} - \bar{U}_{i.} - \bar{U}_{.j} + \bar{U}_{..}) + \bar{T}_{.} + (T_k - \bar{T}_{.}) \\
&= \mu + \rho_i + \gamma_j + \tau_k + u_{ij}
\end{aligned} \tag{2.2}$$

com $\mu = \bar{U}_{..} + \bar{T}_{.}$ sendo a média da variável resposta; $\rho_i = \bar{U}_{i.} - \bar{U}_{..}$ definido como o efeito da i -ésima linha, com $\sum_i \rho_i = 0$; $\gamma_j = \bar{U}_{.j} - \bar{U}_{..}$ sendo o efeito da j -ésima coluna com $\sum_j \gamma_j = 0$; $\tau_k = T_k - \bar{T}_{.}$ sendo o efeito do k -ésimo tratamento com $\sum_k \tau_k = 0$ e

$$\begin{aligned}
u_{ij} &= U_{ij} - \bar{U}_{i.} - \bar{U}_{.j} + \bar{U}_{..} \\
&= (U_{ij} - \bar{U}_{..}) - (\bar{U}_{i.} - \bar{U}_{..}) - (\bar{U}_{.j} - \bar{U}_{..})
\end{aligned} \tag{2.3}$$

explicando a heterogeneidade das unidades experimentais, uma vez que se define u_{ij} como o efeito da interação linha-coluna, já que a contribuição das UE (ij) não é a adição do efeito linha (ρ_i) e o efeito coluna (γ_j).

Denote a resposta observada na i -ésima linha e j -ésima coluna por z_{ij} . Seja δ_{ij}^k uma variável aleatória que toma o valor de 1 se o tratamento k é aplicado à UE (ij), e 0 caso contrário. Então,

$$z_{ij} = \sum_k \delta_{ij}^k T_{ijk} \tag{2.4}$$

Substituindo (2.2) em (2.4) obtém-se

$$z_{ij} = \mu + \rho_i + \gamma_j + \sum_k \delta_{ij}^k \tau_k + u_{ij} \tag{2.5}$$

Por outro lado, pode-se expressar uma observação sobre uma UE em termos do tratamento recebido. Seja x_{kl} a observação da l -ésima aplicação do tratamento k e seja ζ_{ij}^{kl} uma variável aleatória com $\zeta_{ij}^{kl} = 1$ se a l -ésima aplicação do tratamento k é aplicada sobre a UE (ij) e $\zeta_{ij}^{kl} = 0$ caso contrário. Então,

$$x_{kl} = \sum_{ij} \zeta_{ij}^{kl} z_{ij} \tag{2.6}$$

O interesse principal está na média dos tratamentos e o contraste entre eles. Assim, de (2.6) tem-se

$$\begin{aligned}
\bar{x}_k &= \frac{1}{t} \sum_l \sum_{ij} \zeta_{ij}^{kl} z_{ij} \\
&= \frac{1}{t} \sum_{ij} \delta_{ij}^k z_{ij} \\
&= \mu + \tau_k + \frac{1}{t} \sum_{ij} \delta_{ij}^k u_{ij}
\end{aligned} \tag{2.7}$$

onde $\sum_l \zeta_{ij}^{kl} = \delta_{ij}^k$ e $\sum_{ij} \delta_{ij}^k = t$.

A distribuição conjunta das δ_{ij}^k é determinada pela família particular de quadrados latinos do qual um é escolhido aleatoriamente. Usando o procedimento de aleatorização tem-se que

$$P\left(\delta_{ij}^k = 1\right) = \frac{1}{t}$$

e

$$P\left(\delta_{ij}^k = 0\right) = 1 - \frac{1}{t}.$$

A estrutura do quadrado latino implica que se $\delta_{ij}^k = 1$ então $\delta_{i'j}^k = 0$ e $\delta_{ij}^{k'} = 0$. Logo

$$\begin{aligned}
P\left(\delta_{ij}^k = 1, \delta_{i'j}^k = 1\right) &= 0 & (i \neq i'), \\
P\left(\delta_{ij}^k = 1, \delta_{ij}^{k'} = 1\right) &= 0 & (j \neq j').
\end{aligned}$$

Também obtém-se,

$$P\left(\delta_{ij}^k = 1, \delta_{i'j'}^k = 1\right) = \frac{1}{t(t-1)} \quad (i \neq i', j \neq j').$$

De acordo com essas propriedades de δ_{ij}^k , pode-se pesquisar propriedades distribucionais dos modelos (2.5), (2.6) e (2.7). Assim,

$$E\left(\delta_{ij}^k\right) = \frac{1}{t}.$$

De (2.7) tem-se que $E(\bar{x}_{k.}) = \mu + \tau_k + \frac{1}{t^2} \sum_{ij} u_{ij}$. Usando (2.3) e o fato de que $\sum_i u_{ij} = \sum_j u_{ij} = 0$, então

$$E(\bar{x}_{k.}) = \mu + \tau_k \quad (2.8)$$

Suponha que haja interesse no contraste entre efeitos de tratamentos definido por

$$\theta = \sum_k c_k \tau_k \quad (2.9)$$

com $\sum_k c_k = 0$. Um estimador não viesado para a expressão 2.9 é dado pelo correspondente contraste entre as médias de tratamento,

$$\hat{\theta} = \sum_k c_k \bar{x}_{k.}$$

O valor esperado e variância de $\hat{\theta}$ são apresentados a seguir.

$$E\left(\sum_k c_k \bar{x}_{k.}\right) = \sum_k c_k \tau_k$$

e

$$\begin{aligned} \text{Var}(\bar{x}_{k.}) &= E[\bar{x}_{k.} - E(\bar{x}_{k.})]^2 \\ &= E\left[\frac{1}{t} \sum_{ij} \delta_{ij}^k u_{ij}\right]^2 \\ &= \frac{1}{t^2} E\left[\sum_{ij} (\delta_{ij}^k)^2 u_{ij}^2 + \sum_{i \neq i'} \sum_j \delta_{ij}^k \delta_{i'j}^k u_{ij} u_{i'j} \right. \\ &\quad \left. + \sum_i \sum_{j \neq j'} \delta_{ij}^k \delta_{ij'}^k u_{ij} u_{ij'} + \sum_{i \neq i'} \sum_{j \neq j'} \delta_{ij}^k \delta_{i'j'}^k u_{ij} u_{i'j'}\right] \\ &= \frac{1}{t^2} \left[\frac{1}{t} \sum_{ij} u_{ij}^2 + 0 + 0 + \frac{1}{t(t-1)} \sum_{i \neq i'} \sum_{j \neq j'} u_{ij} u_{i'j'} \right]. \quad (2.10) \end{aligned}$$

Tendo em conta que,

$$\begin{aligned}
\sum_{i \neq i'} \sum_{j \neq j'} u_{ij} u_{i'j'} &= \left(\sum_{ij} u_{ij} \right)^2 - \sum_{ij} u_{ij}^2 - \sum_{i \neq i'} \sum_j u_{ij} u_{i'j} - \sum_i \sum_{j \neq j'} u_{ij} u_{ij'} \\
&= 0 - \sum_{ij} u_{ij}^2 + \sum_{ij} u_{ij}^2 + \sum_{ij} u_{ij}^2
\end{aligned}$$

Substituindo esta expressão em (2.10) tem-se que

$$\begin{aligned}
Var(\bar{x}_{k.}) &= \frac{1}{t^2} \left(\frac{1}{t} + \frac{1}{t(t-1)} \right) \sum_{ij} u_{ij}^2 \\
&= \frac{1}{t^2(t-1)} \sum_{ij} u_{ij}^2
\end{aligned} \tag{2.11}$$

De forma semelhante, pode-se obter para $k' \neq k$

$$Cov(\bar{x}_{k.}, \bar{x}_{k'.}) = -\frac{1}{t^2(t-1)^2} \sum_{ij} u_{ij}^2 \tag{2.12}$$

Usando (2.11) e (2.12), segue que

$$\begin{aligned}
Var\left(\sum_k c_k \bar{x}_{k.}\right) &= \frac{1}{t^2} \left[\sum_k c_k^2 \frac{1}{t-1} \sum_{ij} u_{ij}^2 - \sum_{k \neq k'} c_k c_{k'} \frac{1}{(t-1)^2} \sum_{ij} u_{ij}^2 \right] \\
&= \sum_k c_k^2 \frac{1}{t(t-1)^2} \sum_{ij} u_{ij}^2
\end{aligned} \tag{2.13}$$

definindo $\sigma_u^2 = \frac{1}{(t-1)^2} \sum_{ij} u_{ij}^2$, pode-se reescrever (2.13) como

$$Var\left(\sum_k c_k \bar{x}_{k.}\right) = \sum_k c_k^2 \frac{\sigma_u^2}{t}$$

Uma estimação de σ_u^2 é obtida pelas médias da análise de variância.

Outro aspecto importante na especificação do modelo linear para um quadrado latino é o caso em que supõe-se o modelo sob aditividade no sentido amplo. A inclusão de erros técnicos é importante, o qual conduz ao caso onde

tem-se aditividade unidade-tratamento no sentido amplo. O importante a ter em conta é que, além do erro unitário u_{ij} , agora adiciona-se outros componentes, o erro experimental $v_{ij(k)}$ e o erro observacional $\eta_{ij(k)}$, ambos com média zero e variância σ_v^2 e σ_η^2 , respectivamente. A $E(QM)$ em termos destes componentes da variância são apresentados ao lado direito da coluna da Tabela 2.1.

Com o objetivo de fazer inferência sobre os tratamentos, define-se o erro experimental como $\epsilon_{ij(k)}$ e o erro observacional como $\eta_{ij(k)}$, os quais são considerados como variáveis aleatórias independentes e identicamente distribuídas (*i.i.d.*) com média zero e variância $\sigma_\epsilon^2 = \sigma_u^2 + \sigma_v^2$ e σ_η^2 , respectivamente. Mais adiante é definido o erro total como $e_{ij(k)}$, o qual é considerado uma variável *i.i.d.* com média zero e variância $\sigma_e^2 = \sigma_\epsilon^2 + \sigma_\eta^2$. Portanto, escrevendo o modelo para uma observação sobre uma UE(ij), ao qual se aplica o k -ésimo tratamento, tem-se

$$y_{ij(k)} = \mu + \rho_i + \gamma_j + \tau_k + e_{ij(k)} \quad (2.14)$$

De acordo com (2.14), o contraste de tratamentos $\sum_k c_k \tau_k$ é estimado como $\sum_k c_k \bar{y}_{..(k)}$, com

$$Var \left(\sum_k c_k \hat{\tau}_k \right) = Var \left(\sum_k c_k \bar{y}_{..(k)} \right) = \sum_k c_k^2 \frac{\sigma_e^2}{t}.$$

Da Tabela 2.1, o quadrado médio do erro experimental, $QM(E)$, é uma estimação para σ_e^2 e a hipótese $H_o : \tau_1 = \tau_2 = \dots = \tau_t$ pode ser testada considerando a estatística $F = \frac{QM(T)}{QM(E)}$, ou seja o cociente do quadrado médio dos tratamentos e o quadrado médio do erro experimental. A estatística se distribui de forma aproximada como uma F com $(t-1)$ e $(t-1)(t-2)$ graus de liberdade.

2.3 Análise de variância

Para escrever a tabela da ANOVA é conveniente denotar a observação sobre a UE(ij) que recebe o tratamento k como $y_{ij(k)}$, onde o símbolo $(.)$ indica que alguns termos (i, j, k) ocorrem no arranjo experimental, ocorrendo t^2 dos t^3 apenas. Portanto,

$$\begin{aligned}\bar{y}_{ij(.)} &= z_{ij} \\ \bar{y}_{i.(.)} &= \bar{z}_{i.}, & \bar{y}_{.j(.)} &= \bar{z}_{.j} \\ \bar{y}_{..(k)} &= x_k. \\ \bar{y}_{..(.)} &= z_{..} = \bar{x}_{..}\end{aligned}$$

A tabela da ANOVA, apresentada em [11] e dada na Tabela 2.1, é baseada na seguinte identidade para $y_{ij(k)}$

$$\begin{aligned}y_{ij(k)} &= \bar{y}_{..(.)} + (\bar{y}_{i.(.)} - \bar{y}_{..(.)}) + (\bar{y}_{.j(.)} - \bar{y}_{..(.)}) \\ &\quad + (\bar{y}_{..(k)} - \bar{y}_{..(.)}) + (\bar{y}_{ij(k)} - \bar{y}_{i.(.)} - \bar{y}_{.j(.)} - \bar{y}_{..(k)} + 2\bar{y}_{..(.)})\end{aligned}$$

e é obtida tomando quadrados em ambos lados e somando para todas as possíveis combinações $ij(k)$.

O valor esperado dos quadrados médios $E(QM)$ da Tabela 2.1, tem-se que o lado esquerdo da coluna é baseado nos modelos (2.5) (2.6) e obtido a partir da teoria de aleatorização. Os resultados da Tabela 2.1, mostram que o quadrado latino é um plano não-viesado, pois o $QM(E)$ fornece uma estimativa não-viesada de σ_u^2 . Para testar a hipótese

$$H_0 : \tau_1 = \tau_2 = \dots = \tau_t = 0$$

$$H_1 : \tau_i \neq \tau_j \quad \text{para qualquer } i \neq j$$

considera-se a estatística de teste $F = \frac{QM(T)}{QM(E)}$, ou seja o cociente do quadrado médio dos tratamentos e o quadrado médio do erro experimental, usada para testar a hipótese de igualdade de médias de tratamentos para mais de duas populações.

Tabela 2.1: ANOVA de um quadrado latino

Fontes	g.l.	SQ	QM	E(QM)	
				Ad. sentido estrito	Ad. sentido amplo
Linhas	$t - 1$	$t \sum_i (\bar{y}_{i.(\cdot)} - \bar{y}_{..(\cdot)})^2 = \text{SQ(F)}$	$\text{QM(F)} = \text{SQ(F)} / (t - 1)$	$t \sum_i \rho_i^2 / (t - 1)$	$\sigma_\eta^2 + \sigma_v^2 + t \sum_i \rho_i^2 / (t - 1)$
Colunas	$t - 1$	$t \sum_j (\bar{y}_{.j(\cdot)} - \bar{y}_{..(\cdot)})^2 = \text{SQ(C)}$	$\text{QM(C)} = \text{SQ(C)} / (t - 1)$	$t \sum_j \gamma_j^2 / (t - 1)$	$\sigma_\eta^2 + \sigma_v^2 + t \sum_j \gamma_j^2 / (t - 1)$
Tratamentos	$t - 1$	$t \sum_k (\bar{y}_{..(k)} - \bar{y}_{..(\cdot)})^2 = \text{SQ(T)}$	$\text{QM(T)} = \text{SQ(T)} / (t - 1)$	$\sigma_u^2 + t \sum_k \tau_k^2 / (t - 1)$	$\sigma_\eta^2 + \sigma_v^2 + \sigma_u^2 + t \sum_k \tau_k^2 / (t - 1)$
Erro	$(t - 1)(t - 2)$	Diferença = SQ(E)	$\text{QM(E)} = \text{SQ(E)} / (t - 1)(t - 2)$	σ_u^2	$\sigma_\eta^2 + \sigma_v^2 + \sigma_u^2$
Total	$t^2 - 1$	$\sum_{ij(k)} (\bar{y}_{ij(k)} - \bar{y}_{..(\cdot)})^2$			

A distribuição do teste F é aproximada pela função de distribuição de uma F com $t - 1$ graus de liberdade no numerador e $(t - 1)(t - 2)$ graus de liberdade no denominador.

Em geral, algumas vezes é discutido que o plano de quadrado latino não é muito usado sob um ponto de vista prático, pois tem duas limitações:

- i) o número de tratamentos deve ser igual ao número de linhas e colunas. Esta restrição impõe um inconveniente para o trabalho experimental real, especialmente para experimentos com um grande número de tratamentos;
- ii) para valores pequenos de t , o número de graus de liberdade do erro experimental é insuficiente para lograr níveis de precisão desejáveis.

2.4 Quadrados latinos replicados

Uma forma de incrementar os graus de liberdade do erro é replicar os quadrados latinos. O número de réplicas depende, geralmente, da situação experimental em questão. Usa-se como exemplo a pesquisa efetuada na tese [2] sobre diversas abordagens na extração e manutenção de linhas de produtos de software para mostrar métodos de replicação que conduzem a diferentes modelos lineares, assim como, diferentes análises, respectivamente.

Consideremos as seguintes situações:

- i) O experimento básico usando um quadrado latino de ordem t é replicado por uma linha de produtos de software r vezes. Nesse caso, tomam-se os mesmos produtos para as r réplicas e os mesmos t analistas são usados em cada réplica.
- ii) Diferentes produtos de software são obtidos para cada réplica, portanto, tem-se t produtos por réplica e mantém-se os mesmos t analistas.

- iii) Para este caso, cada uma das t linhas de produtos de software é dividida em r produtos de software, utilizando cada produto para uma das réplicas, onde procura-se r analistas diferentes, os quais são avaliados nas réplicas.
- iv) Tem-se que cada linha de produto de software é uma réplica. Assim, procura-se r linhas de produtos diferentes para levar em consideração o experimento básico. Cada linha de produtos de software tem seus próprios analistas e seus próprios produtos.

Em cada um destes experimentos realiza-se o procedimento de aleatorização de forma independente para cada componente do quadrado latino. Uma consequência da replicação de um quadrado latino é que uma variável de classificação adicional entra na análise, fazendo referência à variável réplica. Em essência, essa nova variável funciona como se fosse uma terceira variável de bloqueio, particionando do erro experimental alguma variação devido às diferenças gerais entre experimentos. Para isso, mostra-se como é efetuada a análise em cada método de réplica.

2.4.1 Linhas e colunas cruzadas dentro das réplicas.

Na situação (*i*) compartilham-se as mesmas linhas e colunas. Para este caso, tem-se que usar os mesmos analistas (linhas) e os mesmos produtos de software (colunas), assumindo que há diferenças sistemáticas entre cada réplica, sendo tais diferenças causadas por retardos de tempo. Assim, tal como se apresenta em [11], uma extensão do modelo (2.14) apropriado para observações deste plano experimental é:

$$Y_{ijk(l)} = \mu + \alpha_i + \rho_j + \gamma_k + \tau_l + e_{ijk(l)}, \quad (2.15)$$

onde ρ_j , γ_k , τ_l estão definidos ao começo da seção 2.2 e α_i , com $i = 1, 2, \dots, r$, representa o efeito da i -ésima réplica. A ANOVA correspondente é a Tabela (2.2).

Deve-se ressaltar que o modelo (2.15) é apenas um modelo possível. Para justificar esta configuração experimental, separa-se a partir da $SQ(\text{erro})$ a soma dos quadrados devido à réplica*interação entre tratamentos, com $(r - 1)(t - 1)$ graus de liberdade. Na verdade, os graus de liberdade do erro, tal como consta na Tabela (2.2), são a soma dos graus de liberdade do erro para r quadrados latinos individuais e os graus de liberdade para a réplica*linha, réplica*coluna e réplica*interação entre tratamentos.

Tabela 2.2: ANOVA para o modelo (2.15)

Fontes	g.l.	SQ	E(QM)
Réplicas	$r - 1$	$t^2 \sum_i (\bar{y}_{i..(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Produtos	$t - 1$	$rt \sum_j (\bar{y}_{.j.(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Analistas	$t - 1$	$rt \sum_k (\bar{y}_{..k(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Tratamentos	$t - 1$	$rt \sum_l (\bar{y}_{...l} - \bar{y}_{...(\cdot)})^2$	$\sigma_e^2 + rt \sum_l \tau_l^2 / (t - 1)$
Erro	$r(t - 1)(t - 2) + 3(r - 1)(t - 1)$	Diferença	σ_e^2
Total	$rt^2 - 1$	$\sum_{ijk(l)} (y_{ijk(l)} - \bar{y}_{...(\cdot)})^2$	

A seguinte sintaxe no software SAS representa o modelo (2.15) para esta forma de replicar quadrados latinos, pelo qual é fundamental o uso do procedimento GLM (*Generalized Linear Model*), no qual se determina as variáveis explicativas no comando *class*, e escreva-se o modelo em seguida do comando *model*.

```
proc glm data=modelo1 outstat=tabla1 noprint;
class rep col row treat;
model y = rep col row treat;
run;
```

Na figura 2.1 ilustra-se graficamente o processo de replicar linhas e colunas cruzadas dentro das réplicas, onde defina-se os diferentes analistas como A1,

A2 nas linhas e para as colunas, as diferentes linhas de produtos é denominada como LP1, LP2 e finalmente se faz referência as diversas réplicas como REP1, REP2, REP3.

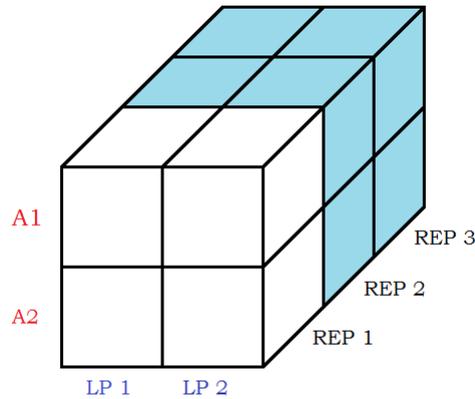


Figura 2.1: Mesmos analistas e mesmas linhas de produto

2.4.2 Colunas cruzadas com linhas embutidas nas réplicas.

De acordo com a situação (ii), compartilham-se as mesmas linhas, mas não as mesmas colunas. Aqui, usam-se os mesmos analistas (linhas), mas modificam-se as linhas de produtos (colunas). Portanto, as colunas estão embutidas dentro das réplicas derivando o modelo 2.16, assim como é citado em [11].

$$Y_{ijk(l)} = \mu + \alpha_i + \rho_{ij} + \gamma_k + \tau_l + e_{ijk(l)} \quad (2.16)$$

onde ρ_{ij} , com $i = 1, 2, \dots, r$ e $j = 1, 2, \dots, t$, é o efeito da linha embutida dentro da réplica. A ANOVA correspondente ao modelo (2.16) é dada na Tabela (2.3). O modelo (2.16) também pode ser alterado para incluir a réplica*interação entre tratamentos.

O código em SAS que representa o modelo (2.16) é apresentado a seguir e o gráfico que ilustra esta forma de replicação pode ser observado na figura

Tabela 2.3: ANOVA para o modelo (2.16)

Fontes	g.l.	SQ	E(QM)
Réplicas	$r - 1$	$t^2 \sum_i (\bar{y}_{i..(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Produtos/Réplicas	$r(t - 1)$	$t \sum_{ij} (\bar{y}_{ij.(\cdot)} - \bar{y}_{i..(\cdot)})^2$	
Analistas	$t - 1$	$rt \sum_k (\bar{y}_{..k(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Tratamentos	$t - 1$	$rt \sum_l (\bar{y}_{...l} - \bar{y}_{...(\cdot)})^2$	$\sigma_e^2 + rt \sum_l \tau_l^2 / (t - 1)$
Erro	$r(t - 1)(t - 2) + 2(r - 1)(t - 1)$	Diferença	σ_e^2
Total	$rt^2 - 1$	$\sum_{ijk(l)} (y_{ijk(l)} - \bar{y}_{...(\cdot)})^2$	

2.2.

```
proc glm data=modelo1 outstat=tabla1 noprint;
class rep col(rep) row treat;
model y = rep col row treat;
run;
```

Neste caso, diferentemente da forma de réplica anterior, o software SAS toma para um quadrado latino diferentes produtos de software, uma vez que as colunas estão embutidas nas réplicas (col(rep)).

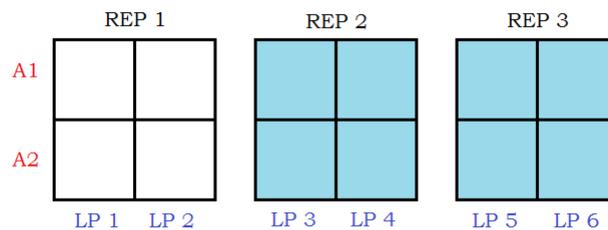


Figura 2.2: Mesmos analistas e diferentes linhas de produto

2.4.3 Linhas cruzadas com colunas embutidas nas réplicas.

De acordo com o exemplo (*iii*), compartilham-se as mesmas colunas, mas não as mesmas linhas. Nesta forma de replicação, tem-se os mesmos produtos de software em todas as réplicas, mas tem-se analistas diferentes em cada uma das réplicas, ver figura 2.3.

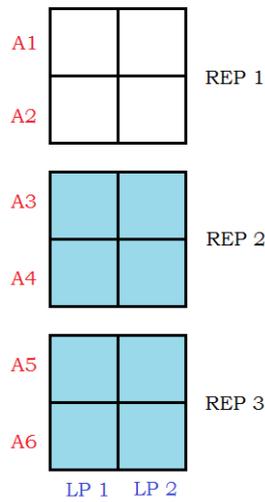


Figura 2.3: Mesmas linhas de produto e diferentes analistas

Tal como em [11], obtém-se um modelo da seguinte forma:

$$Y_{ijk(l)} = \mu + \alpha_i + \rho_j + \gamma_{ik} + \tau_l + e_{ijk(l)} \quad (2.17)$$

onde γ_{ik} , com $i = 1, 2, \dots, r$ e $k = 1, 2, \dots, t$; é o efeito da coluna embutida dentro da réplica. A ANOVA correspondente ao modelo (2.17) é dada na Tabela (2.4). O modelo (2.17) pode ser alterado para incluir a réplica*interação entre tratamentos.

Cabe ressaltar que este é o método de replicar quadrados latinos que foi utilizado para os seis experimentos feitos em [2] e a sintaxe SAS para representar o modelo (2.17) é a seguinte:

```
proc glm data=modelo1 outstat=tabla1 noprint;
```

Tabela 2.4: ANOVA para o modelo (2.17)

Fontes	g.l.	SQ	E(QM)
Réplicas	$r - 1$	$t^2 \sum_i (\bar{y}_{i..(.)} - \bar{y}_{...(.)})^2$	
Produtos	$t - 1$	$rt \sum_j (\bar{y}_{.j.(.)} - \bar{y}_{...(.)})^2$	
Analistas/Réplicas	$r(t - 1)$	$t \sum_{ik} (\bar{y}_{i.k(.)} - \bar{y}_{i..(.)})^2$	
Tratamentos	$t - 1$	$rt \sum_l (\bar{y}_{... (l)} - \bar{y}_{...(.)})^2$	$\sigma_e^2 + rt \sum_l \tau_l^2 / (t - 1)$
Erro	$r(t - 1)(t - 2) + 2(r - 1)(t - 1)$	Diferença	σ_e^2
Total	$rt^2 - 1$	$\sum_{ijk(l)} (y_{ijk(l)} - \bar{y}_{...(.)})^2$	

```
class rep col row(rep) treat;
model y = rep col row treat;
run;
```

2.4.4 Linhas e colunas embutidas dentro das réplicas.

De acordo com o exemplo (*iv*), os analistas (linhas) são diferentes em cada réplica, assim como os produtos de software (colunas). Este seria o método que geraria mais trabalho e custo na hora da realização. Os câmbios produzidos são ilustrados na figura 2.4.

Como em [11] um modelo adequado é:

$$Y_{ijk(l)} = \mu + \alpha_i + \rho_{ij} + \gamma_{ik} + \tau_l + e_{ijk(l)} \quad (2.18)$$

tendo a ANOVA dada na Tabela (2.5). Neste caso, pode ser útil e oportuno alterar o modelo (2.18) e incluir um termo de linha de produto de software*interação entre tratamentos, se o objetivo é investigar as diferenças entre os efeitos do tratamento e linhas de produtos específicos, devido a diferentes contextos empregados nas linhas de produtos. Em todos os casos, a hipótese $H_0 : \tau_1 = \tau_2 = \dots = \tau_t$ é testada na forma usual, empregando o teste $F = QM(T)/QM(E)$ com $(t - 1)$ e v graus de liberdade, onde $QM(erro)$ é

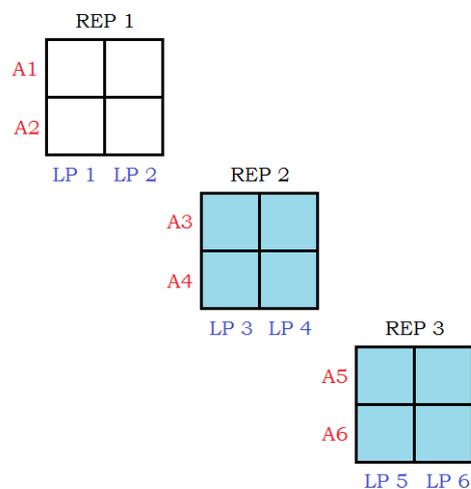


Figura 2.4: Diferentes analistas e diferentes linhas de produto

calculado para as quatro situações.

Tabela 2.5: ANOVA para o modelo (2.18)

Fontes	g.l.	SQ	E(QM)
Réplicas	$r - 1$	$t^2 \sum_i (\bar{y}_{i..(\cdot)} - \bar{y}_{...(\cdot)})^2$	
Produtos/Réplicas	$r(t - 1)$	$t \sum_{ij} (\bar{y}_{ij.(\cdot)} - \bar{y}_{i..(\cdot)})^2$	
Analistas/Réplicas	$r(t - 1)$	$t \sum_{ik} (\bar{y}_{i.k(\cdot)} - \bar{y}_{i..(\cdot)})^2$	
Tratamentos	$t - 1$	$rt \sum_l (\bar{y}_{... (l)} - \bar{y}_{...(\cdot)})^2$	$\sigma_e^2 + rt \sum_l \tau_l^2 / (t - 1)$
Erro	$r(t - 1)(t - 2) + (r - 1)(t - 1)$	Diferença	σ_e^2
Total	$rt^2 - 1$	$\sum_{ijk(l)} (y_{ijk(l)} - \bar{y}_{...(\cdot)})^2$	

As mudanças apresentadas no modelo (2.18) são refletidas no código SAS tais como se mostra a seguir:

```
proc glm data=modelo1 outstat=tabla1 noprint;
class rep col(rep) row(rep) treat;
model y = rep col row treat;
run;
```

CAPÍTULO 3

Engenharia de software

Uma área da engenharia que é considerada uma das mais recentes formalizadas e estudadas é a engenharia de software. De acordo com o foco desta dissertação, os fundamentos dos dois principais itens, serão discutidos ao longo desta seção. Inicialmente, o item referente à definição de engenharia de software e seus principais conceitos, em seguida, um segundo item respeito às linhas de produtos de software, fazendo uma introdução e uma abordagem sucinta desta área. Por último, devido ao grau de complexidade, algumas semelhanças e diferenças entre as abordagens de linhas de produto PLUSS e MSVCM são apresentadas.

A engenharia de software objetiva o desenvolvimento de sistemas de software com boa relação custo-benefício. Não existem limitações físicas para o potencial do software, mas a ausência de limitações naturais significa que o software pode se tornar complexo e muito difícil de ser compreendido.

Devido aos avanços da tecnologia de hardware, particularmente ao aumento da capacidade de processamento e armazenamento, aplicações até recentemente inimagináveis passaram a ser demandadas, exigindo um software maior e mais complexo. A experiência inicial de construção desses sistemas

mostrou que uma abordagem informal de desenvolvimento de software não era suficiente. Projetos importantes sofriam atrasos por apresentarem custos maiores do que os inicialmente previstos, portanto, eram não confiáveis, de difícil manutenção e tinham desempenho inferior.

Os custos de hardware caíam, enquanto os de software subiam rapidamente. Novas técnicas e novos métodos eram necessários para controlar a complexidade inerente aos grandes sistemas de software [14].

Graças a esse aumento constante de demanda, foi possível observar um contínuo progresso desde 1968 e o desenvolvimento da engenharia de software melhorou a qualidade do software que se produz. Tem-se uma compreensão muito melhor das atividades envolvidas no desenvolvimento do software e se procuraram métodos eficazes de especificação, projeto e implementação de software. Novas ferramentas reduzem o esforço necessário para produzir sistemas grandes e complexos.

3.1 Definição

Antes de definir engenharia de software é ideal conhecer primeiro o significado do que é software. De acordo com Somerville [14], software não é apenas o programa de computador, é também toda a documentação associada e os dados de configuração necessários para fazer com que esses programas operem corretamente. Um sistema de software consiste em uma série de programas separados com arquivos de configuração dos programas, documentação do sistema que descreve a estrutura do mesmo e documentação do usuário que explica como utilizar o sistema.

Portanto, tem-se que a engenharia de software é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema depois que ele entrou em operação. No entanto, a engenharia de software não se dedica apenas aos processos técnicos de desenvolvimento de software, mas também à atividades como o gerenciamento de projetos de software e ao desenvol-

vimento de ferramentas, métodos e teorias que dêem apoio à produção de software.

Em geral, os engenheiros de software adotam uma abordagem sistemática e organizada no trabalho, uma vez que essa é a maneira mais eficaz de produzir software de alta qualidade. A engenharia de software está relacionada com a seleção do método mais apropriado para um conjunto de circunstâncias e uma abordagem mais criativa e informal para que o desenvolvimento possa ser eficaz nessas circunstâncias.

Para isto, determina-se um processo de software, o qual é um conjunto de atividades e resultados associados que objetivam o desenvolvimento de um produto de software. Há quatro atividades de processo que são fundamentais e comuns a todos os processos de software. Essas atividades são:

1. Especificação do software: a funcionalidade do software e as restrições em sua operação devem ser definidas.
2. Desenvolvimento do software: o software deve ser produzido de modo que atenda a suas especificações.
3. Validação do software: o software tem de ser validado para garantir que ele faça o que o cliente deseja.
4. Evolução do software: o software deve evoluir para atender as necessidades mutáveis do cliente.

Diferentes processos de software organizam essas atividades de maneiras diversas e são descritos em diferentes níveis de detalhes. Os prazos das atividades variam, assim como os resultados de cada atividade. No entanto, alguns processos são mais adequados do que outros para alguns tipos de aplicações. Se um processo inadequado for utilizado, provavelmente a qualidade ou a utilidade do produto de software a ser desenvolvido será reduzida.

Apresentados de forma breve alguns conceitos básicos e introdutórios da engenharia de software, em seguida apresenta-se uma abordagem mais es-

pecializada e avançada dentro da engenharia de software. Este é o caso da abordagem de engenharia de linhas de produtos de software.

A próxima seção apresenta os conceitos introdutórios de linhas de produtos de software, enfatizando a sua importância na lógica da aplicação em processos de engenharia de software.

3.2 Linhas de produtos de software

Diversos motivos levam às empresas a embarcar em uma abordagem de engenharia como a linha de produto de software. Estes vão desde aspectos como o custo e o tempo, assim como aspectos de confiabilidade para o usuário final e a consistência da interface do usuário.

De acordo com a definição citada em [2], tem-se que uma linha de produto de software é um conjunto de sistemas intensivos de software que compartilham um conjunto de características que satisfaçam as necessidades específicas de um determinado segmento de mercado ou missão, e que são desenvolvidos a partir de um conjunto comum de artefatos de software essenciais de uma forma prescrita.

A abordagem de linha de produtos de software geralmente é baseada fortemente em considerações de ordem econômica. Devido ao apoio de reuso em grande escala, tal abordagem melhora o processo especialmente no desenvolvimento de software, ou seja, reduz custos e tempo de lançamento do produto e melhora a qualidade dos produtos resultantes, como a sua fiabilidade.

A redução de custos e tempo de lançamento é fortemente correlacionada na engenharia em linha de produtos de software: a abordagem apóia o reuso em grande escala durante o desenvolvimento de software. Ao contrário de abordagens tradicionais, o reuso pode chegar a ser 90% do software pelo geral, [15].

O reuso é mais rentável do que o desenvolvimento sem reuso, por ordens de magnitude. Assim, os custos de desenvolvimento e tempo de lançamento podem ser drasticamente reduzidos pela engenharia de linha de produtos.

Infelizmente, essa melhoria não vem de graça, mas exige algum investimento inicial extra. Isso é necessário para a construção de artefatos reutilizáveis, transformando a organização, entre outras características. Diferentes estratégias existem para fazer este investimento, mas a necessidade básica para um conjunto de investimentos se mantém. A mensagem positiva é, no entanto, que geralmente, um ponto de equilíbrio é alcançado após três produtos (Figura 3.1) [15].

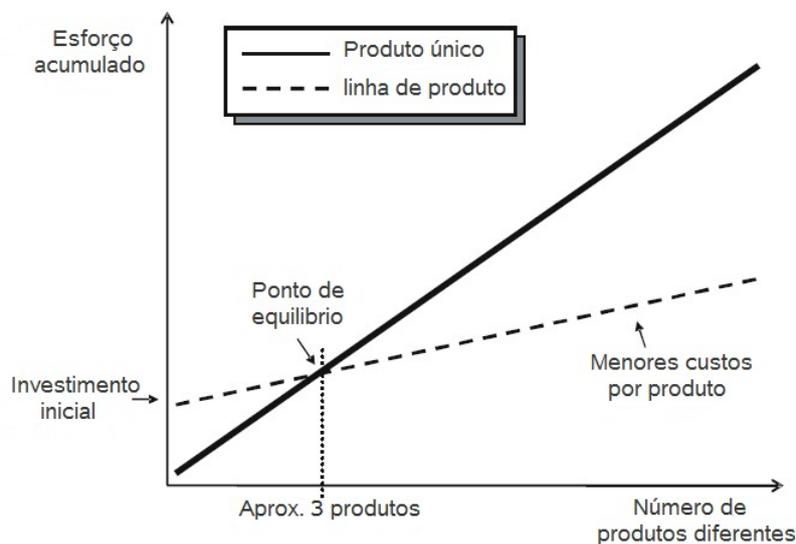


Figura 3.1: Economia em linha de produtos de software. [15]

Além disso, dependendo da abordagem para o desenvolvimento de uma linha de produtos de software, uma organização pode atingir um ponto de equilíbrio anterior, quando uma linha de produtos de software é lançada. Por exemplo, a abordagem proativa (também conhecido como big-bang, veja a Figura 3.2(a)) lança uma linha de produtos de software a partir do zero. Neste caso, todos os artefatos da linha de produtos são construídos a partir da análise de domínio. Diferentemente, a abordagem extrativista lança uma linha de produtos de software a partir de produtos existentes. Esta abordagem ilustrada na Figura 3.2 (b), pode reduzir o investimento inicial para institucionalizar o desenvolvimento de uma linha de produtos, uma vez que

os artefatos existentes estão disponíveis e podem ser reutilizados [2].

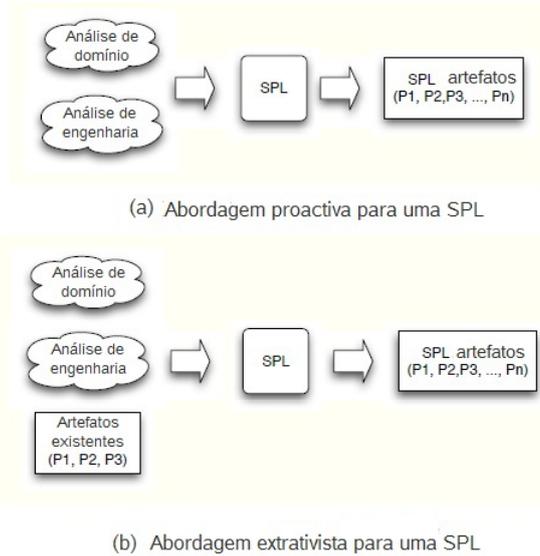


Figura 3.2: Diferentes abordagem para o desenvolvimento de uma linha de produtos de software

Normalmente, junto com a redução dos custos de desenvolvimento, uma redução dos custos de manutenção também é atingida. Vários aspectos contribuem para essa redução, o mais notório é o fato de que o montante global de código e documentação que deve ser mantida é drasticamente reduzido. Como o tamanho global dos projetos de desenvolvimento de aplicativos é fortemente reduzido, o risco do projeto de acompanhamento também reduz-se.

A engenharia de linha de produtos de software também tem um forte impacto sobre a qualidade do software resultante. Um novo aplicativo consiste, em grande medida, de componentes reusados e previamente testados. Isto implica que a densidade de defeitos de tais produtos pode vir a ser drasticamente mais baixa do que os produtos que são desenvolvidos tudo de novo. Isto leva a sistemas mais confiáveis e seguros, como resultado, a segurança é impactada positivamente também. Engenharia de linha de produtos de software pode apoiar a garantia da qualidade, por exemplo, quanto a um produto e sua simulação em duas variantes. Especialmente para sistemas incorporados, a simulação permite testes extensos que seriam impossíveis em

outros casos. Se ambas variantes são derivadas do mesmo código, as simulações podem ser utilizadas como base para a análise da qualidade do produto final. Embora os argumentos dos custos normalmente dominem o debate na engenharia de linha de produto, a capacidade de produzir maior qualidade é, para algumas organizações, o grande argumento, especialmente em domínios de segurança crítica, onde se faz um grande esforço na garantia da qualidade e dos esforços de certificação. Além da qualidade do processo, a engenharia de linha de produtos de software impacta positivamente os aspectos do produto, como a usabilidade do produto final, melhorando a consistência da interface do usuário.

3.2.1 Fundamentos da abordagem de engenharia de linha de produto de software

De acordo com [15], vários princípios são fundamentais para o sucesso da engenharia de linha de produto de software, podendo ser descritos como segue:

- **Gestão de variabilidade:** sistemas individuais são considerados como variações de um tema comum. Esta variabilidade é explícita e deve ser gerida de forma sistemática.
- **Centrar o negócio:** a engenharia de linha de produtos de software visa conectar completamente a engenharia de linha de produtos com a estratégia de longo prazo do negócio.
- **Centrar arquitetura:** o lado técnico do software deve ser desenvolvido de uma forma que permite tirar partido das semelhanças entre os sistemas individuais.
- **Abordagem dos dois ciclos de vida:** os sistemas individuais são desenvolvidos com base em uma plataforma de software. Estes produtos - bem como a plataforma - devem ser projetados e ter seus próprios ciclos de vida.

Procurando levar estas noções teóricas ao entendimento do processo de experimentação efetuado em [2], é importante mencionar que o princípio fundamental abordado no estudo preliminar das abordagens PLUSS e MSVCM foi a gestão de variabilidade.

Basicamente, neste ponto é onde se pretende fazer a diferença na hora de aperfeiçoar o desenvolvimento e a manutenção de uma linha de produto de software.

3.2.2 PLUSS e MSVCM

Uma linha de produto de software é fortemente caracterizada por um segmento de mercado ou domínio, no qual os conceitos relevantes são usualmente documentados como features¹, ou seja, aspectos ou características do domínio² visíveis para o usuário. O objetivo da modelagem de features, que é uma técnica amplamente usada nas linhas de produto de software, detalha as relações entre os conceitos de um domínio. Estas relações definem o escopo, o espaço de variabilidade e os membros do conjunto de uma linha de produtos de software. Como consequência, as configurações específicas de features são usadas como contribuição dos processos de derivação de produtos.

Estas duas técnicas são avaliadas em [2], onde o interesse principal é verificar se existem diferenças nos tempos médios de extração e manutenção de diferentes linhas de produtos de software. De acordo com [3], procura-se avaliar o uso das duas técnicas para representar a variabilidade em cenários de caso de uso. Estas foram selecionadas porque são bastante similares, de acordo com as especificações de entrada, e representam muito bem os estilos composicionais (MSVCM) e invasivos (PLUS) para gestão da variabilidade.

Para apoiar a atividade de derivação de produtos, representam-se primeiro os pontos de variação, através dos diferentes artefatos das linhas de produto de software, e então é descrito como resolver, dada a configuração de features, estes pontos de variação. Realmente, estas tarefas são relaci-

¹representa graficamente o domínio da linha de produtos

²areá de atuação do software dentro de uma areá de conhecimento.

onadas à gestão da variabilidade, a qual é inerentemente transversal, dado que a contribuição de uma feature é usualmente espalhada através de muitos artefatos de uma linha de produto de software. Dependendo de vários fatores (tais como técnicas de modularização, features granuladas e assim sucessivamente), o efeito de um feature relativo aos artefatos de uma linha de produtos de software pode ser de anotação ou composicional. O estilo de anotação emaranha a representação de especificações comuns e variantes, enquanto que o estilo composicional suporta melhor a separação de comportamentos variantes em distintos módulos.

Para entender um pouco as diferenças e semelhanças das duas abordagens PLUSS e MSVCM, em [2] apresenta-se linhas de produtos de software de comércio eletrônico (*e-commerce*), para este caso, usa-se os cenários compra de produtos *Buy Product* e procura de produtos *Search Product*.

Considere uma linha de produto para o domínio *e-commerce*, a qual pode gerar produtos que suportam compras por carro *shopping carts*, ou compras por bônus *bonus* e métodos de remessa diferentes *shipping methods*. Adicionalmente, configurações econômicas podem ser também avaliadas, no entanto, estas configurações não podem suportar compras por carro e compras por bônus.

Cenários de produtos pertencentes a esta linha de produtos de software são apresentados na figura 3.3. O cenário compra de produtos do primeiro produto (no lado esquerdo da figura) está configurado com compras por carro e por bônus, como se observa nos dois primeiros passos do cenário compra de produtos. Adicionalmente, este produto suporta dois métodos de remessa: econômico *economical* e rápido *fast*, como é indicado no quarto passo. Finalmente, o primeiro produto não registra usuários preferenciais depois de uma procura ou compra.

Diferentemente, o segundo produto (lado direito da figura 3.3) não está configurado com shopping cart e bonus, como se observa nos dois primeiros passos do cenário Buy Product. Além disso, suporta unicamente o método de remessa Economical, onde este registra as preferências dos usuários depois

SC01: Compra de produtos	
Ação de usuário	Artefato resposta
Seleciona a opção de saída.	Apresenta os itens no carro de compra e a quantia a ser paga. O usuário poder remover itens de seu carro de compra.
Selecione o confirme opção.	Pedido de bônus e informação de pagamento.
Encher pedido de informação e seleciona a opção de procedimento.	Pedido do método de remessa e endereço.
Seleciona um dos métodos de remessa (econômico, rápido), enche o endereço de destino e seleciona a opção de procedimento.	Calcula custos da remessa.
Confirmação da compra.	Executa a ordem e envia o pedido ao sistema de entrega, em ordem a despachar o produto.
	Registra usuários preferencias.

SC02: Procura de produtos	
Ação de usuário	Artefato resposta
Seleciona a opção de procura de produto.	Apresenta a avaliação de critérios de procura.
Informe do critério de procura.	Recobre os produtos que satisfazem o critério de procura. Mostra lista com produtos resultantes.
	Registra usuários preferencias.

SC01: Compra de produtos	
Ação de usuário	Artefato resposta
Seleciona a opção de compra de produtos.	Apresenta o produto selecionado. O usuário pode mudar a quantia de artigos que quer comprar.
Selecione o confirme opção.	Pedido informação de pagamento.
Encher pedido de informação e seleciona a opção de procedimento.	Pedido do método de remessa e endereço.
Seleciona um dos métodos de remessa (econômico, rápido), enche o endereço de destino e seleciona a opção de procedimento.	Calcula custos da remessa.
Confirmação da compra.	Executa a ordem e envia o pedido ao sistema de entrega, em ordem a despachar o produto.
	Registra usuários preferencias.

SC02: Procura de produtos	
Ação de usuário	Artefato resposta
Seleciona a opção de procura de produto.	Apresenta a avaliação de critérios de procura.
Informe do critério de procura.	Recobre os produtos que satisfazem o critério de procura. Mostra lista com produtos resultantes.
	Registra usuários preferencias.

Figura 3.3: Diferentes configurações dos cenários compra e procura de produtos. [2]

de uma operação de procura e compra, tal como é indicado no passo final de ambos cenários no segundo produto.

De acordo com uma abordagem de linha de produto de software, deveria ser possível gerar as duas configurações de um conjunto de artefatos reusáveis. Por tal razão, na engenharia de software foi proposta uma abordagem para representar os pontos de variação em cenários, tal como linhas de produtos modelando casos de uso para engenharia de sistemas e de software (Product Line Use Case Modeling for Systems and Software Engineering-PLUS). Porém, apesar dos benefícios de representação de variabilidade, esta abordagem não apresenta uma separação clara entre gestão de variabilidade e especificação de cenários.

Na figura 3.4 são descritos os cenários de Buy Product escritos na notação PLUS. Note que um único artefato é usado para representar todas as configurações válidas relacionadas a este cenário, misturando comportamentos comuns, comportamentos variáveis e configuração de informação referente à

feature que está dentro de colchetes. Por exemplo, os passos 1(a) e 1(b) nunca são executados juntos, ou seja, são passos alternados: o passo 1(a) está presente unicamente se o feature Shopping Cart é selecionada, caso contrario está presente o passo 1(b). Do mesmo jeito, tem-se que escolher entre a opção (a) e (b) para o passo 2, dependendo se o feature Bonus é selecionado ou não. Finalmente, o passo 6 é opcional e estará presente unicamente se o feature Update User Preference é selecionado.

Id	Ação de usuário	Artefato resposta
1(a)	Seleciona a opção de saída. [Carro de compra]	Apresenta os itens no carro de compra e a quantia a ser paga. O usuário poder remover itens de seu carro de compra.
1(b)	Seleciona a opção de compra de produtos. [não Carro de compra]	Apresenta o produto selecionado. O usuário pode mudar a quantia de artigos que quer comprar. Calcula e mostra a quantia a ser paga.
2(a)	Selecione o confirme opção. [Bônus]	Pedido bônus e informação de pagamento.
2(b)	Selecione o confirme opção. [não Bônus]	Pedido informação de pagamento.
3	Encher pedido de informação e seleciona a opção de procedimento.	Pedido do método de remessa e endereço.
4	Seleciona um método de remessa, enche o endereço de destino e seleciona a opção de procedimento.	Calcula custos da remessa.
5	Confirmação da compra.	Executa a ordem e envia o pedido ao sistema de entrega, em ordem a despachar o produto.
(6)	[Registra usuários preferenciais]	Registra usuários preferenciais.

Figura 3.4: Cenários Buy Product usando PLUSS. [2]

Como consequência, todas as possíveis variantes são descritas dentro do mesmo artefato. Isto dificulta entender o comportamento dos produtos especificados, usando a abordagem PLUSS. Adicionalmente, resultam envolvimento entre comportamentos comuns e variantes aumentando o processo de manutenção, uma vez que ao introduzir uma nova variante do produto mudanças são requeridas em outros pontos de cenários existentes.

Em conclusão, PLUSS não apresenta uma clara separação entre a variabilidade dos artefatos e a especificação de cenários, comprometendo a evolução de uma linha de produtos de software.

Por outro lado, tem-se a abordagem de modelagem de cenários de variabilidade por mecanismos transversais (Modeling Scenario Variability as

Crosscutting Mechanisms - MSVCM) para representar a variabilidade em cenários de casos de uso. Procura-se aumentar a separação de interesses entre comportamentos comuns e variantes, assim como incrementar a separação de interesse entre o cenário de especificações e a gestão de variabilidade.

Para lidar com variabilidades textuais nos cenários, constrói-se casos de uso aspectual e parâmetros. Os casos de uso aspectual tratam com a variabilidade em fluxo de controle, como é requerido pelos feature Shopping Cart e Bonus no cenário Buy Product, comportamento que é representado com cinza no primeiro passo da primeira configuração da figura 3.3. A parametrização de cenários, por outro lado, trata com a variabilidade nos dados, como é requerido pelo feature Shipping Method.

Assim, pode-se representar as duas configurações da figura 3.3 usando:

- i) Um cenário parametrizado SC01, o qual define os comportamentos comuns para a compra de produtos.
- ii) Um cenário base SC02 para comportamentos comuns de procura de produtos.
- iii) Casos de uso aspectual (*Advice-ADV*), que especifica os comportamentos opcionais requeridos pelas configurações Shopping Cart e Bonus, não (Shopping Cart e Bonus) e Update User Preferences (ADV01, ADV02 e ADV03 respectivamente).

Finalmente, as configurações ficam representadas na figura 3.5.

Os parâmetros são representados por nomes incluídos dentro de parênteses de ângulo (“<”e “>”). Há uma referência a parâmetros no cenário etiquetado como SC01 da figura 3.5. No segundo passo, SM faz referência a um parâmetro de um ponto de variação para as opções do Shipping Method. Adicionalmente, os casos de uso aspectuais diferem dos cenários porque têm uma cláusula de ponto de corte (antes, depois ou ao redor). Esta cláusula refere-se às anotações especiais aos passos de cenários (ou ainda, passos de advice). A avaliação de um advice combina os passos antes (depois ou ao

SC01: Compra de produtos	
Ação de usuário	Artefato resposta
Encher pedido de informação e seleciona a opção de procedimento.	Pedido do método de remessa e endereço. @ConfirmaçãoCompra
Seleciona um dos métodos de remessa <SM>, enche o endereço de destino e seleciona a opção de procedimento.	Calcula custos da remessa.
Confirmação da compra.	Executa a ordem e envia o pedido ao sistema de entrega, em ordem a despachar o produto. @RegistraPreferenciais

SC02: Procura de produtos	
Ação de usuário	Artefato resposta
Selecione a opção de procura de produto.	Apresenta a avaliação de critérios de procura.
Informe do critério de procura.	Recobre os produtos que satisfazem o critério de procura. Mostra lista com produtos resultantes. @RegistraPreferenciais

ADV01: Compra de produtos com carro de compras e bônus	
Ponto de corte: antes @ConfirmaçãoCompra	
Ação de usuário	Artefato resposta
Seleciona a opção de saída.	Apresenta os itens no carro de compra e a quantia a ser paga. O usuário poder remover itens de seu carro de compra.
Selecione o confirme opção.	Pedido de bônus e informação de pagamento.

ADV02: Compra de produtos sem carro de compras e bônus	
Ponto de corte: antes @ConfirmaçãoCompra	
Ação de usuário	Artefato resposta
Seleciona a opção de compra de produtos.	Apresenta o produto selecionado. O usuário pode mudar a quantia de artigos que quer comprar.
Selecione o confirme opção.	Pedido informação de pagamento.

ADV03: Registra usuários preferenciais	
Ponto de corte: depois @RegistraPreferenciais	
Ação de usuário	Artefato resposta
	Registra usuários preferenciais.

Figura 3.5: Cenários MSVCM de Buy Product e Search Product. [2]

redor) de algum passo que tenha uma cláusula de ponto de corte. Por exemplo, o primeiro passo do cenário SC01 tem anotação @ConfirmPurchase. Consequentemente, avaliar o advice ADV01 resulta em um fluxo de eventos similares aos cenários descritos no lado esquerdo da figura 3.3.

Para representar o conhecimento das configurações, note que os cenários (ou advices) não fazem referências explícitas aos features. O conhecimento das configurações é responsável por relacionar os artefatos da linha de produtos com os features. A estrutura do conhecimento das configurações relaciona expressões do feature a transformações: traduzir artefatos de linhas de produtos em artefatos específicos dentro dos produtos. Se a expressão de um feature é avaliada como verdadeira para uma determinada configuração de feature, as transformações relacionadas são aplicadas.

Três transformações distintas lidam com os tipos de variabilidade abordados: selecionar cenários (lida com a variabilidade em função), avaliar advices (lida com a variabilidade no controle do fluxo) e parâmetros de união (lida com a variabilidade dos dados). Por exemplo, o conhecimento da configuração da Figura 3.6 abrange a configurabilidade dos cenários Buy Products e Search Products.

Expressão artefatos	Transformações
Serviço compra de bens	seleciona cenário SC01
Serviço procura de produtos	seleciona cenário SC02
Carro de compras e bônus	avaliar requerimentos ADV01
não (Carro de compras e bônus)	avaliar requerimentos ADV02
Registra usuários preferenciais	avaliar requerimentos ADV03
Métodos de remessa	aponta parâmetros SM ao método de remessa
...	...

Figura 3.6: Exemplo do conhecimento de configurações. [2]

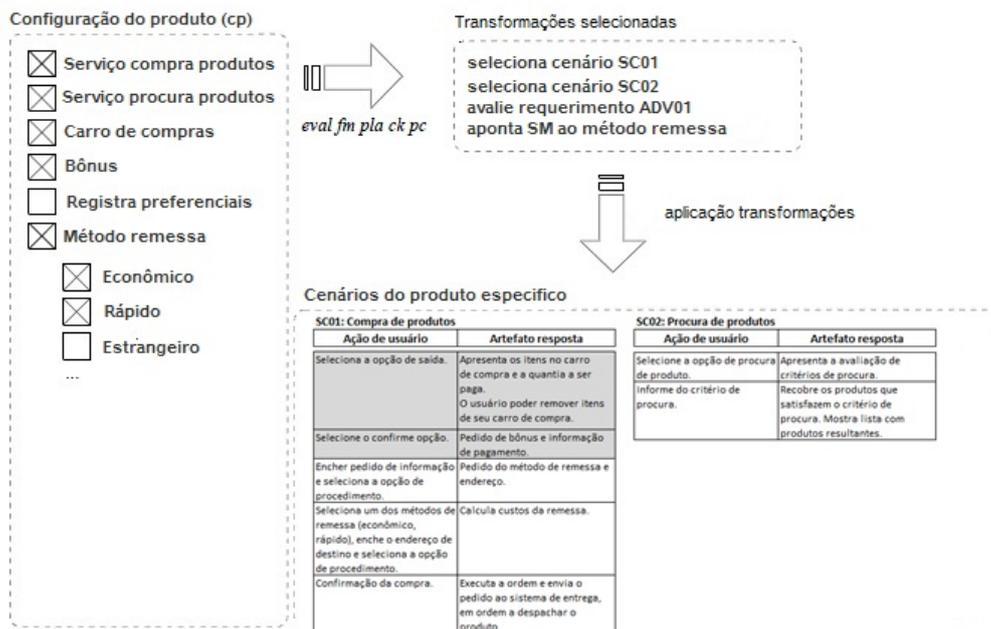


Figura 3.7: Construção do conhecimento de configurações. [2]

A Figura 3.7 mostra o fluxo de trabalho para construção de artefatos de produtos específicos, usando a anterior representação do conhecimento das configurações. O processo começa quando uma seleção válida de features fornece um produto (a configuração do produto). Em seguida, no processo de construção, que também considera o modelo de feature, os artefatos de uma linha de produtos de software e o conhecimento das configurações recupera a lista de transformações que devem ser aplicadas para o produto específico. A última etapa consiste principalmente em aplicar a lista de transformações.

Quadrados latinos 2x2 e aplicações a engenharia de software

4.1 Processo de aleatorização para um plano experimental

Tem-se um plano aleatorizado se, e só se, há uma alocação aleatória dos tratamentos às UE. Explica-se com apoio do SAS o processo de aleatorização, especialmente no procedimento PROC PLAN. De acordo com [13] o procedimento PLAN constrói modelos em planos aleatorizados para experimentos fatoriais, sobretudo para experimentos embutidos e cruzados e planos aleatorizados em blocos completos. PROC PLAN também é usado para gerar listas de permutações e combinações de números. PROC PLAN gera um plano fazendo primeiro uma seleção dos níveis para o primeiro fator, enquanto que para o segundo, o procedimento gera uma seleção dos níveis para cada nível do primeiro fator.

Em geral, para um fator dado, o procedimento PLAN origina uma seleção

dos níveis para todas as combinações dos níveis de todos os fatores que o precedem. A seleção pode ser feita através de cinco possibilidades diferentes:

- Seleção aleatória: os níveis são volvidos em ordem aleatória. Seleciona-se m níveis do fator de maneira aleatória sem substituição dos inteiros $1, \dots, n$, com $n \in \mathbb{N}$. Se n não é especificado então seleciona-se níveis aleatoriamente ordenando os inteiros $1, \dots, m$, com $m \in \mathbb{N}$.
- Seleção ordenada: os níveis são tornados em ordem padrão cada vez que a seleção seja gerada. Seleciona-se os níveis do fator como inteiros $1, \dots, m$ nessa ordem.
- Seleção cíclica: os níveis retornados são calculados por permutações cíclicas dos níveis das seleções prévias. Seleciona-se os níveis do fator permutando ciclicamente os inteiros $1, \dots, n$.

Exemplo: $t = 5$ *cyclic*; seleciona-se os inteiros de $1, \dots, 5$; na próxima iteração $2, 3, 4, 5, 1$; na seguinte $3, 4, 5, 1, 2$ e assim sucessivamente.

- Seleção permutada: os níveis são uma permutação dos inteiros de 1 até n . Seleciona-se m níveis do fator como uma permutação dos inteiros $1, \dots, m$ de acordo com um algoritmo cíclico que gera todas as $m!$ permutações. As permutações são produzidas ordenadamente.

Exemplo: $t = 5$ *perm*; listam os inteiros de 1 até 5 na primeira iteração, na segunda iteração a lista $1, 2, 3, 5, 4$ e para a iteração 119 produz a lista $5, 4, 3, 1, 2$ e na última iteração (120) se tem $5, 4, 3, 2, 1$.

- Seleção de combinações: m níveis são selecionados como uma combinação dos inteiros de 1 até n com $n \in \mathbb{N}$, tomando m cada vez. De acordo com um algoritmo cíclico gera-se todas as $n!/(m!(n-m)!)$ combinações. As combinações são produzidas ordenadamente.

Exemplo: $t = 3$ *of 5 comb*; mostra todas as combinações de 5 inteiros escolhendo 3 por vez. A primeira iteração é $1, 2, 3$, a segunda é $1, 2, 4$, a terceira $1, 2, 5$ e assim sucessivamente, onde a última iteração é $3, 4, 5$.

Com a finalidade de introduzir conceitos fundamentais que são a base e o ponto de partida no estudo de planos experimentais mais complexos, inicialmente é abordado um plano experimental básico como um exemplo simples, o que ajudará no entendimento das análises e estudos dos mesmos.

Suponha a existência de $N = tr$ unidades experimentais (UE) homogêneas e t tratamentos. Separa-se aleatoriamente as tr unidades experimentais em t conjuntos, cada um com r UE. Os t tratamentos são alocados aos t conjuntos tal que o i -ésimo tratamento é aplicado a cada uma das r UE no i -ésimo conjunto ($i = 1, 2, \dots, t$). A realização deste protocolo é chamada de experimento completamente aleatorizado com igual número de réplicas.

Assim, nesta seção, para simulação de um experimento completamente aleatorizado usa-se basicamente o seguinte programa em SAS para gerar os possíveis planos. Para o exemplo seguinte, tem-se dois tratamentos com cinco réplicas.

```
proc plan seed=1234;
    factors unit=10 /noprint;
    treatments treat=10 cyclic (1 1 1 1 1 2 2 2 2 2);
    output out=OutFromProcPlan;
run;
```

A base do programa é o uso do procedimento *plan*, onde se especifica uma semente para gerar o plano aleatório ($seed=1234$), posteriormente se define o número de unidades experimentais tal como se observa no comando $unit=10$ e conseqüentemente o número de tratamentos com o respectivo número de réplicas ($treat=10$ *cyclic* (1 1 1 1 1 2 2 2 2 2)). Finalmente o programa gera uma lista aleatória das dez unidades experimentais associada ao fator *treat* que vai ser uma variável com uma lista ordenada (1 1 1 1 1 2 2 2 2), cuja saída é apresentada a seguir:

unit	7	1	3	2	8	6	9	10	4	5
treat	1	1	1	1	1	2	2	2	2	2

A partir dessa forma como o experimento foi aleatorizado, deriva-se a seguinte variável aleatória:

$$\delta_{ij}^k = \begin{cases} 1 & \text{se a UE } k \text{ é associada com a etiqueta } ij \\ 0 & \text{caso contrário} \end{cases}$$

cujas propriedades são descritas nas seguintes probabilidades determinadas pelo processo da aleatorização:

$$\begin{aligned} P(\delta_{ij}^k = 1) &= 1/N \\ P(\delta_{ij}^k = 1, \delta_{i'j'}^{k'} = 1) &= 1/N(N-1), \quad k \neq k', \quad ij \neq i'j' \\ P(\delta_{ij}^k = 1, \delta_{i'j'}^{k'} = 1, \delta_{i''j''}^{k''} = 1) &= 1/N(N-1)(N-2), \\ & \quad k, k', k'' \text{ diferentes, } \quad ij, i'j', i''j'' \text{ diferentes} \end{aligned}$$

As variáveis $\delta_{ij}^k (k = 1, 2, \dots, N = rt; i = 1, 2, \dots, t; j = 1, 2, \dots, r)$ são nomeadas de variáveis aleatórias do modelo. Têm-se $(rt)^2$ destas variáveis, as quais são dependentes entre si. Por exemplo, se a UE 2 recebe a etiqueta 12, ou seja, o tratamento 1 com a aplicação 2 do tratamento 1, então $\delta_{11}^2 = \delta_{21}^2 = \delta_{22}^2 = 0$ e também $\delta_{12}^1 = \delta_{12}^3 = \delta_{12}^4 = 0$, uma vez que a UE 2 não pode receber alguma outra etiqueta e a etiqueta 12 não pode ser associada a nenhuma outra unidade experimental. Assim, em termos gerais,

$$\sum_k \delta_{ij}^k = 1, \quad \sum_{ij} \delta_{ij}^k = 1$$

Isto implica que,

$$\begin{aligned} P(\delta_{ij}^k = 1, \delta_{i'j'}^{k'} = 1) &= 0 \quad (ij) \neq (i'j') \\ P(\delta_{ij}^k = 1 \mid \delta_{i'j'}^{k'} = 1) &= 0 \quad (ij) \neq (i'j') \\ P(\delta_{ij}^k = 1 \mid \delta_{ij}^{k'} = 1) &= 0 \quad k \neq k' \end{aligned}$$

A partir dessas definições das variáveis aleatórias, pode-se derivar o modelo linear para um plano completamente aleatorizado. Assim, se a UE k recebe a etiqueta ij , o tratamento i é aplicado à UE k . Ao finalizar o período

experimental uma observação é obtida, denotada por y_{ij} , ou seja, a observação sobre a j -ésima ocorrência do i -ésimo tratamento. Se o tratamento i é aplicado à UE k , a verdadeira resposta é o número T_{ik} . Supondo que todos os tratamentos pudessem ser aplicados a uma mesma UE, observaríamos todos os T_{ik} . Mas não é possível realizar este procedimento, pois aplicar o processo de aleatorização implica aplicar apenas um tratamento a uma UE. Usando as variáveis aleatórias modelo δ_{ij}^k , pode-se relacionar os Y_{ij} com os T_{ik} como

$$y_{ij} = \sum_{k=1}^N \delta_{ij}^k T_{ik}$$

Depois de alguns desenvolvimentos matemáticos, para mais detalhe ver [11], considera-se o seguinte modelo:

$$y_{ij} = \mu + \tau_i + e_{ij} \tag{4.1}$$

4.2 Teste de aleatorização em planos experimentais

De acordo com [7], o teste de aleatorização é uma função (chamada de função teste) $\phi(x)$, definida por $x \in \chi$, tal que $0 \leq \phi(x) \leq 1$ para todo $x \in \chi$. Observando X , lançaria-se uma moeda com probabilidade $\phi(x)$ de obter cara. H_0 seria rejeitada se acontecesse cara, obtendo-se coroa H_0 não seria rejeitada. Note-se que $\phi(\cdot)$ é também chamada de função crítica.

Um critério de rejeição para o teste de aleatorização é dado pelos mínimos quadrados. Escolhe-se aleatoriamente um plano de um conjunto de planos experimentais e calcula-se o critério do teste para o atual plano, igual a C_a , em seguida, calcula-se o critério para cada um dos planos do conjunto de planos experimentais e obtém-se o nível de significância para a hipótese de que os tratamentos não têm diferença na média de seus efeitos, como:

$$\frac{1}{s} [\text{número de } C \geq C_a]$$

onde s é o número de planos possíveis.

Considerando a hipótese de que os tratamentos não têm diferenças na média de seus efeitos, deveria-se observar respostas iguais em uma UE independente do tratamento que seja aplicado.

Como exemplo básico, apresenta-se detalhadamente o teste exato para um plano completamente aleatorizado. O número total de planos experimentais que podem ser usados no processo da aleatorização para um plano completamente aleatorizado é:

$$s = \frac{N!}{(r!)^t} = \frac{(tr)!}{(r!)^2}. \quad (4.2)$$

Denota-se cada um desses planos como Π_γ ($\gamma = 1, 2, \dots, s$). Então, se os tratamentos não têm efeitos diferenciais, o mesmo resultado é obtido, ou seja, as mesmas respostas para todos os Π_γ .

Agora, tem-se que construir um critério de avaliação da hipótese, que é determinado por um arranjo experimental real, ou potencial, a ser usado. Um critério depende do plano de tratamento se, e somente se, as observações são indexadas por (ij) , com i denotando o tratamento e j a réplica dentro do tratamento, a qual é invariante sob permutações de j dentro de i . A variedade desses critérios C é ilimitada. Alguns exemplos são:

- a soma dos quadrados dos tratamentos, $SQ(T)$, na tabela ANOVA;
- $QM(T)/QM(E)$;
- O posto do total de tratamentos.

Uma vez escolhido o critério C , calcula-se o valor de C para todos os possíveis planos Π_γ , obtendo um conjunto de números $\{C_\gamma, \gamma = 1, 2, \dots, s\}$. Cabe ressaltar que existe $s^* = s/(t!)$ valores diferentes de C_γ , dado que cada permutação dos tratamentos marcados tem o mesmo valor para C . Denota-se por s^* os diferentes valores para C_{γ^*} . Entre estes números, encontra-se o valor associado ao plano usado, o qual será denotado por C_0 . Assim, o nível de significância contra a hipótese nula de não diferença na média dos efeitos de tratamentos com o critério C escolhido é:

$$\begin{aligned}
SL &= \frac{1}{s} [\text{número de } C_\gamma(\gamma = 1, 2, \dots, s) \geq C_0] \\
SL &= \frac{1}{s^*} [\text{número de } C_{\gamma^*}(\gamma^* = 1, 2, \dots, s^*) \geq C_0].
\end{aligned}
\tag{4.3}$$

Assumindo α como o nível de significância, o qual pode ser um múltiplo de $1/s^*$, sob a hipótese nula de efeitos de tratamentos não diferenciáveis, pode-se concluir que a probabilidade de obter um nível de significância menor ou igual que α é α . O argumento é consistente, sob a hipótese nula, uma vez que observa-se, com probabilidade $1/s^*$, um dos números do conjunto $\{C_\gamma, \gamma = 1, 2, \dots, s\}$. Este procedimento é conhecido como teste aleatório ou teste exato [11].

Por outro lado, espera-se que a distribuição aleatória de QM(T)/QM(E) esteja bem representada pela distribuição F com $t - 1$ e $t(r - 1)$ graus de liberdade. Neste sentido, considera-se o teste F para testar a hipótese nula de efeitos de tratamentos não diferenciáveis como uma boa aproximação ao teste aleatório (4.3). A sugestão de que o teste aleatório possa ser aproximado pelo teste F não é sempre tão certa. Algumas vezes, quando o número de réplicas é pequeno ou quando o número de tratamentos é pequeno, pode chegar a apresentar diferenças significativas entre o nível de significância nominal (Nominal Significance Level-NSL) e o nível de significância aleatório (Randomized Significance Level-RSL). Para ilustrar esse fato, efetuou-se um estudo de simulação para um plano completamente aleatorizado, especialmente trabalhando no caso de dois tratamentos de interesse, pelo qual se definem seis marcos de estudo, sendo eles um plano completamente aleatorizado (PCA) com três, cinco, oito, dez, doze e quinze réplicas, respectivamente. Para o caso de três réplicas, tem-se seis UE com variáveis respostas iguais a 0, 1, 2, 3, 4, 5, atribuídas a cada uma das seis UE. Para cinco réplicas, os valores de 0, 1, ..., 9. No caso de oito réplicas, atribuem-se os valores resposta 0, 1, ..., 6, 7, a cada uma das dezesseis UE, de forma similar, procede-se para o caso de doze réplicas. Para dez e quinze réplicas, atribui-se os valores resposta de 0, 1, ..., 8, 9 a cada uma das vinte e trinta UE, respectivamente.

O estudo de simulação consiste em selecionar aleatoriamente um determinado número de todos os possíveis planos aleatórios de dois tratamentos com o valor de réplica correspondente. Para os seis casos de estudo, escolhe-se aleatoriamente 1500 planos aleatórios. Cabe ressaltar que a capacidade e poder computacional do software SAS, este número de planos pode ser bem maior, mas para este exemplo usa-se um número relativamente baixo. Para cada um desses planos escolhidos, calcula-se o valor $F = QM(T)/QM(E)$, denotado por $F^{(1)}, F^{(2)}, \dots, F^{(1500)}$, obtendo o nível e significância em dois sentidos:

- Níveis de significância baseados na distribuição F com $(t - 1)$ e $t(r - 1)$ graus de liberdade, *NSL*.
- Níveis de significância baseados no posto dos 1500 valores F calculados como em (4.3), *RSL*.

Realizado este processo os resultados obtidos são apresentados nos seis gráficos que compõem a Figura 4.1, onde pode-se observar que quase todos os valores do *RSL* são maiores que os valores do *NSL*, com algumas exceções, por exemplo, nos cinco últimos gráficos, para valores próximos de um, o valor de *RSL* é quase o mesmo do *NSL*. Similarmente, para os valores entre 0 e 0.1, com exceção do caso em que se usou três réplicas. Por outro lado, para os valores que estão na faixa de 0.3 a 0.9, é notável que os *RSL* sejam maiores que os *NSL*, com a característica de que a medida que aumenta o número de réplicas essa diferença parece diminuir, fazendo que os valores do *NSL* aproximem-se dos valores do *RSL*. Estas grandes discrepâncias são, naturalmente, devido à singularidade dos *RSL*, especialmente quando o número de tratamentos e réplicas é pequeno, confirmando a sugestão de que o teste de aleatorização não sempre pode ser aproximado pelo teste F , pois o número de possíveis planos aleatórios é baixo, tal como se pode calcular com (4.2), e também o fato de que o mesmo valor de F pode ocorrer mais de uma vez.

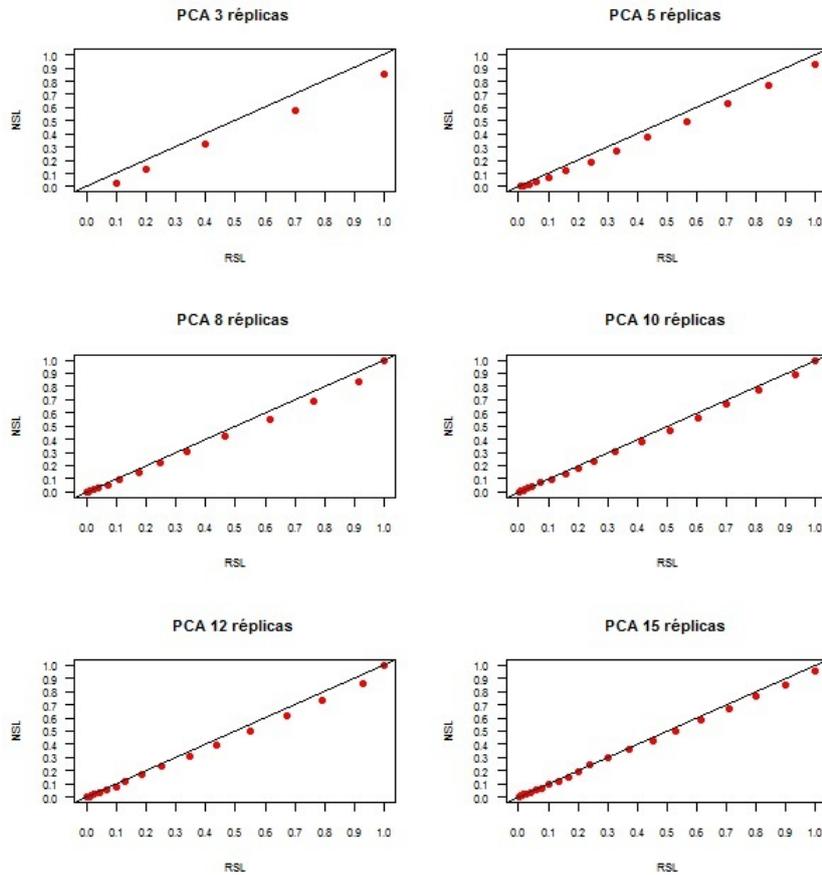


Figura 4.1: Níveis de significância do teste aleatório frente ao teste aproximado F para um PCA

Assim, para os casos apresentados neste exemplo, tem-se que os respectivos números de planos aleatórios são:

- PCA dois tratamentos e três réplicas: $s = 20$ e $s^* = 10$
- PCA dois tratamentos e cinco réplicas: $s = 252$ e $s^* = 126$
- PCA dois tratamentos e oito réplicas: $s = 12870$ e $s^* = 6435$
- PCA dois tratamentos e dez réplicas: $s = 184756$ e $s^* = 92378$
- PCA dois tratamentos e doze réplicas: $s = 2704156$ e $s^* = 1352078$

- PCA dois tratamentos e quinze réplicas: $s = 155117520$ e $s^* = 77558760$

4.3 Análise de um quadrado latino 2×2 com réplicas

Quando são identificados dois tipos de fontes de heterogeneidade que afetam sistematicamente a resposta das UE devido à natureza ou arranjo das mesmas, o quadrado latino é um plano efetivo para afastar e diferenciar estas fontes de variabilidade do erro experimental.

Nas seções seguintes são apresentadas as razões principais do uso conveniente de um plano aleatorizado em quadrados latinos, especialmente de ordem 2, assim como as possíveis dificuldades que pode gerar o uso deste plano experimental. Posteriormente, é desenvolvido um estudo de simulação para verificar o comportamento do teste aproximado frente ao teste aleatório, e finalmente calcula-se o valor do teste exato para os dados gerados em [2], como também apresenta-se a função poder para o teste aproximado, usando as variabilidades obtidas no estudo inicial efetuado em [2].

Conhecer a importância que tem os quadrados latinos 2×2 na engenharia de software e, especialmente, no processo de experimentação, para comparar abordagens diferentes em linhas de produtos de software é fundamental para o desenvolvimento deste trabalho. Por que é preciso planos em quadrados latinos na aplicação à engenharia de software? Para responder a esta pergunta, apresenta-se, inicialmente, um plano mais simples, o plano aleatorizado em blocos completos (PABC).

De acordo com [11] o PABC pode ser definido como segue: "O material experimental é dividido dentro de b conjuntos de t UE cada um, onde t é o número de tratamentos, tal que as UEs dentro de cada conjunto sejam homogêneas como seja possível e que as diferenças entre as UEs seja considerado, por muito, as diferenças entre os conjuntos. Os conjuntos são chamados de blocos. Dentro de cada bloco os t tratamentos são alocados aleatoriamente

as UEs, onde cada tratamento acontece exatamente uma vez dentro de cada bloco, por outro lado aleatorizações independentes são usadas nos b blocos".

Assim, para o caso do experimento, um possível fator que define o bloco são os produtos de software a serem usados. Portanto, dentro de cada bloco tem-se duas UEs, ou seja, dois analistas para cada bloco, aplicando os dois tratamentos objeto de estudo, PLUSS e MSVCM, em cada bloco. Tomando o caso de oito réplicas, que foi o número mais usado nas fases de experimentação em [2], seriam necessários oito produtos de software diferentes e dezesseis analistas para execução do experimento, tal como se observa na Figura 4.2.

O plano aleatorizado em blocos completos é um processo de experimentação bastante custoso e até inviável na maioria dos casos, pois reunir tal quantidade de analistas e efetuar um processo de capacitação em cada um dos produtos de software empregados, tal como realizou-se em [2] é uma tarefa difícil de ser cumprida.

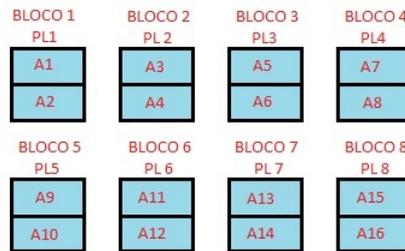


Figura 4.2: Gráfico de um PABC com oito réplicas

Por exemplo, um quadrado latino, neste caso 2×2 , com linhas embutidas nas réplicas, tal como na figura 2.3, mas tomando o mesmo número de réplicas do exemplo anterior, oito réplicas, precisa dos mesmos dezesseis analistas para o processo de experimentação, mas usaria apenas dois produtos de software, o qual facilitaria o processo de capacitação e realização do experimento.

A razão mais importante pela qual justifica o uso do plano aleatorizado em quadrados latinos 2×2 é que, além de controlar o fator produtos de software, tal como em PABC, também controla possíveis diferenças dos analistas que

não seria controlada em PABC, pois tal efeito pode ser confundido com o efeito dos tratamentos. É válido ressaltar que dificilmente se encontraria dois prováveis analistas que gerem as mesmas capacidades de desenvolvimento de uma linha de produtos de software, assim como condições de experiência na área, habilidade, conhecimento, entre outras condições que, com certeza, não serão homogêneas no total de analistas.

Tem-se em conta também que para quadrados pequenos, geralmente de ordem menor que 4, a realização de um número de réplicas baixo pode dificultar o fato de obter níveis de precisão desejáveis, o que se reflete nos poucos graus de liberdade disponíveis para o erro experimental. Na figura 4.3 se compara, esquematicamente, os graus de liberdade das fontes de variação em planos experimentais com controle de fatores de ruído gradativo.

Assumindo cinco tratamentos alocados à vinte e cinco unidades experimentais com igual número de réplicas por tratamento, a Figura 4.3 ilustra a perda de graus de liberdade do erro experimental a medida que se escolhe um plano experimental mais complexo. A menos que haja variação suficiente no erro experimental do PCA, que pode ser atribuído à alguma variável escolhida de bloqueio, para compensar a perda de graus de liberdade do erro experimental, o plano experimental resultante será menos eficiente.

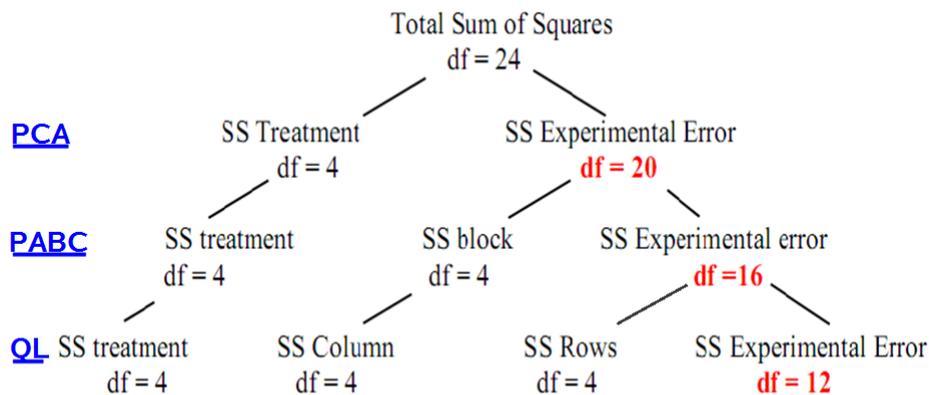


Figura 4.3: Graus de liberdade da ANOVA para um PCA, PABC e QL [8]

Um exemplo numérico do efeito que causa a perda de graus de liberdade no erro sobre o valor crítico F é apresentado na figura 4.4.

<u>3x3 Quadrado latino</u>	<u>5x5 Quadrado latino</u>
LS $df_e = 2$ (PABC $df_e = 4$)	LS $df_e = 12$ (PABC $df_e = 16$)
LS $F_{0.05,2,2} = 19.0$	LS $F_{0.05,4,12} = 3.26$
PABC $F_{0.05,2,4} = 6.94$	PABC $F_{0.05,4,16} = 3.01$

Figura 4.4: Efeito numérico na perda de graus de liberdade no erro

Desta maneira, observa-se que a perda de graus de liberdade no erro tem um efeito enorme na sensibilidade dos quadrados latinos, especialmente quando se tem poucos tratamentos e baixo número de réplicas. Caso em que um quadrado latino de ordem cinco com valores de F ao redor de três pula para um valor de dezenove quando o quadrado latino é de ordem três, assim como os graus de liberdade do erro experimental passam de doze para dois, sendo notório o efeito que causa a perda de graus de liberdade.

4.3.1 Estudo de simulação

O fundamento básico do processo de simulação é baseado na aleatorização, devido à importância da propriedade de aleatoriedade que é crucial para validar as conclusões obtidas, assim como neutralizar os efeitos de algum viés sistemático. Uma característica de um arranjo em quadrado latino é que, cada tratamento só pode aparecer em uma única linha e coluna do quadrado (seção 2.1). Sendo assim, tem-se um número determinado de possíveis quadrados latinos, onde o objetivo é escolher um destes possíveis arranjos de forma aleatória. Assim um procedimento para obter um quadrado aleatoriamente é o seguinte:

1. Seleciona-se um quadrado latino padrão para o número de tratamentos envolvidos. Por exemplo, se os tratamentos são quatro, se pode selecionar o seguinte quadrado latino padrão:

A B C D
B C D A
C D A B
D A B C

2. Por algum procedimento de geração de números aleatórios, seleciona-se dois conjuntos de números aleatórios com comprimento igual ao número de tratamentos envolvidos e depois ordena-se estes conjuntos de números aleatórios. Por exemplo:

	Grupo 1 (para colunas)	Grupo 2 (para linhas)
Números aleatórios	9 1 6 4	8 5 3 7
Ordem	4 1 3 2	4 2 1 3

3. Rotula-se as linhas e colunas do quadrado latino reduzido, escolhido no primeiro item, com a ordem obtida no item anterior (a). Depois ordenam-se as linhas de acordo com a lista ordenada (b) e por último ordenam-se as colunas de acordo a lista ordenada (c).

a. Alocar a ordem

	<u>4</u>	<u>1</u>	<u>3</u>	<u>2</u>
4	A	B	C	D
2	B	C	D	A
1	C	D	A	B
3	D	A	B	C

b. Ordenar linhas

	<u>4</u>	<u>1</u>	<u>3</u>	<u>2</u>
1	C	D	A	B
2	B	C	D	A
3	D	A	B	C
4	A	B	C	D

c. Ordenar colunas

	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
1	D	B	A	C
2	C	A	D	B
3	A	C	B	D
4	B	D	C	A

4. Finalmente, o novo quadrado latino é gerado e os tratamentos gerais A-D são alocados aleatoriamente aos 4 tratamentos. Assim, o quadrado mostra quais tratamentos são alocados às unidades experimentais.

Uma forma de automatizar o processo de aleatorização através do software estatístico SAS é por meio do procedimento PROC PLAN apresentado na seção 4.1. Neste caso, a seguinte sintaxe gera um quadrado latino 2 × 2:

```
proc plan seed=1234;
```

```

factors row=2 ordered
      col=2 ordered ;
treatments treat=2 cyclic;
output out=OutFromProcPlan
      Row nvals=(1 2) random
      Col nvals=(1 2) random
      treat nvals=(1 2) random;

run;

```

Observa-se no código anterior que as linhas e colunas são identificadas como ordenadas (ordered) e os tratamentos como cíclicos (cyclic). As duas declarações dos fatores Row e Col geram aleatoriamente a ordem para reorganizar as linhas e colunas, como se indica no item três do processo de aleatorização acima explicado. Na figura 4.5, apresenta-se a saída correspondente ao código descrito.

```

■
The PLAN Procedure
Plot Factors
Factor      Select    Levels    Order
row         2          2        Ordered
col         2          2        Ordered

Treatment Factors
Factor      Select    Levels    Order    Initial
              Block /
              Increment
treat       2          2        Cyclic   (1 2) / 1

      row    -col-    -treat-
      1     1 2     1 2
      2     1 2     2 1

```

Figura 4.5: Saída gerada por SAS no processo de aleatorização

Uma vez escolhido o quadrado latino no processo de aleatorização, momentaneamente para 1500 iterações de Monte Carlo, cabe ressaltar que a capacidade e poder computacional do software SAS este número de iterações pode ser bem maior. O cálculo do valor da estatística $F = QM(T)/QM(E)$ é facilmente obtido usando o procedimento PROC GLM, tal como se apresenta na seguinte sintaxe.

```

proc glm data=modelo1 outstat=tabla1 noprint;
class rep col row treat;
model y = rep col row treat;
run;

```

Calculando os valores de F e denotando-os por $F^{(1)}, F^{(2)}, \dots, F^{(1500)}$, obtém-se o nível de significância em dois sentidos:

- Níveis de significância baseados na distribuição F com $(t - 1)$ graus de liberdade no tratamento e uma quantidade de graus de liberdade no erro experimental de acordo com cada um dos cenários de replicação.
- Níveis de significância baseados no posto dos 1500 valores F calculados como em (4.3), mas com os valores gerados para um plano em quadrado latino.

Para isso foi desenvolvido um programa utilizando o software estatístico SAS 9.2, (o código para gerar as simulações de Monte Carlo para uma das rodadas com um número de réplicas determinado esta disponível no Apêndice A), para gerar os valores do teste de aleatorização e o teste aproximado para diferentes números de réplicas nas possíveis formas de fazer replicação nos experimentos em quadrados latinos.

Com base no trabalho realizado na fase de experimentação de [2], três etapas experimentais foram estabelecidas, denominadas por rodada 1, rodada 2 e rodada 3, respectivamente. Em cada uma das rodadas o processo experimental é dividido em duas fases, uma referente à avaliação dos tempos de desenvolvimento de uma SPL, chamada fase 01, e outra referente à avaliação dos tempos de manutenção de uma SPL, considerado como fase 02, e para cada uma das fases foi usado um número determinado de réplicas. Os cenários finalmente simulados são apresentados na Tabela 4.1.

Em suma, foram realizados seis experimentos comparando os métodos PLUSS e MSVCM em termos do esforço requerido para extrair e fazer manutenção de especificações nas linhas de produtos de software. Para cada

Tabela 4.1: Cenários de simulação quadrado latino 2×2 com réplicas

Etapa	Processo	# de réplicas
Rodada 1	fase 01	5
	fase 02	5
Rodada 2	fase 01	11
	fase 02	8
Rodada 3	fase 01	8
	fase 02	8

um dos cenários planejados, tomam-se como variáveis respostas no estudo de simulação, os dados adquiridos no processo de experimentação de [2], fornecidos pelo autor e com a autorização do mesmo para uso deles nesta dissertação.

4.3.2 Resultados do estudo de simulação

Para cada um dos cenários apresentados na Tabela 4.1 foram estudadas as quatro formas de obter réplicas em quadrados latinos, com o fim de entender os quatro processos ilustrados na seção 2.4. É válido salientar que quando o número de tratamentos é pequeno, neste caso simplesmente dois, e tem-se uma variabilidade considerável para que possa ser tratada em um experimento com dois fatores de bloqueio, emprega-se quadrados latinos incrementando o número de réplicas para aumentar o número de graus de liberdade do erro experimental, sendo cada um dos quadrados latinos considerado independente dos demais quadrados latinos realizados na totalidade do experimento.

Em seguida, são apresentados a análise e os resultados do estudo de simulação, tanto graficamente como numa tabela com alguns valores de significância pequenos para cada um dos cenários da Tabela 4.1.

4.3.2.1 Primeira rodada de experimentação

FASE 01

Realizado o processo de simulação anteriormente mencionado, os resultados obtidos são apresentados na Figura 4.6, onde observa-se que a maioria de valores do RSL são maiores que o NSL, especialmente no intervalo acima de 0.8 aparece a diferença máxima. Poucas exceções no caso em que as linhas estão embutidas nas réplicas apresentam valores de NSL maiores que RSL.

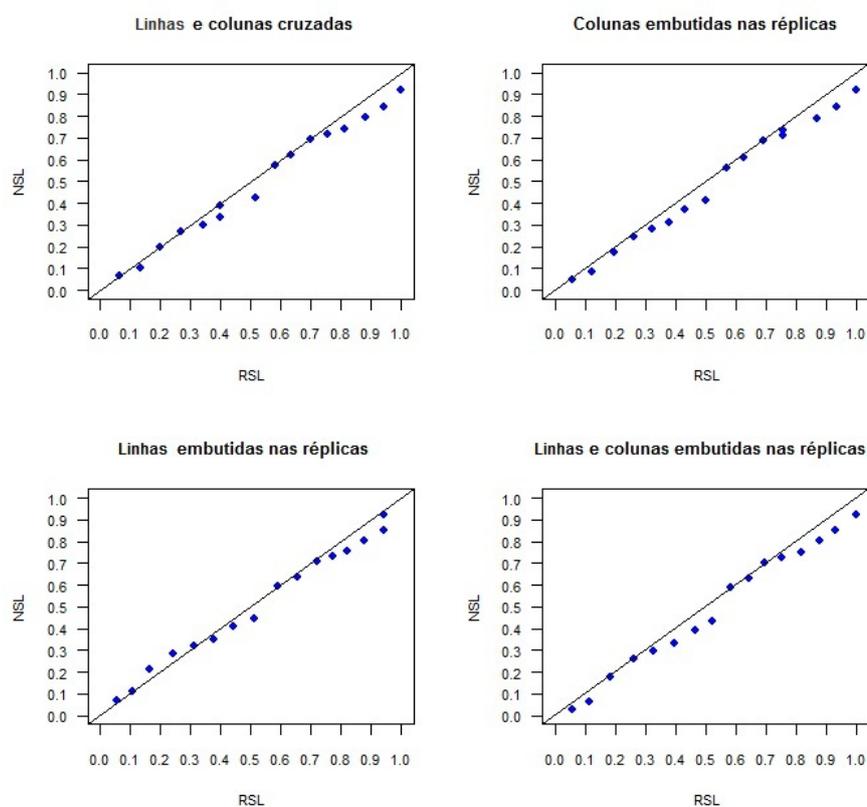


Figura 4.6: Quadrados latinos 2×2 com 5 réplicas. Rodada 1. Fase 01

Essas grandes discrepâncias são, naturalmente, devido à singularidade dos RSL, especialmente quando o número de tratamentos e réplicas são pequenos, pois o número de possíveis planos aleatórios é baixo e também o fato de que o mesmo valor de F pode ocorrer mais de uma vez.

Em geral, é evidente que no caso onde cinco réplicas são realizadas que a forma menos apropriada é linhas embutidas nas réplicas, já que a aproxi-

Tabela 3.3 Linhas e colunas cruzadas. R1.F01

Valor F	NSL	RSL	Frequência
4.015	0.068	0.065	0.065
3.091	0.104	0.135	0.070
1.855	0.198	0.201	0.066
1.348	0.268	0.270	0.069
1.174	0.300	0.342	0.072
1.015	0.334	0.398	0.056
0.802	0.388	0.398	0.062
0.677	0.427	0.518	0.058
0.330	0.576	0.581	0.063
0.256	0.622	0.635	0.054
0.162	0.694	0.700	0.065
0.136	0.719	0.757	0.057

Tabela 3.4: Colunas embutidas na réplica. R1.F01

Valor F	NSL	RSL	Frequência
5.470	0.048	0.057	0.057
3.971	0.081	0.123	0.065
2.215	0.175	0.197	0.075
1.564	0.246	0.262	0.065
1.349	0.279	0.322	0.060
1.156	0.314	0.381	0.059
0.904	0.370	0.433	0.052
0.758	0.409	0.499	0.066
0.363	0.564	0.569	0.070
0.280	0.611	0.627	0.058
0.177	0.685	0.692	0.065
0.148	0.710	0.757	0.065

Tabela 3.5: Linhas embutidas na réplica. R1.F01

Valor F	NSL	RSL	Frequência
4.316	0.071	0.057	0.057
3.209	0.111	0.110	0.053
1.842	0.212	0.164	0.054
1.315	0.285	0.243	0.079
1.138	0.317	0.311	0.068
0.979	0.352	0.378	0.067
0.768	0.406	0.443	0.065
0.645	0.445	0.514	0.071
0.311	0.592	0.589	0.075
0.240	0.637	0.657	0.068
0.152	0.707	0.723	0.066
0.127	0.730	0.774	0.051

Tabela 3.6 Linhas e colunas embutidas na réplica. R1.F01

Valor F	NSL	RSL	Frequência
12.208	0.025	0.057	0.057
6.398	0.065	0.116	0.059
2.692	0.176	0.183	0.067
1.742	0.257	0.263	0.079
1.462	0.293	0.327	0.065
1.224	0.331	0.398	0.071
0.928	0.390	0.466	0.068
0.764	0.431	0.525	0.059
0.350	0.586	0.583	0.059
0.267	0.632	0.644	0.061
0.167	0.704	0.699	0.055
0.140	0.728	0.753	0.054

mação ao teste exato não é tão efetiva. Um fato importante a mencionar é que de acordo com as Tabelas 3.3 a 3.6 apresentadas para os níveis de significância mais baixos não existe tendência alguma a rejeitar a hipótese nula de igualdade nos tempos de desenvolvimento de uma linha de produto.

FASE 02

Na análise do esforço nas especificações de manutenção de uma linha de produto com cinco réplicas, conseguiu-se os resultados da Figura 4.7. Diferentemente da fase 01 da rodada 1, tem-se que os valores de NSL são maiores que os níveis de significância aleatório a partir de 0.0 até 0.4 – 0.5. Acima dessa faixa, ou seja de 0.4 – 0.5 até 1.0, os valores do RSL são maiores que o NSL. Em nenhuma das quatro formas de replicação obtém-se uma aproximação satisfatória do teste F ao teste exato. A melhor aproximação é aquela em que linhas e colunas são embutidas nas réplicas. No entanto, cinco réplicas não é o cenário mais apropriado para avaliar à hipótese nula de tratamentos com efeitos diferenciais, pois as discrepâncias são grandes entre NSL e RSL.

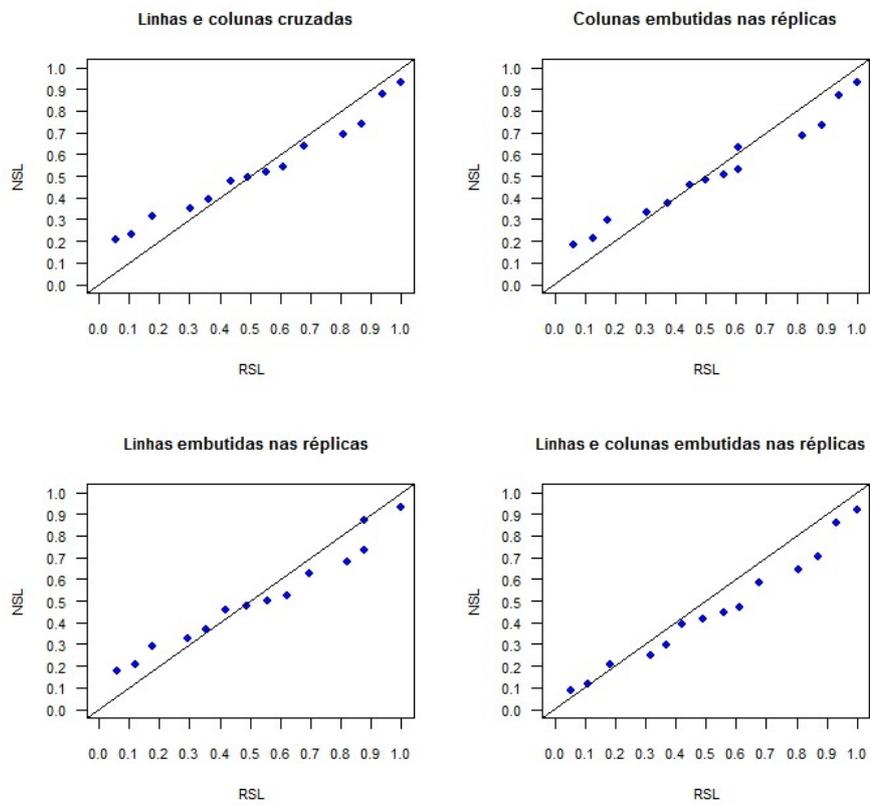


Figura 4.7: Quadrados latinos 2×2 com 5 réplicas. Rodada 1. Fase 02

Cabe notar que com respeito às Tabelas 3.7 a 3.10 apresentadas para os níveis de significância mais baixos, não se tem uma tendência clara a rejeitar a hipótese nula de igualdade nos tempos de manutenção de uma linha de produto. Como se pode observar na primeira linha das Tabelas 3.7, 3.8 e 3.9, o teste aproximado pode incorrer em conclusões erradas, pois para um $\alpha = 10\%$ não se rejeitaria a hipótese nula quando realmente ela deveria ser rejeitada de acordo com o teste exato.

Tabela 3.7 Linhas e colunas cruzadas. R1.F02

Valor F	NSL	RSL	Frequência
1.800	0.205	0.056	0.056
1.577	0.233	0.107	0.051
1.101	0.315	0.177	0.071
0.940	0.351	0.301	0.123
0.795	0.390	0.364	0.063
0.546	0.474	0.435	0.071
0.492	0.497	0.493	0.059
0.441	0.519	0.551	0.058
0.393	0.543	0.611	0.059
0.231	0.640	0.678	0.067
0.167	0.690	0.808	0.130
0.113	0.742	0.869	0.061

Tabela 3.8: Colunas embutidas na réplica. R1.F02

Valor F	NSL	RSL	Frequência
2.114	0.184	0.062	0.062
1.829	0.213	0.128	0.066
1.245	0.297	0.177	0.049
1.054	0.335	0.305	0.129
0.884	0.375	0.372	0.067
0.599	0.461	0.447	0.075
0.538	0.484	0.497	0.051
0.481	0.507	0.559	0.062
0.428	0.531	0.607	0.047
0.249	0.631	0.607	0.072
0.179	0.683	0.817	0.139
0.122	0.736	0.885	0.067

Tabela 3.9 Linhas embutidas na réplica. R1.F02

Valor F	NSL	RSL	Frequência
2.179	0.178	0.059	0.059
1.884	0.207	0.119	0.060
1.280	0.291	0.175	0.057
1.084	0.328	0.295	0.120
0.908	0.368	0.356	0.061
0.615	0.456	0.420	0.064
0.552	0.479	0.487	0.067
0.494	0.502	0.559	0.072
0.439	0.526	0.621	0.062
0.256	0.627	0.694	0.073
0.184	0.679	0.823	0.129
0.125	0.733	0.879	0.056

Tabela 3.10 Linhas e colunas embutidas na réplica. R1.F02

Valor F	NSL	RSL	Frequência
5.074	0.087	0.055	0.055
3.967	0.117	0.111	0.055
2.254	0.208	0.184	0.073
1.811	0.250	0.318	0.134
1.452	0.295	0.367	0.049
0.917	0.393	0.421	0.054
0.812	0.419	0.491	0.070
0.716	0.445	0.560	0.069
0.629	0.472	0.613	0.053
0.352	0.585	0.679	0.066
0.249	0.644	0.805	0.126
0.167	0.704	0.870	0.065

4.3.2.2 Segunda rodada de experimentação

FASE 01

Para realizar a análise do esforço no desenvolvimento de uma linha de produto, ou seja, fase 01, foram feitas onze réplicas e obtidos os resultados da Figura 4.8. Os valores de RSL são menores que os valores dos NSL em quase todos os pontos. Vale ressaltar que para o caso em que se replica os quadrados latinos embutindo tanto linhas como colunas, a aproximação dos valores dos níveis de significância nominal e aleatório é bastante satisfatória.

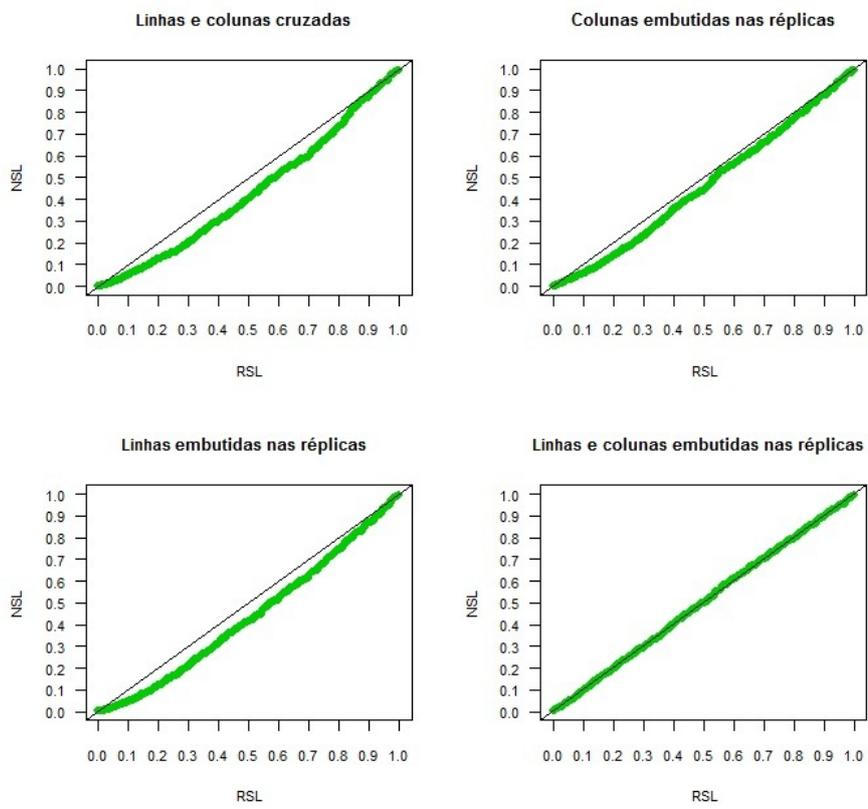


Figura 4.8: Quadrados latinos 2×2 com 11 réplicas. Rodada 2. Fase 01

Para as outras três formas de replicação, tem-se que na faixa de 0.1 a 0.8 a discrepância é relevante, sendo os RSL bem menores que os NSL. Dentro dessas três formas de replicar, quando se tem as colunas embutidas nas réplicas, a discrepância é menor, sendo a segunda melhor opção no caso de que não seja realizável embutir linhas e colunas nas réplicas. Realizando um aumento da figura na área de rejeição, particularmente de 0.0 até 0.1, tal como se observa na Figura 4.9, encontra-se que um bom número de pontos estão dentro da área de rejeição, mas nesta situação pode-se cair no caso em que através do teste F se pode rejeitar a hipótese nula, enquanto que para o teste exato esses mesmos pontos teria-se a situação contrária, ou seja, a hipótese nula não é rejeitada.

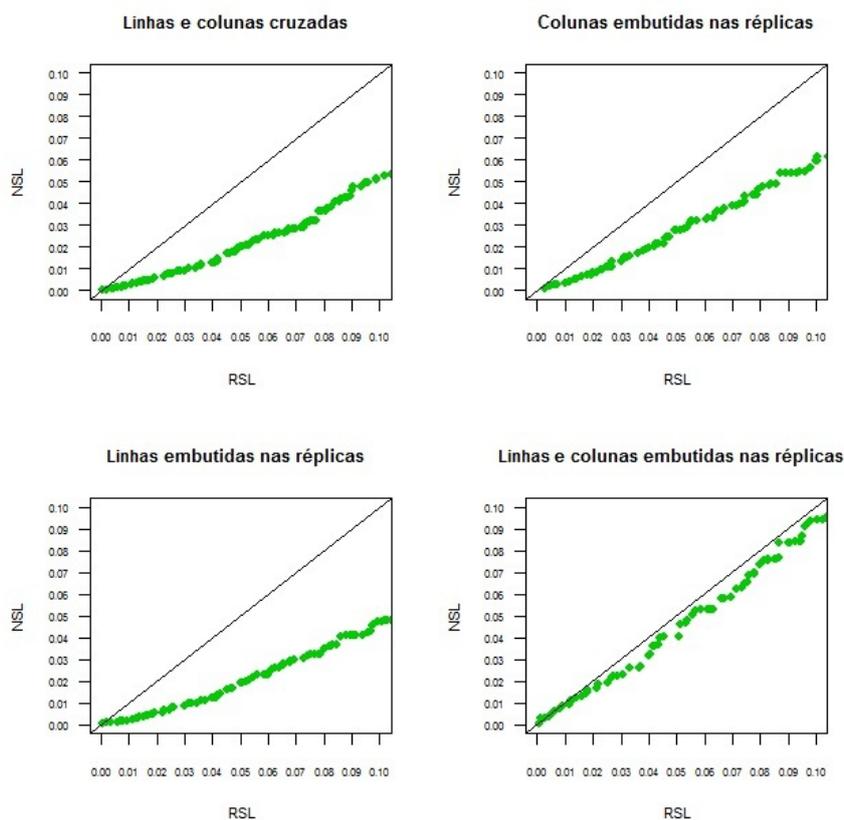


Figura 4.9: Quadrados latinos 2×2 com 11 réplicas. Rodada 2. Fase 01

Tabela 3.11 Linhas e colunas cruzadas. R2.F01

Valor F	NSL	RSL	Frequência
17.403	0.000	0.001	0.001
15.595	0.000	0.002	0.001
13.611	0.001	0.004	0.002
13.013	0.001	0.005	0.001
12.530	0.001	0.005	0.001
12.194	0.002	0.006	0.001
12.159	0.002	0.007	0.001
11.655	0.002	0.008	0.001
11.219	0.002	0.009	0.001
10.649	0.003	0.011	0.002
10.149	0.003	0.011	0.001
10.119	0.003	0.013	0.001

Tabela 3.12: Colunas embutidas na réplica. R2.F01

Valor F	NSL	RSL	Frequência
15.071	0.001	0.003	0.003
12.901	0.002	0.004	0.001
12.263	0.002	0.005	0.001
11.754	0.003	0.006	0.001
11.402	0.003	0.007	0.001
11.365	0.003	0.010	0.003
10.394	0.004	0.011	0.001
9.812	0.005	0.013	0.002
9.756	0.005	0.014	0.001
9.308	0.006	0.016	0.002
8.923	0.007	0.018	0.002
8.894	0.007	0.019	0.001

Tabela 3.13: Linhas embutidas na réplica. R2.F01

Valor F	NSL	RSL	Frequência
18.287	0.000	0.001	0.001
15.450	0.001	0.002	0.001
15.395	0.001	0.003	0.001
13.980	0.001	0.006	0.003
13.533	0.001	0.007	0.001
13.486	0.002	0.008	0.001
12.826	0.002	0.009	0.001
12.263	0.002	0.011	0.002
11.537	0.003	0.013	0.001
11.467	0.003	0.013	0.001
10.912	0.004	0.014	0.001
10.875	0.004	0.015	0.001

Tabela 3.14 Linhas e colunas embutidas na réplica. R2.F01

Valor F	NSL	RSL	Frequência
25.333	0.001	0.001	0.001
15.608	0.003	0.001	0.001
15.530	0.003	0.003	0.001
14.440	0.003	0.004	0.001
13.550	0.004	0.005	0.001
12.957	0.005	0.005	0.001
12.048	0.006	0.006	0.001
11.348	0.007	0.008	0.002
10.475	0.009	0.009	0.001
10.393	0.009	0.011	0.002
9.750	0.011	0.012	0.001
9.708	0.011	0.013	0.001

Com respeito à rodada 1, fase 01, é mais apropriado usar um bom número de réplicas, pois a aproximação é melhor. Por outro lado, é importante mencionar que se evidenciam valores nos quais se rejeitaria a hipótese nula a um nível de significância de $\alpha = 5\%$, com uma boa aproximação dos dois testes de acordo com as Tabelas 3.11 a 3.14.

FASE 02

Para a análise do esforço nas especificações de manutenção de uma linha de produto, nesta segunda rodada foram usadas oito réplicas, dando como resultado a Figura 4.10. Observa-se que tanto para o caso de linhas e colunas cruzadas como para o caso de colunas embutidas nas réplicas, os valores de NSL são maiores que os de RSL, sendo ao mesmo tempo as aproximações péssimas, especialmente a segunda. Por outro lado, nos casos de linhas embutidas nas réplicas e linhas e colunas embutidas nas réplicas, as aproximações são muito boas, praticamente iguais. No mais, nas linhas embutidas, os valores de RSL são levemente maiores que os de NSL, enquanto que nas linhas e colunas embutidas tem-se comportamentos mistos.

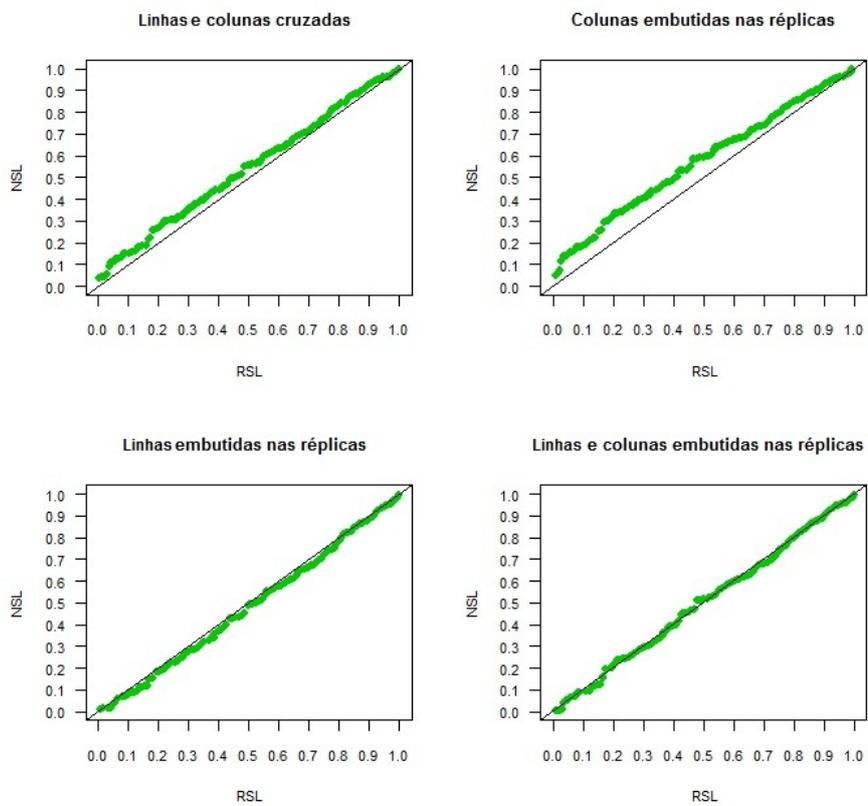


Figura 4.10: Quadrados latinos 2×2 com 8 réplicas. Rodada 2. Fase 02

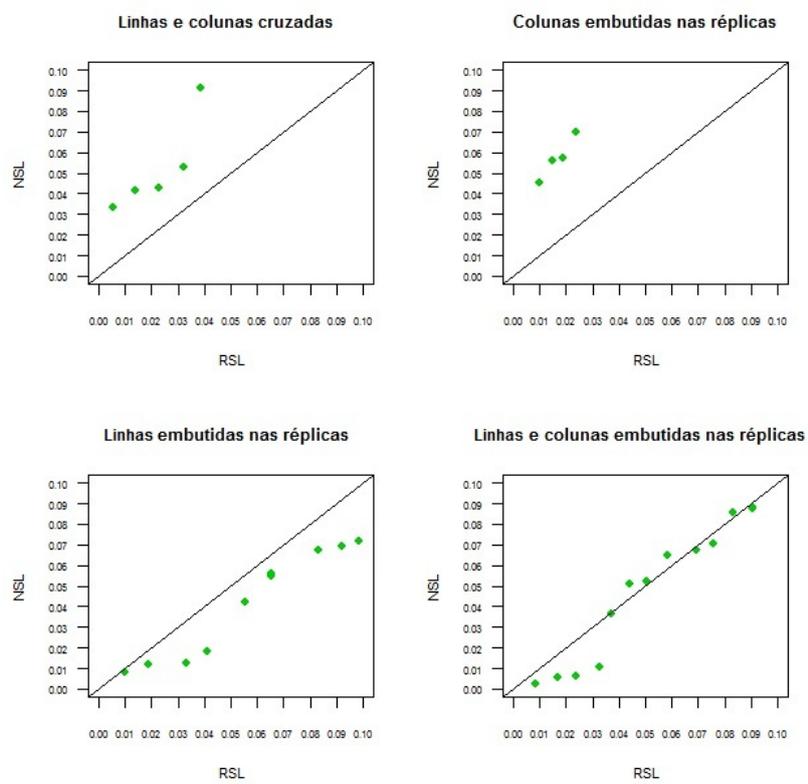


Figura 4.11: Quadrados latinos 2×2 com 8 réplicas. Rodada 2. Fase 02

Tabela 3.15 Linhas e colunas cruzadas. R2.F02

Valor F	NSL	RSL	Frequência
5.191	0.033	0.005	0.005
4.701	0.042	0.014	0.009
4.658	0.043	0.023	0.009
4.208	0.053	0.032	0.009
3.133	0.091	0.039	0.007
2.796	0.109	0.045	0.007
2.767	0.111	0.055	0.009
2.542	0.126	0.063	0.008
2.508	0.128	0.071	0.009
2.458	0.132	0.077	0.006
2.250	0.148	0.085	0.007
2.225	0.151	0.093	0.008

Tabela 3.16: Colunas embutidas na réplica. R2.F02

Valor F	NSL	RSL	Frequência
4.822	0.045	0.010	0.010
4.334	0.056	0.015	0.005
4.292	0.057	0.019	0.004
3.851	0.070	0.024	0.005
2.823	0.115	0.029	0.005
2.507	0.136	0.037	0.008
2.480	0.138	0.043	0.007
2.271	0.154	0.057	0.013
2.239	0.157	0.065	0.008
2.193	0.161	0.073	0.008
2.002	0.179	0.082	0.009
1.978	0.181	0.088	0.006

Tabela 3.17 Linhas embutidas na réplica. R2.F02

Valor F	NSL	RSL	Frequência
9.438	0.008	0.010	0.010
8.278	0.012	0.019	0.009
8.182	0.013	0.033	0.015
7.184	0.018	0.041	0.008
5.015	0.042	0.055	0.014
4.390	0.055	0.065	0.010
4.337	0.056	0.065	0.005
3.935	0.067	0.083	0.013
3.874	0.069	0.092	0.009
3.786	0.072	0.099	0.007
3.426	0.085	0.105	0.007
3.383	0.087	0.110	0.005

Tabela 3.18 Linhas e colunas embutidas na réplica. R2.F02

Valor F	NSL	RSL	Frequência
20.491	0.003	0.009	0.009
15.424	0.006	0.017	0.008
15.066	0.006	0.024	0.007
11.804	0.011	0.033	0.009
6.678	0.036	0.037	0.005
5.542	0.051	0.044	0.007
5.452	0.052	0.051	0.007
4.787	0.065	0.059	0.008
4.690	0.067	0.069	0.011
4.551	0.070	0.076	0.007
4.005	0.085	0.083	0.007
3.942	0.087	0.091	0.007

Aumentando a imagem na área de 0.0 até 0.1 da Figura 4.11, para o caso de linhas e colunas cruzadas e o caso de colunas embutidas nas réplicas surge a situação em que analisando o teste F não se tem evidência suficiente para rejeitar H_0 , mas por outro lado o teste exato rejeita a hipótese nula para estes mesmos valores ou pontos. Já para as outras duas formas de replicar quadrados latinos dificilmente obteria-se contradições nas conclusões.

Nas Tabelas 3.15 a 3.18 são apresentados os níveis de significância mais baixos, nos quais poderiam rejeitar a hipótese nula para um $\alpha = 10\%$. É válido ter em conta que a frequência desses valores é muito baixa, ao redor de 1%. Também se evidencia a boa aproximação dos valores para as formas de réplica apresentadas nas Tabelas 3.17 e 3.18.

4.3.2.3 Terceira rodada de experimentação

FASE 01

Usando os resultados obtidos na rodada 3 para analisar o esforço no desenvolvimento de linhas de produto de software, fase 01, gera-se a Figura

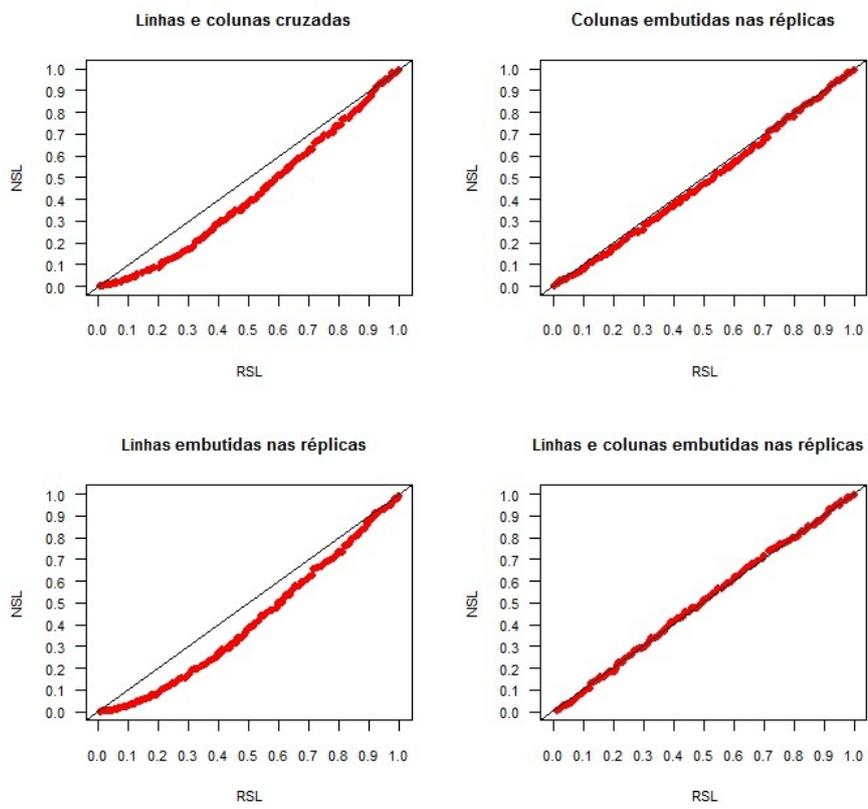


Figura 4.12: Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 01

4.12, na qual foram utilizadas oito réplicas, dando como resultado uma boa aproximação para o caso em que as colunas são embutidas nas réplicas e o caso em que tanto linhas como colunas são embutidas nas réplicas, sendo este o padrão para a fase 01.

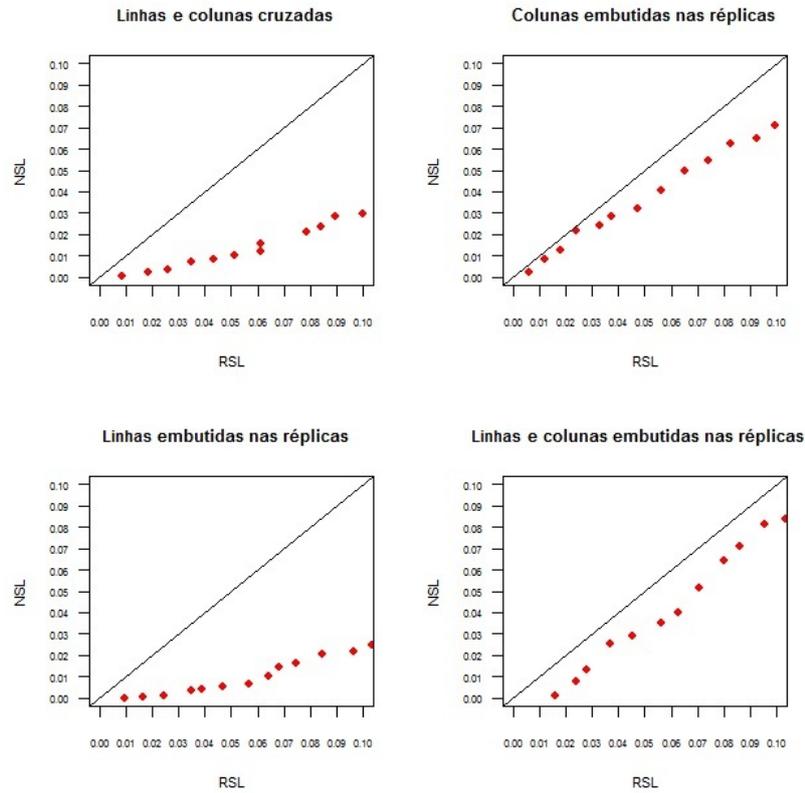


Figura 4.13: Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 01

Para os casos de linhas e colunas cruzadas e para o caso de linhas embutidas, as aproximações são muito ruins. Os valores que estão entre 0.5 e 0.9 distam bastante entre si, sendo os valores de RSL maiores que os valores de NSL. Efetuando um zoom da Figura 4.12 na área de rejeição, de 0.0 até 0.1, tal como se observa na Figura 4.13, encontra-se que tanto para linhas e colunas cruzadas como para linhas embutidas nas réplicas, pode-se cair no caso em que através do teste F pode-se rejeitar a hipótese nula, enquanto que a diferentemente do teste exato, nos mesmos pontos, teria o caso contrario, ou

Tabela 3.19 Linhas e colunas cruzadas. R3.F01

Valor F	NSL	RSL	Frequência
17.840	0.000	0.009	0.009
12.400	0.002	0.019	0.010
10.904	0.003	0.026	0.007
8.955	0.007	0.035	0.009
8.563	0.008	0.043	0.009
8.010	0.010	0.051	0.008
7.628	0.012	0.061	0.010
6.893	0.016	0.061	0.007
6.231	0.021	0.079	0.011
5.947	0.024	0.084	0.005
5.545	0.028	0.089	0.005
5.442	0.030	0.100	0.011

Tabela 3.20: Colunas embutidas na réplica. R3.F01

Valor F	NSL	RSL	Frequência
13.824	0.002	0.006	0.006
9.395	0.008	0.012	0.006
8.211	0.012	0.018	0.006
6.691	0.022	0.024	0.006
6.388	0.024	0.033	0.009
5.962	0.028	0.037	0.005
5.669	0.032	0.047	0.010
5.108	0.040	0.056	0.009
4.605	0.050	0.065	0.009
4.390	0.055	0.074	0.009
4.087	0.063	0.083	0.009
4.009	0.065	0.093	0.010

Tabela 3.21 Linhas embutidas na réplica. R3.F01

Valor F	NSL	RSL	Frequência
36.213	0.000	0.009	0.009
19.567	0.001	0.017	0.007
16.214	0.001	0.025	0.008
12.385	0.003	0.035	0.010
11.678	0.004	0.039	0.004
10.714	0.006	0.047	0.008
10.071	0.007	0.057	0.010
8.877	0.010	0.064	0.007
7.850	0.014	0.068	0.004
7.423	0.016	0.075	0.007
6.833	0.020	0.085	0.010
6.684	0.022	0.097	0.012

Tabela 3.22 Linhas e colunas embutidas na réplica. R3.F01

Valor F	NSL	RSL	Frequência
31.500	0.001	0.016	0.016
13.669	0.008	0.024	0.008
10.894	0.013	0.028	0.004
7.974	0.026	0.037	0.009
7.461	0.029	0.045	0.009
6.775	0.035	0.056	0.011
6.324	0.040	0.063	0.007
5.505	0.051	0.071	0.008
4.816	0.064	0.080	0.009
4.534	0.071	0.086	0.006
4.148	0.081	0.095	0.009
4.051	0.084	0.103	0.008

seja, não rejeição da hipótese nula. Assim, usando-se alguns desses métodos de réplica poderia-se chegar a conclusões erradas.

Um fato importante é que com um número menor de réplicas, neste caso oito, pode-se ter aproximações boas, tal como se observa nas Tabelas 3.20 e 3.22. Observa-se também que nas quatro tabelas (3.19 a 3.22) os níveis de significância são baixos, dando para rejeitar a hipótese nula a um nível de $\alpha = 5\%$.

FASE 02

Utilizando oito réplicas para analisar o esforço nas especificações de manutenção de uma linha de produto, fase 02, tem-se como resultado a Figura 4.14, onde observa-se um comportamento estranho, pois as discrepâncias entre os testes exato e F são enormes, tanto para o caso de linhas e colunas cruzadas como para colunas embutidas nas réplicas, onde não é possível fazer uso do teste F , pois como se observa nas Tabelas 3.23 e 3.24, a um nível de $\alpha = 5\%$, não se rejeitaria a hipótese nula quando em realidade a hipótese deveria ser rejeitada.

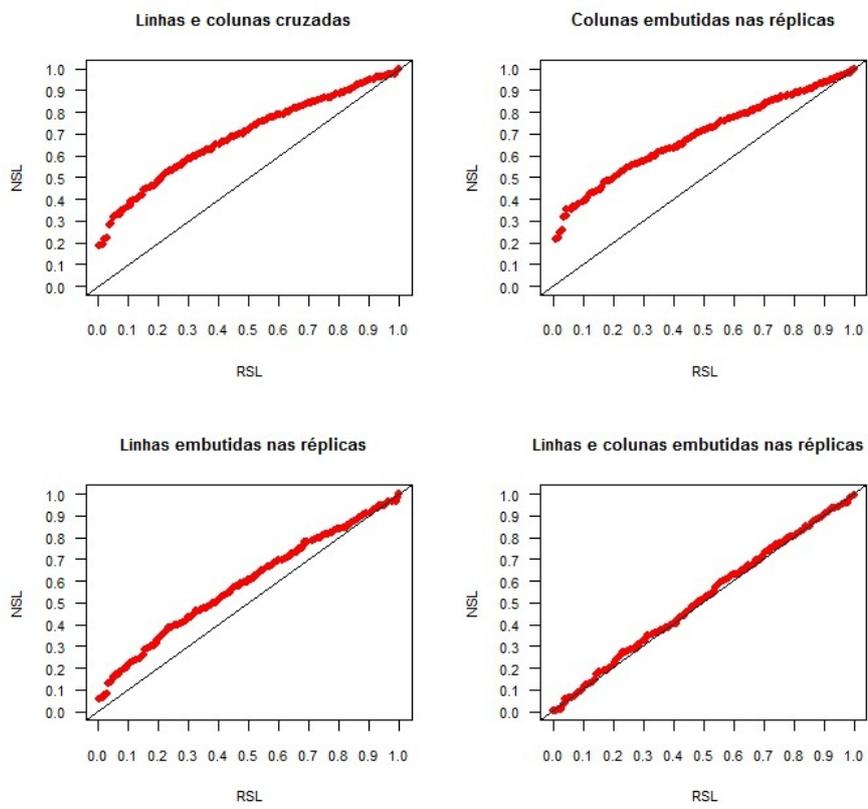


Figura 4.14: Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 02

Tabela 3.23: Linhas e colunas cruzadas. R3.F02

Valor F	NSL	RSL	Frequência
1.878	0.185	0.005	0.005
1.825	0.191	0.015	0.010
1.632	0.215	0.023	0.008
1.583	0.222	0.031	0.007
1.224	0.281	0.039	0.008
1.183	0.289	0.043	0.005
1.041	0.319	0.053	0.010
1.035	0.321	0.056	0.003
1.004	0.328	0.063	0.007
0.997	0.329	0.073	0.010
0.936	0.344	0.080	0.007
0.901	0.353	0.088	0.008

Tabela 3.24: Colunas embutidas na réplica. R3.F02

Valor F	NSL	RSL	Frequência
1.683	0.216	0.013	0.013
1.634	0.222	0.017	0.005
1.457	0.247	0.025	0.007
1.412	0.254	0.031	0.007
1.086	0.315	0.039	0.007
1.049	0.323	0.046	0.007
0.921	0.353	0.046	0.008
0.915	0.355	0.061	0.007
0.888	0.362	0.069	0.008
0.882	0.364	0.075	0.007
0.827	0.379	0.085	0.010
0.795	0.388	0.101	0.015

Tabela 3.25: Linhas embutidas na réplica. R3.F02

Valor F	NSL	RSL	Frequência
4.334	0.056	0.005	0.005
4.187	0.060	0.015	0.011
3.668	0.076	0.023	0.008
3.540	0.081	0.031	0.008
2.639	0.127	0.036	0.005
2.540	0.133	0.044	0.008
2.204	0.160	0.052	0.008
2.189	0.161	0.056	0.004
2.117	0.168	0.061	0.005
2.102	0.169	0.069	0.008
1.961	0.183	0.075	0.006
1.881	0.192	0.086	0.011

Tabela 3.26 Linhas e colunas embutidas na réplica. R3.F02

Valor F	NSL	RSL	Frequência
18.888	0.003	0.004	0.004
17.185	0.004	0.012	0.008
12.487	0.010	0.021	0.009
11.565	0.011	0.030	0.009
6.712	0.036	0.037	0.007
6.306	0.040	0.041	0.005
5.065	0.059	0.041	0.006
5.013	0.060	0.055	0.008
4.773	0.065	0.062	0.007
4.724	0.066	0.069	0.007
4.276	0.077	0.077	0.007
4.033	0.085	0.085	0.008

Essas são as consequências de usar o teste F quando se tem um número de tratamentos pequeno e um número de réplicas não adequado. No entanto, para o caso de linhas e colunas embutidas nas réplicas, curiosamente, a aproximação é muito boa, são praticamente iguais tal como se pode ver na Tabela 3.26. Fazendo um zoom na área de 0.0 até 0.1 da Figura 4.15, observa-se que para o caso de linhas embutidas nas réplicas surge a situação em que analisando o teste F não se tem evidência suficiente para rejeitar H_0 , mas por outro lado, o teste exato rejeita a hipótese nula para estes mesmos valores, assim, é possível obter conclusões erradas ou facilmente não poder obter conclusão alguma, devido às contradições provenientes das análises. Cabe ressaltar que para linhas e colunas embutidas nas réplicas o comportamento do teste F ao teste exato é muito parecido e consistente, no sentido de que se obtém as mesmas conclusões analisando qualquer um dos testes empregados.

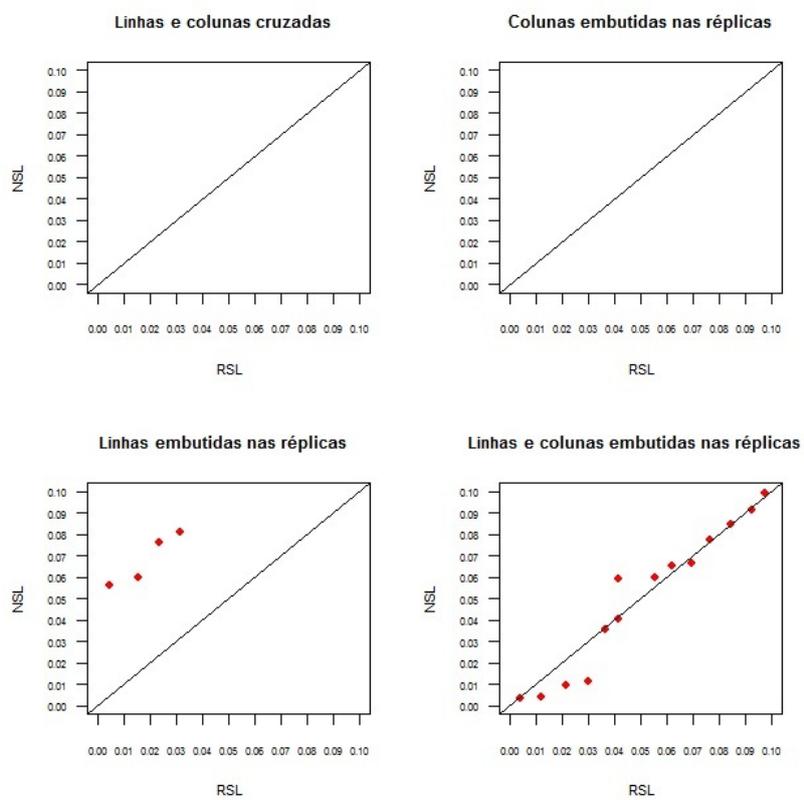


Figura 4.15: Quadrados latinos 2×2 com 8 réplicas. Rodada 3. Fase 02

CAPÍTULO 5

Cálculo da função poder do teste F

Neste capítulo é realizado o cálculo do teste exato para os dados fornecidos pelo autor de [2]. Como pode ser observado a Tabela 5.1 apresenta os valores da estimativa do erro experimental, o valor da estatística F, o nível de significância nominal (NSL), ou seja, o nível de significância para o teste F, e por último, o nível de significância aleatório (RSL), ou seja, o nível de significância para o teste exato. Este valor RSL é uma das contribuições desta dissertação cujo código em SAS para o cálculo dele encontra-se no Apêndice B.

Analisando os dados obtidos neste trabalho e comparando-os com os trabalhos em [2], tem-se que para a primeira rodada do experimento (Tabela 5.1) na fase 01 de extração de linhas de produtos de software, os valores dos dois níveis de significância estão próximos, dando como resultado que, a um nível de 5%, não se encontra evidência estatística suficiente para rejeitar a hipótese nula de igualdade nos tempos médios de extração para cada tratamento. A mesma conclusão é obtida para a fase de manutenção das linhas de produtos de software, mas tendo em conta que os níveis de significância NSL e RSL diferem notavelmente, embora não cheguem a conclusões contraditórias.

Tabela 5.1: Níveis de significância do teste F e do teste exato

	$\hat{\sigma}_e^2$	Teste F	NSL	RSL
Rodada 1				
Fase 01	116.17	4.42	0.0619	0.0625
Fase 02	119.12	1.88	0.2071	0.125
Rodada 2				
Fase 01	205.47	18.40	0.0004	0.0039
Fase 02	212.54	1.38	0.2598	0.2734
Rodada 3				
Fase 01	112.29	21.64	0.0009	0.0156
Fase 02	101.10	3.76	0.0812	0.0234

Para o caso da rodada 2, nota-se que os valores dos níveis de significância são muito próximos, isto é facilmente justificado pelos resultados achados no estudo de simulação. Tal como se pode observar na Figura 4.8 e Figura 4.10, esta última especialmente, a aproximação dos dois testes é muito boa para o caso de linhas embutidas nas réplicas, método usado em [2]. Cabe ressaltar que para a fase de extração, os dois testes concordam em rejeitar a hipótese nula, dessa forma, evidência-se a existência de diferença dos tempos médios dos tratamentos, além de ser a fase que apresentava maior variância do erro experimental.

Em contrapartida, nos valores obtidos para a rodada 3 (Tabela 5.1), os níveis de significância afastam-se entre si. Para a primeira fase, apesar de não serem tão próximos, os níveis fornecem a mesma conclusão de rejeitar a hipótese de igualdade dos tempos médios dos tratamentos. No entanto, para o caso da fase 2 de manutenção de linhas de produtos de software, acontece uma contradição nas conclusões obtidas para cada um dos testes ao nível de significância de 5%, no mais, NSL não rejeita a hipótese nula, mas por outro lado RSL sugere o contrario, rejeitar H_0 . Vale ressaltar que nesta fase o erro experimental apresentou o menor valor de variabilidade e o número de

réplicas efetuadas foi de oito.

Para complementar as análises feitas até agora, apresenta-se um estudo da função poder do teste F , com a finalidade de indicar o número de réplicas apropriado para valores diferentes de $\hat{\sigma}_e^2$ contemplando diversos valores para a diferença nas médias de tempo entre tratamentos.

O conceito de poder baseado em [5] define a função poder de um teste de hipótese com região de rejeição $A1$, para testar $H_0 : \theta = \theta_0$ contra $H_1 : \theta = \theta_1$, como uma função de θ expressada da seguinte forma,

$$\beta(\theta) = P_\theta(X \in A1)$$

O ideal é que a função poder seja 0 para todo $\theta \in \Theta_0$ e 1 para todo $\theta \in \Theta_0^c$. Exceto em situações triviais, este ideal não pode ser atendido. Qualitativamente, um bom teste tem a função de poder próximo a 1 para a maioria dos $\theta \in \Theta_0^c$ e próximo de 0 para a maioria dos $\theta \in \Theta_0$.

A análise do poder é geralmente usada em uma das duas seguintes maneiras, a determinação do poder do teste ou a determinação do tamanho da amostra necessária para atingir um certo poder. Um dos aspectos importantes em execução de experimentos é o custo. Em geral, quanto maior tamanho do uso de UE, mais custoso é a execução do experimento, portanto, é importante conhecer o número mínimo do tamanho de amostra necessários para alcançar um valor alto do poder ou um valor pré-especificado.

Outro ponto importante na análise do poder do teste, são os parâmetros necessários para conseguir efetuar o cálculo. Neste trabalho, os parâmetros são fixados com base nos dados obtidos em [2]. O primeiro a ser definido foi α , isto é, a probabilidade de rejeitar a hipótese nula dada que ela é verdadeira. Para este estudo, considerou-se $\alpha = 5\%$. O segundo parâmetro se refere ao desvio padrão do erro experimental. Para isto, aproveitou-se os dados calculados em [2] e tomou-se um intervalo de desvios padrão que abrange os valores mínimo, máximo e alguns valores intermédios, neste caso, $\hat{\sigma}_e^2 = 10.055, 11.402, 12.649, 14.579$. No terceiro parâmetro, definiu-se uma sequência de valores para os diferentes tamanhos de réplica, neste caso

$n = 5, 8, 10, 15, 20$. Por último, considerou-se o tamanho do efeito δ , que é como uma pequena ou grande diferença em relação à hipótese nula que se quer detectar. Em cada um dos gráficos apresentados na Figura 5.1 usou-se diferentes valores de tamanhos de efeitos, $\delta = 5, 10, 15, 20$ os quais são representados pelos valores obtidos em [2], assim, tem-se 60 – 65 para um tamanho de 5 e 40 – 50, 85 – 100 e 70 – 90 para tamanhos de 10, 15 e 20, respectivamente, tal como se mostra na legenda da Figura 5.1.

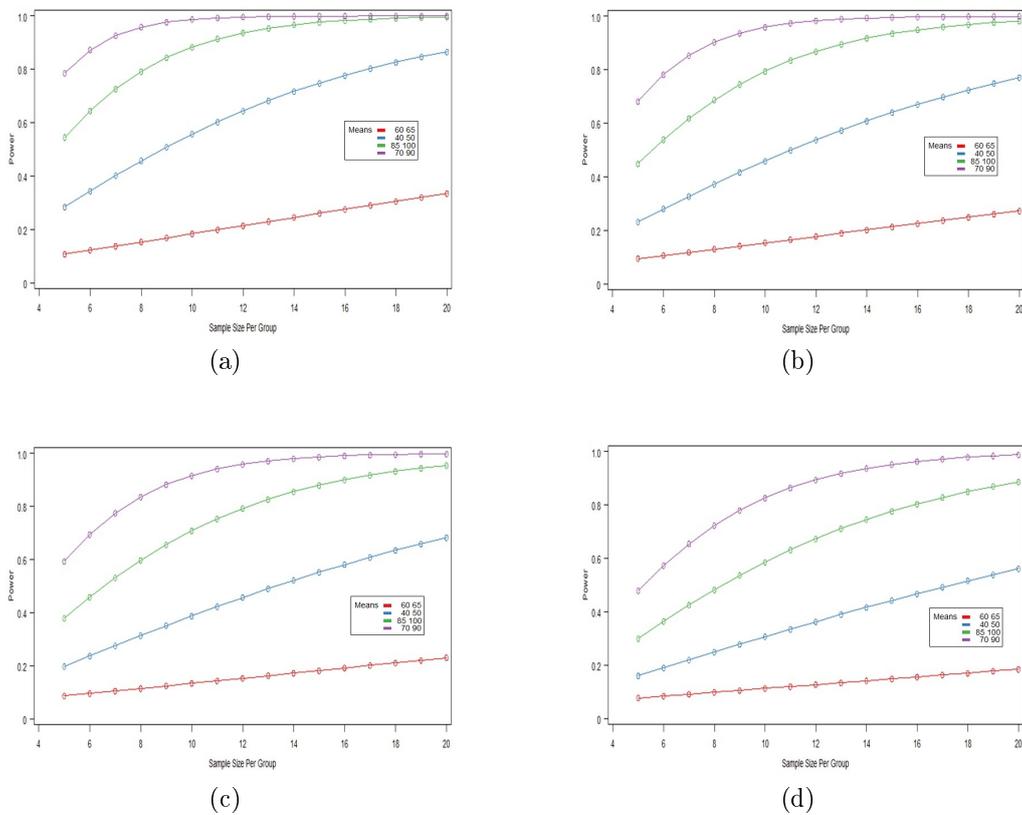


Figura 5.1: Gráficos da função de poder para diferentes valores de σ_e^2

Na Figura 5.1 são apresentadas as possíveis combinações dos valores mencionados no parágrafo anterior, onde cada uma das sub-figuras apresenta o valor do poder do teste F para um valor de desvio padrão do erro experimental específico e os diferentes valores de número de réplicas. Cada curva representa uma das diferenças, em médias, dos efeitos de tratamento. As-

sim, para o primeiro caso, Figura 5.1a, em que o valor do desvio padrão é $\hat{\sigma}_e = 10.055$, como é esperado, o menor valor de δ implica menor poder, e caso contrário, o maior valor de tamanho do efeito implica maior poder do teste. Isto é um comportamento constante nas quatro sub-figuras, pelo qual tem-se a mesma ordem das curvas. Voltando à Figura 5.1a, um poder de 80% só é possível para diferenças maiores que 5, assim, para um valor de $\delta = 10$ precisa-se o maior número de réplicas, praticamente 20, mas a medida que esse valor de δ aumenta para 15, precisaria-se de oito réplicas e para um valor de $\delta = 20$ um número de cinco réplicas. Cabe ressaltar que este desvio padrão para o erro experimental foi obtido na rodada 3 da fase 02, na qual determinou-se o número de réplicas para conseguir um poder do teste de 80% para uma diferença de 15, mas devido à uma má escolha da forma de replicar quadrados latinos em [2], chegou-se a uma contradição no cálculo do teste exato.

Para as seguintes três Figuras 5.1b, 5.1c e 5.1d, só é possível alcançar um poder de 80% para valores de $\delta = 15, 20$, isto devido em parte ao fato de que se tem valores maiores do desvio padrão do erro experimental, $\hat{\sigma}_e = 11.402, 12.649, 14.579$ respectivamente. Das sub-figuras 5.1b, 5.1c entre oito e doze réplicas chega-se a um poder de teste próximo do 80%, levando em consideração que no caso do tamanho da amostra, seria preciso muito mais amostras para detectar uma diferença de efeitos pequena. No caso 5.1d, praticamente só se tem um bom poder para uma diferença de vinte e um número de réplicas superior a onze. Portanto, se o desvio padrão é pequeno, resultará um poder maior do que se o desvio padrão for grande, e por outro lado, se o desvio padrão é grande, então o poder será reduzido. Assim, um pequeno desvio-padrão necessitará de menos amostras para alcançar o mesmo poder que um grande desvio padrão.

CAPÍTULO 6

Conclusões e recomendações

Com os estudos de simulação de Monte Carlo feitos neste trabalho, conclui-se que a forma de replicação em que melhor se aproxima o teste F para o teste exato é linhas e colunas embutidas nas réplicas, mas é preciso considerar que esta seria uma das formas mais custosas de se obter as réplicas. Cabe ressaltar que outras opções de replicação apresentam comportamentos bons na aproximação dos testes. Assim, para a fase 01 de extração de linhas de produtos de software, as colunas embutidas nas réplicas foi uma forma aceitável de realizar replicação, considerando a qualidade da aproximação do teste F para o exato. Para a fase 02 de manutenção de uma linha de produção de software, as linhas embutidas nas réplicas apresentam um bom comportamento na qualidade da aproximação.

Um aspecto complementar e importante é que a melhor forma de replicar quadrados latinos é através de linhas e colunas embutidas nas réplicas. No entanto, uma questão importante é de quantas réplicas se precisa para realizar um processo de experimentação. Para isso, é importante estabelecer junto com os especialistas na área as diferenças que se desejam detectar por meio da estatística do teste, para assim, efetuar uma análise do poder. Neste

ponto, conclui-se que para um número razoável de réplicas poderiam detectar diferenças entre as médias dos valores dos efeitos. Para $\Delta = 15$ e $\Delta = 20$ minutos e valores de $\hat{\sigma}_e^2 < 150$, este número razoável de réplicas seria de $8 \leq r \leq 12$. Já para valores de $\hat{\sigma}_e^2 \approx 200$, diferenças são possíveis de detectar só com um número de réplicas alto, $r > 10$ para valores de $\Delta = 20$ e $r > 16$ para valores de $\Delta = 15$.

Para trabalhos futuros cujo objetivo seja comprovar as vantagens do uso da abordagem MSVCM, especialmente na manutenção de linhas de produção, recomenda-se ter em conta o número de requerimentos solicitados para chegar ao ponto de quebra em que MSVCM deixa de ser uma abordagem custosa, tornando-se vantajoso o seu uso, em semelhança à discussão acerca das vantagens ao empregar linhas de produção de software frente a sistemas convencionais, os quais num ponto inicial são sistemas econômicos mas que a partir de mais de três sistemas terminam sendo mais custosos que as linhas de produção de software, tal como se apresenta na figura 6.1.

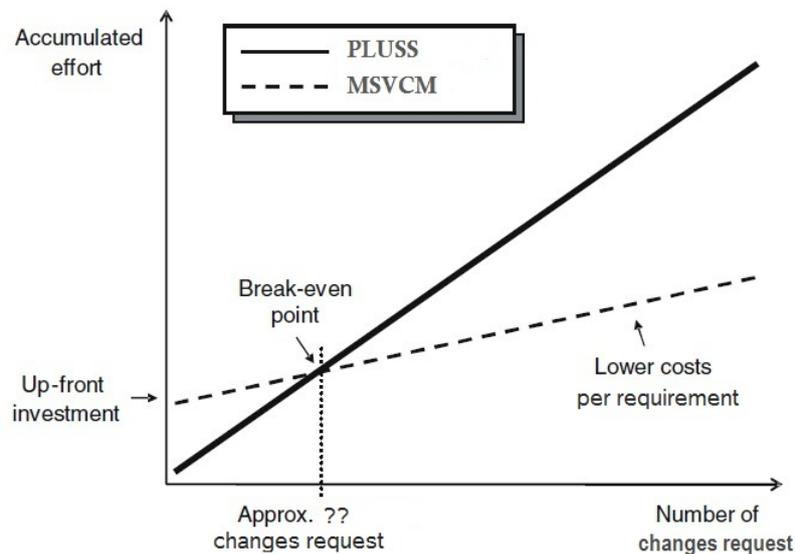


Figura 6.1: Relação de custos entre PLUSS e MSVCM

Por outro lado, é recomendável explorar, se for possível, planos experimentais mais complexos. Se é possível contar com um número razoável de

analistas, o plano aleatorizado em retângulos latinos seria uma opção, já que poderia ser obtido mais de dois níveis no fator linha, que representa os analistas empregados no experimento. Tendo em conta que não é viável empregar um analista para testar abordagens em mais de dois produtos de software, o fator coluna é restringindo a dois níveis.

Por último, um comentário para ressaltar a rigorosidade e organização com que foram realizados os experimentos em [2], o que permitiu realizar um estudo posterior com ferramentas e informação de ótima qualidade.

Bibliografia

- [1] Balasubramani, G.K., Wisniewski, S., Zhang, H. & Heather, F. Development of an efficient SAS[®] macro to perform permutation tests for two independent samples. *Computer Methods and Programs in Biomedicine*, n. 79, p. 179-187, 2005.
- [2] Bonifácio,R. (2010). *Modeling Software Product Line Variability in Use Case Scenarios*. Tese doutorado em ciência da computação UFPE.
- [3] Bonifácio,R. Borba, P. & Ferraz, C. (2010). *Assessing Software Product Line Approaches for Specifying Use Case Scenarios*. Artigo em revisão.
- [4] Casella, G. & Berger, R.L. (2008). *Statistical Design*. First Edition. Springer.
- [5] Casella, G. & Berger, R.L. (2002). *Statistical Inference*. Second Edition. Duxbury.
- [6] Dean,A & Voss,D. (1999). *Design and Analysis of Experiments*.Springer.
- [7] Dudewicz, E. & Mishra, S. (1998). *Modern Mathematical Statistics*. First Edition. Wiley.

- [8] Dubcovsky, J., Bottoms, T. & Murphy, S. (2010). *Design, Analysis, and Interpretation of Experiments*. Department of plant sciences. UC Davis. <http://www.plantsciences.ucdavis.edu/agr205/Lectures/2010%20Iago/Topic%207/Topic%207%20Reading.pdf>
- [9] Generating Combinations and Permutations. (1999). SAS Communications. TS Docs. <http://support.sas.com/techsup/technote/ts498.html>
- [10] Hinkelmann, K. (2001). Remembering Oscar Kempthorne. *Statistical Science*, Vol. 16, No. 2, 169 - 183.
- [11] Kempthorne, O. & Hinkelmann, K. (1994). *Design and Analysis of Experiments*. Vol. I: Introduction to Experimental Design. Wiley-Interscience.
- [12] Osmena, P. (Marzo 2010). *Statistical Power Analysis Using SAS and R* A Senior Project Presented to The Faculty of the Statistics Department California Polytechnic State University, San Luis Obispo. <http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1002&context=statsp>
- [13] SAS/STAT(R) 9.22 User's Guide. (2007). SAS Publishing. Second Edition http://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_anova_sect029.html
- [14] Sommerville, I. (2004). *Engenharia de Software*. 6ª edição. Addison Wesley.
- [15] Van der Linden, F., Schmid, K. & Rommes, E. (2007). *Software Product Lines in Action*. The best industrial practice in product line engineering. 1ª edição. Springer.
- [16] Vasconcelos, A. (Fevereiro 2010). *Tópicos em engenharia de software e sistemas* <http://www.cin.ufpe.br/~if682/>.

Apêndice A: código SAS para o calculo do teste exato de um plano em quadrado latino 2x2

```
/*
*****
UNIVERSIDADE FEDERAL DE PERNAMBUCO
AUTOR: IVAN HERRERA SANCHEZ
DATA DE CRIAÇÃO: 09/2010
ULTIMA MODIFICAÇÃO: 12/2010
CALCULO DO TESTE EXATO PARA UM PLANO EM QUADRADOS LATINOS
*****
*/
proc factex; /* procedimento para gerar os possiveis valores de 2^r*/
    factors rep1 rep2 rep3 rep4 rep5;
    output out=Factorial
rep1 nvals=(1 2)
rep2 nvals=(1 2)
rep3 nvals=(1 2)
rep4 nvals=(1 2)
rep5 nvals=(1 2);
run;
/* declaração de variáveis globais */
%let tabla=factorial;
%let var1=rep1;
%let var2=rep2;
%let var3=rep3;
%let var4=rep4;
%let var5=rep5;
/*macro para transposição das filas do data gerado pelo proc factex*/
%macro trasponeperm;
data paso1;set &tabla.;ind=_n_;run;
proc sql noprint;select max(ind) into:maxfact from paso1;quit;
%let maxfact=%eval(&maxfact*1);
%do i=1 %to &maxfact;
data paso2;set paso1;if ind=&i;run;
proc transpose data=paso2 out=paso3;var &var1. &var2. &var3. &var4. &var5.;
run;
%if &i=1 %then %do;
data salida;set paso3;corrida=&i;run;
%end;
%if &i > 1 %then %do;
data salida1;set paso3;corrida=&i;run;
proc append base=salida data=salida1 force;run;
%end;
%end;
%mend trasponeperm;
%trasponeperm; /* execução da macro trasponeperm*/
/* macro para gerar as possiveis combinações de um quadrado latino 2x2*/
```

```

%macro combinaciones;
data paso1;set salida;ind=_n_;run;
proc sql noprint;select max(ind) into:maxfact from paso1;quit;
%let maxfact=%eval(&maxfact*1);
%do i=1 %to &maxfact;
data _null_;
set paso1;
if ind=&i then do;                                /* inclusion */
call symput ('varcol1', col1);
call symput ('replica',_NAME_);
call symput ('corrida',corrida);
end;                                              /* inclusion */
run;
%if &varcol1=1 %then
%do;
proc plan seed=123; /* gera um dos dois quadrados latinos possiveis */
factors row=2 ordered
col=2 ordered / noprint;
treatments treat=2 cyclic;
output out=OutFromProcPlan
Row nvals=(1 2) random
Col nvals=(1 2) random
treat nvals=(1 2) random;

run;
proc sort data=OutFromProcPlan out=OutFromProcPlan;
by Row Col;
run;
%end;
%else %if &varcol1=2 %then
%do;
proc plan seed=1234; /* gera o outro dos dois quadrados latinos*/
factors row=2 ordered
col=2 ordered / noprint;
treatments treat=2 cyclic;
output out=OutFromProcPlan
Row nvals=(1 2) random
Col nvals=(1 2) random
treat nvals=(1 2) random;

run;
proc sort data=OutFromProcPlan out=OutFromProcPlan;
by Row Col;
run;
%end;
data OutFromProcPlan;set OutFromProcPlan;
format itera best8.0;
rep="&replica.";
itera=&corrida.;
run; /**/
proc append base=combinaciones data=OutFromProcPlan force;

```

```

run;
%end;
%mend combinaciones;
%combinaciones; /* execução da macro combinaciones*/
data combinaciones;set combinaciones;rep1=rep;drop rep;run;
data combinaciones;set combinaciones;
format rep best8.0;
rep=substr(rep1,4,1)*1;
drop rep1;
run;
proc sql noprint;select max(itera) into:niter from combinaciones;
quit;
%let niter=%eval(&niter*1);
/* criação do data fixando as variáveis resposta a cada UE*/
data diseno;set combinaciones;
if row=1 and col=1 and rep=1 then y=55;
if row=1 and col=2 and rep=1 then y=83;
if row=2 and col=1 and rep=1 then y=50;
if row=2 and col=2 and rep=1 then y=55;
if row=1 and col=1 and rep=2 then y=60;
if row=1 and col=2 and rep=2 then y=76;
if row=2 and col=1 and rep=2 then y=82;
if row=2 and col=2 and rep=2 then y=57;
if row=1 and col=1 and rep=3 then y=61;
if row=1 and col=2 and rep=3 then y=46;
if row=2 and col=1 and rep=3 then y=69;
if row=2 and col=2 and rep=3 then y=40;
if row=1 and col=1 and rep=4 then y=55;
if row=1 and col=2 and rep=4 then y=40;
if row=2 and col=1 and rep=4 then y=70;
if row=2 and col=2 and rep=4 then y=60;
if row=1 and col=1 and rep=5 then y=63;
if row=1 and col=2 and rep=5 then y=42;
if row=2 and col=1 and rep=5 then y=60;
if row=2 and col=2 and rep=5 then y=60;
run;
/* macro para a estimação do modelo linear para um quadrado latino
e calculo do teste aproximado F*/
%macro modeloglm(niter);
%put niter= &niter.;
%do i=1 %to &niter.;
data modelo1;set diseno;if itera=&i.;
run;
proc glm data=modelo1 outstat=tabla1 noprint;
class rep col row treat;
model y = rep col row(rep) treat;
run;
%if &i=1 %then
%do;

```

```

data tabla;set tabla1;itera=&i;run;
%end;
%if &i>1 %then
%do;
data tabla1;set tabla1;itera=&i;run;
proc append base=tabla data=tabla1;run;
%end;
proc delete data=tabla1;run;
%end;
%mend modeloglm;
%modeloglm(&niter.); /*execução da macro modeloglm para cada uma dos
    possíveis arranjos de quadrados latinos 2x2 com réplicas*/
/* macro para o calculo do teste exato */
%macro exato(niter);
%put niter= &niter.;
data tabla1;set tabla;
if _type_='SS3' and _source_='treat';keep itera prob f;
run;
%do i=1 %to &niter.;
data _null_;set tabla1;
if itera=&i. then call symput ('valf',f);
run;
data exato;set tabla1;
valf=round(&valf.*1,0.0000000001);
f=round(f*1,0.0000000001);
if f<valf then var1="menores";
if f>=valf then var1="mayor_igual";
run;
%if &i=1 %then
%do;
data rsl;set exato;itera=&i;run;
%end;
%if &i>1 %then
%do;
data exato;set exato;itera=&i;
run;
proc append base=rsl data=exato;
run;
%end;
%end;
proc delete data=exato;run;
%mend exato;
%exato(&niter.); /*execução da macro exato para cada uma dos
    possíveis arranjos de quadrados latinos 2x2 com réplicas*/
proc sql; /* edição do resultado do teste exato */
create table consulta as select
itera, var1,count(*) as numt
from rsl
where var1='mayor_i'

```

```

group by itera,var1;
quit;
data simulacion; /* edição da data com os resultados*/
merge tabla1 consulta;
by itera;
exato=numt/&niter.;
exato=round(exato*1,0.00000001);
keep itera f prob exato;
run;
/* procedimento para graficar o teste exato e aproximado F */
symbol v=dot;
proc gplot data=simulacion;
    plot prob*exato;
run;
quit;
/* procedimento para exportar a data com todos os resultados */
proc export data=simulacion
outfile='E:\UFPE\Dissertação\resultados\filas_embutidas_exato
\rodada1\fase01_5rep'
dbms=tab;
run;

```

Apêndice B: código SAS para simulação das aproximações do teste exato e o teste F em um plano em quadrado latino 2x2

```

/*
*****
UNIVERSIDADE FEDERAL DE PERNAMBUCO
AUTOR: IVAN HERRERA SANCHEZ
DATA DE CRIAÇÃO: 09/2010
ULTIMA MODIFICAÇÃO: 12/2010
CALCULO DO TESTE EXATO PARA UM PLANO EM QUADRADOS LATINOS
*****
*/
%global niter; /*declaração de variaveis globais*/
%global nrep;
%let niter=1500;
%let nrep=5;
data combinaciones;
input rep row col treat itera;
cards;
run;
/* macro para gerar arranjos de um quadrado latino 2x2*/
%macro simulacion(niter,nrep);

```

```

%do i=1 %to &niter.;
%do j=1 %to &nrep.;
data _null_;
  x=round(ranuni(0)*1000000); /*geração da semente*/
call symput ('seed', x);
run;
  proc plan seed=&seed.;
    factors row=2 ordered
    col=2 ordered / noprint;
    treatments treat=2 cyclic;
    output out=OutFromProcPlan
      Row nvals=(1 2) random
      Col nvals=(1 2) random
      treat nvals=(1 2) random;

  run;
  proc sort data=OutFromProcPlan out=OutFromProcPlan;
    by Row Col;
  run;
data OutFromProcPlan;set OutFromProcPlan;
rep=%eval(&j*1);
itera=%eval(&i*1);
run;
proc append base=combinaciones
data=OutFromProcPlan force;
run;
%end;
%end;
%mend simulacion;
%simulacion(&niter.,&nrep.); /* execução da macro simulacion */
/* criação do data fixando as variavéis resposta a cada UE*/
data diseno;set combinaciones;
if row=1 and col=1 and rep=1 then y=55;
if row=1 and col=2 and rep=1 then y=83;
if row=2 and col=1 and rep=1 then y=50;
if row=2 and col=2 and rep=1 then y=55;
if row=1 and col=1 and rep=2 then y=60;
if row=1 and col=2 and rep=2 then y=76;
if row=2 and col=1 and rep=2 then y=82;
if row=2 and col=2 and rep=2 then y=57;
if row=1 and col=1 and rep=3 then y=61;
if row=1 and col=2 and rep=3 then y=46;
if row=2 and col=1 and rep=3 then y=69;
if row=2 and col=2 and rep=3 then y=40;
if row=1 and col=1 and rep=4 then y=55;
if row=1 and col=2 and rep=4 then y=40;
if row=2 and col=1 and rep=4 then y=70;
if row=2 and col=2 and rep=4 then y=60;
if row=1 and col=1 and rep=5 then y=63;
if row=1 and col=2 and rep=5 then y=42;

```

```

if row=2 and col=1 and rep=5 then y=60;
if row=2 and col=2 and rep=5 then y=60;
run;
/* macro para a estimação do modelo linear para um quadrado latino
e calculo do teste F*/
%macro modeloglm(niter);
%put niter= &niter.;
%do i=1 %to &niter.;
data modelo1;set diseno;if itera=&i.;
run;
proc glm data=modelo1 outstat=tabla1 noprint;
class rep col row treat;
model y = rep col row treat;
run;
%if &i=1 %then
%do;
data tabla;set tabla1;itera=&i;run;
%end;
%if &i>1 %then
%do;
data tabla1;set tabla1;itera=&i;run;
proc append base=tabla data=tabla1;run;
%end;
proc delete data=tabla1;run;
%end;
%mend modeloglm;
%modeloglm(&niter.); /*execução da macro modeloglm para cada uma dos
possíveis arranjos de quadrados latinos 2x2 com réplicas*/
/* macro para o calculo do teste exato */
%macro exato(niter);
%put niter= &niter.;
data tabla1;set tabla;
if _type_='SS3' and _source_='treat';keep itera prob f;
run;
%do i=1 %to &niter.;
data _null_;set tabla1;
if itera=&i. then call symput ('valf',f);
run;
data exato;set tabla1;
valf=round(&valf.*1,0.000000001);
f=round(f*1,0.000000001);
if f<valf then var1="menores";
if f>=valf then var1="mayor_igual";
run;
%if &i=1 %then
%do;
data rsl;set exato;itera=&i;run;
%end;
%if &i>1 %then

```

```

%do;
data exato;set exato;itera=&i;run;
proc append base=rsl data=exato;run;
%end;
%end;
proc delete data=exato;run;
%mend exato;
%exato(&niter.); /*execução da macro exato para cada uma dos
possíveis arranjos de quadrados latinos 2x2 com réplicas*/
proc sql; /* edição do resultado do teste exato */
create table consulta as select
itera, var1,count(*) as numt
from rsl
where var1='mayor_i'
group by itera,var1;
quit;
data simulacion; /* edição da data com os resultados*/
merge tabla1 consulta;
by itera;
exato=numt/&niter.;
exato=round(exato*1,0.00000001);
keep itera f prob exato;
run;
/* procedimento para graficar o teste exato e aproximado F */
symbol v=dot;
proc gplot data=simulacion;
plot prob*exato;
run;
quit;
/* procedimento para calcular frequências */
ods output frequencies=frequencies;
proc univariate data=simulacion freq;
var F;
run;
proc sort data=simulacion out=simula1 nodupkey;
by f;run;
proc sort data=frequencies out=frequencies1 nodupkey;
by Value;run;
data frecuencias;
merge simula1 frequencies1;
keep F PROB itera exato Count;
run;
/* procedimento para exportar a data com todos os resultados */
proc export data=simulacion
outfile='E:\UFPE\Dissertação\resultados\cruzados\rodada1
\fase01_5rep'
dbms=tab;
run;
proc export data=frecuencias

```

```
outfile='E:\UFPE\Dissertação\resultados\cruzados\rodada1  
\freq_fase01_5rep'  
dbms=tab;  
run;
```