UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI FACULDADE DE CIÊNCIAS AGRÁRIAS

CURSO DE ENGENHARIA FLORESTAL

**SCRIPT PARA PROCESSAMENTO DE INVENTÁRIO FLORESTAL UTILIZANDO LINGUAGEM R**

**Sollano Rabelo Braga**

Diamantina – MG

2017

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI FACULDADE DE CIÊNCIAS AGRÁRIAS

**SCRIPT PARA PROCESSAMENTO DE INVENTÁRIO FLORESTAL UTILIZANDO LINGUAGEM R**

**Sollano Rabelo Braga**

Orientador: **Dr. Marcio Leles Romarco de Oliveira**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Florestal, como parte dos requisitos exigidos para a conclusão do curso.

Diamantina – MG

2017

**SCRIPT PARA PROCESSAMENTO DE INVENTÁRIO FLORESTAL UTILIZANDO LINGUAGEM R**

**Sollano Rabelo Braga**

Orientador: **Dr. Marcio Leles Romarco de Oliveira**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Florestal, como parte dos requisitos exigidos para a conclusão do curso.

APROVADO em ... / ... / ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. Eric Bastos Gorgens – UFVJM

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eng. Florestal Lidia Gabriela Sousa - UFVJM

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. Marcio Leles Romarco de Oliveira - UFVJM

***Dedico***

*À minha mãe Leana Maria Rabelo,*

*Ao meu pai Militão Sampaio Braga.*

**AGRADECIMENTOS**

Agradeço à minha mãe Leana Maria Rabelo e ao meu pai Militão Sampaio Braga, pelo apoio e amor recebidos.

Aos professores do Departamento de Engenharia Florestal, em especial o professor Márcio Leles Romarco de Oliveira, pela orientação.

Aos membros do laboratório de Mensuração e Manejo Florestal, pelo companheirismo e apoio.

Aos membros do site Stackoverflow pela ajuda.

À Universidade Federal dos Vales do Jequitinhonha e Mucuri pela oportunidade oferecida.

**RESUMO**

Informações necessárias ao manejo e planejamento de povoamentos, que normalmente são referentes ao volume de madeira e/ou outra variável dendrométrica, no sentido qualitativo e quantitativo, são obtidas por meio do inventário florestal. Para o processamento dos dados de inventário, como dados de cubagem ou amostragem, muitas empresas optam por softwares de edição de planilhas, como o Microsoft Excel, LibreOffice, softwares comerciais específicos, ou até mesmo algumas empresas desenvolvem seu próprio software. No caso das planilhas eletrônicas, por não serem softwares estatísticos, podem levar a imprecisões nas análises. Com isso este trabalho teve como objetivo criar scripts que possibilitem o processamento de inventário utilizando *R,* uma poderosa linguagem e ambiente de programação estatística e gráfica. O script criado pode processar dados de cubagem utilizando o método de Smalian, e processar dados de amostragem utilizando os métodos de Amostragem Casual Simples, Amostragem Casual Estratificada e Amostragem Sistemática. Sua validação foi feita comparando seus resultados com exemplos apresentados em literatura específica, além de utilizar uma base de dados real, sendo estes resultados comparados com os obtidos por meio do Microsoft Excel. Os scripts produzidos foram eficientes no processamento de inventário florestal quando comparado com es exemplos testados e validados.

**Palavras – chave:** Mensuração, Processamento de Dados, Cubagem.

**ABSTRACT**

Information necessary to management and planning of forest population that are normally referred to timber volume or another dendrometric variable, be it quantitative or qualitative, are obtained through forest inventory. To process forest inventory data, most companies use spreadsheet editors like Microsoft Excel or LibreOffice, specific commercial software, or software created by the own company. Regarding electronic spreadsheets, since they are not statistic-based software, its analysis may be biased or imprecise. This study was developed with the purpose of creating a script to process forest inventory data utilizing *R,* a powerful language and environment for statistical computing and graphics. Cubic volume can be determined using the script by Smalian’s method, and inventory data can be processed using Simple Random Sampling, Stratified Sampling, and Systematic Sampling. The script was tested by comparisons between results from literature, and utilizing real database, comparing its results with the ones achieved using Microsoft Excel. The created scripts proved efficient when compared to the tested and validated examples.

**Keywords:** Forest Mensuration, Data processing, Cubic volume.

**LISTA DE FIGURAS**

[Figura 1 – Primeiras seis linhas da base de dados utilizada como exemplo no script da cubagem pelo método de Smalian. 23](file:///D:\Documents\R\trabalhos_manejo\tcc_inventario\TCC_parte_escrita\VERSAO_FINAL_SCRIPT%20PARA%20PROCESSAMENTO%20DE%20INVENTÁRIO%20FLORESTAL%20UTILIZANDO%20LINGUAGEM%20R%20-%20Copy.docx#_Toc477991888)

[Figura 2 – Primeiras seis linhas do resultado da cubagem pelo método de Smalian utilizando *R* base. 25](#_Toc477991889)

[Figura 3 - Volume estimado pelo método de Smalian utilizando *dplyr*. 26](#_Toc477991890)

[Figura 4 - Resultados da compilação dos dados por árvore e cálculo do fator de forma utilizando *R* base. 28](#_Toc477991891)

[Figura 5 - Volume estimado por meio de fator de forma utilizando *R* base. 29](#_Toc477991892)

[Figura 6 - Separação das chaves para o ajuste de equação de volume utilizando *R* base. 30](#_Toc477991893)

[Figura 7 - Resultados dos ajustes de equação de volume utilizando *R* base. 31](#_Toc477991894)

[Figura 8 - Volume estimado por meio de equação volumétrica utilizando *R* base. 32](#_Toc477991895)

[Figura 9 - Dados compilados por árvore e cálculo do fator de forma utilizando *dplyr*. 33](#_Toc477991896)

[Figura 10 - Volume estimado pelo fator de forma utilizando *dplyr*. 34](#_Toc477991897)

[Figura 11 - Coeficientes do ajuste do modelo volumétrico com casca utilizando *dplyr*. 35](#_Toc477991898)

[Figura 12 - Coeficientes do ajuste do modelo volumétrico sem casca utilizando *dplyr*. 36](#_Toc477991899)

[Figura 13 - Volume com casca e sem casca estimado por meio de equação volumétrica utilizando *dplyr*. 37](#_Toc477991900)

[Figura 14 - Dados de inventário utilizados como exemplo. 38](#_Toc477991901)

[Figura 15 - Tabela de altura dominante gerada utilizando o *R* base. 39](#_Toc477991902)

[Figura 16 - Variável altura dominante unida aos dados originais por meio do *R* base. 40](#_Toc477991903)

[Figura 17 - Coeficientes do ajuste do modelo hipsométrico utilizando o *R* base. 41](#_Toc477991904)

[Figura 18 - Altura estimada para as árvores não medidas utilizando *R* base. 42](#_Toc477991905)

[Figura 19 - *Área Seccional*, *Idade*, Volume com e sem casca utilizando *R* base. 44](#_Toc477991906)

[Figura 20 - Dados de inventário por parcela gerados utilizando *R* base. 46](#_Toc477991907)

[Figura 21 - Dados por parcela por talhão. 46](#_Toc477991908)

[Figura 22 - Resultados da Amostragem Casual Simples considerando população infinita utilizando o *R* base. 48](#_Toc477991909)

[Figura 23 - Resultados por estrato da Amostragem Casual Estratificada considerando população infinita utilizando *R* base. 52](#_Toc477991910)

[Figura 24 - Tabela final para Amostragem Casual Estratificada utilizando *R* base. 52](#_Toc477991911)

[Figura 25 - Tabela de resultados da Amostragem Sistemática utilizando *R* base. 53](#_Toc477991912)

[Figura 26 - Altura dominante estimada utilizando *dplyr*. 54](#_Toc477991913)

[Figura 27 - Coeficientes das equações de altura gerados utilizando *dplyr*. 55](#_Toc477991914)

[Figura 28 - Altura estimada (*HT\_EST*) para árvores não medidas utilizando *dplyr*. 56](#_Toc477991915)

[Figura 29 - Cálculo da área seccional (*AS*), idade (*IDADE*) e estimação dos volumes com e sem casca (*VCC* e *VSC*) utilizando *dplyr*. 57](#_Toc477991916)

[Figura 30 - Dados de inventário por parcela gerados utilizando *dplyr*. 58](#_Toc477991917)

[Figura 31 - Amostragem Casual Simples considerando população infinita utilizando *dplyr*. 61](#_Toc477991918)

[Figura 32 - Resultados por estrato do inventário Casual Estratificado considerando população infinita utilizando *dplyr*. 64](#_Toc477991919)

[Figura 33 - Resultados finais do inventário Casual Estratificado considerando população infinita utilizando *dplyr*. 65](#_Toc477991920)

[Figura 34 - Resultados do inventário sistemático utilizando *dplyr*. 66](#_Toc477991921)

[Figura 35 - Resultado do inventário piloto do estudo de caso da Amostragem Casual Simples. 67](#_Toc477991922)

[Figura 36 - Resultados do inventário piloto da Amostragem Casual Estratificada. 68](#_Toc477991923)

[Figura 37 - Volume e sentido de caminhamento das parcelas para o estudo de caso da Amostragem Sistemática. 70](#_Toc477991924)

**LISTA DE TABELAS**

[Tabela 1 - Tabela comparativa para o volume de 10 árvores utilizando o método de Smalian 27](#_Toc477991925)

[Tabela 2 - Volume das parcelas sorteadas para o inventário piloto da Amostragem Casual Simples. 66](#_Toc477991926)

[Tabela 3 - Parcelas sorteadas para o inventário definitivo com Amostragem Casual Simples. 67](#_Toc477991927)

[Tabela 4 - Tabela comparativa entre os principais estimadores para a Amostragem Casual Simples. 67](#_Toc477991928)

[Tabela 5 - Parcelas sorteadas para o inventário piloto na Amostragem Casual Estratificada. 68](#_Toc477991929)

[Tabela 6 - Parcelas sorteadas para o inventário definitivo utilizando Amostragem Casual Estratificada. 69](#_Toc477991930)

[Tabela 7 - Tabela comparativa entre os principais estimadores para a Amostragem Casual Estratificada. 70](#_Toc477991931)

[Tabela 8 - Tabela comparativa entre os principais estimadores para a Amostragem Sistemática. 71](#_Toc477991932)

**SUMÁRIO**

[1 INTRODUÇÃO 12](#_Toc477991933)

[2 OBJETIVO 13](#_Toc477991934)

[2.1 Geral 13](#_Toc477991935)

[2.2 Específico 14](#_Toc477991936)

[3 MATERIAL E MÉTODOS 14](#_Toc477991937)

[3.1 Linguagem de Programação 14](#_Toc477991938)

[3.2 Ambiente de Desenvolvimento Integrado 15](#_Toc477991939)

[3.3 Hardware 16](#_Toc477991940)

[3.4 Método de Cubagem 16](#_Toc477991941)

[3.5 Estimadores de Volume 16](#_Toc477991942)

[3.6 Estimador de Altura 17](#_Toc477991943)

[3.7 Amostragem Casual Simples 17](#_Toc477991944)

[3.8 Amostragem Casual Estratificada 19](#_Toc477991945)

[3.9 Amostragem Sistemática 21](#_Toc477991946)

[3.10 Avaliação do script 21](#_Toc477991947)

[4 RESULTADOS E DISCUSSÃO 22](#_Toc477991948)

[4.1 Cubagem pelo método de Smalian utilizando *R* base 22](#_Toc477991949)

[4.2 Cubagem pelo método de Smalian utilizando *dplyr* 26](#_Toc477991950)

[4.3 Estudo de caso para o método de Smalian 27](#_Toc477991951)

[4.4 Compilar dados por árvore e cálculo do fator de forma utilizando *R* base 27](#_Toc477991952)

[4.5 Estimação de volume utilizando *R* base 29](#_Toc477991953)

[4.5.1 Estimação de volume por meio do fator de forma – *R* base 29](#_Toc477991954)

[4.5.2 Ajuste de equação volumétrica – *R* base 30](#_Toc477991955)

[4.5.3 Estimação de volume por meio de equação volumétrica – *R* base 32](#_Toc477991956)

[4.6 Compilar dados por árvore e cálculo do fator de forma utilizando *dplyr* 33](#_Toc477991957)

[4.7 Estimação de volume utilizando *dplyr* 34](#_Toc477991958)

[4.7.1 Estimação de volume por meio do fator de forma - *dplyr* 34](#_Toc477991959)

[4.7.2 Ajuste de equação volumétrica - *dplyr* 35](#_Toc477991960)

[4.7.3 Estimação de volume por meio de equação volumétrica utilizando *dplyr* 36](#_Toc477991961)

[4.8 Inventário utilizando *R* base 37](#_Toc477991962)

[4.8.1 Ajuste de modelo hipsométrico e estimação da altura para árvores não medidas – *R* base 38](#_Toc477991963)

[4.8.2 Estimação de volume e quantificação de parcelas – *R* base 43](#_Toc477991964)

[4.8.3 Amostragem Casual Simples – *R* base 47](#_Toc477991965)

[4.8.4 Amostragem Casual Estratificada – *R* base 49](#_Toc477991966)

[4.8.5 Amostragem Sistemática – *R* base 52](#_Toc477991967)

[4.9 Inventário utilizando *dplyr* 53](#_Toc477991968)

[4.9.1 Ajuste de modelo hipsométrico e estimação da altura para árvores não medidas – *dplyr* 53](#_Toc477991969)

[4.9.2 Estimação de volume e quantificação de parcelas – *dplyr* 56](#_Toc477991970)

[4.9.3 Amostragem Casual Simples – *dplyr* 59](#_Toc477991971)

[4.9.4 Amostragem Casual Estratificada – *dplyr* 61](#_Toc477991972)

[4.9.5 Amostragem Sistemática – *dplyr* 65](#_Toc477991973)

[4.10 Estudo de caso para Amostragem Casual Simples 66](#_Toc477991974)

[4.11 Estudo de caso para Amostragem Casual Estratificada 68](#_Toc477991975)

[4.12 Estudo de caso para Amostragem Sistemática 70](#_Toc477991976)

[5 CONCLUSÃO 71](#_Toc477991977)

[REFERÊNCIAS BIBLIOGRÁFICAS 72](#_Toc477991978)

[ANEXOS 74](#_Toc477991979)

[Anexo A: Modelo para dados de cubagem 74](#_Toc477991980)

[Anexo B: Modelo para dados de inventário 75](#_Toc477991981)

# INTRODUÇÃO

Devido a expansão do setor florestal brasileiro, este vem ganhando destaque no mercado mundial de produtos florestais. Esse aumento da demanda destaca cada vez mais o emprego efetivo de técnicas de manejo florestal, cujo objetivo é a manutenção da produção de bens e serviços em quantidade e qualidade de povoamentos florestais.

Informações necessárias ao manejo e planejamento de povoamentos, que normalmente são referentes ao volume de madeira e/ou outra variável dendrométrica, no sentido qualitativo e quantitativo, são obtidas por meio do inventário florestal (MELLO, *et al.*, 2009).

Um inventário florestal pode fornecer diversas informações, como estimativas de área, descrição da topografia, mapeamento da propriedade, estimativas de quantidade e da qualidade de diferentes recursos florestais e estimativas de crescimento (caso o inventário seja realizado mais de uma vez), sendo que a ênfase em determinado elemento pode variar de acordo com o objetivo do inventário (HUSCH; BEERS; KERSHAW, 2003).

Apesar da literatura específica citar diversos tipos de delineamentos amostrais, 3 tipos de amostragem ganham destaque, sendo as mais usadas: Casual Simples, Sistemática e Estratificada. Sendo que outros delineamentos utilizados geralmente são modificações destes (CAMPOS; LEITE, 2013).

Para se obter o volume de cada parcela, modelos hipsométricos são utilizados em conjunto com modelos volumétricos, de *taper* ou de múltiplos volumes (LEITE; ANDRADE, 2002). Portanto, para o ajuste destes modelos são utilizados dados provindos de cubagem rigorosa em árvores abatidas, ou cubadas ainda em pé, por meio de dendrômetros como o Criterion (OLIVEIRA, *et al.*, 2009).

Outro método citado por Campos e Leite (2013) é o emprego do fator de forma médio, uma razão entre volumes que permite o cálculo do volume real.

Após a coleta, o processamento dos dados de inventário geralmente é feito em escritórios, utilizando diversos tipos de software. Esta etapa é crucial, já que produz as informações da produção e/ou crescimento da população.

Para o processamento dos dados de inventário, muitos usuários optam por softwares de edição de planilhas, como o Microsoft Excel, LibreOffice. Por não serem softwares estatísticos, o engenheiro muitas vezes precisa lidar com diversas planilhas, que não trabalham bem com uma grande base de dados, o que leva o usuário a dividir o dado original várias planilhas. Todo este trabalho pode prejudicar o rendimento, levar a inconsistências e consequentemente elevar o custo.

Entretanto, a maioria destes softwares possuem licença paga e código de fonte fechado. Sendo de difícil acesso ao usuário comum, como o pequeno empresário ou estudantes da área que desejam prestar um serviço, ou até mesmo desenvolver seus trabalhos nas disciplinas.

Visto isso, softwares livres vêm ganhando destaque no processamento de dados, já que não possuem licença, e são de código fonte aberto, ou seja, pode-se verificar a qualidade das análises utilizadas. Entende-se como software livre “respeitar a liberdade e senso de comunidade dos usuários. A grosso modo, os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software” (FREE SOFTWARE FOUNDATION, INC, 2016).

Um destes softwares é o *R,* uma poderosa linguagem e ambiente de programação estatística e gráfica, que possibilita a criação de scripts para diversas funções (R CORE TEAM, 2016).

Empresas florestais podem utilizar um script criado em linguagem *R* específico para a área florestal para processar seus inventários, sem a necessidade de adquirir uma licença paga, diminuindo os custos do inventário. Docentes podem utilizar em aulas, para demonstrações e exemplificações do conteúdo. E discentes da área podem ter acesso a um programa específico para engenharia florestal de forma gratuita.

# OBJETIVO

## Geral

O objetivo deste trabalho foi desenvolver scripts específicos para área florestal, no que se refere a processamento de dados de inventário florestal utilizando o software *R,* demonstrando a aplicabilidade do mesmo.

## Específico

* Criar um script para processamento de dados de cubagem utilizando linguagem *R* sem pacotes adicionais;
* Criar um script para processamento de dados de cubagem utilizando linguagem *R* em conjunto com o pacote *dplyr*;
* Criar um script para processamento de dados de inventário florestal utilizando linguagem *R* sem pacotes adicionais;
* Criar um script para processamento de dados de inventário florestal utilizando linguagem *R* em conjunto com o pacote *dplyr*;

# MATERIAL E MÉTODOS

## Linguagem de Programação

Para o desenvolvimento deste script foi utilizado o *R,* uma poderosa linguagem e ambiente de programação estatística e gráfica. O *R* está disponível para Windows (7,8, 10), Linux e Mac OS X. (R CORE TEAM, 2016).

No *R,* a forma mais comum de se compartilhar processamentos e programas são scripts. Scripts são arquivos de texto, com extensão .R, que contém o código necessário para aquela determinada análise. Além do código geralmente são inseridos comentários, para melhorar o entendimento do script, utilizando “#” no início das frases. Qualquer usuário pode criar um script, que realiza um determinado tipo de processamento, e compartilha-lo por e-mail, ou mídias físicas. A vantagem do script é que, por se tratar de um arquivo de texto, mesmo que ele realize um processamento pesado, e que contenha centenas de linhas de código, ele sempre possui um tamanho muito pequeno, o que facilita o seu compartilhamento. Qualquer pessoa que tenha o *R* instalado pode abrir um script e rodá-lo em seu computador, e, caso o script utilize pacotes, é necessário tê-los instalados na máquina.

Outra vantagem desta linguagem é que ela é expansiva por meio de pacotes, que se tratam de compilações de códigos criados por usuários da comunidade, gerando ferramentas que o *R* padrão (ou *R* base) não possui de forma nativa. Isto abre uma gama enorme de possiblidades, dando ao usuário inúmeras ferramentas de análise.

Um destes pacotes é o *dplyr*, que fornece uma série de ferramentas para a manipulação eficiente de dados no *R* (WICKHAM; FRANCOIS, 2015). Todos os cálculos e análises neste script foram feitos utilizando o *R* base, e repetidos utilizando o *dplyr*. Isto dá ao usuário a opção de escolher qual código prefere utilizar. Os códigos do *dplyr* são mais amigáveis para usuário leigo, pois suas funções (ou verbos) possuem nomes objetivos, de fácil entendimento. As computações utilizando o *R* base podem ser um pouco menos intuitivas, principalmente a alguém que não tem muita familiaridade com a linguagem, porém não há a necessidade de instalar pacotes. Com o *dplyr*, não é necessário criar chaves adicionais, realizar loops, ou manipulações mais complexas, sendo muito mais prático de se utilizar. Todo o processo é feito em um único *pipe*, que são linhas de comando seguidas de *%>%*, um comando que pode ser lido como “depois”.

Portanto, cada processamento foi separado em 2 scripts, um utilizando apenas o *R* base, e outro o pacote *dplyr*.

## Ambiente de Desenvolvimento Integrado

Uma linguagem de programação pode ter diversos ambientes. Para a linguagem *R,* um dos Ambientes de Desenvolvimento Integrado (ADI) ou Integrated Development Environment (IDE) mais utilizados é o RStudio, que inclui console, destaque de sintaxes e ferramentas especializadas, como controle de versão e criações de aplicativos html e geração de documentos em .pdf. O RStudio está disponível para Windows (Vista, 7,8,10) Mac OS X e Linux.

Utilizando o RStudio como IDE, o usuário pode abrir um ou mais scripts, visualizar os seus resultados no console, visualizar objetos salvos e gráficos direto na interface, e importar arquivos por linhas de comando ou utilizando a interface. É possível importar/exportar arquivos nos formatos .txt e .csv de forma nativa, e com o auxílio de pacotes, outros tipos de formatos, como .xlsx. É possível ainda criar projetos, que salvam todas as informações do projeto e definem o diretório de trabalho.

## Hardware

Para o desenvolvimento do script foi utilizado um computador com CPU intel i5 de 3ª geração, 2.5GHz, 6GB de memória RAM e 1T de armazenamento.

## Método de Cubagem

O script pode processar dados de cubagem utilizando o método de Smalian. Este método consiste na medição de diâmetros ao longo do fuste das árvores (HUSCH; BEERS; KERSHAW, 2003). O volume da árvore é calculado pelo somatório dos volumes de cada seção, sendo que o volume de cada seção é calculado por meio das fórmulas a seguir. Outros métodos podem ser implementados com pequenas alterações ao código.

()

(2)

Em que:

= volume com casca da seção, em m³;

= área seccional com casca, obtida com o diâmetro mensurado no início da seção, em m²;

= área seccional com casca, obtida com o diâmetro mensurado no final da seção, em m²;

= comprimento da seção, em m;

= volume total da árvore, em m³.

## Estimadores de Volume

Como o objetivo do script não é o ajuste de modelos, foram utilizados apenas exemplos. Como exemplo foi utilizado o fator de forma descrito por Campos e Leite (2013) e o modelo proposto por Schumacher e Hall (1933), como visto a seguir:

(3)

(4)

(5)

Em que:

*f* = Fator de forma;

*Vreal* = Volume real calculado na cubagem, em m³

*Vcilindro* = Volume de um cilindro perfeito, em m³;

*Ln* = logaritmo Neperiano;

*Vcc* = volume total com casca, em m³;

*DAP* = diâmetro a 1,30 m do solo, em cm;

*HT* = altura total, em m;

β0, β1 e β2 = parâmetros do modelo;

εi = erro aleatório.

## Estimador de Altura

Para estimar a altura das árvores que não foram mensuradas em campo foi utilizado como exemplo o modelo proposto por Leite e Andrade (2003). O usuário pode ajustar outros modelos caso deseje, fazendo poucas alterações no código. O modelo é descrito abaixo:

(6)

em que:

*HD* = Média das alturas dominantes, em m.

## Amostragem Casual Simples

Os algoritmos do script seguem as metodologias descritas por Husch, Beers e Kershaw (2003). Foram implementados os métodos de cálculo considerando a população finita, e a população infinita, dessa forma, cabe ao usuário decidir qual utilizar. Os estimadores deste método são:

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

(17)

(18)

(19)

(20)

(21)

(22)

(23)

Em que:

*N* = Número de parcelas cabíveis na população;

*n* = Número de parcelas amostradas;

*Y𝑖* = Quantidade da variável *Y* medida na *i*-ésima unidade de amostra;

= Média de *Y* por unidades de amostra; uma estimativa da média populacional;

*Ŷ* = Estimativa do total da variável *Y* para a população.

*S²* = Variância da amostra;

*S* = Desvio-padrão da amostra;

*CV* = Coeficiente de variação da amostra;

= Variância da média;

= Erro-padrão da média;

= Variância do total da população;

= Erro-padrão do total da população;

*tα*= estatística *t*-student considerando probabilidade α e *n* - 1 graus de liberdade;

*E%adm* = Erro admitido, em porcentagem;

*E* = Erro absoluto do inventário;

*E%* = Erro relativo do inventário;

= Tamanho necessário da amostra para atender a precisão requerida;

= Intervalo de confiança para a média da população;

= Intervalo de confiança para o total da população.

## Amostragem Casual Estratificada

A computação da Amostragem Casual Estratificada foi implementada seguindo a metodologia descrita por Husch, Beers e Kershaw (2003). Foi utilizado o método de Neyman (ótimo) para alocação de parcelas. Foram inseridas fórmulas considerando população finita e infinta, assim, o usuário tem a opção de escolher qual método deseja utilizar. Os principais estimadores são os seguintes, sendo que, os demais, são os mesmos da Amostragem Casual Simples:

(24)

(25)

(26)

(27)

(28)

(29)

(30)

(31)

(32)

(33)

(34)

(35)

(36)

(37)

(38)

(39)

(40)

Em que:

*N* = Número total de unidades de amostra na população;

*M* = Número de estratos na população;

*Nj* = Número total de unidades de amostra em cada *j*-ésimo estrato, *j* = 1, 2, ..., *M*;

*n* = Número de unidades lançadas em todos os estratos;

*nj* = Número de unidade de amostra lançada em cada *j*-ésimo estrato;

*Pj* = Proporção do número de unidades de amostra em cada estrato em relação ao número total de unidades de amostra;

= Média da variável *Y* para cada *j*-ésimo estrato;

*Yij* = Quantidade da variável *Y* na *i*-ésima unidade de amostra, do *j*-ésimo estrato;

= Média estratificada ou média ponderada;

= Valor total estimado de *Y* para a população;

= Valor total estimado de *Y* para cada *j*-ésimo estrato;

= Variância estimada de *Y* em cada *j*-ésimo estrato;

= Desvio-padrão de *Y* em cada *j*-ésimo estrato;

= Erro admitido absoluto;

= Tamanho da amostra necessária para o erro admitido, utilizando o método de alocação de Neyman (ótimo);

= Fixação do número de unidades de amostra, utilizando o método de alocação de Neyman (ótimo).

## Amostragem Sistemática

A computação da Amostragem Sistemática foi implementada seguindo a metodologia descrita por Soares, Neto e Souza (2012). Seus estimadores são os mesmos da Amostragem Casual Simples, exceto pelo erro-padrão da média. Caso o usuário realize uma Amostragem Sistemática e deseje processar como Casual Simples, basta utilizar os scripts descritos no tópico 3.7. O método utilizado para o cálculo foi o método das diferenças sucessivas. Sua fórmula é apresentada abaixo:

(41)

## Avaliação do script

A avaliação do script foi feita comparando-se os resultados de estudos de caso presentes no livro de Soares, Neto e Souza (2012), no caso do inventário, e uma base de dados real processada utilizando o software Microsoft Excel no caso da cubagem.

# RESULTADOS E DISCUSSÃO

Foram produzidos no total 4 scripts: 2 para cubagem pelo método de Smalian, sendo um utilizando apenas o *R* base, e outro utilizando o *dplyr*; e 2 para inventário, sendo um utilizando apenas o *R* base, e outro utilizando o *dplyr*. Cada script funciona como um programa, portanto eles foram separados para facilitar a interação com o usuário. Caso o usuário deseje processar o inventário, ele abre apenas o script de inventário, e o processa, podendo ainda escolher se deseja usar apenas os comandos do *R* base ou utilizando o pacote *dplyr*, que facilita o processamento.

Em todos os scripts deste trabalho, os primeiros comandos são de importação de dados, onde os dados são salvos em um objeto. Isso é exemplificado uma vez, mas deve ser repetido sempre que uma base de dados externa for utilizada. A nomenclatura das variáveis deve ser respeitada para que o script funcione corretamente. Portanto, para a reprodução exata dos códigos apresentados, os nomes de objetos e variáveis devem ser respeitados. A função *head* será utilizada frequentemente, para demonstrar os resultados. Ela mostra as 6 primeiras linhas de um objeto.

É comum entre as linhas de comando ou até mesmo dentro de linhas de comando mais complexas, o uso do símbolo “#” seguido de textos. Estes textos geralmente explicam a linha de comando que o acompanha, de forma a facilitar o entendimento do script.

## Cubagem pelo método de Smalian utilizando *R* base

Os dados utilizados neste script estão em formato *.csv*, e localizados em uma pasta chamada "dados", dentro do diretório de trabalho. Portanto, para se realizar a importação destes, utiliza-se a função *read.csv2*, especificando apenas o nome da pasta e nome do arquivo separados por "/".

Caso o diretório de trabalho não esteja definido, pode-se utilizar a função *setwd("caminho da pasta desejada")* para defini-lo.

*dados\_sma <- read.csv2("dados/dados\_sma.csv")*

Obs: A função *read.csv2* foi utilizada pois o seu padrão de importação é "," para decimal, e ";" para separador, que é o padrão utilizado em computadores brasileiros. Caso o padrão dos dados utilizados seja o americano, deve-se utilizar a função *read.csv*.

O zero pode atrapalhar em médias (aumentando o *n*) e gerar erro em regressões. Portanto, para se trabalhar melhor no *R*, deve-se converter zeros em NA:

*dados\_sma[dados\_sma == 0] <- NA*

Na figura 1 são ilustrados os dados utilizados da forma em que são visualizados no console do *R* utilizando a função *head*. Estes seguem o modelo apresentado no Anexo A:



Figura – Primeiras seis linhas da base de dados utilizada como exemplo no script da cubagem pelo método de Smalian.

Caso os dados possuam a variável *e\_casca*, é possível calcular AS sem casca, e consequentemente, volume sem casca. A seguir é feito o cálculo do diâmetro sem casca, onde a casca está em mm; caso esta já esteja em cm, não se deve dividir por 10.

*dados\_sma$di\_sc <- dados\_sma$di\_cc - 2 \* (dados\_sma$e\_casca/10)*

Cálculo da área seccional:

*dados\_sma$AS\_CC<-(dados\_sma$di\_cc^2\*pi)/40000  
dados\_sma$AS\_SC <- (dados\_sma$di\_sc^2 \* pi)/40000*

É necessário criar-se uma chave para se separar o dado em classes. A chave pode ser composta de uma ou mais variáveis. Cada variável da chave possuí níveis, por exemplo: a variável TALHAO indica a qual talhão aquela árvore pertence, Outros exemplos de variáveis que podem ser utilizadas como chaves são talhão, código genético, etc. Caso seja utilizado mais de uma chave, é necessário uni-las em uma única variável. Para isso utiliza-se a função *paste*. Neste exemplo serão unidas as variáveis TALHAO e ARVORE, separadas por “\_”, como indicado pelo argumento “sep”:

*dados\_sma$TALHAO\_ARV <- paste(dados\_sma$TALHAO, dados\_sma$ARVORE,   
sep = "\_")*

Variáveis criadas com paste possuem classe *character*, por isso, é importante a conversão em fator, definindo-se os níveis:

*dados\_sma$TALHAO\_ARV <- factor(dados\_sma$TALHAO\_ARV, levels = unique(dados\_sma$TALHAO\_ARV))*  
Agora que as variáveis necessárias foram criadas, estima-se o volume pelo método de Smalian. Primeiro, fixa-se os dados para facilitar digitação com a função *attach*:

*attach(dados\_sma)*

Como a variável ARVORE por ter *n* número de linhas dependendo do dado utilizado, e é necessário se diferenciar quando uma árvore termina e outra começa para que o método de Smalian seja aplicado, é necessário utilizar um *for loop.*

Como será utilizado um *for loop*, primeiro cria-se as listas e dataframes a serem utilizados dentro dele antecipadamente. Gera-se duas listas, pois tem-se volume com casca e volume sem casca, neste caso.

*my.list1 <- vector("list", nrow(dados\_sma))  
my.list2 <- vector("list", nrow(dados\_sma))  
dataC1 <- data.frame()  
dataC2 <- data.frame()*

Cria-se dois dataframes e duas listas pois o cálculo será feito do volume com o sem casca. Agora segue-se com o loop:

*# Indica que o loop vai da primeira até a última linha do data frame  
for(i in 1:nrow(dados\_sma))   
{   
 # Inserir NA quando árvore for diferente,   
 # garantindo que não se faça cálculos misturando as duas árvores  
 if(is.na(TALHAO\_ARV[i+1]) | TALHAO\_ARV[i] != TALHAO\_ARV [i+1] )   
 {   
 my.list1[[i]] <- NA  
 my.list2[[i]] <- NA  
 }  
 # cálculo do volume quando a árvore for igual  
 else if( TALHAO\_ARV[i] == TALHAO\_ARV[i+1] )   
 {   
 # Volume com casca  
 my.list1[[i]] <- ((AS\_CC[i] + AS\_CC[i+1]) / 2) \* (hi[i+1] - hi[i])   
 # Volume sem casca  
 my.list2[[i]] <- ((AS\_SC[i] + AS\_SC[i+1]) / 2) \* (hi[i+1] - hi[i])   
 } }  
# desafixa-se os dados  
detach(dados\_sma)*

Após os volumes terem sido salvos em listas, as listas são convertidas em matrizes, e unidas aos dados originais:

*# Conversão de lista para dataframe  
dataC1 <- rbind(dataC1, do.call(rbind, my.list1))   
# Conversão de lista para dataframe  
dataC2 <- rbind(dataC2, do.call(rbind, my.list2))   
# Renomear lista com o nome desejado  
names(dataC1) <- "VCC"   
# Renomear lista com o nome desejado  
names(dataC2) <- "VSC"  
# União com os dados originais   
dados\_vol\_secao <- cbind(dados\_sma, dataC1, dataC2)*   
 O resultado é salvo no objeto *dados\_vol\_secao*, como pode ser visto na figura 2:

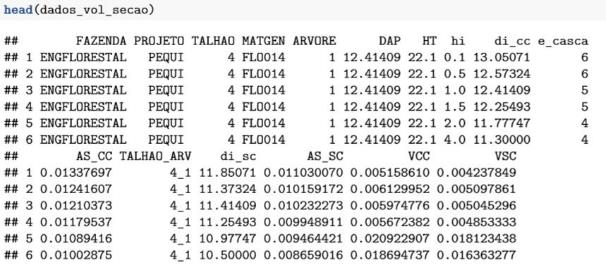


Figura – Primeiras seis linhas do resultado da cubagem pelo método de Smalian utilizando *R* base.

## Cubagem pelo método de Smalian utilizando *dplyr*

Os dados utilizados são os mesmos do método anterior, e podem ser visualizados na figura 1. Tudo é processado como se fosse uma única linha de comando, de forma direta, e salvo no objeto *dados\_vol\_secao*:

*dados\_vol\_secao <- dados\_sma %>% # definicao do df  
 group\_by(TALHAO, ARVORE) %>% # grupo  
 mutate( # função para adicionar novas variáveis  
 di\_sc = di\_cc-2\*(e\_casca/10), # cálculo do diâmetro sem casca  
 AS\_CC = (di\_cc^2 \* pi) / 40000, # AS com casca  
 AS\_SC = (di\_sc^2 \* pi) / 40000, # AS sem casca  
 VCC = ( (AS\_CC + lead(AS\_CC) )/2) \* (lead(hi) - hi ), # Volume com casca  
 VSC = ( (AS\_SC + lead(AS\_SC) )/2) \* (lead(hi) - hi ) ) # Volume sem casca  
# lead: acessa a linha seguinte*

O resultado pode ser visualizado na figura 3. Percebe-se que este é idêntico ao obtido quando se utiliza o *R* base:

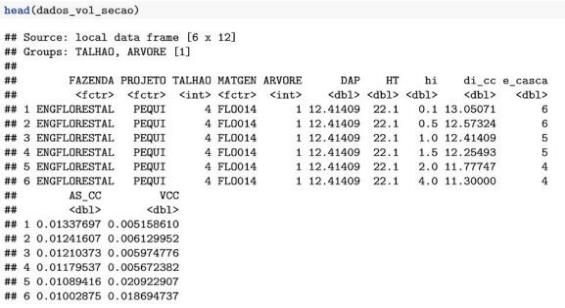


Figura - Volume estimado pelo método de Smalian utilizando *dplyr*.

## Estudo de caso para o método de Smalian

Utilizando dados reais de uma empresa florestal localizada em Minas Gerais, foi processado o volume de 10 árvores utilizando o script, e o Microsoft Excel, segundo a metodologia de Smalian. Os resultados foram idênticos, porém, seu processamento com o script foi muito mais prático de se fazer, utilizando os códigos demonstrados no tópico 4.2. Na tabela 1 são exibidos os volumes obtidos:

Tabela - Tabela comparativa para o volume de 10 árvores utilizando o método de Smalian

|  |  |  |
| --- | --- | --- |
| **Árvore** | **Script** | **Microsoft Excel** |
| 1 | 0,7486 | 0,7486 |
| 2 | 0,7720 | 0,7720 |
| 3 | 0,7380 | 0,7380 |
| 4 | 0,7727 | 0,7727 |
| 5 | 0,7144 | 0,7144 |
| 6 | 0,8022 | 0,8022 |
| 7 | 0,8580 | 0,8580 |
| 8 | 0,8528 | 0,8528 |
| 9 | 0,8749 | 0,8749 |
| 10 | 0,8745 | 0,8745 |

## Compilar dados por árvore e cálculo do fator de forma utilizando *R* base

A seguir, são mostrados os códigos para se compilar os dados por árvore utilizando o *R* base. Foi utilizado como exemplo o volume obtido no tópico 4.1.

Para se compilar os dados por árvore, precisa-se da média dos valores de *DAP*, e *HT* por árvore e da soma dos volumes; a variável que será utilizada como chave é inserida com a função *unique*,

Para a compilação faz-se o uso da função *by* em conjunto com *do.call*. No caso, isso é feito duas vezes, utilizando a função *data.frame* para se converter o resultado em colunas de um dataframe. Caso a chave não tenha sido criada ainda, ela deve ser criada:

*dados\_vol\_secao$TALHAO\_ARV <- paste(dados\_vol\_secao$TALHAO,*

*dados\_vol\_secao$ARVORE, sep = "\_")*

É importante a conversão em fator, definindo-se os níveis:

*dados\_vol\_secao$TALHAO\_ARV <- factor(dados\_vol\_secao$TALHAO\_ARV,*

*levels = unique(dados\_vol\_secao$TALHAO\_ARV))*

Agora prossegue-se com os cálculos:

*dados\_vol\_arvore <- data.frame(CHAVE = unique(dados\_vol\_secao$TALHAO\_ARV),*

*do.call(rbind, by(dados\_vol\_secao[c("DAP", "HT")], dados\_vol\_secao$TALHAO\_ARV,*

*colMeans, na.rm = T)), do.call(rbind, by(dados\_vol\_secao[c("VCC",*

*"VSC")], dados\_vol\_secao$TALHAO\_ARV, colSums, na.rm = T)))*

A seguir é feito o cálculo da porcentagem da casca:

dados\_vol\_arvore$PORC\_CASCA <- ((dados\_vol\_arvore$VCC -

dados\_vol\_arvore$VSC)/dados\_vol\_arvore$VCC) \* 100

A seguir é feito o cálculo do fator de forma com e sem casca:

*dados\_vol\_arvore$FFCC <- dados\_vol\_arvore$VCC/(pi \* dados\_vol\_arvore$DAP^2/40000 \* dados\_vol\_arvore$HT)*

*dados\_vol\_arvore$FFSC <- dados\_vol\_arvore$VSC/(pi \* dados\_vol\_arvore$DAP^2/40000 \* dados\_vol\_arvore$HT)*

Utilizando a função *head* visualiza-se as seis primeiras linhas do objeto em que os cálculos foram salvos, como pode ser visto na figura 4:

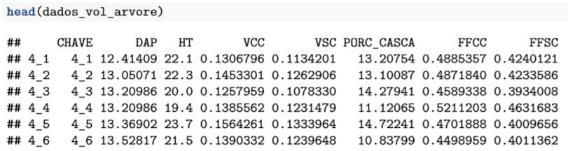


Figura - Resultados da compilação dos dados por árvore e cálculo do fator de forma utilizando *R* base.

## Estimação de volume utilizando *R* base

Caso o objetivo seja estimar o volume utilizando os dados de cubagem, para fins comparativos, por exemplo, pode-se utilizar do fator de forma, ou equações volumétricas.

Os dados utilizados de exemplo para a estimação de volume foram os demostrados na figura 4.

### Estimação de volume por meio do fator de forma – *R* base

Para se estimar o volume utilizando o fator de forma, primeiro calcula-se o fator de forma médio:

*FFMED <- cbind(mean(dados\_vol\_arvore$FFCC), mean(dados\_vol\_arvore$FFSC))*

Em seguida basta estimar o volume utilizando o fator de forma médio:

*dados\_vol\_arvore$VFFCC <- FFMED[1] \* (pi \* dados\_vol\_arvore$DAP^2/40000 \**

*dados\_vol\_arvore$HT)*

*dados\_vol\_arvore$VFFSC <- FFMED[2] \* (pi \* dados\_vol\_arvore$DAP^2/40000 \**

*dados\_vol\_arvore$HT)*

O volume estimado pelo fator de forma pode ser visto na figura 5:

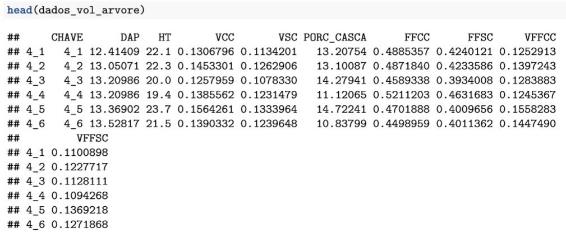


Figura - Volume estimado por meio de fator de forma utilizando *R* base.

### Ajuste de equação volumétrica – *R* base

Como exemplo, o ajuste das equações volumétricas foi feito por talhão. Este procedimento é muito simples de ser feito com o pacote *dplyr*, porém, utilizando apenas o *R* base ele se torna um pouco mais complexo.

O modelo de Schumacher e Hall (1993) será ajustado por talhão, porém, a variável talhão foi unida com a variável árvore, gerando uma chave combinada, e as variáveis originais foram perdidas no *loop for*. Portanto, é necessário transformar a chave em variáveis separadas novamente. Para isso utiliza-se a função *strsplit*. Como o resultado desta função é uma lista, a mesma é convertida utilizando *do.call*:

*aux1 <- data.frame(do.call(rbind, strsplit(as.character(dados\_vol\_arvore$CHAVE), "\_")))  
names(aux1) <- c("TALHAO", "ARVORE")*

Agora basta unir as chaves ao dado original, e converte-las para fator:

*dados\_vol\_arvore$TALHAO <-factor(aux1$TALHAO, levels= unique(aux1$TALHAO))*

*dados\_vol\_arvore$ARVORE <-factor(aux1$ARVORE, levels =unique(aux1$ARVORE))*

Observa-se na figura 6 que as chaves foram separadas com sucesso:

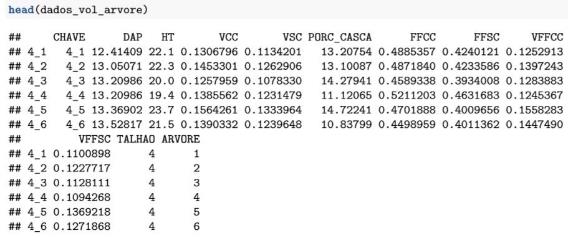


Figura - Separação das chaves para o ajuste de equação de volume utilizando *R* base.

Agora ja é possivel realizar a regressao por talhão. Utilizando a função *by* em conjunto com *lm*, executa-se a regressão linear por grupo. Sao utilizados 4 argumentos:

1. Matriz ou dataframe que contém os dados a serem utilizados;
2. Variável classificatória indicando o grupo que será utilizado;
3. Indica função a ser executada (*lm*);
4. Aqui é inserido o argumento "formula", da função lm, onde indica-se o modelo que será utilizado.

*aux2 <- by(dados\_vol\_arvore, dados\_vol\_arvore$TALHAO, lm,*

*formula = log(VCC) ~ log(DAP) + log(HT))*

Para converter esta saída em um *dataframe* contendo os coeficientes, *R²* ajustado e erro, utiliza-se a função *vapply*, que é composta por 3 argumentos:

1. Matriz ou vetor em que se deseja aplicar uma função em cada elemento;
2. Função a ser aplicada na matriz ou vetor; neste caso, cria-se uma função que extrai os coeficientes, o r quadrado e o erro do ajuste;
3. *template* que fornece os nomes das colunas a serem criadas pela função.

Esta função gera uma matriz com duas linhas e *n* colunas, dependendo do número de coeficientes. Então transpõe-se o seu resultado com a função *t*:

*regbase <- t(vapply(aux2, function(x) c(coef(x), summary(x)$adj.r.squared,*

*summary(x)$sigma), c(b0 = 0, b1 = 0, b2 = 0, Rsqr = 0, Std.Error = 0)))*

A função *by* salva a chave na linha do objeto; devido a isso, é preciso uni-la novamente ao *dataframe*:

*tabcoef\_vcc <- cbind(TALHAO = as.numeric(rownames(regbase)), as.data.frame(regbase))*

Remoção dos rownames (nomes das linhas):

*row.names(tabcoef\_vcc) <- NULL*

O resultado dos ajustes pode ser visualizado na figura 7:

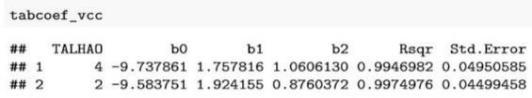


Figura - Resultados dos ajustes de equação de volume utilizando *R* base.

Este procedimento pode ser repetido utilizando o volume sem casca como variável dependente, obtendo-se as equações para volume sem casca.

### Estimação de volume por meio de equação volumétrica – *R* base

Como existe um conjunto de coeficientes por talhão, é necessário unir os dados originais com os dados de coeficiente com base na variável chave, *TALHAO*. Para isso utiliza-se a função *merge*:

*dados\_vol\_arvore <- merge(dados\_vol\_arvore, tabcoef\_vcc, by = "TALHAO")*

Merge pode mudar a ordem dos dados, portanto, utiliza-se *order* para reorganiza-los. Em seguida remove-se os nomes das linhas, pois estes foram alterados por *merge*:

*dados\_vol\_arvore <- dados\_vol\_arvore[order(dados\_vol\_arvore$ARVORE), ]*

*rownames(dados\_vol\_arvore) <- NULL*

Agora basta estimar o volume utilizando a equação:

*dados\_vol\_arvore$VEQCC <- exp(dados\_vol\_arvore$b0 + dados\_vol\_arvore$b1 \**

*log(dados\_vol\_arvore$DAP) + dados\_vol\_arvore$b2 \* log(dados\_vol\_arvore$HT))*

Remove-se os coeficientes após a estimação:

*dados\_vol\_arvore[c("b0", "b1", "b2", "Rsqr", "Std.Error")] <- NULL*

O resultado da estimação pode ser visualizado na figura 8:

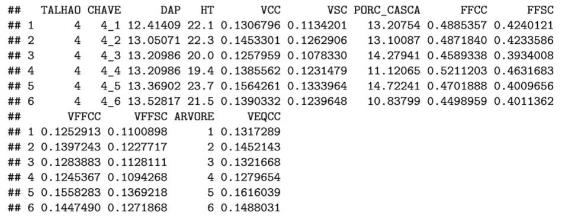


Figura - Volume estimado por meio de equação volumétrica utilizando *R* base.

O volume sem casca pode ser estimado de forma similar, utilizando os coeficientes provindos do ajuste sem casca.

## Compilar dados por árvore e cálculo do fator de forma utilizando *dplyr*

A seguir são demonstrados os códigos utilizados para compilar os dados por árvore, cálculo do fator de forma e porcentagem de casca utilizando o *dplyr*. Tudo é feito com apenas um comando e sem a necessidade de chaves adicionais. Foi utilizado como exemplo os dados de volume obtidos pelo método de Smalian, demonstrados na figura 1.

A principal função utilizada é a *summarise*, que funciona de forma semelhante à tabela dinâmica do Microsoft Excel:

*dados\_vol\_arvore <- dados\_vol\_secao %>% # define data frame utilizado*

*group\_by(TALHAO, ARVORE) %>% # definição da chave*

*summarize( # Função que compila os dados*

*DAP = mean(DAP) , # Média de DAP*

*HT = mean(HT) , # média de HT*

*VCC = sum(VCC, na.rm = TRUE) , # Soma de volume com casca*

*VSC = sum(VSC, na.rm = TRUE) , # Soma de volume sem casca*

*PORC\_CASCA = ((VCC - VSC)/VCC)\*100 , # Porcentagem da casca*

*FFCC = VCC/(pi \* DAP^2 / 40000 \* HT), # Fator de forma com casca*

*FFSC = VSC/(pi \* DAP^2 / 40000 \* HT) # Fator de forma sem casca*

*) %>% # fim do summarise*

*arrange(ARVORE) # Organizar por árvore (opcional)*

O resultado obtido pelas linhas de comando anteriores pode ser visto na figura 9:

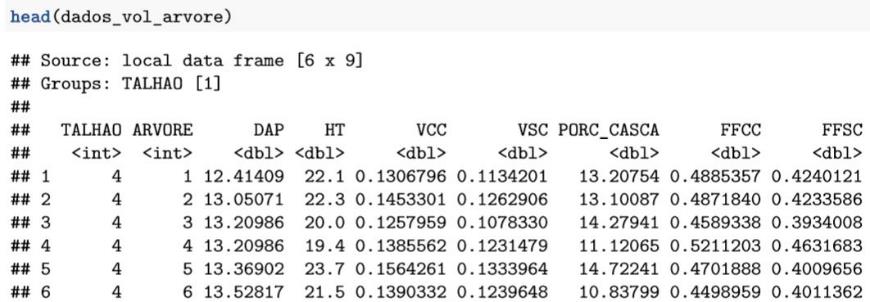


Figura - Dados compilados por árvore e cálculo do fator de forma utilizando *dplyr*.

## Estimação de volume utilizando *dplyr*

Caso o objetivo seja estimar o volume utilizando os dados de cubagem, para fins comparativos, por exemplo, pode-se utilizar do fator de forma, ou de equações volumétricas.

Os dados utilizados de exemplo para a estimação de volume seguem o modelo apresentado no anexo B e podem ser vistos na figura 9.

### Estimação de volume por meio do fator de forma - *dplyr*

Para se estimar o volume por meio do fator de forma utilizando *dplyr*, calcula-se a média do fator de forma direta, e multiplica-se esta pelo volume do cilindro.

*dados\_vol\_arvore <- dados\_vol\_arvore %>%*

*mutate(VFFCC = mean(FFCC) \* (pi \* DAP^2 / 40000 \* HT),*

*VFFSC = mean(FFSC) \* (pi \* DAP^2 / 40000 \* HT) )*

O volume estimado pelo fator de forma pode ser visto na figura 10:

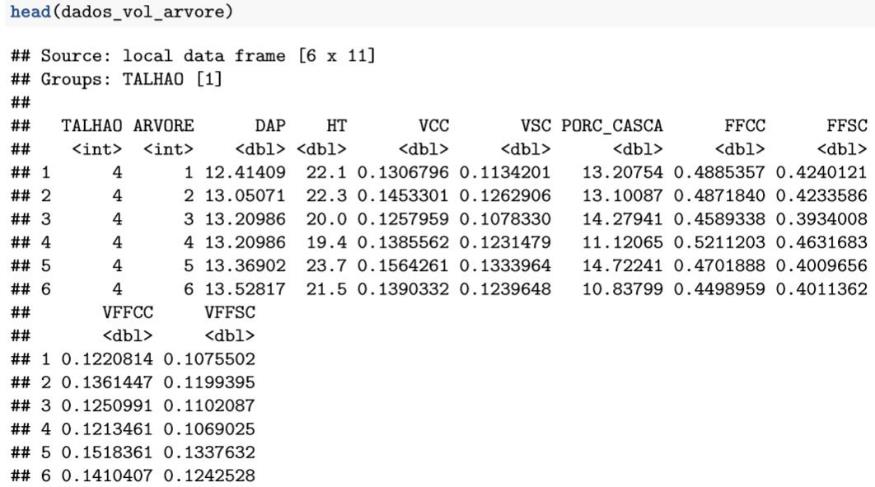


Figura - Volume estimado pelo fator de forma utilizando *dplyr*.

### Ajuste de equação volumétrica - *dplyr*

A seguir, o modelo de Schumacher & Hall (1933) é ajustado para volume com casca, por talhão:

*tabcoef\_vcc <- dados\_vol\_arvore %>% # selecao do df*

*group\_by(TALHAO) %>% # seleção da chave*

*do(mod = lm(log(VCC) ~ log(DAP) + log(HT), data =.)) %>% # ajuste do modelo*

*mutate(b0=coef(mod)[1], # a variável mod, criada anteriormente, possui os coefs.*

*b1=coef(mod)[2], # na ordem, por isso eles são extraídos com []*

*b2=coef(mod)[3],*

*Rsqr=summary(mod)[[9]], # extrai-se r quadrado ajustado do sumário de mod*

*Std.Error=summary(mod)[[6]]) %>% # extrai-se o erro do sumário de mod*

*select(-mod) # agora que as variáveis de interesse foram obtidas, remove-se a variável com os ajustes*

Na figura 11 visualiza-se o objeto contendo os coeficientes do ajuste do modelo volumétrico com casca:

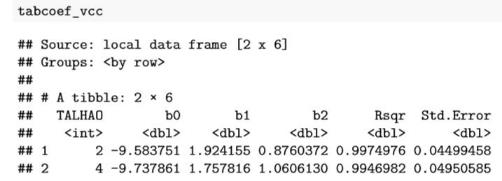


Figura - Coeficientes do ajuste do modelo volumétrico com casca utilizando *dplyr*.

Para se realizar o ajuste para o volume sem casca, basta alterar a variável “VCC” para “VSC”:

*tabcoef\_vsc <- dados\_vol\_arvore %>% # seleção do df*

*group\_by(TALHAO) %>% # seleção da chave*

*do(mod = lm(log(VSC) ~ log(DAP) + log(HT), data =.)) %>% # ajuste do modelo*

*mutate(b0=coef(mod)[1], # a variável mod, criada anteriormente, possui os coefs.*

*b1=coef(mod)[2], # na ordem, por isso eles são extraídos com []*

*b2=coef(mod)[3],*

*Rsqr=summary(mod)[[9]], # extrai-se r quadrado ajustado do sumário de mod*

*Std.Error=summary(mod)[[6]]) %>% # extrai-se o erro do sumário de mod*

*select(-mod) # agora que as variáveis de interesse foram obtidas, remove-se a variável com os ajustes*

Na figura 12 são demonstrados os coeficientes obtidos pelo ajuste do modelo volumétrico sem casca:

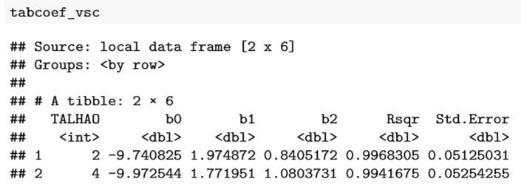


Figura - Coeficientes do ajuste do modelo volumétrico sem casca utilizando *dplyr*.

### Estimação de volume por meio de equação volumétrica utilizando *dplyr*

Para a estimação, pode-se utilizar a função *left\_join*, que diferente de *base::merge*, mantem a ordem original dos dados. Feito isso, utiliza-se *mutate* para estimar o volume, e remove-se os coeficientes com *select*:

*dados\_vol\_arvore <- dados\_vol\_arvore %>% # Dado utilizado*

*left\_join(tabcoef\_vcc, by = "TALHAO") %>% # objeto a ser unido, e chave utilizada*

*mutate(VEQCC = exp(b0 + b1 \* log(DAP) + b2\* log(HT) ) ) %>% # cálculo do volume*

*select(-b0,-b1,-b2,-Rsqr, -Std.Error) %>% #remoção de variáveis adicionadas pelo join*

*left\_join(tabcoef\_vsc, by = "TALHAO") %>%*

*mutate(VEQSC = exp(b0 + b1 \* log(DAP) + b2\* log(HT) ) ) %>%*

*select(-b0,-b1,-b2,-Rsqr, -Std.Error)*

O volume estimado pode ser visto na figura 13:

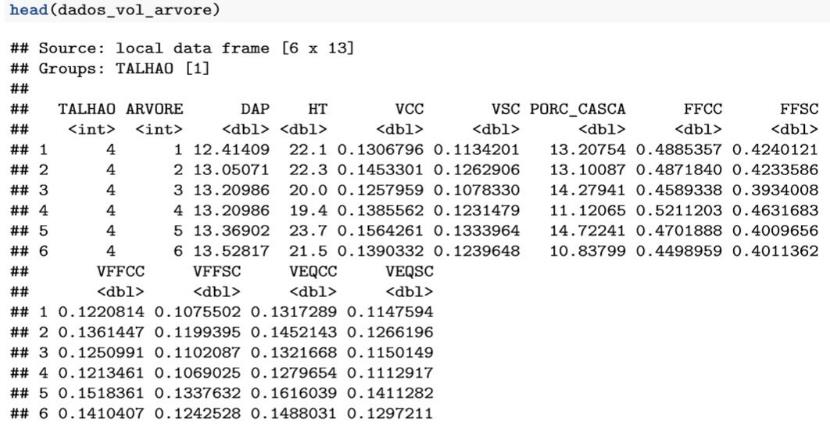


Figura - Volume com casca e sem casca estimado por meio de equação volumétrica utilizando *dplyr*.

## Inventário utilizando *R* base

Quando se carrega dados utilizando as funções padrões do R, colunas contendo informações sobre datas são entendidas como caracteres ou fatores. Como serão feitos cálculos com datas neste script, precisa-se que as datas estejam no formato certo. Para isso, converte-se as colunas que contém estas informações com a função *as.Date*, informando o formato de data utilizado. Portanto, após a importação dos dados, é necessário realizar esta conversão:

*dados\_invt <- read.csv2("dados/dados\_invent.csv")*

*dados\_invt$DATA\_PLANTIO <- as.Date(dados\_invt$DATA\_PLANTIO, format = "%d/%m/%Y")*

*dados\_invt$DATA\_MEDICAO <- as.Date(dados\_invt$DATA\_MEDICAO, format = "%d/%m/%Y")*

Os códigos a seguir realizam o inventário apenas para um talhão. Para realizar a estimativa direto para dois talhões (como feito utilizando *dplyr*), seria necessário um código muito mais avançado do que os apresentados, dificultando o entendimento. Por tanto, caso o objetivo seja realizar o inventário Casual Simples para várias áreas, recomenda-se o uso do pacote *dplyr*.

Na figura 14 é apresentado o dado utilizado como exemplo para o processamento de inventário. Estes seguem o modelo apresentado no Anexo B:

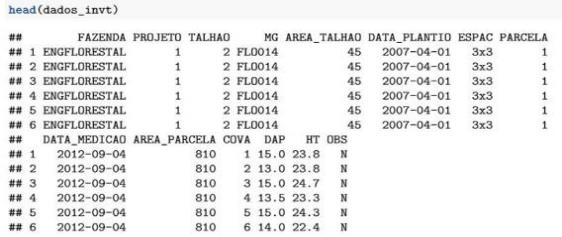


Figura - Dados de inventário utilizados como exemplo.

### Ajuste de modelo hipsométrico e estimação da altura para árvores não medidas – *R* base

O modelo utilizado no ajuste como exemplo foi o de Campos & Leite, que utiliza a variável altura dominante (HD), portanto é necessário estima-la. Para isso, o primeiro passo é selecionar as árvores dominantes, filtra-se os dados utilizando a variável de qualidade do tronco:

*dados\_dom <- dados\_invt[dados\_invt$OBS == "D", ]*

A seguir, cria-se uma nova chave que une talhão e parcela utilizando a função *paste*. Este passo é necessário, pois tem-se dois grupos (talhão e parcela), e a função *by* trabalha apenas com um. Além disso, os nomes das parcelas não são únicos neste exemplo, portanto, precisa-se distingui-los com outra variável, no caso, *TALHAO*.

A função *paste* une duas ou mais variáveis, e o argumento “sep” indica o que irá separa-las; neste caso, une-se as variáveis TALHAO e PARCELA, separadas por underline ("\_"); esta nova variável será chamada de TALHAO\_PAR.

*dados\_dom$TALHAO\_PAR <- paste(dados\_dom$TALHAO, dados\_dom$PARCELA,*

*sep = "\_")*

Para calcular altura dominante média por chave / parcela no *R* base, utiliza-se a função *by*, em conjunto com *as.list*, *do.call* e *as.data.frame*.

*tabhd <- as.data.frame(do.call(rbind, as.list(by(dados\_dom["HT"],*

*dados\_dom$TALHAO\_PAR, colMeans))))*

A função *by* move a chave para a linha, por isso é preciso adiciona-la novamente:

*tabhd <- cbind(rownames(tabhd), tabhd)*

Renomeia-se as variáveis utilizando a função *names*:

*names(tabhd) <- c("TALHAO\_PAR", "HD")*

Remove-se dos nomes das linhas utilizando a função *row.names*:

*row.names(tabhd) <- NULL*

Na figura 15 pode-se visualizar a tabela de altura dominante por parcela:

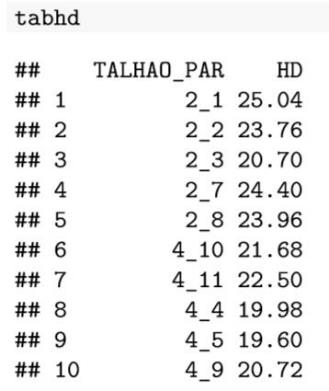


Figura - Tabela de altura dominante gerada utilizando o *R* base.

Em seguida cria-se a chave nos dados originais, pois esta é utilizada como referência para unir os *dataframes*:

*dados\_invt$TALHAO\_PAR <- paste(dados\_invt$TALHAO, dados\_invt$PARCELA,*

*sep = "\_")*

Agora que ambos os *dataframes* possuem a chave, é possível uni-los com a função merge (equivalente ao *procv*); esta possui 4 argumentos:

1. Primeiro objeto a ser unido;
2. Segundo objeto a ser unido;
3. Nome da variável classificatória (chave) que deve ser utilizada como base para unir estes objetos.
4. *all.x* garante que os dados originais sejam mantidos (perfoma um *left join*).

É importante frisar que esta variável deve estar presente nos dois objetos, e que o nome da mesma deve ser idêntico nos dois objetos.

Remove-se as árvores mortas, ou seja, que tem *DAP* com valor *NA*, para não alterar a média:

*dados\_hd <- merge(dados\_invt[!is.na(dados\_invt$DAP), ], tabhd, by = "TALHAO\_PAR", all.x = TRUE)*

Remove-se a chave adicional, pois não é mais necessária:

*dados\_invt$TALHAO\_PAR <- NULL*

Na figura 16 observa-se a variável altura dominante unida aos dados originais:

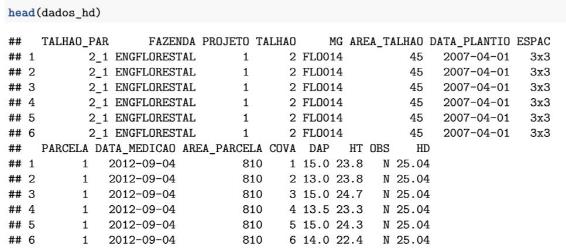


Figura - Variável altura dominante unida aos dados originais por meio do *R* base.

Agora já é possível realizar o ajuste. Primeiramente remove-se as árvores não medidas do objeto dados\_hd, que contém os dados de inventário unido com os valores de altura dominante.

*dados\_adj <-dados\_hd[!is.na(dados\_hd[c("DAP")]) & !is.na(dados\_hd[c("HT")]),]*

Agora calcula-se as variáveis que serão utilizadas na regressão:

*dados\_adj$LN\_HT <- log(dados\_adj$HT)*

*dados\_adj$INV\_DAP <- 1/dados\_adj$DAP*

*dados\_adj$LN\_HD <- log(dados\_adj$HD)*

Utilizando a função *by*, executa-se a regressão linear. São utilizados 4 argumentos:

1. Matriz ou *dataframe* que contém os dados a serem utilizados;
2. Variável classificatória indicando o grupo que será utilizado;
3. Indica função a ser executada (*lm*);
4. Aqui é inserido o argumento "formula", da função *lm*, onde indica-se o modelo que será utilizado.

*aux1 <- by(dados\_adj, dados\_adj$TALHAO, lm, formula = LN\_HT ~ INV\_DAP + LN\_HD)*

Em seguida utiliza-se a função *vapply*, para criar uma tabela que contém apenas os coeficientes, R² ajustado e erro-padrão. A função é composta por 3 argumentos:

1. Matriz ou vetor em que se deseja aplicar uma função em cada elemento;
2. Função a ser aplicada na matriz ou vetor; neste caso, cria-se uma função que extrai os coeficientes, o r quadrado e o erro do ajuste;
3. *template* que fornece os nomes das colunas a serem criadas pela função.

Esta função gera uma matriz com duas linhas e *n* colunas, dependendo do número de coeficientes. Portanto, transpõe-se o seu resultado com a função *t*, para facilitar a sua manipulação.

*regbase <- t(vapply(aux1, function(x) c(coef(x), summary(x)$adj.r.squared, summary(x)$sigma), c(b0 = 0, b1 = 0, b2 = 0, Rsqr = 0, Std.Error = 0)))*

A função *by* salva a chave na linha do objeto; devido a isso, é preciso uni-lo novamente ao *dataframe*:

*tabcoef\_ht <- cbind(TALHAO = as.numeric(rownames(regbase)), as.data.frame(regbase))*

Por fim, remove-se os nomes das colunas:

*row.names(tabcoef\_ht) <- NULL*

Na figura 17 visualiza-se a tabela de coeficientes do ajuste do modelo hipsométrico:

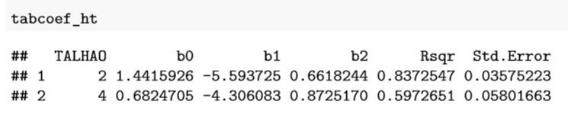


Figura - Coeficientes do ajuste do modelo hipsométrico utilizando o *R* base.

A seguir une-se os dados que contém a variável altura dominante com os coeficientes estimados pela equação hipsométrica:

*dados\_invt2 <- merge(dados\_hd, tabcoef\_ht, by = "TALHAO", all.x = T)*

Para se estimar a altura das árvores não medidas, utiliza-se a função *ifelse*, que funciona de forma semelhante à função *se* do Microsoft Excel. Esta é composta por 3 argumentos:

1. Condicional: “Existe um valor para altura?”;
2. Caso seja verdadeiro, repetir altura;
3. Caso contrário, estimar altura.

*dados\_invt2$HT\_EST <- ifelse(!is.na(dados\_invt2$HT), dados\_invt2$HT,*

*exp(dados\_invt2$b0 + dados\_invt2$b1 \* 1/dados\_invt2$DAP +*

*dados\_invt2$b2 \* log(dados\_invt2$HD)))*

A função *ifelse* é vetorizada, ou seja, ela é aplicada para cada linha dos vetores utilizados. Esta é uma grande vantagem, e dispensa a utilização de loops, como for, por exemplo, agilizando a computação.

Agora remove-se os coeficientes, pois não serão mais utilizados:

*dados\_invt2[c("LN\_HT", "INV\_DAP", "LN\_HD", "b0", "b1", "b2",*

*"Rsqr", "Std.Error")] <- NULL*

Na figura 18 são demonstradas as primeiras e últimas seis linhas do objeto *dados\_invt2*, que contém a variável altura estimada:

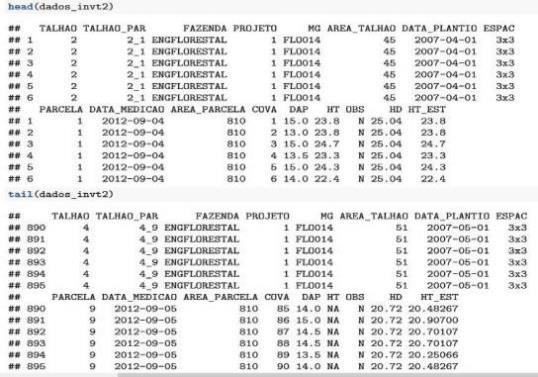


Figura - Altura estimada para as árvores não medidas utilizando *R* base.

Analisando a variável *HT\_EST* percebe-se que apenas as árvores não medidas (*NA*) foram estimadas, e que as medidas tiveram seu valor repetido.

### Estimação de volume e quantificação de parcelas – *R* base

Como exemplo, o volume será estimado utilizando equações de volume. Caso o objetivo seja estimar utilizando o fator de forma, basta alterar a parte do cálculo do volume, multiplicando o fator de forma médio pelo volume do cilindro. Os dados utilizados foram do objeto *dados\_invt2*, apresentado na figura 18.

Primeiro para se estimar o volume com casca, deve-se unir os dados de inventário aos coeficientes da equação de volume, utilizando a função *merge*:

*dados\_invt3 <- merge(dados\_invt2, tabcoef\_vcc, by = "TALHAO", all.x = T)*

A seguir são feitos os cálculos de área seccional, idade e volume.

*# Área seccional:*

*dados\_invt3$AS <- pi \* dados\_invt3$DAP^2/40000*

*# Idade:*

*dados\_invt3$IDADE <- as.numeric( (dados\_invt3$DATA\_MEDICAO - dados\_invt3$DATA\_PLANTIO)/30 )*

*# Volume com casca:*

*dados\_invt3$VCC <- exp(dados\_invt3$b0 + dados\_invt3$b1 \**

*log(dados\_invt3$DAP) + dados\_invt3$b2 \* log(dados\_invt3$HT\_EST))*

Após estimar o volume, remove-se os coeficientes utilizados na estimação:

*dados\_invt3[c("b0", "b1", "b2", "Rsqr", "Std.Error")] <- NULL*

Agora, repete-se o procedimento para o volume sem casca:

*dados\_invt3 <- merge(dados\_invt3, tabcoef\_vsc, by = "TALHAO", all.x = T)*

*dados\_invt3$VSC <- exp(dados\_invt3$b0 + dados\_invt3$b1 \**

*log(dados\_invt3$DAP) + dados\_invt3$b2 \* log(dados\_invt3$HT\_EST))*

Novamente remove-se os coeficientes utilizados:

*dados\_invt3[c("b0", "b1", "b2", "Rsqr", "Std.Error")] <- NULL*

As variáveis estimadas podem ser visualizadas na figura 19:

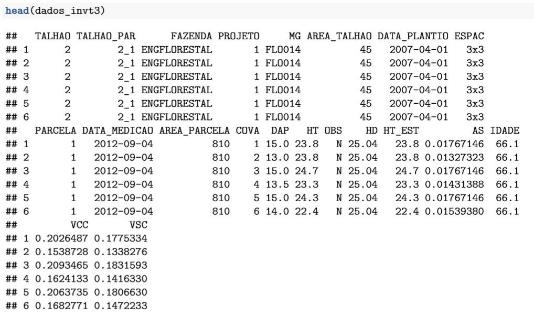


Figura - *Área Seccional*, *Idade*, Volume com e sem casca utilizando *R* base.

A seguir será feita quantificação de parcelas. Os cálculos a seguir consideram a área do talhão em ha, e área da parcela em m². Novamente, precisa-se unir as variáveis *TALHAO* e *PARCELA*:

*dados\_invt3$TALHAO\_PAR <- paste(dados\_invt3$TALHAO, dados\_invt3$PARCELA, sep = "\_")*

Agora calcula-se a média e a soma das variáveis, utilizando a função *by*. A função é utilizada duas vezes, uma para soma, e outra média. Os resultados são unidos em um *dataframe*:

*tab\_invt <- data.frame(*

*do.call(rbind, by(dados\_invt3[,c("IDADE", "AREA\_TALHAO", "AREA\_PARCELA", "DAP","HT\_EST", "HD", "AS")], dados\_invt3$TALHAO\_PAR, colMeans ) ),*

*do.call(rbind, by(dados\_invt3[,c("AS", "VCC", "VSC")], dados\_invt3$TALHAO\_PAR, colSums ) ) )*

A área seccional foi inserida duas vezes, como média e soma, pois a média será utilizada no cálculo do diâmetro quadrático.

A chave foi movida para os nomes das linhas durante este processo, portanto, é necessário transforma-la em coluna novamente. Separa-se a chave em duas variáveis separadas, que são adicionadas ao *dataframe*:

*aux <- data.frame(do.call(rbind, strsplit(row.names(tab\_invt), "\_")))*

*names(aux) <- c("TALHAO", "PARCELA")*

*tab\_invt <- cbind(tab\_invt, aux)*

Converte-se as variáveis criadas para classe numérica novamente:

*tab\_invt$TALHAO <- as.numeric(as.character(tab\_invt$TALHAO)) tab\_invt$PARCELA <- as.numeric(as.character(tab\_invt$PARCELA))*

A seguir arredonda-se algumas variáveis, converte-se os valores de área seccional e volume para hectare, e calcula-se o diâmetro quadrático:

Arredondamento da idade e do *DAP*:

*tab\_invt$IDADE <- round(tab\_invt$IDADE, 1)*

*tab\_invt$DAP <- round(tab\_invt$DAP, 2)*

*# Cálculo do diâmetro quadrático:*

*tab\_invt$q <- round(sqrt(tab\_invt$AS \* 40000/pi), 2)*

Arredondamento da altura total e dominante para duas casas decimais:

*tab\_invt$HT <- round(tab\_invt$HT\_EST, 2)*

*tab\_invt$HD <- round(tab\_invt$HD, 2)*

Conversão para hectare:

*tab\_invt$G <- round(tab\_invt$AS.1 \* 10000/tab\_invt$AREA\_PARCELA, 4) tab\_invt$VCC <- round(tab\_invt$VCC \* 10000/tab\_invt$AREA\_PARCELA, 4) tab\_invt$VSC <- round(tab\_invt$VSC \* 10000/tab\_invt$AREA\_PARCELA, 4)*

Por ﬁm, reordena-se as colunas, reorganiza-se as linhas utilizando a função *order* (colocada dentro do colchete e do lado esquerdo da vírgula) e altera-se a ordem das variáveis, citando os nomes das variáveis utilizando a função *c* no lado direito da vírgula:

*tab\_invt <- tab\_invt[order(tab\_invt$TALHAO, tab\_invt$PARCELA) ,*

*c("TALHAO", "PARCELA", "IDADE", "AREA\_TALHAO", "AREA\_PARCELA", "DAP", "q", "HT", "HD", "G", "VCC", "VSC")]*

Remove-se os nomes das linhas:

*row.names(tab\_invt) <- NULL*

Os dados de inventário por parcela podem ser visualizados na figura 20:

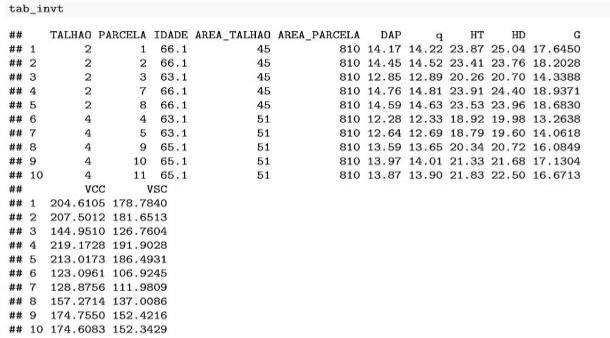


Figura - Dados de inventário por parcela gerados utilizando *R* base.

No *R* base, sem a criação de funções compostas, só é possível fazer um talhão por vez, ou seja, não é possível agrupar o inventário. Portanto, como exemplo, foi utilizado o talhão 2 para as estimativas de inventário, como visto na figura 21:

*x <- na.omit(tab\_invt[tab\_invt$TALHAO == 2, ])*

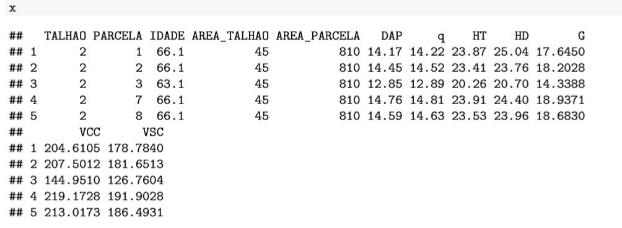


Figura - Dados por parcela por talhão.

### Amostragem Casual Simples – *R* base

A seguir são apresentados os códigos para se realizar um inventário de Amostragem Casual Simples utilizando o *R* base. Os dados utilizados foram os do objeto *x*, demonstrado na figura 21.

Primeiramente, define-se os valores de alpha e erro previamente:

*alpha <- 0.05*

*Erro <- 10*

Em seguida calcula-se variáveis básicas para se calcular o erro-padrão da média:

*n <- length(x$VCC) # número de amostras*

*N <- mean(x$AREA\_TALHAO)/(mean(x$AREA\_PARCELA)/10000) # N*

*Y <- mean(x$VCC) # média do volume*

*CV <- sd(x$VCC)/Y \* 100 # Coeficiente de variância*

O *n* calculado e o erro padrão da média podem ser calculados considerando uma população finita ou infinita, ficando a critério do usuário qual utilizar. Considerando população finita, as linhas de comando são as seguintes:

*t <- qt(alpha/2, df = n - 1, lower.tail = FALSE)*

*t\_rec <- qt(alpha/2, df = ceiling(t^2 \* CV^2/(Erro^2 + (t^2 \* CV^2/N))) - 1, lower.tail = FALSE)*

*n\_recalc <- ceiling(t\_rec^2 \* CV^2/(Erro^2 + (t\_rec^2 \* CV^2/N)))*

*Sy <- sqrt(var(x$VCC)/n) \* (1 - (n/N))*

Poucas alterações ao script são necessárias para se realizar os cálculos considerando população finita:

*t <- qt(alpha/2, df = n - 1, lower.tail = FALSE)*

*t\_rec <- qt(alpha/2, df = ceiling(t^2 \* CV^2/Erro^2) - 1, lower.tail = FALSE)*

*n\_recalc <- ceiling(t\_rec^2 \* CV^2/Erro^2)*

*Sy <- sqrt(var(x$VCC)/n)*

Agora são realizados os cálculos de erro absoluto, erro relativo, volume total estimado e erro total:

*Erroabs <- t \* Sy*

*Erroperc <- Erroabs/Y \* 100*

*Yhat <- Y \* N*

*Erro\_Total <- Erroabs \* N*

Para gerar a tabela de resultados da Amostragem Casual Simples, basta unir os objetos criados anteriormente em um único *dataframe*, e realizar alguns cálculos a mais para a estimação dos intervalos de confiança. Alguns valores foram arredondados utilizando a função *round.*

*tab\_cs <- data.frame(*

*TALHAO = mean(x$TALHAO),*

*IDADE = mean(x$IDADE),*

*n,*

*N,*

*CV,*

*t\_student = t,*

*n\_recalc,*

*Y,*

*Sy = round(Sy, 4),*

*Erroabs = round(Erroabs, 4),*

*Erroperc = round(Erroperc, 4),*

*Yhat,*

*Erro\_Total,*

*IC\_ha\_Inf = Y - Erroabs,*

*IC\_ha\_Sup = Y + Erroabs,*

*IC\_Total\_inf = Yhat - Erro\_Total,*

*IC\_Total\_Sup = Yhat + Erro\_Total )*

A tabela de resultados da Amostragem Casual Simples pode ser visualizada na figura 22:

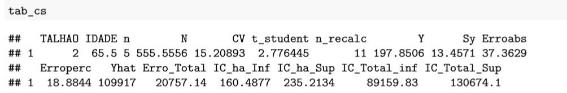


Figura - Resultados da Amostragem Casual Simples considerando população infinita utilizando o *R* base.

### Amostragem Casual Estratificada – *R* base

Para se realizar o inventário por meio da Amostragem Casual Estratificada utilizando *R* base, primeiro define-se quais os dados as serem utilizados:

*x <- tab\_invt*

Em seguida calcula-se variáveis básicas para se calcular o erro-padrão da média.

As variáveis que necessitam ser calculadas por estrato são calculadas utilizando a função *by*; as funções que podem ser utilizadas dentro dela são limitadas (soma, média), por isso em alguns casos foram usadas funções próprias;

Serão feitos vários cálculos por grupo, que serão salvos no objeto *aux*, utilizando a função *by*. Esta função retorna um objeto de classe by, que se comporta como uma lista. Como precisa-se de um *dataframe*, roda-se a função *by* dentro da função *do.call(rbind, )*; esta retorna uma matriz, portanto, insere-se dentro da função *data.frame,* ficando assim: *data.frame( do.call( rbind,by() ) ).*

*x$Nj <- x$AREA\_TALHAO/(x$AREA\_PARCELA/10000)*

*aux <- data.frame(do.call(rbind, by(x, as.factor(x$TALHAO), colMeans)))*

*aux$N <- sum(aux$Nj)*

*aux$nj <- do.call(rbind, as.list(by(x, as.factor(x$TALHAO),*

*function(x) x$n <- nrow(x))))*

*aux$var <- do.call(rbind, as.list(by(x, as.factor(x$TALHAO),*

*function(x) x$var <- var(x$VCC))))*

*aux$Eyj <- do.call(rbind, as.list(by(x, as.factor(x$TALHAO),*

*function(x) x$Eyj <- sum(x$VCC))))*

*aux$Eyj2 <- do.call(rbind, as.list(by(x, as.factor(x$TALHAO),*

*function(x) x$Eyj <- sum(x$VCC^2))))*

*aux <- do.call(rbind, by(aux, as.factor(aux$TALHAO), function(x) {*

*x$Pj = x$Nj/mean(x$N)*

*x$Pj\_Sj2 = x$Pj \* x$var*

*x$Pj\_Sj = x$Pj \* sqrt(x$var)*

*x$Pj\_Yj = x$Pj \* x$VCC*

*return(x) }))*

Após os cálculos por grupo, são feitos os somatórios e coeficiente de variância:

*N <- sum(aux$Nj)*

*EPj\_Sj2 <- sum(aux$Pj\_Sj2)*

*EPj\_Sj <- sum(aux$Pj\_Sj)*

*EPj\_Yj <- sum(aux$Pj\_Yj)*

*Y <- sum(aux$Pj\_Yj)*

*CV <- EPj\_Sj/Y \* 100*

Assim como na Amostragem Casual Simples, o *n* calculado e o erro padrão da média podem ser calculados considerando uma população finita ou infinita, ficando a critério do usuário qual utilizar. Considerando população finita, as linhas de comando são as seguintes:

*t <- qt(alpha/2, df = sum(aux$nj) - 1, lower.tail = FALSE)*

*t\_rec <- qt(alpha/2, df = ceiling(t^2 \* EPj\_Sj^2/((Erro \*Y/100)^2 + (t^2 \* EPj\_Sj2/N))) - 1, lower.tail = FALSE)*

*n\_recalc <- ceiling(t\_rec^2 \* EPj\_Sj^2/((Erro \* Y/100)^2 + (t\_rec^2 \* EPj\_Sj2/N)))*

*nj\_otimo <- ceiling(n\_recalc \* aux$Pj\_Sj/EPj\_Sj)*

*n\_otimo <- sum(nj\_otimo)*

*Sy <- sqrt(sum(aux$Pj\_Sj)^2/sum(aux$nj) - (EPj\_Sj2/N))*

Para se considerar uma população infinita as seguintes modificações são necessárias:

*t <- qt(alpha/2, df = sum(aux$nj) - 1, lower.tail = FALSE)*

*t\_rec <- qt(alpha/2,*

*df = ceiling( t ^2 \* EPj\_Sj^2 / (Erro\*Y/100)^2 ) -1,*

*lower.tail = FALSE)*

*n\_recalc <- ceiling(t\_rec^2 \* EPj\_Sj^2/(Erro \* Y/100)^2)*

*nj\_otimo <- ceiling(n\_recalc \* aux$Pj\_Sj/EPj\_Sj)*

*n\_otimo <- sum(nj\_otimo)*

*Sy <- sqrt(sum(aux$Pj\_Sj)^2/sum(aux$nj))*

A seguir, calcula-se o restante das variáveis da Amostragem Casual Estratificada, como média por hectare, erro absoluto, erro relativo, volume total, volume total por estrato, erro total, e os intervalos de confiança por hectare e total:

*Y <- EPj\_Yj*

*Erroabs <- Sy \* t*

*Erroperc <- Erroabs/Y \* 100*

*Yhat <- N \* Y*

*Yhatj <- aux$Nj \* aux$VCC*

*Erro\_Total <- Erroabs \* N*

*IC\_ha\_Inf <- Y - Erroabs*

*IC\_ha\_Sup <- Y + Erroabs*

*IC\_Total\_inf <- Yhat - Erro\_Total*

*IC\_Total\_Sup <- Yhat + Erro\_Total*

Em seguida, une-se as variáveis calculadas com valores por estrato em uma única tabela, que pode ser visualizada na figura 23:

*tab\_pstrat <- data.frame(*

*TALHAO = aux$TALHAO,*

*IDADE = aux$IDADE,*

*nj = aux$nj,*

*Nj = aux$Nj,*

*Pj = aux$Pj,*

*Eyj = aux$Eyj,*

*Eyj2 = aux$Eyj2,*

*Yj = aux$VCC,*

*Pj\_Sj2 = aux$Pj\_Sj2,*

*Pj\_Sj = aux$Pj\_Sj,*

*Pj\_Yj = aux$Pj\_Yj,*

*CV,*

*t\_rec,*

*nj\_otimo,*

*n\_otimo,*

*Yhatj )*

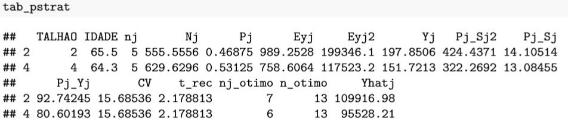


Figura - Resultados por estrato da Amostragem Casual Estratificada considerando população infinita utilizando *R* base.

Por fim, une-se agora as demais variáveis em uma segunda tabela, que pode ser vista na figura 24:

*tab\_strat\_f <- data.frame(*

*t\_student = round(t, 4),*

*Sy = round(Sy , 4),*

*Y = round(Y, 4),*

*Erroabs = round(Erroabs, 4),*

*Erroperc = round(Erroperc, 4),*

*Yhat = round(Yhat, 2),*

*Erro\_Total = round(Erro\_Total, 2),*

*IC\_ha\_Inf,*

*IC\_ha\_Sup,*

*IC\_Total\_inf,*

*IC\_Total\_Sup )*

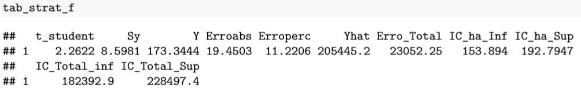
**

Figura - Tabela final para Amostragem Casual Estratificada utilizando *R* base.

### Amostragem Sistemática – *R* base

O inventário utilizando a Amostragem Sistemática foi realizado utilizando a fórmula das diferenças sucessivas para o cálculo do erro padrão da média. O que significa que a ordem de entrada dos dados influencia, por tanto, os dados devem ser inseridos na ordem, para que a estatística seja válida. Os dados utilizados foram os do objeto *x*, demonstrado na figura 21.

No que diz respeito às estimativas de amostragem, a Amostragem Sistemática diferencia-se da Amostragem Casual Simples apenas no cálculo do erro padrão da média. Por tanto, os demais cálculos são idênticos aos apresentados no tópico 4.8.3. Para o cálculo do erro padrão da média para a Amostragem Sistemática, pode-se utilizar:

*Sy <- sqrt( (sum(diff(x$VCC)^2) / (2 \* n \* (n-1) ) ) \* ((N-n)/N) )*

O resultado da Amostragem Sistemática pode ser visto na figura 25:

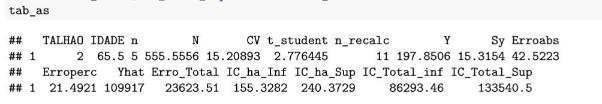


Figura - Tabela de resultados da Amostragem Sistemática utilizando *R* base.

## Inventário utilizando *dplyr*

Em todas as estimativas de inventário utilizando o *dplyr*, o cálculo do inventário é feito por grupo utilizando a função *group\_by*, permitindo que se faça um inventário para cada talhão por exemplo, no caso da Amostragem Casual Simples. Isto pode ser expandido para diferentes variáveis classificatórias, permitindo que se faça inúmeros inventários simultaneamente. Esta é uma grande vantagem pois este tipo de agrupamento utilizando o *R* base requer um conhecimento mais avançado da linguagem, e a construção de códigos complexos. Os dados utilizados seguem o modelo apresentado no anexo B, e podem ser vistos na figura 14.

### Ajuste de modelo hipsométrico e estimação da altura para árvores não medidas – *dplyr*

Para a estimação da altura foi utilizado o modelo de Campos & Leite, que utiliza a variável altura dominante (*HD*). Portanto primeiro estima-se a altura dominante por parcela, e em seguida realiza-se o ajuste do modelo por talhão. Os dados utilizados como exemplo foram os demonstrados na figura 14. A seguir estima-se a altura dominante, e adiciona-se a variável aos dados originais.

*tabhd <- dados\_invt %>% # Definição do df*

*group\_by(TALHAO, PARCELA) %>% # determina as chaves*

*filter(!is.na(HT), !is.na(DAP), OBS == "D") %>% # filtro*

*summarise(HD = mean(HT) ) %>% # média da altura dom / parcela / talhão*

*ungroup # remoção dos grupos*

A tabela gerada pode ser vista na figura 26:

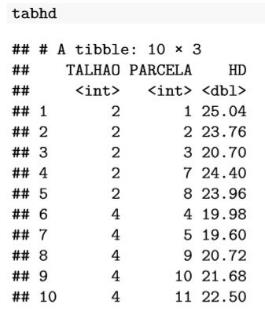


Figura - Altura dominante estimada utilizando *dplyr*.

A seguir é feito o ajuste do modelo hipsométrico de Campos e Leite (2003) por talhão:

*tabcoef\_ht <- dados\_invt %>% # Definição do df*

*filter(!is.na(DAP), !is.na(HT) ) %>% # Remover todos os NAs*

*inner\_join(tabhd, by = c("TALHAO", "PARCELA") )%>% # União dos dados originais com HD*

*mutate(LN\_HT = log(HT),*

*INV\_DAP = 1/DAP,*

*LN\_HD = log(HD) ) %>% # variáveis necessárias para regressão*

*group\_by(TALHAO) %>% # chave*

*do(mod = lm(LN\_HT ~ INV\_DAP + LN\_HD, data =.)) %>% # Ajuste do Modelo*

*mutate(b0=coef(mod)[1],*

*b1=coef(mod)[2],*

*b2=coef(mod)[3],*

*Rsqr=summary(mod)[[9]],*

*Std.Error=summary(mod)[[6]] ) %>%*

*select(-mod)*

Os coeficientes estimados podem ser visualizados na figura 27:

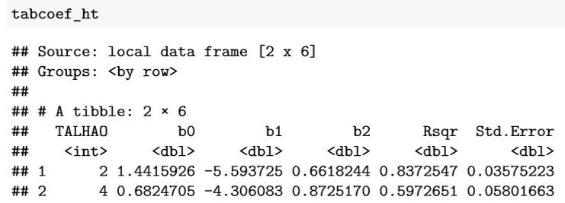


Figura - Coeficientes das equações de altura gerados utilizando *dplyr*.

A seguir, estima-se a altura apenas para as árvores não medidas:

*dados\_invt2 <- dados\_invt %>% # Definição do df*

*filter(!is.na(DAP) ) %>% # Remover árvores mortas*

*left\_join(tabhd, by = c("TALHAO", "PARCELA") ) %>% # União com tabela de HD*

*left\_join(tabcoef\_ht, by = c("TALHAO") ) %>% # união com equações de altura*

*mutate( # criar novas variáveis*

*HT\_EST = ifelse(!is.na(HT), # condicional: Existe um valor para altura?*

*HT, # caso seja verdadeiro, repetir altura*

*exp(b0 + b1\*1/DAP + b2\*log(HD) ) ) ) %>% # caso contrário, estimar*

*select(-b0,-b1,-b2,-Rsqr, -Std.Error) # remoção de variáveis utilizadas*

Percebe-se que a variável *HT\_EST* repetiu as alturas medidas, e estimou as alturas não medidas, como era esperado. Este resultado é idêntico ao obtido quando se utilizou apenas o *R* base, porém, menor número de linhas de comando utilizadas, e menor número de objetos adicionais foram criados. O resultado pode ser visto na figura 28:

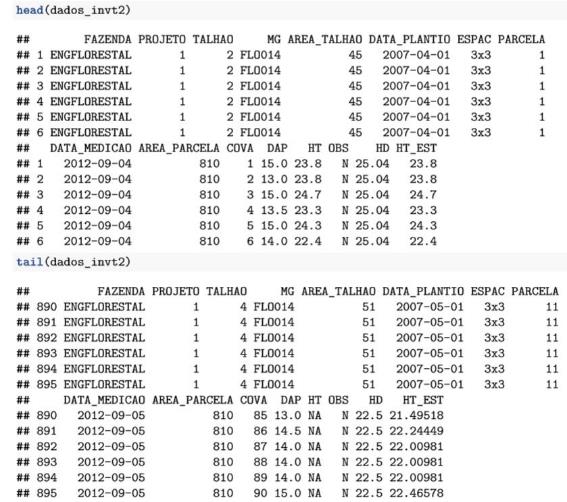


Figura - Altura estimada (*HT\_EST*) para árvores não medidas utilizando *dplyr*.

### Estimação de volume e quantificação de parcelas – *dplyr*

Utilizando o objeto criado no tópico 4.9.1, que contém a variável *HT\_EST*, com as alturas não medidas estimadas, é feito o cálculo das variáveis área seccional, idade, e a estimação do volume. O volume é estimado utilizando os coeficientes obtidos na cubagem, que podem ser vistos nas figuras 11 e 12.

*dados\_invt3 <- dados\_invt2 %>% # remoção das variáveis criadas*

*left\_join(tabcoef\_vcc, by = c("TALHAO") ) %>% # união com coefs da equação de vcc*

*mutate(AS = pi\*DAP^2/40000, # calcular área seccional*

*IDADE = (DATA\_MEDICAO - DATA\_PLANTIO) / 30, # idade*

*VCC = exp(b0 + b1\*log(DAP) + b2\*log(HT\_EST) ) ) %>% # estima-se o volume*

*select(-b0,-b1,-b2,-Rsqr, -Std.Error) %>% # remoção das var criadas*

*# caso não se tenha valores para VSC, deve-se parar por aqui*

*left\_join(tabcoef\_vsc, by = c("TALHAO") ) %>% # união com coefs da equação de vsc*

*mutate(VSC = exp(b0 + b1\*log(DAP) + b2\*log(HT\_EST) ) )%>% # estimar volume sem casca*

*select(-b0,-b1,-b2,-Rsqr, -Std.Error) # remover variáveis*

As novas variáveis criadas podem ser vistas na figura 29:

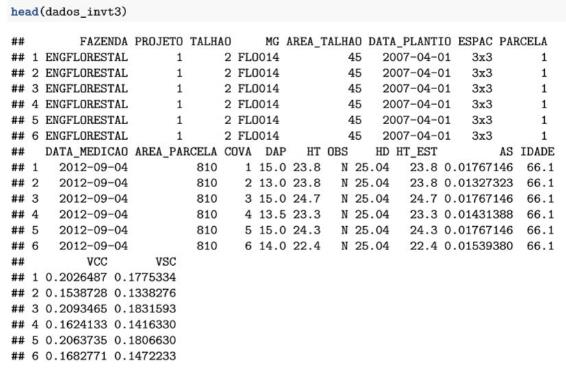


Figura - Cálculo da área seccional (*AS*), idade (*IDADE*) e estimação dos volumes com e sem casca (*VCC* e *VSC*) utilizando *dplyr*.

A quantificação de parcelas pode ser feita utilizando a função *group\_by* para realizar os cálculos por parcela, em conjunto com a função *summarise*:

*tab\_invt <- dados\_invt3 %>%*

*group\_by(TALHAO, PARCELA) %>% # Chaves*

*summarise(*

*IDADE = round(mean(as.numeric(IDADE) )), # Idade em meses*

*AREA\_TALHAO = mean(AREA\_TALHAO), # Área do talhão em ha*

*AREA\_PARCELA = mean(AREA\_PARCELA), # Área da parcela em m²*

*DAP = round(mean(DAP), 2) , # diâmetro a 1,30m do solo em cm*

*q = round(sqrt(mean(AS) \* 40000 / pi), 2) , # diâmetro médio em cm*

*HT = round(mean(HT\_EST), 2) , # Altura total em m*

*HD = round(mean(HD), 2), # Altura dominante em m*

*G = round(sum(AS) \* 10000/ mean(AREA\_PARCELA), 4 ) , # Área basal em m²/ha*

*VCC = round(sum(VCC) \* 10000/mean(AREA\_PARCELA),4),# vol. com casca em m³/ha*

*VSC = round( sum(VSC)\*10000/ mean(AREA\_PARCELA), 4)) # vol. sem casca em m³/ha*

O resultado pode ser visto na figura 30:

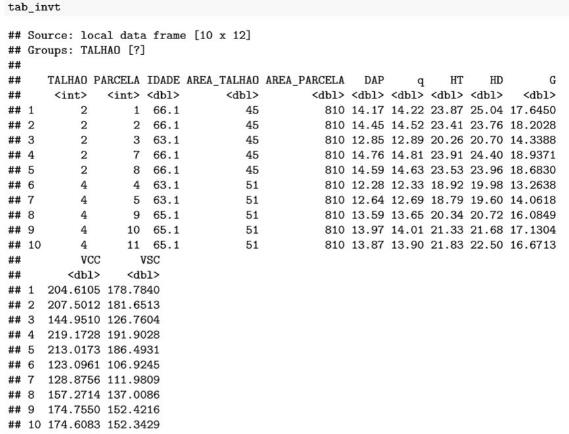


Figura - Dados de inventário por parcela gerados utilizando *dplyr*.

### Amostragem Casual Simples – *dplyr*

A seguir são apresentadas as linhas de comando utilizadas para realizar a Amostragem Casual Simples considerando população finita utilizando o *dplyr*. Foi utilizado o objeto *tab\_invt* apresentadona figura 30, que contém os dados de inventário por parcela.

*alpha <- 0.05*

*Erro <- 10*

*tab\_cs <- tab\_invt %>% # definição do df*

*group\_by(TALHAO) %>% # grupo ou chave*

*summarise(*

*IDADE = mean(IDADE), # usa-se média pois os valores estão repetidos*

*n = n() , # número de amostras*

*N = mean(AREA\_TALHAO) / ( mean(AREA\_PARCELA)/10000 ), # Cálculo do N*

*CV = sd(VCC) / mean(VCC) \* 100,# Cálculo do coeficiente de variação*

*t= qt(alpha/2, df = n-1, lower.tail = FALSE) ,*

*t\_rec = qt(alpha/2, df = ceiling( t^2 \* CV^2 / ( Erro^2 +(t^2 \* CV^2 / N) ) ) - 1, lower.tail= FALSE) , # cálculo do t utilizando o n calculado, de forma direta*

*n\_recalc = ceiling( t\_rec ^2 \* CV^2 / ( Erro^2 +(t\_rec^2 \* CV^2 / N) ) ), # n recalculado*

*Y = mean(VCC, na.rm=T), # Média do volume*

*Sy = sqrt( var(VCC)/n \* (1 - (n/N)) ) , # Erro-padrão da Média*

*Erroabs = Sy \* t , # Erro Absoluto*

*Erroperc = Erroabs / Y \* 100 , # Erro Percentual*

*Yhat = Y \* N, # Media estimada para Área total*

*Erro\_Total = Erroabs \* N, # Erro Estimado Para Área Total*

*IC\_ha\_Inf = Y - Erroabs, # Intervalo de confiança por ha inferior*

*IC\_ha\_Sup = Y + Erroabs, # Intervalo de confiança por ha superior*

*IC\_Total\_inf = Yhat - Erro\_Total, # Intervalo de confiança total inferior*

*IC\_Total\_Sup = Yhat + Erro\_Total) %>% # Intervalo de confiança total superior*

*round(4) # Número de casas decimais*

Para se realizar a Amostragem Casual Simples considerando população infinita, basta fazer pequenas alterações nos estimadores de *n* e *Sy*, de acordo com suas respectivas fórmulas, a linha de comando completa fica da seguinte forma:

*alpha <- 0.05*

*Erro <- 10*

*tab\_cs <- tab\_invt %>% # definição do df*

*group\_by(TALHAO) %>% # grupo ou chave*

*summarise(*

*IDADE = mean(IDADE), # usa-se média pois os valores estão repetidos*

*n = n() , # número de amostras*

*N = mean(AREA\_TALHAO) / ( mean(AREA\_PARCELA)/10000 ), # Cálculo do N*

*CV = sd(VCC) / mean(VCC) \* 100,# Cálculo do coeficiente de variação*

*t= qt(alpha/2, df = n-1, lower.tail = FALSE) ,*

*t\_rec = qt(alpha/2, df = ceiling( t^2 \* CV^2 / Erro^2) - 1, lower.tail = FALSE),*

*n\_recalc = ceiling( t\_rec ^2 \* CV^2 / Erro^2 ), # n recalculado*

*Y = mean(VCC, na.rm=T), # Média do volume*

*Sy = sqrt( var(VCC)/n ), # Erro-padrão da Media*

*Erroabs = Sy \* t , # Erro Absoluto*

*Erroperc = Erroabs / Y \* 100 , # Erro Percentual*

*Yhat = Y \* N, # Média estimada para Área total*

*Erro\_Total = Erroabs \* N, # Erro EStimado Para Área Total*

*IC\_ha\_Inf = Y - Erroabs, # Intervalo de confiança por ha inferior*

*IC\_ha\_Sup = Y + Erroabs, # Intervalo de confiança por ha superior*

*IC\_Total\_inf = Yhat - Erro\_Total, # Intervalo de confiança total inferior*

*IC\_Total\_Sup = Yhat + Erro\_Total) %>% # Intervalo de confiança total superior*

*round(4) # Número de casas decimais*

O resultado da Amostragem Casual Simples pode ser visto na figura 31:

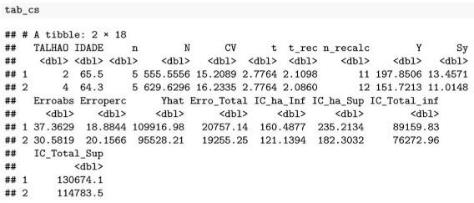


Figura - Amostragem Casual Simples considerando população infinita utilizando *dplyr*.

### Amostragem Casual Estratificada – *dplyr*

A seguir o inventário florestal é processado utilizando a Amostragem Casual Estratificada. Foi utilizado o objeto *tab\_invt* apresentado na figura 30, que contém os dados de inventário por parcela. São geradas duas tabelas, a primeira com resultados por estrato, e a segunda, com os resultados de erro e intervalos de confiança. Primeiro será feito o inventário considerando população finita, e em seguida, considerando população infinita. Para se gerar a tabela por estrato considerando população finita, utiliza-se:

*alpha <- 0.05*

*Erro <- 10*

*tab\_pstrat <- tab\_invt %>%*

*group\_by(TALHAO) %>%*

*summarise(Nj = mean(AREA\_TALHAO) / (mean(AREA\_PARCELA)/10000) ) %>%*

*summarise(N = sum(Nj) ) %>%*

*select(N) %>%*

*cbind(tab\_invt,.) %>%*

*mutate(Nj = AREA\_TALHAO / (AREA\_PARCELA/10000) ) %>%*

*group\_by(TALHAO) %>%*

*summarise(IDADE = mean(IDADE),*

*nj = n() ,*

*Nj = mean(Nj),*

*N = mean(N),*

*Pj = Nj/mean(N),*

*Eyj = sum(VCC),*

*Eyj2 = sum(VCC^2),*

*Yj = mean(VCC, na.rm=T),*

*Pj\_Sj2 = Pj \* var(VCC, na.rm=T),*

*Pj\_Sj = Pj \* sd(VCC, na.rm=T),*

*Pj\_Yj = Pj \* Yj) %>%*

*ungroup %>%*

*mutate( EPj\_Sj2 = sum(Pj\_Sj2),*

*EPj\_Sj = sum(Pj\_Sj),*

*Y = sum(Pj\_Yj),*

*CV = EPj\_Sj / Y \* 100,*

*t = qt(alpha/2, df = sum(nj)-1, lower.tail = FALSE),*

*t\_rec = qt(*

*alpha/2, df = ceiling(*

*t^2 \* EPj\_Sj^2 / ( (Erro\*Y/100)^2 + (t^2 \* EPj\_Sj2 / N ) ) )-1,*

*lower.tail = FALSE),*

*n\_recalc = ceiling(*

*t\_rec^2 \* EPj\_Sj^2 / ( (Erro\*Y/100)^2 + (t\_rec^2 \* EPj\_Sj2 / N ) ) ),*

*nj\_otimo = ceiling(n\_recalc\*Pj\_Sj/EPj\_Sj),*

*n\_otimo = sum(nj\_otimo),*

*Yhatj = Nj \* Yj )*

Para a tabela com valores de erro e intervalo de confiança, utiliza-se as seguintes linhas de comando:

*tab\_strat\_f <- tab\_pstrat %>%*

*summarise(*

*t = mean(t),*

*Sy = sqrt(sum(Pj\_Sj) ^2 / sum(nj) - (mean(EPj\_Sj2) / mean(N) ) ),*

*Y = sum(Pj\_Yj), # Volume médio*

*Erroabs = Sy \* t, # Erro Absoluto*

*Erroperc = Erroabs / Y \* 100, # Erro percentual*

*Yhat = sum(Nj) \* Y, # Volume Total*

*Erro\_Total = Erroabs \* sum(Nj), # Erro Total*

*IC\_ha\_Inf = Y - Erroabs, # Int. de confiança por ha inferior*

*IC\_ha\_Sup = Y + Erroabs, # Int. de confiança por ha superior*

*IC\_Total\_inf = Yhat - Erro\_Total, # Int. de confiança total inferior*

*IC\_Total\_Sup = Yhat + Erro\_Total ) %>% # Int. de confiança total superior)*

*round(4)*

Para se realizar o inventário considerando uma população infinita, utiliza-se, para a tabela por talhão:

*alpha <- 0.05*

*Erro <- 10*

*tab\_pstrat <- tab\_invt %>% # df a ser utilizado*

*group\_by(TALHAO) %>% # grupo*

*summarise(Nj = mean(AREA\_TALHAO) / (mean(AREA\_PARCELA)/10000) ) %>%*

*summarise(N = sum(Nj) ) %>%*

*select(N) %>%*

*cbind(tab\_invt,.) %>%*

*mutate(Nj = AREA\_TALHAO / (AREA\_PARCELA/10000) ) %>%*

*group\_by(TALHAO) %>%*

*summarise(IDADE = mean(IDADE),*

*nj = n() , # número de amostras por estrato*

*Nj = mean(Nj), # N por estrato*

*N = mean(N), # N*

*Pj = Nj/mean(N), # proporção pro estrato*

*Eyj = sum(VCC), # somatório de vcc por estrato*

*Eyj2 = sum(VCC^2), # somatório quadrático de vcc por estrato*

*Yj = mean(VCC, na.rm=T), # média de vcc por estrato*

*Pj\_Sj2 = Pj \* var(VCC, na.rm=T),*

*Pj\_Sj = Pj \* sd(VCC, na.rm=T),*

*Pj\_Yj = Pj \* Yj) %>%*

*ungroup %>%*

*mutate( EPj\_Sj2 = sum(Pj\_Sj2),*

*EPj\_Sj = sum(Pj\_Sj),*

*Y = sum(Pj\_Yj), # média estratificada (ponderada)*

*CV = EPj\_Sj / Y \* 100, # coeficiente de variação*

*t = qt(alpha/2, df = sum(nj)-1, lower.tail = FALSE),*

*t\_rec = qt(alpha/2, df = ceiling(*

*t^2 \* EPj\_Sj^2 / (Erro\*Y/100)^2 )-1, lower.tail = FALSE),*

*n\_recalc = ceiling( t\_rec^2 \* EPj\_Sj^2 / (Erro\*Y/100)^2 ),*

*nj\_otimo = ceiling(n\_recalc\*Pj\_Sj/EPj\_Sj),*

*n\_otimo = sum(nj\_otimo), # n calculado total*

*Yhatj = Nj \* Yj ) # produção por estrato*

A tabela com valores por estrato considerando população infinita por ser vista na figura 32:

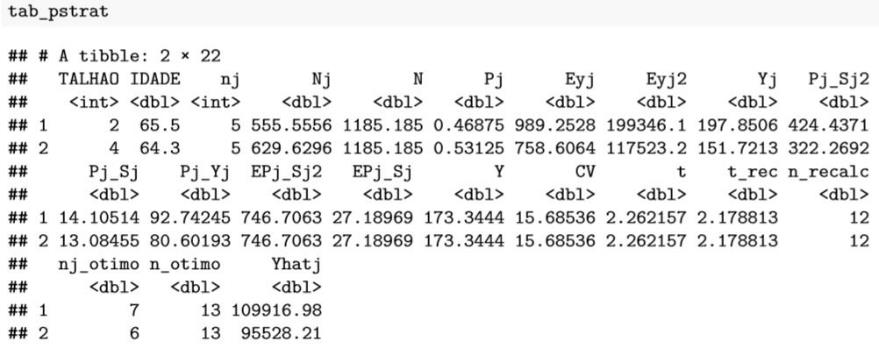


Figura - Resultados por estrato do inventário Casual Estratificado considerando população infinita utilizando *dplyr*.

E para a tabela contendo os valores de erro e intervalo de confiança, utiliza-se:

*tab\_strat\_f <- tab\_pstrat %>%*

*summarise(t = mean(t),*

*Sy = sqrt(sum(Pj\_Sj)^2 / sum(nj) ), # Erro-padrão da média*

*Y = sum(Pj\_Yj), # média de vcc estratificada (ponderada)*

*Erroabs = Sy \* t, # Erro Absoluto*

*Erroperc = Erroabs / Y \* 100, # Erro percentual*

*Yhat = sum(Nj) \* Y, # Volume Total*

*Erro\_Total = Erroabs \* sum(Nj), # Erro Total*

*IC\_ha\_Inf = Y - Erroabs, # Int. de confiança por ha inferior*

*IC\_ha\_Sup = Y + Erroabs, # Int. de confiança por ha superior*

*IC\_Total\_inf = Yhat - Erro\_Total, # Int. de confiança total inferior*

*IC\_Total\_Sup = Yhat + Erro\_Total ) %>% # Int. de confiança total superior)*

*round(4) # Número de casas decimais*

Caso desejado, após a criação da tabela final, pode-se arredondar a tabela por estrato caso desejado. Na figura 33 visualiza-se os resultados finais da Amostragem Estratificada utilizando *dplyr* considerando população infinita:

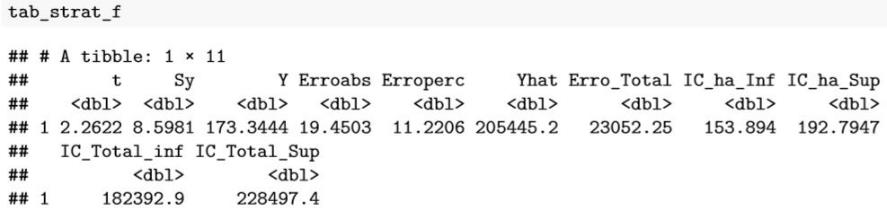


Figura - Resultados finais do inventário Casual Estratificado considerando população infinita utilizando *dplyr*.

### Amostragem Sistemática – *dplyr*

O inventário utilizando a Amostragem Sistemática foi realizado utilizando a fórmula das diferenças sucessivas para o cálculo do erro padrão da média. O que significa que a ordem de entrada dos dados influencia, por tanto, os dados devem ser inseridos na ordem, para que a estatística seja válida.

No que diz respeito estimativas de amostragem, a Amostragem Sistemática diferencia-se da Amostragem Casual Simples apenas no cálculo do erro padrão da média. Portanto, para se realizar a Amostragem Sistemática, basta utilizar o script apresentado no tópico 4.9.3, alterando a linha onde se realiza o cálculo do erro padrão da média, que deve ser substituído pela seguinte linha de comando:

*Sy = sqrt( (sum(diff(VCC)^2) / (2 \* n \* (n-1) ) ) \* ((N-n)/N) )*

Como exemplo foi utilizado o objeto *tab\_invt* apresentado na figura 30, que contém os dados de inventário por parcela. O resultado da Amostragem Sistemática utilizando *dplyr* pode ser visto na figura 34:

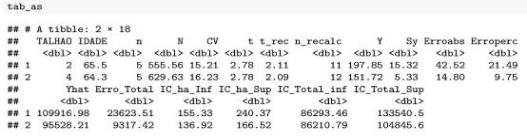


Figura - Resultados do inventário sistemático utilizando *dplyr*.

## Estudo de caso para Amostragem Casual Simples

O exemplo a seguir de aplicação da Amostragem Casual Simples foi retirado do livro de Soares, Neto e Souza (2012). Considerando-se uma floresta com 46,8 ha de área total, têm-se como objetivo do inventário estimar o volume total da população admitindo-se uma precisão de ± 20% e um nível de probabilidade de 95%; realizou-se uma amostragem-piloto de 10 amostras, cujas unidades com área de 3000m² foram selecionadas aleatoriamente na população e seus resultados podem ser vistos na tabela 2:

Tabela - Volume das parcelas sorteadas para o inventário piloto da Amostragem Casual Simples.

|  |  |
| --- | --- |
| **Parcela** | **Volume (m³)** |
| 1 | 41 |
| 2 | 33 |
| 3 | 24 |
| 4 | 31 |
| 5 | 10 |
| 6 | 32 |
| 7 | 62 |
| 8 | 16 |
| 9 | 66 |
| 10 | 25 |

Os dados foram processados utilizando os scripts apresentados. O resultado pode ser visto na figura 35:

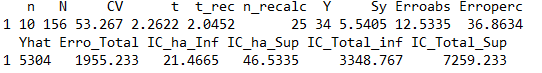


Figura - Resultado do inventário piloto do estudo de caso da Amostragem Casual Simples.

Percebe-se que o número de parcelas (*n*) não atinge o número de parcelas necessário para a precisão inserida (*n\_recalc*), que é 25. Portanto é necessário lançar mais 15 amostras e rodar o inventário novamente. As novas amostras podem ser visualizadas na tabela 3:

Tabela - Parcelas sorteadas para o inventário definitivo com Amostragem Casual Simples.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parcela** | **Volume (m³)** | **Parcela** | **Volume (m³)** | **Parcela** | **Volume (m³)** |
| 1 | 41 | 10 | 25 | 19 | 17 |
| 2 | 33 | 11 | 44 | 20 | 50 |
| 3 | 24 | 12 | 7 | 21 | 38 |
| 4 | 31 | 13 | 57 | 22 | 20 |
| 5 | 10 | 14 | 22 | 23 | 35 |
| 6 | 32 | 15 | 31 | 24 | 31 |
| 7 | 62 | 16 | 40 | 25 | 26 |
| 8 | 16 | 17 | 43 |  |  |
| 9 | 66 | 18 | 27 |  |  |

Utilizando o script apresentado no tópico 4.9.3, o inventário definitivo foi processado, gerando resultados idênticos aos da literatura, como visto na tabela 4:

Tabela - Tabela comparativa entre os principais estimadores para a Amostragem Casual Simples.

|  |  |  |
| --- | --- | --- |
| **Variáveis** | **Script** | **Literatura consultada** |
| Coeficiente de Variância (%) | 45,46 | 45,46 |
| t-student | 2,0639 | 2,064 |
| Média geral (m³ / 0,3 ha) | 33,12 | 33,12 |
| Erro-Padrão da Média (m³ / 0,3 ha) | 2,7595 | 2,76 |
| Erro Absoluto (m³ / 0,3 ha) | 5,6952 | 5,697 |
| Erro Relativo (%) | 17,1957 | 17,20 |
| Volume total estimado (Ŷ) (m³) | 5166,72 | 5166,72 |

## Estudo de caso para Amostragem Casual Estratificada

O exemplo a seguir de aplicação da Amostragem Casual Estratificada foi retirado do livro de Soares, Neto e Souza (2012). Considere uma população florestal com 45,0 ha dividida em três estratos, na qual se realizou um inventário piloto, distribuindo-se sete unidades de amostra de 0,1 ha de área na subárea 1; oito unidades na subárea 2; e sete unidades na subárea 3, e admitiu-se um erro amostral de ± 5%, de acordo com a tabela 5:

Tabela - Parcelas sorteadas para o inventário piloto na Amostragem Casual Estratificada.

|  |  |  |  |
| --- | --- | --- | --- |
| Parcela | Volume (m³) | | |
| Subárea 1 | Subárea 2 | Subárea 3 |
| 1 | 7,9 | 10,2 | 10,65 |
| 2 | 3,8 | 15,25 | 12,15 |
| 3 | 4,4 | 13,4 | 14,6 |
| 4 | 6,25 | 13,6 | 10,9 |
| 5 | 5,55 | 14,2 | 16,55 |
| 6 | 8,1 | 9,85 | 17,9 |
| 7 | 6,1 | 10,2 | 13,35 |
| 8 | - | 11,55 | - |

Utilizando o script apresentado no tópico 4.9.4, o inventário foi processado, obtendo-se os resultados apresentado na figura 36:

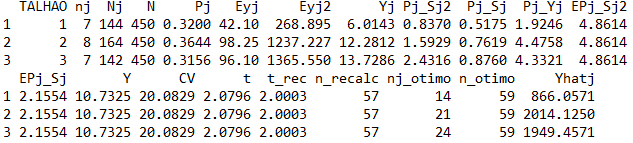


Figura - Resultados do inventário piloto da Amostragem Casual Estratificada.

Percebe-se que o *nj* atual é menor que o *nj\_otimo* em todos os talhões, significando que é necessário lançar mais parcelas em cada talhão para se obter a precisão de ± 5%. O script indica que são necessárias 59 parcelas ao total (*n\_otimo*) para se alcançar a precisão desejada. Porém, este valor não está de acordo com o apresentado na literatura, que é de 57. Isto não é um erro do software, e sim uma questão de arredondamento. Na literatura foram utilizadas apenas 2 casas decimais, e valores decimais na estimativa do *nj* ótimo foram arredondados normalmente, enquanto que no script eles são arredondados sempre para cima (com a função *ceiling*). Este arredondamento para cima assegura que serão lançadas parcelas suficientes para o erro, garantindo uma estimativa mais segura. Isto pode ser alterado pelo usuário do script, caso desejado; bastando alterar a função utilizada, *ceiling*, para *round*, que arredonda normalmente. Além disso, houveram pequenos erros pelos autores nos cálculos de *Pj*, *Pj* *Sj*, e *Pj* *S²j*, levando a pequenas diferenças nos resultados finais.

Com isso, foram lançadas as parcelas necessárias para se atingir o erro estipulado, como visto na tabela 6:

Tabela - Parcelas sorteadas para o inventário definitivo utilizando Amostragem Casual Estratificada.

|  |  |  |  |
| --- | --- | --- | --- |
| Parcela | Volume (m³) | | |
| Subárea 1 | Subárea 2 | Subárea 3 |
| 1 | 7,9 | 10,2 | 10,65 |
| 2 | 3,8 | 15,25 | 12,15 |
| 3 | 4,4 | 13,4 | 14,6 |
| 4 | 6,25 | 13,6 | 10,9 |
| 5 | 5,55 | 14,2 | 16,55 |
| 6 | 8,1 | 9,85 | 17,9 |
| 7 | 6,1 | 10,2 | 13,35 |
| 8 | 6,6 | 11,55 | 14,9 |
| 9 | 7,4 | 9,25 | 9,7 |
| 10 | 5,35 | 11,3 | 15,2 |
| 11 | 5,9 | 13,95 | 13,45 |
| 12 | 4,65 | 12,7 | 12,4 |
| 13 | 4,25 | 10,15 | 14,45 |
| 14 | 8,25 | 14,9 | 13,55 |
| 15 | - | 10,8 | 12,3 |
| 16 | - | 11,55 | 15,65 |
| 17 | - | 13,9 | 14,2 |
| 18 | - | 9,2 | 17,8 |
| 19 | - | 12,45 | 14,8 |
| 20 | - | 11,9 | 9,35 |
| 21 | - | - | 12,6 |
| 22 | - | - | 13,8 |
| 23 | - | - | 15,85 |

Na tabela 7 é feita a comparação entre os resultados obtidos pelos autores, e os resultados obtidos pelo script, no inventário definitivo. Após a justificativa das diferenças presentes nos resultados, fica demonstrado que o software é eficiente para o processamento do inventário utilizando amostragem casual estratificada.

Tabela - Tabela comparativa entre os principais estimadores para a Amostragem Casual Estratificada.

|  |  |  |
| --- | --- | --- |
| **Variáveis** | **Script** | **Literatura consultada** |
| nj subárea 1 | 14 | 14 |
| nj subárea 2 | 21 | 20 |
| nj subárea 3 | 24 | 23 |
| n ótimo | 59 | 57 |
| Média estratificada (m³ / 0,1 ha) | 10,6471 | 10,6470 |
| Erro-Padrão da Média (m³ / 0,1 ha) | 0,2339 | 0,2339 |
| Erro absoluto (m³ / 0,1 ha) | 0,4685 | 0,4678 |
| Erro relativo (%) | 4,4003 | 4,39 |
| Volume total estimado (m³) | 4791,177 | 4791,15 |

## Estudo de caso para Amostragem Sistemática

O exemplo a seguir de aplicação da Amostragem Sistemática foi retirado do livro de Soares, Neto e Souza (2012). para analisar a eficiência do script para a Amostragem Sistemática. O povoamento florestal a ser inventariado possui 10 ha de área, em que foram lançadas 18 parcelas de 0,02 ha de área cada, distribuídas sistematicamente. O volume das parcelas, bem como o sentido de caminhamento, é indicado na figura 37:

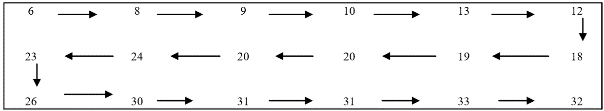


Figura - Volume e sentido de caminhamento das parcelas para o estudo de caso da Amostragem Sistemática.

A tabela de comparação entre os resultados obtidos pelo script e a literatura é apresentada na tabela 8:

Tabela - Tabela comparativa entre os principais estimadores para a Amostragem Sistemática.

|  |  |  |
| --- | --- | --- |
| **Variáveis** | **Script** | **Literatura consultada** |
| Coeficiente de Variância (%) | 44,6505 | 44,64 |
| t-student | 2,1098 | 2,11 |
| Média geral (m³ / 0,02 ha) | 20,2778 | 20,2778 |
| Erro-Padrão da Média (m³ / 0,02 ha) | 0,4008 | 0,4008 |
| Erro Absoluto (m³ / 0,02 ha) | 0,8457 | 0,8457 |
| Erro Relativo (%) | 4,1705 | 4,17 |
| Volume total estimado (m³) | 10138,8889 | 10140 |

# CONCLUSÃO

Os scripts são eficientes para o processamento de inventários florestais, no que diz respeito à precisão dos resultados. O processamento dos dados utilizando o script é mais rápido e eficiente do que utilizando softwares não específicos para a engenharia florestal.

O software R em conjunto com a IDE RStudio é uma ótima alternativa aos softwares de edição de planilhas, para o processamento de inventários florestais e possui grande aplicabilidade na área florestal.

Os scripts criados para o processamento de cubagem e inventário florestal, feitos utilizando o *R* base e o pacote *dplyr* permitem ao usuário decidir a forma de processamento que mais lhe agrada, dando flexibilidade aos scripts.

# REFERÊNCIAS BIBLIOGRÁFICAS

CAMPOS, J. C. C.; LEITE, H. G. **Mensuração florestal:** perguntas e respostas. 3ª. ed. Viçosa: Editora UFV, 2013. 605 p.

CHANG, W. et al. **shiny:** Web Application Framework for R. R package version 0.13.2., 2016. Disponivel em: <https://CRAN.R-project.org/package=shiny>.

FARIAS, C. A. D. et al. Comparação de métodos de amostragem para análise estrutural de florestas ineqüiâneas. **Revista Árvore**, Viçosa, v. 26, n. 5, p. 541-548, 2002.

FREE SOFTWARE FOUNDATION, INC. **GNU Operating System**, 2016. Disponivel em: <https://www.gnu.org/>. Acesso em: 15 junho 2016.

HUSCH, B.; BEERS, T.; KERSHAW, J. **Forest Mensuration**. 4ª. ed. New Jersey: John Wiley & Sons, Inc., 2003.

LEITE, H. G.; ANDRADE, V. C. L. Um método para condução de inventários florestais sem o uso de equações volumétricas. **Revista Árvore**, v. 26, n. 6, p. 321-328, 2002.

LEITE, H. G.; ANDRADE, V. C. L. Importância das variáveis altura dominante e altura total em equações hipsométricas e volumétricas. **Revista Árvore**, v. 27, n. 3, p. 301-310, 2003.

MELLO, J. M. et al. Métodos de amostragem geoestatística para estimativa do número de fustes e volume em plantios de *Eucalyptus grandis.* ***Floresta****, Curitiba, v. 39, n. 1, p. 157-166, 2009.*

*OLIVEIRA, M. L. R. et al. Estimação do volume de árvores de clones de eucalipto pelo método da similaridade de perfis.* ***Rev. Árvore****, Viçosa, v. 33, n. 1, 2009.*

*R CORE TEAM. R: A language and environment for statistical computing.* ***R Foundation for Statistical Computing, Vienna, Austria****, 2016. ISSN 3-900051-07. Disponivel em: <http://www.r-project.org/>.*

*RIBEIRO, A. et al. ESTRATÉGIAS E METODOLOGIAS DE AJUSTE DE MODELOS HIPSOMÉTRICOS.* ***Cerne****, Lavras, v. 16, n. 1, p. 22-31, 2010.*

*RSTUDIO INC.* ***Shinyapps****, 2016. Disponivel em: <http://www.shinyapps.io/>.*

*SCHUMACHER, F. X.; HALL, F. S. Logarithmic expression of timber-tree volume.*

***Jornal of Agricultural Research****, Washington, D. C., v. 47, n. 9, Nov. 1933.*

*SOARES, C. P. B.; NETO, F. D. P.; SOUZA, A. L. D.* ***Dendrometria e Inventário Florestal****. 2ª. ed. Viçosa: UFV, 2012. 272 p.*

*WICKHAM, H.; FRANCOIS, R.* ***dplyr:*** *A Grammar of Data Manipulation. R package version 0.4.3., 2015. Disponivel em: <https://CRAN.R-project.org/package=dplyr>.*

# ANEXOS

## Anexo A: Modelo para dados de cubagem

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TALHAO | ARVORE | DAP | HT | hi | di\_cc | e\_casca |
| 101 | 1 | 13,36 | 23,7 | 1,0 | 15,4 | 2 |
| 101 | 1 | 13,36 | 23,7 | 1,5 | 14,0 | 1 |
| 101 | 1 | 13,36 | 23,7 | 2,0 | 13,52 | 6 |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| *n* | *n* | *n* | *n* | *n* | *n* | *n* |
| *n* | *n* | *n* | *n* | *n* | *n* | *n* |
| *n* | *n* | *n* | *n* | *n* | *n* | *n* |

Em que:

TALHAO: variável classificativa que possui classe maior que a variável ARVORE. Pode ser utilizada para designar um talhão, site, estrato, código genético, espécie, etc;

ARVORE: variável classificativa que indica o número da árvore;

DAP: variável numérica que indica o diâmetro à 1,30 metros do solo, em metros;

HT: variável numérica que indica altura total das árvores, em metros;

hi: variável numérica que indica o comprimento da seção, em metros;

di\_cc: variável numérica que indica o diâmetro da seção, em centímetros;

e\_casca: variável numérica que indica a espessura da casca, em milímetros;

## Anexo B: Modelo para dados de inventário

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TALHAO | AREA\_  TALHAO | DATA\_  PLANTIO | DATA\_  MEDICAO | PARCELA | AREA\_  PARCELA | DAP | HT | OBS |
| 101 | 45 | 01/04/2007 | 04/09/2012 | 1 | 810 | 13,36 | 23,7 | 1,0 |
| 101 | 45 | 01/04/2007 | 04/09/2012 | 1 | 810 | 13,36 | 23,7 | 1,5 |
| 101 | 45 | 01/04/2007 | 04/09/2012 | 1 | 810 | 13,36 | 23,7 | 2,0 |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| *n* | *n* | *n/n/n* | *n/n/n* | *n* | *n* | *n* | *n* | *n* |
| *n* | *n* | *n/n/n* | *n/n/n* | *n* | *n* | *n* | *n* | *n* |
| *n* | *n* | *n/n/n* | *n/n/n* | *n* | *n* | *n* | *n* | *n* |

Em que:

TALHAO: variável classificativa que possui classe maior que a variável ARVORE. Pode ser utilizada para designar um talhão, site, estrato, código genético, espécie, etc;

AREA\_TALHAO: variável numérica que indica a área do talhão, em hectares;

DATA\_PLANTIO: variável em caracteres que indica data de plantio das árvores, no formado dd/mm/aaaa;

PARCELA: variável classificativa que indica o número da parcela;

DATA\_MEDICAO: variável em caracteres que indica data de medição das árvores, no formado dd/mm/aaaa;

PARCELA: variável numérica que indica o número da parcela;

AREA\_PARCELA: variável numérica que indica a área da parcela, em metros;

DAP: variável numérica que indica o diâmetro à 1,30 metros do solo, em metros;

HT: variável numérica que indica altura total das árvores, em metros;

OBS: variável classificatória que indica a qualidade do tronco. Foi utilizado *N* para normal, *D* para dominante e *F* para falha.

**AUTORIZAÇÃO**

Autorizo a reprodução e/ou divulgação total ou parcial do presente trabalho, por qualquer meio convencional ou eletrônico, desde que citada a fonte.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sollano Rabelo Braga

sollanorb@gmail.com

Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM

Campus JK - Rodovia MG 367, n. 5000, Alto do Jacuba

CEP: 39100-000 - Diamantina - MG - Brasil