

Curso Relâmpago de R

# Curso Relâmpago de R

## Objetivo

O objetivo desse curso é fazê-lo adquirir rapidamente certa familiaridade com o software R.

Ao final desse curso você deverá ter atingido dez metas, devendo ser capaz de:

- **1)** saber que o R não é um “aplicativo” mas um *ambiente de trabalho*;
- **2)** saber se o R é o software adequada para você;
- **3)** iniciar, salvar e concluir uma sessão no R;
- **4)** ler arquivos de dados tipo CSV;
- **5)** realizar operações matemáticas simples necessárias para criação e transformação de variáveis;
- **6)** obter estatísticas descritivas de variávies;
- **7)** construir gráficos exploratórios simples de análise de dados;
- **8)** construir modelos lineares clássicos;
- **9)** realizar inferências gráficas e numéricas nos modelos lineares; e
- **10)** conhecer as fontes para você continuar se desenvolvendo no R.

## 1. O Esquema R

---

O R **não é um “aplicativo”** que lhe possibilita através de uma interface gráfica amigável realizar algumas tarefas sem saber exatamente o que está fazendo.

O R é um **ambiente de trabalho para realização de análises estatísticas**.

Como software, suas características principais são:

- **INTERFACE:** o R é uma interface para análises de dados, criando um ambiente de trabalho.
- **INTERATIVIDADE:** essa interface é interativa, isto é, você digita comandos e obtem os resultados.
- **FUNCIONAL:** a linguagem **S**, a linguagem que se fala dentro do R, é uma *linguagem funcional*, isto é, todas as análises e ações são realizadas por *funções*.
- **ORIENTAÇÃO PARA OBJETOS:** a linguagem **S** é uma linguagem de programação orientada para objetos, isto é, todas as *entidades* no ambiente R (dados, análises, gráficos, funções) são efetivamente objetos.
- **MODULAR:** o R é composto por módulos, que são chamados de pacotes (**packages**). O pacote

básico traz a funcionalidade necessária para as análises matemáticas e estatísticas mais usuais. Existem literalmente milhares de outros pacotes para realizar análises específicas nas mais diversas áreas do conhecimento científico.
- **COLABORATIVO:** o R é um esforço mundial de programação em código aberto.

| **Nesse ponto você já deve ter atingido a meta 1.** |
| --- |
| **Se isso não aconteceu, re-leia o material acima e realize uma profunda meditação.** |
| **Se mesmo assim você não alcançou a meta 1, procure um amigo que entenda de computação para traduzir o que você leu acima. Nesse caso, o curso está deixando de ser relâmpago.** |

# 2. Filosofia de trabalho ou o R é para mim?

Do ponto de vista da análise de dados, o R tem uma filosofia que o diferencia radicalmente dos outros softwares estatísticos:

- **ANALISAR DADOS É PROGRAMAR COM DADOS:** toda a análise de dados deve ser pensada como a construção de programa que ao ser executado gera a análise esperada.

**PORTANTO**, reflita nos seguintes pontos antes de prosseguir:

## O R não é para você!

Você pode ter certeza que o R não é para você se:

- Você não quer aprender uma linguagem e realizar análises utilizando comandos.
- Você acredita que análise estatística é um simples protocolo para obter alguns resultados numéricos.
- Você acredita que para cada situação ou conjunto de dados existe A análise estatística correta.
- Você não sabe o que é **análise estatística baseada em modelos** e não está nenhum pouco interessado em saber.

## Não perca seu tempo com o R!

- Se você acredita que basta aprender algumas receitas no R para depois copiá-las quando quiser realizar as mesmas análises, sem entender realmente o que você está fazendo: NÃO PERCA SEU TEMPO COM O R, APRENDA UM SOFTWARE AMIGÁVEL.

## Não se iluda com o R!

- Se você imagina que é possível “tolerar essa coisa de programação com dados” por um tempo,

pois, um dia, alguém vai desenvolver uma interface amigável e aí você já conhecerá o R e tudo será mais fácil.
- Não se iluda!!! Alguém pode até desenvolver uma interface amigável (de fato, já existe pelo menos uma). Mas a equipe de desenvolvedores do R não está interessada em "oficializar" interfaces amigáveis.
- O R não é um software comercial. Não há interesse em atrair usuários que não compartilhem da sua filosofia de trabalho.

**Nesse ponto você já deve ter atingido a meta 2. Mas cuidado com o auto-engano!**

# 3. Iniciando, Gravando e Concluindo

---

Nesse curso assumimos que você já tem o R instalado e funcionando no seu computador. Não será discutida a instalação, pois essa operação é análoga a instalação de qualquer software no sistema operacional que você trabalha. Para obter o R e informações de como instalá-lo, visite o site: http://www.r-project.org/ .

Para iniciar o R você deve:

- **No Ruindows:** realizar um clique duplo no ícone do R no seu "desktop" ou procurar o R na lista de programas instalados a partir do menu "Iniciar".
- **No Linux:** criar um diretório ou se dirigir a um diretório já criado e executar o comando "R".

Apesar das diferenças de interface devido a o sistema operacional, ao iniciar o R você deverá obter uma janela muito parecida com a seguinte janela:

```
R version 2.7.0 (2008-04-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de
ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Área de trabalho anterior carregada]
```

```
>
```

### A Linha de Comando

O R é uma linguagem interativa, ou seja, que permite ao usuário enviar um comando por vez e receber o resultado[1)]. Para isso, usamos a linha de comando, que tem o sinal “>” quando o R está pronto para receber um comando.

Os outros dois estados da linha de comando são o de execução e o de espera para a conlusão do comando. No modo de execução não é exibido nenhum sinal e não é possível digitar outro comando. Você só perceberá isso se der um comando que tenha um tempo de execução muito longo. Experimente o seguinte comando:

```
> for(i in 1:10000) prod(1:i)
>
```

O estado de espera ocorre quando o usuário envia um comando incompleto, o que é indicado por um sinal de “+”:

```
> log(1
+ )
[1] 0
>
```

Na primeira linha, não fechamos os parênteses da função `log` e demos *enter*. O R responde com o sinal de espera (+), indicando que o comando está incompleto. Digitando o parêntese que falta e apertando a tecla *enter* novamente o R retorna o resultado do comando, precedido de um índice numérico.

Ao longo desse curso, os exemplos mostrados dentro do R (como o exemplo acima) são totalmente funcionais. Você deveria ser capaz de digitá-los no seu computador, mas os resultados apresentados pelo R nem sempre serão mostrados nos exemplos.

Para tirar o máximo proveito desse curso relâmpago, você deve repetir todos os comandos exemplificados e não ter medo de experimentar variações dos exemplos apresentados.

### Primeiros Comandos

Se você voltar à mensagem inicial que o R apresenta, verá que ele já lhe ensina alguns comandos:

- `license`[2)] detalha as condições de distribuição do R.
- `contributors` lista a equipe do *time-cerne* de desenvolvimento.
- `citation` ensina como citar o R em trabalhos acadêmicos.
- `demo` inicia uma sessão interativa de demonstração do R.
- `help` para obter uma página de ajuda on-line.
- `help.start` para iniciar um navegador de internet com as páginas de auxílio.
- `q` é o comando para encerrar a sessão e sair do R ( no inglês q = quit ).

### Comando mais utilizado no R: help

Da mesma forma que pessoas pouco letradas pensam que pessoas cultas possuem dicionário só para enfeitar a estante de livros e que raramente os utilizam, os iniciantes no R tendem a pensar que a função `help` é para iniciantes.

Ledo engano!! Até os maiores experts em R utilizam o comando `help` com frequência, pelo simples fato que a quantidade de funções e comandos no R é tão grande que é humanamente impossível alguém conhecer todos eles e com todos os seu detalhes.

Use e abuse da função `help`. Para obter ajuda da própria função utilize o comando:

```
> help( help )
```

Outros exemplos:

```
> # auxílio sobre o símbolo "*"
> help( "*" )
> # auxílio sobre a função "sin" = seno
> help( sin )
> # variante da função "help"
> ?sin
> # auxílio sobre os conjuntos de dados (datasets) que vêm com o R
> help( datasets )
```

### Objetos no R

Como foi dito, no R tudo é considerado um objeto. Até mesmo as funções que você vem utilizando são objetos.

Para ver um objeto no R, basta digitar o seu **NOME**. O nome de uma função é o comando sem os parênteses:

```
> help
> q
> citation
```

No R é fácil criar objetos numéricos[3)]:

```
> x = 2
> x
> y = 7
> y
> z = x * y
> z
> w = x - y
> w
```

Vejamos alguns exemplos com objetos vetoriais:

```
> k = c(1.3, -5, 6.7, 4.8)
> k
> x * k
> m = c(0, 1, 1, 0, 1, 1)
> m
> m * k
```

### A Área de Trabalho

Você deve imaginar que ao iniciar uma sessão do R e criar uma série de objetos, você está fazendo isso numa ÁREA DE TRABALHO (**workspace**).

Para saber quais objetos você criou no seu workspace use a função:

```
> ls()
```

Para apagar os objetos indesejados, utilize a função `rm`, fornecendo os objetos que você deseja apagar:

```
> rm(x, y, z, w)
> ls()
```

É importante lembrar que ao criar os objetos, o R não grava automaticamente o workspace no seu disco. Fica tudo na memória do computador. Basta uma interrupção momentânea de energia e você perde tudo. Por isso, é importante lembrar de periodicamente gravar o seu workspace, principalmente nas sessões mais longas:

```
> # grava o seu workspace com o nome genérico: ".RData"
> save.image()
> # grava o seu workspace com o nome indicado
> save.image(file="minha_sessao_inicial_de_R.RData")
```

Lembre-se que na maioria dos sistemas operacionais, arquivos cujo o nome se inicia com um ponto (como “.RData”) são arquivos ocultos!

| **Nesse ponto você deve ter alcançado a meta 3.** |
| --- |
| **Inseguro? Experimente criar uma nova área de trabalho, criando variações das operações acima e gravando o workspace. Inicie uma nova sessão no workspace que foi gravado!** |

# 4. Lendo Dados para Dentro do R: Formato CSV

O R é um ambiente para análise de dados. Não é um ambiente para **digitação** ou **organização** dos seus dados.

O aplicativo mais utilizado para digitar e organizar os dados são as **planilhas eletrônicas**, como o Excell e assemelhados. As planilhas conseguem gravar os dados em vários formatos, além do formato nativo da planilha[4)].

## Formato CSV

No caso do R, o melhor é gravar os dados no formato CSV (CSV = *Comma-Separeted Values*).

O formato CSV consiste em uma única tabela (spreadsheet) da sua planilha eletrônica, onde os valores são gravados linha-a-linha, sendo que numa mesma linha os valores são separados por vírgulas.

O formato UNIVERSAL de tabela de dados para análise estatística é o seguinte:

- cada **LINHA** é uma observação,
- cada **COLUNA** é uma variável ou atributo que foi tomado em cada observação.

No formato CSV, cada linha será uma observação e as colunas serão separadas por vírgulas.

## Lendo um Arquivo CSV

Como exemplo trabalharemos com um arquivo que apresenta três parcelas, cada uma em um "caxetal" diferente. As primeiras linhas desse arquivo formam a seguinte tabela:

| local | parcela | arvore | fuste | cap | h | especie |
|---|---|---|---|---|---|---|
| chauas | 1 | 1 | 1 | 210 | 80 | Myrcia sulfiflora |
| chauas | 1 | 3 | 1 | 170 | 80 | Myrcia sulfiflora |
| chauas | 1 | 4 | 1 | 720 | 70 | Syagrus romanzoffianus |
| chauas | 1 | 5 | 1 | 200 | 80 | Tabebuia cassinoides |
| chauas | 1 | 6 | 1 | 750 | 170 | indet.1 |
| chauas | 1 | 7 | 1 | 320 | 80 | Myrcia sulfiflora |
| chauas | 1 | 8 | 1 | 480 | 160 | Tabebuia cassinoides |
| chauas | 1 | 9 | 1 | 240 | 140 | Tabebuia cassinoides |

Grave o arquivo no diretório que estiver trabalhando com o R: exemplo-caixeta.csv.

Esse arquivo pode ser visualizado em qualquer editor de textos, pois o formato CSV é um formato texto:

```
"local","parcela","arvore","fuste","cap","h","especie"
```

```
"chauas",1,1,1,210,80,"Myrcia sulfiflora"
"chauas",1,3,1,170,80,"Myrcia sulfiflora"
"chauas",1,4,1,720,70,"Syagrus romanzoffianus"
"chauas",1,5,1,200,80,"Tabebuia cassinoides"
"chauas",1,6,1,750,170,"indet.1"
"chauas",1,7,1,320,80,"Myrcia sulfiflora"
"chauas",1,8,1,480,160,"Tabebuia cassinoides"
"chauas",1,9,1,240,140,"Tabebuia cassinoides"
```

Note nos seguintes pontos:

- A primeira linha do arquivo contem o nome das variáveis ou nome das colunas da tabela.
- Os valores de cada variável (coluna) estão seprados por vírgulas.
- As “palavras” estão envoltas em aspas duplas (“).

Para ler esse arquivo no R, basta utilizar a função `read.csv`.

Cuidado!! Se você digitar simplesmente:

```
> read.csv(file="exemplo-caixeta.csv")
```

as linhas do arquivo aparecerão na tela mas não serão gravadas em nada.

É necessário gravar a leitura do arquivo num **objeto**:

```
> cax = read.csv(file="exemplo-caixeta.csv")
```

Agora se você digitar o nome do objeto:

```
> cax
```

o R lhe apresentará todas as 198 linhas de dados!!

Para visualizar apenas as primeiras linhas do dataframe (objeto de dados) use o comando head:

```
> head(cax)
   local parcela arvore fuste cap   h                especie
1 chauas       1      1     1 210  80      Myrcia sulfiflora
2 chauas       1      3     1 170  80      Myrcia sulfiflora
3 chauas       1      4     1 720  70 Syagrus romanzoffianus
4 chauas       1      5     1 200  80   Tabebuia cassinoides
5 chauas       1      6     1 750 170                indet.1
6 chauas       1      7     1 320  80      Myrcia sulfiflora
```

## Problema da Planilha ou do CSV

Um grande problema surge quando se utiliza uma planilha eletrônica onde o **separador decimal** também é a vírgula.

Nesse caso, o arquivo CSV será uma grande confusão, pois a vírgula não só delimitará cada coluna da tabela de dados como também indicará o separador decimal do números que não forem inteiros.

Dois aspectos devem ser considerados:

- PRIMEIRO: jamais digitar dados com marcador decimal (no Brasil a vírgula). Essa é uma regra importante, pois a maior parte dos erros de digitação envolvem a utilização do marcador decimal na definição dos números. Se você mediu o diâmetro de uma árvore em centímetros com uma casa decimal, anote os dados em milímetros.
- SEGUNDO: se a digitação com marcador decimal se tornou inevitável[5)] e você utilizou a **vírgula** como marcador decimal, ao gravar o arquivo CSV, procure fazer com que a planilha utilize um outro símbolo para separação de valores, como por exemplo o **ponto-e-vírgula (;)**. A maneira de fazer isso depende da planilha que você estiver utilizando.

O arquivo exemplo-caixeta-2.csv é um exemplo de arquivo CSV com ponto-e-vírgula como separação de valores. No R, esse arquivo deverá ser lidos com o argumento `sep` definindo o símbolo usado na separação de valores:

```
> cax2 = read.csv(file="exemplo-caixeta-2.csv",sep=";")
> head( cax2 )
```

A página de auxílio da função `read.csv` detalha diferentes formas de leitura de dados que podem ser utilizadas no R.

# 5. Manipulando e Criando Variáveis

---

## Entendendo um Data Frame no R

O comando head sempre apresenta as primeiras linhas da **tabela de dados** (*data frame*).

- As colunas da tabela (variáveis) são mostradas pelos nomes que estavam na primeira linha do arquivo CSV.
- As linhas da tabela (observações) são geralmente numeradas. Note que a primeira coluna de números não é uma variável! Ela indica cada linha do arquivo CSV, ou seja cada observação. No R, as linhas do data frame (observações) também devem ter nomes únicos. Se o usuário não os fornece, o R simplesmente os nomeia segundo a numeração da ordem em que os dados são lidos.

Se quisermos apenas uma variável (coluna) desse data frame, basta unir o nome do data frame (`cax`) ao nome da coluna desejada com o símbolo especial **“$”**:

```
> cax$local
> cax$cap
```

## Novas Variáveis no Data Frame

Novas variáveis (colunas) podem ser criadas também utilizando o símbolo $. Por exemplo:

```
> # constante universal PI
> pi
[1] 3.141593
> # cálculo do DAP (cm) a partir do CAP (mm)
> cax$dap = (cax$cap/10) / pi
> # cálculo da área transversal (m2) a partir do DAP (cm)
> cax$g = (pi/4) * (cax$dap/100)^2
> # cálculo do volume cilíndrico a partir da área transv. e da altura (dm)
> cax$vol.cilindrico = cax$g * (cax$h/10)
>
> head(cax)
```

Note que:

- A palavra `pi` no R está reservada para representar a constante universal pi.
- Ao utilizar a expressão `cax$dap =` estamos criando uma nova variável (coluna) no dataframe `cax` com o nome `dap`, cujos valores serão o resultado da expressão matemática apresentada após o sinal de igualdade. O mesmo acontece com as colunas `g` e `vol.cilindrico`.

Não há como **apagar** as variáveis de dentro de um data frame! É necessário criar um novo data frame apenas com as variáveis desejadas, mas isso não é assunto para um curso relâmpago!

| **As metas 4 e 5 foram atingidas?** |
| --- |
| **Se você está inseguro, repita todos os passos dos itens 4 e 5 com os seus próprios dados!** |

# 6. Descrevendo as Observações

---

## Contagens

A forma mais simples de descrever quantitativamente observações é agrupá-las em categorias e contar quantas observações pertence a cada categoria.

No R a forma mais direta de obter contagens (frequências) é através da função `table`. Tomando como exemplo o dataframe `cax`, podemos nos perguntar quantas árvores foram observadas em cada caixetal (variável `local`):

```
> table(cax$local)
```

Também é interessante saber o número de árvores por `local` e `parcela`:

```
> table(cax$local, cax$parcela)
```

**Observação:** note que os **argumentos** de uma função são separados por vírgula (,)

Podemos verificar a abundância de cada espécie (`especie`) em cada caixetal (`local`):

```
> table( cax$especie, cax$local )
```

#### Gráficos de Contagem

Dados de contagem também podem ser apresentado na forma de **gráficos de barra**:

```
> barplot( table(cax$local) )
```

**Observação:** veja que no R você pode construir um comando chamando função dentro de função em vários níveis. No exemplo acima, o resultado da função `table` foi colocado como argumento para a função `barplot`.

Formas alternativas de construir esses gráficos são:

```
> barplot( table(cax$fuste) )
> plot( table(cax$fuste) )
```

Um gráfico de abundância das espécies presentes nos três caixetais:

```
> # Define margens do gráfico, aumentando a esquerda
> par( mar=c(5,10,4,2) )
> # Gráfico horizontal c/ nomes horizontais
> barplot(sort(table(cax$especie)), horiz=T, las=1, xlab="Abundância")
> # Fecha a janela gráfica
> dev.off()
```

**Observação:** no exemplo acima temos um comando com três funções *aninhadas* (`table` dentro de `sort`, que está dentro de `barplot`).

Embora seja convencional apresentar o gráfico de abundância com barras, um gráfico na forma de pontos é de construção mais simples, sendo mais informativo:

```
> dotchart( sort(table(cax$especie)), xlab="Abundância" )
```

## Sumário de Variáveis

A função `summary` retorna uma conjunto de estatísticas descritivas (**sumário**) de todas as variáveis de um data frame de acordo com o seu tipo:

```
> summary(cax)
```

Note que para as variáveis `parcela` e `arvore` esse sumário não faz sentido, uma vez que elas são simples variáveis indentificadoras da parcela e da árvore[6)].

Mas o sumário também pode ser obtida para cada variável individualmente:

```
> summary(cax$dap)
> summary(cax$h)
> summary(cax$especie)
```

## Estatísticas Descritivas

O R também possui funções para as diversas estatísticas descritivas de variáveis quantitativas:

| Estatística Descritiva | Nome da Função |
|---|---|
| Média | `mean` |
| Mediana | `median` |
| Mínimo | `min` |
| Máximo | `max` |
| Amplitude de variação | `range` |
| Quartis e quantis | `quantile` |
| Distância Interquartil (*Inter Quarter Range*) | `IQR` |
| Variância | `var` |
| Desvio padrão (*Standard Deviation*) | `sd` |
| Desvio abosluto mediano (*Mean Absolut Deviation*) | `mad` |

```
> mean(cax$dap)
> mdap = mean(cax$dap)
> mdap
>
> median(cax$dap)
> min(cax$dap)
> max(cax$dap)
> range(cax$fuste)
> quantile(cax$h)
> IQR(cax$h)
> var(cax$h)
> sd(cax$h)
```

# 7. Gráficos Exploratórios

O R é um ambiente de trabalho onde a análise gráfica de dados é de fácil execução. Entretanto, é necessário diferenciar dois tipos de gráficos:

- **Gráficos para análise de dados:** são gráficos simples que permitam visualizar o mais claro possível padrões presentes nos dados. Esses gráficos são construídos rapidamente no R e as formas de construí-los permitem inúmeras interações com os **elementos de informação** nos gráficos.
- **Gráficos prontos para apresentação:** são construídos para inclusão em documentos e trabalhos técnicos e científicos, como forma de ilustrar resultados e conclusões. Gráficos de apresentação são mais elaborados. Sua construção no R exige mais tempo e conhecimento, pois o R não oferece recursos interativos para manipular os **elementos pictoriais** dos gráficos.

Vejamos alguns gráficos para análise exploratória de dados.

## Histogramas

Histogramas são gráficos tradicionais na análise exploratória de dados, pois nos apresentam um gráfico da distribuição de probabilidade da variável analisada.

```
> hist( cax$dap )
> hist( cax$h, col="red" )
> hist( cax$h, col="blue", probability=T )
```

Uma possibilidade de gráfico que o R permite é adicionar uma curva de **densidade probabilística** ao histograma, para melhor estudar o comportamento da variável.

```
> hist( cax$dap, probability=T , col="blue")
> lines( density(cax$dap) , col="red")
```

Um **gráfico tipo texto** análogo ao histograma é o tradicional **gráfico de ramo-folha** da análise exploratória de dados:

```
> stem(cax$dap)
> stem(cax$h)
```

## Boxplot

Os boxplots são gráficos de uso frequente para se estudar o comportamento das variáveis. Sua construção no R é direta e simples:

```
> boxplot( cax$dap )
> boxplot( dap ~ local, data=cax )
```

Transformar um gráfico de análise em um gráfico de apresentação demanda o conhecimento sobre as ferramentas gráficas presentes no R:

```
> # Altera as margens da janela gráfica
> par( mar=c(5,10,4,2) )
> # Boxplot
> boxplot( dap ~ especie, data=cax , horizontal=T, las=1)
> # Fecha a janela gráfica
> dev.off()
```

### Gráfico de Dispersão

A função `plot` é a função básica para construção de gráficos de dispersão para duas variáveis quantitativas:

```
> plot( cax$dap, cax$h )
> scatter.smooth( cax$dap, cax$h )
```

| **Você se sente seguro em relação às metas 6 e 7?** |
| --- |
| **Não? Então utilize os seus próprios dados para calcular estatísticas descritivas e construir gráficos exploratórios .** |

# 8. Modelos Lineares

---

Utilizaremos alguns modelos lineares para estudar a relação entre o DAP (variável `dap`) e a altura total (variável `h`) das árvores dos caxetais.

A função `lm` (*linear model*) é a função utilizada para **construir** um modelo linear. O primeiro passo é construir um modelo linear gravando-o num objeto na área de trabalho.

```
> hipso1 = lm( formula = h ~  dap, data=cax )
> hipso1 = lm( h ~  dap, data=cax )
```

O primeiro argumento da função `lm` é uma **fórmula estatística** (`formula = h ~ dap`) que descreve a variável `h` como variável resposta e a variável `dap` como variável preditora. Ela deve ser lida da seguinte forma: **modele `h` como uma função linear de `dap`**.

O segundo argumento (`data=cax`) define que as variáveis da fórmula estão no data frame `cax`.

Vejamos o objeto `hipso1`:

```
> hipso1
```

Mas que decepção!! O R não fez nada?

# 9. Inferência sobre Modelos Lineares

O R fez muita coisa! Ele construiu um modelo linear e o gravou no objeto `hipso1`. A questão agora é o que você deseja saber desse objeto que é um modelo linear?

O primeiro interesse é analisar o comportamento dos resíduos do modelo linear, para verificar se o modelo é apropriado aos dados. Para isso basta utilizar a função `plot` com o objeto `hipso1`.

```
> plot( hipso1 )
```

O R entra num modo interativo diferente, apresentando sequencialmente, à medida que você tecle <Return>, um gráfico diferente que lhe permite avaliar o comportamento dos resíduos do modelo. Esses gráficos são os gráficos necessários para se verificar as pressuposições básicas do modelo linear clássico.

E para se fazer inferência sobre as estimativas dos coeficientes de regressão e sobre a qualidade do ajuste do modelo? Se utiliza as funções `summary` e `anova`:

```
> summary( hipso1 )
> anova( hipso1 )
```

## Cada Modelo um Objeto

Você pode ajustar quantos modelos você desejar para estudar a mesma relação.

Podemos considerar que a relação entre DAP e altura é linear na escala logarítmica, assim um modelo apropriado seria:

```
> hipso2 = lm( log(h) ~ log(dap) , data=cax)
> plot(hipso2)
> summary(hipso2)
```

Ou então podemos verificar o modelo conhecido na Mensuração Florestal como *Modelo Schumacher*:

```
> hipso3 = lm( log(h) ~ I(1/dap) , data=cax)
> plot(hipso3)
> summary(hipso3)
```

Alguém aprecia parábolas ?

```
> hipso4 = lm( h ~ dap + I(dap^2) , data=cax)
> plot(hipso4)
> summary(hipso4)
```

Como podemos visualizar todos esses modelos junto com a relação DAP - altura ?

```
> scatter.smooth( cax$dap, cax$h)
> hipso1
> curve( 40.182 + 5.218*x, 0, 26, col="blue", add=T )
>
> hipso2
> curve( exp( 3.2031 + 0.5574*log(x) ), 0, 26, col="red", add=T )
>
> hipso3
> curve( exp( 4.639 - 1.871/x ), 0, 26, col="darkgreen", add=T )
>
> hipso4
> curve( 10.4109 + 11.1879*x - 0.2325*x^2, 0, 26, col="orange", add=T )
```

A abordagem mais apropriada é consideramos que cada caxetal talvez tenha uma relação altura-DAP diferente:

```
> hipso.plus = lm( h ~ dap * local, data=cax)
> plot( hipso.plus )
> summary( hipso.plus )
```

| **As metas 8 e 9 foram alcançadas?** |
| --- |
| **Não? Construa mais modelos com os seus próprios dados.** |

# 10. Para onde ir a partir daqui?

---

Bem! Chegamos a nossa meta final: as fontes para continuar se desenvolvendo no R.

- **http://www.r-project.org/** é o site do Projeto R e possui as informações básicas para continuar aprendendo o R. Lá você encontrará as últimas versões do R e terá acesso aos **milhares** de pacotes que acrescentam funcionalidade extra ao R.
- No site do projeto R, você encontrará também os manuais básicos que são as referências oficiais para o R:
  - An Introduction to R
  - The R language definition
  - Writing R Extensions
  - R Data Import/Export
  - R Installation and Administration
  - R Internals
  - The R Reference Index
- Os livros de **John M. Chambers** também são referências básicas para o R:

    - Chambers, J.M. 2008 **Software for Data Analysis: Programming with R**. New York: Springer.
    - Chambers, J.M. 2004 **Programming with Data: A Guide to the S Language**. New YOrk: Springer.

- Um livro de R escrito para ecologistas:
    - Crawley, Michael J. 2007. **The R Book**. New York: John Wiley.

- Existe muitos sites onde você poderá aprender mais sobre o R, basta fazer uma busca simples na internet. Dois sites **em português** que você pode continuar aprendendo o R são:
    - Uso da Linguagem R e
    - Uso da Linguagem R para Análise de Dados Ecológicos.

Embora ambos sejam semelhantes no conteúdo, o segundo site é mais didático e está mais atualizado.

---

Essa lista está longe de ser exaustiva!! Existem muitos outros recursos para aprender o R, essas são apenas algumas dicas iniciais.

| **Dez metas alcançadas?** |
| --- |
| **Cada meta é um passo. Todo caminho se percorre passo a passo. Continue caminhando!** |
| ***Be an useR ! Be happy !*** |

# Autor

João Luís Ferreira Batista

Laboratório de Mensuração e Biometria Florestal
Centro de Métodos Quantitativos
Departamento de Ciências Florestais
Escola Superior de Agricultura "Luiz de Queiroz"
**UNIVERSIDADE DE SÃO PAULO**

1)
é possível também de executar um lote de comandos, mas neste wiki trabalharemos apenas com o modo interativo.
2)
Note que para executar o comando é necessário digitar o nome da função seguido de parênteses: `license()`
3)
Nota: a maioria dos comandos nesse curso são mostrados **SEM** os resultados apresentados pelo R
4)
No Excell, o formato nativo é o XLS
5)
Você vai se arrepender disso depois!!

6)

Embora codificadas como números essas variáveis são variáveis nominais

From:
http://insilvaarbores.com.br/dokuwiki/ - **In Silva, Arbores ...**

Permanent link:
**http://insilvaarbores.com.br/dokuwiki/doku.php?id=pt:cursos_online:r_relampago:start&rev=1682344058**

Last update: **2023/04/24 13:47**