- Tutorial
- Exercícios
- Apostila

4. Análise Exploratória de Dados

Podemos conduzir a análise exploratória de dados de duas formas:

- análise numérica: computar estatísticas descritivas;
- análise gráfica: explorar o comportamento e a relação entre as variáveis através de gráficos.

Nesse tópico utilizaremos os arquivos de dados:

- Levantamento em caixetais: caixeta.csv
- Dados de biomassa de árvores: esaligna.csv
- Inventário em florestas plantadas:egrandis.csv

Estatísticas Descritivas

A forma mais direta de se obter um resumo estatístico das variáveis num `data.frame` é através da função `summary`. Ela apresenta estatísticas descritivas para as variáveis numéricas.

```
> cax = read.csv("caixeta.csv", header=TRUE, as.is=TRUE)
>
> summary(cax)
    local              parcela          arvore          fuste
cap               h
 Length:1027        Min.   :1.000   Min.   :  1.0   Min.   : 1.000   Min.
:  20.0   Min.   :  5.00
 Class :character   1st Qu.:2.000   1st Qu.: 51.0   1st Qu.: 1.000   1st
Qu.: 190.0   1st Qu.: 60.00
 Mode  :character   Median :3.000   Median : 99.0   Median : 1.000   Median
: 270.0   Median : 90.00
                    Mean   :2.821   Mean   :108.5   Mean   : 1.711   Mean
: 299.7   Mean   : 90.28
                    3rd Qu.:4.000   3rd Qu.:159.0   3rd Qu.: 2.000   3rd
Qu.: 360.0   3rd Qu.:110.00
                    Max.   :5.000   Max.   :291.0   Max.   :11.000   Max.
:2100.0   Max.   :480.00
   especie
 Length:1027
 Class :character
 Mode  :character



>
```

Outras estatísticas devem ser calculadas individualmente pelo analista:

```
> resumo1 = aggregate( cax[ , c("cap","h")], list(local=cax$local), mean )
> resumo2 = aggregate( cax[ , c("cap","h")], list(local=cax$local), sd )
>
> resumo = merge( resumo1, resumo2, by="local", suffixes=c(".mean",".sd") )
> resumo
   local cap.mean    h.mean   cap.sd     h.sd
1 chauas 293.6385  89.60094 139.8761 37.00023
2 jureia 404.4813 109.70954 213.8512 31.68068
3 retiro 236.5972  78.08333 137.2203 30.46426
```

Exercícios

Exercício: *Estatísticas do Caixetal*

Construa um `data.frame` com os dados de **área basal** por `local` e `parcela`.

Encontre a média e desvio padrão da área basal por `local`.

Calcule o intervalo de confiança de 95% da área basal por `local`.

Analisando a Distribuição das Variáveis: Gráficos Univariados

Histogramas

A primeira abordagem ao se estudar a distribuição de uma variável é o uso de **histogramas**:

```
> cax$dap = (pi/4)* (cax$cap/10)
> hist( cax$dap )
> hist( cax$dap[ cax$local == "chauas" ] )
> hist( cax$dap[ cax$local == "jureia" ] )
> hist( cax$dap[ cax$local == "retiro" ] )
```

A função `hist` também pode fornecer os dados do histograma, sem gerar o histograma propriamente dito:

```
> hist( cax$dap, plot=FALSE )
$breaks
 [1]   0  10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170

$counts
 [1]  72 408 329 116  62  20  10   3   4   1   1   0   0   0   0   0   1
```

```
$intensities
 [1] 7.010709e-03 3.972736e-02 3.203505e-02 1.129503e-02 6.037001e-03
 [6] 1.947420e-03 9.737098e-04 2.921130e-04 3.894839e-04 9.737098e-05
[11] 9.737098e-05 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
[16] 0.000000e+00 9.737098e-05

$density
 [1] 7.010709e-03 3.972736e-02 3.203505e-02 1.129503e-02 6.037001e-03
 [6] 1.947420e-03 9.737098e-04 2.921130e-04 3.894839e-04 9.737098e-05
[11] 9.737098e-05 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
[16] 0.000000e+00 9.737098e-05

$mids
 [1]   5  15  25  35  45  55  65  75  85  95 105 115 125 135 145 155 165

$xname
[1] "cax$dap"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
>
```

Note que o objeto gerado pela função `hist` tem a classe `histogram`, logo ele pode ser guardado e grafado posteriormente:

```
> dap.hist = hist( cax$dap, plot=FALSE )
> class(dap.hist)
[1] "histogram"
>
> plot(dap.hist)
```

Alguns parâmetros gráficos podem tornar o gráfico mais apresentável. Esses parâmetros gráficos pode ser utilizados como *argumentos* em diversas funções gráficas onde são pertinentes:

- `xlab` e `ylab` = nomes dos eixos X e Y, respectivamente;
- `main` = nome do título do histograma;
- `col` = cor da barra (histograma), ou de linhas e símbolos plotados;

```
> hist( cax$dap[ cax$local=="chauas" ],
+ xlab="Diâmetro à Altura do Peito - DAP (cm)",
+ ylab="Freqüência",
+ main="Histograma DAP - Chauás",
+ col = "blue" )
>
```

Muitas vezes desejamos comparar gráficos, sendo útil termos mais de uma janela gráfica. A função `X11()` abre janelas gráficas, sendo que podemos abrir várias janelas:

```
> hist( cax$dap[ cax$local=="chauas" ] , main="Chauás" )
> X11()
> hist( cax$dap[ cax$local=="jureia" ] , main="Juréia" )
> X11()
> hist( cax$dap[ cax$local=="retiro" ] , main="Retiro" )
>
> dev.off()
X11
  2
> dev.off()
X11
  3
>
```

A função `dev.off()` fecha uma janela gráfica e faz parte de um conjunto de funções que manipula as janelas gráficas. Nessa manipulação, somente uma janela gráfica pode estar 'ACTIVE' de cada vez, e as janelas são consideradas como estando num círculo, onde podemos passar de uma para outra:

- `dev.off()` - fecha a janela gráfica;
- `dev.cur()` - diz qual janela gráfica está ACTIVE;
- `dev.set(which=dev.cur())` - define qual janela deverá ficar ativa, o argumento `which` deve ser o número da janela;
- `dev.next(which=dev.cur())` - informa o número da próxima janela gráfica;
- `dev.prev(which=dev.cur())` - informa o número da janela gráfica anterior;
- `graphics.off()` - fecha todas as janelas gráficas.

```
> X11()
> hist( cax$dap[ cax$local=="retiro" ] , main="Retiro" )
> X11()
> hist( cax$dap[ cax$local=="jureia" ] , main="Juréia" )
> dev.cur()
X11
  3
> dev.set(2)
X11
  2
> hist( cax$dap[ cax$local=="retiro" ] , main="Retiro" , col="blue")
> dev.set(3)
X11
  3
> hist( cax$dap[ cax$local=="jureia" ] , main="Juréia" , col="green")
>
> graphics.off()
>
```

Exercício

Exercício: *Altura de Árvores em Caixetais I*

Construa um histograma da altura das árvores do caixetal.

Construa histogramas da altura das árvores para os diferentes caixetais (`local`).

Há diferenças entre as estruturas (distribuição de tamanhos) dos caixetais?

Exercício: *Diâmetros de Árvores de Eucalipto*

Construa um histograma do DAP das árvores de *E. saligna*. Procure interpretar o histograma.

Gráficos de Densidade

Uma outra forma de explorar a distribuição de uma variável é trabalharmos com um gráfico de *densidade*. O gráfico de densidade é gerado como se fosse um histograma com uma classe móvel, isto é, a classe que tem uma certa amplitude, se move da esquerda para direita e em cada ponto estima a *densidade probabilística da variável*. Tecnicamente, a função `density` é um **estimador de densidade de kernel gaussiano**.

A função `density` faz a análise da densidade, mas não faz o gráfico, devendo ser utilizada juntamente com a função `plot`, para criar o gráfico, ou a função `lines`, para adicionar uma linha a um gráfico já criado:

```
> plot( density(cax$dap) )
>
```

O parâmetro que controla o comportamento do estimador de densidade é a amplitude da janela de observação ou *bandwidth* (bw). Janela pequenas geram estimativas de densidade com viés pequeno, mas com variância grande. Janelas grandes geram estimativas de densidade com viés grande, mas pequena variância. O ideal é o equilíbrio entre os extremos e o R possui algumas funções que buscam automaticamente da *bandwidth* apropriada, mas o analista tem controle sobre esse parâmetro:

```
> plot( density(cax$dap, bw=0.5), col="red" )
> lines( density(cax$dap, bw=5), col="blue" )
> lines( density(cax$dap, bw=1.5), col="green" )
>
```

Exercícios

Exercício: *Distribuição de DAP nos Caixetais*

Realize uma análise de densidade do DAP para cada um dos caixetais. Os resultados confirmam o que foi visto nos histogramas?

Boxplot

Boxplot é um gráfico estatístico também utilizado para estudar o comportamento de variáveis. Ele é composto dos elementos:

- Uma caixa (*box*) que representa a região entre o primeiro e o terceiro quartis (quantis 25% e 75%), ou seja, 50% dos dados estão dentro da caixa.
- Uma linha dentro da caixa que representa a posição da mediana (segundo quartil ou quantil 50%).
- Linhas que se prolongam a partir da caixa até no máximo 1,5 vezes a distância interquartil (diferença entre o 1o. e 3o. quartis).
- As observações que passarem essa distância são representadas individualmente por pontos.

```
> boxplot( cax$dap )
>
> esa = read.csv("dados/esaligna.csv",header=TRUE)
> boxplot( esa$dap )
>
```

O **boxplot** é útil para analisar a *simetria* de uma distribuição, o *espalhamento* das observações e a presença de observações discrepantes. Ele é problemático quando a variável analisada **não é unimodal**. Ele também é uma ferramenta útil para comparar distribuições, isso é realizado quando desejamos um **boxplot** para cada situação:

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

Note que o primeiro argumento da função `boxplot` não é um vetor nesse caso:

```
> class( dap ~ local )
[1] "formula"
>
```

O primeiro argumento é uma `formula` estatística onde o símbolo **~** tem um significado especial.

A fórmula `dap ~ local` deve ser lida como: *modele a variável* `dap` *como função da variável* `local`.

O argumento `data` informa em qual `data.frame` estão as variáveis citadas na fórmula e é um argumento essencial toda vez que se utiliza uma fórmula.

A utilização da fórmula permite a construção de gráficos mais complexos, pensando na **interação** entre dois fatores influenciando a variável DAP:

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

Mas os gráficos no R podem ficar realmente *sofisticados*, mas é necessário um pouco mais de programação:

```
> boxname = paste( sort(rep(unique(cax$local),5)), rep(1:5,3) )
```

```
> boxcor = sort(rep(c("navy","darkgreen","salmon1"),5))
> boxplot( dap ~ local * parcela, data=cax , names=boxname, col=boxcor,
horizontal=T, las=1, xlab="DAP (cm)")
>
```

Exercícios

Exercício: *Altura de Árvores em Caixetais II*

Utilize o gráfico boxplot para analisar a altura das árvores em caixetais.

Exercício: *Estrutura de Eucaliptais*

Utilize o gráfico boxplot para analisar o DAP de árvores de *E. grandis* em função das variáveis região (`regiao`) e rotação (`rot`).

Gráficos Quantil-Quantil

Gráficos Quantil-Quantil também são uma forma de estudar o comportamento de variáveis, mas utilizando as propriedades que emergem de uma variável quando trabalhamos com os seus quantis.

O gráfico quantil-quantil mais tradicional é aquele usado para verificar se uma variável possui distribuição Normal. No R isso é realizado com a função `qqnorm`, associada à função `qqline` que adiciona uma linha ao gráfico:

```
> qqnorm( cax$dap )
> qqline( cax$dap )
>
```

A idéia central do gráfico quantil-quantil é a seguinte: quando um variável segue uma dada distribuição (como a distribuição Normal) os **quantis empíricos**, isto é, calculados a partir de uma amostra, formam uma linha reta contra os **quantis teóricos**, calculados a partir das estimativas dos parâmetros da distribuição (no caso da Normal: média e desvio padrão).

É isso que a função `qqnorm` faz para distribuição Normal:

```
> vn1 = rnorm( 10000 )
> qqnorm( vn1 )
> qqline( vn1 )
>
> ve1 = rexp( 100000 )
> qqnorm( ve1 )
> qqline( ve1 )
>
> ve2 = apply( matrix(ve1, ncol=100), 1, mean)
> qqnorm( ve2 )
> qqline( ve2 )
```

```
>
```

Também é possível comparar duas distribuições a partir dos **quantis empíricos**:

```
> qqplot( cax$dap[ cax$local=="retiro" ], cax$dap[ cax$local=="jureia" ] )
> abline( 0, 1, col="red" )
>
> a = min( cax$dap[ cax$local=="jureia" ] )
> abline( a, 1, col="navy" )
>
```

Nota: a função `abline( a, b)` adiciona a um gráfico uma reta com intercepto *a* e inclinação *b*.

Exercícios

> ***Exercício:*** *Inventário em Floresta Plantada*
>
> Verifique se as variáveis quantitativas obtidas no inventário de florestas plantadas tem distribuição Normal: `dap`, `ht` e `hdom`.

> ***Exercício:*** *Altura de Árvores em Caixetais III*
>
> Verifique se a distribuição da altura das árvores tem o mesmo comportamento nos diferentes caixetais.

Gráfico de Variável Quantitativa por Classes

A maneira clássica de se apresentar uma variável quantitativa associada a uma classe é o famoso gráfico de barras.

Vejamos um exemplo comum em *fitossociologia* que é apresentar a densidade relativa das espécies:

```
> da = table( cax$especie[ cax$local=="jureia" ] )
> da = sort(da, decreasing=TRUE )
> dr = da/sum(da) * 100
```

Para obter o gráfico de barras basta usar a função `barplot`:

```
> barplot( dr )
```

O resultado não é muito apropriado para interpretações, mas podemos fazer algumas melhoras:

```
> barplot( dr , xlab="Densidade Relativa (%)", horiz=T, las=1)
```

Os nomes das espécies precisam de mais espaço. É possível alterar o espaço trabalhando os parâmetros da função `par` que controla todos os parâmetros gráficos de uma janela gráfica. Nesse caso, o parâmetro `omd=c(x1,x2,y1,y2)` define o início e final da região de plotagem em termos

relativos. O valor *default* é `omd=c(0, 1, 0, 1)`.

```
> par( omd=c(0.2,1,0,1) )
> barplot( dr , xlab="Densidade Relativa (%)", horiz=T, las=1)
>
```

Na verdade, o gráfico de barras não é um gráfico muito apropriado para o que se propõe, apesar do uso generalizado que se faz dele na comunidade científica.

No gráfico de barras, somos levados a comparar o comprimento das barras para estabelecer um julgamento entre as categorias. No gráfico de densidade relativa, comparamos os comprimentos de barra para obter uma visão das densidades relativas das espécies.

Existe no R, um gráfico que faz a mesma coisa de modo muito mais simples e direto:

```
> par( omd=c(0,1,0,1) )     # Primeiro é necessário re-estabelecer o
parâmetro omd
>
> dotchart( dr )
>
```

No `dotchart`, somos levados a comparar a posição relativa dos pontos, e a relação entre as categorias fica muito mais rápida e direta.

Como nessa floresta a *Tabebuia cassinoides* (caixeta) é a espécie dominante, é interessantes fazer o gráfico na escala logarítmica para enfatizar a diferença entre as outras espécies:

```
> dotchart( log(dr), xlab="Logaritmo Natural da Densidade Relativa (%)")
```

Exercícios

Exercício: *Dominância em Caixetais*

Construa um gráfico da dominância (biomassa relativa) das espécies nos caixetais.

Exercício: *Inventário em Floresta Plantada*

Utilizando a função `dotchart` investigue o número de árvores no inventário em função da região (`regiao`) e rotação (`rot`).

Análise Gráfica: Relação entre Variáveis

Gráfico de Dispersão

Os gráficos de dispersão (ou gráficos x-y) são os gráficos mais utilizados para estudar a relação entre duas variáveis.

A função genérica no R para gráficos de dispersão é a função `plot`:

```
> plot( x = cax$dap, y = cax$h )
```

Na função `plot`, o primeiro argumento é plotado nas *abscissas* (eixo-x) e o segundo argumento nas *ordenadas* (eixo-y).

Ao investigar a relação entre duas variáveis, frequentemente a densidade de pontos no gráfico torna o julgamento da relação problemática, pois é muito difícil considerar a variação da densidade ao se julgar a relação no gráfico de dispersão.

Há no R uma função adicional que auxilia o julgamento adicionando ao gráfico de dispersão uma linha não-paramétrica de tendência (**smooth** ou suavização):

```
> scatter.smooth( cax$dap, cax$h , col="red")
```

Uma série de parâmetros gráficos podem ser utilizados diretamente nas funções `plot` e `scatter.smooth`:

```
> scatter.smooth( cax$dap, cax$h , col="red", xlab="DAP (cm)", ylab="Altura
(dm)", main="Caixetais")
> scatter.smooth( cax$dap, cax$h , col="red", xlab="DAP (cm)", ylab="Altura
(dm)", log="x")
> scatter.smooth( cax$dap, cax$h , col="red", xlab="DAP (cm)", ylab="Altura
(dm)", log="y")
> scatter.smooth( cax$dap, cax$h , col="red", xlab="DAP (cm)", ylab="Altura
(dm)", log="xy")
```

O R também permite um certo grau de **interação** com gráficos de dispersão. Uma delas é a identificação de observações no gráfico:

```
> scatter.smooth( cax$dap, cax$h )
> dim( cax )
[1] 1027    8
> identify( cax$dap, cax$h, 1:1027 )
[1] 362 556 557
>
>
> cax[ c(362, 556, 557), ]
     local parcela arvore fuste  cap   h                  especie       dap
362 chauas       5    232     1  130 480     Tabebuia cassinoides  10.21018
556 jureia       4    105     1 1400 100     Tabebuia cassinoides 109.95574
557 jureia       4    106     1 2100 160 Calophyllum brasiliensis 164.93361
>
```

A função `identify` atua sobre um gráfico produzido (plot) e possui três argumentos. Os dois primeiros são os mesmos argumentos que geraram o gráfico. O terceiro argumento é uma variável de identificação. No exemplo acima a variável de identificação é o índice que identifica a observação (linha do `data.frame`).

Ao executar a função `identify`, o R entra num modo interativo com o gráfico. Ao posicionar o

mouse sobre uma observação no gráfico e pressionar o **botão esquerdo**, o R identifica a observação. É possível identificar tantas observações quanto se desejar. Para sair do modo interativo, pressiona-se o **botão direito** do *mouse*.

No exemplo acima, as três observações discrepantes do gráfico parecem de fato muito erradas. Assim, podemos eliminá-las e continuar o estudo da relação:

```
> cax2 = cax[ -c(362, 556, 557), ]
> scatter.smooth( cax2$dap, cax2$h , col="red" )
```

Também na função `plot` é possível se utilizar como argumento inicial uma `formula`, seguida do `data.frame` que contém as variáveis:

```
> plot( h ~ dap, data=cax2 )
```

Nesse caso, para adicionar a linha não-paramétrica de tendência é necessário um segundo comando:

```
> plot( h ~ dap, data=cax2 )
> lines( lowess( cax2$dap, cax2$h ) , col="red")
```

O uso da `formula` permite a utilização da função `coplot` para formação de gráficos de dispersão em função de variáveis categóricas:

```
> coplot( h ~ dap | local , data=cax2 )
> coplot( h ~ dap | local*parcela , data=cax2 )
```

Também é possível adicionar uma linha de tendência em cada gráfico gerado pela função `coplot`:

```
> coplot( h ~ dap | local , data=cax2 , panel= panel.smooth)
> coplot( h ~ dap | local*parcela , data=cax2 , panel=panel.smooth)
```

Na `formula` acima, surgiram elementos novos:

- A barra vertical indica uma situação condicional, no caso fazer um gráfico de dispersão para cada `local`.
- O asterísco (*) indica **interação**, no caso o gráfico de dispersão é realizado para cada interação entre as variáveis `local` e `parcela`.

A função `coplot` atua de forma diferente, se as variáveis que classificam o gráfico de dispersão são variáveis categóricas (`factor`) ou numéricas (`numeric`):

```
> egr = read.table("dados/egrandis.csv",header=TRUE,sep=";")
> coplot( ht ~ dap | idade, data=egr, panel = panel.smooth )
> coplot( ht ~ dap | idade * rotacao , data=egr, panel = panel.smooth )
> coplot( ht ~ dap | idade * as.factor(rotacao) , data=egr, panel =
panel.smooth )
```

Exercícios

Exercício: *Relação Hipsométrica da Caixeta*

Analise a relação dap-altura (`dap` e `h`) em função do caixetal, mas **somente** para as árvores de caixeta (*Tabebuia cassinoides*).

Exercício: *Inventário em Floresta Plantada II*

Analise a relação entre as variáveis `hdom` (altura das árvores dominantes) e `dap` para diferentes regiões (`regiao`) e rotações (`rotacao`).

Painel de Gráficos de Dispersão

Quando o objetivo é explorar a relação entre variáveis quantitativas com o objetivo de construir modelos ou analisar a estrutura de correlação é útil poder fazer gráficos de dispersão das variáveis duas-a-duas. A função **pairs** realiza essa operação automaticamente:

```
> pairs( egr[ , c("dap","ht","idade")] )
```

Sempre é possível sofisticar os gráficos. No exemplo abaixo o painel apresenta a relação entre as variáveis quantitativas utilizando cores para mostrar as variáveis região e rotação:

```
> pairs( egr[ , c("dap","ht","idade")] , pch=21, bg=c("red","blue","green",
"gold")[unclass(as.factor(egr$regiao))] )
> pairs( egr[ , c("dap","ht","idade")] , pch=21,
bg=c("red","green")[unclass(egr$rotacao)] )
```

Exercícios

Exercício: *Biomassa de Árvores de Eucalipto*

Analise a relação entre as variáveis quantitativas do conjunto de dados sobre biomassa das árvores de *E. saligna*.

Qual a influência da variável classe (`classe`) sobre a relação entre as variáveis?

Gráficos em Painel: O Pacote Lattice

Para ampliar a capacidade de análise gráfica exploratória e mesmo *modelagem gráfica* dos dados, existe no R o pacote **lattice**. Para carregar o pacote usa-se o comando:

```
> library(lattice)
```

O pacote **lattice** oferece uma série de funções análogas às funções gráficas do R, mas permite a construção de **paineis**. Um painel é um série de gráficos de mesmo tipo (dispersão, histograma, etc.) colocados lado-a-lado acompanhando uma variável categórica ou quantitativa.

Gráficos de Dispersão

Para construir gráficos de dispersão no lattice usa-se a função **xyplot**:

```
> egr = read.csv("egrandis.csv",header=T)
> xyplot( ht ~ dap, data=egr )
```

Note que no lattice, os gráficos são construídos com base em fórmulas. Essas fórmulas permitem estrutura mais complexas de análise:

```
> xyplot( ht ~ dap | regiao , data=egr )
> xyplot( ht ~ dap | regiao * rotacao , data=egr )
```

Também é possível construir gráficos com suavização:

```
> xyplot( ht ~ dap | regiao * rotacao , data=egr,
 panel = function(x,y)
 {
         panel.xyplot(x,y)
         panel.loess(x,y, span=1, col="red")
 } )
>
```

Exercícios

Exercício: *Relação Hipsométrica da Caixeta II*

Utilizando o pacote `lattice`, analise a relação dap-altura (`dap` e `h`) em função do caixetal, mas **somente** para as árvores de caixeta (*Tabebuia cassinoides*).

Exercício: *Relação Altura das Dominantes - Idade em Florestas Plantadas*

Utilizando os dados de floresta plantada (*E. grandis*), analise a relação entre altura das árvores dominantes (`hdom`) e idade (`idade`) por rotação (`rotacao`) e região (`regiao`).

Painel de Gráficos de Dispersão

O pacote lattice também possui uma função específica para fazer um painel de gráficos de dispersão: **splom** (*scatter plot*):

```
> splom( egr[ , c("dap","ht","hdom","idade")]  )
```

Identificar grupos em cada gráfico de dispersão é mais fácil com a função **splom**, basta utilizar o argumento `group`:

```
> splom( egr[ , c("dap","ht","hdom","idade")] , group=egr$regiao )
> splom( egr[ , c("dap","ht","hdom","idade")] , group=egr$rot )
```

Também é possível adicionar uma *linha de suavização*, mas é necessário definir a função de painel (argumento `painel`):

```
> splom( egr[ , c("dap","ht","hdom","idade")] , group=egr$regiao,
+ panel = function(x,y,...)
+ {
+     panel.splom(x,y,...)
+     panel.loess(x,y,...)
+ }
+ )
>
```

A função **panel.loess** é a função que efetivamente faz a suavização em cada gráfico de dispersão.

Exercícios

Exercício: *Biomassa de Árvores de Eucalipto*

Analise a relação entre as variáveis quantitativas dos dados de biomassa de *E. saligna* utilizando a função **splom**. Inclua na sua análise a variável `classe`.

Histogramas e Gráficos de Densidade

No lattice, todos os tipos de gráficos podem ser construídos na forma de painel. Para estudar a distribuição de variáveis temos a função **histogram** e **densityplot**:

```
> cax = read.csv("caixeta.csv",header=T)
> cax$dap = cax$cap / pi
>
> histogram( ~ dap, data=cax )
> histogram( ~ dap | local , data=cax )
>
> densityplot( ~ dap, data=cax )
> densityplot( ~ dap | local , data=cax )
```

Também é possível construir um histograma com linhas de densidade, para isso o tipo do histograma deve ser definido como `density`:

```
> histogram( ~ ht | regiao * rot , dat=egr, type="density",
+     panel = function(x, ...){
+             panel.histogram(x, ...)
+             panel.densityplot(x, col="red", ...)
+     }
```

```
+ )
```

As funções de histograma e densidade podem se tornar mais complexas. No exemplo abaixo, uma curva de densidade assumindo a distribuição Normal é adicionada aos histogramas, os quais são construídos com a densidade nas ordenadas:

```
> histogram( ~ ht | regiao * rot , dat=egr, type="density",
+     panel = function(x, ...){
+           panel.histogram(x, ...)
+           panel.mathdensity(dmath=dnorm, col="black",
args=list(mean=mean(x),sd=sd(x)))
+     }
+ )
```

Exercícios

> ***Exercício:*** *Altura das Árvores Dominantes em Florestas Plantadas*
>
> Explore o comportamento da variável altura das árvores dominantes (`hdom`) por região (`regiao`) e rotação (`rot`) na floresta plantada de *E. grandis*.

> ***Exercício:*** *Altura de Árvores de Caixeta*
>
> Analise o comportamento da variável altura (h) das árvores de caixeta.

Gráficos Quantil-Quantil

O pacote **lattice** implementa a construção de gráficos sempre através de fórmulas, isso pode ser conveniente no caso de se verificar a distribuição de uma variável em várias situações:

```
> qqmath( ~ dap | local, data=cax )
```

Para adicionar a linha do gráfico qq é necessário editar a função de painel:

```
> qqmath( ~ dap | local, data=cax,
+     panel = function(x, ...)
+    {
+       panel.qqmath(x, ...)
+       panel.qqmathline(x, ...)
+    }
+ )
```

Uma vantagem do pacote lattice é a possibilidade de gráficos quantil-quantil com outras distribuições além da distribuição normal. Nos gráficos abaixo, a distribuição observada de DAP das árvores dos caixetais é comparada com a distribuição exponencial (`qexp`).

```
> qqmath( ~ dap | local , data=cax, distribution = function(p) qexp(p,
1/mean(x)) )
>
> qqmath( ~ dap | local , data=cax, distribution = function(p) qexp(p,
1/mean(x)),
+    panel = function(x,...)
+    {
+       panel.qqmath(x,...)
+       panel.qqmathline(x,...)
+    }
+ )
>
```

Também é possível fazer gráficos quantil-quantil de um conjuntos de dados empíricos usando a função **qq**:

```
> qq( local ~ dap, data=cax, subset = ( local=="chauas" | local=="jureia" )
)
> qq( local ~ dap, data=cax, subset = ( local=="chauas" | local=="retiro" )
)
> qq( local ~ dap, data=cax, subset = ( local=="jureia" | local=="retiro" )
)
```

Dois aspectos devem ser notados no código acima:

1. A variável `local` (categórica) aparece **à esquerda** do sinal de modelagem.
2. O argumento `subset` faz com que a variável `local` fique com apenas duas categorias.

Exercícios

Exercício: *Altura das Árvores em Florestas Plantadas*

Verifique se a altura das árvores (`ht`) nas florestas plantadas de *E. grandis* segue distribuição Normal.

Faça uma análise geral e depois por região (`regiao`) e rotação (`rot`).

Exercício: *Biomassa de Árvores de Eucalipto*

Verifique se biomassa total (`total`) e a biomassa do tronco (`tronco`) das árvores de *E. saligna* possuem distribuição semelhante. E a biomassa das folhas (`folha`), tem distribuição semelhante à biomassa do tronco?