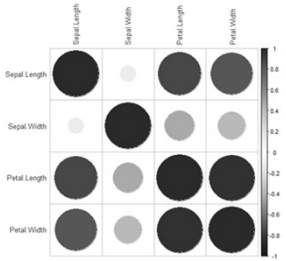
Visualização de Dados Multivariados

- Correlograma:

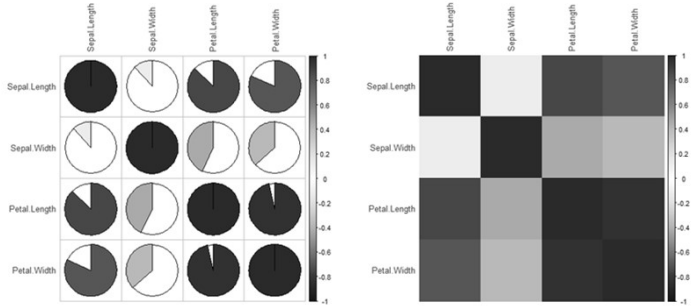
```
        # Matriz de correlações
(iris.cor <- cor(iris[-5]))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
> library(corrplot)
> # correlograma - círculo
> corrplot(iris.cor, method = "circle")
```



√ círculos

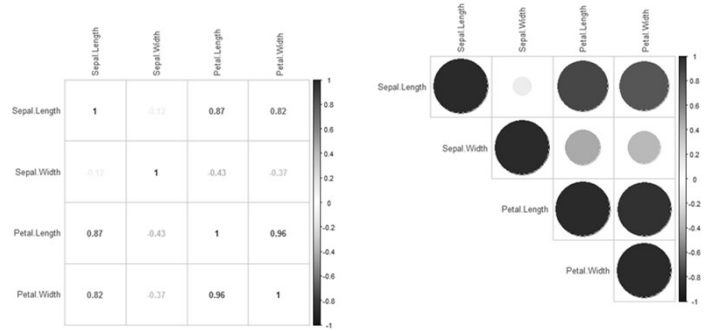
- Correlograma:

```
> # correlograma - pizza
> corrplot(iris.cor, method = "pie")
> # coorelograma - cor
> corrplot(iris.cor, method = "color")
```



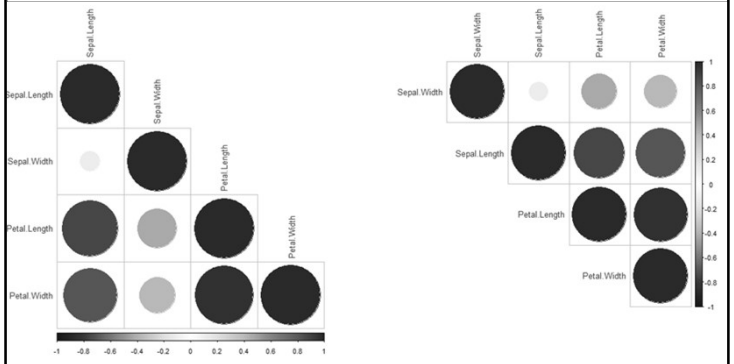
- Correlograma:

```
> # correlograma - valores
> corrplot(iris.cor, method = "number")
> # correlograma - superior
> corrplot(iris.cor, type = "upper")
```



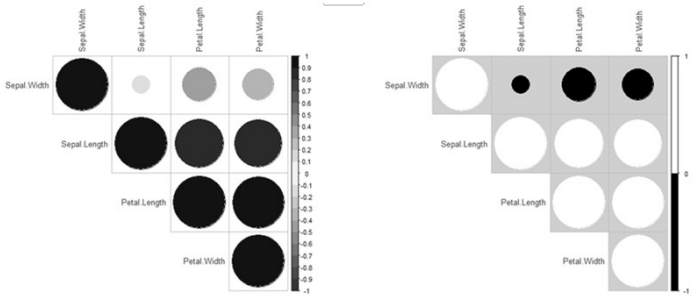
- Correlograma:

```
> # correlograma - inferior
> corrplot(iris.cor, type = "lower")
> # correlograma c/ reordenação por hclust
> corrplot(iris.cor, type="upper", order = "hclust")
```



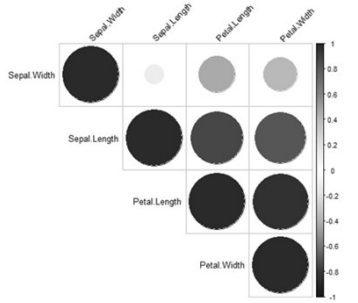
- Correlograma:

```
> # usando espectro de cores diferente
> col <- colorRampPalette(c("red", "white", "blue"))(20)
> corrplot(iris.cor, type = "upper", order = "hclust", col = col)
> # Mudando cor de fundo para lightblue
> corrplot(iris.cor, type = "upper", order = "hclust", col = c("black", "white"),
+          bg = "lightblue")
```



- Correlograma:

```
> # Mudando a cor e a rotação dos rótulos
> corrplot(iris.cor, type = "upper", order = "hclust", tl.col = "black",
+ tl.srt = 45)
> #tl.col (cor do texto) e tl.srt (rotação texto)
```



- Correlograma:
  - √ Função para cálculo de p-valor

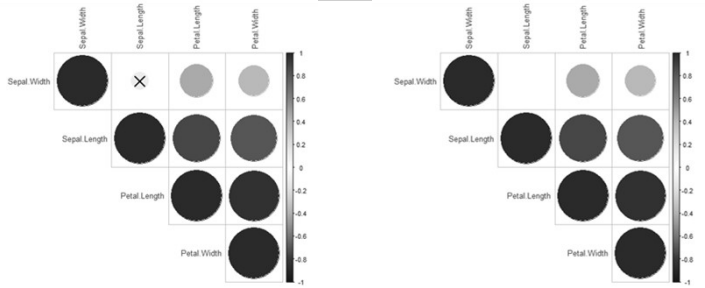
```
> # Função para cálculo do p-valor das correlações
> cor.mteste <- function(mat, ...) {
+     mat <- as.matrix(mat)
+     n <- ncol(mat)
+     p.mat<- matrix(NA, n, n)
+     diag(p.mat) <- 0
+     for (i in 1:(n - 1)) {
+         for (j in (i + 1):n) {
+             tmp <- cor.test(mat[, i], mat[, j], ...)
+             p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
+         }
+     }
+   colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
+   p.mat
+ }
```

  - √ Matriz dos p-valores das correlações

```
> # matriz dos p-valores das correlações
> p.mat <- cor.mteste(iris[-5])
> head(p.mat)
 Sepal.Length  Sepal.Width Petal.Length  Petal.Width
Sepal.Length 0.000000e+00 1.518983e-01 1.038667e-47 2.325498e-37
Sepal.Width  1.518983e-01 0.000000e+00 4.513314e-08 4.073229e-06
Petal.Length 1.038667e-47 4.513314e-08 0.000000e+00 4.675004e-86
Petal.Width  2.325498e-37 4.073229e-06 4.675004e-86 0.000000e+00
```

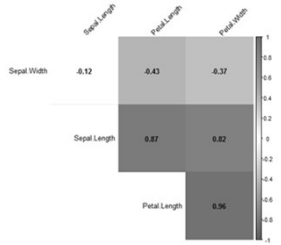
- Correlograma:

```
> # Agregando nível de significância ao correlograma
> corrplot(iris.cor, type="upper", order="hclust", p.mat = p.mat,
+ sig.level = 0.01)
> # Deixando em branco coeficiente não significante
> corrplot(iris.cor, type = "upper", order = "hclust", p.mat = p.mat,
+ sig.level = 0.01, insig = "blank")
```



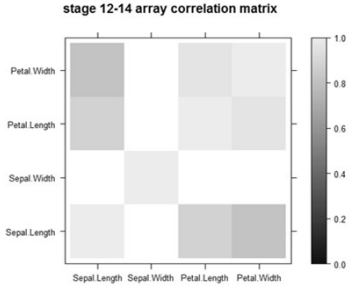
- Correlograma:

```
> # Customizando o correlograma
> col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD",
+ "#4477AA"))
> corrplot(iris.cor, method="color", col=col(200), type="upper", order="hclust",
+          addCoef.col = "black", # Adiciona coeficiente de correlação
+          tl.col="black", tl.srt=45, # Rotação e cor de texto rótulo
+          # Combinação com significância
+          p.mat = p.mat, sig.level = 0.01, insig = "blank",
+          diag=FALSE # elimina valores da diagonal principal
+          )
```



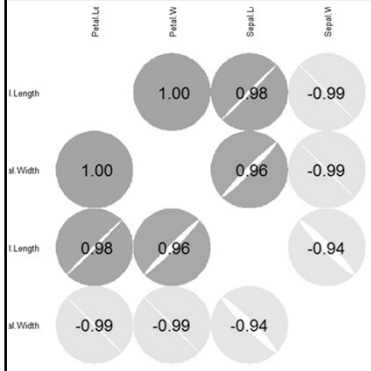
- Matriz de Correlações – pacote `lattice`:

```
> library(lattice)
> rgb.palette <- colorRampPalette(c("blue", "yellow"), space = "rgb")
> levelplot(iris.cor, main = "stage 12-14 array correlation matrix",
+ xlab = "", ylab = "", col.regions = rgb.palette(120),
+ cuts = 100, at = seq(0, 1, 0.01)
+ )
```



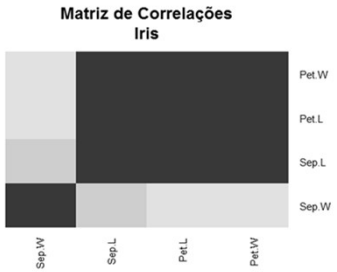
- Matriz de Correlações – pacote `lattice`:

```
> source("https://github.com/JVAdams/jvamisc/blob/master/R/plotcor.R")
> library(plotrix)
> library(seriation)
> library(MASS)
> plotcor(cor(iris.cor), mar = c(0.1, 4, 4, 0.1))
```



- Mapa de Calor:

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(iris.cor, col = brewer.pal(9, "GnBu"), trace ="none",
+      key = FALSE, dend = "none", cexCol = 1.1, cexRow = 1.1, srtCol = 90,
+      labRow = c("Sep.L", "Sep.W", "Pet.L", "Pet.W"),
+      labCol = c("Sep.L", "Sep.W", "Pet.L", "Pet.W"),
+      main = "\n\n\nMatriz de Correlações\nIris")
```



- Gráfico em html:
  - √ Gráfico 1:

```
> library(plotly)
> p1 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = F)
> p2 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = T)
> subplot(p1,p2)
```

  - √ Gráfico 2:

```
> p1 <-
+   iris %>%
+   group_by(Species) %>%
+   plot_ly(x =~Sepal.Length, color = ~Species) %>%
+   add_markers(y = ~Sepal.Width)
> p2 <-
+   iris %>%
+   group_by(Species) %>%
+   plot_ly(x=~Sepal.Length, color= ~Species) %>%
+   add_markers(y =~Sepal.Width, showlegend = F)
> subplot(p1,p2)
```


Técnicas Gráficas Multivariadas

- A visualização de dados multivariados é possível se a dimensionalidade for relativamente baixa:
  - √ Técnicas gráficas
  - √ Técnicas multivariadas como ferramentas de exploração

Alguns Gráficos Multivariados

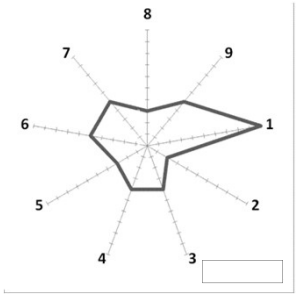
- √ Star plot
- √ Segment diagram
- √ Chernoff faces
- √ Parallel coordinate plot
- √ Andrew's curve
- √ Buble plot
- √ Mosaic plot

- Análise multivariada quantitativa
  - √ Análise de componentes principais
  - √ Biplots
  - √ Análise de componentes independentes
  - √ Análise de agrupamentos
  - √ Análise fatorial

- Análise multivariada categórica:
  - √ *Mosaic plots*
  - √ Análise de correspondência
  - √ Análise de correspondência múltipla

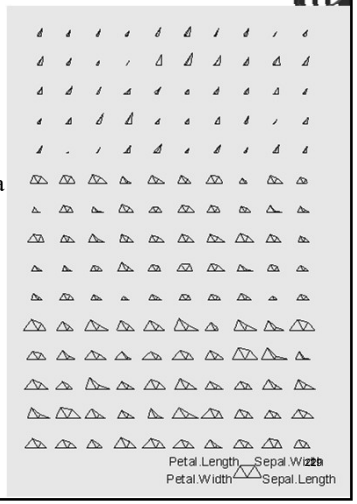
Star Plot

- Estrelas para visualização de dados
- Formação da estrela:
  - √ Raio para cada variável
  - √ Comprimento é proporcional à variável
- Útil para visualização de itens com número arbitrário de variáveis



- Pode ser usado para responder as seguintes perguntas:
  - √ Quais variáveis são dominantes para uma determinada observação?
  - √ Quais observações são similares?
    (Existem agrupamentos de observações?)
  - √ Existem valores discrepantes?

- Exemplo: Flores de íris
  - √ Você vê diamantes?
    - Alguns são grandes, alguns são pequenos
  - √ Dados em sequência
    - Setosa, versicolar e virginica
  - √ Valores iniciais pequenos
    - Setosa é do Alasca!
  - √ Há outliers?



Resumo

- Antes de uma análise mais aprofundada, sempre realizar uma exploração apropriada dos dados
  - √ Verifique se há erros óbvios nos dados
  - √ Familiarize-se bastante com os dados
  - √ Tente identificar a distribuição dos dados
- Análise exploratória de dados não é uma ciência exata
  - √ É uma arte muito importante

Verificação de Normalidade Multivariada

Exemplo

- Estudo poluição do ar
  - √ Amostra: 41 cidades americanas
  - √ Variáveis:
    - SO2: concentração no ar (mg/m3)
    - Temp: temperatura
    - Popul: população, em milhares (censo 1970)
    - Vento: velocidade média anual (milhas/hora)
    - Precip: precipitação média anual (pol)
    - Dias: número médio anual de dias de chuva
  - √ Dados: *USairpollution {HSAUR2}*

- Carregamento e preparação do conjunto de dados:

```
> library(MVA, HSAUR2)# carrega os pacotes
> data(USairpollution)# carrega o banco de dados
> help(USairpollution)# Descrição do banco de dados

> colunas<- c("SO2", "Temp", "Indust", "Pop",
+ "Vento", "Precip", "Dias")
> colnames(USairpollution)<- colunas
> dados <- USairpollution
```

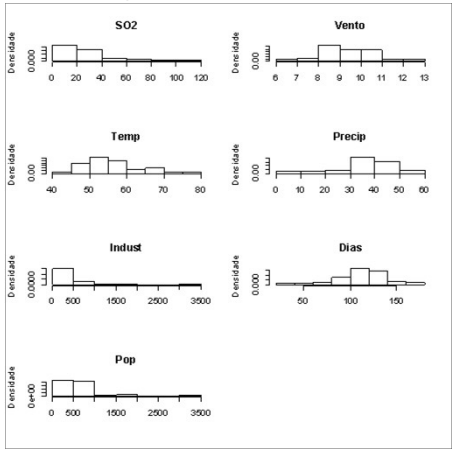
237

- Medidas resumo - univariada

	SO2	Temp	Indust	Pop	Vento	Precip	Dias
Min.	8,0	43,5	35,0	71,0	6,0	7,1	36,0
1º Quartil	13,0	50,6	181,0	299,0	8,7	31,0	103,0
Mediana	26,0	54,6	347,0	515,0	9,3	38,7	115,0
Mean	30,1	55,8	463,1	608,6	9,4	36,8	113,9
3º Quartil	35,0	59,3	462,0	717,0	10,6	43,1	128,0
Max.	110,0	75,5	3344,0	3369,0	12,7	59,8	166,0
D. padrão	23,5	7,2	563,5	579,1	1,4	11,8	26,5
C. variação	78,1%	13,0%	121,7%	95,2%	15,1%	32,0%	23,3%

238

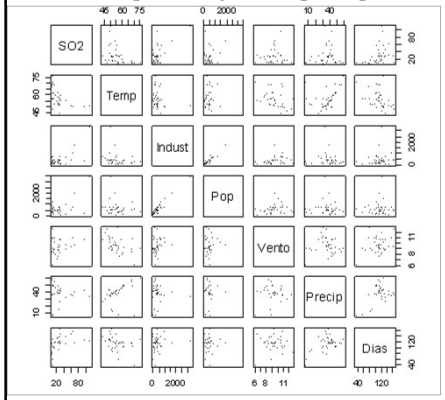
- Histogramas das variáveis



Verificação quanto a:
- √ Simetria
- √ Modalidade
- √ Pontos atípicos

239

- Estrutura de correlação entre as variáveis
  - √ Scatterplot matrix
    - Separar os gráficos para apresentá-los mais adequadamente



240

- Matriz de correlações

```
> round(cor(dados), 4)
             SO2    Temp  Indust     Pop   Vento  Precip    Dias
SO2       1.0000 -0.4336  0.6448  0.4938  0.0947  0.0543  0.3696
Temp     -0.4336  1.0000 -0.1900 -0.0627 -0.3497  0.3863 -0.4302
Indust    0.6448 -0.1900  1.0000  0.9553  0.2379 -0.0324  0.1318
Pop       0.4938 -0.0627  0.9553  1.0000  0.2126 -0.0261  0.0421
Vento     0.0947 -0.3497  0.2379  0.2126  1.0000 -0.0130  0.1641
Precip    0.0543  0.3863 -0.0324 -0.0261 -0.0130  1.0000  0.4961
Dias      0.3696 -0.4302  0.1318  0.0421  0.1641  0.4961  1.0000
```

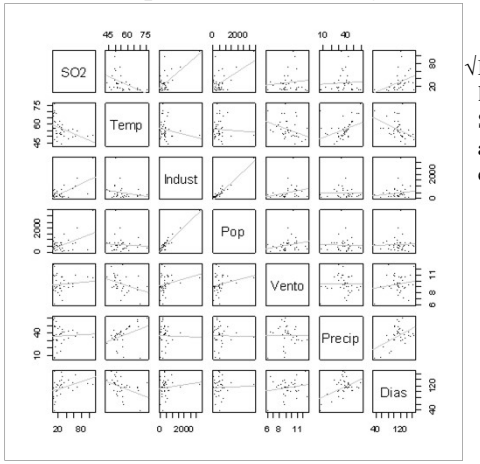
√ Forte correlação entre SO2 e Industr e Popul
- Indust e Popul são fortemente correlacionadas
- Provavelmente predizem SO2 da mesma maneira

√ Correlação entre SO2 e Precip é muito pequena

√ Correlação entre SO2 e Dias é moderada

241

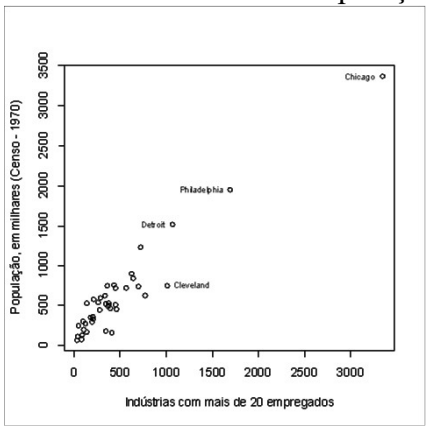
- Scatterplot matrix com ajuste linear



√É provável que modelo linear entre SO2 e Precip e SO2 e dias não perceberá adequadamente a relação entre cada par de variáveis

242

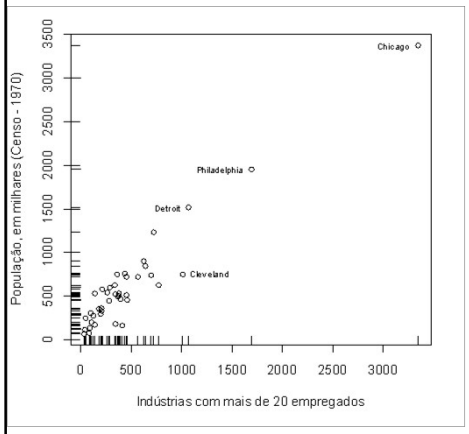
- Variáveis Indústria e População



√Há pontos que se afastam do padrão dos dados

243

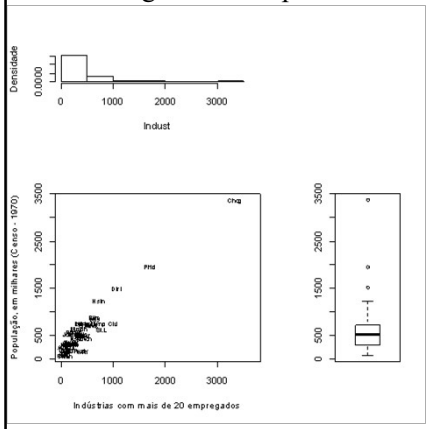
- Gráfico de dispersão com distribuição marginal



√Concentração de pontos na faixa inferior de ambas as variáveis

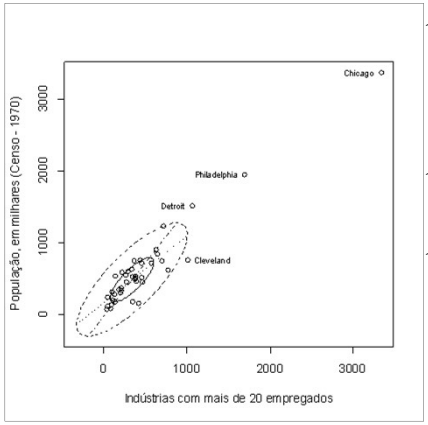
244

- Gráfico de dispersão com distribuição marginal
  - √ Histograma e boxplot



√Boxplot identifica alguns pontos extremos (univariado)

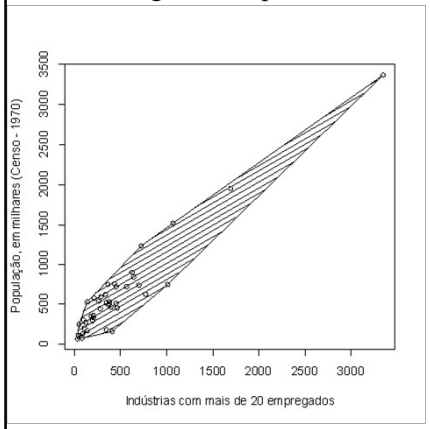
- Boxplot bivariado
  - √ Análogo ao boxplot univariado



√Boxplot bivariado identifica pontos extremos do vetor de dados (bivariado)

√Correlação
Todas: 0,9553
Exceto identificadas: 0,7956
√A redução não é considerável

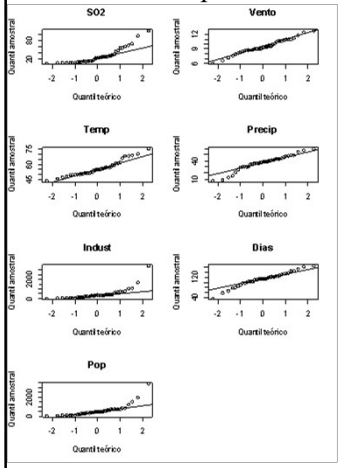
- Envelope convexo dos dados
  - √ Análogo ao boxplot univariado



√Envelope convexo apara valores
√Estimação robusta da correlação

√Correlação
Todas: 0,9553
Exceto envoltória: 0,9225
√Correlação estimada após remoção é maior que a relacionado com pontos identificados pelo boxplot bivariado

- Gráficos de probabilidade univariados



√ Gráficos para SO2 e Precipitação desviam-se consideravelmente da linearidade
√ Gráficos para Indústrias e População evidenciam outliers

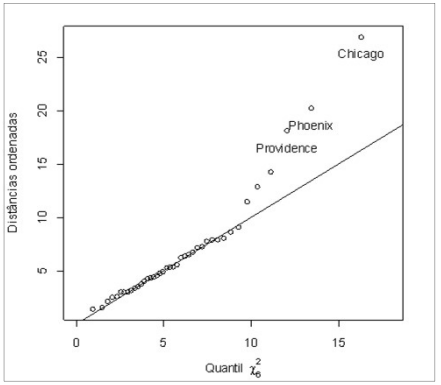
- Teste de normalidade
  - √ H_0: dados se ajustam à distribuição normal
  - √ H_1: dados não se ajustam à distribuição normal

```
> teste<- sapply(colnames(dados), function(x)shapiro.test(dados[[x]])$p.value)
> as.matrix(teste)
              [,1]
SO2     9.723376e-06
Temp    2.214972e-02
Indust 2.781101e-09
Pop     3.622798e-08
Vento   6.972580e-01
Precip 3.725311e-02
Dias    2.419457e-01
```

  - √ Não se rejeita a hipótese de normalidade para as variáveis Vento e Dias

254

- Gráfico das distâncias generalizadas (χ^2):



  - √ Gráfico detectou possíveis outliers nos dados multivariados
    - Desvio da variabilidade natural dos dados

255

Comentários

- É difícil construir um bom teste global de normalidade conjunta em mais de duas dimensões
- Hipótese de normalidade aparenta estar violada
  - √ As marginais aparentam ser normais? E aalgumas combinações lineares de componentes X_i?
  - √ Diagramas de dispersão de diferentes características têm aparência elíptica?
  - √ Há outliers que devessem ser verificados?

256

- Hipótese de normalidade individual é menos crucial em situações em que o tamanho amostral é grande e as técnicas dependem da média amostral (ou de distâncias envolvendo essa média)

257

Exemplos de Aplicação

Conjunto de Dados – `honolulu`

- Doenças cardiovasculares
  - √ 7.683 casos coletados no Havaí em 1969
  - √ Fator de exposição: fumante
- Universo:
  - √ Homens doentes com idade entre 45 e 67 anos
  - √ Média de Idade da população: 54,36 anos
- Tamanho da amostra: 100
- Dados: *honolulu.txt*

√ Variáveis codificadas:

- educacao : nível de instrução (1 = nenhuma, 2 = primeiro grau incompleto; 3 = primeiro grau completo; 4 = segundo grau completo; 5 = curso técnico; 6 = curso superior)
- peso, em Kg
- altura, em cm
- Idade, em anos
- fumante : status de fumante (0 = não; 1 = sim)
- atividade : atividade física em casa (1 = sedentário; 2 = moderada; 3 = alta)
- glicose: nível de glicose no sangue em mg percentuais
- colesterol: nível de colesterol sérico em miligramas percentuais
- pressao: pressão sanguínea sistólica, em mmHg

- Importação do conjunto de dados:

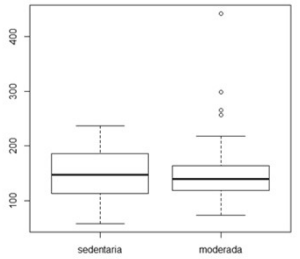
```
> dados <- read.table("honolulu.txt", head = TRUE)
> honolulu <- dados[-1]
> dim(honolulu)
[1] 100   9
> str(honolulu)
'data.frame':   100 obs. of  9 variables:
 $ educacao  : int  2 1 1 2 2 4 1 3 5 2 ...
 $ peso      : int  70 60 62 66 70 59 47 66 56 62 ...
 $ altura    : int  165 162 150 165 162 165 160 170 155 167 ...
 $ idade     : int  61 52 52 51 51 53 61 48 54 48 ...
 $ fumante   : int  1 0 1 1 0 0 0 1 0 0 ...
 $ atividade : int  1 2 1 1 1 2 1 1 2 1 ...
 $ glicose   : int  107 145 237 91 185 106 177 120 116 105 ...
 $ colesterol: int  199 267 272 166 239 189 238 223 279 190 ...
 $ pressao   : int  102 138 190 122 128 112 128 116 134 104 ...
> head(honolulu)
  educacao peso altura idade fumante atividade glicose colesterol pressao
1        2   70    165    61       1         1     107        199     102
2        1   60    162    52       0         2     145        267     138
3        1   62    150    52       1         1     237        272     190
4        2   66    165    51       1         1      91        166     122
5        2   70    162    51       0         1     185        239     128
6        4   59    165    53       0         2     106        189     112
```

- Preparação do conjunto de dados:

```
> # Transformação variáveis em fatores
> nomes.col <- c("educacao", "fumante", "atividade")
> honolulu[nomes.col] <- lapply(honolulu[nomes.col] , factor)
>
> # Renomeação níveis dos fatores - educacao
> edu.niveis <- c("N", "1I", "1C", "2C", "T", "S")
> levels(honolulu$educacao) <- edu.niveis
> # Renomeação níveis dos fatores - atividade
> ativ.niveis <- c("sedentaria", "moderada", "alta")
> levels(honolulu$atividade) <- ativ.niveis
> # Renomeacao níveis e reordenação níveis - fumante
> fuma.niveis <- c("N", "S")
> levels(honolulu$fumante) <- fuma.niveis
> # transformacao em fator ordenado
> honolulu$educacao <- ordered(honolulu$educacao)
> honolulu$atividade <- ordered(honolulu$atividade)
> head(honolulu)
  educacao peso altura idade fumante  atividade glicose colesterol pressao
1       1I   70    165    61       S sedentaria     107         199     102
2        N   60    162    52       N   moderada     145         267     138
3        N   62    150    52       S sedentaria     237         272     190
4       1I   66    165    51       S sedentaria      91         166     122
5       1I   70    162    51       N sedentaria     185         239     128
6       2C   59    165    53       N   moderada     106         189     112
```


- *Boxplot*: `glicose` vs. `atividade`:

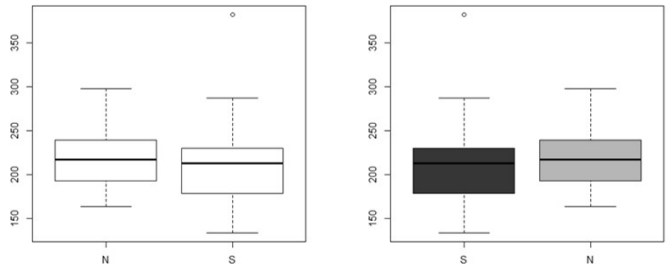
```
> # Default
> boxplot(glicose ~ atividade, data = honolulu)
```



√ Há diferença no nível médio de glicose entre os grupos
√ Há diferença das variabilidades dos grupos?
√ Há *outliers*?

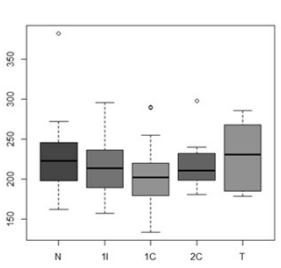
- *Boxplot*: `colesterol` vs. `fumo`:

```
> # Default
> boxplot(colesterol ~ fumante, data = honolulu)
> # Boxplot com caixas coloridas (inversão ordem fatores)
> fator <- relevel(honolulu$fumante, "S")
> boxplot(colesterol ~ fator, data = honolulu, col = c("red", "green"))
```

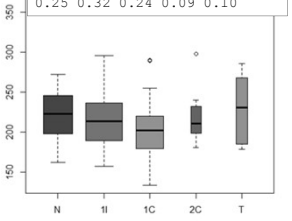


- *Boxplot*: `colesterol` vs. `educacao`:

```
> # Boxplot com caixas coloridas - qualquer qte. categorias
> library(RColorBrewer)
> QteNiveis <- length(levels(honolulu$educacao))
> cores <- brewer.pal(n = QteNiveis, name = "Set1")
> boxplot(colesterol ~ educacao, data = honolulu, col = cores)
>  # Boxplot com caixas proporcionais
> PropNiveis <- prop.table(table(honolulu$educacao))
> boxplot(colesterol ~ educacao, data = honolulu, width = PropNiveis, col = cores)
```



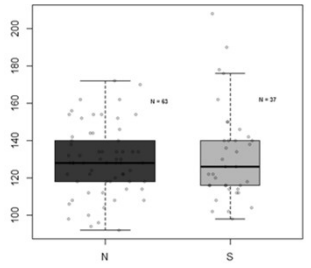

```
> PropNiveis
    N    1I    1C    2C     T
0.25 0.32 0.24 0.09 0.10
```



- *Boxplot*: `pressao` vs. `fumante`:
  - √ *Boxplot* customizado

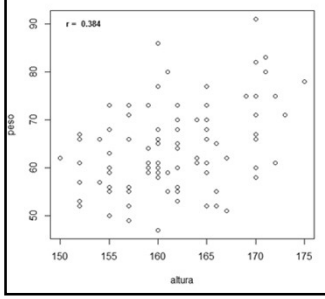
```
> # Boxplot customizado - pontos e caixas proporcionais
> PropNiveis <- prop.table(table(honolulu$fumante))
> boxplot(pressao ~ fumante, data = honolulu, col = c("red", "green"),
+ width = PropNiveis, outpch = NA)
>   # Acrescenta pontos
> niveis <- levels(honolulu$fumante)
> for(i in 1:length(niveis))
+ {
+   este.nivel <- niveis[i]
+   valores <- honolulu[honolulu$fumante == este.nivel, "pressao"]
+      # Adiciona perturbação, proporcional à N em cada nível (eixo X)
+   perturbacao <- jitter(rep(i, length(valores)), amount = PropNiveis[i]/2)
+   points(perturbacao, valores, pch = 20, col = rgb(0, 0, 0, 0.2))
+      # Adiciona texto do tamanho dos grupos
+   tipica <- min(max(valores), median(valores) + IQR(valores)*1.5)
+   text(i + PropNiveis[i]/2, tipica, cex = 0.6, font = 2, pos = 4,
+ labels = paste("N = ", length(valores), sep=""))
+ }
```


- *Boxplot* customizado:
  - √ Pontos, caixas proporcionais ao tamanho dos grupos, cores



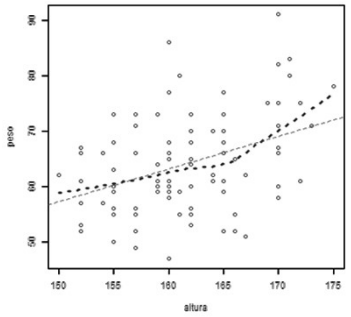
- *Scatter plot* – Doença cardiovascular:

```
> # Box-plots
> variaveis <- names(iris[-5])
> par(mfrow = c(2, 2))
> for(i in 1: length(variaveis)) {
+ with(iris, {
+ dados <- eval(parse(text = variaveis[i]))
+ boxplot(dados ~ Species, data = iris, main = variaveis[i], cex.axis = 0.85)
+ })
+ }
```



- √ Qual a relação entre o peso e a altura das pessoas?
- √ Percebem-se 'clusters'?
- √ Há diferenças na variabilidade de uma variável, considerados os valores da outra?
- √ Há valores atípicos?

- Relação entre as variáveis:
  - √ Reta de regressão e suavização

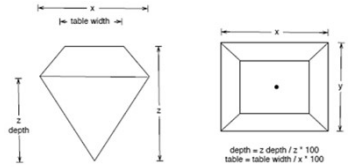


Conjunto de Dados – `diamonds`

- Preços e outros atributos de diamantes
  - √ Conjunto de dados com informações (preços e outros 9 atributos) sobre 53.940 diamantes
  - √ Fonte não informada
- Dados: `diamonds{ggplot2}`

- √ Variáveis codificadas:
  - price: preço, em US$ ($326 a $18.823)
  - carat: peso do diamante, em quilates (0,2 a 5,01)
  - cut: qualidade do corte (Fair, Good, Very Good, Premium, Ideal)
  - colour: cor do diamante (de J = pior para D = melhor)
  - clarity: medida de quão claro o diamante é (I1 = pior, SI1, SI2, VS1, VS2, VVS1, VVS2, IF = melhor)
  - x: comprimento, em mm (0 a 10,74)
  - y: largura, em mm (0 a 58,9)
  - z: espessura, em mm (0 a 31,8)
  - depth: espessura total percentual (43 a 79) $\frac{z}{\frac{x+y}{2}} = \frac{2\,z}{x+y}$
  - table: largura do topo do diamante em relação ao ponto mais largo (43 a 95)

- Desenho esquemático diamante

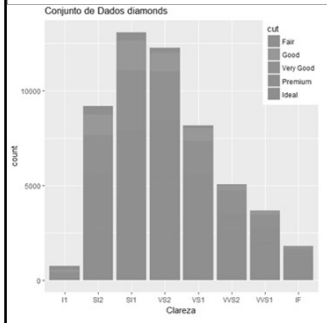


- Importação do conjunto de dados:

```
> library(ggplot2)
> dim(diamonds)
[1] 53940    10
> str(diamonds)
Classes 'tbl_df', 'tbl' and 'data.frame':       53940 obs. of  10 variables:
 $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 ...
 $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
> head(diamonds)
# A tibble: 6 x 10
  carat       cut color clarity depth table price     x     y     z
  <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1  0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43
2  0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31
3  0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31
4  0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63
5  0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75
6  0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48
> head(diamonds)
```

- Gráfico de barras – `clarity` *versus* `cut`

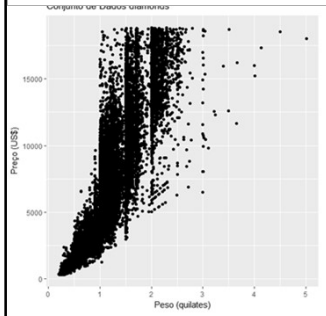
```
> # Diagrama de barras de 'clareza' categorizado com 'corte'
> ggplot(diamonds, aes(clarity, fill = cut)) + geom_bar() +
+ xlab("Clareza") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_color_discrete(name = "Corte") +
+ theme(legend.position = c(1, 1), legend.justification = c(1,1))
```



√ Qual a relação entre as duas variáveis?
√ Frequências de `cut` mudam nos níveis de `clarity`?

- *Scatter plot* – `price` *versus* `carat`

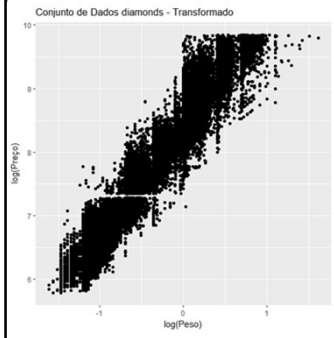
```
> # Peso vs. Preço
> p <- ggplot(data = diamonds, aes(x = carat, y = price)) +
+ geom_point() +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds")
> p
```



√ Gráfico aponta forte relação entere as variáveis
  – Há outliers importantes
  – Estrias verticais interessante
√ Relação aparenta ser exponencial
  – Interessante transformar os dados

- *Scatter plot* – `log price` *versus* `carat`

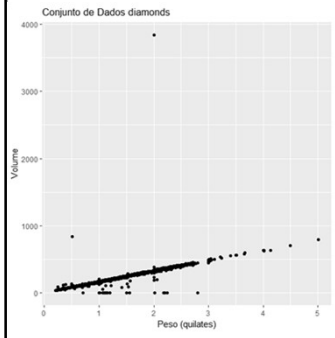
```
> # Transformação dos dados
> ggplot(data = diamonds, aes(x = log(carat), y = log(price))) +
+ geom_point() +
+ xlab("log(Peso)") +
+ ylab("log(Preço)") +
+ ggtitle("Conjunto de Dados diamonds - Transformado")
```



√ Relação agora parece ser linear
  – Necessária cautela devido a *overplotting*

- Relação entre volume e peso

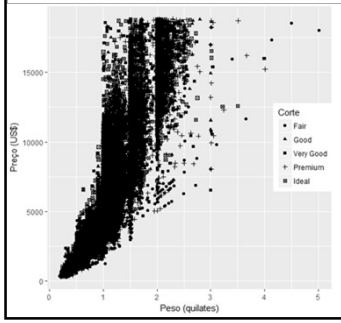
```
> # Relação entre volume (x*y*z) e peso do diamante
> ggplot(data = diamonds, aes( x = carat, y = x * y * z)) +
+ geom_point() +
+ xlab("Peso (quilates)") +
+ ylab("Volume") +
+ ggtitle("Conjunto de Dados diamonds")
```



√ Espera-se que densidade seja constante
  – Relação linear entre volume e peso
  – Maioria dos pontos parece situar-se em uma linha
  – Há *outliers* grandes

- *Scatter plot* – `price`, `carat` e `cut`

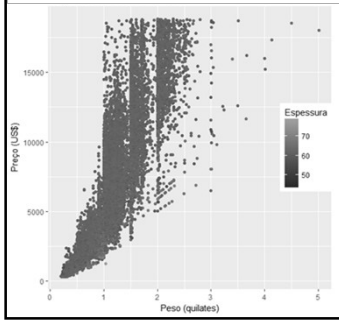
```
> # Peso vs. Preço, com caracter diferentes em cut
> ggplot(data = diamonds, aes(x = carat, y = price, shape = cut)) +
+ geom_point() +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_shape_discrete(name = "Corte") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5)
```



√ Difícil a leitura devido ao overplotting

- *Scatter plot* – `price`, `carat` e `depth`

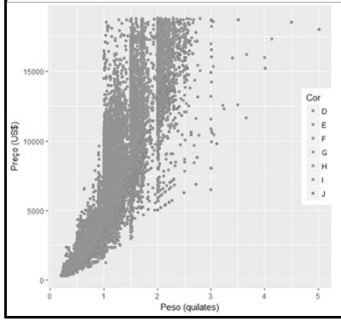
```
> # Preço vs. peso com codificação de cor para depth
> ggplot(data = diamonds, aes(x = carat, y = price, colour = depth)) +
+ geom_point() +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_color_continuous(name = "Espessura") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5))
```



√ Difícil a leitura devido ao overplotting

- *Scatter plot* – `price`, `carat` e `color`

```
> # Peso vs. Preço, com cores em color
> ggplot(data = diamonds, aes(x = carat, y = price, color = color)) +
+ geom_point() +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_color_discrete(name = "Cor") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5))
```



√ Aparentemente há diferentes relações entre `price` e `carat`, de acordo com os níveis de `color`
  – Leitura ainda prejudicada pelo overplotting

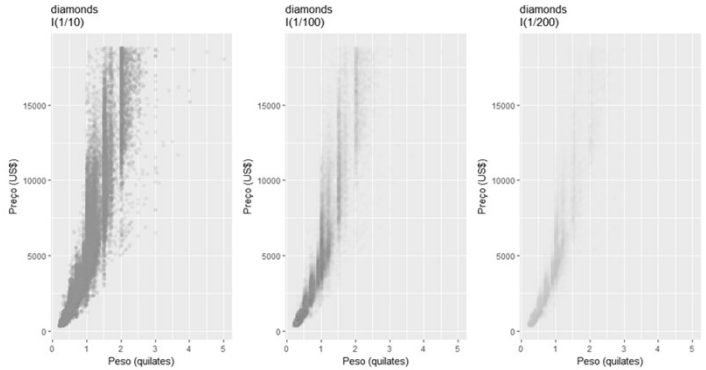
- *Scatter plot* – `price`, `carat` e `color`

√ Redução do overplotting

```
> # Peso vs. Preço, com cores em color - Redução overplotting
> dp1 <- ggplot(data = diamonds, aes(x = carat, y = price, color = color)) +
+ geom_point(alpha = I(1/10)) +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("diamonds\nI(1/10)") +
+ guides(colour=FALSE)
> dp2 <- ggplot(data = diamonds, aes(x = carat, y = price, color = color)) +
+ geom_point(alpha = I(1/100)) +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("diamonds\nI(1/100)") +
+ guides(colour=FALSE)
> dp3 <- ggplot(data = diamonds, aes(x = carat, y = price, color = color)) +
+ geom_point(alpha = I(1/200)) +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("diamonds\nI(1/200)") +
+ guides(colour=FALSE)
> library(gridExtra)
> grid.arrange(dp1, dp2, dp3, nrow=1
```

denominador especifica a quantidade de pontos que devem se sobrepor para se obter uma cor completamente opaca

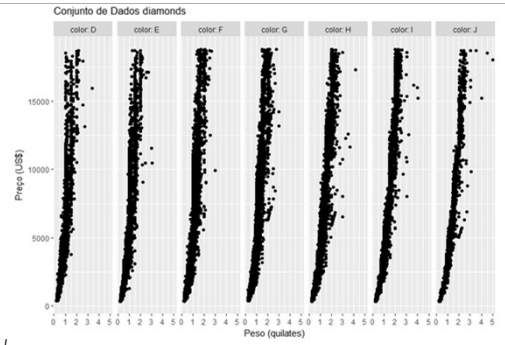
- *Scatter plot* – Redução overplotting



√ Relações diferentes para os níveis de color.

- *Scatter plot* – `price`, `carat` e `color`

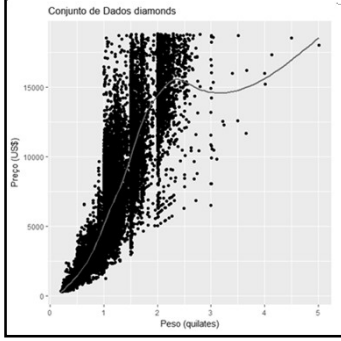
```
> # Preço vs. Peso para cada nível de color
> p + geom_point() + facet_grid(. ~ color, labeller = label_both)
```



√ Diferentes tendências de crescimento por cor

- Relação entre `price` e `carat`

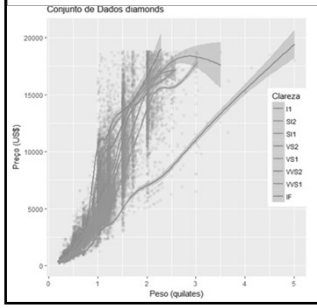
```
> # Exploração relação entre peso e preço
> ggplot(data = diamonds, aes(x = carat, y = price)) +
+ geom_point() +
+ geom_smooth(se = FALSE) +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds")
```



√ Suavização
- Conjunto de dados pequenos `method = loess`
- Conjunto de dados grandes: `method = gam`

- Relação `price`, `carat` e `clarity`

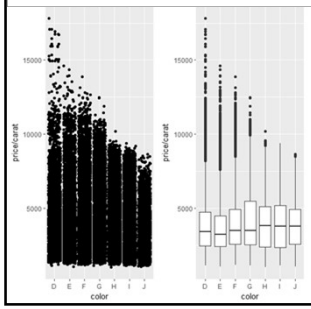
```
> # Preço vs. peso, com suavização para clarity
> ggplot(data = diamonds, aes(x = carat, y = price, colour = clarity)) +
+ geom_point(alpha = 0.1) +
+ geom_smooth() +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_color_discrete(name = "Clareza") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5))
```



√ Relação entre peso e preço aparentam ser diferentes com relação aos níveis de clareza

• Comparação preço unitário por cor

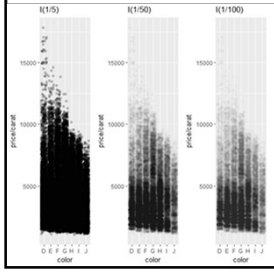
```
> # Comparação preço unitário para níveis de color
>   # gráfico de pontos
> bp1 <- ggplot(data = diamonds, aes(x = color, y = price/carat)) +
+ geom_point(position = position_jitter(width = 0.4))
>   # box-plot
> bp2 <- ggplot(data = diamonds, aes(x = color, y = price/carat)) +
+ geom_boxplot()
>   # painel
> grid.arrange(bp1, bp2, nrow=1)
```



√ À medida em que a cor melhora (da esquerda para a direita):
- Diminui a dispersão dos valores
- Há pouca alteração no centro da distribuição

• Comparação preço unitário por cor

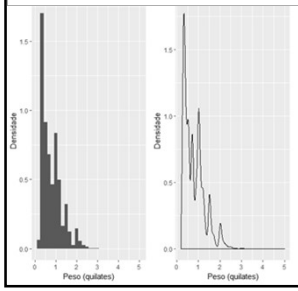
```
> # Comparação preço unitário para níveis de color - redução overplotting
> bp1 <- ggplot(data = diamonds, aes(x = color, y = price/carat)) +
+ geom_point(position=position_jitter(width=0.4), alpha = I(1/5))+
+ ggtitle("I(1/5)")
> bp2 <- ggplot(data = diamonds, aes(x = color, y = price/carat)) +
+ geom_point(position=position_jitter(width=0.4), alpha = I(1/50))+
+ ggtitle("I(1/50)")
> bp3 <- ggplot(data = diamonds, aes(x = color, y = price/carat)) +
+ geom_point(position=position_jitter(width=0.4), alpha = I(1/100))+
+ ggtitle("I(1/100)")
> grid.arrange(bp1, bp2, bp3, nrow=1)
```



√ À medida em que a opacidade diminui começamos visualiza-se onde se situa a maior parte dos dados
- Boxplot efetua melhor esta tarefa

• Distribuição do peso

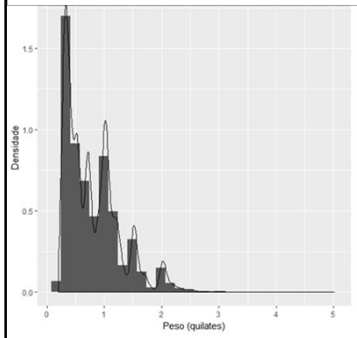
```
> # Distribuição do Peso
> hs1 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..)) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
> hs2 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_density() +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
> grid.arrange(hs1, hs2, nrow=1)
```



√ Aparentemente à grupos de preços
- Diferentes distribuições de preços

• Distribuição do peso – suavização

```
> hs3 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..)) +
+ geom_density() +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
> hs3
```

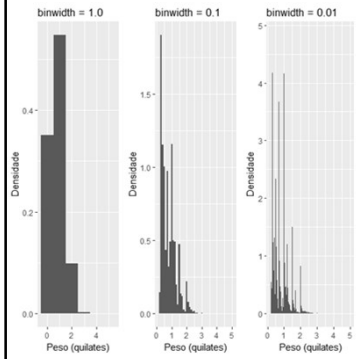


√ Aparência de multimodalidade dos dados
√ Importante tentar vários graus de suavização
- No histograma: `binwidth` controla a quantidade de suavização

- Distribuição do peso
  - √ Pesquisa da quantidade de suavização

```
> # Distribuição do peso - pesquisa da qte de suavização
>
> hs1 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..), binwidth = 1.0) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")+
+ ggtitle("binwidth = 1.0")
> hs2 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..), binwidth = 0.1) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade") +
+ ggtitle("binwidth = 0.1")
> hs3 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..), binwidth = 0.01) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade") +
+ ggtitle("binwidth = 0.01")
> grid.arrange(hs1, hs2, hs3, nrow=1)
```

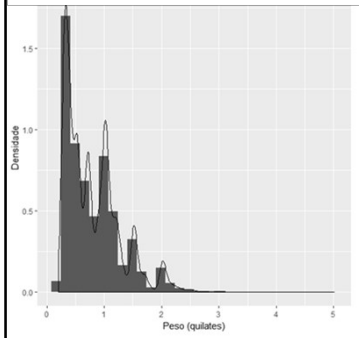

- Estimativa de densidade do preço



√ Gráfico com menor intervalo de classe apresenta as estrias percebidas
  – No scatterplot a maioria ocorre em númeors "bonitos"

- Distribuição do peso – suavização

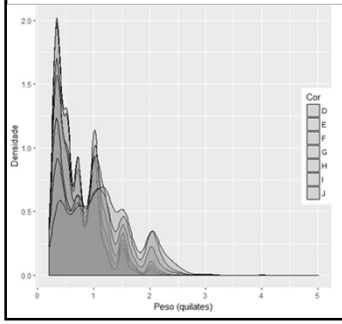
```
> hs3 <- ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..)) +
+ geom_density() +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
> hs3
```



√ Aparência de multimodalidade dos dados
√ Importante tentar vários graus de suavização
  – No histograma: `binwidth` controla a quantidade de suavização

- Densidades do peso por nível de cor

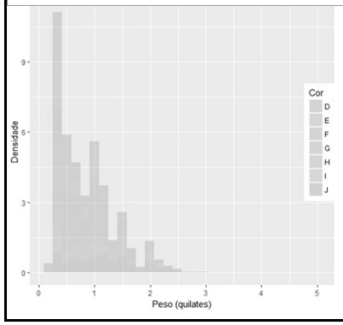
```
> # Comparação distribuição de peso entre níveis de cor
> ggplot(data = diamonds, aes(x = carat, fill = color)) +
+  geom_density(alpha = 0.3) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_fill_discrete(name = "Cor") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5))
```



√ As densidades parecem fáceis de ser lida na comparação as diferentes curvas.
√ Importante tentar vários graus de suavização
√ Supõem hipóteses que podem não ser verdade para os dados:
  – Densidade contínua, suave e ilimitada

- Histogramas do peso por nível de cor

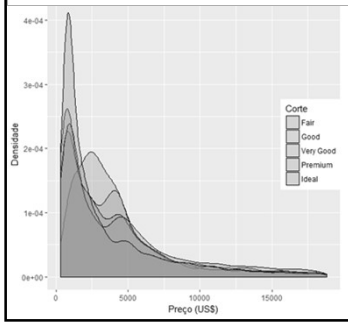
```
> # Comparação histogramas de peso entre níveis de cor
> ggplot(data = diamonds, aes(x = carat, fill = color)) +
+  geom_histogram(aes(y = ..density..), alpha = 0.3) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_fill_discrete(name = "Cor") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5))
```



√ Interpretação do gráfico é mais difícil

- Densidades de preço por nível de corte

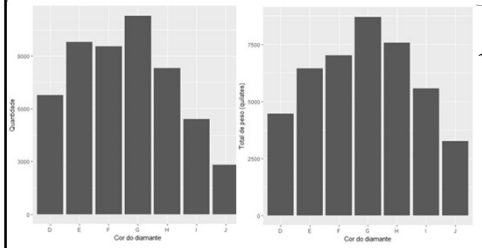
```
> # Comparação distribuição de preço entre níveis de corte
>  ggplot(data = diamonds, aes(x = price, fill = cut)) +
+  geom_density(alpha = 0.3) +
+ xlab("Preço (US$)") +
+ ylab("Densidade") +
+ ggtitle("Conjunto de Dados diamonds") +
+ scale_fill_discrete(name = "Corte") +
+ theme(legend.position = c(1, 0.5), legend.justification = c(1,0.5)
```



√ Distribuições diferentes por nível de corte
- Forte assimetria

- Distribuição da variável `color`:

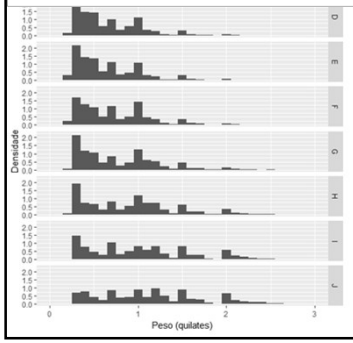
```
> # Distribuição variável color
> ggplot(data = diamonds, aes(x = color)) +
+ geom_bar() +
+ xlab("Cor do diamante") +
+ ylab("Quantidade")
> ggplot(data = diamonds, aes(x = color)) +
+ geom_bar(aes(weight = carat)) +
+ xlab("Cor do diamante") +
+ ylab("Total de peso (quilates)")
```



√ Cor dos diamantes ponderado por carat
- Peso total dos diamantes por nível de cor

- Histogramas de peso por cor:

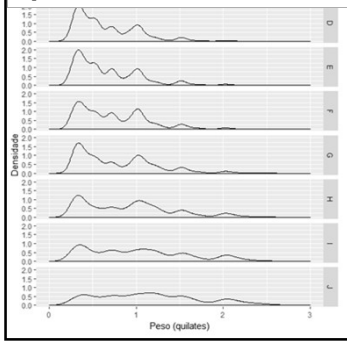
```
> # Histogramas de Peso condicionado a cor
> ggplot(data = diamonds, aes(x = carat)) +
+ geom_histogram(aes(y = ..density..), binwidth = 0.1) +
+ xlim(0, 3) +
+ facet_grid(color ~ .) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
```



√ Assimetria em direção aos menores valores para diamantes de alta qualidade (cor D)
- Distribuição torna-se mais plana à medida em que a qualidade diminui

- Histogramas de peso por cor:

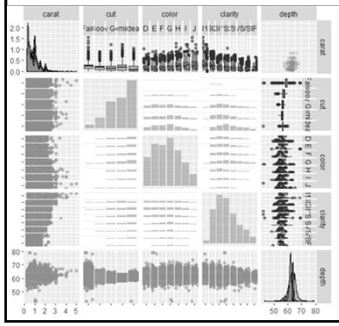
```
> # Density plot de Peso condicionado a cor
> ggplot(data = diamonds, aes(x = carat)) +
+ geom_density() +
+ xlim(0, 3) +
+ facet_grid(color ~ .) +
+ xlab("Peso (quilates)") +
+ ylab("Densidade")
```



√ Distribuições dos diamantes são mais fáceis de serem comparadas
- Ignora a quantidade de diamantes em cada nível de cor

- *Scatter plot matrix*:
  - √ `Carat`, `cut`, `color`, `clarity`, `depth`

```
> library(GGally)
> ggpairs(diamonds[,1:5], upper = list(continuous = "density", combo = "box"),
+      lower = list(continuous = "points", combo = "dot"),
+      mapping = ggplot2::aes(colour = cut, alpha = 0.4),
+    title = "Conjunto de Dados - diamonds"
+  )
```



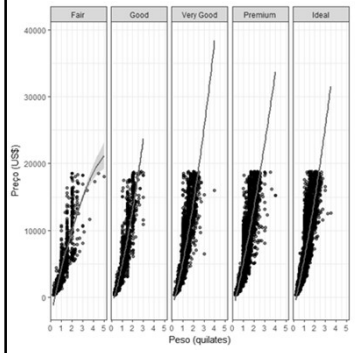
√ Visualização de pares de variáveis
- Métricas ou não métricas

- Relação entre preço e peso:
  - √ Gráficos com suavização por corte

```
> library(dplyr)
>
> diamonds %>%
+ ggplot(aes(x = carat,y = price)) +
+ geom_point(alpha = 0.5) +
+ facet_grid(~ cut) +
+ stat_smooth(method = lm, formula = y ~ poly(x,2)) +
+ xlab("Peso (quilates)") +
+ ylab("Preço (US$)") +
+ theme_bw()
```

√ Operador %>% passa a saída do operador da esquerda como o primeiro argumento para o operador da direita

- Relação entre preço e peso:
  - √ Gráficos com suavização por corte



√ Preço aumenta com tamanho do diamante
√ Relacionamento é não-linear
√ Há alguns outliers
√ Relacionamento com corte não é forte

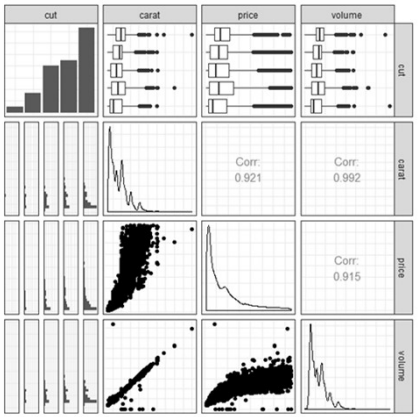
- *Scatter plot matrix*:

```
diamonds %>%
        mutate(volume = x*y*z) %>%
        select(cut, carat, price, volume) %>%
        sample_frac(0.5, replace = TRUE) %>%
        ggpairs(axisLabels = "none") +
        theme_bw()
```

√ Gráfico informativo:

- Podemos aprender muito sobre a estrutura de covariância dos dados
- Scatterplots (contínua vs. contínua) ou histogramas por grupos (contínua vs. categórica)
- Diagonal: estimativas densidades (dados contínuos), histogramas (categóricos)
- upper: correlação (dados contínuos) ou boxplots por grupos (contínuos vs. categóricos)

- *Scatter plot matrix*:



√ Facilita visualização dos dados

Referências

Bibliografia Recomendada

- DALGAARD, P. *Introductory statistics with R*. Springer, 2002.
- MURRELL, P. *R graphics*. Chapman & Hall, 2006.
- EVERITT, B.; HOTHORN, T. *An Introduction to Applied Multivariate Analysis with R*. Springer, 2011.
- ZELTERMAN, D. *Applied Multivariate Statistics with R*. Springer, 2015.