Capítulo 3

Estatística Descritiva e Gráficos Gerais

Gustavo Mello Reis
José Ivo Ribeiro Júnior

Universidade Federal de Viçosa
Departamento de Informática
Setor de Estatística

Viçosa 2007

Neste capítulo serão apresentadas algumas estatísticas descritivas e gráficos freqüentemente utilizados na área de controle de qualidade, com o objetivo de medir a posição, a variação e a distribuição da variável reposta Y, de forma geral ou estratificada.

1. Medidas

Como exemplo, considere os dados dos vetores Y e X, no R:

```
Y<-c(20,23,23,28,33,37,37,37,40,44)  # Entrar com Y
X<-c(1,1,1,1,1,2,2,2,2,2)            # Entrar com X
```

Pode-se calcular a média de Y, de forma geral ou para cada nível de X, da seguinte forma:

```
mean(Y)              # média de Y
[1] 32.2
tapply(Y, X, mean)  # média de Y para cada nível de X
   1    2
25.4 39.0
```

A função tapply pode ser usada, não só para a média, mas para qualquer outra medida, de duas maneiras:

a) tapply(variável Y, variável X, medida);

b) tapply(variável Y, c(variável X, variável XX), medida).

Neste último caso, a variável Y será estratificada em dois fatores diferentes (variável X e variável XX).

A mediana de Y é dada por:

```
median(Y)            # mediana de Y
[1]
35
tapply(Y, X, median)  # mediana de Y para cada nível de X
 1  2
23 37
```

Já os percentis, que por padrão no R são calculados o P_o, P_{25}, P_{50}, P_{75} e P_{100}, são obtidos através de:

```
quantile(Y)
   0%   25%   50%   75%  100%
```

```
20.00 24.25 35.00 37.00 44.00
```

A especificação de outros percentis pode ser feita por:

quantile(Y, c(0.1, 0.15, 0.615, 0.89999, 0.99))

```
     10%       15%     61.5%   89.999%       99%
22.70000 23.00000 37.00000 40.39964 43.64000
```

No R, existe a função summary capaz de resumir vários tipos de objetos, como por exemplo:

summary(Y)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   24.25   35.00   32.20   37.00   44.00
```

tapply(Y,X,summary)

```
$"1"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   20.0    23.0    23.0    25.4    28.0    33.0


$"2"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     37      37      37      39      40      44
```

A função table fornece as freqüências dos diferentes valores de Y. Quando a amostra é pequena, pode-se obter a moda através desta função. Caso contrário, utiliza-se a função subset em conjunto com a função table para que retorne apenas a moda, como segue:

table(Y) # Para amostras pequenas

```
Y
20 23 28 33 37 40 44
 1  2  1  1  3  1  1
```

subset(table(Y), table(Y)= =max(table(Y))) # Retorna apenas a moda

```
37
 3
```

As medidas de dispersão variância, desvio padrão, erro padrão da média, coeficiente de variação e amplitude total, da variável Y, são calculadas, respectivamente, por:

var(Y) # Variância (S_Y^2)

```
[1] 67.28889
```

sd(Y) # Desvio padrão (S_Y)

```
[1] 8.20298
```

sd(Y)/sqrt(length(Y)) # Erro padrão da média ($S_{\bar{Y}}$)

```
[1] 2.59401
```

100*sd(Y)/mean(Y) # Coeficiente de variação ($\hat{CV}_Y$)

```
[1] 25.47509
```

max(Y) - min(Y) # Amplitude total ($\hat{AT}_Y$)

```
[1] 24
```

range(Y) # Exibe o menor e o maior valor de Y, respectivamente

```
[1] 20  44
```

2. Intervalos de Confiança

Como exemplo, considere o vetor de dados da variável Y:

```
Y<-c(11, 16, 22, 27, 31, 35, 39, 43, 47, 50)
```

Para a média de Y, serão estimados dois intervalos com 100(1-α)% de confiança, um com base na distribuição de z e outro com base na de t, dados, respectivamente por:

a) $IC(\mu_Y)_{1-\alpha} : \overline{Y} \pm z_{\alpha/2}\sigma_{\overline{Y}}$; onde: $\sigma_{\overline{Y}} = \sigma_Y / \sqrt{n}$

b) $IC(\mu_Y)_{1-\alpha} : \overline{Y} \pm t_{\alpha/2}S_{\overline{Y}}$. onde: $S_{\overline{Y}} = S_Y / \sqrt{n}$

No primeiro caso, será considerado $\sigma_Y = 5$ e, os comandos no R, para a obtenção de um intervalo com 100(1-0.05)% de confiança, são fornecidos a seguir:

```
alfa<-0.05
z<-qnorm(1-alfa/2)   # Valor de z para α = 0.05
n<-length(Y)         # Tamanho da amostra
sinal<-c(-1,+1)      # Sinal mais ou menos
mean(Y)+sinal*z*5/sqrt(n)  # Fórmula: Ȳ ± z_{α/2}σ_Ȳ
```

```
[1] 29.00102 35.19898
```

Os resultados correspondem aos limites inferior e superior do intervalo.

No segundo caso, os comandos no R, para a obtenção de um intervalo com 100(1-0.05)% de confiança, são:

```
gl<-n-1                  # Graus de liberdade
alfa<-0.05
t<-qt(1-alfa/2, gl)   # Valor de t para α = 0.05
n<-length(Y)          # Tamanho da amostra
sinal<-c(-1,+1)       # Sinal mais ou menos
mean(Y)+sinal*t*sd(Y)/sqrt(n)   # Fórmula: Ȳ ± t_{α/2} S_Ȳ
```

```
[1] 22.7094 41.4906
```

Do mesmo modo, ao final, têm-se os limites inferior e superior do intervalo.

3. Gráficos

No R, cada tipo de gráfico possui uma função específica. No entanto, algumas de suas configurações, serão controladas pela função par, de acordo com os argumentos apresentados na Tabela 1:

Tabela 1. Argumentos para função par

mfrow	divide a janela onde os gráficos serão construídos, cujo valor é do tipo c(nl, nc), em que nl é o número de linhas e nc o número de colunas em que a janela será dividida
mfcol	idêntica à mfrow, porém seu valor será do tipo c(nc, nl)
ps	controla o tamanho de todos os textos nos gráficos, cujo valor deve ser um número inteiro
pty	indica a área em que o gráfico será construído, seus valores são: “m” (área máxima) ou “s” (área quadrada)
bg	controla a cor de fundo da janela dos gráficos, sendo que as 657 cores disponíveis são visualizadas por meio do comando colors( )
fg	controla a cor dos eixos e das bordas dos símbolos dos gráficos
col.main, col.lab, col.sub, col.axis	controla as cores do título, dos nomes dos eixo, do rodapé e dos valores dos eixos, respectivamente
cex.main, cex.lab, cex.sub, cex.axis	controla o tamanho da fonte, do título, dos nomes dos eixos, do rodapé e dos valores dos eixos, respectivamente, sendo que os valores positivos menores ou maiores do que 1, diminuem ou aumentam o tamanho, respectivamente
font.main, font.lab, font.sub, font.axis	controla a fonte a ser usada, com base em números inteiros de 1 a 20, sendo que o número 1 indica texto normal, o 2 negrito, o 3 itálico e o 4 negrito + itálico

adj	indica a posição dos textos através de valores de 0 a 1, sendo que 0 indica texto à esquerda, 0.5 centralizado e 1 à direita
las	indica a posição dos valores nos eixos x e y dos gráficos, sendo o padrão neste material de las=1, que indica que os valores dos eixos x e y devem ser postos na horizontal

Os argumentos, com exceção dos quatro primeiros, podem também ser usados nas funções responsáveis por criar os gráficos. A diferença é que quando são utilizados na função "par", as suas propriedades serão aplicadas em todos os gráficos criados, enquanto a janela dos gráficos não for fechada. Quando são utilizados dentro de uma função específica de um gráfico, os seus efeitos serão aplicados apenas naquele gráfico.

A cópia dos gráficos do R para editores de textos, como por exemplo, Word e Writer (do pacote OpenOffice), podem ser feitas sob duas opções:

a) pressionar Ctrl+c para copiar e Ctrl+v para colar o gráfico (apenas no tamanho da janela reduzida);

b) clicar com o botão direito sobre o gráfico e depois com o esquerdo, em "Copy as metafile" para copiar e, Ctrl+v, para colar o gráfico (no tamanho em que estiver apresentado no R).

Para salvar os gráficos gerados no R, durante uma sessão, deve-se estar com a janela dos mesmos aberta e ativa, ou seja, à frente das outras janelas. Neste caso, o menu do R será modificado em relação ao menu principal e, neste novo menu, clicar em "Histórico" e em "Gravando". Para acessá-los posteriormente, basta teclar PageUp ou PageDown para ver os gráficos anteriores ou posteriores, respectivamente. O processo de gravação será finalizado após clicar, novamente, em "Histórico" e em "Gravando", desmarcando assim, esta opção. Porém, os gráficos gravados anteriormente continuarão disponíveis. Caso haja necessidade de deletá-los, deve-se clicar em "Histórico" e em "Limpar Histórico". Outra maneira, é salvá-los em um objeto dentro do R. Para isso, deve-se clicar em "Histórico" e em "Salvar para variável...", digitar um nome para o objeto e clicar em "OK", cujo objeto estará disponível apenas na sessão que foi salvo. Para ver os gráficos, deve-se clicar em "Histórico" e em "Pegar da variável...", digitar o nome do objeto e clicar em "OK". E, para excluir o objeto, deve-se digitar no console o comando: rm(nome.do.objeto).

Além disso, os gráficos podem ser salvos em um arquivo externo ao R. Com a janela do gráfico de interresse aberta, clicar em "Arquivo" e em "Salvar como" e escolher um formato para salvar o gráfico. Os formatos mais usuais e mais compatíveis com a maioria dos programas de imagens, são Bmp (Bitmap) e Jpeg.

3.1. Histograma

Como exemplo, será simulada uma variável Y em duas amostras com $n_1 = 50$ e $n_2 = 50$, baseada nas distribuições normais com $\mu_{Y1} = 150$ e $\sigma_{Y1} = 20$, para X_1 e $\mu_{Y2} = 100$ e $\sigma_{Y2} = 20$, para X_2. A simulação no R será feita da seguinte forma:

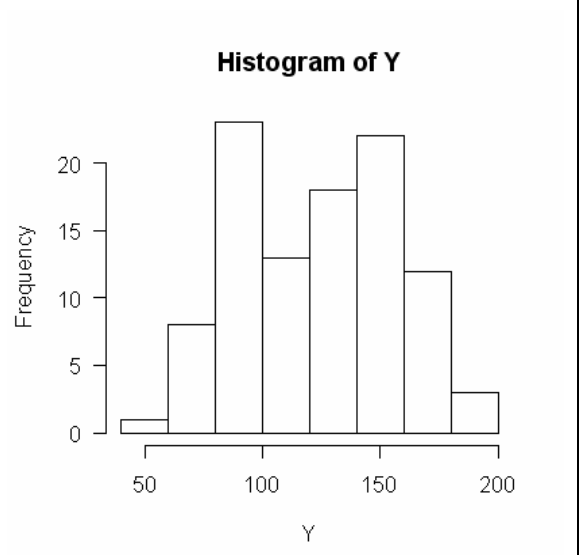
```
Y1<-rnorm(50,150,20) # Primeira simulação
Y2<-rnorm(50,100,20) # Segunda simulação
Y<-c(Y1, Y2)         # Agrupar as duas simulações em um mesmo vetor
rm(Y1, Y2)           # Remover os vetores Y1 e Y2
X<-gl(2, 50)         # Construir o vetor X com os dois níveis X1 e X2
```

Dois histogramas serão criados:

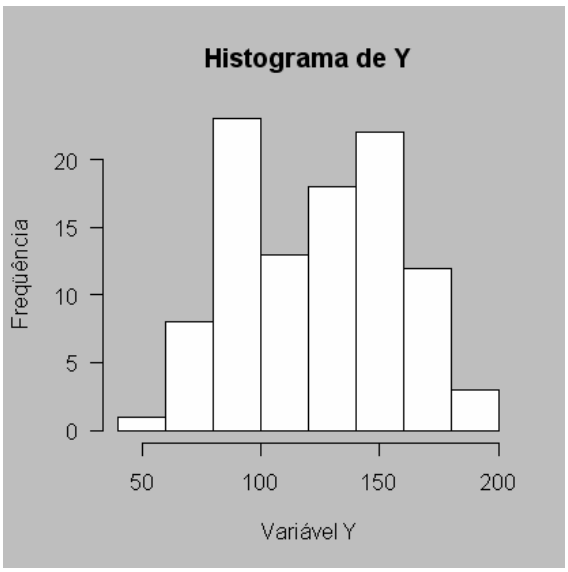
```
par(las=1)          # Exibir os valores dos eixos x e y na horizontal
hist(Y)             # Figura 1a
par(bg="grey")      # Modificar a cor de fundo para cinza
# Figura 1b
hist(Y, col="white", main="Histograma de Y", ylab="Freqüência", xlab="Variável Y")
```

Figura 1. Histograma

a



b

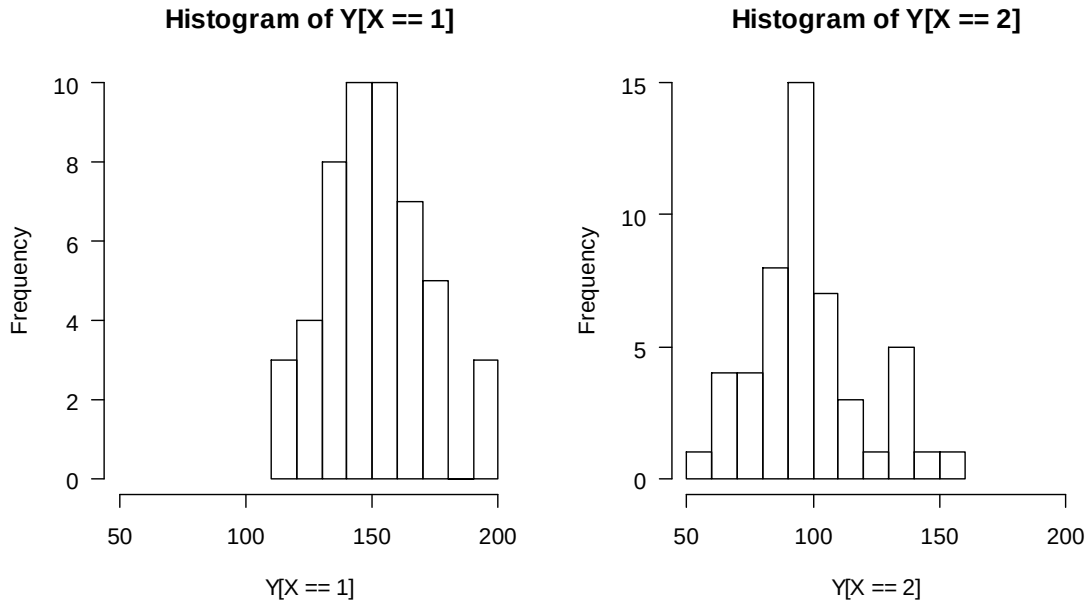


Para construir o histograma estratificado (Figura 2) de acordo com os níveis da variável X na mesma janela, serão utilizados os comandos a seguir:

```
par(bg = "white")       # Modificar a cor de fundo para branco
par(mfrow=c(1,2))       # Dividir a janela dos gráficos em uma linha e duas colunas
hist(Y[X= =1], xlim=c(50,200)) # Estratificar Y e estabelecer limites para o eixo x
```

hist(Y[X= =2], xlim=c(50,200)) # Estratificar Y e estabelecer limites para o eixo x

Figura 2. Histograma estratificado



3.2. Gráfico de Ramos e Folhas

Para a variável Y simulada com n = 100 (n_1 + n_2), tem-se:

stem(Y)

```
The decimal point is 1 digit(s) to the right of the |

   4 | 9
   6 | 55601666
   8 | 34448888000122456668899
  10 | 3345567134679
  12 | 47890012234667889
  14 | 01134555688902345556799
  16 | 013449911267
  18 | 258
```

Caso haja necessidade de estratificar o gráfico, deve-se proceder de acordo com o exemplo:

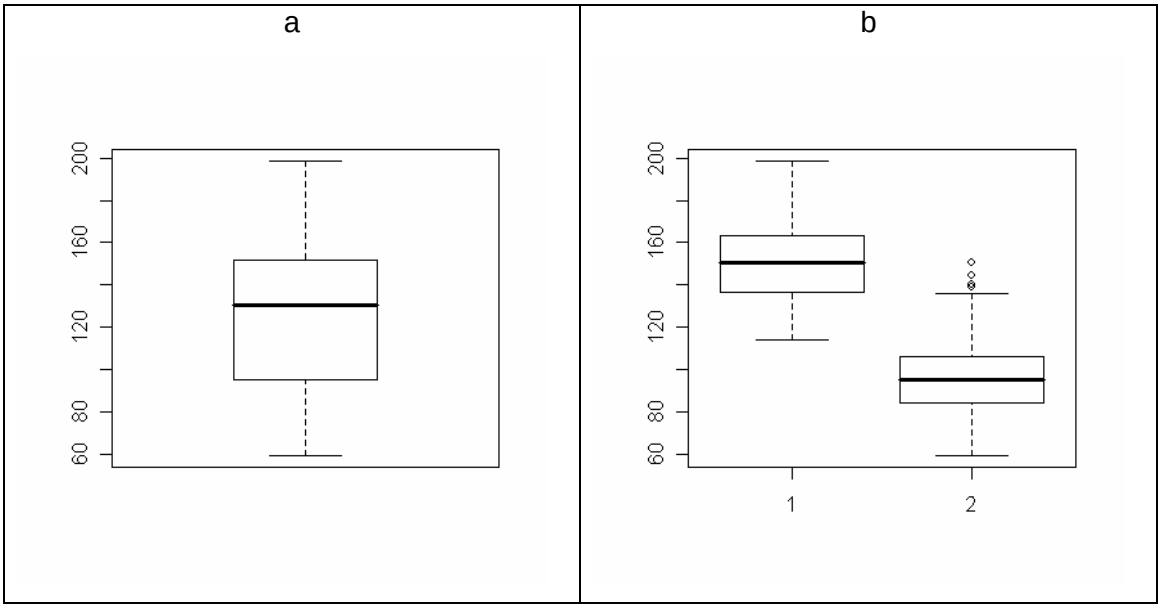
stem(Y[X= =1])
stem(Y[X= =2])

3.3. Box-plot

Para a variável Y, tem-se:

boxplot(Y) # Box-plot simples para n = 100 (Figura 3a)
boxplot(Y~X) # Box-plot estratificado por X para $n_1 = 50$ e $n_2 = 50$ (Figura 3b)

Figura 3. Box-plot simples e estratificado



3.4. Gráfico de Pareto

O gráfico de Pareto pode ser construído tanto para efeitos (variáveis Ys) como para causas (variáveis Xs), cujo objetivo é de destacar os níveis prioritários da variável estudada.

As variáveis são, normalmente, expressas em número de ocorrências ou em unidades monetárias, sendo que o gráfico pode ser estratificado com o objetivo de verificar se o comportamento do processo, a nível geral, ocorre ou não de acordo com as diferentes particularidades do mesmo.

Como exemplo, considere que uma indústria girou o ciclo PDCA com o objetivo de diminuir o número de televisores defeituosos. A amostragem foi feita sobre a produção de um mês de acordo os tipos de defeitos (X) e estratificada em função dos locais 1 e 2 de produção (XX). No estudo, foram analisadas duas variáveis: número de ocorrência de cada tipo de defeito (Y) e custo devido ao tipo de defeito (YY).

A entrada de dados pode ser feita da seguinte forma:

dados.p<-read.csv2(“pareto.csv”, dec=”.”)
dados.p

XX	X	Y	YY
1	def.A	15	30
1	def.B	12	60
1	def.C	6	120
1	def.D	4	40
1	def.E	7	25
2	def.A	6	12
2	def.B	16	80
2	def.C	12	240
2	def.D	6	60
2	def.E	2	15

```
attach(dados.p)
```

A função utilizada para construir o gráfico de pareto faz parte do pacote qcc, que deverá ser instalado, caso ainda não tenha sido. Após a instalação, deve-se ativá-lo na sessão:

```
library(qcc)        # Ativar o pacote qcc
```

As barras serão organizadas de acordo com a ordem de freqüência, sendo que por padrão, a ordem das linhas é associada com a ordem das letras do alfabeto. As cores, por padrão (default), terão uma tonalidade vermelha que diminui com a importância da barra.

O gráfico de Pareto para o número de defeitos (Y) será construído, de forma estratificada em XX, da seguinte forma:

```
names(Y)<-X             # Atribuir os nomes dos tratamentos aos valores de Y
par(mfrow=c(1,2))       # Ver os dois gráficos na mesma janela
pareto.chart(Y[XX= =1], las=1)  # Figura 4a
```

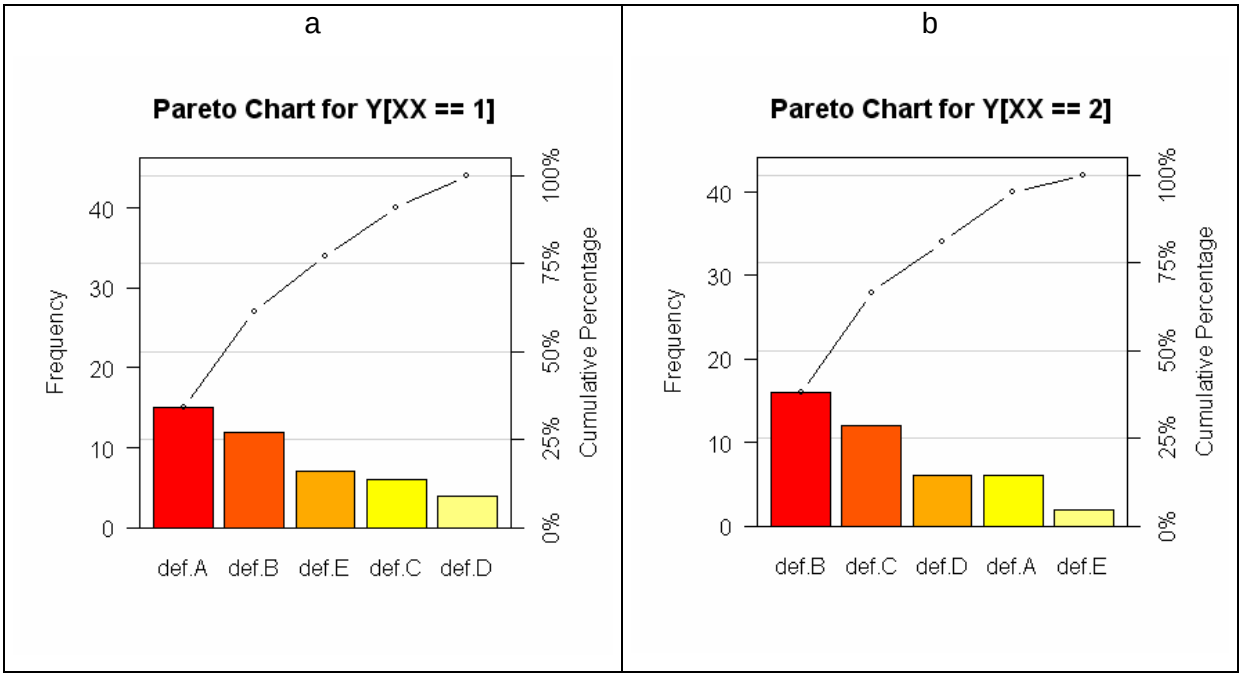
```
Pareto chart analysis for Y[XX == 1]
        Frequency Cum.Freq. Percentage Cum.Percent.
  def.A        15        15   34.09091     34.09091
  def.B        12        27   27.27273     61.36364
  def.E         7        34   15.90909     77.27273
  def.C         6        40   13.63636     90.90909
  def.D         4        44    9.09091    100.00000
```

```
pareto.chart(Y[XX= =2], las=1) # Figura 4b
```

```
Pareto chart analysis for Y[XX == 2]
        Frequency Cum.Freq. Percentage Cum.Percent.
```

def.B	16	16	38.095238	38.09524
def.C	12	28	28.571429	66.66667
def.D	6	34	14.285714	80.95238
def.A	6	40	14.285714	95.23810
def.E	2	42	4.761905	100.00000

Figura 4. Gráficos de Pareto para Y estratificados em função de XX



Alguns nomes no eixo x podem não aparecer no gráfico, para não haver sobreposição de nomes, não foi este caso. No entanto, uma forma de evitar esta falha é alinhar os nomes do eixo x na vertical, usando o argumento las=2.

A construção do gráfico de Pareto para os custos dos defeitos (YY) segue os mesmos passos anteriores:

```
names(YY)<-X            # Atribuir os nomes dos tratamentos aos valores de Y
pareto.chart(YY[XX= =1], las=1)
pareto.chart(YY[XX= =2], las=1)
```

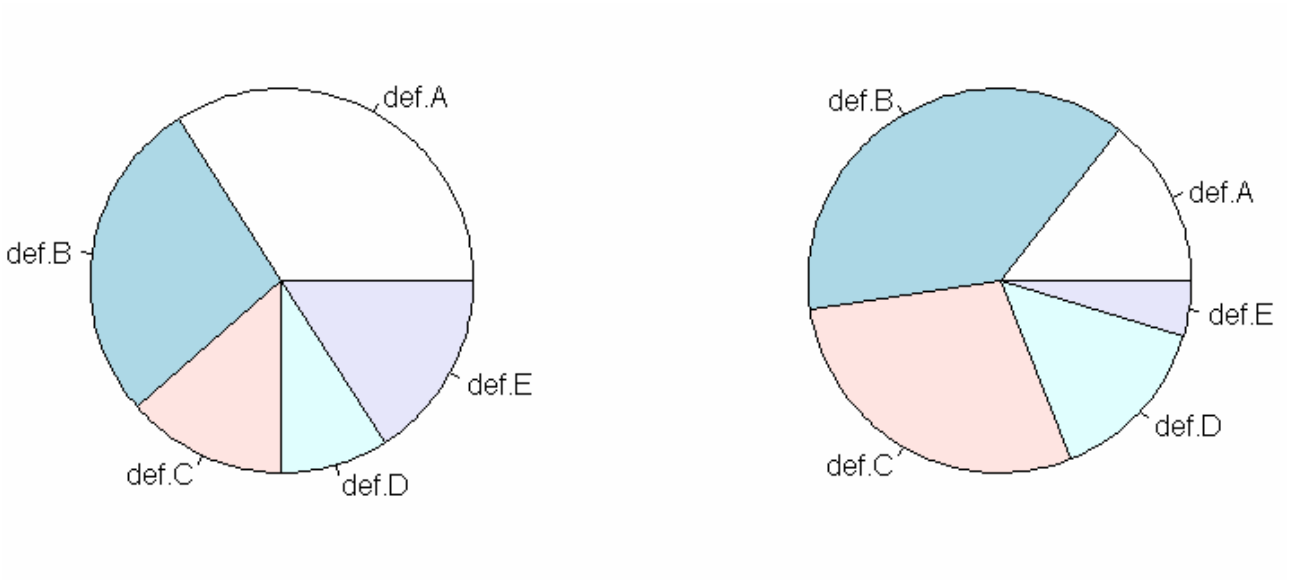
3.5. Gráfico de Pizza

Como exemplo considere os dados armazenados no objeto “dados.p”. De forma geral e estratificada em XX, os gráficos de Pizza são construídos por meio de:

```
par(mfrow=c(1,2), pty= “s”)
names(Y)<-X
```

pie(Y[XX= =1])
pie(Y[XX= =2])

Figura 5. Gráficos de Pizza para Y estratificados em função de XX



3.6. Diagrama de Dispersão

Este gráfico é utilizado com o objetivo de mostrar a relação entre duas variáveis Ys, entre uma variável X e uma variável Y ou entre duas variáveis Xs.

Numa empresa metal-mecânica, após perceber que o desgaste das ferramentas de corte nos tornos parecia excessivo, foi solicitado ao encarregado pelo controle de qualidade que comprovasse ou não a suspeita. Para isso, ele realizou um experimento que consistiu em variar a velocidade do torno (Y) e observar o tempo em que a ferramenta era capaz de manter a afiação para a qual foi padronizada (YY). Ao todo, foram utilizadas 22 ferramentas de dois fornecedores (X).

No R, foram utilizados os seguintes dados do arquivo C:\Rdados\dispersão.csv, como segue:

dados.disp<-read.csv2(“dispersão.csv”,dec=”.”) # Entrada de dados
dados.disp

X	Y	YY
A	30	25
A	30	23
A	32	30
B	32	16
A	34	30

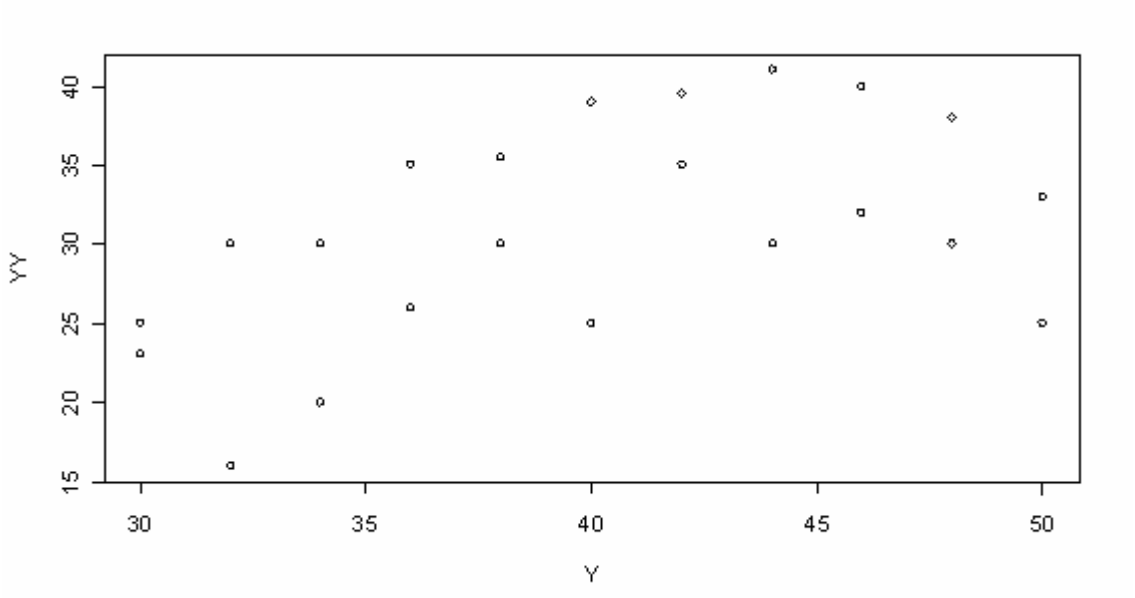
B	34	20
A	36	35
B	36	26
A	38	35.5
B	38	30
A	40	39
B	40	25
B	42	35
A	42	39.5
B	44	30
A	44	41
B	46	32
A	46	40
B	48	38
B	48	30
B	50	25
B	50	33

```
attach(dados.disp)
```

O digrama de dispersão (Figura 6) será construído com o seguinte comando:

```
plot(YY~Y)
```

Figura 6. Diagrama de dispersão geral



De acordo com a Figura 6, parece que a variável Y, velocidade do torno, não possui relação com a variável YY, tempo em que mantêm a afiação. Para confirmar, será calculado o coeficiente de correlação (r) entre as duas variáveis:

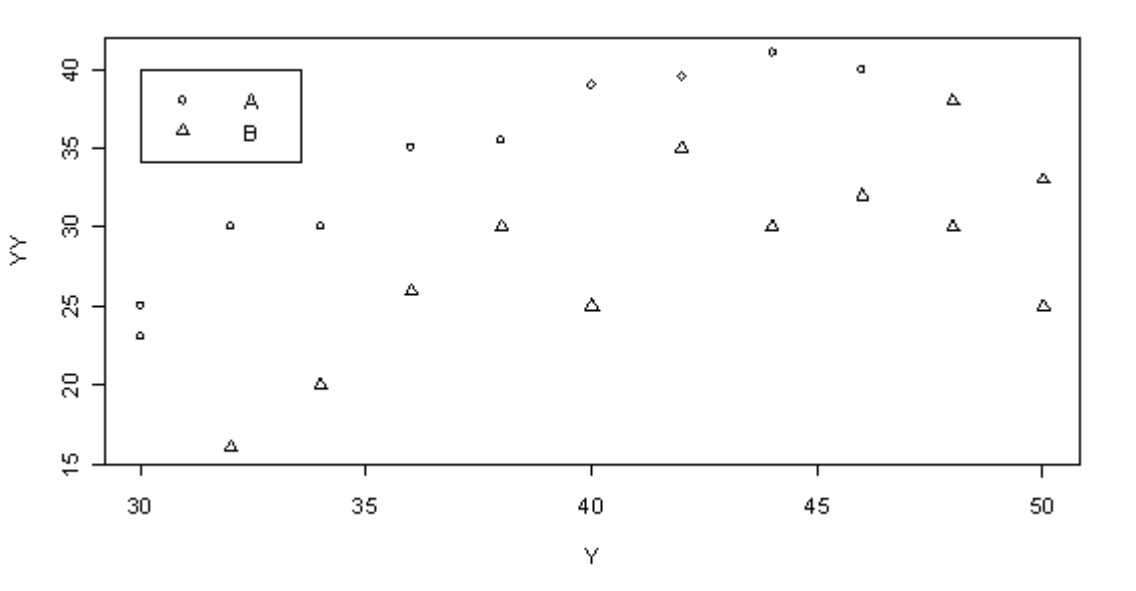
```
cor(Y,YY)   # r
[1] 0.5041802
```

No entanto, após a identificação dos tempos (YY) referentes às ferramentas de cada fabricante, uma terceira variável (X) foi acrescentada e o diagrama estratificado (Figura 7) foi construído da seguinte forma:

```
plot(Y,YY,pch=c(1,2)[unclass=X])) # construir o diagrama estratificado por X
legend(30,40,c("A","B"),pch=c(1,2)) # construir uma legenda na coordenada (30, 40)
```

O argumento "pch" indica o tipo de símbolo que será usado, podendo receber valores de 1 a 25 e também qualquer caractere diretamente especificado utilizando aspas, como por exemplo, pch = "*".

Figura 7. Diagramas de dispersão estratificados em função de X



De acordo com a Figura 7, é possível observar que o aumento de Y dentro do intervalo de X estudado, ou seja, fabricantes A e B, faz com que exista uma tendência do aumento de YY. Além disso, observa-se que as ferramentas do fabricante A parecem ser mais resistentes que as do B, visto que a dispersão dos dados referentes às ferramentas de A está acima das de B, ou seja, o tempo que as ferramentas de A mantêm a afiação é maior que das de B, para uma dada velocidade do torno (Y).

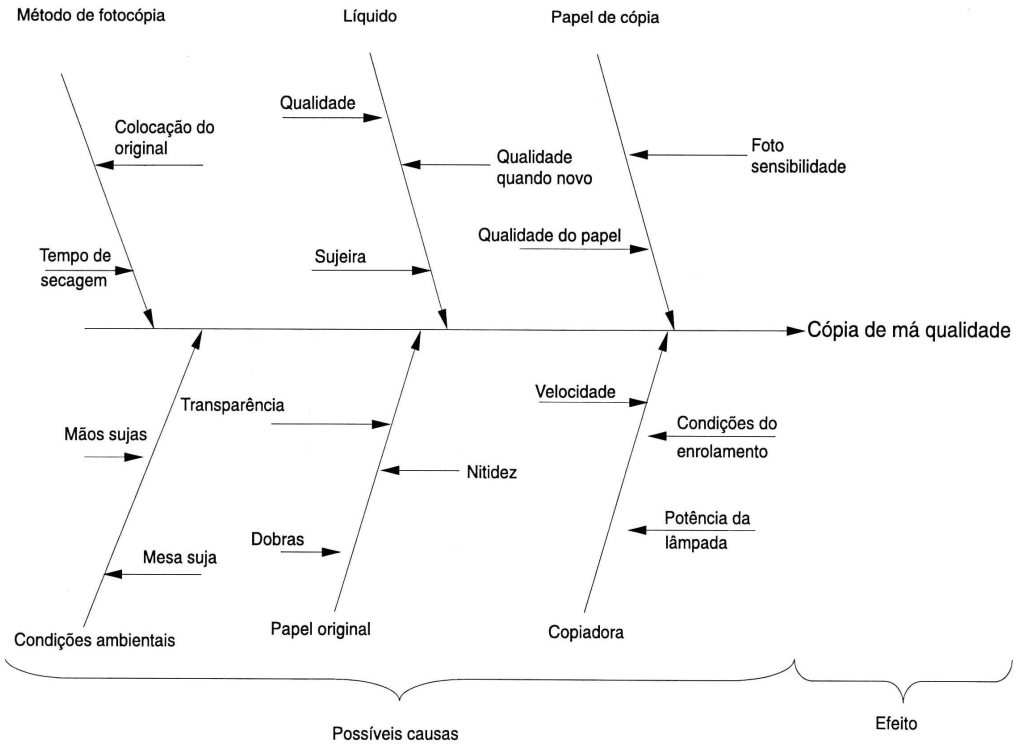
Para confirmar a tendência de aumento de YY, tempo capaz de manter a afiação, com um acréscimo em Y, velocidade do torno, serão calculados os coeficientes de correlação para cada fabricante:

```
cor(Y[X= =“A”], YY[X= =“A”])  # fabricante A
[1] 0.950448
cor(Y[X= =“B”], YY[X= =“B”])   # fabricante B
[1] 0.6917471
```

3.7. Diagrama de Causa e Efeito

O exemplo do capítulo anterior para a construção do Diagrama de Causa e Efeito será repetido aqui: considere uma pequena empresa do ramo de cópias e impressões. Depois de ter passado por um período muito bom de crescimento, devido ao seu preço diferenciado em relação ao dos concorrentes, surgiu uma fase de muitas dificuldades. O dono da empresa, percebendo que estava perdendo a sua clientela, tentou identificar o problema que vinha acontecendo. Preparou alguns questionários e os aplicou diretamente aos seus clientes, na maioria estudantes. Depois de respondidos e analisados por ele próprio, a conclusão foi que a qualidade das cópias das outras empresas eram bastante superiores. Sem perder tempo, tratou de convocar uma reunião com todos os funcionários, de forma a buscar as causas do problema observado, ou seja, a cópia de má qualidade (Figura 8)

Figura 8. Diagrama de causas primárias e secundárias



No R, o diagrama (Figura 9) será criada pela função cause.and.effect, que faz parte do pacote qcc. Esta função possui cinco argumentos: cause = causas primárias (sem.espaços.no.nome) e secundárias, effect = efeito, tittle = título, cex = tamanho dos textos e font = fonte do texto. De acordo com a Figura 8, têm-se:

```
library(qcc)  # Ativar o pacote qcc
cause.and.effect(cause = list(Método.de.foto.cópia = c("Colocação do Original", "Tempo de secagem"), Liquido = c("Qualidade", "Qualidade quando novo", "Sujeira"), Papel.de.cópia = c("Qualidade do papel", "Foto sensibilidade"), Condições.ambientais = c("Mesa suja", "Mãos sujas"), Papel.original = c("Dobras", "Nitidez", "Transparência"), Copiadora = c(" Velocidade", "Condições do enrolamento", "Potência da lâmpada")), effect = "Cópia de má qualidade", cex = c(0.7, 0.5, 1))
```

Figura 9. Diagrama de Causa e Efeito no R