EDA-R

Danilo A C Souto

Amostragem

Amostragem é uma técnica para remover amostras de um conjunto de dados. Pode ser utilizado para facilitar uma análise primária. Mas lembro que, sempre que possível, deve-se utilizar o conjunto todo de dados para as análises.

Para retirar uma amostra pode-se utilizar a função `sample(x, size, replace = FALSE, prob = NULL)`

```
sample(airquality$Temp, size=15, replace = TRUE)
```

```
##  [1] 66 86 81 93 79 82 90 61 83 81 74 82 76 82 82
```

Amostra sem reposição

```
sample(airquality$Temp, size=15, replace = FALSE)
```

```
##  [1] 93 66 76 80 77 82 64 73 84 88 86 92 72 67 81
```

Caso queira amostrar todos os atributos do data frame pode-se utilizar a técnica de selecionar pelo índice.

```
i <- 1:nrow(airquality)
s <- sample(i, size = 5)
```

	Ozone	Solar.R	Wind	Temp	Month	Day
24	32	92	12.0	61	5	24
34	NA	242	16.1	67	6	3
11	7	NA	6.9	74	5	11
88	52	82	12.0	86	7	27
138	13	112	11.5	71	9	15

Estrutura dos dados

Uma forma de visualizar a estrutura de dados é a partir do comando `str()`. Desta forma podemos ver todos os atributos e o tipo de cada atributo.

Ozone	Solar.R	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6
23	299	8.6	65	5	7
19	99	13.8	59	5	8
8	19	20.1	61	5	9
NA	194	8.6	69	5	10

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
nas <- airquality[is.na(airquality$Ozone), ]
```

	Ozone	Solar.R	Wind	Temp	Month	Day
5	NA	NA	14.3	56	5	5
10	NA	194	8.6	69	5	10
25	NA	66	16.6	57	5	25
26	NA	266	14.9	58	5	26
27	NA	NA	8.0	57	5	27
32	NA	286	8.6	78	6	1
33	NA	287	9.7	74	6	2
34	NA	242	16.1	67	6	3
35	NA	186	9.2	84	6	4
36	NA	220	8.6	85	6	5

Variável Categórica

Para variáveis categóricas podemos utilizar as tabelas de frequência. A função `table()` faz a contagem dos elementos.

```
knitr::kable(mtcars[1:10,])
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

```
df <- mtcars
df$cyl <- factor(df$cyl,
                 levels = c(4,6,8),
                 labels = c("4 cilindros", "6 cilindros", "8 cilindros"))

df$am <- factor(df$am,
                 levels = c(0,1),
                 labels = c("Automático", "Manual"))
```

```r
table(df$am)
```

```
##
## Automático     Manual
##         19         13
```

```r
freq <- table(df$am)
freq
```

```
##
## Automático     Manual
##         19         13
```

Para criar a frequência relativa, podemos utilizar a funçõ `prop.table()`passando como argumento o resultado da função `table()`

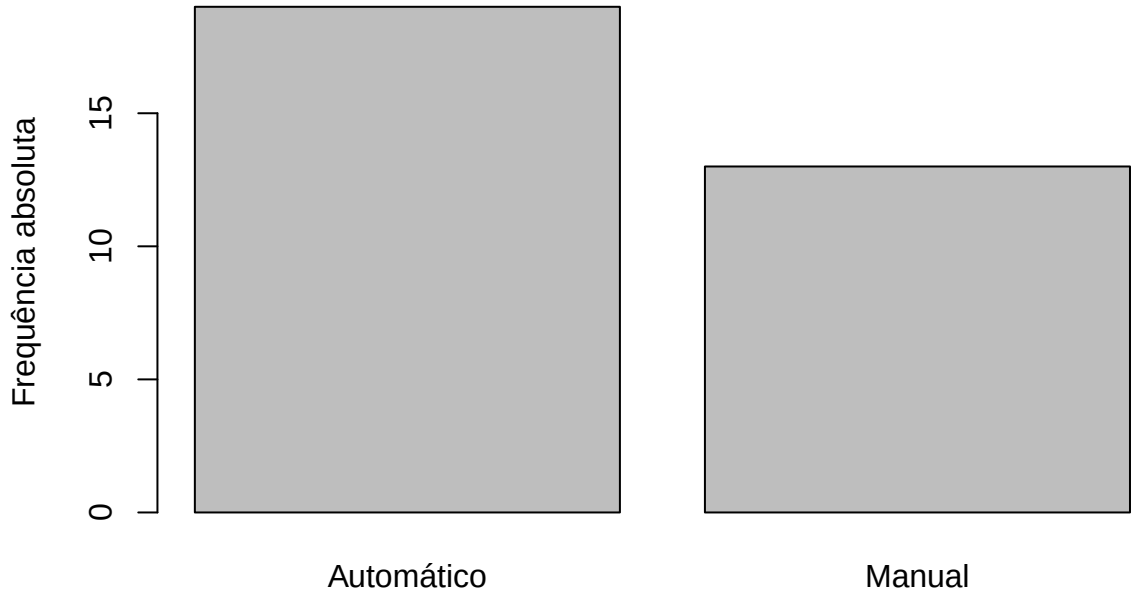
```r
m_freq <- cbind(freq,r=prop.table(table(mtcars$am)))
m_freq
```

```
##            freq       r
## Automático   19 0.59375
## Manual       13 0.40625
```

Gráfico de Barras

Outra ferramenta bastante útil bara uma visualização rápida desse dados é o gráfico de barras. Com ele temos uma boa ideia das escalas e dimensões de cada categoria.

```r
barplot(m_freq[,1], ylab = "Frequência absoluta")
```



Também pode-se utilizar a frequência relativa para construir o gráfico de barras como pode ser visto a seguir.

```r
barplot(m_freq[,2], ylab = "Frequência relativa", ylim = c(0, .6))
```

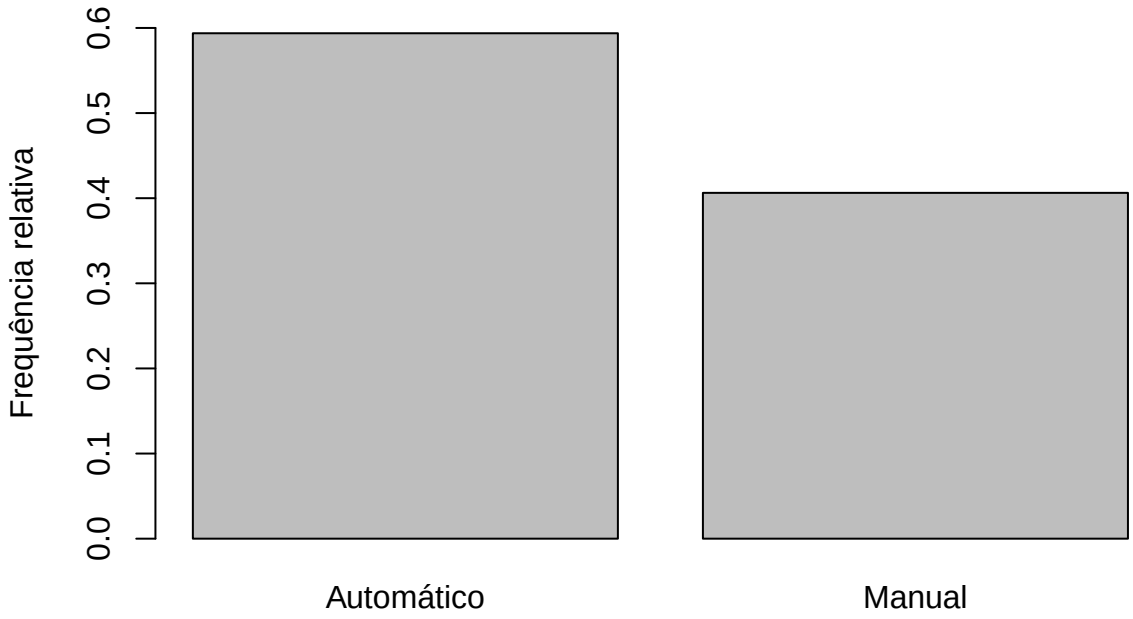
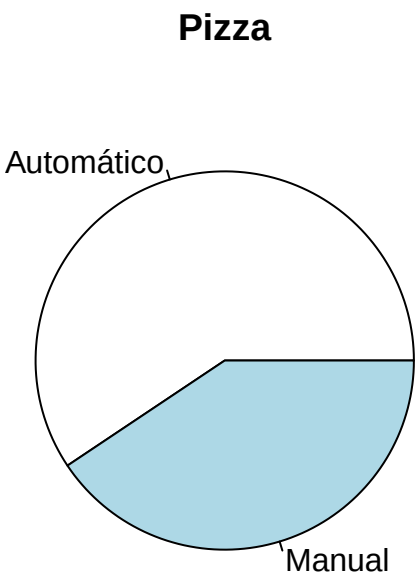


Gráfico de Setores

Outra forma de exibir os dados é pelo gráfico de setores popularmente conhecido como gráfico de pizza.

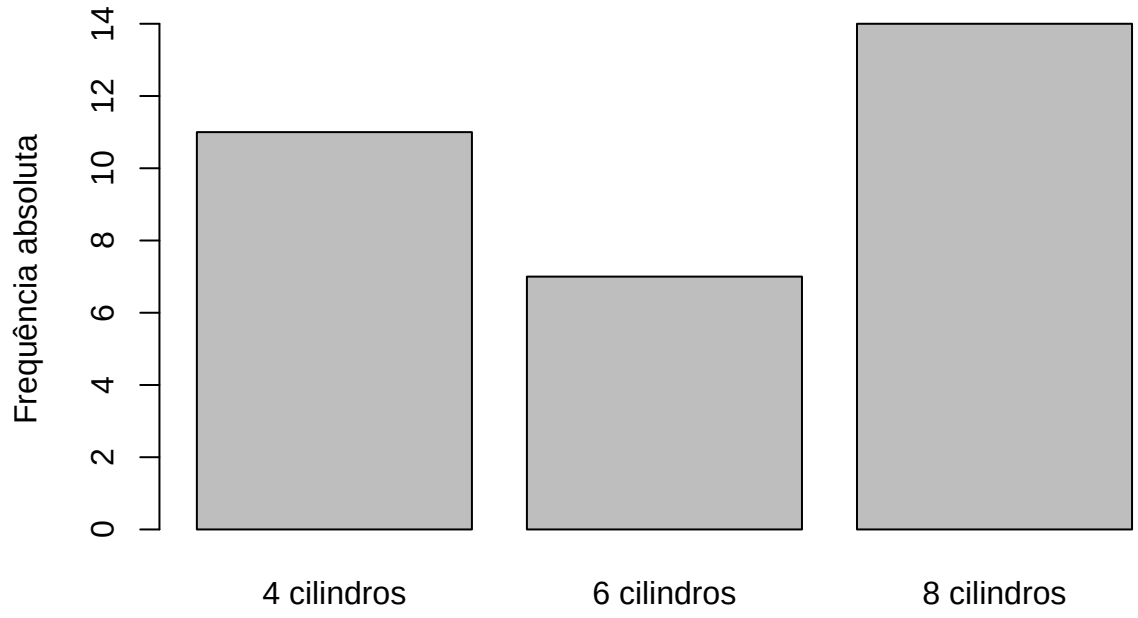
```r
pie(freq, main = "Pizza")
```



Variáveis categóricas ordinais

Para as variáveis categóricas ordinais também podem ser visualizadas pelo gráfico de barras

```r
df$cyl <- factor(mtcars$cyl,
                 levels = c(4,6,8),
                 labels = c("4 cilindros", "6 cilindros", "8 cilindros"),
                 ordered = T)

barplot(table(df$cyl), ylab = "Frequência absoluta" )
```




```r
barplot(prop.table(table(df$cyl)), ylab = "Frequência relativa", ylim = c(0, .6))
```

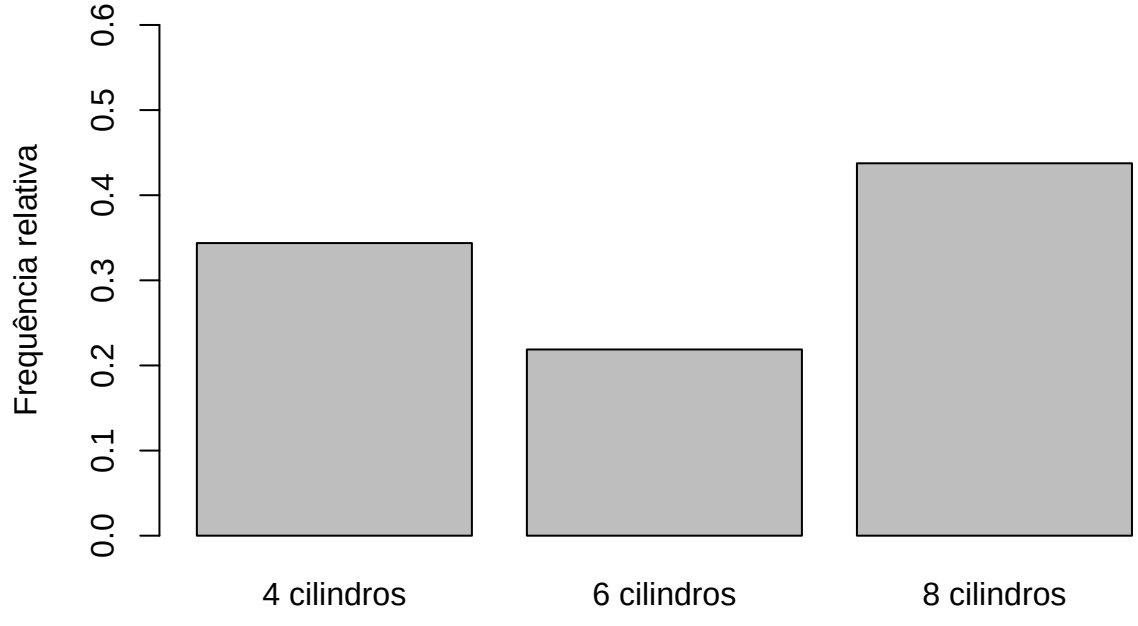


Gráfico de Setores

O gráfico de setores deve-se ser evitado quando há muitas categorias pois é fácil perdermos a referência sobre a dimensões de cada categoria.

```r
pie(table(df$cyl))
```

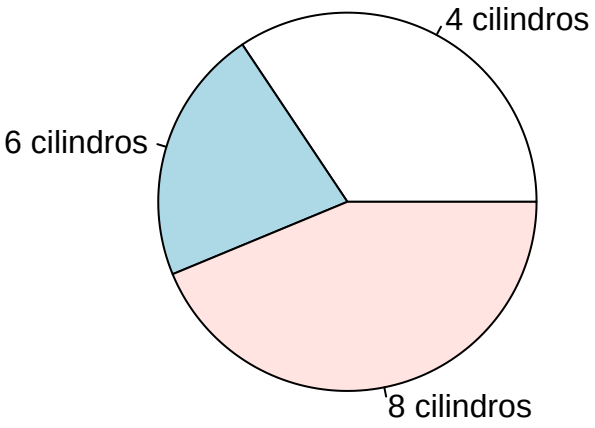


Tabela de frequência

É usual utilizar a tabela de frequencia junto com a frequência relativa

```
tabela <- cbind(f=table(df$cyl))
tabela <- cbind(tabela, "F"=cumsum(tabela[,1]))
tabela <- cbind(tabela, r=prop.table(table(df$cyl)))
tabela <- cbind(tabela, "R"=cumsum(tabela[,3]))
```

	f	F	r	R
4 cilindros	11	11	0.34375	0.34375
6 cilindros	7	18	0.21875	0.56250
8 cilindros	14	32	0.43750	1.00000

Também é usual adicionar os totais nessas tabelas. `addmargins()`.

```
carb.tb <- table(mtcars$carb)
tb <- cbind("f" = addmargins(carb.tb), "p"=addmargins(prop.table(carb.tb)),"F"=c(cumsum(carb.tb),NA), "
```

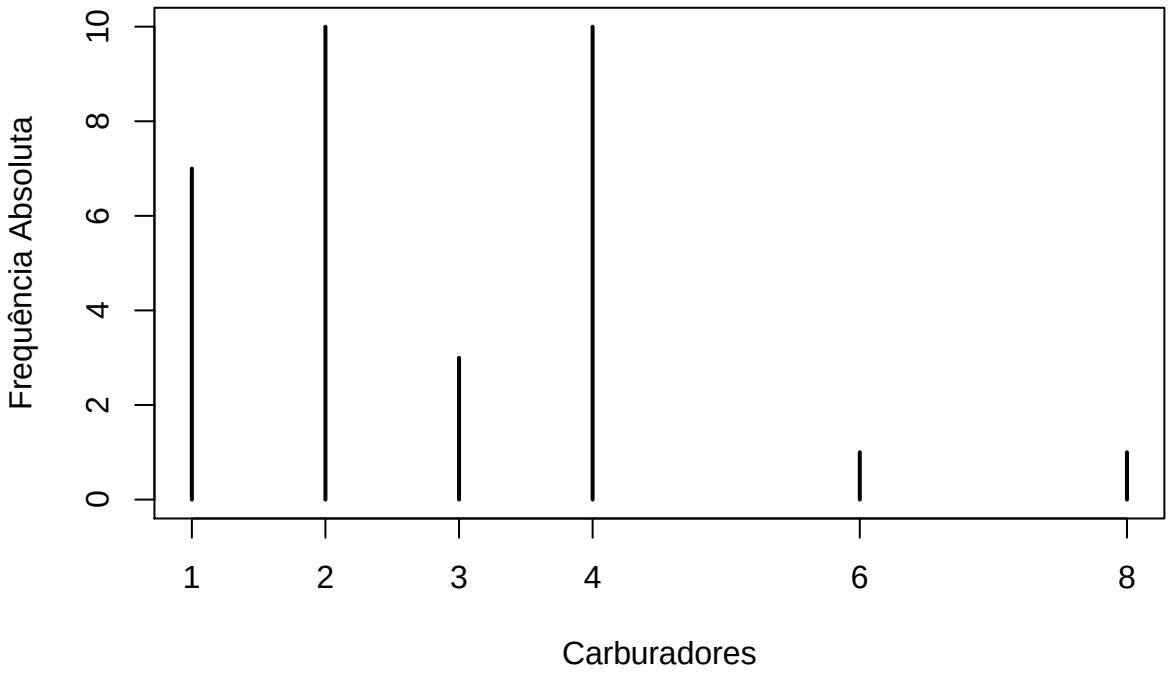
	f	p	F	R
1	7	0.21875	7	0.21875
2	10	0.31250	17	0.53125
3	3	0.09375	20	0.62500
4	10	0.31250	30	0.93750
6	1	0.03125	31	0.96875
8	1	0.03125	32	1.00000
Sum	32	1.00000	NA	NA

Variáveis Numéricas Discretas

Gráfico de Linhas

Para variáveis discretas utiliza-se o gráfico de barras mas as barras são finas. Na verdade linhas apenas nos números pois assim representa-se a não continuidade.

```
plot(carb.tb, ylab = "Frequência Absoluta", xlab="Carburadores")
```

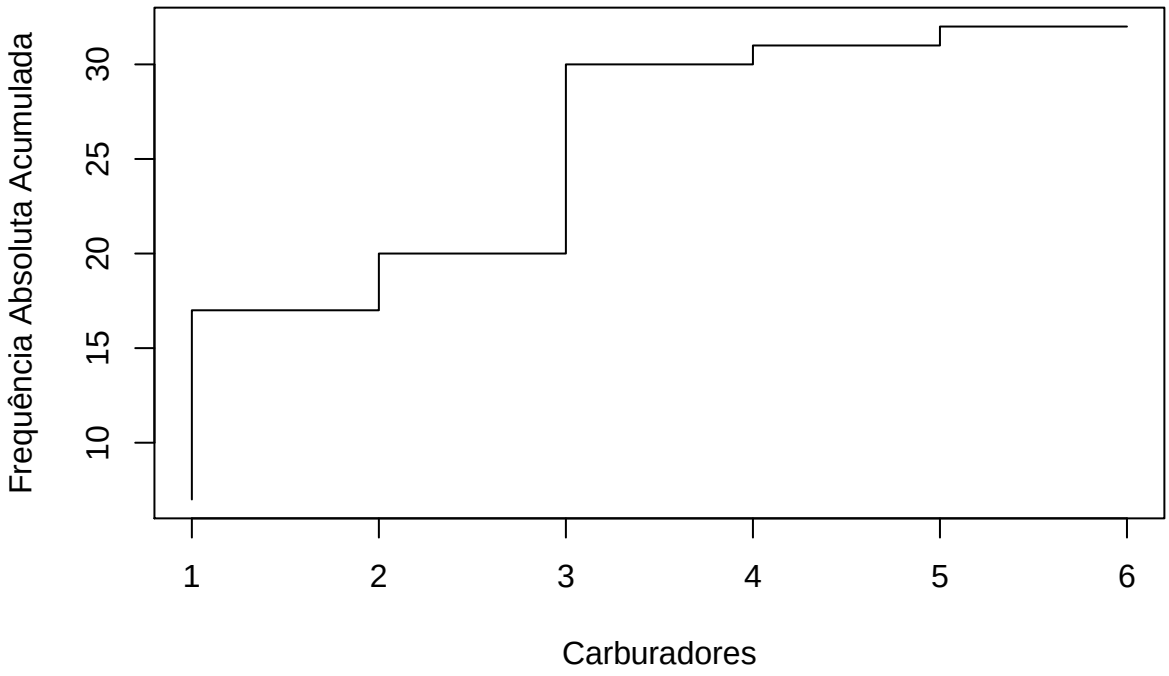


###

Gráfico para frequências acumuladas

Neste caso utiliza-se o gráfico do tipo Step. Basta colotar type = “S”

```
plot(cumsum(carb.tb), type="S", ylab="Frequência Absoluta Acumulada", xlab="Carburadores")
```



Variáveis Numéricas Contínuas

Variáveis contínuas podem ser discretizadas caso seja necessário. Para isso precisamos da Amplitude Total e k. Com isso podemos calcular os pontos de quebra

```
mpg <- sort(mtcars$mpg)
AT <- max(mpg) - min(mpg)
```

```r
k <- sqrt(length(mpg))
AT/k
```

```
## [1] 4.154252
```

```r
(quebra <- seq(10, 36, 4))
```

```
## [1] 10 14 18 22 26 30 34
```

```r
quebra
```

```
## [1] 10 14 18 22 26 30 34
```

Para cortar os dados em intervalos utilizamos a função `cut()` passando os pontos de quebra.

```r
classes <- cut(mpg, breaks = quebra, right = FALSE)
classes.tb <- table(classes)
```

Aqui podemos ver os intervalos já com suas frequências

classes	Freq
[10,14)	3
[14,18)	10
[18,22)	10
[22,26)	3
[26,30)	2
[30,34)	4

E calcular a tabela completa.

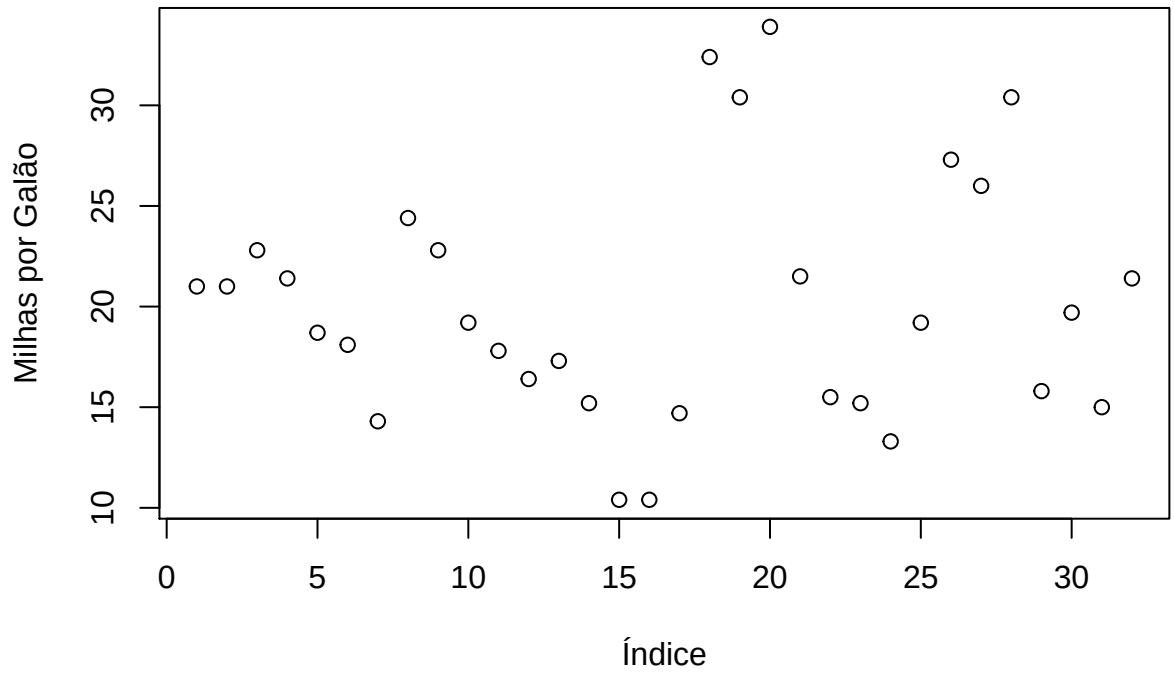
```r
tb <- cbind("f" = addmargins(classes.tb),
            "r"= addmargins(prop.table(classes.tb)),
            "F" = c(cumsum(classes.tb),NA),
            "R" = c(cumsum(prop.table(classes.tb)),NA),
            "Dens"= c(prop.table(classes.tb)/5,NA)
            )
```

	f	r	F	R	Dens
[10,14)	3	0.09375	3	0.09375	0.01875
[14,18)	10	0.31250	13	0.40625	0.06250
[18,22)	10	0.31250	23	0.71875	0.06250
[22,26)	3	0.09375	26	0.81250	0.01875
[26,30)	2	0.06250	28	0.87500	0.01250
[30,34)	4	0.12500	32	1.00000	0.02500
Sum	32	1.00000	NA	NA	NA

Gráfico de disperção

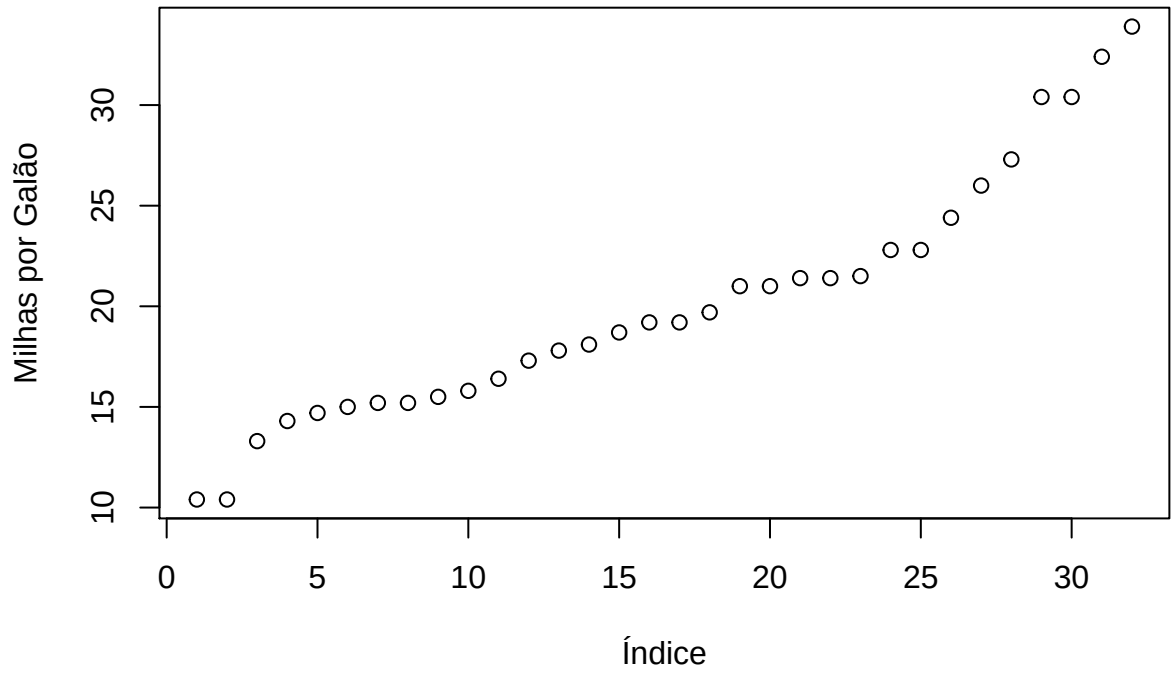
O gráfico de dispersão pode ser utilizado para visualizar uma variável numérica. Pode nos fornecer informações quanto à sua dispersão. O valor da variável fica no eixo y enquanto no x fica somente o índice.

```
plot(mtcars$mpg, ylab="Milhas por Galão", xlab="Índice")
```



Se necessário também será útil ordenar pelos valores.

```
plot(sort(mtcars$mpg), ylab="Milhas por Galão", xlab="Índice")
```

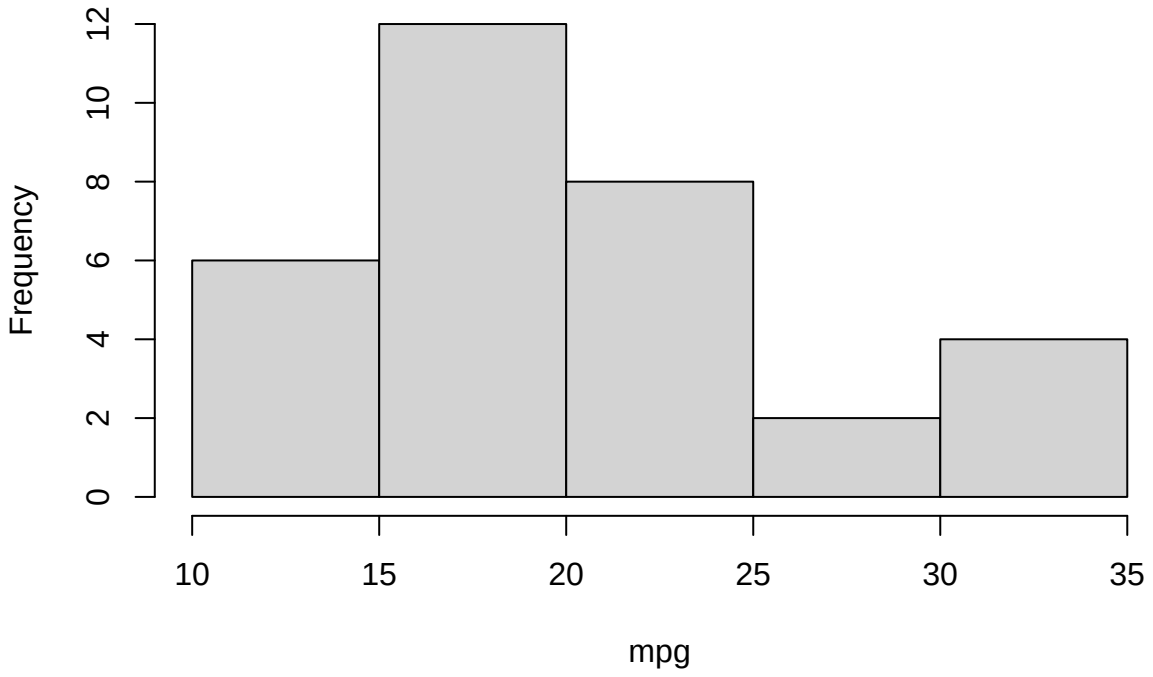


##

Histograma

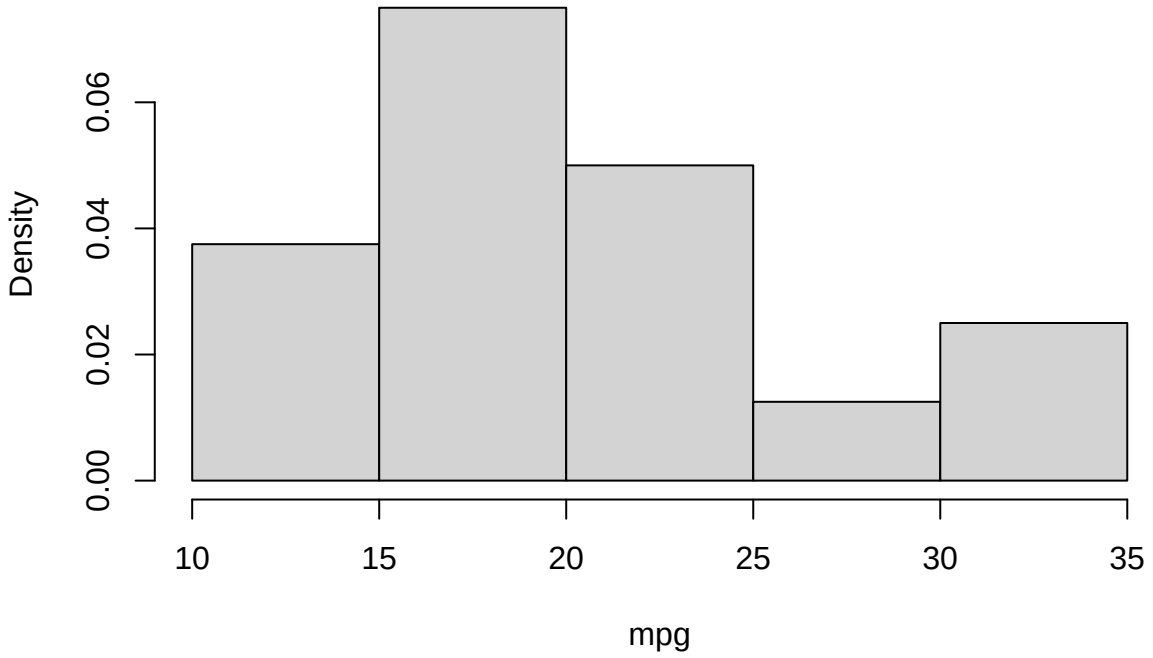
Esta ferramenta é útil para visualizar a distribuição dos dados. Ele agrupa os valores por intervalos para poder contar.

```
hist(mpg, main = "")
```



Aqui temos o mesmo histograma mas em vez de utilizarmos frequência, utilizamos densidade.

```
hist(mpg, freq = F, main = "")
```



Medidadas de Centralidade

Mediana

```
median(mpg)
```

```
## [1] 19.2
```

Quando os dados possuem valores NA a mediana não é calculada

```r
median(airquality$Ozone)
```

```
## [1] NA
```

Para forçar o cálculo temos mo argumento na.rm = T

```r
median(airquality$Ozone, na.rm = T)
```

```
## [1] 31.5
```

Média

```r
mean(mpg)
```

```
## [1] 20.09062
```

Quando os dados possuem valores NA a média não é calculada

```r
mean(airquality$Ozone)
```

```
## [1] NA
```

Para forçar o cálculo temos mo argumento na.rm = T

```r
mean(airquality$Ozone, na.rm = T)
```

```
## [1] 42.12931
```

Medidas de dispersão

Amplitude

Amplitude é a diferença entre o maximo e mínimo

```r
AMP <- max(airquality$Temp) - min(airquality$Temp)
```

Desvio Médio

$$DM = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

```r
sum(abs(airquality$Temp - mean(airquality$Temp)))/ length(airquality$Temp)
```

```
## [1] 7.568627
```

Variância

```r
var(mpg)
```

```
## [1] 36.3241
```

Desvio Padrão

```r
sd(mpg)
```

```
## [1] 6.026948
```

Coeficiente de Variação

Com o coeficiente de variação podemos comparar a variação de diferentes atributos.

$$CV = \frac{s}{\bar{x}}.100\%$$

```
cbind(
  mpg=(sd(mtcars$mpg)/mean(mtcars$mpg))*100,
  wt=(sd(mtcars$wt)/mean(mtcars$wt))*100)
```

```
##           mpg       wt
## [1,] 29.99881 30.41285
```

Quartil

O quartil sera os dados em 4 conjuntos iguais ou muito próximos. O segundo quartil possui 50% dos dados para cada lado. É a mediana.

```
quantile(mpg)
```

```
##     0%    25%    50%    75%   100%
## 10.400 15.425 19.200 22.800 33.900
```

Podemos utilizar a função `cut()` e `table()` para calcular as frequências de cada classe.

```
cbind(freq=table(cut(mpg, breaks = quantile(mpg), right = F)))
```

```
##             freq
## [10.4,15.4)    8
## [15.4,19.2)    7
## [19.2,22.8)    8
## [22.8,33.9)    8
```

Decil

O quartil sera os dados em 10 conjuntos iguais ou muito próximos.

```
quantile(mpg, probs = seq(0,1,0.1))
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
## 10.40 14.34 15.20 15.98 17.92 19.20 21.00 21.47 24.08 30.09 33.90
```

Percentil

O quartil sera os dados em 100 conjuntos iguais ou muito próximos.

```
quantile(mpg, probs = seq(0,1,0.01))
```

```
##     0%     1%     2%     3%     4%     5%     6%     7%     8%     9%    10%
## 10.400 10.400 10.400 10.400 11.096 11.995 12.894 13.470 13.780 14.090 14.340
##    11%    12%    13%    14%    15%    16%    17%    18%    19%    20%    21%
## 14.464 14.588 14.709 14.802 14.895 14.988 15.054 15.116 15.178 15.200 15.200
##    22%    23%    24%    25%    26%    27%    28%    29%    30%    31%    32%
## 15.200 15.239 15.332 15.425 15.518 15.611 15.704 15.797 15.980 16.166 16.352
##    33%    34%    35%    36%    37%    38%    39%    40%    41%    42%    43%
## 16.607 16.886 17.165 17.380 17.535 17.690 17.827 17.920 18.013 18.112 18.298
##    44%    45%    46%    47%    48%    49%    50%    51%    52%    53%    54%
## 18.484 18.670 18.830 18.985 19.140 19.200 19.200 19.200 19.260 19.415 19.570
```

```
##    55%    56%    57%    58%    59%    60%    61%    62%    63%    64%    65%
## 19.765 20.168 20.571 20.974 21.000 21.000 21.000 21.088 21.212 21.336 21.400
##    66%    67%    68%    69%    70%    71%    72%    73%    74%    75%    76%
## 21.400 21.400 21.408 21.439 21.470 21.513 21.916 22.319 22.722 22.800 22.800
##    77%    78%    79%    80%    81%    82%    83%    84%    85%    86%    87%
## 22.800 23.088 23.584 24.080 24.576 25.072 25.568 26.052 26.455 26.858 27.261
##    88%    89%    90%    91%    92%    93%    94%    95%    96%    97%    98%
## 28.168 29.129 30.090 30.400 30.400 30.400 30.680 31.300 31.920 32.505 32.970
##    99%   100%
## 33.435 33.900
```

As cinco Medidas

Medidas muito importantes para resumir os dados temos a função `fivenum()`

```
fivenum(mpg)
```

```
## [1] 10.40 15.35 19.20 22.80 33.90
```

ou `summary()`

```
summary(mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.40   15.43   19.20   20.09   22.80   33.90
```

Aqui vamos gerar as cinco medidas para o conjunto de dados que analisa o número de insetos mortos por cada tipo de inseticida.

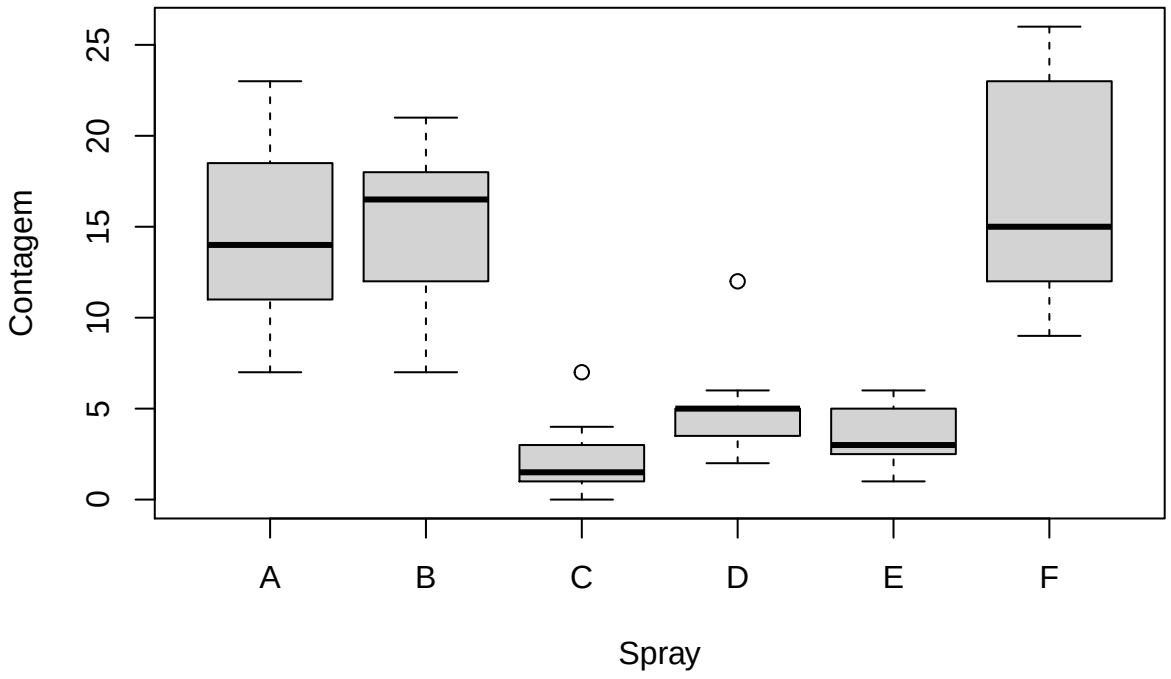
```
fn <- rbind(
A=fivenum(InsectSprays[InsectSprays[,2]=="A",1]),
B=fivenum(InsectSprays[InsectSprays[,2]=="B",1]),
C=fivenum(InsectSprays[InsectSprays[,2]=="C",1]),
D=fivenum(InsectSprays[InsectSprays[,2]=="D",1]),
E=fivenum(InsectSprays[InsectSprays[,2]=="E",1]),
F=fivenum(InsectSprays[InsectSprays[,2]=="F",1]))

colnames(fn) <- c("Min","Q1","Q2","Q3","Max")
```

	Min	Q1	Q2	Q3	Max
A	7	11.0	14.0	18.5	23
B	7	12.0	16.5	18.0	21
C	0	1.0	1.5	3.0	7
D	2	3.5	5.0	5.0	12
E	1	2.5	3.0	5.0	6
F	9	12.0	15.0	23.0	26

Podemos ver que apesar de termos as informações calculada fica difícil a visualização das informações. ### Box plot

Para uma melhor visualizar as mesmas informações de uma maneira muito mais rápida temos o diagrama de caixas

```
boxplot(count ~ spray, data = InsectSprays,
        xlab = "Spray", ylab = "Contagem",
          col = "lightgray")
```

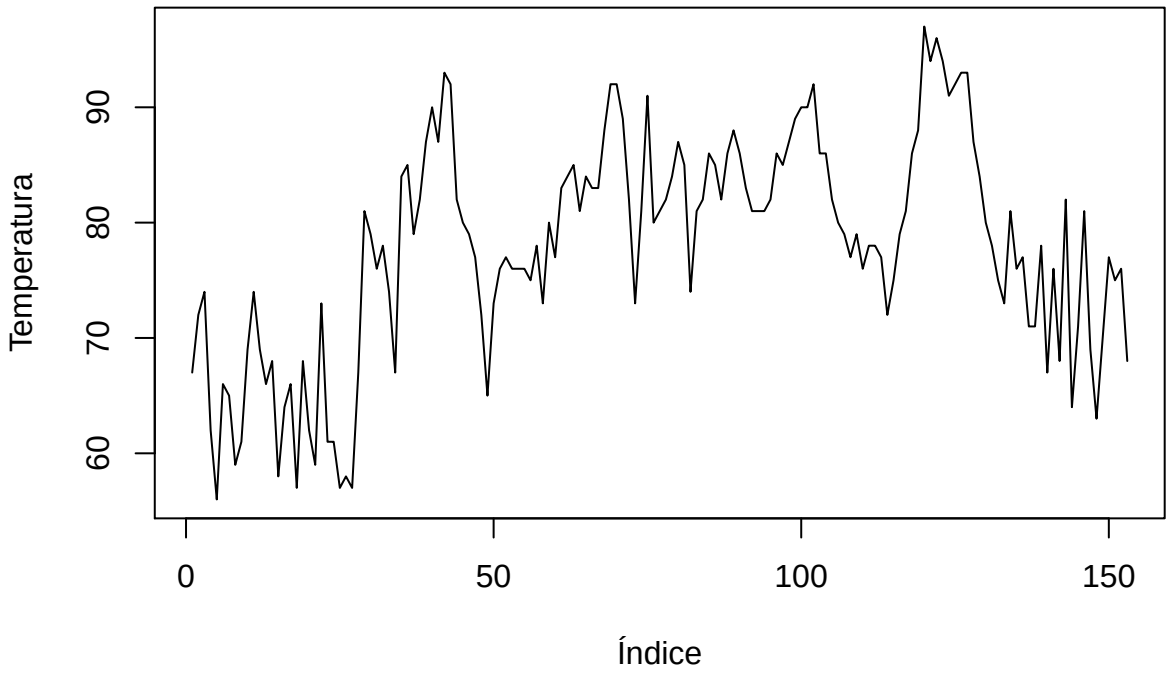


##

Gráfico de linhas

Outro gráfico bastante utilizado para dados contínuos é o de linhas

```
plot(airquality$Temp, type="l", ylab = "Temperatura", xlab="Índice")
```



##

Extra

Vocabulário de Gráficos

A parte de visualização de gráficos é muito importante da estatística para comunicar a informação.

Para ajudar na escolha temos o guia a seguir Vocabulário de gráficos

Biblioteca de gráficos

O ggplot2 é um pacote integrante do tidyverse muito eficaz na contrução de gráficos. Ela trabalha de uma forma um pouco diferente do modo normal e utiliza a chamada grammar of graphics.

Segue o link para consulta ggplot2

Gráficos no R

Aqui segue um guia com inúmeros gráficos a serem utilizados e seu guia. Além desses gráficos existem outras bibliotecas para gráficos.

Segue o guia Tipos de graficos

Toolbox

Existem vários pacotes de EDA

verifique o DataExplorer

install.packages("DataExplorer") library(DataExplorer) create_report(df)