Exemplos Ilustrativos do uso do R

Luís Torgo
FEP, Universidade do Porto
ltorgo@liacc.up.pt

12 de Dezembro de 2006

Esta pequena introdução tem como objectivo principal apresentar alguns exemplos ilustrativos de utilização do R em tarefas de análise de dados, sem que nesta altura se espere que os alunos entendam todo o código R usado.

1. O famoso conjunto de dados "iris"

Começamos por carregar o conjunto de dados para o R e depois ver as primeiras linhas desta tabela de dados (de duas formas alternativas).

```
> data(iris)
> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> str(iris)

'data.frame':        150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Vejamos agora alguns exemplos de obtenção de estatísticas descritivas básicas dos dados,

```
> summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

Uma outra alternativa usando uma função disponível na package extra chamada Hmisc,

```
> library(Hmisc)
> describe(iris)

iris

 5  Variables      150  Observations
---------------------------------------------------------------------------
Sepal.Length
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
    150       0      35   5.843   4.600   4.800   5.100   5.800   6.400   6.900   7.255

lowest : 4.3 4.4 4.5 4.6 4.7, highest: 7.3 7.4 7.6 7.7 7.9
---------------------------------------------------------------------------
Sepal.Width
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
    150       0      23   3.057   2.345   2.500   2.800   3.000   3.300   3.610   3.800

lowest : 2.0 2.2 2.3 2.4 2.5, highest: 3.9 4.0 4.1 4.2 4.4
---------------------------------------------------------------------------
Petal.Length
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
    150       0      43   3.758    1.30    1.40    1.60    4.35    5.10    5.80    6.10

lowest : 1.0 1.1 1.2 1.3 1.4, highest: 6.3 6.4 6.6 6.7 6.9
---------------------------------------------------------------------------
Petal.Width
      n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
    150       0      22   1.199     0.2     0.2     0.3     1.3     1.8     2.2     2.3

lowest : 0.1 0.2 0.3 0.4 0.5, highest: 2.1 2.2 2.3 2.4 2.5
---------------------------------------------------------------------------
Species
      n missing  unique
    150       0       3

setosa (50, 33%), versicolor (50, 33%), virginica (50, 33%)
---------------------------------------------------------------------------
```

Vamos agora aplicar uma função às colunas de um conjunto de dados (data frame no dialecto R). A função "apply" permite-nos este efeito. No exemplo, aplicamos a função "sd" a todas menos a 5ª coluna da tabela, obtendo desta forma os desvios padrões de cada coluna

```
> apply(iris[, -5], 2, sd)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.8280661    0.4358663    1.7652982    0.7622377
```

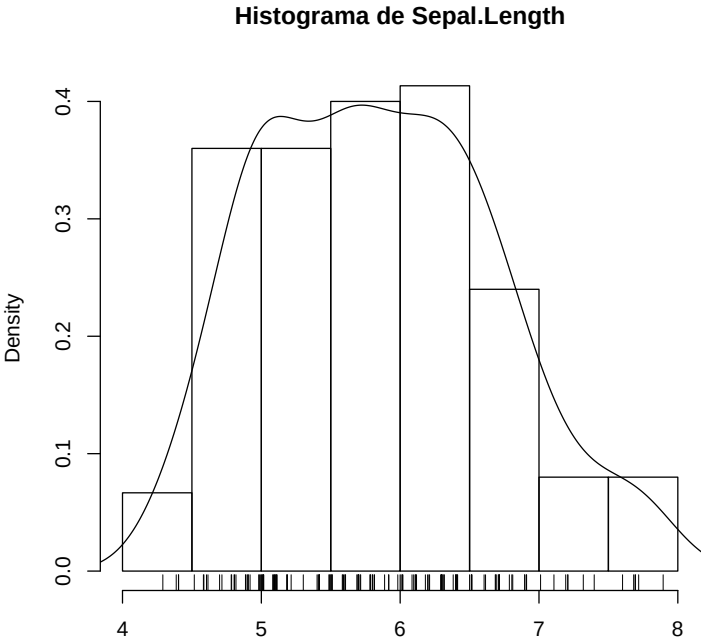
A função "by" permite-nos aplicar "qualquer" função a sub-grupos dos nossos dados que são determinados por um factor (uma variável discreta dos nossos dados),

```
> by(iris[, -5], iris$Species, mean)

iris$Species: setosa
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.006        3.428        1.462        0.246
------------------------------------------------------------
iris$Species: versicolor
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.936        2.770        4.260        1.326
------------------------------------------------------------
iris$Species: virginica
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       6.588        2.974        5.552        2.026
```

1.1. Exemplos de Visualização de Dados

1.1.1. Histogramas

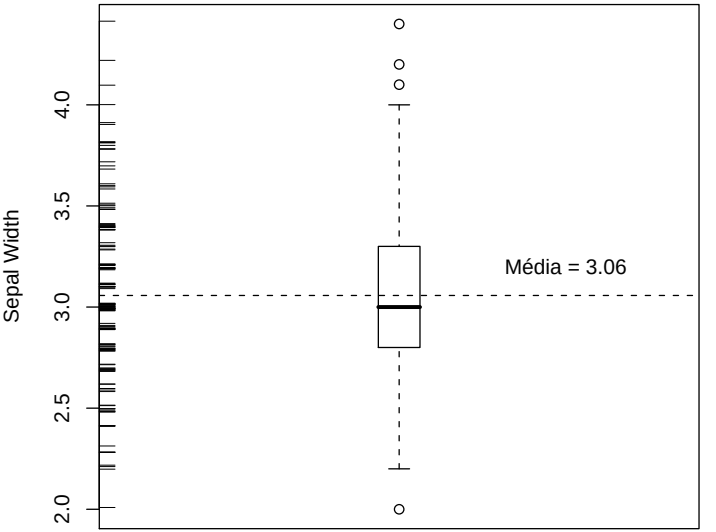
```
> hist(iris$Sepal.Length, prob = T, xlab = "", main = "Histograma de Sepal.Length")
> lines(density(iris$Sepal.Length, na.rm = T))
> rug(jitter(iris$Sepal.Length))
```



X11 2

1.1.2. "Boxplots"

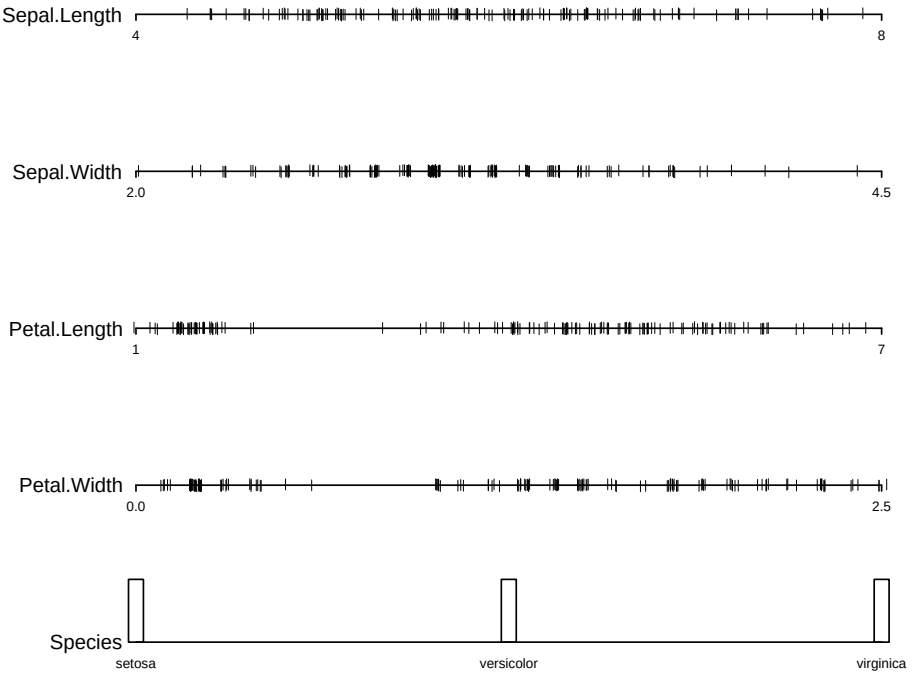
```
> boxplot(iris$Sepal.Width, boxwex = 0.15, ylab = "Sepal Width")
> rug(jitter(iris$Sepal.Width), side = 2)
> med <- mean(iris$Sepal.Width, na.rm = T)
> abline(h = med, lty = 2)
> text(1.3, 3.2, paste("Média =", round(med, 2)))
```



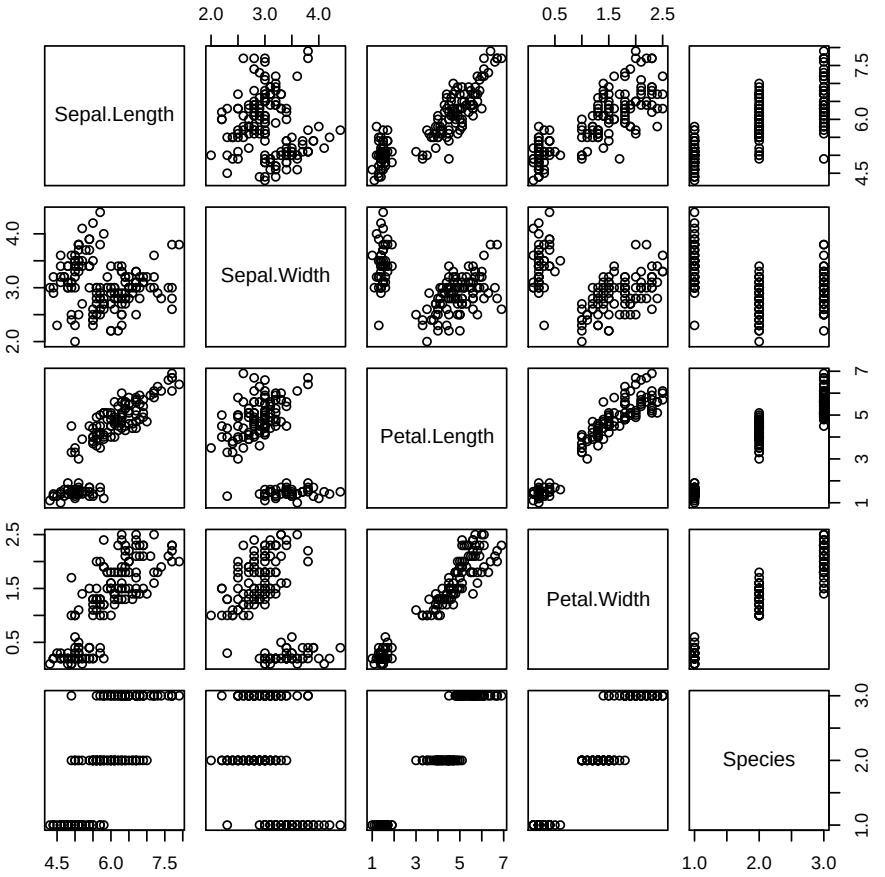
X11 2

1.1.3. Global Overviews of the Data

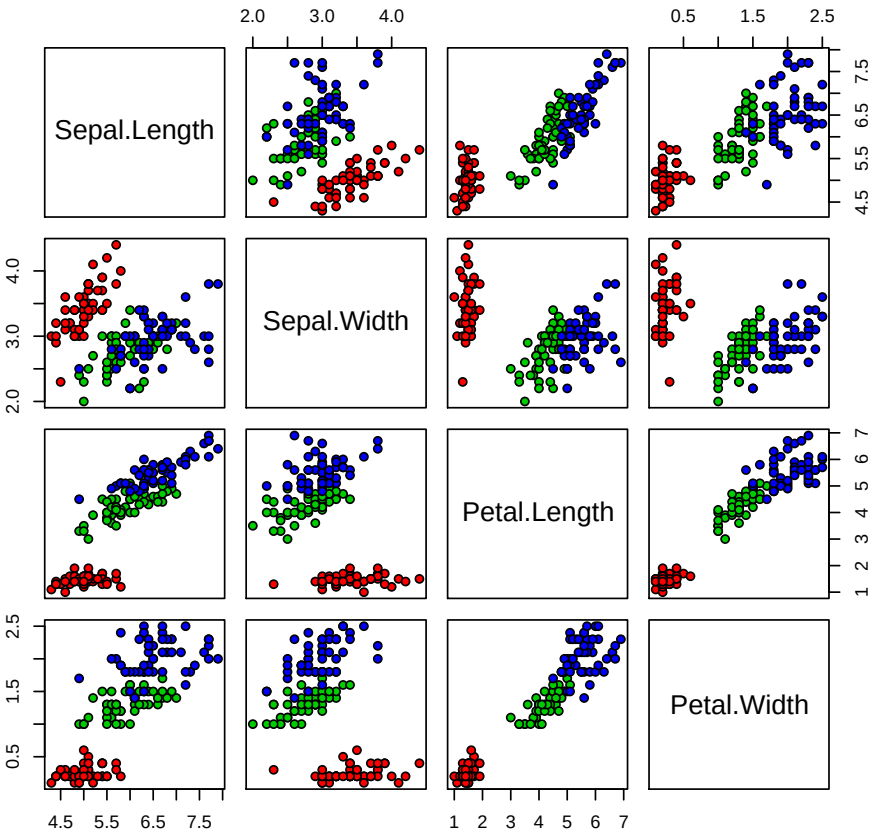
```
> library(Hmisc)
> datadensity(iris)
```



```
> plot(iris)
```

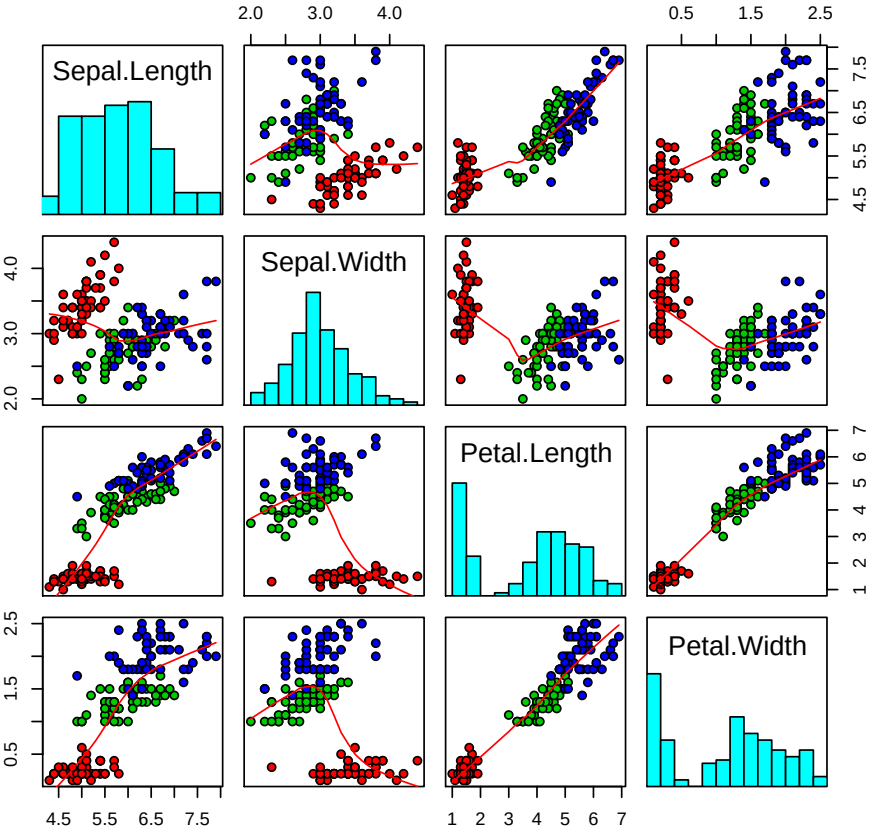

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
2.0 3.0 4.0
0.5 1.5 2.5
4.5 6.0 7.5
2.0 3.0 4.0
1 3 5 7
0.5 1.5 2.5
4.5 6.0 7.5
1 3 5 7
1.0 2.0 3.0

```
> pairs(iris[1:4], main = "Iris -- 3 espécies", pch = 21, bg = c("red",
+       "green3", "blue")[iris$Species])
```



```
> panel.hist <- function(x, ...) {
+     usr <- par("usr")
+     on.exit(par(usr))
+     par(usr = c(usr[1:2], 0, 1.5))
+     h <- hist(x, plot = FALSE, cex.main = 0.7)
+     breaks <- h$breaks
+     nB <- length(breaks)
+     y <- h$counts
+     y <- y/max(y)
+     rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
+ }
> pairs(iris[1:4], panel = panel.smooth, main = "Iris -- 3 espécies",
+     pch = 21, bg = c("red", "green3", "blue")[iris$Species],
+     diag.panel = panel.hist, cex.labels = 1.5)
```

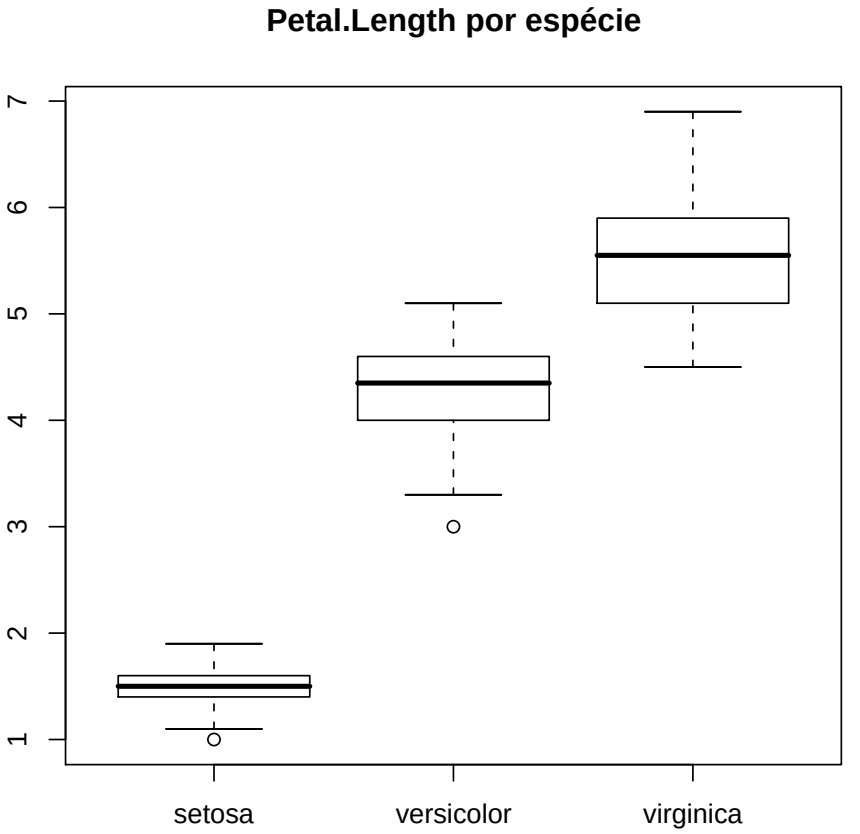
Iris –– 3 espécies



1.1.4. Gráficos Condicionados

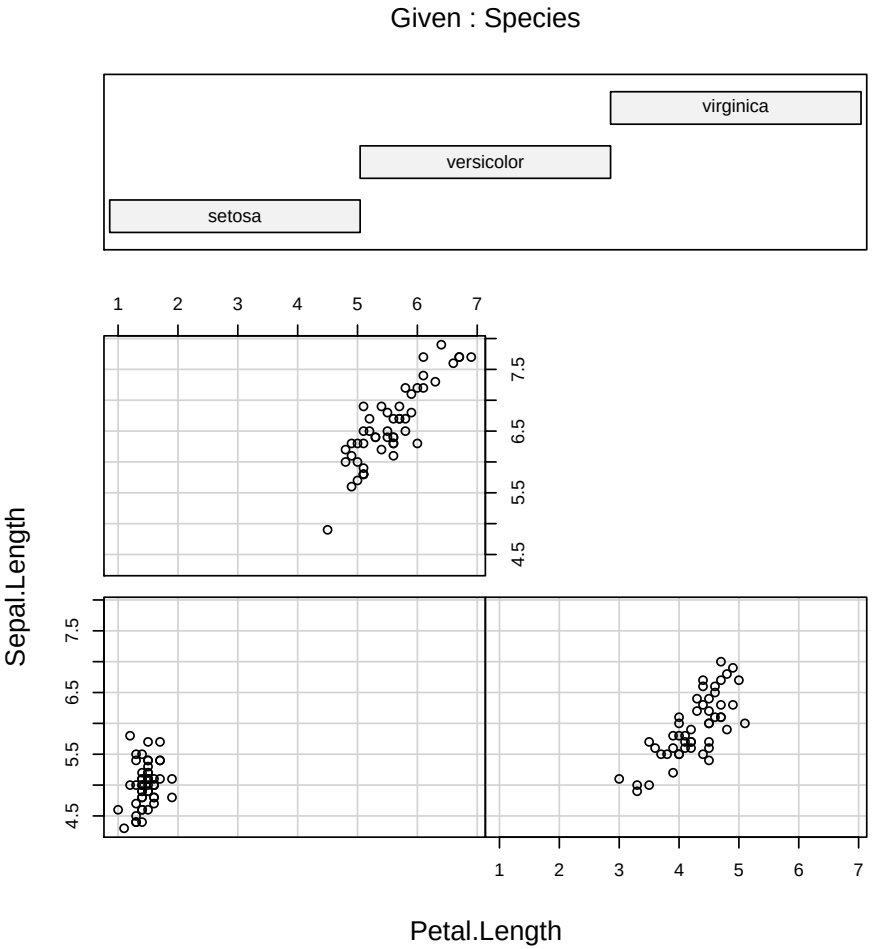
"Box Plots" Condicionados

```
> boxplot(Petal.Length ~ Species, iris, main = "Petal.Length por espécie")
```



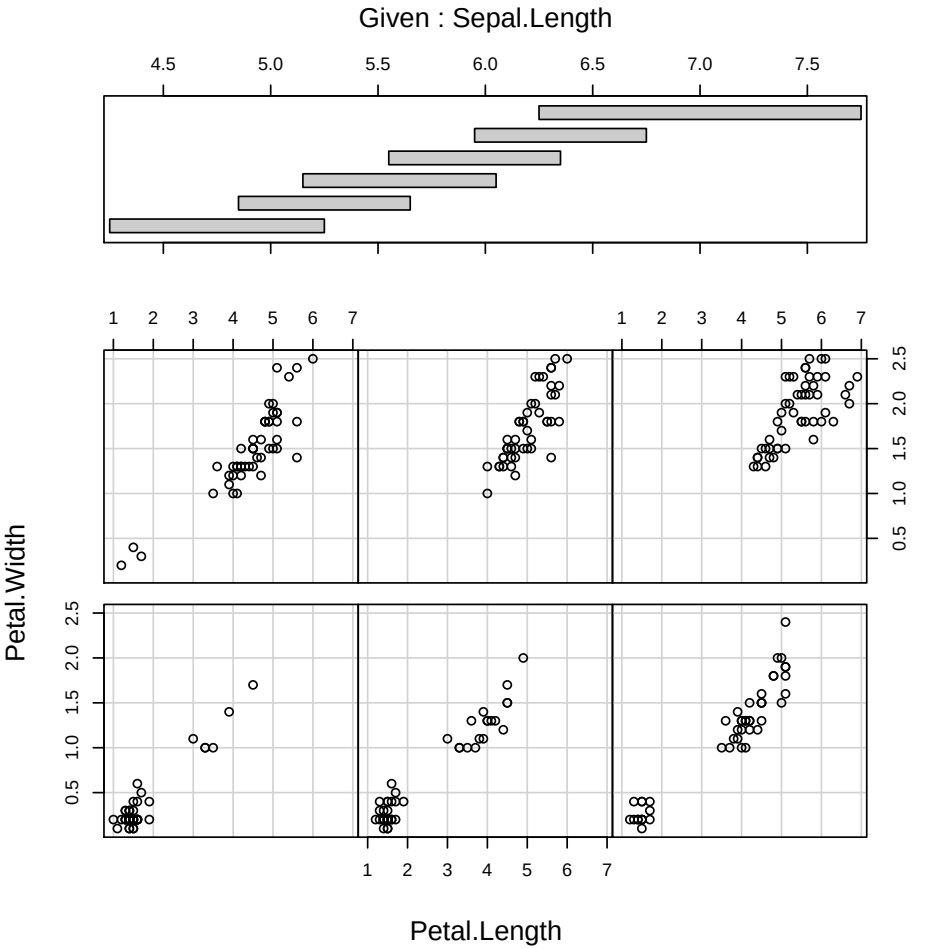
Gráficos X-Y Condicionados

```
> coplot(Sepal.Length ~ Petal.Length | Species, data = iris)
```



Gráficos X-Y Condicionados (por uma variável numérica)

```
> coplot(Petal.Width ~ Petal.Length | Sepal.Length, data = iris)
```



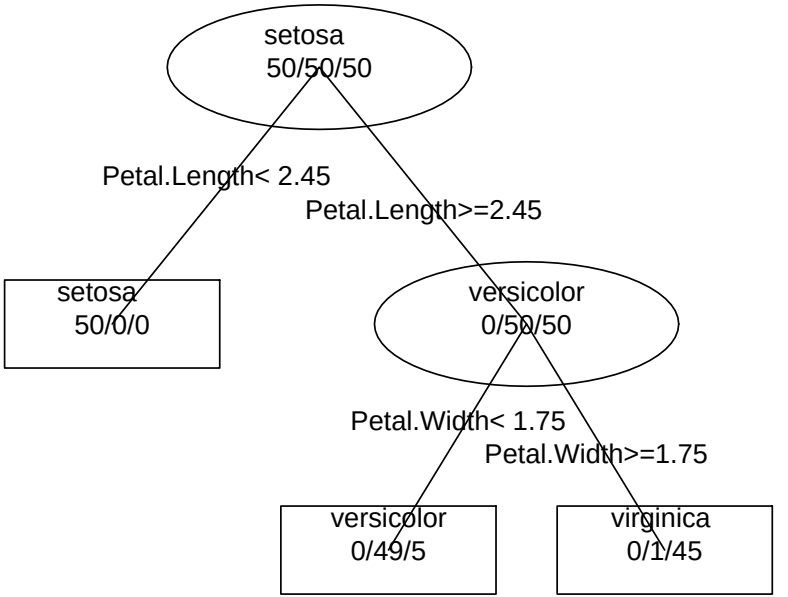
1.2. Um Exemplo de um Modelo dos Dados

As árvores de classificação são um exemplo de um modelo que pode ser obtido com este tipo de dados.

```
> library(rpart)
> ac <- rpart(Species ~ ., iris)
> ac

n= 150

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
  2) Petal.Length< 2.45 50   0 setosa (1.00000000 0.00000000 0.00000000) *
  3) Petal.Length>=2.45 100  50 versicolor (0.00000000 0.50000000 0.50000000)
    6) Petal.Width< 1.75 54   5 versicolor (0.00000000 0.90740741 0.09259259) *
    7) Petal.Width>=1.75 46   1 virginica (0.00000000 0.02173913 0.97826087) *
```

Podemos obter uma representação gráfica da árvore...

```
> plot(ac, margin = 0.15, branch = 0)
> text(ac, use.n = T, fancy = T, all = T, fheight = 0.9)
```



2. Dados do Censos de 1977 nos USA

```
> data(state)
> summary(state.x77)
```

```
   Population        Income       Illiteracy       Life Exp
 Min.   :  365   Min.   :3098   Min.   :0.500   Min.   :67.96
 1st Qu.: 1080   1st Qu.:3993   1st Qu.:0.625   1st Qu.:70.12
 Median : 2838   Median :4519   Median :0.950   Median :70.67
 Mean   : 4246   Mean   :4436   Mean   :1.170   Mean   :70.88
 3rd Qu.: 4968   3rd Qu.:4814   3rd Qu.:1.575   3rd Qu.:71.89
 Max.   :21198   Max.   :6315   Max.   :2.800   Max.   :73.60
     Murder          HS Grad          Frost             Area
 Min.   : 1.400   Min.   :37.80   Min.   :  0.00   Min.   :  1049
 1st Qu.: 4.350   1st Qu.:48.05   1st Qu.: 66.25   1st Qu.: 36985
 Median : 6.850   Median :53.25   Median :114.50   Median : 54277
 Mean   : 7.378   Mean   :53.11   Mean   :104.46   Mean   : 70736
 3rd Qu.:10.675   3rd Qu.:59.15   3rd Qu.:139.75   3rd Qu.: 81162
 Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432
```

Os 5 estados menos populosos...

```
> state.name[order(state.x77[, "Population"])[1:5]]

[1] "Alaska"   "Wyoming"  "Vermont"  "Delaware" "Nevada"
```

E os 5 mais...

```
> state.name[order(state.x77[, "Population"], decreasing = T)[1:5]]

[1] "California"   "New York"     "Texas"        "Pennsylvania" "Illinois"
```

Os estados com uma esperança de vida acima da média de todos os estados

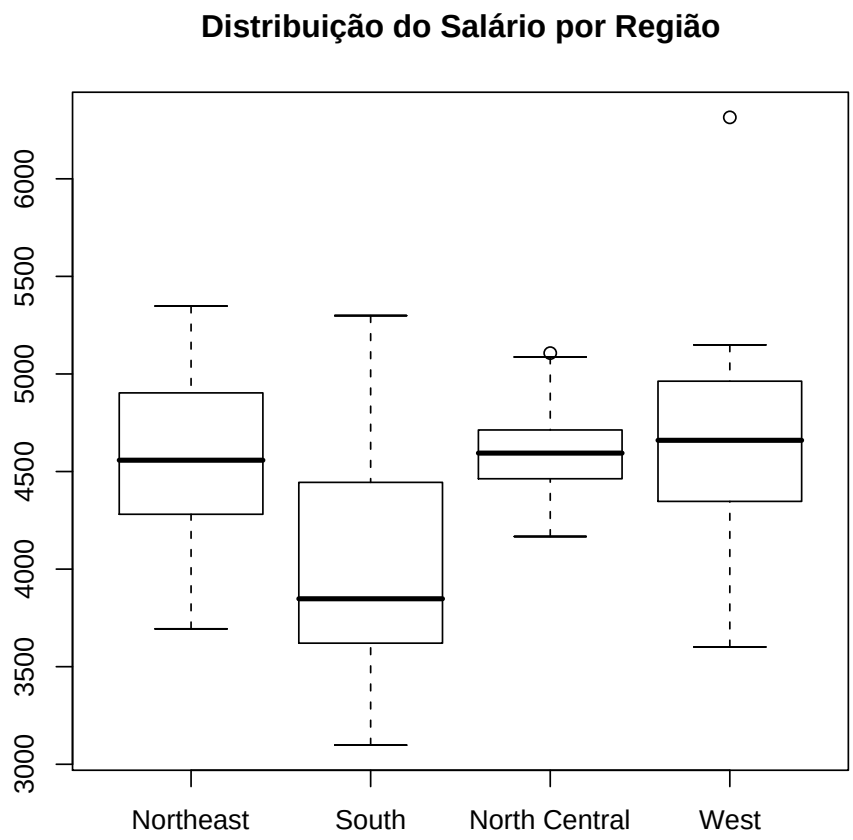
```
> state.name[state.x77[, "Life Exp"] > mean(state.x77[, "Life Exp"])]

 [1] "California"    "Colorado"      "Connecticut"   "Hawaii"
 [5] "Idaho"         "Indiana"       "Iowa"          "Kansas"
 [9] "Massachusetts" "Minnesota"     "Nebraska"      "New Hampshire"
[13] "New Jersey"    "North Dakota"  "Oklahoma"      "Oregon"
[17] "Rhode Island"  "South Dakota"  "Texas"         "Utah"
[21] "Vermont"       "Washington"    "Wisconsin"
```

Qual a distribuição do salário per capita nas diferentes regiões?

```
> by(state.x77[, "Income"], state.region, summary)

INDICES: Northeast
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3694    4281    4558    4570    4903    5348
------------------------------------------------------------
INDICES: South
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3098    3622    3848    4012    4316    5299
------------------------------------------------------------
INDICES: North Central
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4167    4466    4594    4611    4694    5107
------------------------------------------------------------
INDICES: West
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3601    4347    4660    4703    4963    6315
```

O mesmo visualmente...

```
> boxplot(state.x77[, "Income"] ~ state.region, main = "Distribuição do Salário por Região")
```

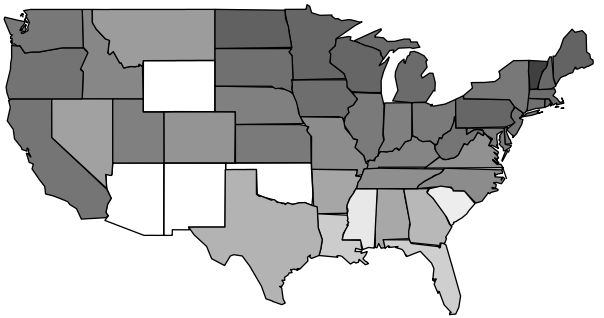


Qual o valor da estatística R^2 quando tentamos obter um modelo de regressão entre a variável "Income" e todas as outras variáveis medidas no Censos?

```
> dados <- as.data.frame(state.x77)
> colnames(dados)[c(4, 6)] <- c("LifeExp", "HSgrad")
> ind <- setdiff(colnames(dados), "Income")
> r2s <- rep(NA, length(ind))
> names(r2s) <- ind
> for (i in ind) {
+     m <- lm(as.formula(paste("Income ~", i)), dados)
+     r2s[i] <- summary(m)$r.squared
+ }
> round(r2s, 2)

Population Illiteracy    LifeExp     Murder     HSgrad      Frost       Area
      0.04       0.19       0.12       0.05       0.38       0.05       0.13
```

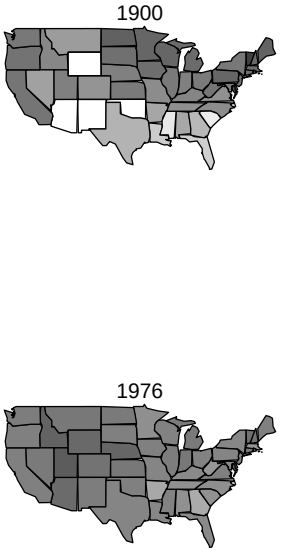
Um exemplo envolvendo distribuições geográficas: a percentagem de votantes no partido Republicano em 1900.

```
> library(maps)
> data(votes.repub)
> state.to.map <- match.map("state", state.name)
> x <- votes.repub[state.to.map, "1900"]
> gray.colors <- function(n) gray(rev(0:(n - 1))/n)
> color <- gray.colors(100)[floor(x)]
> map("state", fill = TRUE, col = color)
```



E a evolução ao longo do tempo...

```
> par(mfrow = c(2, 1))
> x00 <- votes.repub[state.to.map, "1900"]
> x76 <- votes.repub[state.to.map, "1976"]
> color00 <- gray.colors(100)[floor(x00)]
> color76 <- gray.colors(100)[floor(x76)]
> map("state", fill = TRUE, col = color00)
> mtext("1900", side = 3)
> map("state", fill = TRUE, col = color76)
> mtext("1976", side = 3)
```



3. Inquérito aos Empregados de uma Empresa

Uma empresa fez um inquérito aos seus empregados. O inquérito possui dezenas de perguntas que visam saber o que pensam os empregados sobre uma série de assuntos relacionados com a vida da empresa. Para cada inquérito (empregado) existe ainda informação contextual sobre uma série de características (a sua posição na empresa, sexo, grupo ectário, etc.). Pretende-se, entre outras coisas saber se as respostas às perguntas está de alguma forma correlacionada com esses factores contextuais.

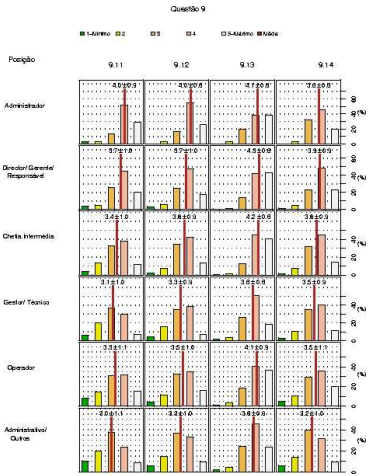
Algumas das dificuldades principais:

- Muitas perguntas (muitas combinações possíveis!)
- Como representar as respostas de forma informativa mas sucinta?
- Tempo para produzir um relatório exaustivo
- etc.

Algumas das soluções encontradas (possíveis graças ao R!):

- Desenvolvido um tipo de gráficos específico para este cenário
- Usada função "Sweave" da package "utils" para produzir relatórios de forma "automática". Cerca de 1000 páginas de relatórios (figuras, tabelas, etc.) foram produzidos em minutos e de forma automática usando um programa em R.

Um dos gráficos desenvolvidos...



O tipo de código usado para obter os relatórios...

```
...
for(q in c(7:33,35,37:42)) {
  barras.cond.descr(q,'Posição',save=T)
  cat('\\includegraphics{Figuras/b',q,'Pos.eps}\n\n',sep="")
  cat('\\clearpage \n')
}
...
```

4. Séries Temporais

Um dos principais objectivos da análise de séries temporais é obter um modelo baseado em observações passadas de uma variável, $y_1, y_2, \ldots, y_{t-1}, y_t$, que nos permite fazer previsões sobre os valores futuros da série, $y_{t+1}, \ldots, y_n$.

Vamos mostrar uma pequena ilustração disto com dados sobre preços de acções, nomeadamente as suas cotações diárias: valor de abertura, de fecho e máximo e mínimo durante o dia.

O R tem várias packages dedicadas à análise de séries temporais.

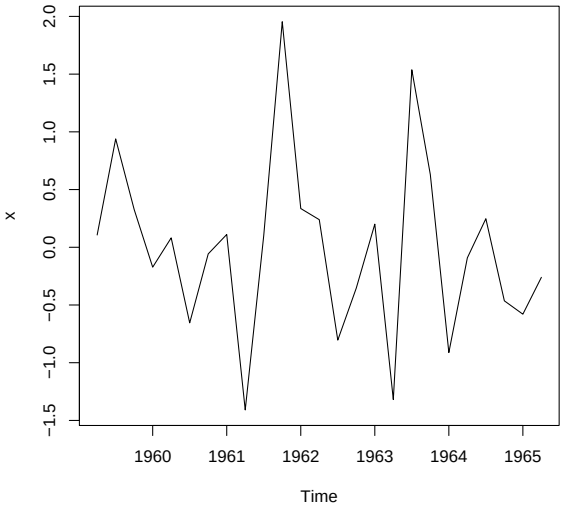
Podemos distinguir dois tipos principais de séries temporais: regulares e irregulares.

4.1. Séries Regulares

Valores igualmente espaçados (em termos temporais).

```
> ts(rnorm(15), frequency = 4, start = c(1959, 2))

           Qtr1       Qtr2       Qtr3       Qtr4
1959             0.9400534  0.4836605  0.1295230
1960  0.6921757  0.2944689  0.7074430  0.2663582
1961  0.6421597 -0.9735027 -0.3859516  0.3750973
1962 -1.3026701  1.3963150  0.2848089 -0.5794639

> ts(rnorm(15), frequency = 12, start = c(1959, 2))

            Jan        Feb        Mar        Apr        May        Jun
1959             2.2969052  0.8767484 -1.4039698 -0.2988863 -0.8898592
1960  0.1982984  1.0108153 -1.0308821 -1.0062780
            Jul        Aug        Sep        Oct        Nov        Dec
1959 -1.0091047 -0.6374327  0.1238307  0.1633279 -0.5258319  0.6676121
1960
```

```
> x <- ts(rnorm(25), frequency = 4, start = c(1959, 2))
> window(x, start = c(1960, 3), end = c(1962, 1))

           Qtr1        Qtr2        Qtr3        Qtr4
1960                          -0.65513579 -0.05734034
1961  0.11080975 -1.40919018  0.10168438  1.95392119
1962  0.33553384

> plot(x)
```



4.2. Séries Irregulares

O R tem vários packages para lidar com este tipo de dados:

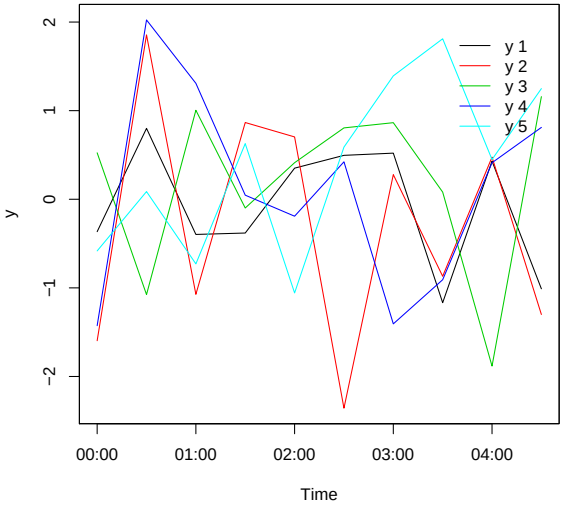
- A package `its`
- A package `tseries`
- A package `fBasics`
- A package `zoo`

Vamos mostrar exemplos com a package `zoo`.

```
> library(zoo)
> x1 <- zoo(rnorm(100), seq(as.POSIXct("2000-01-01"), len = 100,
+     by = "day"))
> x1[1:5]

2000-01-01 2000-01-02 2000-01-03 2000-01-04 2000-01-05
-2.1442609  0.4426790 -0.7544800 -0.8438207 -0.5584058

> x2 <- zoo(rnorm(3), as.Date(c("2005-01-01", "2005-01-10", "2005-01-12")))
> x2

2005-01-01 2005-01-10 2005-01-12
 0.1748442  0.3955837  1.4343807
```

```
> y <- zoo(matrix(rnorm(50), 10, 5), seq(as.POSIXct("2000-01-01"),
+     len = 10, by = "30 sec"))
> y[1:5, ]

2000-01-01 00:00:00 -0.3657535 -1.5963686  0.52418051 -1.42659695 -0.58297690
2000-01-01 00:00:30  0.8002492  1.8554143 -1.07585318  2.02375055  0.08681937
2000-01-01 00:01:00 -0.3970112 -1.0730203  1.00442740  1.30865383 -0.72929109
2000-01-01 00:01:30 -0.3810517  0.8661157 -0.09859802  0.04634589  0.62796968
2000-01-01 00:02:00  0.3510529  0.7035074  0.41266587 -0.19108791 -1.05719173

> z <- zoo(matrix(rnorm(50), 10, 5), seq(as.POSIXct("2000-01-01"),
+     len = 10, by = "sec"))
> z[1:5, ]

2000-01-01 00:00:00 -2.137443 -1.1133699  0.5150506 -0.1608773 -0.02425833
2000-01-01 00:00:01  1.034706 -0.2604301 -0.5848931 -0.8202178 -0.75730997
2000-01-01 00:00:02  1.605525 -0.9775292  0.6597066  0.0406544 -0.15549174
2000-01-01 00:00:03  1.194794 -1.7096078  0.7523290  0.1417471  1.68069598
2000-01-01 00:00:04 -1.520159 -1.0700318 -0.2265825  1.0175490  0.43173557
```

```
> plot(y, plot.type = "single", xlab = "Time", col = 1:5)
> library(gplots)
> smartlegend("right", "top", paste("y", 1:5), col = 1:5, lty = 1,
+     bty = "n")
```

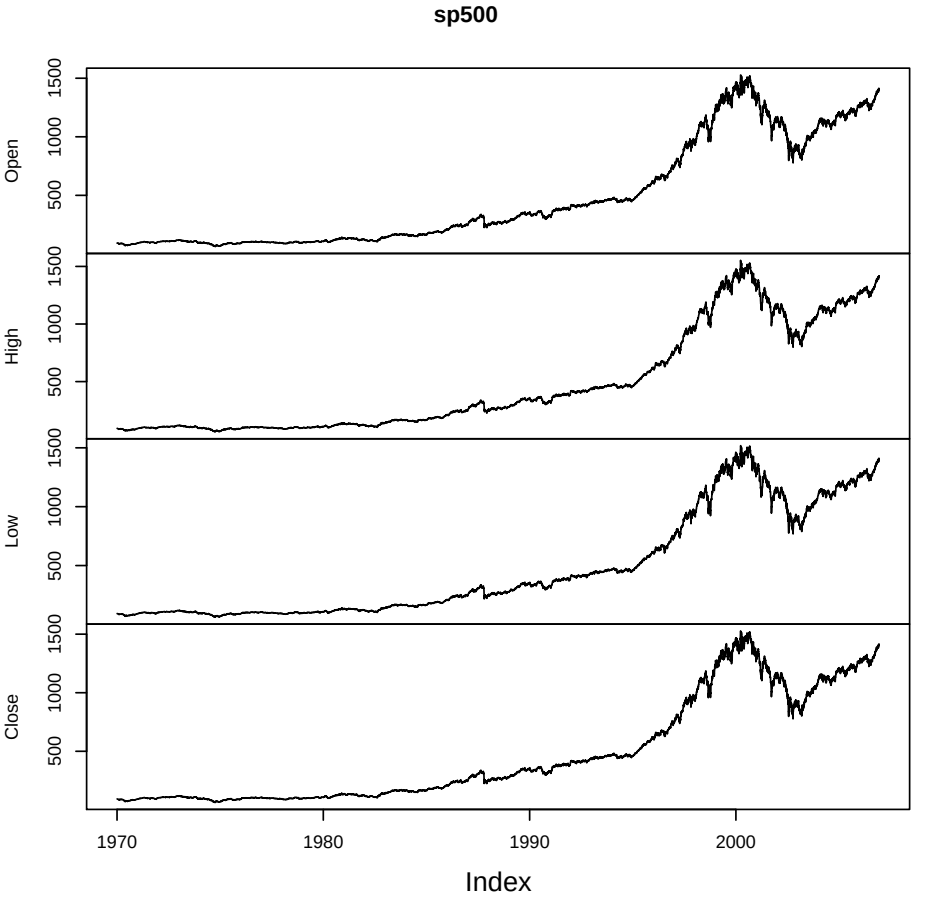


Ir buscar à Internet algumas cotações do índice bolsista SP500...

```
> library(tseries)
> sp500 <- get.hist.quote("^GSPC", start = "1970-01-02", quote = c("Open",
+     "High", "Low", "Close"), origin = "1970-01-01")
> sp500 <- as.zoo(sp500)
> time(sp500) <- as.Date(time(sp500))
> sp500 <- na.omit(sp500)
> head(sp500)

            Open  High   Low Close
1970-01-02 92.06 93.54 91.79 93.00
1970-01-05 93.00 94.25 92.53 93.46
1970-01-06 93.46 93.81 92.13 92.82
1970-01-07 92.82 93.38 91.93 92.63
1970-01-08 92.63 93.47 91.99 92.68
1970-01-09 92.68 93.25 91.82 92.40
```

Gráficos de séries temporais...

```
> plot(sp500)
```



Médias móveis...

```
> ma <- function(x, size = 10) {
+     rollmean(x, size, na.pad = T, align = "right")
+ }

> plot(cbind(sp500[1:1000, "Close"], ma(sp500[1:1000, "Close"],
+     50), ma(sp500[1:1000, "Close"], 100)), plot.type = "single",
+     main = "", type = "l", ylab = "Value", xlab = "", col = 1:3)
> smartlegend("right", "bottom", c("Close", "MA(50)", "MA(100)"),
+     col = 1:3, lty = 1)
```