

Minha primeira sessão com o **knitr**

Walmes Zeviani

1 Fontes

Para uma introdução em português ao knitr visite o tutorial do [Fernando Mayer](#). Para conhecer as opções disponíveis a serem usadas nos chunks visite a página mantida pelo desenvolvedor do knitr [Yihui Xie](#).

2 Opções para os chunks

De forma bem resumida, as opções básicas para controle do código nos chunks são:

- `echo = {TRUE, FALSE};`
- `results = {"markup", "hide", "asis", "hold"};`
- `eval = {TRUE, FALSE};`
- `cache = {TRUE, FALSE};`
- `message = {TRUE, FALSE};`
- `warning = {TRUE, FALSE};`
- `error = {TRUE, FALSE};`

3 Exemplos de uso para código

Vamos carregar o conjunto de dados `iris` no chunk abaixo.

1. Mostra input e output:

```
data(iris)
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 ...
```

2. Esconde output:

```
data(iris)
str(iris)
```

3. Esconde input:

```
## '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 ...
```

4. Remove os realces;

```
data(iris)
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 ...
```

5. Sem os hash para comentários;

```
data(iris)
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 ...
```

6. Com cor diferente para o fundo;

```
data(iris)
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 ...
```

4 Inserindo valores calculados

```
x <- iris$Sepal.Length
m <- mean(x); m
s <- sd(x); s
n <- length(x); n
st <- Sys.time(); st
h <- format(st, format="%H:%M")
d <- format(st, format="%d/%m/%Y")
r <- R.version.string; r
r <- gsub(".*version (.*) \\(\\.\\.\\.\"", "\\1", r)

## [1] 5.843
## [1] 0.8281
## [1] 150
## [1] "2013-11-28 17:31:00 BRST"
## [1] "R version 3.0.1 (2013-05-16)"
```

A média do comprimento das sépalas é de 5.843 ± 0.828 e o tamanho da amostra é de 150 elementos. Esse cálculo foi feito às 17:31 do dia 28/11/2013 usando a versão 3.0.1 do R.

5 Inserindo tabelas com xtable::xtable e Hmisc::latex

1. Tabela em saída R:

```
tb <- aggregate(Sepal.Length~Species, data=iris, mean)
tb

##      Species Sepal.Length
## 1      setosa      5.006
## 2 versicolor      5.936
## 3 virginica      6.588

require(plyr)

## Loading required package: plyr

tb <- ddply(iris, .(Species), colwise(mean))
tb

##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1      setosa      5.006      3.428      1.462      0.246
## 2 versicolor      5.936      2.770      4.260      1.326
## 3 virginica      6.588      2.974      5.552      2.026
```

2. Usando o pacote `xtable` para exportar:

```
options(OutDec=",")
require(xtable)

## Loading required package: xtable

cap <- "Tabela com as médias para as espécies."
xt <- xtable(tb, digits=4, caption=cap)
xt
```

	Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	setosa	5,0060	3,4280	1,4620	0,2460
2	versicolor	5,9360	2,7700	4,2600	1,3260
3	virginica	6,5880	2,9740	5,5520	2,0260

Tabela 1: Tabela com as médias para as espécies.

3. Controlando mais detalhes na saída, como legenda em cima:

```
print(xt, caption.placement="top", include.rownames=FALSE)
```

Tabela 2: Tabela com as médias para as espécies.

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5,0060	3,4280	1,4620	0,2460
versicolor	5,9360	2,7700	4,2600	1,3260
virginica	6,5880	2,9740	5,5520	2,0260

4. Usando o pacote `Hmisc`:

```
require(Hmisc)
latex(tb, file="", caption=cap, where="h", rowname=NULL)
```

5. Usando o pacote `tables` para obter tabelas descritivas:

Tabela 3: Tabela com as médias para as espécies.

Species	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5,006	3,428	1,462	0,246
versicolor	5,936	2,770	4,260	1,326
virginica	6,588	2,974	5,552	2,026

```
cat(sprintf("\\caption{%s}\\label{tab:tabular}\\n", cap))
```

Tabela 4: Média e desvio-padrão para comprimento e largura das sépalas

```
cat("\\begin{center}\\n")
```

```
latex(xtt, caption=cap, table.env=TRUE)
```

Species	n	Sepal.Length		Sepal.Width	
		mean	sd	mean	sd
setosa	50	5,01	0,35	3,43	0,38
versicolor	50	5,94	0,52	2,77	0,31
virginica	50	6,59	0,64	2,97	0,32
All	150	5,84	0,83	3,06	0,44

```
cat("\\end{center}\\n\\end{table}\\n")
```

```
require(tables)
## help(tabular, help_type="html")
xtt <- tabular((Species+1)~(n=1)+Format(digits=2)*
               (Sepal.Length+Sepal.Width)*(mean+sd), data=iris)
xtt

##
##          Sepal.Length      Sepal.Width
## Species  n  mean      sd  mean      sd
## setosa   50 5,01      0,35 3,43      0,38
## versicolor 50 5,94      0,52 2,77      0,31
## virginica 50 6,59      0,64 2,97      0,32
## All      150 5,84      0,83 3,06      0,44
```

6. Usando o `Hmisc` para converter:

```
cap <- "Média e desvio-padrão para comprimento
e largura das sépalas"
cat("\\begin{table}\\n")
```

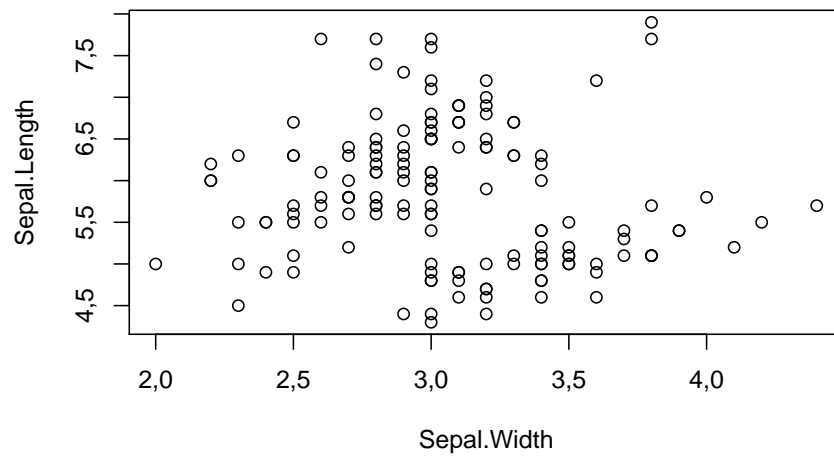
7. Consulte a documentação das funções para ter mais detalhes:

```
help(xtable, help_type="html")
help(print.xtable, help_type="html")
help(latex, help_type="html")
```

6 Figuras

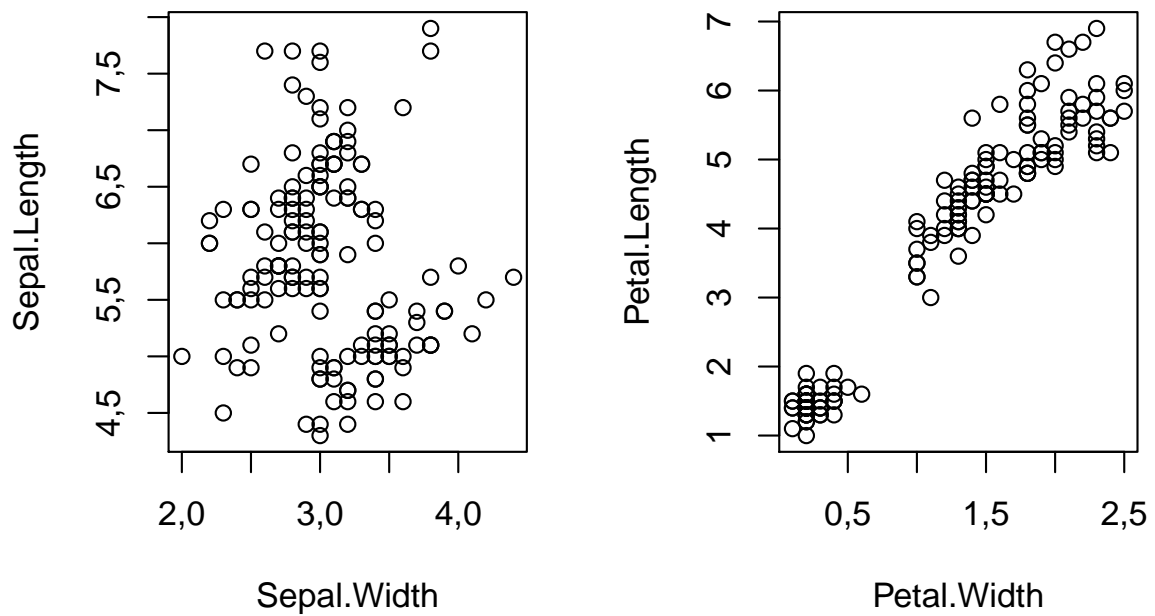
1. Figura sem ser flutuante:

```
plot(Sepal.Length~Sepal.Width, iris)
```



2. Dois gráficos juntos na mesma janela:

```
par(mfrow=c(1,2))
plot(Sepal.Length~Sepal.Width, iris)
plot(Petal.Length~Petal.Width, iris)
```



```
layout(1)
```

3. Figura como objeto flutuante e legenda:

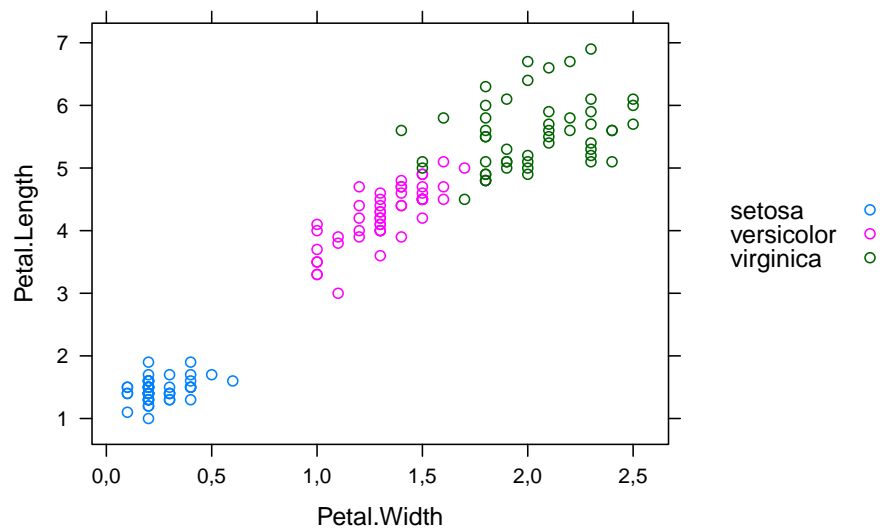


Figura 1: Comprimento em função da largura para a pétala.

```
require(lattice)

## Loading required package: lattice

xyplot(Petal.Length~Petal.Width, groups=Species,
       data=iris, auto.key=list(space="right"))
```

4. Duas figuras uma sobre a outra:

```
xyplot(Petal.Length~Petal.Width, groups=Species,
       data=iris, auto.key=list(space="right"))
xyplot(Petal.Length~Petal.Width|Species, type=c("p", "r"),
       data=iris, layout=c(NA, 1))
```

5. Duas figuras lado a lado:

```
xyplot(Petal.Length~Petal.Width, groups=Species,
       data=iris, auto.key=list(space="top", columns=3))
xyplot(Petal.Length~Petal.Width|Species, type=c("p", "r"),
       data=iris, layout=c(1, NA))
```

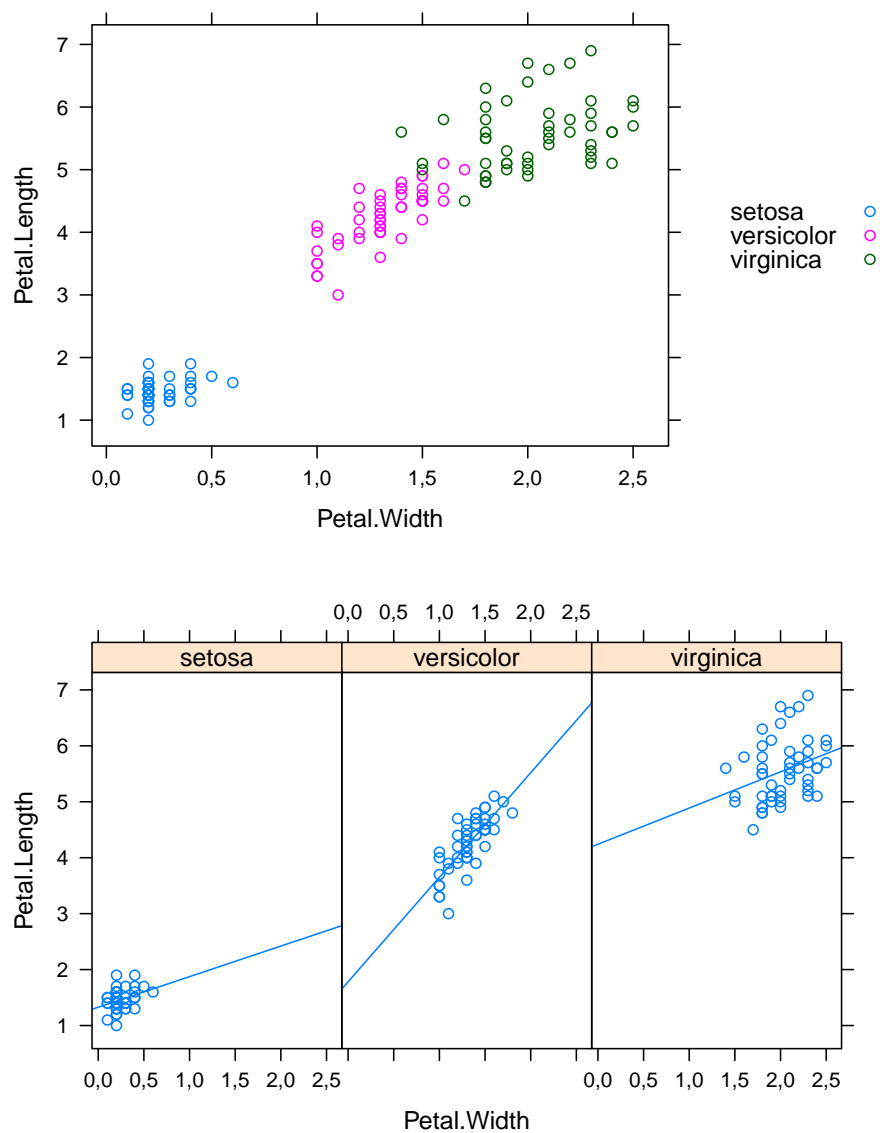


Figura 2: Comprimento em função da largura para a pétala.

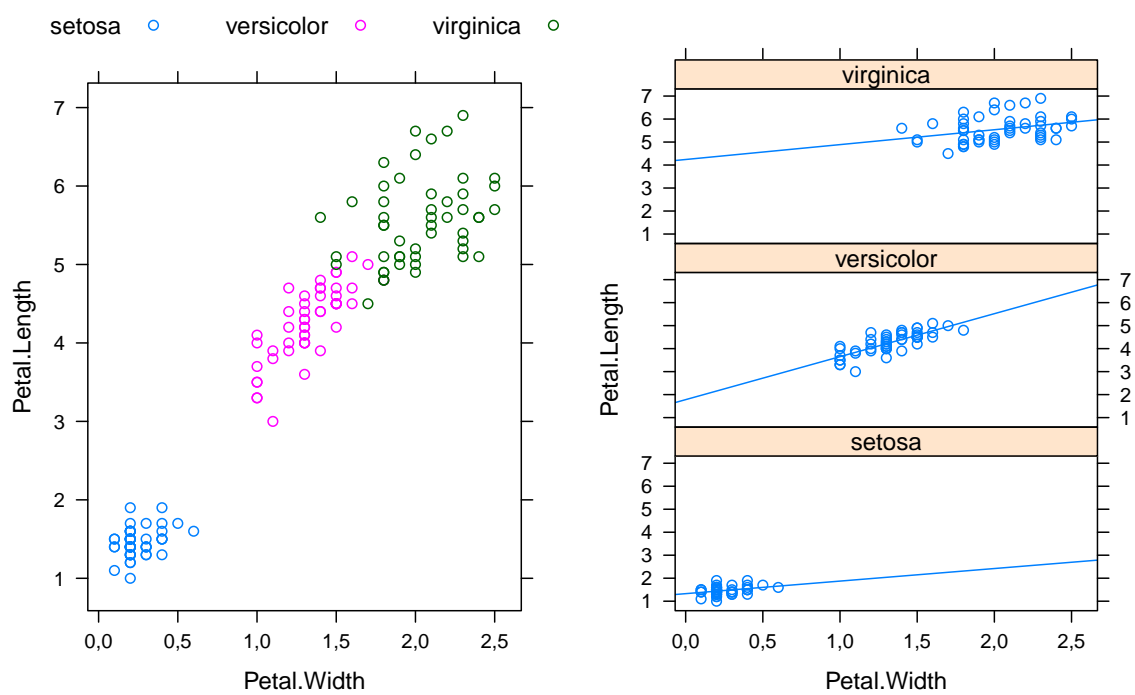


Figura 3: Comprimento e largura.