



# Introdução a RStudio

(v 1.3)

*Oscar Torres-Reyna*

*otorres@princeton.edu*

*Tradução: Bárbara Bettencourt*



August 2013

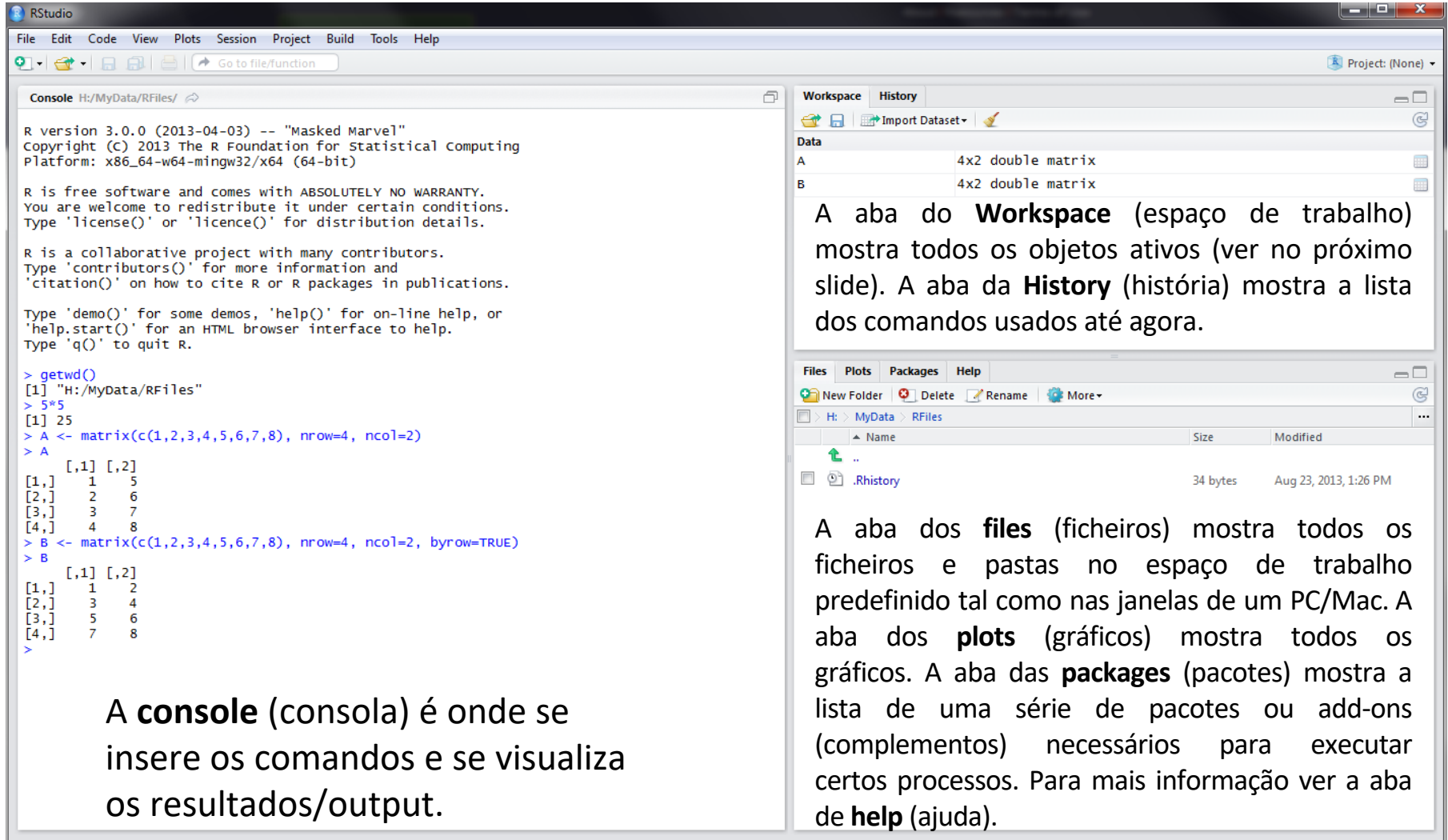
 <http://dss.princeton.edu/training/>

# Introdução

RStudio permite ao utente executar R numa forma mais acessível. É de open-source (ou seja grátis) e está disponível em <http://www.rstudio.com/>

Para tutoriais ou recursos relativos ao R veja os links seguintes: <http://dss.princeton.edu/training/>  
<http://libguides.princeton.edu/dss>

# Tela do RStudio



The screenshot displays the RStudio environment with three main panels:

- Console:** Shows the R version (3.0.0), copyright information, and a series of commands and their outputs. The commands include `getwd()`, `5*5`, `matrix()`, and `matrix()` with `byrow=TRUE`.
- Workspace:** Lists active objects in the environment, including `A` and `B`, both identified as `4x2 double matrix`.
- Files:** Displays the file explorer for the current project, showing a folder named `.Rhistory` with a size of 34 bytes and a modification date of August 23, 2013.

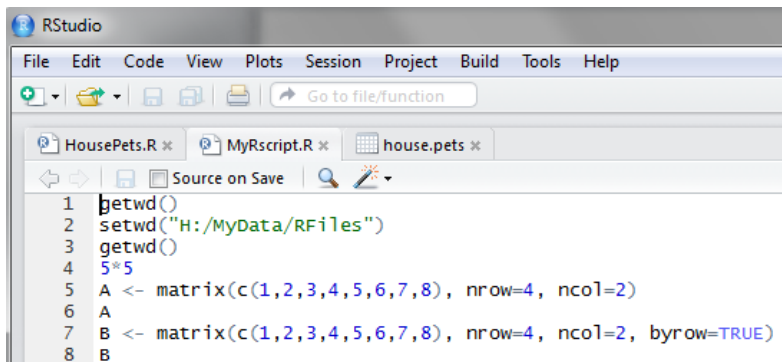
**Aba do **Workspace** (espaço de trabalho)** mostra todos os objetos ativos (ver no próximo slide). A aba da **History** (história) mostra a lista dos comandos usados até agora.

**A aba dos **files** (ficheiros)** mostra todos os ficheiros e pastas no espaço de trabalho predefinido tal como nas janelas de um PC/Mac. A aba dos **plots** (gráficos) mostra todos os gráficos. A aba das **packages** (pacotes) mostra a lista de uma série de pacotes ou add-ons necessários para executar certos processos. Para mais informação ver a aba de **help** (ajuda).

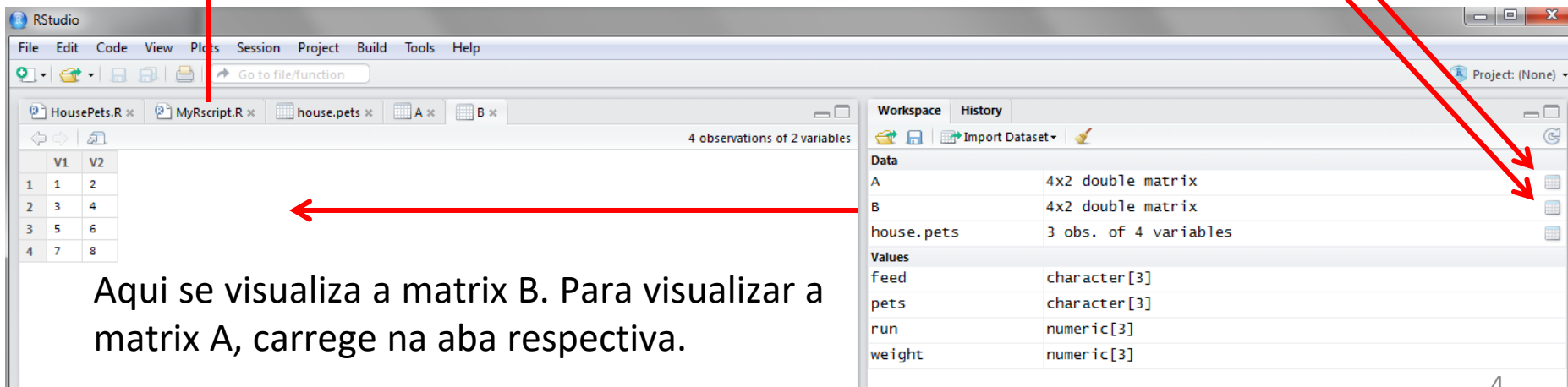
**A console (consola) é onde se insere os comandos e se visualiza os resultados/output.**

# Aba do Workspace (1)

A aba do workspace armazena qualquer objeto, valor, função ou algo que tenha sido criado durante a sessão de R. No exemplo abaixo, se clicar nos quadrados indicados, a data consegue-se ser visualizada no ecrã ao lado esquerdo.



```
1 setwd()
2 setwd("H:/MyData/RFiles")
3 getwd()
4 5*5
5 A <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2)
6 A
7 B <- matrix(c(1,2,3,4,5,6,7,8), nrow=4, ncol=2, byrow=TRUE)
8 B
```



4 observations of 2 variables

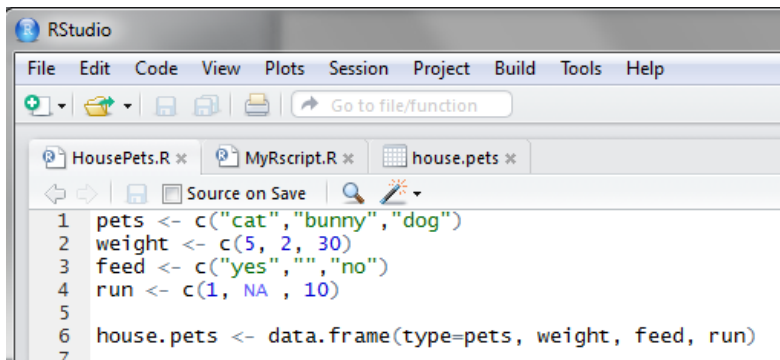
	V1	V2
1	1	2
2	3	4
3	5	6
4	7	8

Aqui se visualiza a matrix B. Para visualizar a matrix A, carregue na aba respectiva.

Workspace	
Data	
A	4x2 double matrix
B	4x2 double matrix
house.pets	3 obs. of 4 variables
Values	
feed	character[3]
pets	character[3]
run	numeric[3]
weight	numeric[3]

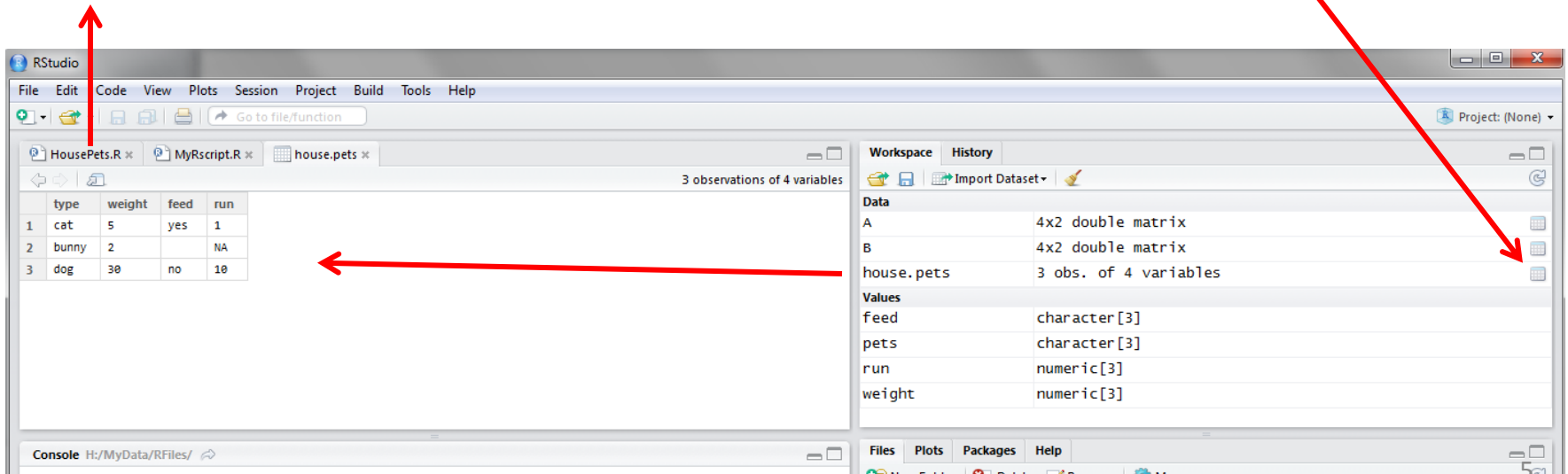
# Aba do Workspace (2)

Aqui temos outro exemplo sobre o aspeto da aba do workspace quando mais objetos são adicionados. Note que a data frame (estrutura de dados) `house.pets` é constituída por diferentes valores or vectores individuais.



```
1 pets <- c("cat","bunny","dog")
2 weight <- c(5, 2, 30)
3 feed <- c("yes","", "no")
4 run <- c(1, NA, 10)
5
6 house.pets <- data.frame(type=pets, weight, feed, run)
7
```

Clique nos quadrados indicados para visualizar o dataset (conjunto de dados) em estrutura de spreadsheet (planilha).



The screenshot shows the RStudio interface with the 'house.pets' data frame loaded. The Data Viewer on the left displays the data in a spreadsheet format, and the Environment pane on the right shows the variables and their types.

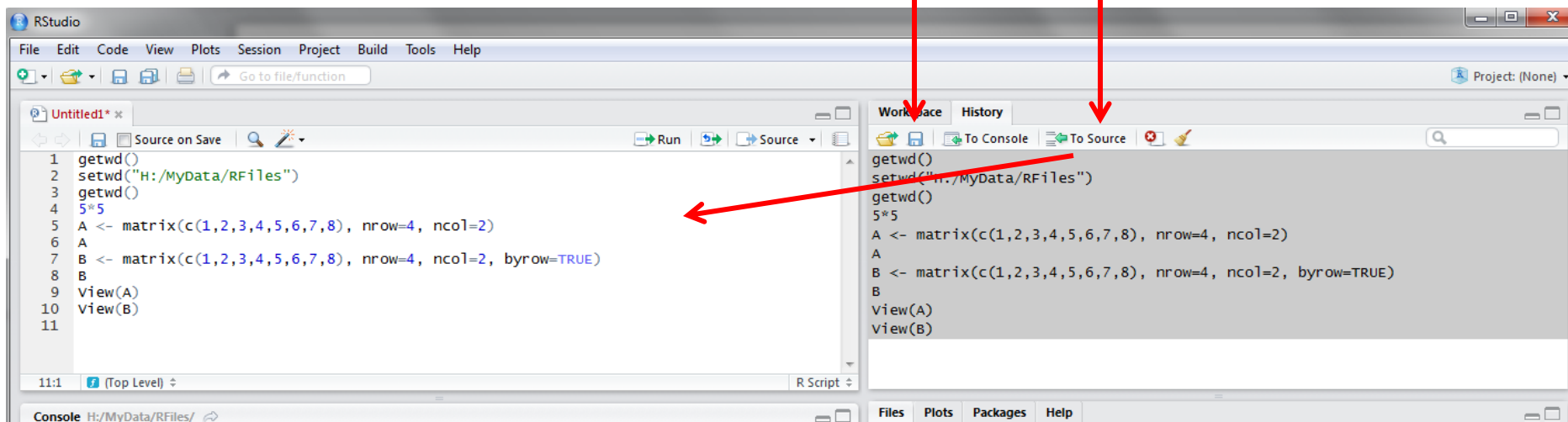
	type	weight	feed	run
1	cat	5	yes	1
2	bunny	2	NA	NA
3	dog	30	no	10

Environment pane:

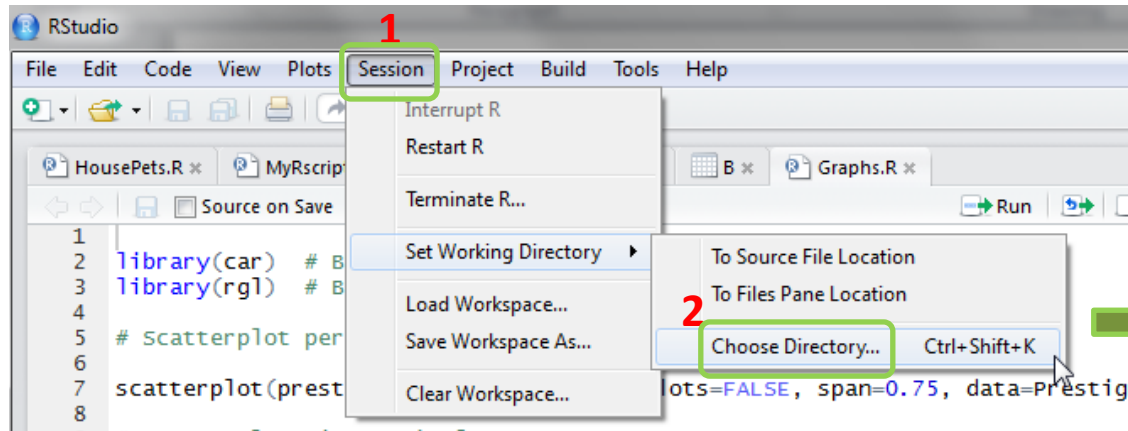
- Data
- A: 4x2 double matrix
- B: 4x2 double matrix
- house.pets: 3 obs. of 4 variables
- Values
- feed: character [3]
- pets: character [3]
- run: numeric [3]
- weight: numeric [3]

# Aba da History

A aba da history mantém um registo de todos os comandos previamente utilizados. Isto é útil quando se está a testar e executar processos. Aqui pode-se **guardar** uma lista inteira ou **selecionar** certos comandos e enviá-los para um R script para registar os passos do seu trabalho. Neste exemplo, selecionamos todos e clicamos no ícone “To Source” (para a origem), uma janela á esquerda vai abrir com uma lista de comandos. Não se esqueça de guardar o ficheiro ‘untitled1’ como um \*.R script.



# Alterando o working directory



Se tiver diferentes projetos pode mudar a working directory de essa sessão, ver acima.  
Ou pode escrever:

```
# Mostra o working directory (wd)
```

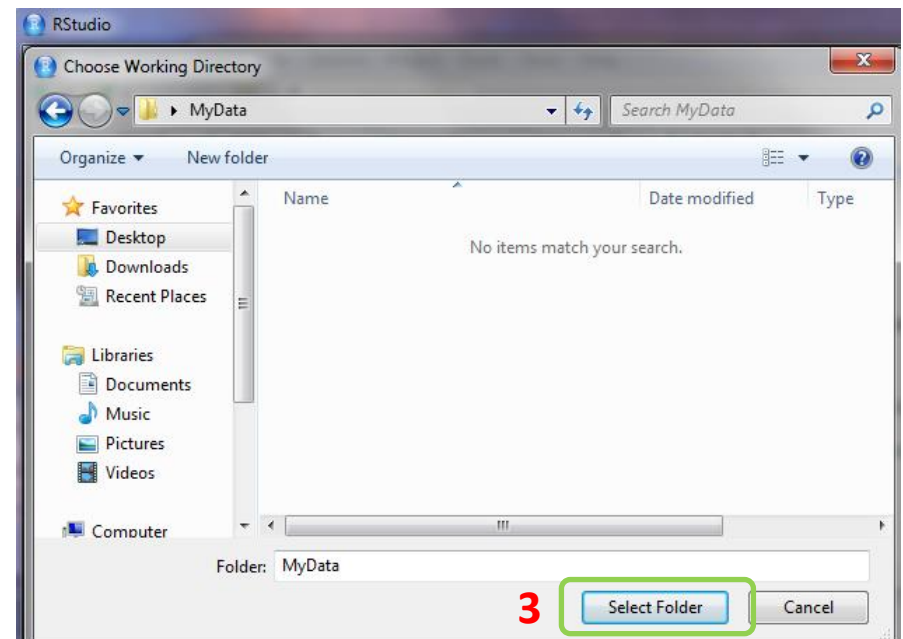
```
getwd()
```

```
# Muda o wd
```

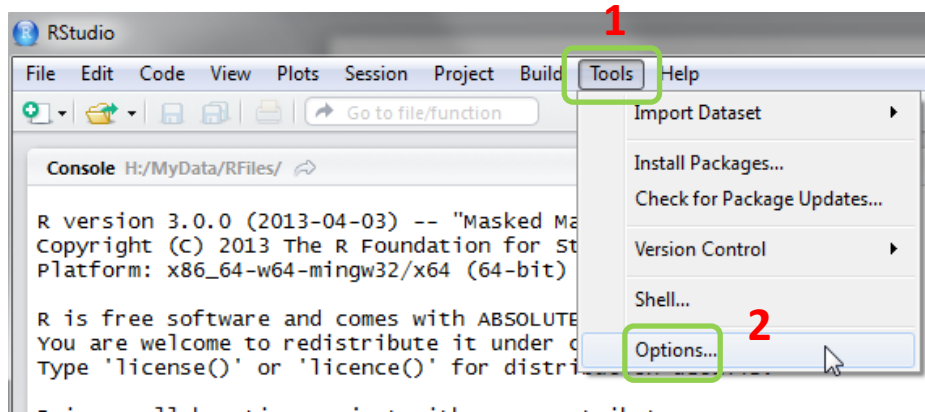
```
setwd("C:/myfolder/data")
```

Para mais informações veja este documento:

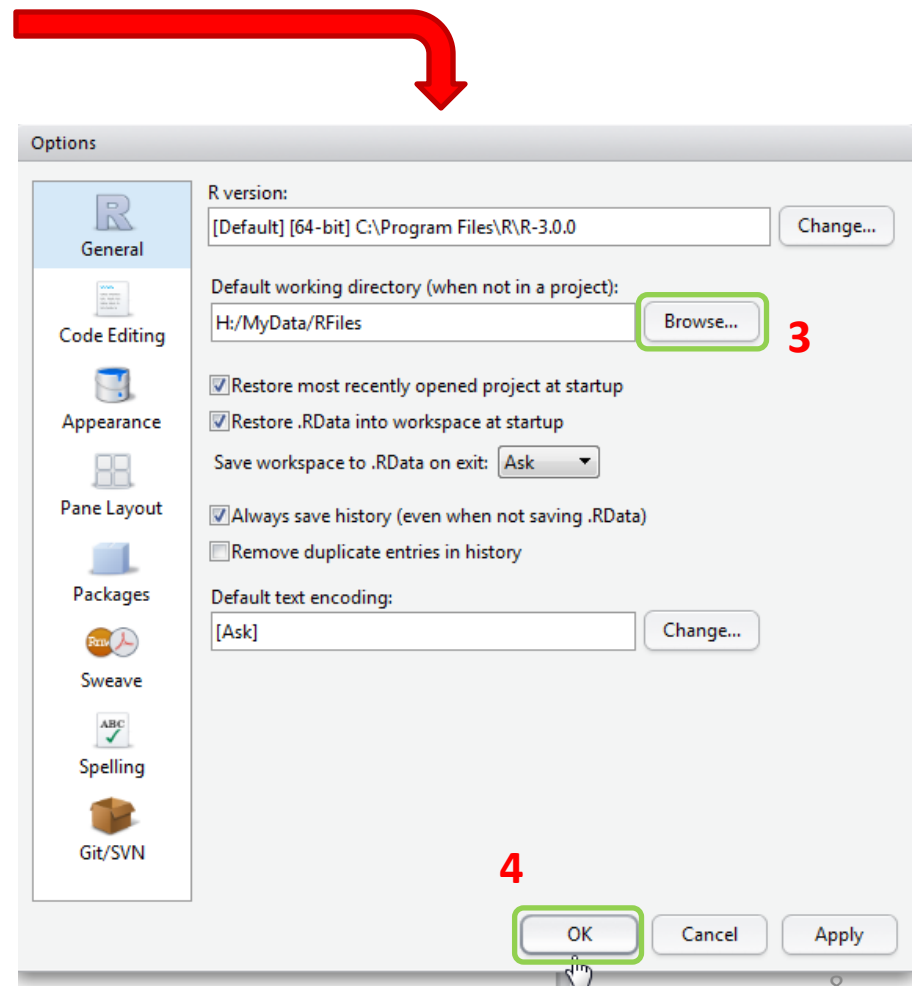
<http://dss.princeton.edu/training/RStata.pdf>



# Configurando um working directory predefinido



Cada vez que se abre o RStudio, ele abre diretamente para o directory predefinido. Pode mudar o predefinido para uma pasta onde contém os seus datafiles para que não precise de definir o working directory todas as vezes. No menu vá para Tools->Options



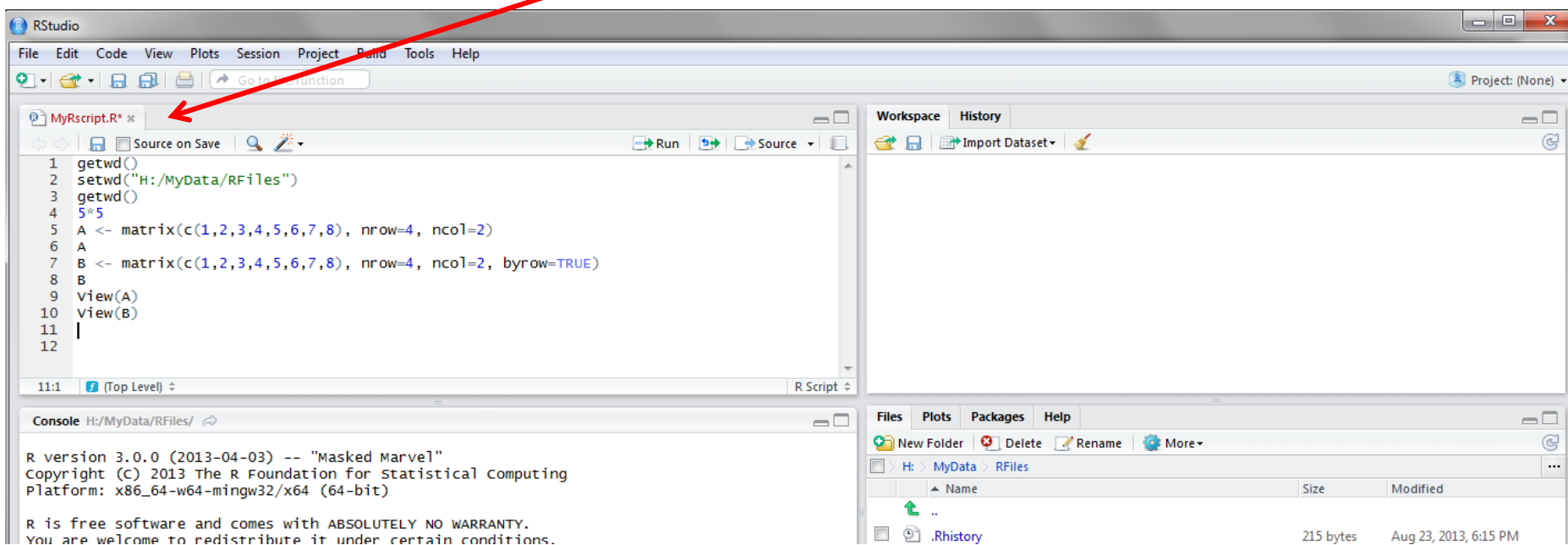


# R script (1)

O ecrã usual do Rstudio tem quatro janelas:

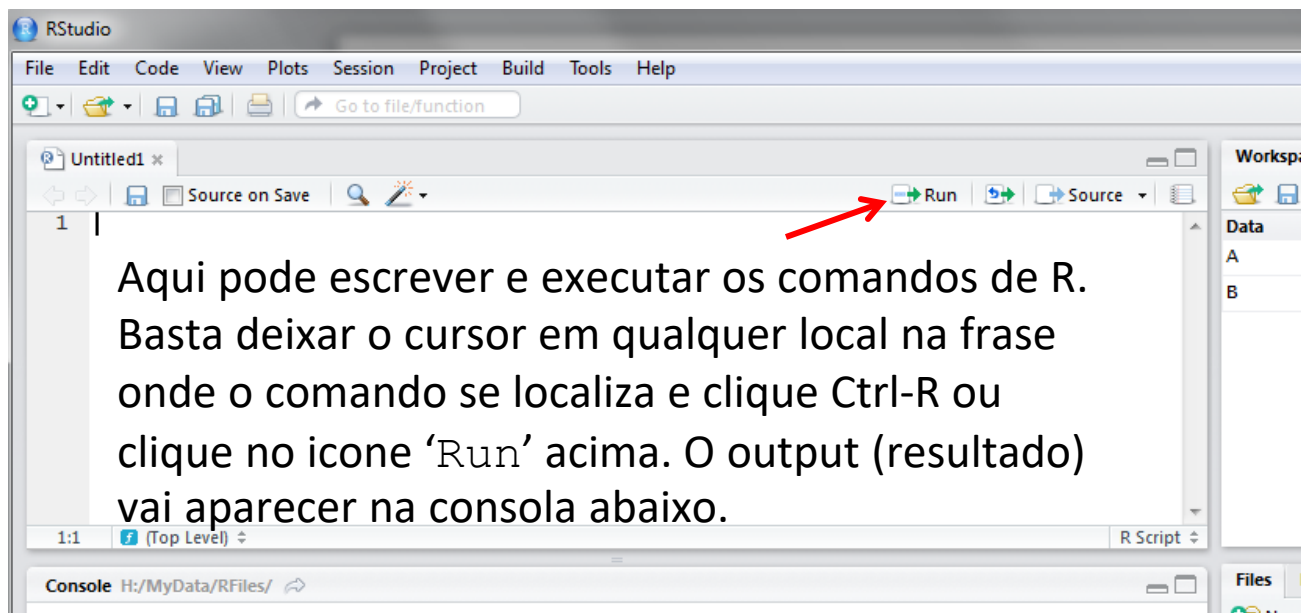
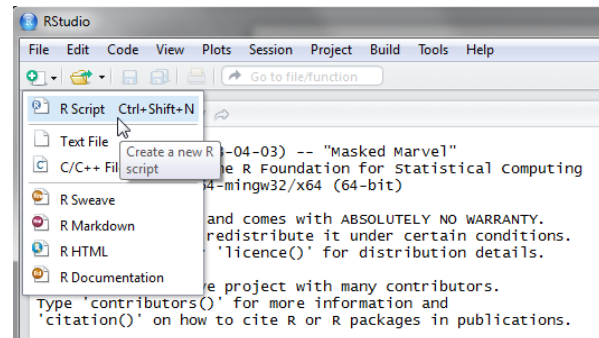
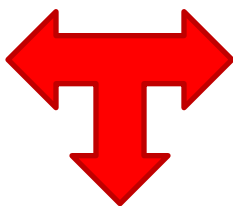
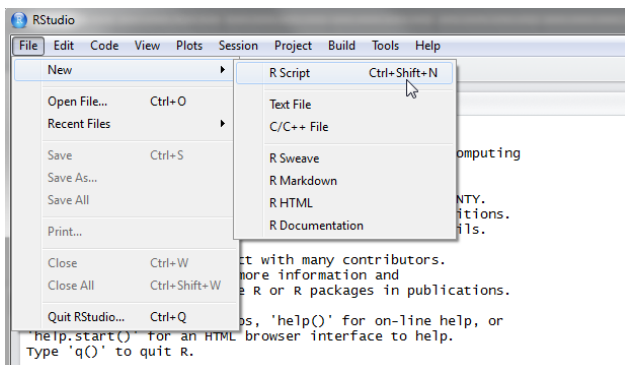
1. Console (Consola).
2. Workspace e history (Espaço de trabalho e história).
3. Files, plots, packages e help (Ficheiros, gráficos, pacotes e ajuda).
4. Os R script(s) e data view (visualizar a data).

O R script é onde se regista os passos do trabalho. Para utilizadores de Stata esta seria a sua página preferida, para utilizadores de SPSS é como a syntax e para utilizadores de SAS é como o programa SAS.



# R script (2)

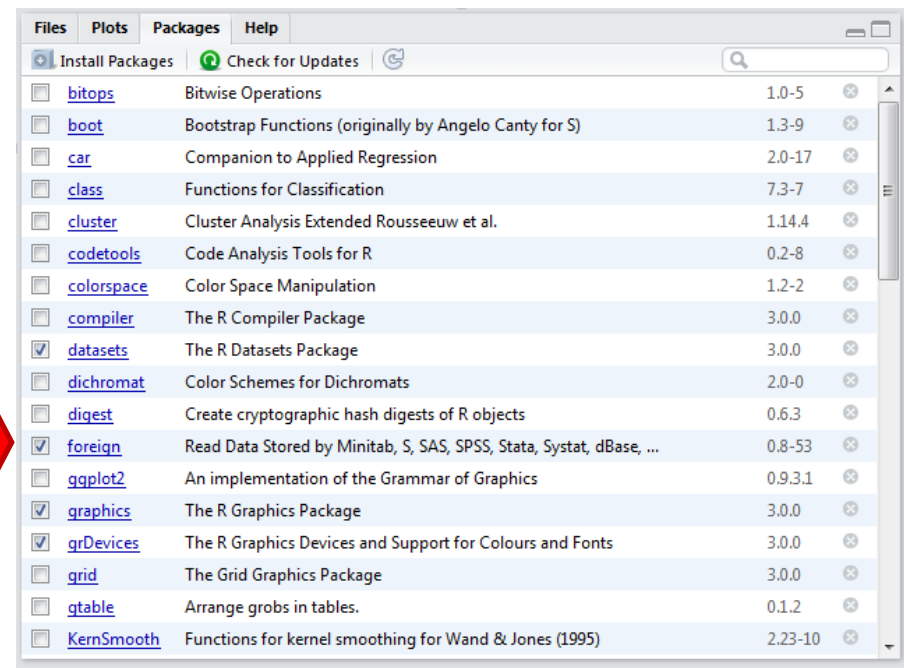
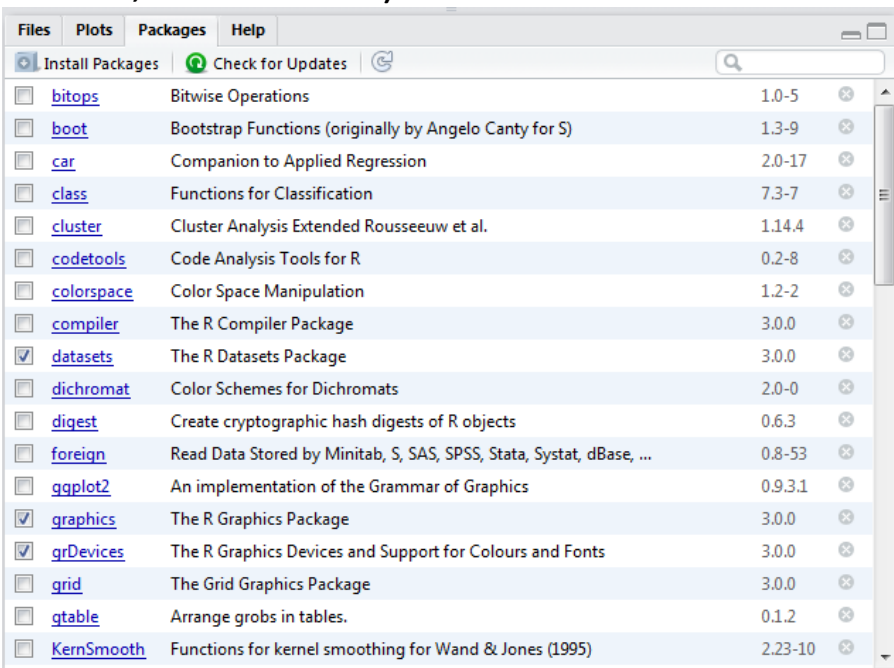
Para criar um novo R Script carregue em `File -> New -> R Script`, ou clique no icon com o sinal “+” e clique em “R Script”, ou então simplesmente clique em `Ctrl+Shift+N`. Não se esqueça de guardar o script.



Aqui pode escrever e executar os comandos de R. Basta deixar o cursor em qualquer local na frase onde o comando se localiza e clique `Ctrl-R` ou clique no icone ‘Run’ acima. O output (resultado) vai aparecer na consola abaixo.

# Aba das Packages

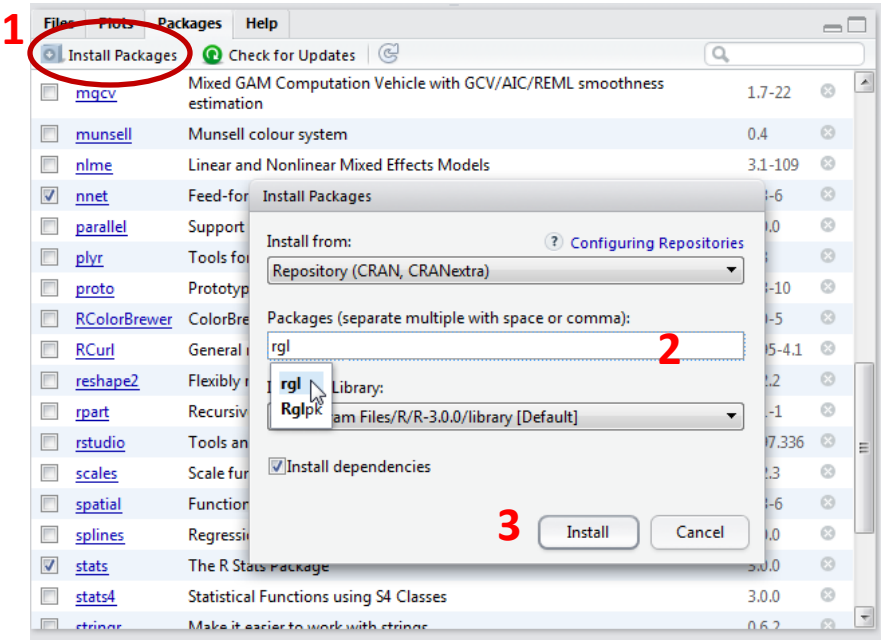
A aba das packages (pacotes) mostra a lista dos add-ons incluídos na instalação do RStudio. Se verificados, o pacote é carregado dentro de R, se não, qualquer comando relacionado com esse pacote não vai funcionar, vai ter de seleccioná-lo. Também pode instalar outros add-ons ao carregar no ícone 'Install Packages'. Outro método para ativar o pacote é ao escrever, por exemplo, `library(foreign)`. Isto vai automaticamente verificar o `--foreign` pacote (isto ajuda a trazer a data de formatos diferentes como a Stata, SAS or SPSS).



# Instalando as Packages

<input type="checkbox"/>	<a href="#">RCurl</a>	General network (HTTP/FTP/...) client interface for R	1.95-4.1	×
<input type="checkbox"/>	<a href="#">reshape2</a>	Flexibly reshape data: a reboot of the reshape package.	1.2.2	×
<input type="checkbox"/>	<a href="#">rpart</a>	Recursive Partitioning	4.1-1	×

**Before**



Vamos instalar o pacote-`rgl` (útil para criar gráficos de imagens 3D). Não vem automaticamente com o R original.

Carregue em “Install Packages”, escreva o nome na janela que abriu e carregue em “Install”.

**After**

<input type="checkbox"/>	<a href="#">RCurl</a>	General network (HTTP/FTP/...) client interface for R	1.95-4.1	×
<input type="checkbox"/>	<a href="#">reshape2</a>	Flexibly reshape data: a reboot of the reshape package.	1.2.2	×
<input type="checkbox"/>	<a href="#">rgl</a>	3D visualization device system (OpenGL)	0.93.952	×
<input type="checkbox"/>	<a href="#">rpart</a>	Recursive Partitioning	4.1-1 12	×

# Aba dos Plots (1)

RStudio

File Edit Code View Plots Session Project Build Tools Help

Go to file/function

HousePets.R \* MyRscript.R \* house.pets \* A \* B \* Graphs.R \* \*

Source on Save Run Source

```
1
2 library(car) # By John Fox and Sanford Weisberg
3 library(rgl) # By Daniel Adler and Duncan Murdoch
4
5 # Scatterplot per group
6
7 scatterplot(prestige ~ income|type, boxplots=FALSE, span=0.75, data=Prestige)
8
9 # Scatterplots in matrix form
10
11 scatterplotMatrix(~ prestige + income + education, span=0.7, data=Prestige)
12
13 # 3D graph, scatter3d is from the --car package. It will open in a separate window.
14
15 scatter3d(prestige ~ income + education, id.n=3, data=Duncan)
```

11:7 (Top Level) R Script

Console H:/MyData/RFiles/

```
> scatterplot(prestige~income|type, boxplots=FALSE, span=0.75, data=Prestige)
> |
```

Workspace History

Import Dataset

Data	
A	4x2 double matrix
B	4x2 double matrix
house.pets	3 obs. of 4 variables
Values	
feed	character[3]
pets	character[3]
run	numeric[3]
weight	numeric[3]

Files Plots Packages Help

Zoom Export Clear All

type

- bc
- △ prof
- + wc

prestige

income

A aba **plots** exibe os gráficos. O gráfico há direita é criado utilizando o comando na linha 7 no script acima.

Ver o slide seguinte para ver o que acontece com mais do que um gráfico.



# Aba dos Plots (2)

The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Project, Build, Tools, and Help. The toolbar below the menu contains icons for file operations and running code. The source editor on the left shows R code for loading packages, creating scatterplots, and a 3D plot. The console on the bottom left shows the execution of the scatterplot and scatterplotMatrix functions. The right pane is divided into Workspace and History tabs, with a table of objects in the workspace. Below this is a sub-menu with Files, Plots, Packages, and Help, and a toolbar with Zoom, Export, and Clear All buttons. The main plot area on the right shows a 3x3 grid of plots for the variables prestige, income, and education. A red arrow points from the 'Plots' icon in the sub-menu to the first plot in the grid.

```
1 library(car) # By John Fox and Sanford Weisberg
2 library(rgl) # By Daniel Adler and Duncan Murdoch
3
4 # Scatterplot per group
5
6 scatterplot(prestige ~ income|type, boxplots=FALSE, span=0.75, data=Prestige)
7
8 # Scatterplots in matrix form
9
10 scatterplotMatrix(~ prestige + income + education, span=0.7, data=Prestige)
11
12 # 3D graph, scatter3d is from the --car package. It will open in a separate window.
13
14 scatter3d(prestige ~ income + education, id.n=3, data=Duncan)
```

Console:

```
> scatterplot(prestige~income|type, boxplots=FALSE, span=0.75, data=Prestige)
> scatterplotMatrix(~ prestige + income + education, span=0.7, data=Prestige)
>
```

Workspace:

Object	Class
A	4x2 double matrix
B	4x2 double matrix
house.pets	3 obs. of 4 variables
feed	character [3]
pets	character [3]
run	numeric [3]
weight	numeric [3]

Plots:

prestige

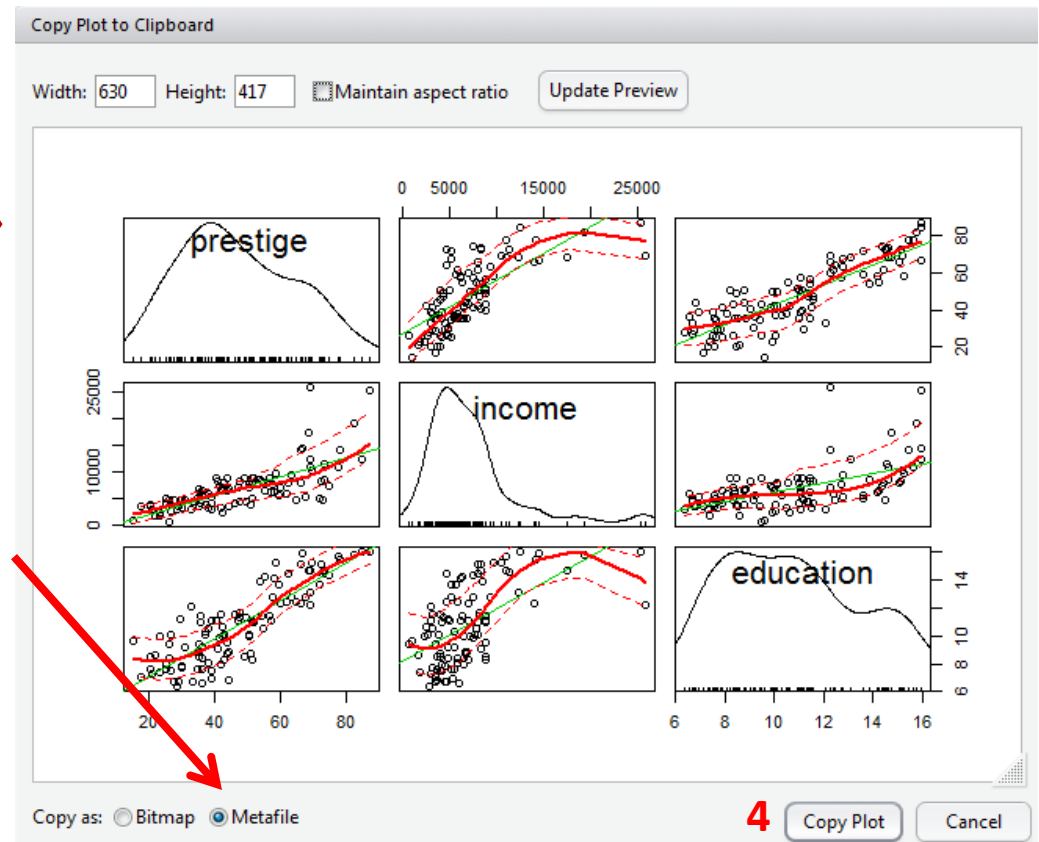
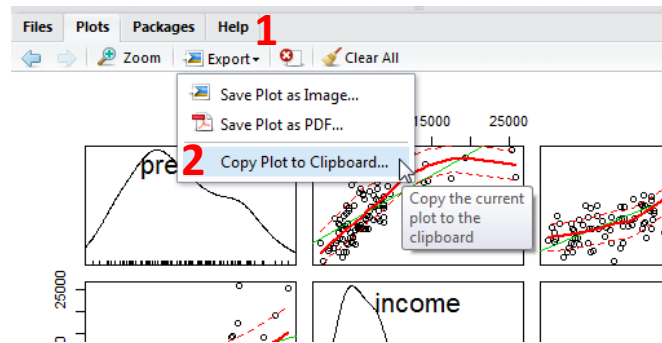
income

education

Aqui temos o segundo gráfico (ver acima linha 11). Se quiser ver o primeiro gráfico, clique no ícone com uma seta que aponta para a esquerda.

# Aba dos Plots (3) – Exportar Gráficos

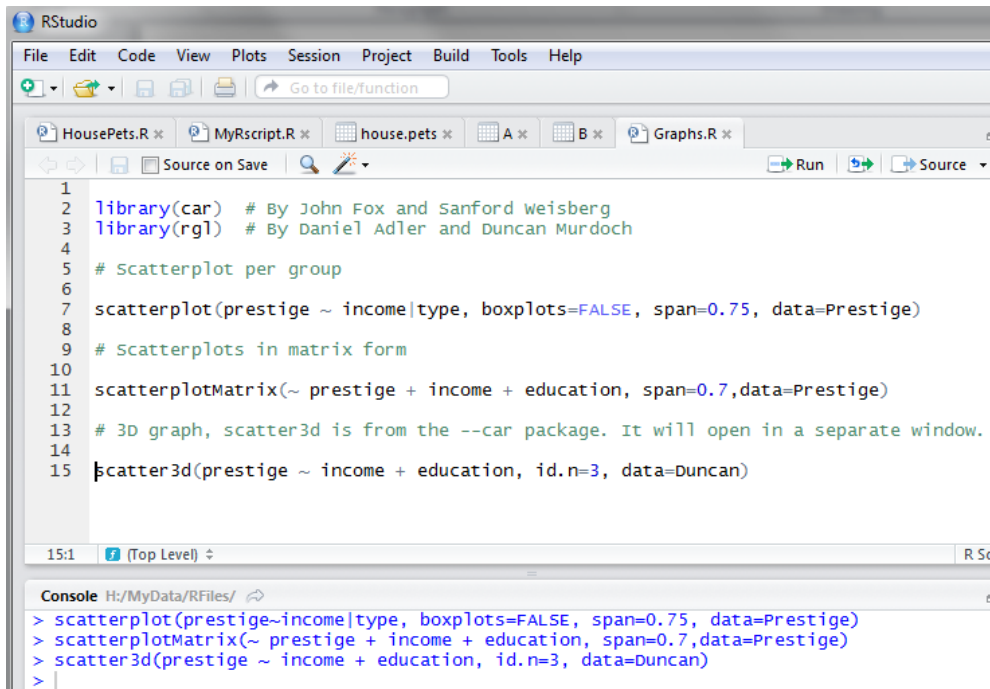
Para extrair o gráfico, clique em “Export” onde pode guardar o ficheiro como uma imagem (PNG, JPG, etc.) ou como PDF, estas opções são úteis quando se quer partilhar o gráfico ou utilizá-lo num documento de LaTeX. A maneira mais fácil de exportar é provavelmente copiar o gráfico para o clipboard e depois colar diretamente para o documento Word.



**3** Verifique que está selecionado ‘Metafile’

**5** Cole para o documento Word

# Gráficos 3D



```
1 library(car) # By John Fox and Sanford Weisberg
2 library(rgl) # By Daniel Adler and Duncan Murdoch
3
4 # Scatterplot per group
5
6 scatterplot(prestige ~ income|type, boxplots=FALSE, span=0.75, data=Prestige)
7
8 # Scatterplots in matrix form
9
10 scatterplotMatrix(~ prestige + income + education, span=0.7, data=Prestige)
11
12 # 3D graph, scatter3d is from the --car package. It will open in a separate window.
13
14
15 scatter3d(prestige ~ income + education, id.n=3, data=Duncan)
```

Console H:/MyData/RFiles/

```
> scatterplot(prestige~income|type, boxplots=FALSE, span=0.75, data=Prestige)
> scatterplotMatrix(~ prestige + income + education, span=0.7, data=Prestige)
> scatter3d(prestige ~ income + education, id.n=3, data=Duncan)
```

Gráficos 3D vão ser exibidos num ecrã separado (ver linha 15 acima). Não vai poder guardá-lo, mas depois de movimentá-lo, quando conseguir encontrar o ângulo preferível, pode fazer um screenshot e copiar e colar para o documento Word.

