

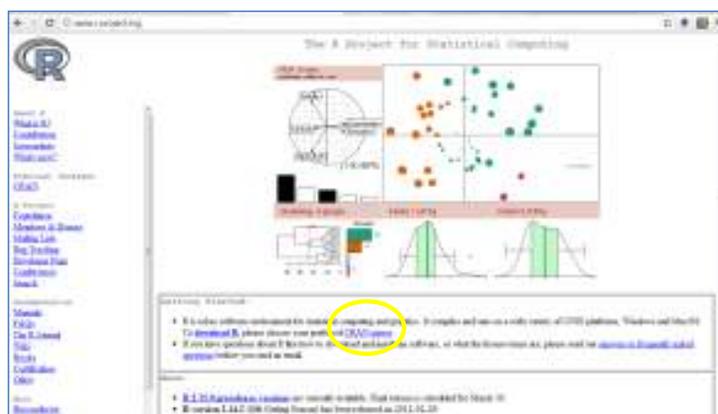
APOSTILA BÁSICA - SOFTWARE R

Luiz Guilherme Scorzafave

O objetivo dessa apostila é apresentar os comandos básicos do software R, especialmente os voltados para a estimação de modelos econométricos.

1. Instalando o R

O R é um software livre que pode ser baixado gratuitamente através do site <http://www.r-project.org>



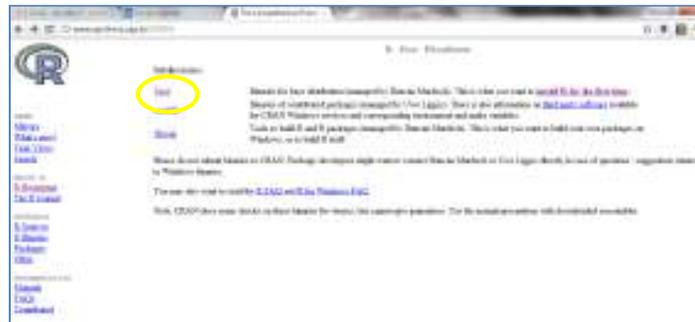
Em seguida, deve-se escolher de qual servidor (clique em CRAN, assinalado em amarelo na imagem acima), o software será baixado para seu computador.



Escolha o seu sistema operacional



Escolha a opção *base*



Clique em download e o software será baixado para sua máquina (45Mb)



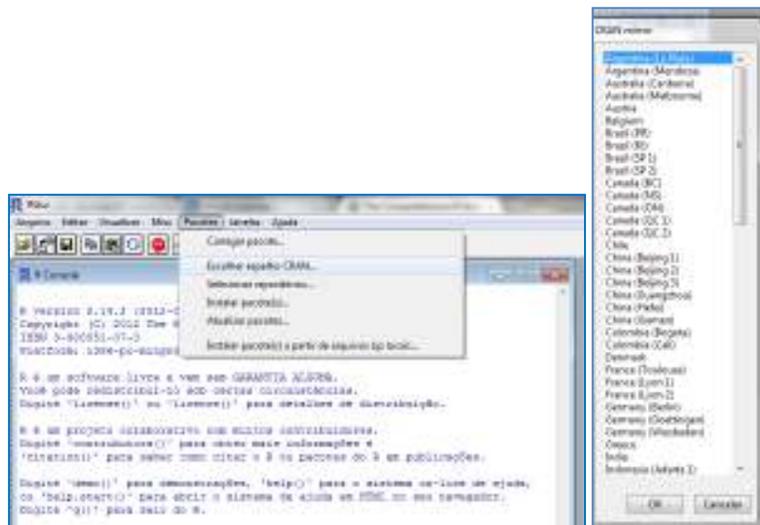
Durante a instalação, escolher a opção de inicialização padrão

Pronto, o R será instalado no seu computador!

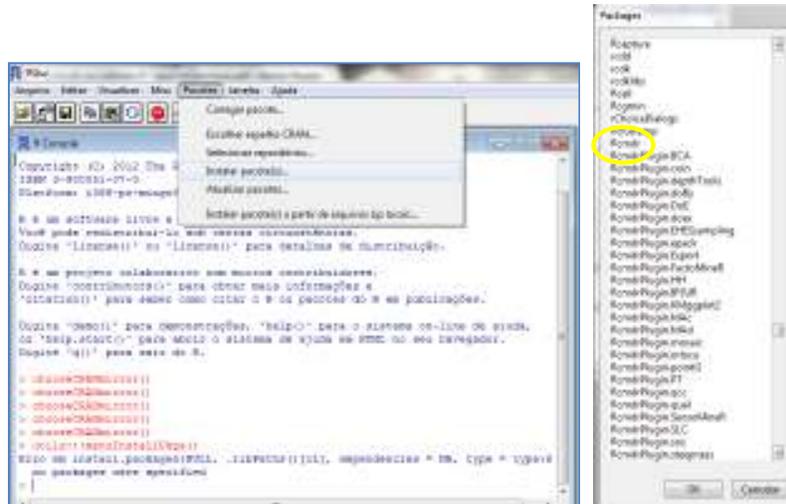
2. Trabalhando com o R

O R é um software que trabalha com pacotes. Os pacotes contêm funções que facilitam a execução das análises e a realização da programação. Ao instalarmos o R, alguns pacotes são instalados também. No entanto, com certeza serão necessários pacotes extras para realização das atividades. Esses pacotes podem ser instalados de dentro do R ou através do site www.r-project.org, onde também estão os manuais de instrução de alguns pacotes.

Para instalar os pacotes, a primeira coisa a se fazer é escolher de qual CRAN os pacotes serão baixados. Clique em **Pacotes** → **Escolher espelho CRAN** → **Brazil (SP1)**, mas pode escolher outro.



Agora, clique em **Pacotes** → **Instalar pacote(s)** → Escolher o pacote **Rcmdr**, que irá instalar o pacote “R Commander”



Ao clicar pela primeira vez em um pacote a ser instalado, o R perguntará se quer criar um diretório para servir como biblioteca de pacotes. Pode aceitar.

Esse pacote nada mais é do que uma interface gráfica (GUI – graphical user interface) para facilitar a vida de quem usa o R.

3. Importando dados para o R

Na grande maioria das vezes, as bases de dados não estão prontas para serem usadas na linguagem do R. Assim, será necessário importar as bases de dados para dentro do R.

Vamos fazer isso de duas formas:

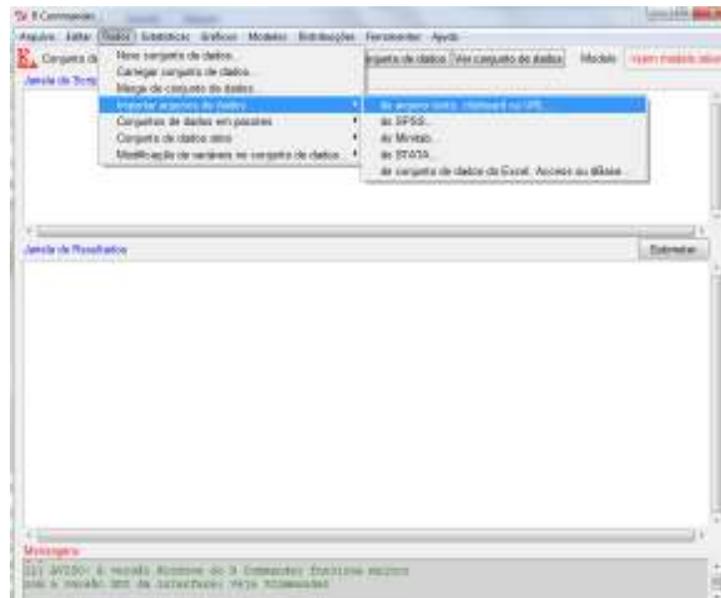
a) através do pacote **Rcmdr**

A título de exemplo, vamos importar a base de dados **ceosall**, pertencente ao livro “*Introdução a Econometria – Uma abordagem moderna*” de J. Wooldridge.

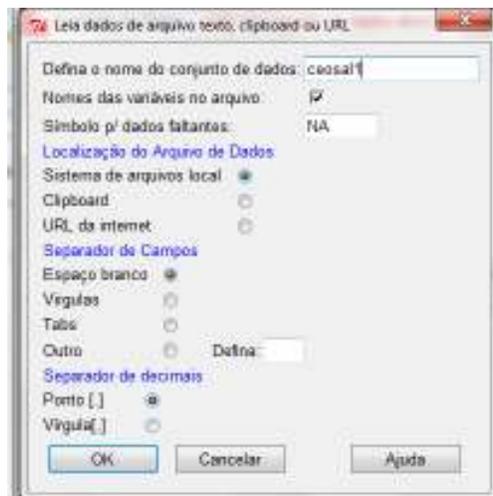
difícil. É necessário verificar se a planilha de dados possui ou não o nome das variáveis usadas na primeira linha e como os valores que faltam na base de dados são representados. Os arquivos **.raw** do Woodridge podem ser abertos diretamente como texto. No entanto, você pode importar diretamente do Excel, se o banco estiver como **.xls**, apesar de ser recomendável transformar os dados em texto (separado por tabulações no Excel)

No R, vamos carregar o pacote **Rcmdr**, clicando em **Pacotes** → **Carregar pacote** → **Rcmdr**

Abrirá a janela do R Commander.



Clique em **Dados**→**Importar arquivos de dados**→**de arquivo texto, clipboard ou URL**. Irá aparecer a seguinte tela:



Nessa tela:

- 1) Iremos definir o nome da base de dados: **ceosal1**
- 2) Deixamos marcada a opção nomes das variáveis no arquivo (pois neste caso a base de dados contém os nomes das variáveis na primeira linha)
- 3) No campo “símbolo para dados faltantes, deixamos a opção NA. Note que outros arquivos podem vir preenchidas por pontos ou outro caracter, quando se tratam de dados

faltantes. Assim, antes de escolher a opção NA, devemos ver como estão os dados na planilha. Por exemplo, os dados do arquivo **GPA3**, os valores que faltam são marcados por pontos. Neste caso, marcamos “símbolo para dados faltantes” com um ponto.

4) Definir separador de colunas como “espaço”

5) Definir separador decimal como “.”

6) Clicar em **OK**, e escolher a base de dados no computador.

Então o **Rcmdr** apresentará o comando completo. Basta copiar e colar no Rgui,

```
ceosal1 <-read.table("C:/caminho de diretorio/CEOSAL1.raw", header=TRUE, sep="",  
na.strings="NA", dec=".", strip.white=TRUE)
```

OBS: Ao copiar qualquer comando não copiem o sinal >, pois senão o R não reconhecerá o comando. Comecem exatamente a partir da primeira letra.

O R é um software que trabalha com a geração de objetos. Assim, **ceosal1** é o nome do objeto criado. Tudo que se escreve antes do símbolo “<-“ refere-se ao nome do objeto. Esse nome é sensível a letras maiúsculas. Assim, o R não reconhecerá o objeto **Ceosal1**, se assim for digitado

b) a segunda opção de importação é através da digitação do comando abaixo diretamente na janela Rgui (Rconsole) ou rodar esse comando de dentro de um script (falaremos sobre scripts depois)

```
ceosal1 <-read.table("C:/caminho de diretorio/CEOSAL1.raw", header=TRUE, sep="",  
na.strings="NA", dec=".", strip.white=TRUE)
```

4. Trabalhando com os dados no R

Uma vez os dados estando dentro do R, é muito útil usar a função **attach**. Essa função fixa a base de dados e permite acessar as variáveis do banco de dados pelo nome, sem a necessidade de especificar, a cada comando dado, de qual base estamos falando.

attach(ceosal1)

O comando abaixo mostrará o nome das variáveis contidas no banco de dados ceosal1

names(ceosal1)

Se não usarmos este comando **attach**, deve-se explicitar qual é a base de dados usada.

Por exemplo, se quiser calcular alguma estatística descritiva de uma das variáveis do banco de dados:

summary(salary), se o banco estiver “attachado”

ou

summary(ceosal[, 1]), se o banco não estiver “attachado”, pois estaremos informando ao R que queremos as estatísticas descritivas da variável que está na primeira coluna do banco de dados **ceosal1**, no caso a variável *salary*

Se simplesmente for digitado o nome do banco de dados, o R automaticamente mostra todas as informações de todas as variáveis.

ceosal1

Mas, se queremos editar os dados:

fix(ceosal1)

Em uma nova janela aparecerão todos os dados, onde os nomes das variáveis podem ser alterados. Basta clicar no nome da variável que se deseja mudar e digitar na caixa de diálogo “nome da variável” o novo nome atribuído à variável. Feito isso, basta fechar a janela e o R reconhecerá os novos nomes das variáveis.

Obs: Quando houver necessidade de selecionar um certo número de linhas, basta:

ceosal1[1:50,]

Neste caso estamos selecionando as linhas de 1 a 50 e todas as colunas do banco de dados.

5. Estimando uma regressão simples no R

Vamos usar o banco de dados **ceosal1** para estimar uma equação no R, por Mínimos Quadrados Ordinários

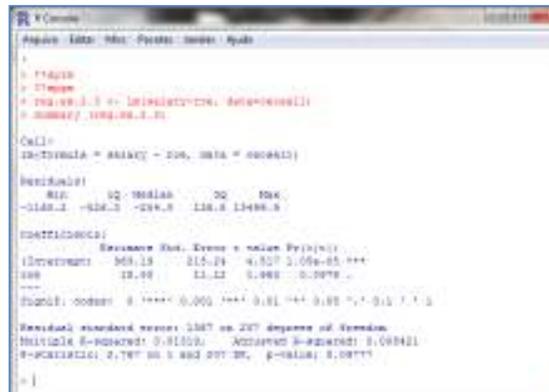
reg.ex2.3 <- lm(salary~roe)

reg.ex2.3: é um objeto que armazenará todos os resultados dessa regressão. Assim, todas as vezes que nos reportarmos aos resultados desta regressão, não é necessário estimá-la novamente. Note que podemos atribuir qualquer nome a uma regressão, mas devemos ter o cuidado de atribuir nomes diferentes, quando estimarmos diversos modelos.

lm(): se refere ao modelo a ser estimado por Mínimos Quadrados Ordinários (MQO). A variável que vem antes do sinal ~ é a variável dependente. As que vem depois desse símbolo são as variáveis explicativas.

O comando para que sejam visualizados os resultados da regressão é:

summary(reg.ex2.3)



```
R Console
> # Regressão
> # Formula: salary ~ exp, data = roe
> # summary(roe.ex2.3)

Call:
lm(formula = salary ~ exp, data = roe)

Residuals:
    1.00  -1.00  -1.00  -1.00  -1.00  -1.00  -1.00  -1.00  -1.00  -1.00

Coefficients:
(Intercept)  65.15  215.24  4.827  1.05e-05 ***
exp           12.98   11.12   1.166   0.2978 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.057 on 297 degrees of freedom
Multiple R-squared:  0.9102, Adjusted R-squared:  0.9091
F-statistic: 2.787 on 1 and 297 DF, p-value: 0.00077
```

Alguns resultados importantes

names(reg.ex2.3)

- [1] "coefficients" "residuals" "effects" "rank" "fitted.values"
- [6] "assign" "qr" "df.residual" "xlevels"
- [10] "call" "terms" "model"

Alguns comandos:

reg.ex.2.3\$coefficients : retorna os coeficientes da regressão

reg.ex.2.3\$resid: retorna os resíduos da regressão

reg.ex.2.3\$fitted: retorna os valores ajustados da regressão

Podemos criar objetos contendo os valores dos resíduos e os valores ajustados:

resid1<- reg.ex.2.3\$resid

fitted1<- reg.ex.2.3\$fitted

Grau de ajustamento da reta da regressão:

plot(roe, salary)

abline(reg.ex.2.3)