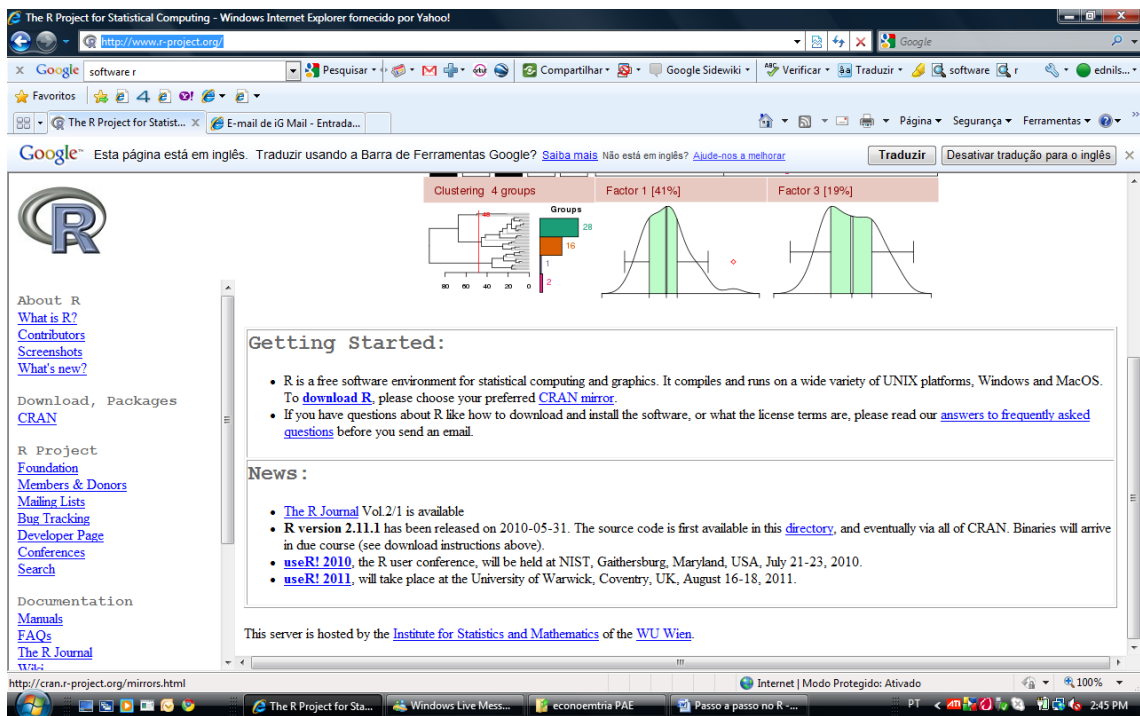


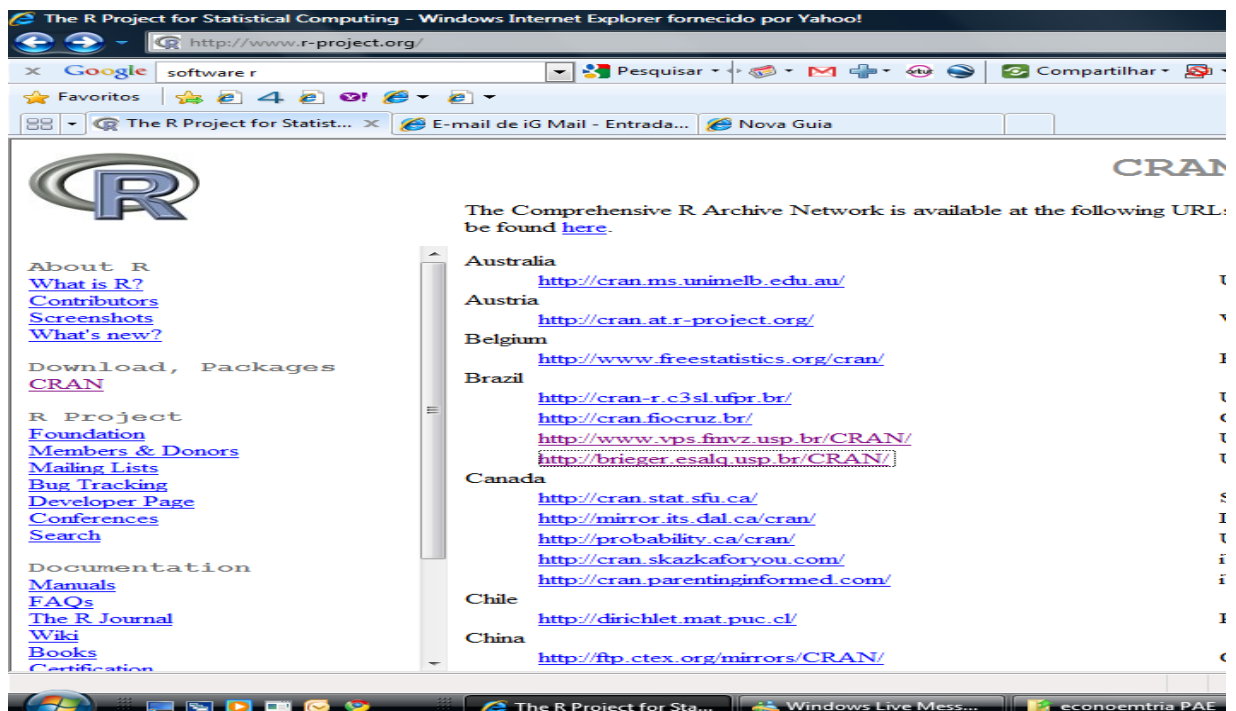
Primeiros passos no software R

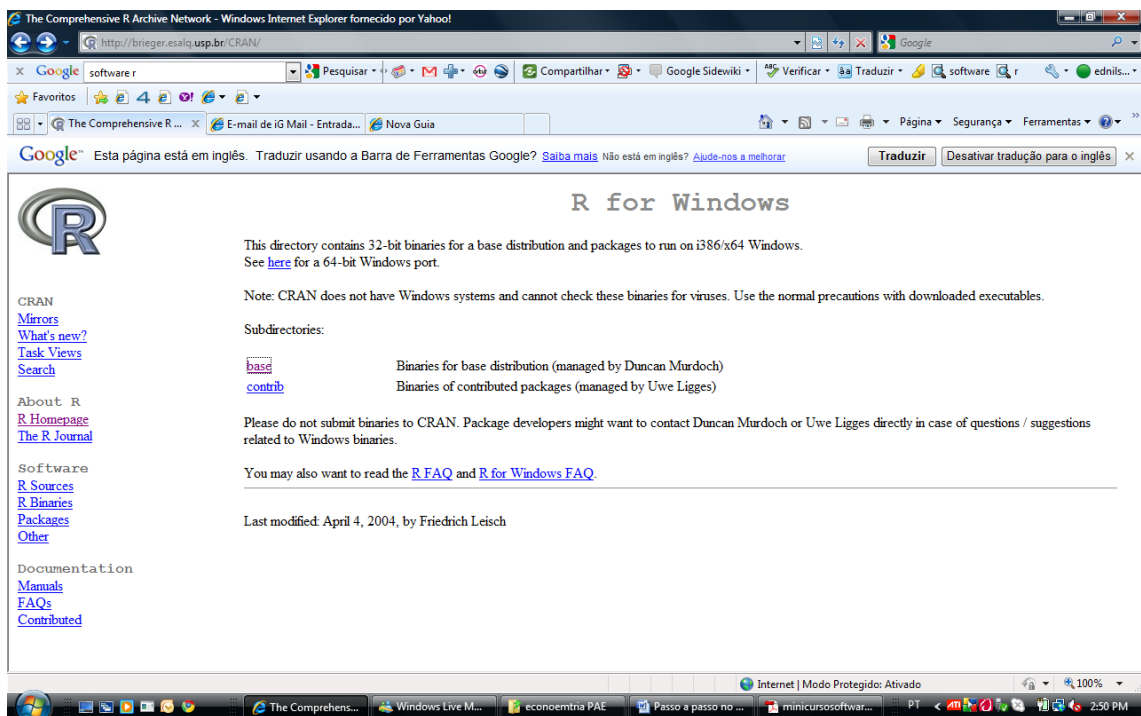
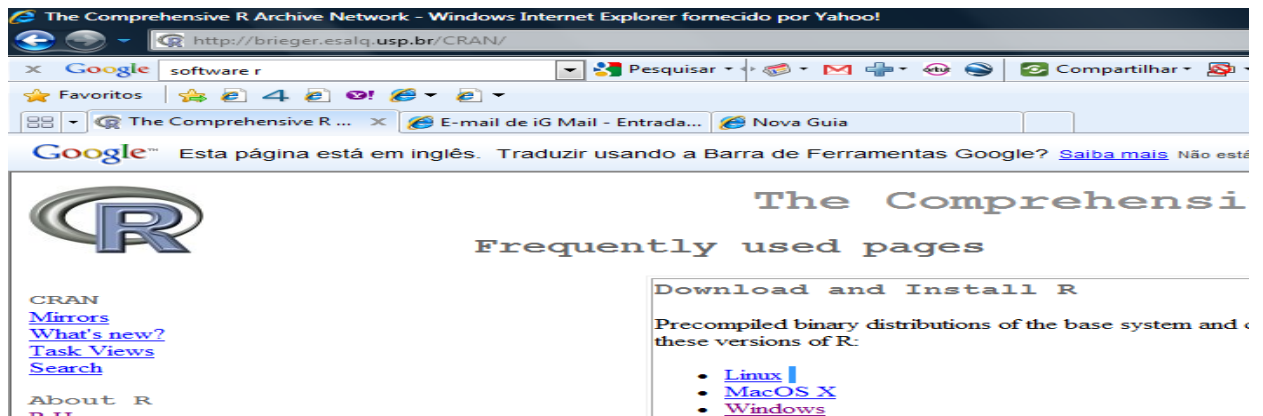
- O software pode ser encontrado na página do R projet:

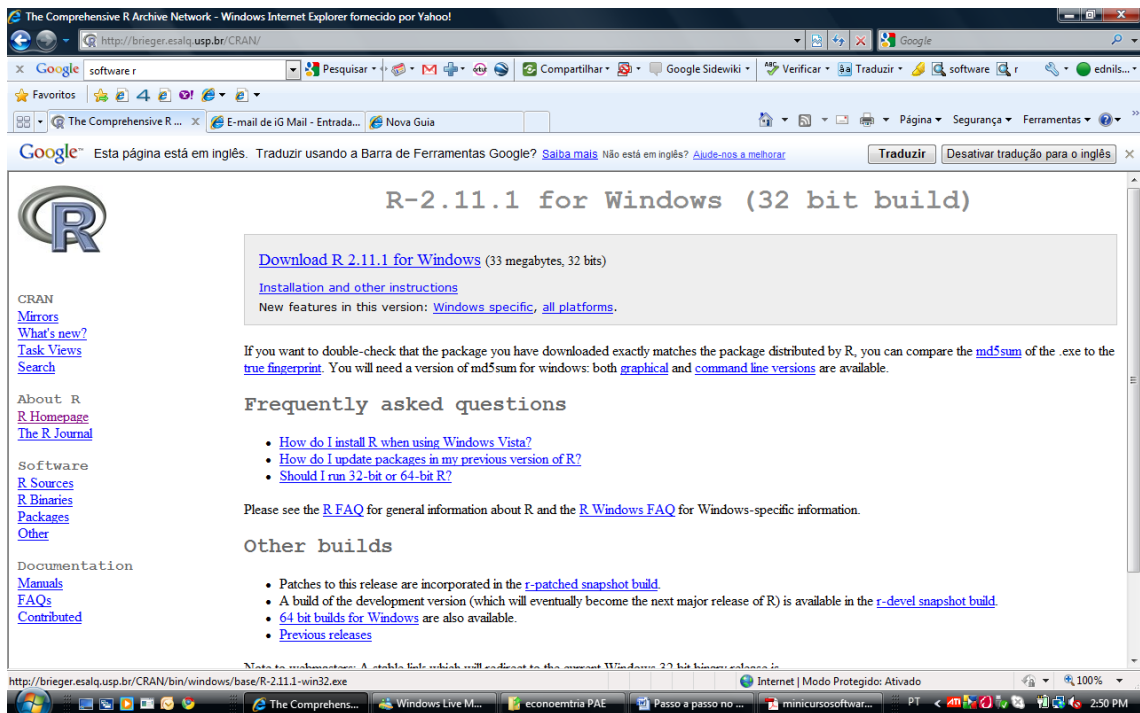
<http://www.r-project.org/>



- O próximo passo é escolher o CRAN:







Depois basta fazer o dowload e instalar.

**escolher a opção instalação completa...

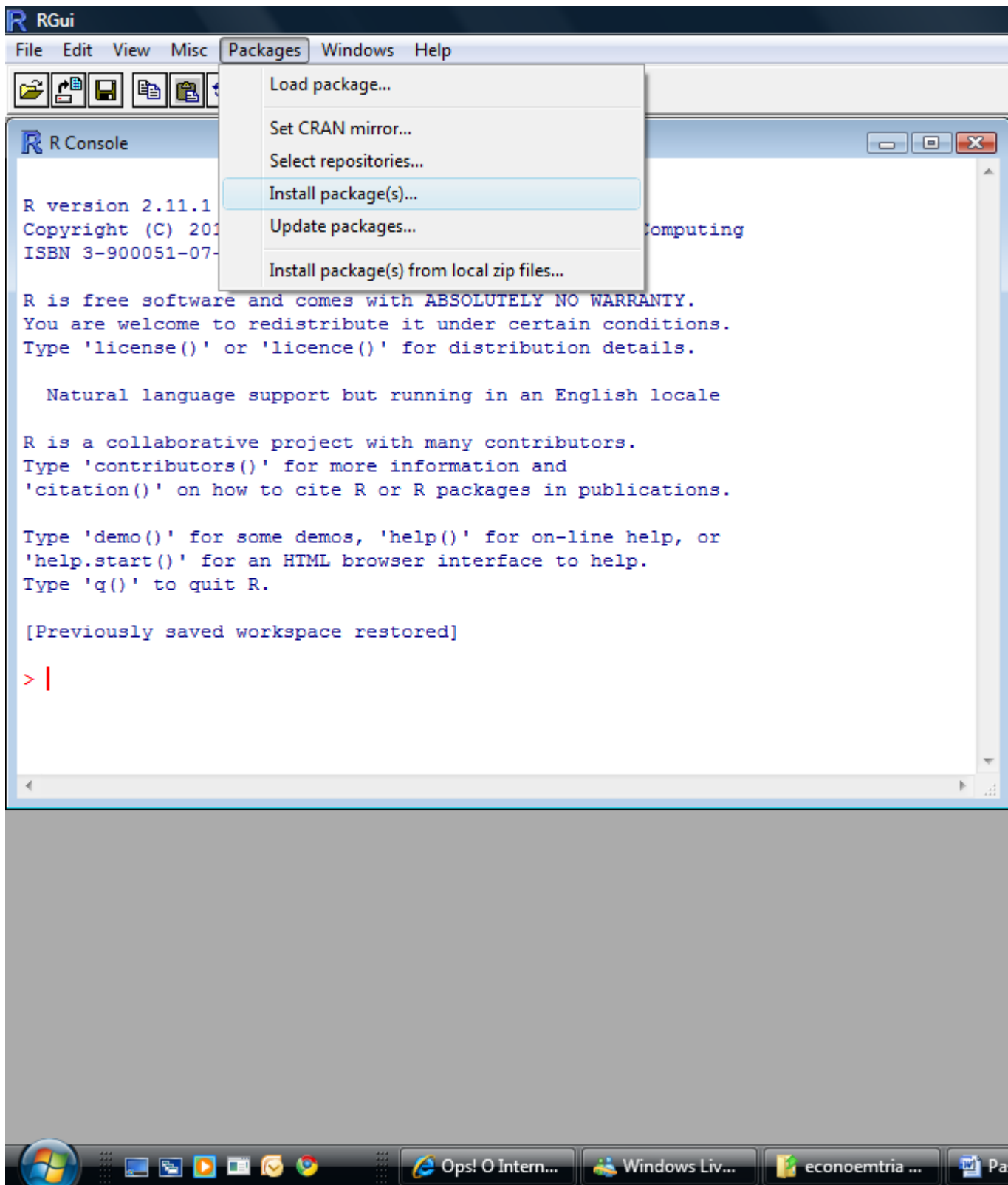
**escolher opção de inicialização padrão...

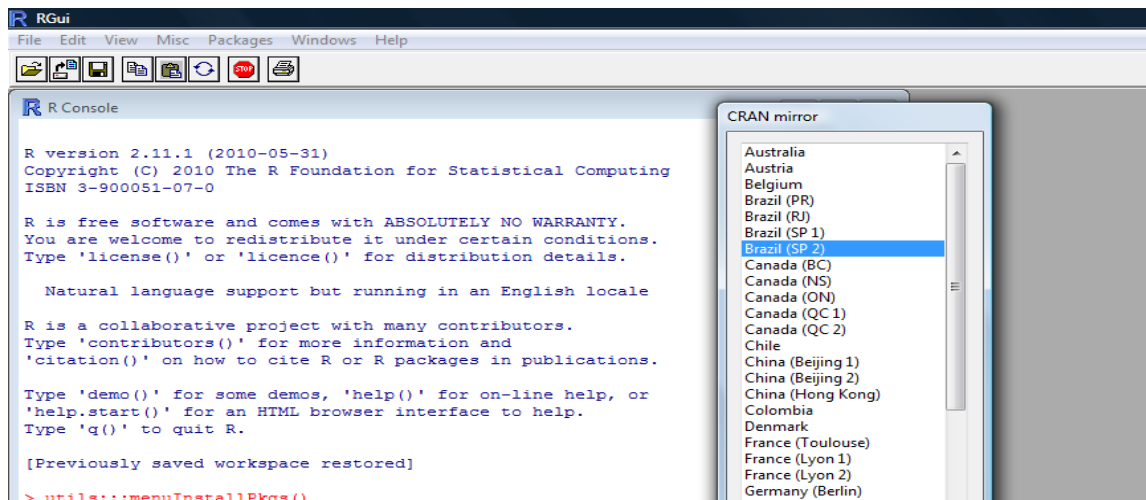
Pacotes do R

No R, existe uma grande diversidade de pacotes. Estes pacotes contêm um conjunto de funções que permitem ou facilitam a realização das análises estatísticas, além de possuírem ajuda para suas funções e demonstrações de execução.

Ao instalar o R, apenas alguns pacotes vêm juntos, os quais são essenciais para o funcionamento do programa, os quais denominamos de módulo ou pacote básico, além de muitos poderem servir de base e pré-requisito para o funcionamento de outros pacotes. Os pacotes extras podem ser encontrados com base no próprio programa ou pelo site, www.r-project.org, ressaltando que em ambos os casos, o usuário deve estar conectado na internet.

Mas vale a pena consultar o site, porque certos pacotes possuem manual de instrução.

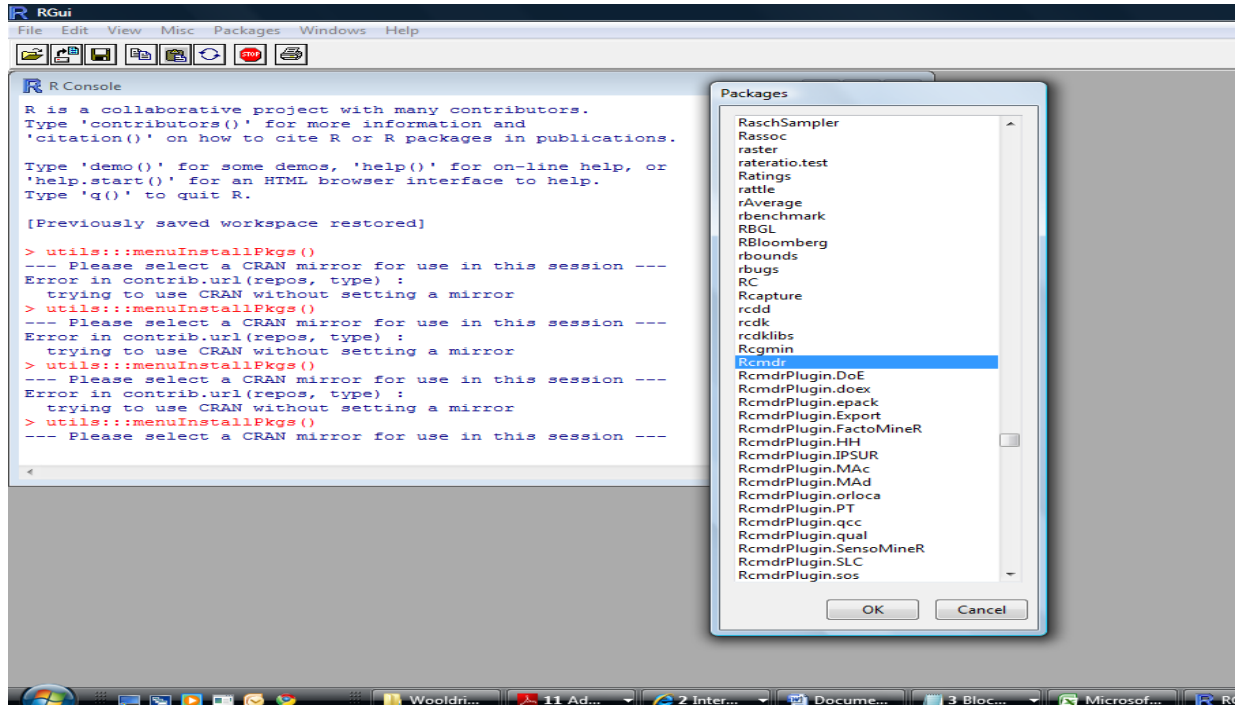




E depois basta escolher quais pacotes instalar.

Por acaso, escolhi o Cran Brazil(SP 2), pode-se escolher qualquer Cran para instalar os pacotes.

Pacote Rcmdr



Basta clicar em ok, que o pacote será instalado no R.

****Quando clicarem a primeira vez em Rcmdr, o programa lhes informará que faltam pacotes do Rcmdr. Basta confirmar que desejam a instalação destes pacotes que faltam.**

Importando dados e estatísticas

2.1- Importando dados.

Para importar os dados para o R podemos fazer de uma maneira fácil e uma difícil. Se optarmos importar os dados via pacote Rcmdr, o processo é feito de uma maneira simples.

Porém, antes disto precisamos verificar a planilha que contém os dados. Mais precisamente é importante verificar se a planilha contém os nomes das variáveis na primeira linha e como os valores que faltam na base de dados são representados. No caso em questão, como estamos trabalhando com os dados do Wooldridge, sabemos que a planilha com os dados não apresentam os nomes das variáveis.

Para importar os dados, é recomendável transformar os dados de excel para um arquivo de texto. Para isso, basta abrir a série em questão no excel, depois salvarmos como arquivo texto:

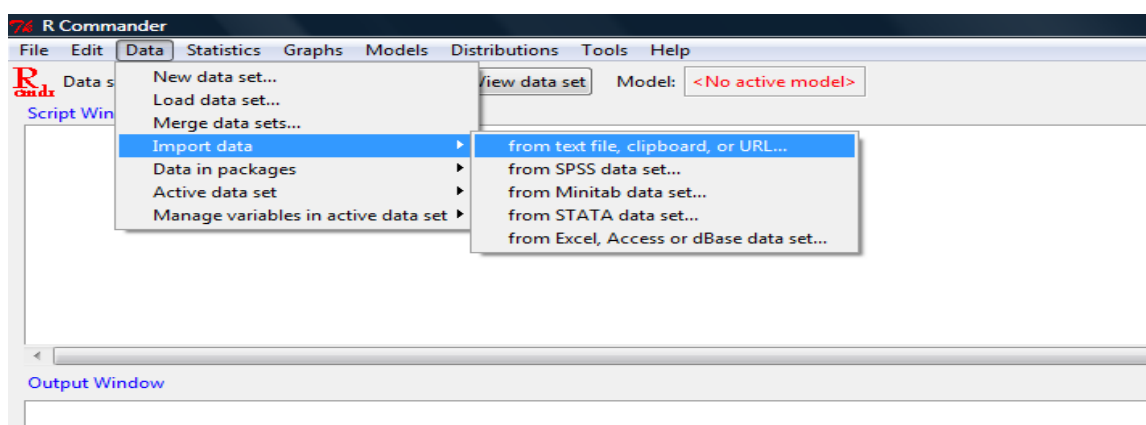
:salvar como\pasta de trabalho do excel\ texto (separado por tabulações)\

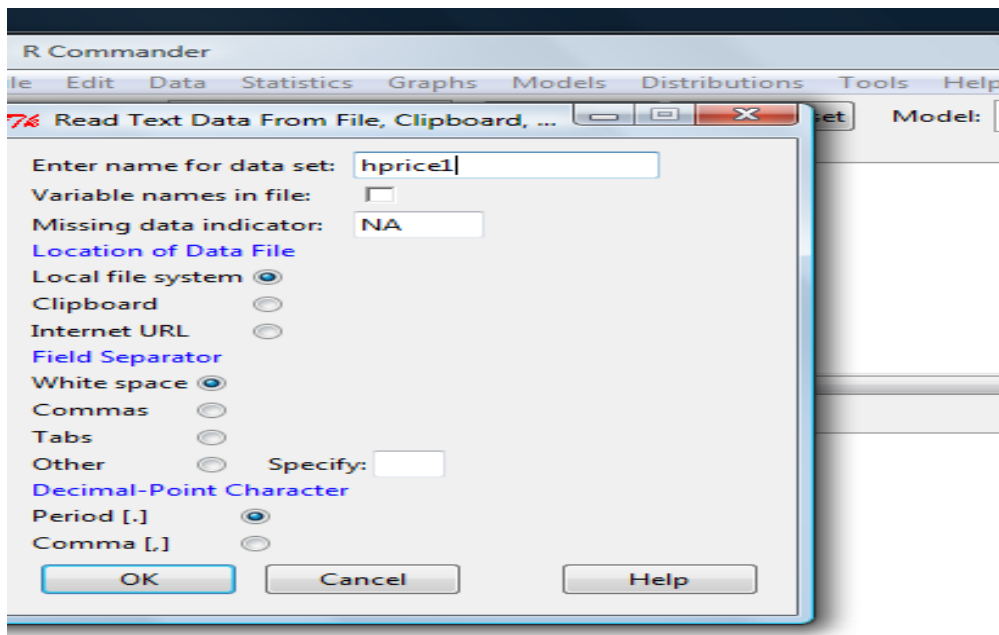
escolher a pasta onde os dados serão salvos.

Feito isto, basta fazer o seguinte caminho no R:

Packages\load\Rcmdr\

Abriá a janela do R Commander.





1º: Definimos o nome da variável: hprice1

2º: Deixamos desmarcada a opção variable names in file (pois neste caso a base de dados não contém os nomes das variáveis na primeira linha)

3º: Na opção missing data indicator deixamos a opção NA,

(no entanto, algumas planilhas costumam vir preenchidas por pontos ou outro caracter, quando falta alguma observação na amostra. Antes de escolher a opção NA, devemos ver como estão os dados na planilha. Por exemplo, os dados ao arquivo GPA3, os valores que faltam são marcados por pontos. Neste caso, marcamos missing data indicator com um ponto.

4º: O restante deixa como aparece na janela sendo que field separator indica que as colunas são separadas por espaços.

5º Depois basta clicar em ok, e escolher a base de dados no computador.

Então o Rcmdr apresentará o caminho e depois disto basta copiar e colar no Rgui, e então ficará assim:

```
> hprice1 <- read.table("C:/Users/Ednilson/Documents/Wool/hprice1.txt",
+ header=FALSE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
```

**quando forem copiar qualquer comando não copiem o sinal >, pois senão o R não reconhecerá o comando. Comecem exatamente a partir da primeira letra.

Onde:

> hprice1 < :é o nome do objeto criado. O R só reconhecerá hprice1 escrito desta forma, se por descuido for digitado Hprice1, o R não o reconhecerá.

- read.table: é o comando para importar os dados. Na verdade este comando já vem instalado no R.

Os demais comandos são os mesmos descritos anteriormente.

Feito isso, usamos a função

```
attach(hprice1)
```

```
names(hprice1)
```

Usamos a função attach para tornar as variáveis acessíveis pelo nome, sem termos que especificar qual é a base de dados que estamos trabalhando.

Se não usarmos este comando, quando formos estimar uma regressão, devemos explicitar qual é a base de dados:

```
Reg <- lm(y~x, data = nome da série)
```

Quando digitamos o nome da série, o R automaticamente mostra todos os dados.

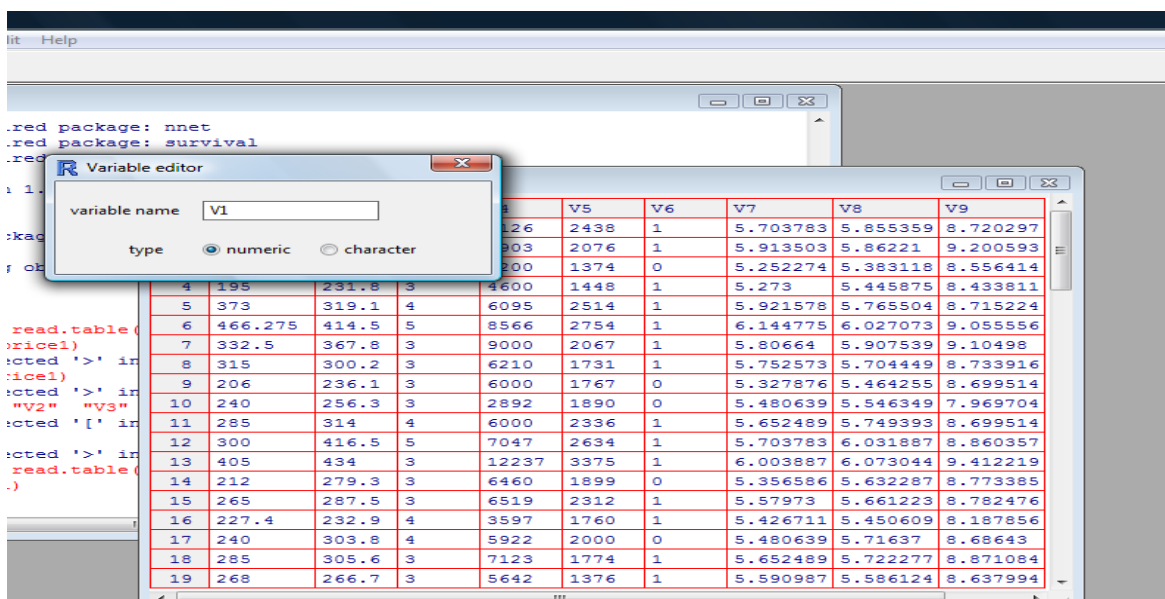
```
hprice1
```

Mas, se queremos editar os dados:

```
Fix(hprice1)
```

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 variable name o nome atribuído à variável, como mostra a figura abaixo:



Feito isso, basta fechar a janela e o R reconhecerá os novos nomes das variáveis.

Ou então pode-se nomear as variáveis da seguinte forma:

Sabemos que V1 se refere a price, e V3 ao número de quartos, então:

```
price <- V1
```

```
bdrms <- V3
```

Assim, foram criados os objetos price e bdrms, que se referem aos valores de V1 e V3 respectivamente. Mas para fazermos isso, temo que dar o comando attach(nome da série)

Se a função attach() não for acionada, o procedimento é o que se segue:

```
price <- hprice1[, 1]
```

pois price está na primeira coluna da base de dados.

```
**[linhas, colunas]
```

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

```
hprice1[1:50,]
```

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

ANÁLISE ESTATÍSTICA

Estatística Descritiva das variáveis: Este comando apresenta os valores da média, mediana, o valor máximo e mínimo de cada variável.

```
summary(price)
```

*fornece os valores para V1

```
summary(hprice1)
```

*fornece os valores para todas as variáveis do banco de dados em questão

Correlação:

cor(price, bdrms) : para um par de variáveis

cor(hprice1): para todas as variáveis da série

Variância:

teste entre duas variáveis:

```
var.test(price,bdrms)
```

Estimando uma regressão:

```
> regex <- lm(price~bdrms)
```

regex < é um objeto criado para todas as vezes que nos reportarmos aos resultados desta regressão, não seja necessário estimá-la novamente. Podemos atribuir qualquer nome a regressão. Mas devemos ter o cuidado de atribuir nomes diferentes, quando estimarmos diversos modelos em um mesmo workspace

- lm(): se refere ao modelo linear que será estimado. Portanto, para outros modelos de regressão, obviamente, o comando será outro.

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

```
summary(regex)
```

Alguns resultados importantes

```
names(regex)
```

```
[1] "coefficients" "residuals"    "effects"      "rank"         "fitted.values"
```

```
[6] "assign"       "qr"           "df.residual"  "xlevels"
```

```
[10] "call"         "terms"        "model"
```

Alguns comandos:

regex\$coefficients : retorna os coeficientes da regressão

regex\$resid: retorna os resíduos da regressão

regex\$fitted: retorna os valores ajustados da regressão

** é comum nomear os resíduos como **uhat** e os valores ajustados como **yhat**, já que se referem a \hat{u} e \hat{y} .

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

```
resid1<- regex$resid
```

```
fitted1<- regex$fitted
```

Para encontrar a matriz de var/cov

```
vcov(regex)
```

Para encontrar o intervalo de confiança do valor dos coeficientes:

```
> confint(regex, level=.95): nome da regressão e o nível de significância.
```

Análise gráfica

Histograma

```
hist(nome da variável)
```

Se quisermos o histograma colorido:

```
hist(nome da variável, col= "red")
```

Irá gerar um histograma vermelho.

Podemos ter também o histograma dos resíduos:

```
hist(regex$resid)
```

Tipos de gráficos:

1- Barras;

```
barplot(nome da variável)
```

2- Bolhas:

```
plot(nome da variável)
```

```
plot(nome da regressão)
```

Se quisermos verificar o grau de ajustamento da reta da regressão:

```
plot(bdrms, price)
```

**primeiro vem a variável independente e depois a dependente.

`abline(regex)`

******cria a reta da regressão