Econometria I em R

Paloma Vaissman Uribe

19/08/2016

Ler conjunto de dados

O tutorial em formato de script do R pode ser encontrado na página do Professor Hedibert Lopes: http://hedibert.org/wp-content/uploads/2016/08/.

- Existem várias formas de abrir um conjunto de dados no R, dependendo do formato do arquivo e da origem.
- Por exemplo, para fazer *upload* de um arquivo .txt que está numa página da internet pode-se digitar o seguinte comando no *console* do R:

```
data1 = read.table("http://hedibert.org/wp-content/uploads/2016/02/wage.txt",header=TRUE)
```

- Note que o argumento **header=TRUE** faz com que a primeira linha do arquivo seja lida como sendo o título das colunas/ variáveis.
- Voce pode verificar todos os argumentos de uma função do R digitando:

```
?read.table
```

- Uma outra opção é salvar o arquivo no seu computador e ler os dados acessando o diretório indicado:

```
setwd("~/Desktop/EconometriaI_2016_02")
data2 = read.table("wage.txt",header=TRUE)
data1 == data2 #compara arquivos
```

- Note que dependendo da extensão do arquivo, deve-se usar outro comando, por exemplo:

```
setwd("~/Desktop/EconometriaI_2016_02")
data3 = read.csv("wage.csv",header=TRUE)
data3 == data2 #compara arquivos
```

- Sempre que desejar, pode-se ler o conjunto de dados utilizando o botão **Import Dataset** do RStudio e seguir os passos. Há duas opções: importar arquivo do tipo texto (.txt,.csv) de um diretório ou arquivo de uma URL.

Manipulando um conjunto de dados

- Quando trata-se de uma matriz (o que pode ser verificado usando o comando **dim**, que retorna a dimensão do objeto), pode-se acessar variáveis digitando após o nome do conjunto de dados: [i,] para acessar a linha i, e [,j] para acessar a coluna j.

```
dim(data1) #dimensão
names(data1) #comando que retorna os nomes de todas as variáveis
data1[,1] #mostra os dados da variável wage (que está na primeira coluna)
data1[1:3,1] #mostra as três primeiras observações (linhas) de wage
```

- Uma outra forma de manipular um *dataframe* é usar o símbolo **$** para acessar as variáveis ou então usar o comando **attach** para carregar as variáveis pelo nome e daí usar para outros comandos.

```
salario = data1$wage #note que é criado um valor, não um conjunto de dados
dim(data1$wage) #dimensão de um valor é zero
length(data1$wage) #usa-se este comando para objetos que não são conjuntos de dados
data1$wage[1:3] #mostra as três primeiras observações de wage
attach(data1)
wage[1:3] #mostra três primeiras observações de wage
```

Criando dummies e variáveis categóricas

- Pode-se usar valores lógicos para criar variáveis binárias específicas:

```
attach(data1)
female==0
singleman=(female==0)&(married==0)
marriedman=(female==0)&(married==1)
singlewoman=(female==1)&(married==0)
marriedwoman=(female==1)&(married==1)
```

- E também construir variáveis categóricas (por exemplo classes de experiência):

```
attach(data1)
cat_exper = 0
cat_exper[exper<=5]=1
cat_exper[(exper>5)&(exper<=10)]=2
cat_exper[(exper>10)&(exper<=15)]=3
cat_exper[(exper>15)&(exper<=20)]=4
cat_exper[(exper>20)&(exper<=25)]=5
cat_exper[(exper>25)&(exper<=30)]=6
cat_exper[(exper>30)&(exper<=35)]=7
cat_exper[(exper>35)&(exper<=40)]=8
cat_exper[(exper>40)&(exper<=45)]=9
cat_exper[(exper>45)]=10
data1$cat_exper=cat_exper #adiciona a nova variável ao conjunto de dados
```

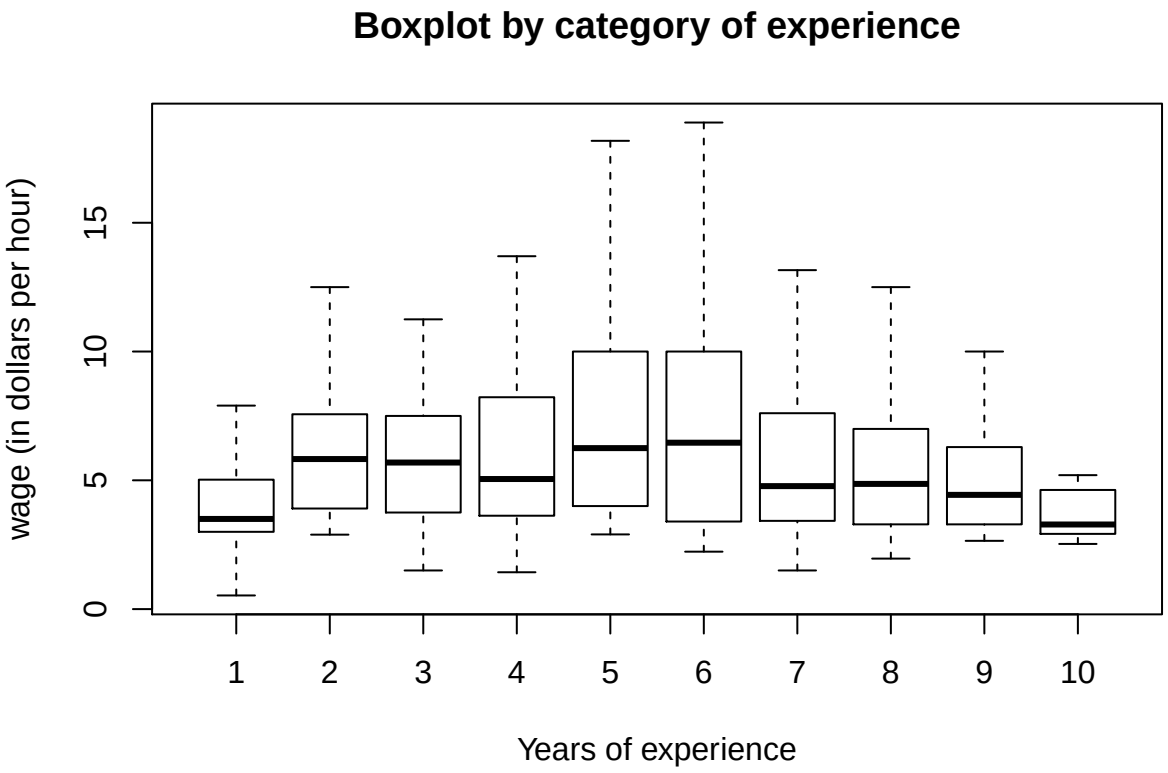
- Pode-se fazer um boxplot do salário por categoria de experiência usando:

```
par(mfrow=c(1,1))
boxplot(wage[exper<=5],
        wage[(exper>5)&(exper<=10)],
        wage[(exper>10)&(exper<=15)],
        wage[(exper>15)&(exper<=20)],
```

```
        wage[(exper>20)&(exper<=25)],
        wage[(exper>25)&(exper<=30)],
        wage[(exper>30)&(exper<=35)],
        wage[(exper>35)&(exper<=40)],
        wage[(exper>40)&(exper<=45)],
        wage[(exper>45)],
        names=c("<=5","(5,10]","(10,15]","(15,20]","(20,25]",
                "(25,30]","(30,35]","(35,40]","(40,45]",">45"),
        xlab="Years of experience",ylab="wage (in dollars per hour)",outline=FALSE)
```

- Ou usando a variável categórica criada anteriormente:

```
boxplot(wage~cat_exper,main="Boxplot by category of experience",
        xlab="Years of experience",ylab="wage (in dollars per hour)",outline=FALSE)
```



- Perceba aqui os argumentos usados: **main** (título do gráfico), **xlab** e **ylab** (título dos eixos x e y), e **outline** (não plota os valores extremos ou *outliers*).

Fazendo gráficos para verificar associação entre variáveis

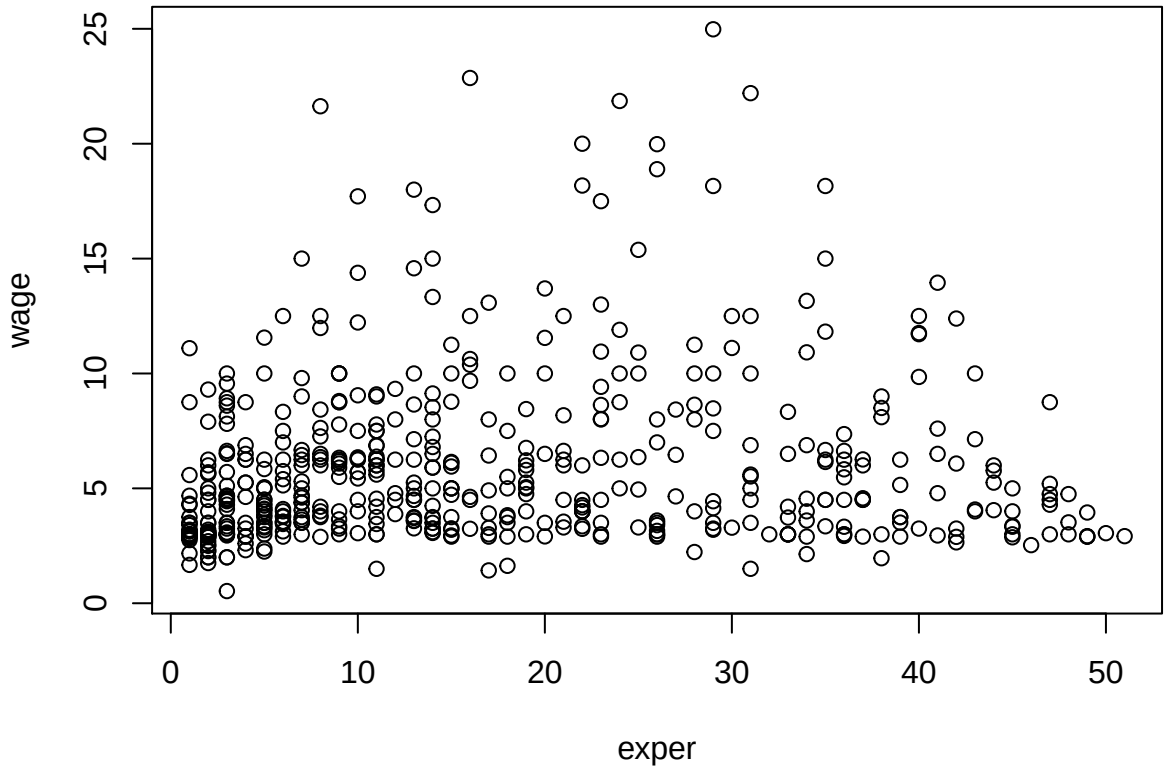
- Um primeiro passo para verificar se existe relação linear entre as variáveis é fazer um gráfico de dispersão usando a função **plot** e calcular a correlação linear através do comando **cor**:

```
attach(data1)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     cat_exper

## The following objects are masked from data1 (pos = 3):
##
##     educ, exper, female, married, wage
```

```
plot(exper,wage)
```




```
cor(exper,wage)
```

```
## [1] 0.1129034
```

Como rodar uma regressão linear no R:

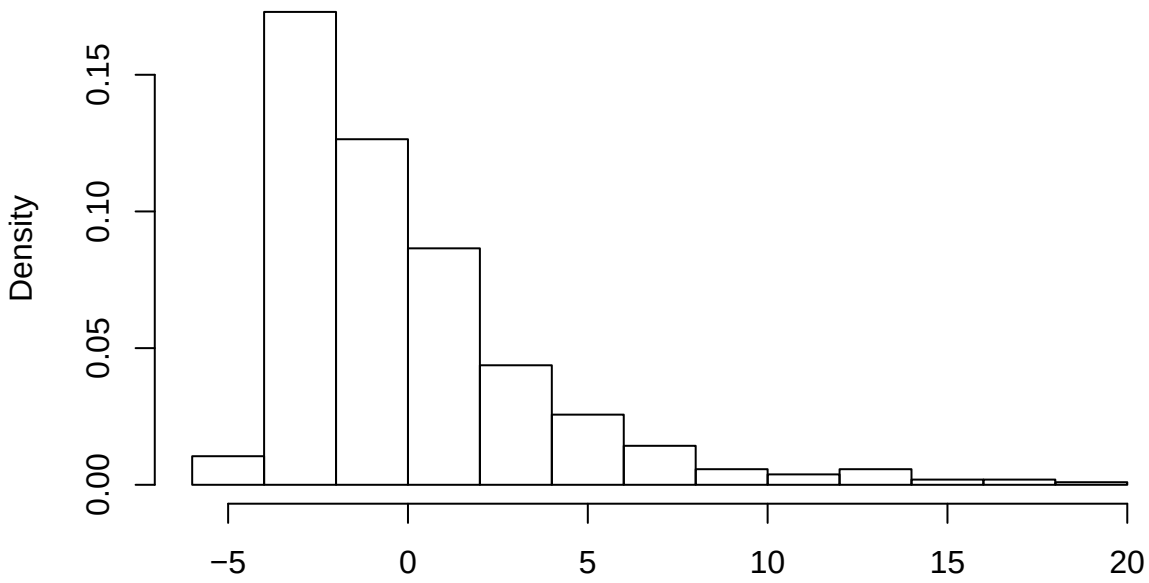
```
reg = lm(wage ~ exper)
sum_reg = summary(reg)
sum_reg
```

```
##
## Call:
## lm(formula = wage ~ exper)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.936 -2.458 -1.112  1.077 18.716
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.37331    0.25699  20.908  < 2e-16 ***
## exper        0.03072    0.01181   2.601  0.00955 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.673 on 524 degrees of freedom
## Multiple R-squared:  0.01275,    Adjusted R-squared:  0.01086
## F-statistic: 6.766 on 1 and 524 DF,  p-value: 0.009555
```

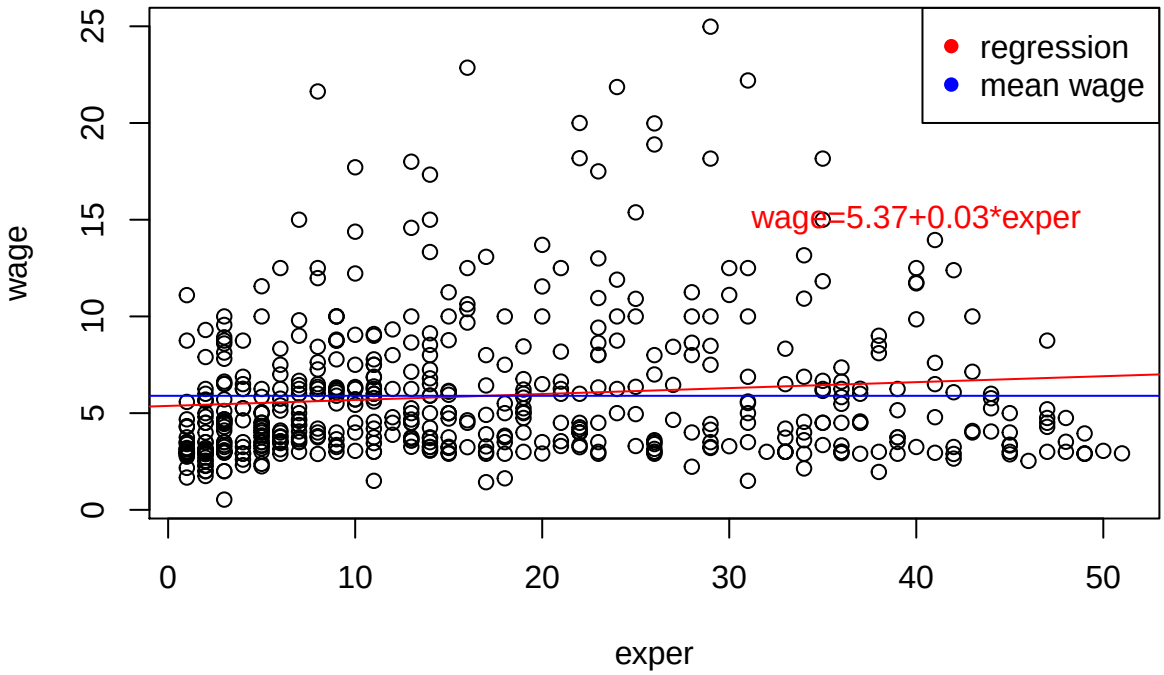
- Perceba aqui que a função **lm** roda a regressão linear, sendo seu resultado uma lista com coeficientes, resíduos, valores previstos, dentre outros. Já o comando **summary** pode ser usado para guardar uma lista com mais itens, e deve se referir à regressão já estimada usando a função **lm**. O comando **summary** gera uma lista que contém os resíduos, os coeficientes, a estimativa da variância do erro, o coeficiente de determinação R2, o R2 ajustado e a estatística F.
- Note que para guardar essas listas de itens gerados deve-se dar um nome para a regressão e/ou sumário da regressão. Feito isso, é sempre possível acessar qualquer item da lista usando o símbolo **$**, por exemplo, pode-se fazer um histograma dos resíduos:

```
hist(reg$residuals,freq = FALSE,xlab="",main="")
```



- Para fazer um gráfico da reta ajustada aos dados, pode-se usar:

```
plot(exper,wage)
legend("topright",legend=c("regression","mean wage"),col=c("red","blue"),pch=16)
abline(reg$coef,col="red")
abline(h=mean(wage),col="blue") #argumento h gera linhas horizontais
text(40,15,label=paste("wage=",round(reg$coef[1],2),
                       "+",round(reg$coef[2],2),"*exper",sep=""),col="red")
```



- Note que foi usada função **abline**, a função **text**, e a função **legend** que adicionam linhas retas, textos, e legendas, respectivamente. O comando **paste** foi usado para concatenar textos e escrever o valor previsto para a variável dependente.

Coeficiente de determinação de uma regressão

- Sabe-se que numa regressão simples $y_i = \beta_0 + \beta_1 x_i + \varepsilon$, o coeficiente de determinação de uma regressão ou R^2 é uma medida que denota o percentual da variação de y explicado pela variação de x. Ou seja:

$$R^2 = 1 - SQR/SQT = SQE/SQT,$$

sendo SQT a soma dos quadrados totais, ou $\sum_{i=1}^{n}(y_i - \bar{y})^2$, SQR a soma dos quadrados dos resíduos, ou $\sum_{i=1}^{n}(y_i - \hat{y})^2$, e SQE a soma dos quadrados explicada, ou $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$, em que $\bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$ e $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_i$ (valor previsto).

- Para a última regressão, o R^2 pode ser obtido no R calculando-se a fórmula acima ou usando o comando **summary**:

```
yhat = reg$coef[1]+reg$coef[2]*exper # calculando valores previstos
yhat2 = reg$fitted.values # também pode obter usando os resultados guardados
SQR = sum((wage - yhat)^2)
SQT = sum((wage-mean(wage))^2)
R2 = 1-SQR/SQT
R2_sum = sum_reg$r.squared # usando summary
c(R2,R2_sum) # mostra os resultados em vetor de ambas as alternativas
```

```
## [1] 0.01274719 0.01274719
```