Econometria I em R

Paloma Vaissman Uribe

09/09/2016

Lendo um 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
```

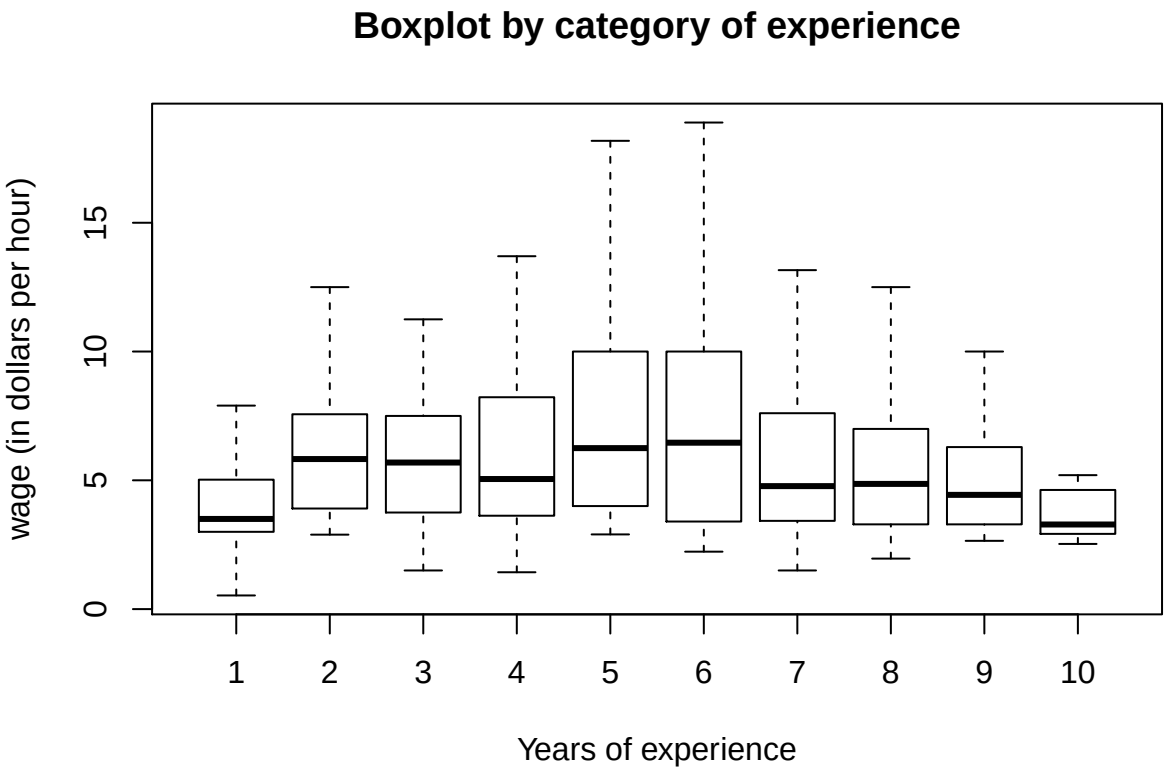
- Pode-se fazer um boxplot do salário por categoria de experiência usando:

```
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

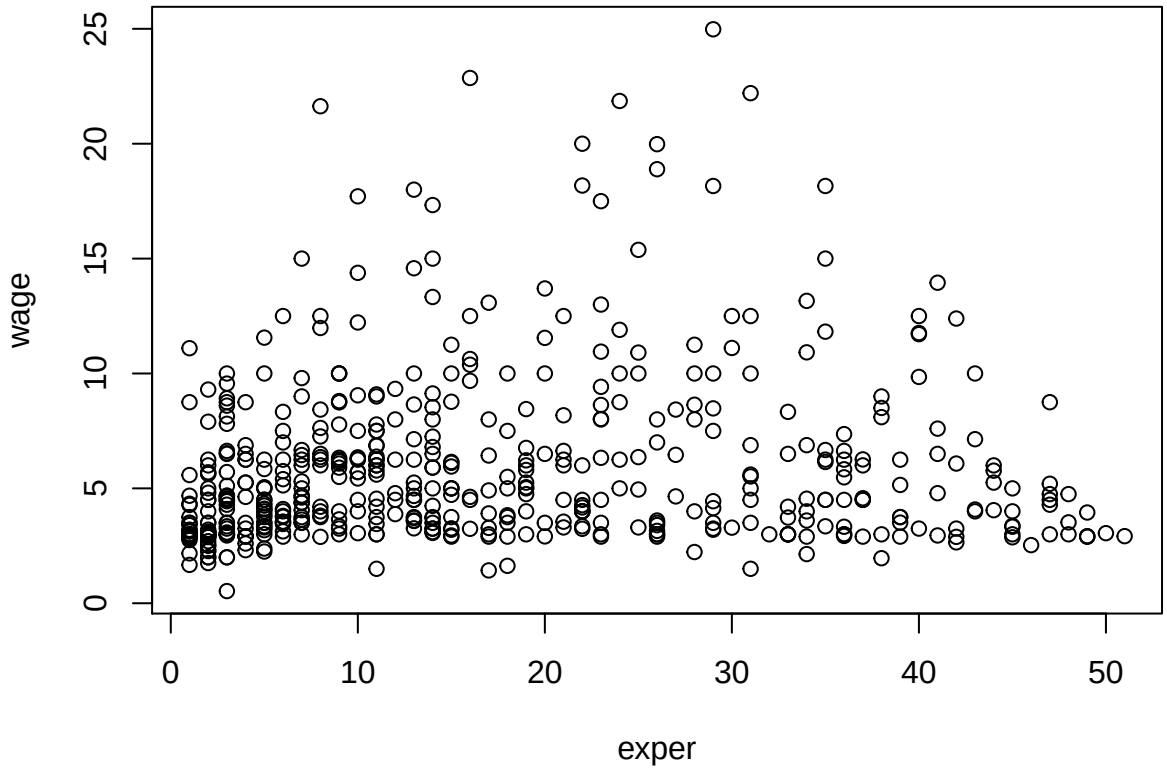
- Um primeiro passo para verificar se existe relação linear entre as variáveis é fazer um gráfico de dispersão usando a função **plot** e calcular a correlação linear através do comando **cor**:

```
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:

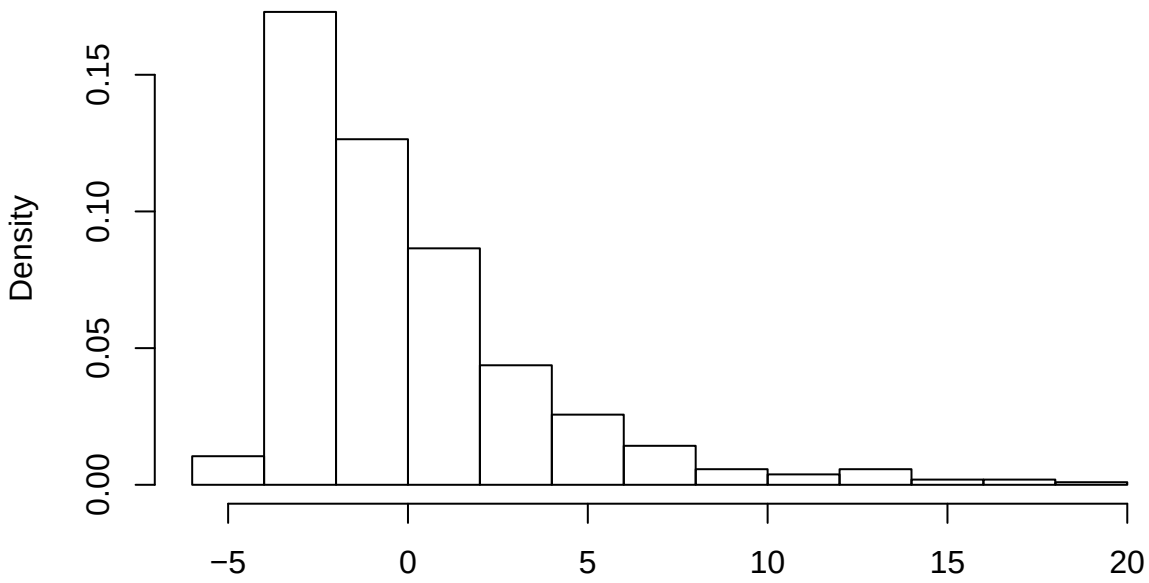
```
reg = lm(wage ~ exper)
sum_reg = summary(reg)
sum_reg
```

```
##
## 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
```

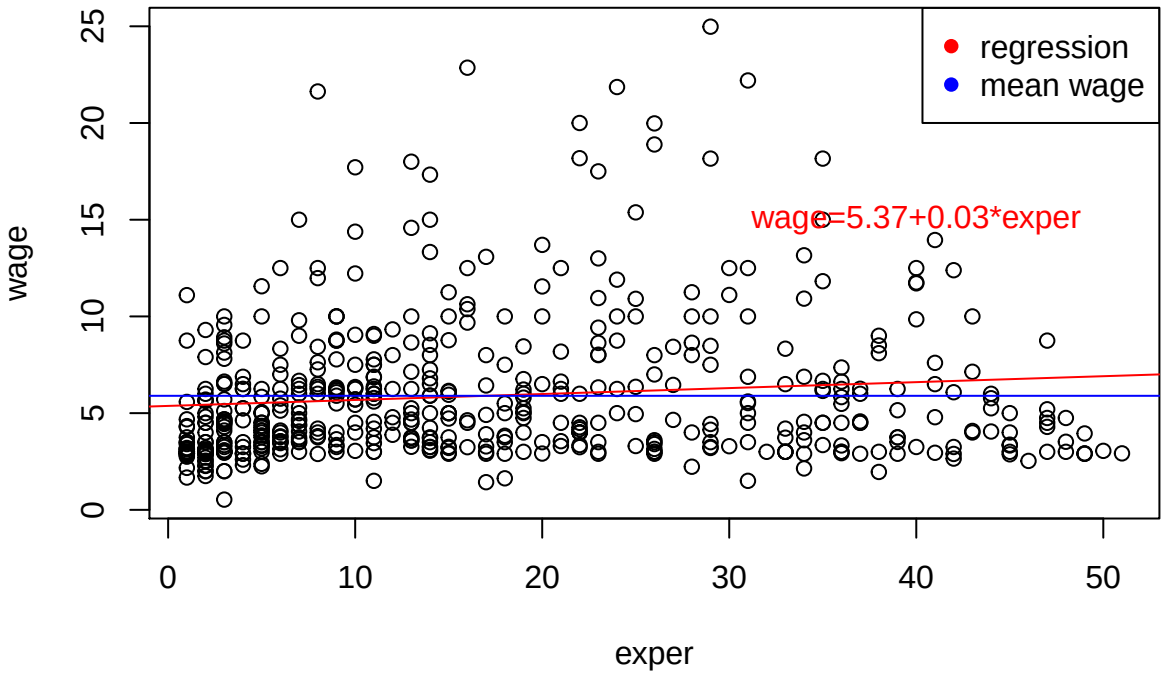
- Perceba aqui que a função **lm** roda a regressão linear, sendo seu resultado uma lista com coeficientes, resíduos, valores previstos, dentre outros. Já o comando **summary** pode ser usado para guardar uma lista com mais itens, e deve se referir à regressão já estimada usando a função **lm**. O comando **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
```

Testes de significância individual

- Para determinar se um coeficiente β_j, referente a um determinado regressor x_j, é estatisticamente significante, pode-se realizar um teste de hipótese: $H_0 : \beta_j = 0$, contra $H_1 : \beta_j \neq 0$.
- Também pode-se testar a hipótese de que o coeficiente β_j é maior (ou menor) que determinado valor. Nesse caso, faz-se um teste unicaudal: por exemplo, $H_0 : \beta_j = 0$, contra $H_1 : \beta_j > 0$.
- Em um modelo de regressão linear simples com intercepto, sob determinadas suposições (1. Linearidade nos parâmetros; 2. Amostragem aleatória; 3. Variação amostral no regressor; 4. Média condicional zer0; 5. Homocedasticidade; e 6. Normalidade dos erros), podemos testar a signficância individual do único regressor utilizando a estatística de teste $\frac{\hat{\beta}_1}{\hat{\sigma}_{\hat{\beta}_1}} \sim t_{n-2}$, em que $\hat{\sigma}_{\hat{\beta}_1}$ é o erro-padrão do estimador de mínimos quadrados de β_1, $\hat{\sigma}_{\hat{\beta}_1} = \frac{\hat{\sigma}}{\sqrt{SST_x}}$, sendo $\hat{\sigma}^2 = \frac{SSR}{n-2}$.
- Para testar hipóteses sobre o intercepto β_0, utiliza-se a estatística $\frac{\hat{\beta}_0}{\hat{\sigma}_{\hat{\beta}_0}} \sim t_{n-2}$, em que $\hat{\sigma}_{\hat{\beta}_0}$ é o erro-padrão do estimador de mínimos quadrados de β_0, $\hat{\sigma}_{\hat{\beta}_0} = \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{SST_x}}$.
- Ou seja, sob determinadas hipóteses, a estatística $\frac{\hat{\beta}_j - b}{\hat{\sigma}_{\hat{\beta}_j}}$ terá distribuição t-student com $n-2$ graus de liberdade, sendo que para o teste de significância individual $b = 0$.
- Na regressão linear múltipla com intercepto, a lógica é a mesma: sob suposições análogas, a estatística $\frac{\hat{\beta}_j - b}{\hat{\sigma}_{\hat{\beta}_j}}$ terá distribuição t-student com $n-k-1$ graus de liberdade, sendo k o número de regressores. As fórmulas das variâncias dos estimadores de mínimos quadrados de $\beta_j, j = 1, ..., k$ também dependerão do parâmetro $\sigma^2 = Var(\varepsilon|X)$. Sendo este desconhecido, será estimado usando $\hat{\sigma}^2 = \frac{SSR}{n-k-1}$. Feito isto, os desvios padrões dos estimadores das variâncias de $\hat{\beta}_j$ serão denominados erros-padrão.
- No R, é possível obter os erros padrões das estimativas utilizando o comando **summary**, acessando a matriz de coeficientes que fica guardada, em que as linhas correspondem à lista de regressores, e as colunas aos valores de estimativas, erros-padrão, estatística-t e p-valor, respectivamente. O índice a ser buscado vai depender do número de regressores. No exemplo anterior:

```
sum_reg$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 5.37330523 0.25699185 20.908465 4.830847e-71
## exper       0.03072187 0.01181106  2.601109 9.554854e-03
```

```
dim(sum_reg$coefficients) #dimensão da matriz de coeficientes
```

```
## [1] 2 4
```

```
sum_reg$coefficients[2,2] #erro padrão (coluna 2) do estimador de beta1 (linha 2)
```

```
## [1] 0.01181106
```

```
sum_reg$coefficients[2,1]/sum_reg$coefficients[2,2] #equivale à estatística-t da coluna 4
```

```
## [1] 2.601109
```

- Podemos testar $H_0 : \beta_0 = 5$, contra $H_1 : \beta_0 > 5$, utilizando:

```
t = (sum_reg$coefficients[1,1]-5)/sum_reg$coefficients[1,2] #estatística-t teste unicaudal
n = nrow(data1)
t_critico = qt(0.95,n-2) #qt() calcula o quantil 95% da distribuição t
p_value = 1-pt(t,n-2) #pt() calcula a função de dist. acumulada da distribuição t
c(t,t_critico,p_value) #teste com nível de significância de 5%
```

```
## [1] 1.45259555 1.64776676 0.07346731
```

- Nesse caso, não rejeita-se H_0 ao nível de signficância de 5%.
- Note que para amostras grandes, em geral, $n > 30$, podemos usar a tabela da normal padrão para aproximar os valores críticos:

```
z_critico = qnorm(0.95,mean=0,sd=1) #qnorm() calcula o quantil 95% da distribuição N(0,1)
p_value = 1-pnorm(t,mean=0,sd=1) #pnorm() calcula a função de dist. acumulada da N(0,1)
c(t,z_critico,p_value) #teste com nível de significância de 5%
```

```
## [1] 1.45259555 1.64485363 0.07316804
```

Intervalos de confiança

- Utilizando os conceitos da sessão anterior, prova-se que o intervalo de confiança (IC) com coeficiente $\gamma = \alpha/2$ (sendo α o nível de significância) para β_j é igual à $[\hat{\beta}_j - t_{df}^{\alpha/2}\hat{\sigma}_{\hat{\beta}_j}; \hat{\beta}_j + t_{df}^{\alpha/2}\hat{\sigma}_{\hat{\beta}_j}]$, sendo df os graus de liberdade da distribuição t correspondente. No caso da regressão linear simples com intercepto: $df = n - 2$, e no caso da regressão linear simples com intercepto $df = n - k - 1$.
- No R, os intervalos podem ser calculados usando as funções que calculam os quantis da distribuição t e acessando os coeficientes. Para o exemplo anterior, o IC 95% para o intercepto é:

```
IC = c(sum_reg$coefficients[1,1]-qt(0.975,n-2)*sum_reg$coefficients[1,2],
       sum_reg$coefficients[1,1]+qt(0.975,n-2)*sum_reg$coefficients[1,2])
IC
```

```
## [1] 4.868444 5.878166
```

Um parêntese:"Loops" no R:

- Muitas vezes é necessário usar o comando **for()** do R para executar comandos de forma sucessiva. Por exemplo, imagine que queremos estimar todas as possíveis regressões simples, utilizando um banco de dados. Usando banco de dados do exemplo anterior, temos que a variável dependente é o salário (*wage*) e há quatro possíveis regressores (*educ*,*exper*,*female*, e *married*). Uma forma de fazer isso no R é a seguinte:

```
vetor_R2 = rep(0,4)
for (i in 2:5){
  vetor_R2[i-1]=summary(lm(data1[,1]~data1[,i]))$r.squared
}
vetor_R2
```

```
## [1] 0.16475751 0.01274719 0.11566656 0.05235730
```

- Note que para usar o **for()** a sintaxe é **for (i in indices) {expressão}**. Ou seja, para o exemplo supracitado, foram estimados 4 modelos sucessivamente. Nesse caso, para os índices i pertencendo (in) ao conjunto *{2,3,4,5}*, substituimos o vetor dimensionado anteriormente (em que definimos um vetor de zeros com tamanho 4 através do comando **rep(0,4)**) pelos coeficientes de determinação de cada um dos modelos. Isto é, cada índice do vetor correponde à um modelo. Note que indexamos por $[i-1]$, pois ainda que os índices das colunas do banco de dados variam de 2 à 5, os índices do vetor de R2 variam de 1 à 4.
- Também pode-se estimar todos os possíveis modelos de regressão tendo como salário a variável dependente: no total serão 4 (C_4^1) modelos de regressão simples , 6 (C_4^2) modelos de regressão com duas variáveis explicativas, 4 (C_4^3) modelos com três variáveis explicativas e 1 (C_4^4) modelo com 4 variáveis.

```
modelos1v = rep(0,4)
modelos2v = rep(0,6)
modelos3v = rep(0,4)
ind_1 = c(2,3,4,5)
ind_2 = matrix(c(2,3,2,4,2,5,3,2,3,4,3,5),6,2,byrow=T)
ind_3 = matrix(c(2,3,4,2,3,5,2,4,5,3,4,5),4,3,byrow=T)
for (i in 1:4){
  modelos1v[i] = summary(lm(data1[,1]~data1[,ind_1[i]]))$adj.r.squared
}
for (i in 1:6){
  X = as.matrix(data1[,ind_2[i,]]) #colocar em formato matricial
  modelos2v[i] = summary(lm(data1[,1]~X))$adj.r.squared
}
for (i in 1:4){
  X = as.matrix(data1[,ind_3[i,]]) #colocar em formato matricial
  modelos3v[i] = summary(lm(data1[,1]~X))$adj.r.squared
}
modelos4v = summary(lm(data1[,1]~as.matrix(data1[,2:5])))$adj.r.squared
R2_adj = c(modelos1v,modelos2v,modelos3v,modelos4v)
which.max(R2_adj) #indice que maximiza R2
```

```
## [1] 15
```

```
which.min(R2_adj) #indice que minimiza R2
```

```
## [1] 2
```