

# Aula EN TP6: Ficha de trabalho 4

*Tiago A. Marques*

*October 22, 2018*

## Introdução

Neste documento vamos implementar a ficha de trabalho 4, sobre amostragem, realizada na quinta semana de aulas.

## Exercício 1

### Enunciado

Pretende comparar o número de espécies de aves em diferentes habitats: montado, olival e estepe cerealífera. Dispõe de alguns dados preliminares:

Número de unidades territoriais com os 3 habitats: 272, 562, 56. (**nota do tradutor: aqui devia dizer, “respectivamente para montado, olival e estepe”**)

Dados preliminares de transectos com várias distâncias percorridas: ficheiro DataTP4avesdist.csv (**nota do tradutor: aqui devia dizer, “Estes transectos com diferentes comprimentos foram realizados aleatoriamente nos vários habitats, mas não sabemos em qual habitat”**)

Número (**nota do tradutor: aqui devia dizer, “cumulativo, em função do número de transectos realizados”**) de espécies identificadas em amostragens preliminares: ficheiro DataTP4avesnum.csv

Estabeleça um plano de amostragem apropriado, sabendo que dispõe de 50000 euros para o estudo e que cada dia de amostragem tem um custo de 500 euros. (**nota do tradutor: aqui devia dizer, “assuma que, em cada dia, realiza apenas um ponto de amostragem, ou seja, dias = pontos de amostragem”**)

### Resolução

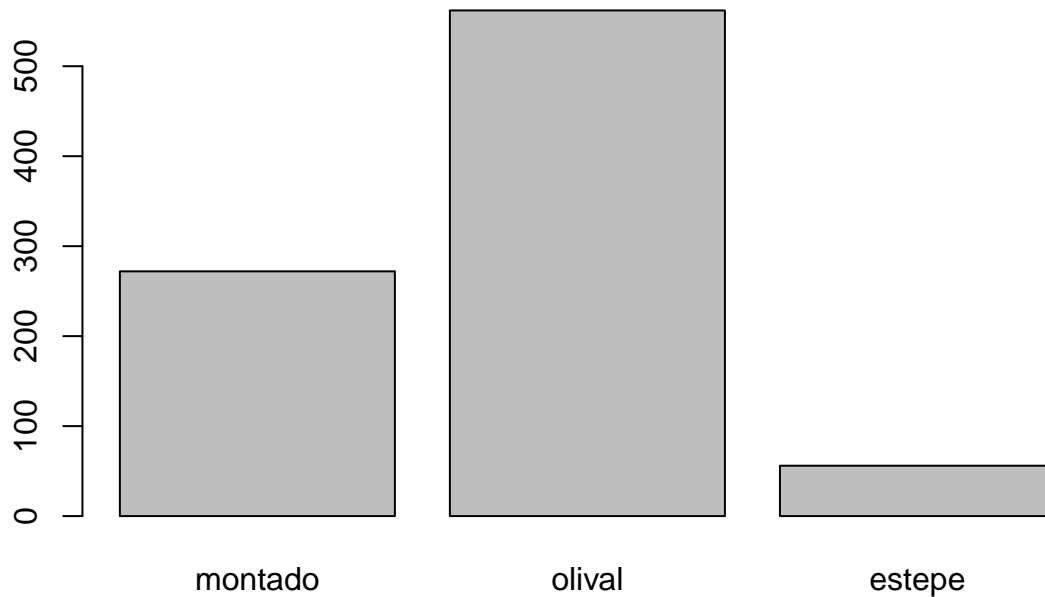
A primeira questão a resolver é quantas unidades de amostragem fazer em cada habitat.

Começamos por definir algumas variáveis necessárias que eram dadas no enunciado

```
habitats=c("montado","olival","estepe")
#número de unidades territoriais
nuts=c(272,562,56)
#orçamento
budget=50000
#custo por unidade
cpu=500
```

Vejamos qual é a distribuição do número de unidades por habitat

```
par(mfrow=c(1,1))
barplot(nuts,names.arg = habitats)
```



Quantas unidades podemos amostrar? Bem, podemos amostrar  $50000/500=100$  unidades.

Qual a melhor forma de fazer a alocação das unidades de amostragem?

A primeira opção seria uma divisão igual por habitat: 33, 33, 34. No entanto, isso queria dizer que ficávamos com uma representação proporcionalmente desequilibrada por estrato: muito maior representação proporcional, por exemplo, de estepe do que de olival.

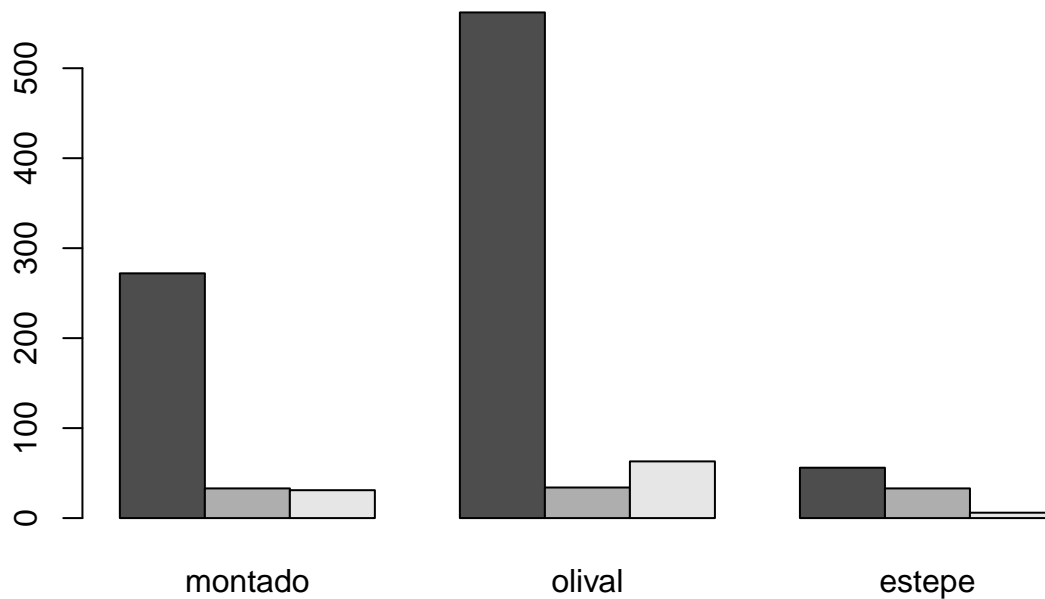
```
alocar1=c(33,34,33)
```

A segunda opção será fazê-lo proporcional à área de habitat disponível. Nesse caso, poderíamos escolher, por exemplo

```
#unidades possiveis de amostrar
upa=budget/cpu
#como as alocar
alocar2=round(upa*nuts/sum(nuts))
```

Podemos ver qual seria a representação dos habitats em cada esquema

```
par(mfrow=c(1,1))
barplot(matrix(c(nuts,alocar1,alocar2),ncol=3,nrow=3,byrow=TRUE),names.arg = habitats,beside=TRUE)
```



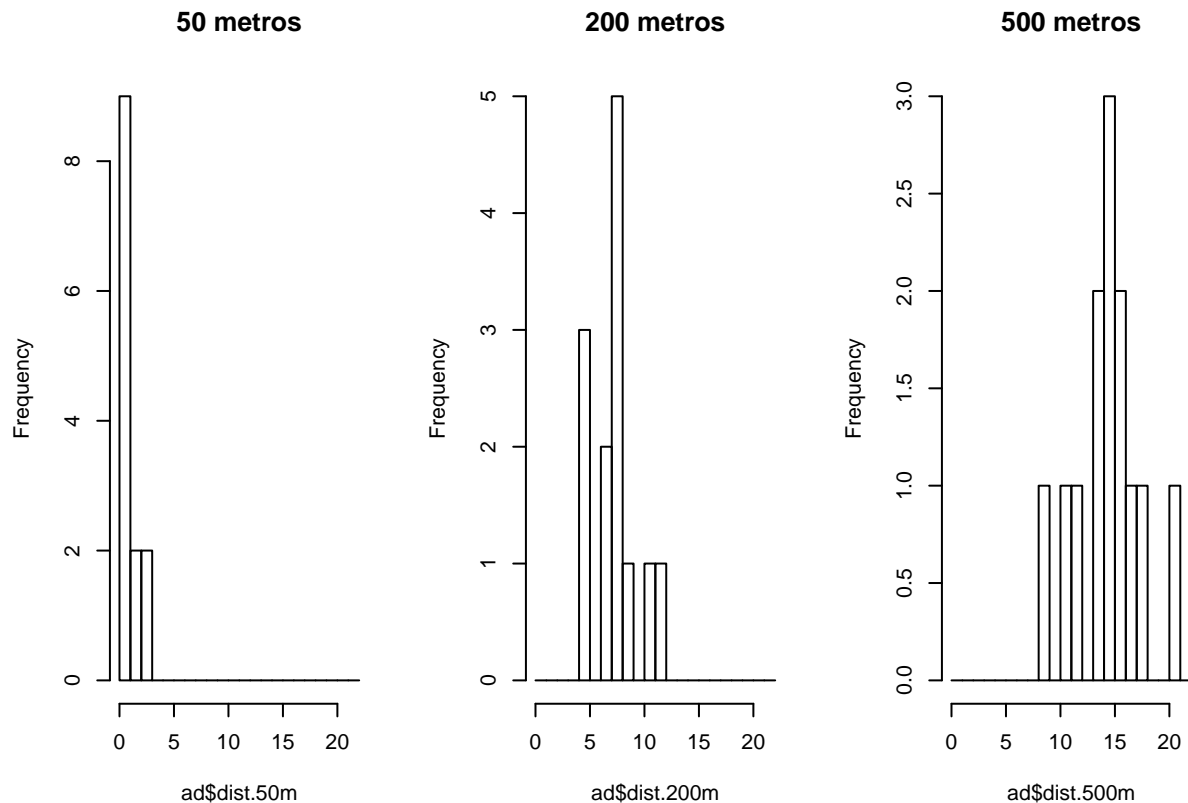
O segundo esquema parece de facto mais sensato. Mas note-se que poderíamos alterar o esquema se tivéssemos mais informação sobre o número de espécies em cada habitat. O que na realidade temos, por isso vamos tentar usar essa informação.

Começamos por ler os dados

```
ad <- read.csv("DataTP4avesdist.csv", header=TRUE, sep=";")
an <- read.csv("DataTP4avesnum.csv", header=TRUE, sep=";")
```

Podemos ver que temos 3 tipos de transectos, com 50, 200 e 500 metros. Os números de indivíduos detectados por transecto são naturalmente maiores quanto mais compridos são os transectos.

```
par(mfrow=c(1,3))
hist(ad$dist.50m,main="50 metros",breaks=0:22)
hist(ad$dist.200m,main="200 metros",breaks=0:22)
hist(ad$dist.500m,main="500 metros",breaks=0:22)
```

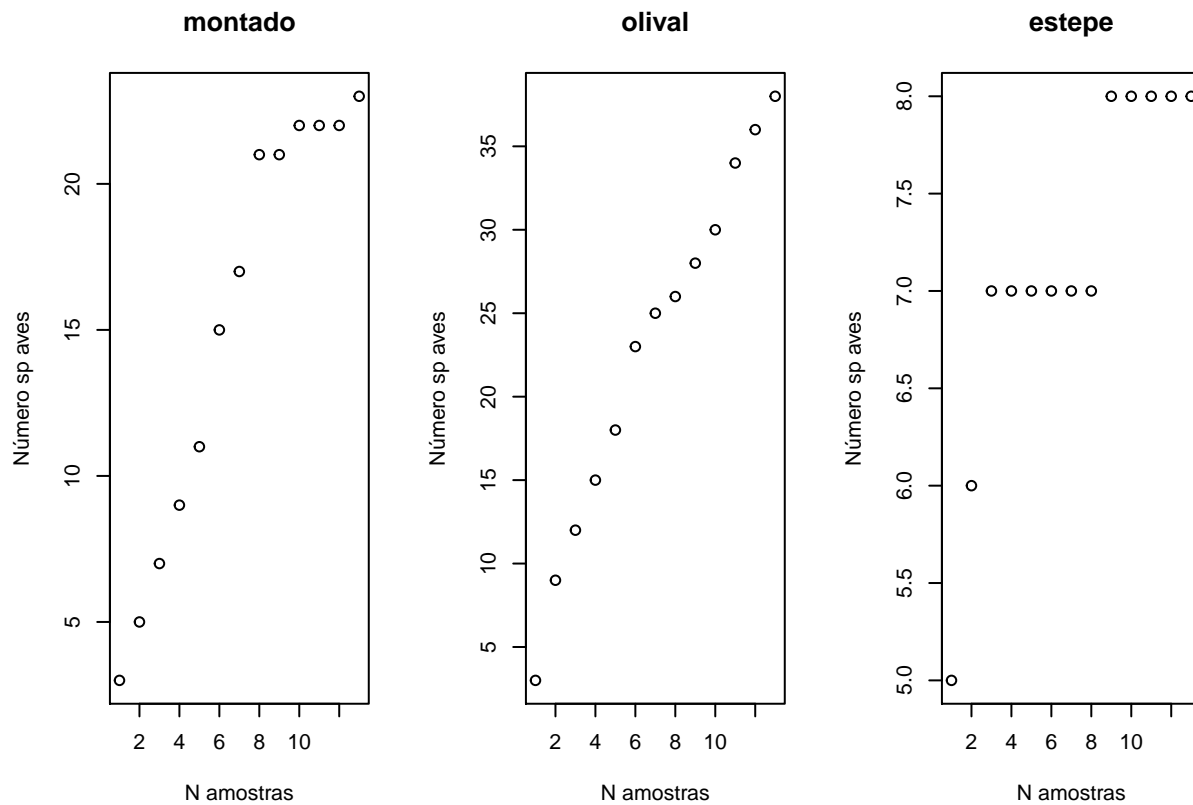


Sendo que é omissa o custo da amostragem em função do comprimento do transecto, podíamos assumir que o principal custo é o transporte e a logística até ao sítio, mas depois de lá estar, a diferença no custo de fazer 50 vs. 500 metros é pouco relevante. Neste caso, os transectos com 500 metros parecem a abordagem mais sensata, porque nos dão uma amostra maior. Além disso, desaparece o “problema” dos muitos 0’s presentes nos transectos mais curtos, e a distribuição do número de indivíduos por transecto é mais “simpática”: simétrica, provavelmente bem aproximada a uma Gaussiana.

Se, por outro lado, o número de espécies detectadas em 200 ou 500 metros fosse equivalente, a opção de 200 metros seria a mais indicada, pois obtinhamos o mesmo resultado com menos de metade do esforço.

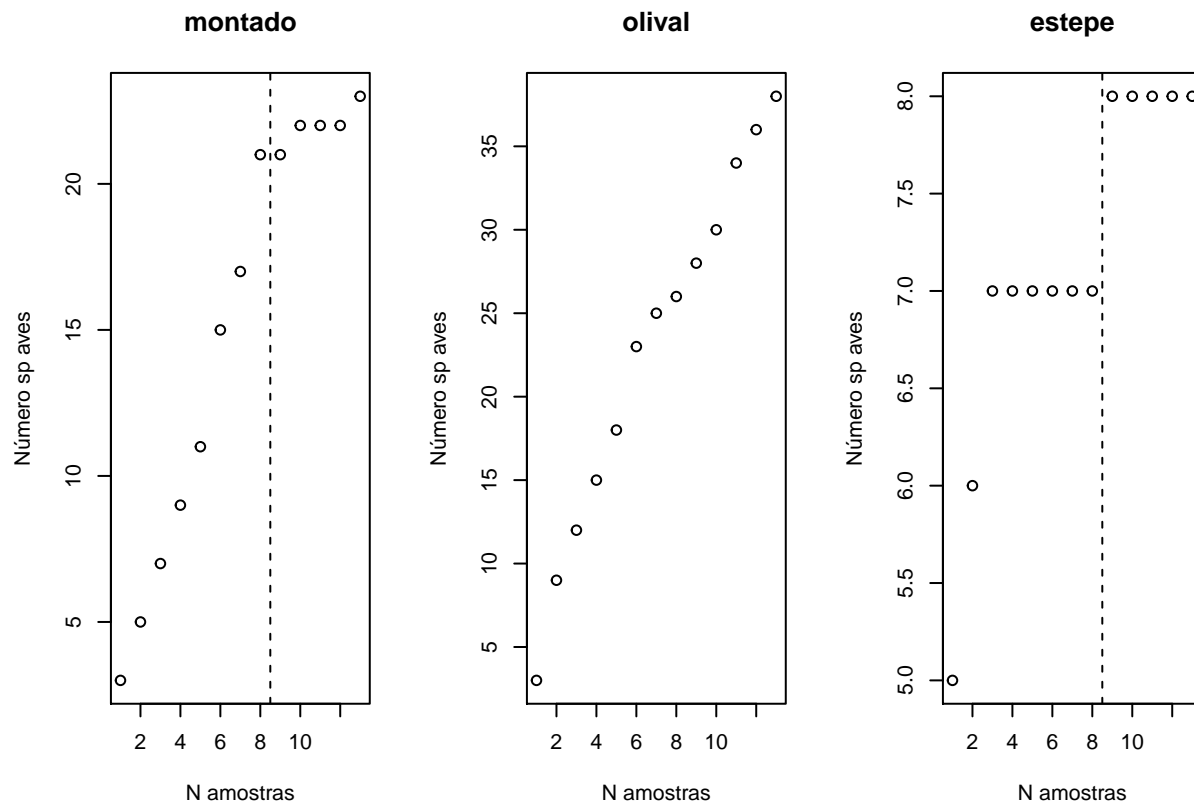
No segundo ficheiro temos o número de espécies detectadas em função do número de unidades de habitat amostradas (nota que é um número cumulativo!)

```
par(mfrow=c(1,3))
plot(an$montado,xlab="N amostras", ylab="Número sp aves",main="montado")
plot(an$olival,xlab="N amostras", ylab="Número sp aves",main="olival")
plot(an$estepe,xlab="N amostras", ylab="Número sp aves",main="estepe")
```



Aparentemente, com mais de 8 amostras já conseguimos ver o número total de espécies presentes no montado e estepe, mas nem sequer 12 amostras são suficientes para que a função cumulativa estabilize no olival, que parece ter uma maior riqueza específica.

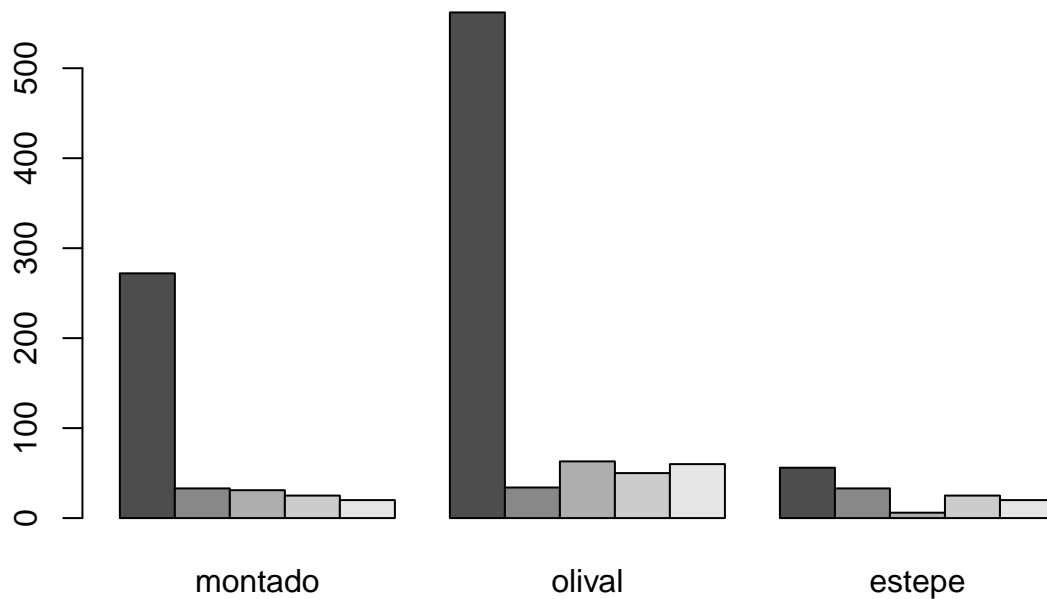
```
par(mfrow=c(1,3))
plot(an$montado,xlab="N amostras", ylab="Número sp aves",main="montado")
abline(v=8.5,lty=2)
plot(an$olival,xlab="N amostras", ylab="Número sp aves",main="olival")
plot(an$estepe,xlab="N amostras", ylab="Número sp aves",main="estepe")
abline(v=8.5,lty=2)
```



Assim sendo, vemos que a alocação proporcional sugerida no início não seria adequada, porque dificilmente conseguiríamos obter o número total de espécies na estepe. Por outro lado, a amostragem equitativa também não é adequada, porque será melhor ter de facto mais amostras de olival.

Um possível esquema final seria o seguinte

```
finaloc1=c(25,50,25)
finaloc2=c(20,60,20)
#matriz de alocações
mataloc=matrix(c(nuts,alocar1,alocar2,finaloc1,finaloc2),ncol=3,nrow=5,byrow=TRUE)
par(mfrow=c(1,1))
barplot(mataloc,names.arg = habitats,beside=TRUE)
```



# Exercício 2

## Enunciado

Pretende avaliar o efeito de dois contaminantes (Hg e Pb) no crescimento de uma espécie de crustáceo. Para o efeito vai realizar uma experiência em laboratório.

Tem acesso a alguns dados preliminares:

Estimativas de crescimento preliminares em função do tratamento: `DataTP4contamcrescimento.csv`

Estimativas de contaminação no ambiente natural: `DataTP4contamambiente.csv`

Tem uma sala disponível para realizar a experiência com 5 bancadas de 20 m, onde é possível instalar até 10 aquários por bancada.

O projecto onde está integrado não tem uma verba muito elevada para a realização desta experiência, pelo que lhe é solicitado que estabeleça um plano com muita contenção de custos. Faça o delineamento de uma experiência nestas condições.

## Resolução

Ler os dados

```
ca <- read.csv2("DataTP4contamambiente.csv", row.names=1)
ca=ca[,1:2]
cc <- read.csv2("DataTP4contamcrescimento.csv")
```

Vejamos qual a variação, na natureza, que cada elemento apresenta

```
apply(ca,2,range)
```

```
##      Hg  Pb  
## [1,]  5.1 0.1  
## [2,] 15.2 18.1
```

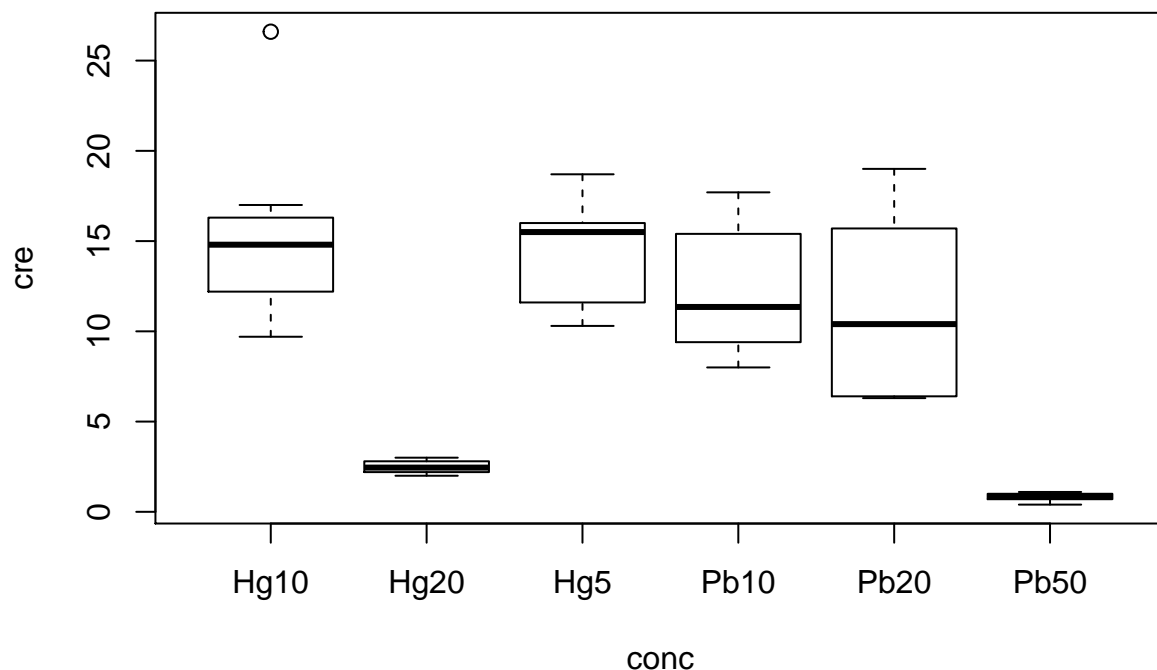
Temos por isso o Hg a variar entre 5.1 e 15.2, enquanto que o Pb varia entre 0.1 e 18.1.

E em relação aos crescimentos? Primeiro podemos reorganizar a informação

```
conc=c("Hg5", "Hg10", "Hg20", "Pb10", "Pb20", "Pb50")  
names(cc)=conc  
#colocar todos os crescimentos juntos  
cresc=data.frame(conc=rep(conc,each=nrow(cc)),cre=as.numeric(as.matrix(cc)))
```

e depois fazer um boxplot com os crescimentos por concentração

```
with(cresc,boxplot(cre~conc))
```



Independentemente do que já vi noutras soluções deste mesmo problema, e assumindo que fazemos um controlo, prefiro ter 1 controlo e duas concentrações de cada elemento. Neste caso, uma alocação proporcional optima será 10 aquários para cada tratamento, e dois aquários por bancada para cada tratamento (na realidade, esta alocação poderia ser uma variante de um quadrado latino, um esquema de amostragem que usa blocos equilibrados).

Assim sendo, a única coisa que ficava a faltar era fazer a alocação concreta dos aquários a cada bancada. Um forma simples de o fazer seria por exemplo definir que em cada bancada tinhamos apenas de definir as ordens dos tratamentos, o que faríamos aleatoriamente



```

set.seed(1234)
tratamentos=c("Hg10","Hg20","Pb10","Pb50","C")
tratamentos2=rep(tratamentos,each=2)
B1=sample(tratamentos2,size=10,replace=FALSE)
B2=sample(tratamentos2,size=10,replace=FALSE)
B3=sample(tratamentos2,size=10,replace=FALSE)
B4=sample(tratamentos2,size=10,replace=FALSE)
B5=sample(tratamentos2,size=10,replace=FALSE)

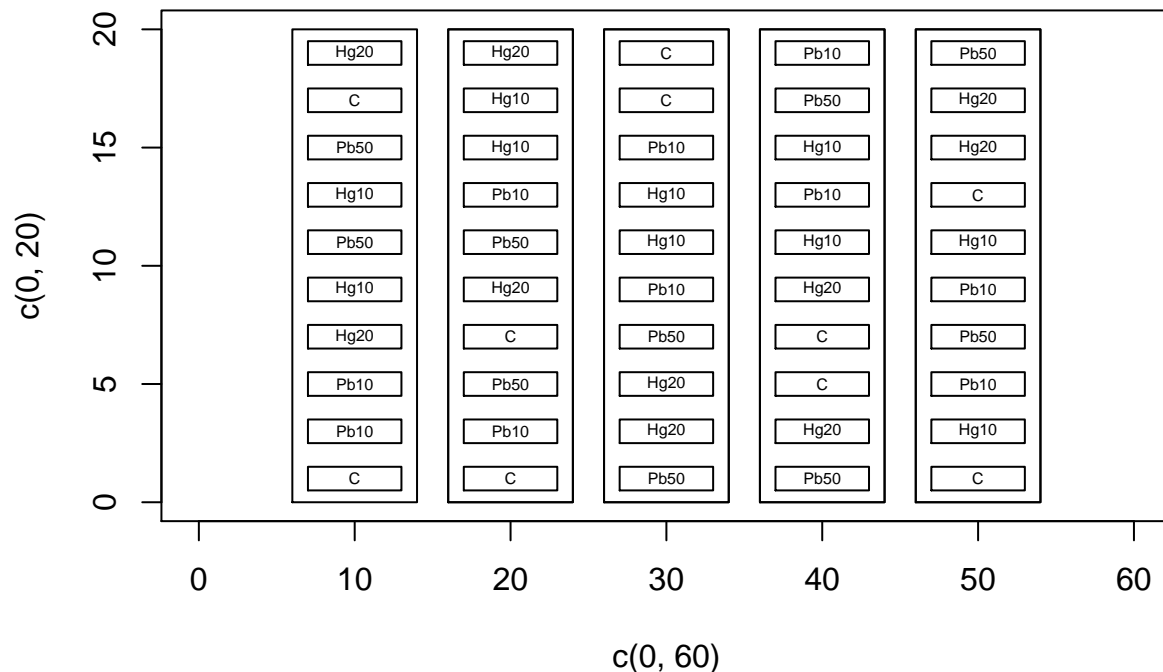
```

Assim sendo, os tratamentos nos aquários ficariam definidos da seguinte forma (note-se que estou apenas a usar as capacidades gráficas do R para fazer uma representação esquemática dos aquários, isto não seria necessário para responder à pergunta de forma adequada)

```

plot(c(0,60),c(0,20),type="n")
polygon(x=c(6,14,14,6),y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+10,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+20,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+30,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+40,y=c(0,0,20,20))
for(i in 1:10) polygon(x=c(7,13,13,7),y=c(2*i-1-0.5,2*i-1-0.5,2*i-1+0.5,2*i-1+0.5))
for(i in 1:10) polygon(x=c(7,13,13,7)+10,y=c(2*i-1-0.5,2*i-1-0.5,2*i-1+0.5,2*i-1+0.5))
for(i in 1:10) polygon(x=c(7,13,13,7)+20,y=c(2*i-1-0.5,2*i-1-0.5,2*i-1+0.5,2*i-1+0.5))
for(i in 1:10) polygon(x=c(7,13,13,7)+30,y=c(2*i-1-0.5,2*i-1-0.5,2*i-1+0.5,2*i-1+0.5))
for(i in 1:10) polygon(x=c(7,13,13,7)+40,y=c(2*i-1-0.5,2*i-1-0.5,2*i-1+0.5,2*i-1+0.5))
polygon(x=c(6,14,14,6)+10,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+20,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+30,y=c(0,0,20,20))
polygon(x=c(6,14,14,6)+40,y=c(0,0,20,20))
for(i in 1:10) text(10,2*i-1,B1[i],cex=0.5)
for(i in 1:10) text(20,2*i-1,B2[i],cex=0.5)
for(i in 1:10) text(30,2*i-1,B3[i],cex=0.5)
for(i in 1:10) text(40,2*i-1,B4[i],cex=0.5)
for(i in 1:10) text(50,2*i-1,B5[i],cex=0.5)

```



Note-se, que apenas por acaso, 3 dos aquários na parte superior (1a linha) ficaram com o tratamento Hg20, e os restantes 2 com o Pb50.

Numa situação como esta seria possível criar uma alocação verdadeiramente equilibrada, usando dois quadrados latinos de 5 por 5.

Por outro lado, poderíamos decidir usar menos aquários (por exemplo 5 em cada bancada, com apenas um aquário por bancada por tratamento), para poupar dinheiro. Nesse caso poderíamos usar um quadrado latino puro. Não sendo um tópico dado neste curso, um a forma de o criarmos seria recorrendo a uma função do package `magic`

```
library(magic)
```

```
## Loading required package: abind
```

```
mydesign=rlatin(5)
```

```
mydesign2=matrix(NA,5,5)
```

```
for (i in 1:5) mydesign2=ifelse(mydesign==i,tratamentos[i],mydesign2)
```

```
mydesign2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "Pb50" "Hg10" "Pb10" "Hg20" "C"
## [2,] "Hg20" "Pb10" "C" "Pb50" "Hg10"
## [3,] "C" "Hg20" "Pb50" "Hg10" "Pb10"
## [4,] "Pb10" "Pb50" "Hg10" "C" "Hg20"
## [5,] "Hg10" "C" "Hg20" "Pb10" "Pb50"
```