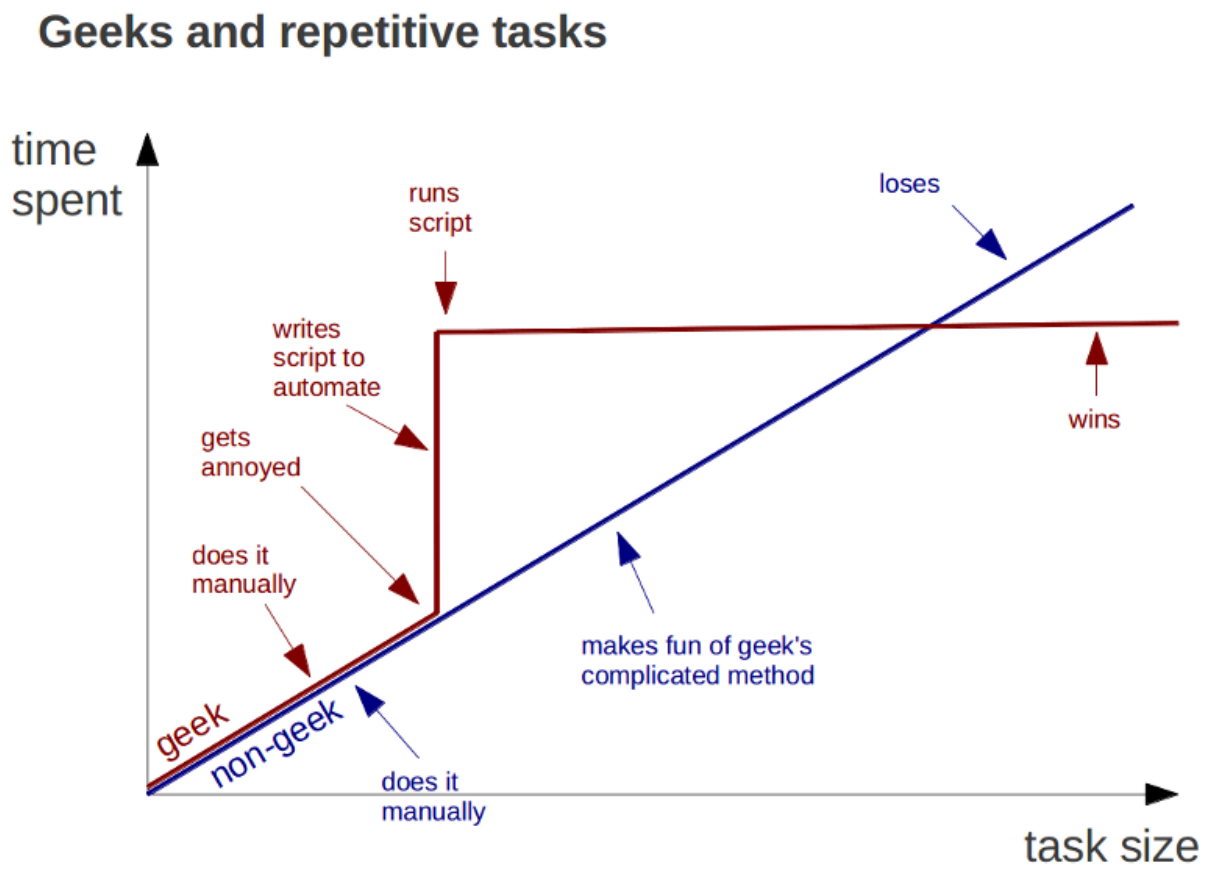
# Repeat “things” in R - loops, functions, apply

author: Dorothea Hug Peter
date: 23.05.2016 autosize: true

## Introduction

type:section




## Why?

- Fewer mistakes
- Your future self

## Tools

- Write a loop

```
for (x in 1:length(input)){
boxplot(input[,x]~category)
}
```

- Write a function

```
correction<-function(input,correction){
  input-correction
}
```

==========

- Use the apply family

```
apply(input,2,mean)
```

# My examples

- Read Data
- Treat Data
- Create Graphs

# My examples

- Read Data
- **Treat Data**
- **Create Graphs**

Dataset: Doubs river chemistry, provided in ADE4 library

```
  dfs alt   slo flo pH har pho nit amm oxy bdo
1   3 934 6.176  84 79  45   1  20   0 122  27
2  22 932 3.434 100 80  40   2  20  10 103  19
3 102 914 3.638 180 83  52   5  22   5 105  35
4 185 854 3.497 253 80  72  10  21   0 110  13
5 215 849 3.178 264 81  84  38  52  20  80  62
6 324 846 3.497 286 79  60  20  15   0 102  53
```

# Functions

- Break long script into small, formalized operations

Why? - Readability - Avoid errors - Facilitate code recycling

## Functions

Define a function

```
function.name<-function(arg1, arg2, arg3=5,...){
  newVal<-arg1+arg2
  newVal/arg3
}
```

Source from file or execute in console

```
function.name(arg1,arg2)
```

## Functions - Example

Define a function

```
np_ratio<-function(arg1, arg2, arg3,...){
  newVal<-arg1+arg2
  newVal/arg3
}
```

```
ratio_doubs<-n_pratio(doubs$env$nit,
                      doubs$env$amm,
                      doubs$env$pho)
```

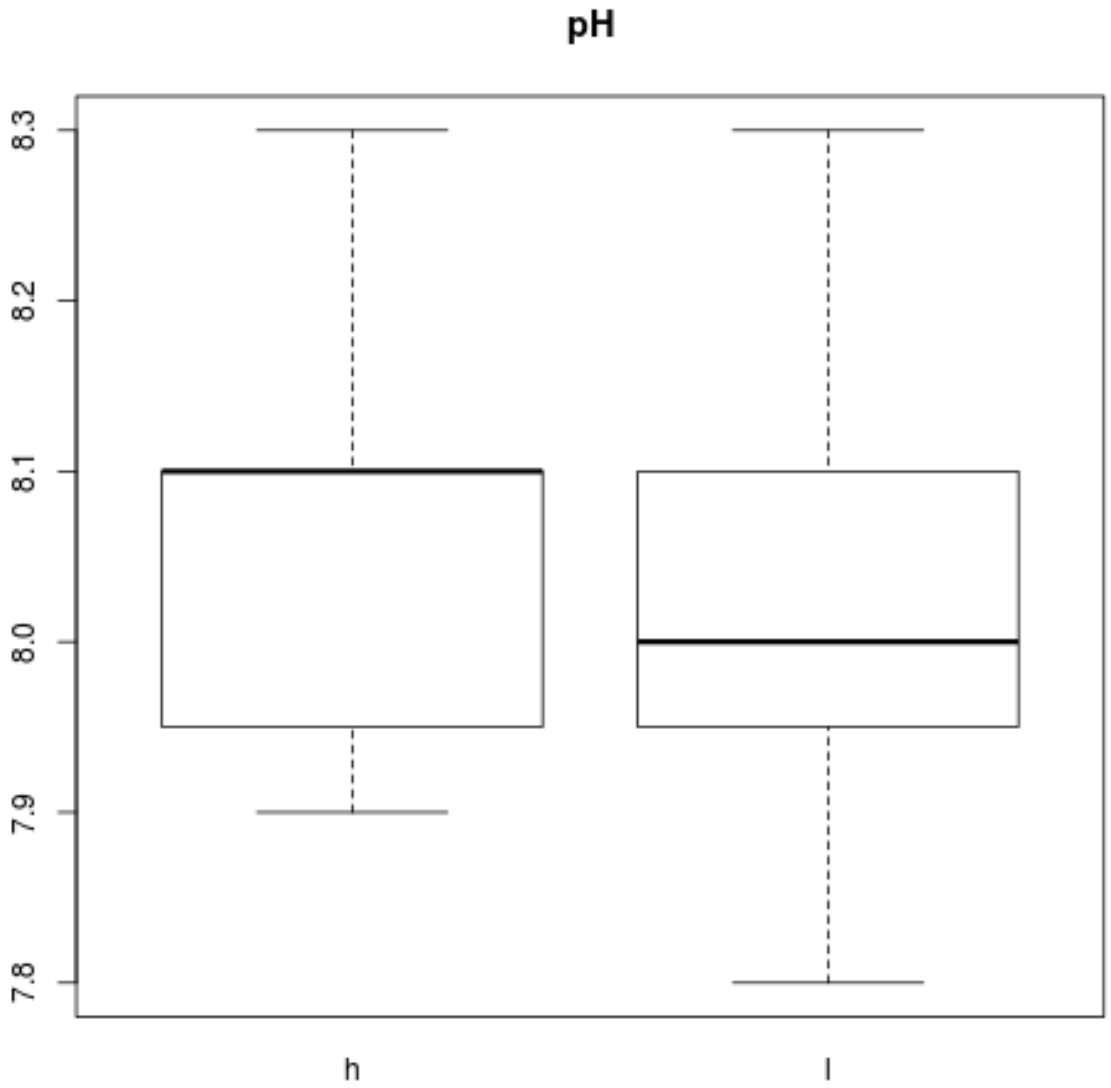
## Functions - Example2

id: function Define a function

```
bp_me<-function(x,cat,t,
                cnames=levels(as.factor(cat))){
  #make a boxplot
  boxplot(x~cat,xlab="",ylab="",names.arg=NA,
          axes=F,outline=F)
  #do some fine tuning
  par(mgp=c(3,1,0))
  axis(2)
  #par(mgp=c(3,0,0))
  par(las=1)
  axis(side=1,at=c(1:length(cnames)),
       labels=cnames,lty=0)
  #par(mgp=c(3,1,0))
  par(las=2)
  box()
  title(main=t)
}
```

## Functions - Example2

```
categs<-ifelse(
  doubs$env$alt>=median(doubs$env$alt),"h","l")
bp_me(doubs$env$pH/10,categs,"pH")
```




## Functions - General remarks

- Choose clear names, make them easy to understand
- Break long functions into small steps
- Avoid using global variables

- use ... in the function definition to allow flexible input

## Loops

type:section

- Avoid!
- Very versatile
- Slow, more difficult to recycle and read

## Loops

```
par(mfrow=c(2,2),las=2,oma = c(1.2,2, 1, 0.5),mar=c(2,4,2,1))
input<-doubs$env[,c("har","pho","nit","amm")]
input[,2:4]<-input[,2:4]/100
```

## Loops

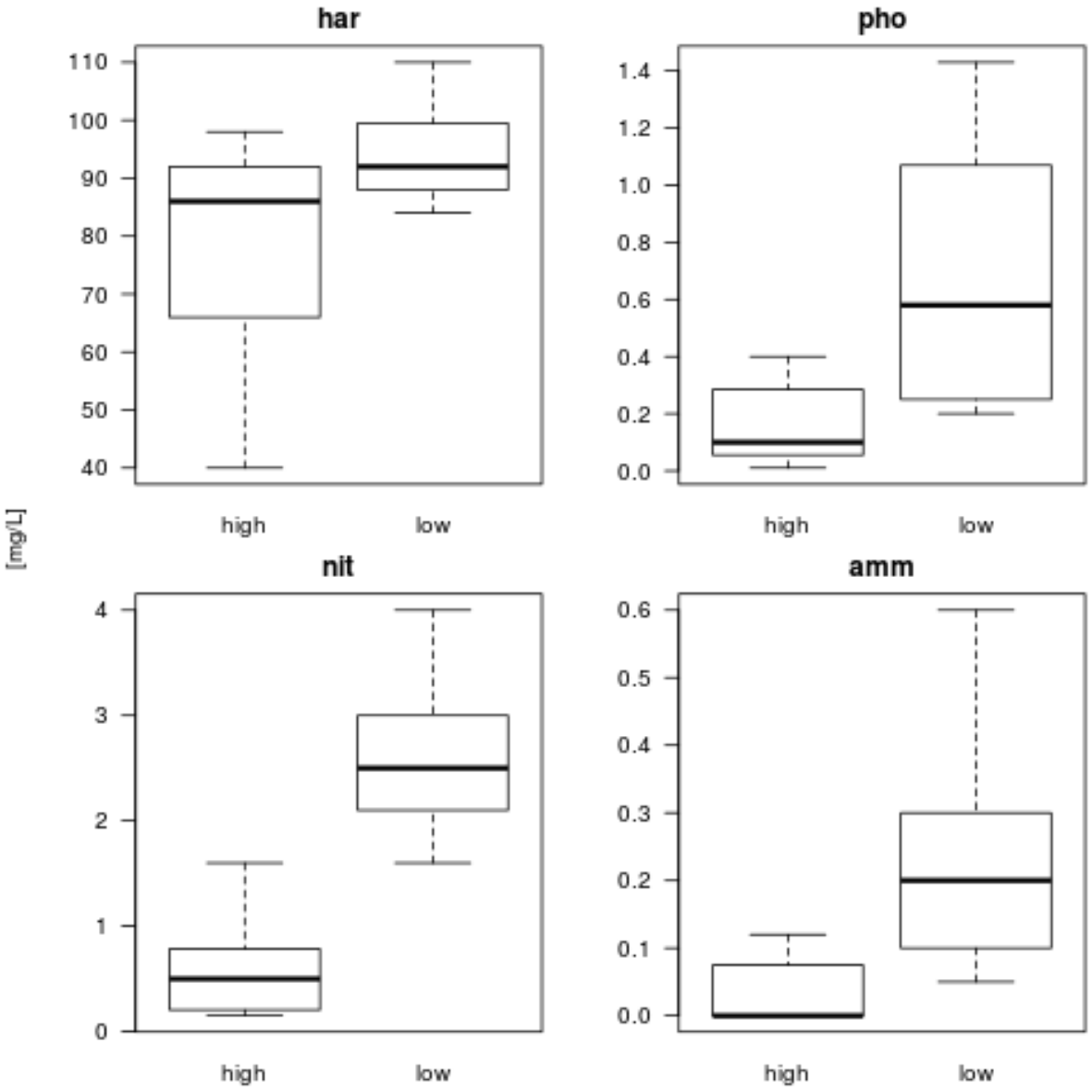
title:false

```
for (x in 1:length(input)){

boxplot(input[,x]~categs,xlab="",ylab="",
        names.arg=NA,axes=F,outline=F)
  axis(2)
  par(las=1)
  axis(side=1,at=c(1:length(
    levels(as.factor(categs)))),
  labels=c("high","low"),lty=0)
  box()
  title(colnames(input)[x])
}
par(las=0)
mtext("[mg/L]", side=2,outer = TRUE, cex = 0.8,)
```

## Loops

title:false




## The apply family

type:section

- Apply a function to a series of objects
- Input and output can vary
- Used in the dplyr package to construct some useful tools

## apply()

```
apply(x,MARGIN,fun(x))
```

```
a<-apply(doubs$env,2,mean)
b<-apply (doubs$fish,1,sum)
head(as.data.frame(b))
```

```
   b
1  3
2 12
3 16
4 21
5 34
6 21
```

==========

```
head(as.data.frame(b))
```

```
   b
1  3
2 12
3 16
4 21
5 34
6 21
```

## lapply()

```
lapply(X,fun(x))
```

## lapply()

title:false

```
lapply(as.list(doubs$env),mean)
```

```
$dfs
[1] 1879.033

$alt
[1] 481.5

$slo
[1] 2.757733
```

```
$flo
[1] 2220.1

$pH
[1] 80.5

$har
[1] 86.1

$pho
[1] 55.76667

$nit
[1] 165.4

$amm
[1] 20.93333

$oxy
[1] 93.9

$bdo
[1] 51.16667
```

## lapply()

title:false

```
a<-lapply(doubs,function(x){apply(x,2,mean)})
summary(a)
```

```
        Length Class  Mode
env     11     -none- numeric
fish    27     -none- numeric
xy       2     -none- numeric
species  4     -none- numeric
```

## mapply()

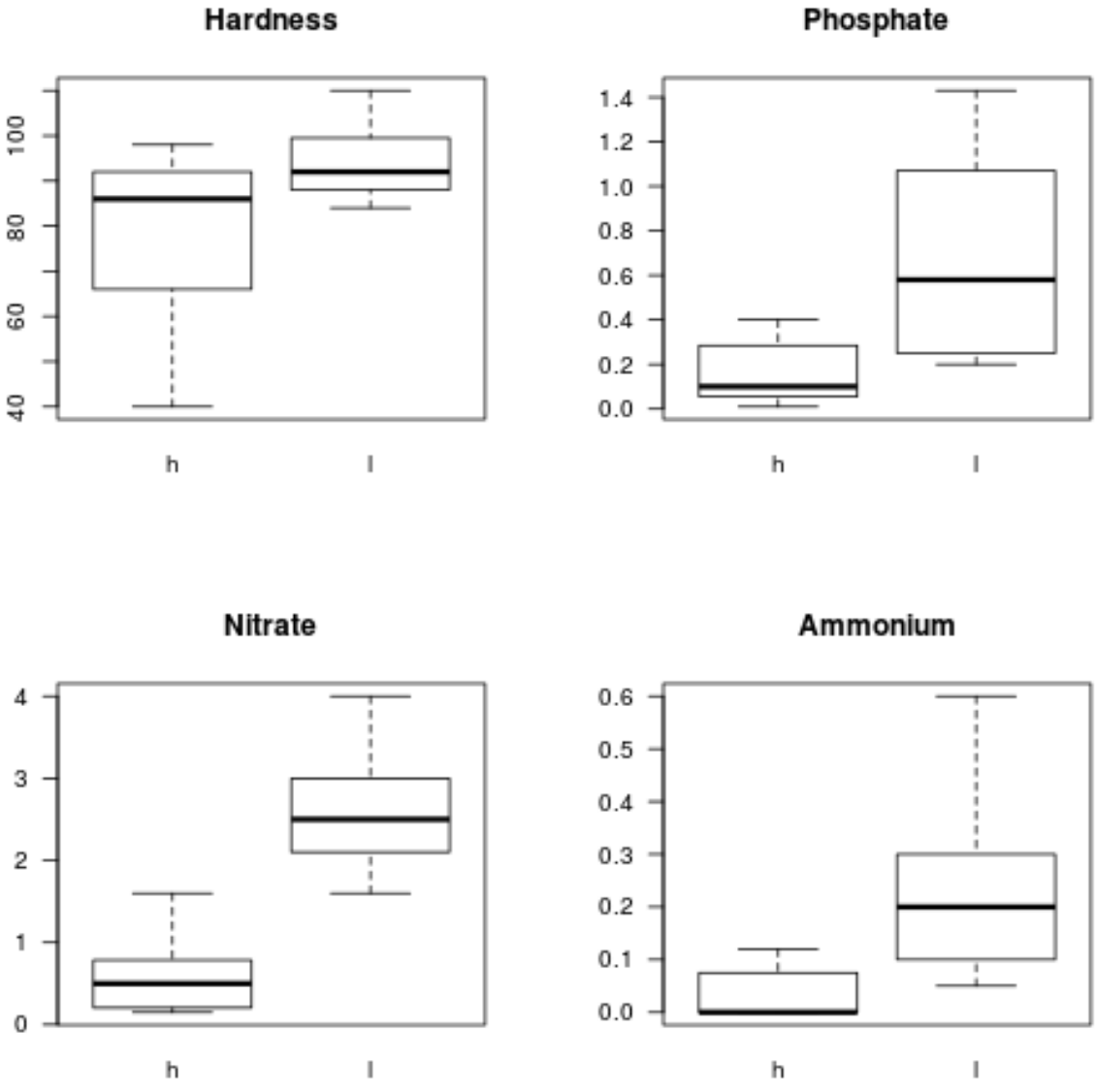
id: application

```
mapply(FUN, arg1, arg2,..., MoreArgs = NULL)
```

bp_me definition

```
titles<-c("Hardness","Phosphate","Nitrate","Ammonium")
par(mfrow=c(2,2)))
mapply(bp_me,x=input,t=titles,MoreArgs=list(cat=categs))
summary(a)
```

## mapply()




```
$har
NULL

$pho
NULL

$nit
NULL

$amm
NULL
```

```
        Length Class  Mode
env     11     -none- numeric
fish    27     -none- numeric
xy       2     -none- numeric
species  4     -none- numeric
```

## Ressources

Two blog posts that I like

http://nicercode.github.io/guides/repeating-things/ http://nicercode.github.io/guides/functions/

And for the advanced:

http://adv-r.had.co.nz/Functions.html

And to do nice R presentations wit R Studio:

http://support.rstudio.com/hc/en-us/articles/200486468 http://rmarkdown.rstudio.com/authoring_pandoc_markdown.html