

# AnalyseExplo

Berry et Lemeilleur

2022-10-05

## R markdown

### 1. Vectors

#### 1.1 Let's start simple

Create the following vectors e1, e2, e3, e4

```
e1 = c(2, 5, 0, 8)
e2 = seq(1, 200, by=1)
e3 = seq(-200, -210, by=-2)
e4 = c(2, 4, 8, 16, 32, 64, 128)
```

- Create the vector v of 50 elements with conditions

Nous avons trouvé deux solutions pour cette question, voici la première (plus complexe) :

```
v <- rep(1,50)
for (i in 1:50)
  if (i %% 2 == 0)
    v[i] = -1
```

et la seconde qui utilise les puissances de moins 1 :

```
v <- c(-1)^(1:50)
```

- Create the vector e5 = (1, 2, ... 210) by concatenating e2 and e3.

```
e5 = c(e2,e3)
```

- Read seq's help vignette: ?seq C'est fait!
- Create the vector e6 containing 70 equally spaced values between 0 and 1.

```
e6 = seq(0, 1, length.out = 70)
```

- Create the vector e7 containing 10 times the sequence e1. Tip: use rep.

```
e7 = rep (e1,10)
```

- What is the result of the operation e2 moins e3 ? This is called recycling, and this is dangerous.

```
e2-e3
```

```
## Warning in e2 - e3: la taille d'un objet plus long n'est pas multiple de la
## taille d'un objet plus court
```

```
## [1] 201 204 207 210 213 216 207 210 213 216 219 222 213 216 219 222 225 228
## [19] 219 222 225 228 231 234 225 228 231 234 237 240 231 234 237 240 243 246
## [37] 237 240 243 246 249 252 243 246 249 252 255 258 249 252 255 258 261 264
## [55] 255 258 261 264 267 270 261 264 267 270 273 276 267 270 273 276 279 282
```

```
## [73] 273 276 279 282 285 288 279 282 285 288 291 294 285 288 291 294 297 300
## [91] 291 294 297 300 303 306 297 300 303 306 309 312 303 306 309 312 315 318
## [109] 309 312 315 318 321 324 315 318 321 324 327 330 321 324 327 330 333 336
## [127] 327 330 333 336 339 342 333 336 339 342 345 348 339 342 345 348 351 354
## [145] 345 348 351 354 357 360 351 354 357 360 363 366 357 360 363 366 369 372
## [163] 363 366 369 372 375 378 369 372 375 378 381 384 375 378 381 384 387 390
## [181] 381 384 387 390 393 396 387 390 393 396 399 402 393 396 399 402 405 408
## [199] 399 402
```

1.2 Character vectors • Create a vector vowels containing all vowels

```
vowels = c('a', 'e', 'i', 'o', 'u', 'y')
```

- letters is a character vector containing all the letters in alphabetical order.

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

- What does letters %in% vowels do ?

```
letters %in% vowels
```

```
## [1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
## [13] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
## [25] TRUE FALSE
```

Pour chaque element de letters si cet element est egalement dans vowels ça renvoie true, et ça renvoie false si cet element n'est pas dans vowels

- Extract the number of each vowel. Tip: use which

```
which(letters%in% vowels)
```

```
## [1] 1 5 9 15 21 25
```

- Extract the number of each nonvowel

```
x <- letters[! letters %in% vowels]
for (i in x)
  which(letters==letters[i])
```

Moyen plus simple de le faire (sans boucle) :

```
which(! letters%in% vowels)
```

```
## [1] 2 3 4 6 7 8 10 11 12 13 14 16 17 18 19 20 22 23 24 26
```

- What are the letters right after a vowel ?

```
letters[which(letters%in% vowels) + 1]
```

```
## [1] "b" "f" "j" "p" "v" "z"
```

- Create the string myname containing your name (in lower case)

```
myname = "clemence"
```

- Use strsplit to extract individual letters of your name. Observe that strsplit returns a list. Access its first element. !!!! A Revoir !!!!

```
namelet = strsplit(myname,"")[[1]]
```

- Give for each character of your name its number in the alphabet.

```
which(letters %in% namelet)
```

```
## [1]  3  5 12 13 14
```

- Do the same with your right neighbor's name.
- Who is on average lower in the alphabet ?

```
myname2 = "orlane"  
namelet2 = strsplit(myname2, "")[[1]]  
indics1 = which(letters %in% namelet2)  
indics2 = which(letters %in% namelet)  
sum(indics1)/length(indics1)
```

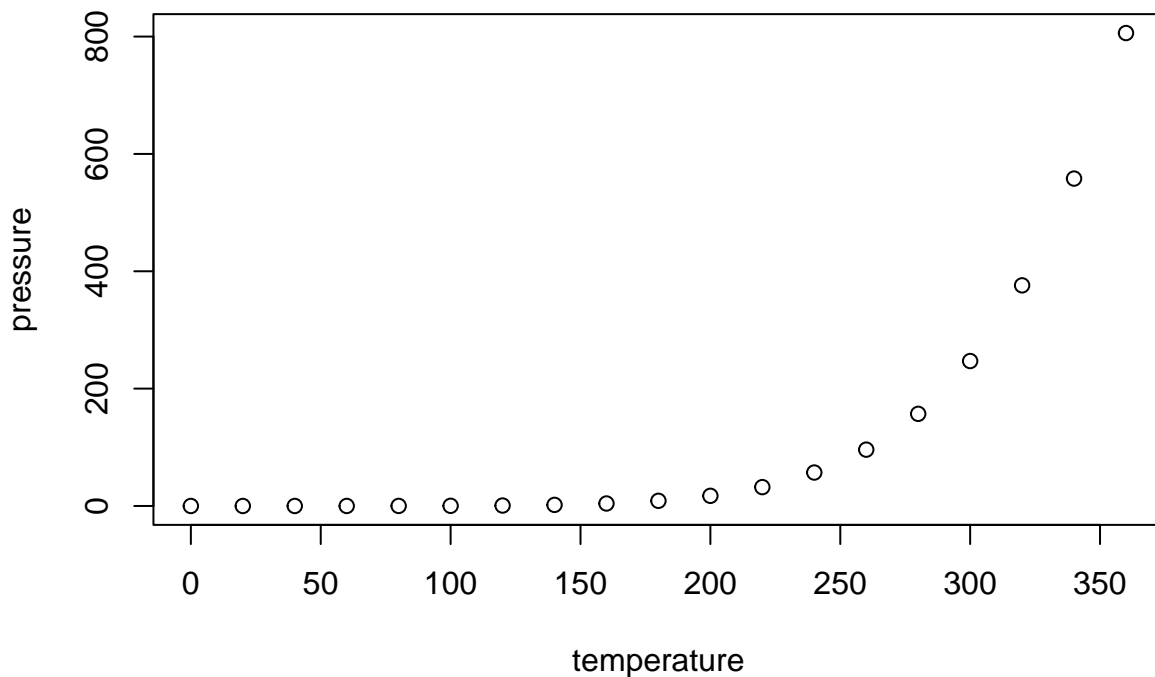
```
## [1] 10.33333
```

```
sum(indics2)/length(indics2)
```

```
## [1] 9.4
```

```
## Including Plots
```

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.