

# Parallel Computing for Data Science

## Lab x002 : Clean & Fast Code

Jairo Cugliari

S1 2016–2017

### 1 Code propre

Un collègue vous envoie cette fonction qui est sensée calculer la somme des premières  $n$  entiers :

```
f1<-function(n)for(i in 1:n)res<-res+1
```

1. Commencez un nouvel script R, nommez le fichier, rajoutez une en-tête descriptive du contenu du fichier et collez le code de votre collègue.
2. Utilisez les règles de style R pour mieux visualiser la fonction.
3. Testez la fonction à partir de deux ou trois cas simples.
4. Utilisez l'outil de débogage interactif pour corriger le code. Vérifiez que le nouveau code est correct avec les cas simples que vous avez utilisé plus haut.

### 2 Chronométrage du temps d'exécution

Cet exercice continue l'exercice précédant.

Voici deux autres versions de la fonction qui renvoie la somme des premières  $n$  entiers :

```
f2 <- function(n) sum(1:n)
```

```
f3 <- function(n) n * (n + 1) / 2
```

Examinez le temps d'exécution du code pour chacune des trois fonctions. Pour chaque fonction, déterminez le ressource limitant du calcul.

### 3 Session de débogage

Débogez le code qui suit :

```
joe=function(x,k){  
  n=length(x)  
  r=NULL  
  for(i in 1:(n-k)) if(all(x[i:i+k-1]==1)) r<-c(r,i)  
  r  
}
```