FACULTÉ DES SCIENCES ECONOMIQUES

UNIVERSITÉ DE RENNES 1

**M1 Statistique & Économétrie**

Ewen Gallic [1]
http://egallic.fr

2015

# Logiciel R et programmation

Exercices

## Partie 2 : Fonctions

### Exercice 1 (création d'une fonction)

1. Créer une fonction nommée `somme_n_entiers` qui retourne la somme des $n$ premiers entiers. Son seul paramètre sera `n` ;
2. Utiliser la fonction `somme_n_entiers()` pour calculer la somme des 100 premiers entiers ;
3. Terminer la fonction par l'assignation du résultat dans un objet nommé `res`, puis évaluer l'expression suivante : `somme_n_entiers(100)`. Que peut-on constater ?
4. Charger les données `diamonds` du *package* `ggplot2` dans la session `R` à l'aide de l'expression suivante :

```
data(diamonds, package = "ggplot2")
```

   Créer une fonction que l'on appellera `prix_diamant_coupe()`, qui, quand on lui fournit la valeur de la coupe du diamant sous forme de caractères (`Fair`, `Good`, `Very Good`, `Premium`, ou `Ideal`), filtre le tableau de données `diamonds` pour ne conserver que les observations pour lesquelles la coupe du diamant correspond à celle indiquée en paramètre, et retourne le prix moyen des observations de la base ainsi filtrée ;
5. Reprendre le code de la fonction précédente, et le modifier pour retourner à présent une liste de deux éléments : (i) la moyenne des prix et (ii) l'écart-type ;
6. Créer la fonction `resume_diamant_coupe_couleur()`, qui pour une coupe et une couleur de diamant données, retourne une liste de deux éléments : (i) la moyenne des prix et (ii) l'écart-type pour les diamants possédant cette coupe et cette couleur (la couleur du diamant est une lettre allant de `J` pour les pires, à `D` pour les meilleurs). Tester la fonction pour la coupe `Fair` et la couleur `D` ;
7. Reprendre la fonction précédente, et lui attribuer la valeur `D` (en chaîne de caractères) comme paramètre effectif pour la couleur. Tester alors l'appel à la fonction en précisant :
   (a) la coupe `Fair` et la couleur `D`,
   (b) la coupe `Fair`, mais pas de paramètre pour la couleur,
   (c) la coupe `Fair` et la couleur `E`,
   (d) la coupe non précisée mais la couleur `E` ;
8. Soit le code suivant :

1. ewen.gallic[at]gmail.com

```r
# @x : (int)
f_test <- function(x){
  x^2
}# Fin de f_test()

# @x : (int)
f_test_2 <- function(y){
  x^2
}# Fin de f_test_2()

# @x : (int)
f_test_3 <- function(y){
  x <- y
  x^2
}# Fin de f_test_3()


x <- 3
```

Expliquez ce qui se passe dans chacun des cas suivants :

```r
f_test(x = 2)
```

```r
f_test_2(y = 2)
```

```r
f_test_2()
```

```r
f_test_3(4)
x
```

**Exercice 2** (création d'une fonction, traitement de chaînes de caractères)
Supposons que les adresses e-mails des étudiants de l'Université de Rennes 1 sont constituées de la manière suivante : le prénom et le nom de famille séparés par un point, le symbole arobase et le enfin le nom de domaine. Supposons de plus que les étudiants ont un seul prénom, et aucune particule au nom de famille. La syntaxe des adresses e-mail est donc comme suit : `nom.prenom@etudiant.univ-rennes1.fr`.

```r
emails <- c("marie.petit@etudiant.univ-rennes1.fr",
            "jean.dupont@etudiant.univ-rennes1.fr",
            "isabelle.martinez@etudiant.univ-rennes1.fr",
            "pierre.moreau@etudiant.univ-rennes1.fr")
```

Créer une fonction, qui à partir d'**une** adresse e-mail d'un étudiant, retourne un `data.frame` contenant trois variables : le prénom, le nom et l'adresse e-mail de cet étudiant.