

Introduction à la programmation en S

Vincent Goulet

École d'actuariat
Université Laval

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

Sommaire

Introduction
à la pro-
grammation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

■ Commandes S

- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- Indixage

Affectations et expressions

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Toute commande S est soit une **affectation**, soit une **expression**
- Une expression est immédiatement évaluée et le résultat est affiché à l'écran :

```
> 2 + 3
```

```
[1] 5
```

```
> pi
```

```
[1] 3.141593
```

```
> cos(pi/4)
```

```
[1] 0.7071068
```

Affectations et expressions

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Lors d'une affectation, une expression est évaluée, mais le résultat est stocké dans un objet (variable) et rien n'est affiché à l'écran
- Le symbole d'affectation est `<-` (ou `->`)

```
> a <- 5
```

```
> a
```

```
[1] 5
```

```
> b <- a
```

```
> b
```

```
[1] 5
```

- Éviter d'utiliser `=`

Astuce

- Dans le mode ESS de Emacs, la touche `_` génère `_<-_`
- Appuyer deux fois pour obtenir le caractère `_`

Astuce

- Pour affecter le résultat d'un calcul **et** voir le résultat, placer l'affectation entre parenthèses
- L'opération d'affectation devient alors une nouvelle expression :

```
> (a <- 2 + 3)
```

```
[1] 5
```

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- Indixage

Caractères permis dans les noms d'objets

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Les lettres a–z, A–Z
- Les chiffres 0–9
- Le point «.»
- «_» est maintenant permis dans R

Règles pour les noms d'objets

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Les noms d'objets ne peuvent commencer par un chiffre
- Le S est sensible à la casse : `foo`, `Foo` et `F00` sont trois objets distincts
- Moyen simple d'éviter des erreurs liées à la casse : employer seulement des lettres minuscules

Noms déjà utilisés et réservés

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Éviter d'utiliser les noms
c, q, t, C, D, I, diff, length, mean, pi, range,
var
- Noms réservés :
Inf, NA, NaN, NULL, TRUE, FALSE
break, else, for, function, if, in, next,
repeat, return, while

TRUE et FALSE

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Les variables T et F prennent par défaut les valeurs TRUE et FALSE, mais peuvent être réaffectées :

```
> T
```

```
[1] TRUE
```

```
> TRUE <- 3
```

Erreur dans TRUE <- 3 : membre gauche de
l'assignation (do_set) incorrect

```
> (T <- 3)
```

```
[1] 3
```

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- **Les objets S**
 - Vecteurs
 - Matrices et tableaux
 - Listes
 - Data frames
 - Indixage

Tout est un objet

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Tout dans le langage S est un objet, même les fonctions et les opérateurs
- Les objets possèdent au minimum un **mode** et une **longueur**
- Certains objets sont également dotés d'un ou plusieurs **attributs**

Modes et types de données

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Prescrit ce qu'un objet peut contenir

- Obtenu avec la fonction `mode` :

```
> v <- c(1, 2, 5, 9)
```

```
> mode(v)
```

```
[1] "numeric"
```

- Principaux modes :

| | |
|------------|--------------------------------|
| numeric | nombres réels |
| complex | nombres complexes |
| logical | valeurs booléennes (vrai/faux) |
| character | chaînes de caractères |
| function | fonction |
| list | données quelconques |
| expression | expressions non évaluées |

Longueur

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Égale au nombre d'éléments que contient un objet
- Obtenue avec la fonction `length` :

```
> length(v)
```

```
[1] 4
```


Longueur (suite)

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Longueur d'une chaîne de caractères est toujours 1
- Un objet de mode character peut contenir plusieurs chaînes de caractères :

```
> v <- "actuariat"
```

```
> length(v)
```

```
[1] 1
```

```
> v <- c("a", "c", "t", "u", "a", "r",  
+       "i", "a", "t")
```

```
> length(v)
```

```
[1] 9
```

Longueur (suite)

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- La longueur d'un objet peut être 0

- Contenant vide :

```
> v <- numeric(0)
```

```
> length(v)
```

```
[1] 0
```

L'objet spécial NULL

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- NULL représente «rien», ou le vide
- Son mode est NULL
- Sa longueur est 0
- Différent d'un objet vide :
 - un objet de longueur 0 est un contenant vide
 - NULL est «pas de contenant»
- Fonction `is.null` pour tester si un objet est NULL

L'objet spécial NA

Introduction
à la pro-
grammation
en S

Vincent
Goulet

Bases du
langage S

- Utilisé pour représenter les données manquantes ou l'absence de données
- Son mode est `logical`
- **Toute** opération impliquant NA a comme résultat NA
- Certaines fonctions (par ex. `sum`, `mean`) peuvent éliminer les données manquantes avant de faire un calcul
- Fonction `is.na` pour tester si un objet est NA

Valeurs spéciales IEEE 754

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- `Inf` : $+\infty$
- `-Inf` : $-\infty$
- `NaN` : $\frac{0}{0}$, $\infty - \infty$, ...
- Fonctions `is.finite`, `is.infinite`, `is.nan`

Attributs

- Éléments d'information liés à un objet
- Principaux attributs

| | |
|-----------------------|----------------------------------------------------|
| <code>class</code> | affecte le comportement d'un objet |
| <code>dim</code> | dimensions des matrices et tableaux |
| <code>dimnames</code> | étiquettes des dimensions des matrices et tableaux |
| <code>names</code> | étiquettes des éléments d'un objet |

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- **Vecteurs**
- Matrices et tableaux
- Listes
- Data frames
- Indixage

En S, tout est un vecteur

- Dans un vecteur simple (*atomic*), tous les éléments sont du même mode
- Possible (et souvent souhaitable) de donner une étiquette aux éléments d'un vecteur :

```
> (v <- c(a = 1, b = 2, c = 5))
```

```
a b c  
1 2 5
```

```
> v <- c(1, 2, 5)
```

```
> names(v) <- c("a", "b", "c")
```

```
> v
```

```
a b c  
1 2 5
```


Fonctions de base pour créer des vecteurs

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- `c` (concaténation)
- `numeric` (vecteur de mode `numeric`)
- `logical` (vecteur de mode `logical`)
- `character` (vecteur de mode `character`).

Indiçage

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Se fait avec []
- Extraction d'un élément par
 - sa position
 - son étiquette (plus sûr)

```
> v[3]
```

```
c
```

```
5
```

```
> v["c"]
```

```
c
```

```
5
```

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- **Matrices et tableaux**
- Listes
- Data frames
- Indixage

Une matrice est un vecteur

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Les matrices et tableaux sont des vecteurs dotés d'un attribut `dim`
- À l'interne, une matrice est donc stockée sous forme de vecteur
- Fonction de base pour créer des matrices est `matrix`
- Fonction de base pour créer des tableaux est `array`

Remplissage d'une matrice

Introduction
à la pro-
grammation
en S

Vincent
Goulet

Bases du
langage S

- **Important** : les matrices et tableaux sont remplis en faisant d'abord varier la première dimension, puis la seconde, etc.

```
> matrix(1:6, nrow = 2, ncol = 3)
```

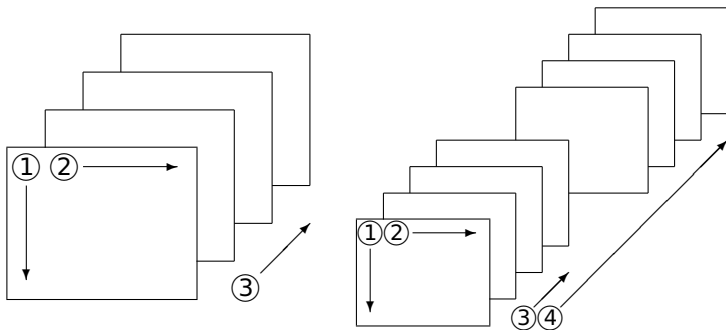
| | [,1] | [,2] | [,3] |
|------|------|------|------|
| [1,] | 1 | 3 | 5 |
| [2,] | 2 | 4 | 6 |

Remplissage des tableaux

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S



Indiçage d'une matrice

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Extraction d'une éléments par sa position (ligne, colonne) dans la matrice, ou encore par sa position dans le vecteur sous-jacent

```
> (m <- matrix(c(40, 80, 45, 21, 55, 32),  
+               nrow = 2))
```

```
      [,1] [,2] [,3]  
[1,]   40   45   55  
[2,]   80   21   32
```

```
> m[1, 2]
```

```
[1] 45
```

```
> m[3]
```

```
[1] 45
```

Fusion verticale de matrices

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Fonction `rbind` permet de fusionner verticalement deux matrices (ou plus) ayant le même nombre de colonnes

```
> n <- matrix(1:9, nrow = 3)  
> rbind(m, n)
```

| | [,1] | [,2] | [,3] |
|------|------|------|------|
| [1,] | 40 | 45 | 55 |
| [2,] | 80 | 21 | 32 |
| [3,] | 1 | 4 | 7 |
| [4,] | 2 | 5 | 8 |
| [5,] | 3 | 6 | 9 |

Fusion horizontale de matrices

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Fonction `cbind` permet de fusionner horizontalement deux matrices (ou plus) ayant le même nombre de lignes

```
> n <- matrix(1:4, nrow = 2)  
> cbind(m, n)
```

| | [,1] | [,2] | [,3] | [,4] | [,5] |
|------|------|------|------|------|------|
| [1,] | 40 | 45 | 55 | 1 | 3 |
| [2,] | 80 | 21 | 32 | 2 | 4 |

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- Matrices et tableaux
- **Listes**
- Data frames
- Indixage

Un vecteur très général

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Vecteur spécial dont les éléments peuvent être de n'importe quel mode
- Permet d'emboîter des listes \Rightarrow objet *récuratif*
- Fonction de base pour créer des listes est `list`
- Conseil : nommer les éléments d'une liste !

Indiçage d'une liste

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Simples crochets [] retournent une liste
- Doubles crochets [[]] retournent l'élément seul
- Par l'étiquette avec liste\$element

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- Indixage

Un peu d'une liste et d'une matrice

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

- Liste de classe `data.frame` dont tous les éléments sont de la même longueur
- Visuellement identique à une matrice
- Plus général qu'une matrice : les colonnes peuvent être de modes différents
- Fonctions `data.frame` ou `as.data.frame`

Sommaire

Introduction
à la programmation
en S

Vincent
Goulet

Bases du
langage S

1 Bases du langage S

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- **Indiçage**

1. Avec un vecteur d'entiers positifs

Éléments se trouvant aux positions correspondant
aux entiers sont extraits, dans l'ordre :

```
> letters[c(1:3, 22, 5)]
```

```
[1] "a" "b" "c" "v" "e"
```


2. Avec un vecteur d'entiers négatifs

Éléments se trouvant aux positions correspondant aux entiers négatifs sont **éliminés** :

```
> letters[c(-(1:3), -5, -22)]
```

```
[1] "d" "f" "g" "h" "i" "j" "k" "l" "m"  
[10] "n" "o" "p" "q" "r" "s" "t" "u" "w"  
[19] "x" "y" "z"
```

3. Avec un vecteur booléen

- Vecteur d'indilage doit être de la même longueur que le vecteur indicé
- Éléments correspondant à TRUE sont **extraits**, ceux correspondant FALSE sont **éliminés** :

```
> letters > "f" & letters < "q"
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE  
[7] TRUE TRUE TRUE TRUE TRUE TRUE  
[13] TRUE TRUE TRUE TRUE FALSE FALSE  
[19] FALSE FALSE FALSE FALSE FALSE FALSE  
[25] FALSE FALSE
```

```
> letters[letters > "f" & letters < "q"]
```

```
[1] "g" "h" "i" "j" "k" "l" "m" "n" "o"  
[10] "p"
```

4. Avec une chaîne de caractères

- À condition que les éléments soient nommés :
> *x <- c(Rouge = 2, Bleu = 4, Vert = 9,*
+ Jaune = -5)
> *x[c("Bleu", "Jaune")]*

Bleu Jaune
4 -5