

Introduction à la programmation en S

Vincent Goulet

École d'actuariat
Université Laval

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Chapitre 1

PRÉSENTATION DU LANGAGE S

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Le langage S
- Les moteurs S
- Où trouver de la documentation
- Interfaces pour S-Plus et R
- Installation de Emacs avec ESS
- Démarrer et quitter S-Plus ou R
- Stratégies de travail
- Gestion des projets ou environnements de travail
- Consulter l'aide en ligne

Le langage S

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Le S est un langage pour «programmer avec des données» développé chez Bell Laboratories (anciennement propriété de AT&T, maintenant de Lucent Technologies).

- Pas seulement un «autre» environnement statistique, mais bien un langage de programmation complet et autonome.
- Inspiré de plusieurs langages, dont l'APL et le Lisp :
 - interprété (et non compilé) ;
 - sans déclaration obligatoire des variables ;
 - basé sur la notion de vecteur ;
 - particulièrement puissant pour les applications mathématiques et statistiques (et donc actuarielles).

Le langage S

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Le S est un langage pour «programmer avec des données» développé chez Bell Laboratories (anciennement propriété de AT&T, maintenant de Lucent Technologies).

- Pas seulement un «autre» environnement statistique, mais bien un langage de programmation complet et autonome.
- Inspiré de plusieurs langages, dont l'APL et le Lisp :
 - interprété (et non compilé) ;
 - sans déclaration obligatoire des variables ;
 - basé sur la notion de vecteur ;
 - particulièrement puissant pour les applications mathématiques et statistiques (et donc actuarielles).

Le langage S

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Le S est un langage pour «programmer avec des données» développé chez Bell Laboratories (anciennement propriété de AT&T, maintenant de Lucent Technologies).

- Pas seulement un «autre» environnement statistique, mais bien un langage de programmation complet et autonome.
- Inspiré de plusieurs langages, dont l'APL et le Lisp :
 - interprété (et non compilé) ;
 - sans déclaration obligatoire des variables ;
 - basé sur la notion de vecteur ;
 - particulièrement puissant pour les applications mathématiques et statistiques (et donc actuarielles).

Les moteurs S

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Il existe quelques «moteurs» ou dialectes du langage S.

- Le plus connu est S-Plus, un logiciel commercial de Insightful Corporation. (Bell Labs octroie à Insightful la licence exclusive de leur système S.)
- R, ou GNU S, est une version libre (*Open Source*) «*not unlike S*».

S-Plus et R constituent tous deux des environnements intégrés de manipulation de données, de calcul et de préparation de graphiques.

Les moteurs S

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Il existe quelques «moteurs» ou dialectes du langage S.

- Le plus connu est S-Plus, un logiciel commercial de Insightful Corporation. (Bell Labs octroie à Insightful la licence exclusive de leur système S.)
- R, ou GNU S, est une version libre (*Open Source*) «*not unlike S*».

S-Plus et R constituent tous deux des environnements intégrés de manipulation de données, de calcul et de préparation de graphiques.

Où trouver de la documentation

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- S-Plus est livré avec quatre livres, mais aucun ne s'avère vraiment utile pour apprendre le langage S.
- Plusieurs livres — en versions papier ou électronique, gratuits ou non — ont été publiés sur S-Plus et/ou R. On trouvera des listes exhaustives dans les sites de Insightful et du projet R.

Où trouver de la documentation

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- S-Plus est livré avec quatre livres, mais aucun ne s'avère vraiment utile pour apprendre le langage S.
- Plusieurs livres — en versions papier ou électronique, gratuits ou non — ont été publiés sur S-Plus et/ou R. On trouvera des listes exhaustives dans les sites de Insightful et du projet R.

Interfaces pour S-Plus et R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R sont d'abord et avant tout des applications en ligne de commande (`sape.exe` et `rterm.exe` sous Windows).

- S-Plus possède toutefois une interface graphique élaborée permettant d'utiliser le logiciel sans trop connaître le langage de programmation.
- R dispose également d'une interface graphique rudimentaire sous Windows et Mac OS.
- L'édition sérieuse de code S bénéficie cependant grandement d'un bon éditeur de texte.

Interfaces pour S-Plus et R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R sont d'abord et avant tout des applications en ligne de commande (`sape.exe` et `rterm.exe` sous Windows).

- S-Plus possède toutefois une interface graphique élaborée permettant d'utiliser le logiciel sans trop connaître le langage de programmation.
- R dispose également d'une interface graphique rudimentaire sous Windows et Mac OS.
- L'édition sérieuse de code S bénéficie cependant grandement d'un bon éditeur de texte.

Interfaces pour S-Plus et R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R sont d'abord et avant tout des applications en ligne de commande (`sape.exe` et `rterm.exe` sous Windows).

- S-Plus possède toutefois une interface graphique élaborée permettant d'utiliser le logiciel sans trop connaître le langage de programmation.
- R dispose également d'une interface graphique rudimentaire sous Windows et Mac OS.
- L'édition sérieuse de code S bénéficie cependant grandement d'un bon éditeur de texte.

Interfaces pour S-Plus et R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R sont d'abord et avant tout des applications en ligne de commande (`sape.exe` et `rterm.exe` sous Windows).

- S-Plus possède toutefois une interface graphique élaborée permettant d'utiliser le logiciel sans trop connaître le langage de programmation.
- R dispose également d'une interface graphique rudimentaire sous Windows et Mac OS.
- L'édition sérieuse de code S bénéficie cependant grandement d'un bon éditeur de texte.

- À la question 6.2 de la foire aux questions (FAQ) de R, «Devrais-je utiliser R à l'intérieur de Emacs ?», la réponse est : «Oui, définitivement.»
- Nous partageons cet avis, aussi apprendra-t-on à utiliser S-Plus ou R à l'intérieur de GNU Emacs avec le mode ESS.
- Autre option : WinEdt (partagiciel) avec l'ajout R-WinEdt.

- À la question 6.2 de la foire aux questions (FAQ) de R, «Devrais-je utiliser R à l'intérieur de Emacs ?», la réponse est : «Oui, définitivement.»
- Nous partageons cet avis, aussi apprendra-t-on à utiliser S-Plus ou R à l'intérieur de GNU Emacs avec le mode ESS.
- Autre option : WinEdt (partagiciel) avec l'ajout R-WinEdt.

- À la question 6.2 de la foire aux questions (FAQ) de R, «Devrais-je utiliser R à l'intérieur de Emacs ?», la réponse est : «Oui, définitivement.»
- Nous partageons cet avis, aussi apprendra-t-on à utiliser S-Plus ou R à l'intérieur de GNU Emacs avec le mode ESS.
- Autre option : WinEdt (partagiciel) avec l'ajout R-WinEdt.

Installation de Emacs avec ESS

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour une installation simplifiée de Emacs et ESS, consulter le site Internet <http://vgoulet.act.ulaval.ca/pub/emacs/> On y trouve une version modifiée de GNU Emacs et des instructions d'installation détaillées.
- L'annexe A du document d'accompagnement présente les plus importantes commandes à connaître pour utiliser Emacs et le mode ESS.

Installation de Emacs avec ESS

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour une installation simplifiée de Emacs et ESS, consulter le site Internet <http://vgoulet.act.ulaval.ca/pub/emacs/> On y trouve une version modifiée de GNU Emacs et des instructions d'installation détaillées.
- L'annexe A du document d'accompagnement présente les plus importantes commandes à connaître pour utiliser Emacs et le mode ESS.

Démarrer et quitter S-Plus ou R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour démarrer R à l'intérieur de Emacs :

`M-x R RET`

puis spécifier un dossier de travail. Une console R est ouverte dans un *buffer* nommé **R**.

- Pour démarrer S-Plus sous Windows, consulter l'annexe B du document d'accompagnement.
- Pour quitter, deux options sont disponibles :

- Taper `q()` à la ligne de commande.

- Dans Emacs, faire `C-c C-q`. ESS va alors s'occuper de fermer le processus S ainsi que tous les *buffers* associés à ce processus.

Démarrer et quitter S-Plus ou R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour démarrer R à l'intérieur de Emacs :

`M-x R RET`

puis spécifier un dossier de travail. Une console R est ouverte dans un *buffer* nommé **R**.

- Pour démarrer S-Plus sous Windows, consulter l'annexe B du document d'accompagnement.
- Pour quitter, deux options sont disponibles :

■ Taper `q()` à la ligne de commande.

■ Dans Emacs, faire `C-c C-g`. ESS va alors s'occuper de fermer le processus S ainsi que tous les *buffers* associés à ce processus.

Démarrer et quitter S-Plus ou R

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour démarrer R à l'intérieur de Emacs :

`M-x R RET`

puis spécifier un dossier de travail. Une console R est ouverte dans un *buffer* nommé **R**.

- Pour démarrer S-Plus sous Windows, consulter l'annexe B du document d'accompagnement.
- Pour quitter, deux options sont disponibles :

1 Taper `q()` à la ligne de commande.

2 Dans Emacs, faire `C-c C-q`. ESS va alors s'occuper de fermer le processus S ainsi que tous les *buffers* associés à ce processus.

Démarrer et quitter S-Plus ou R

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour démarrer R à l'intérieur de Emacs :

`M-x R RET`

puis spécifier un dossier de travail. Une console R est ouverte dans un *buffer* nommé **R**.

- Pour démarrer S-Plus sous Windows, consulter l'annexe B du document d'accompagnement.
- Pour quitter, deux options sont disponibles :

1 Taper `q()` à la ligne de commande.

2 Dans Emacs, faire `C-c C-q`. ESS va alors s'occuper de fermer le processus S ainsi que tous les *buffers* associés à ce processus.

Démarrer et quitter S-Plus ou R

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- Pour démarrer R à l'intérieur de Emacs :

`M-x R RET`

puis spécifier un dossier de travail. Une console R est ouverte dans un *buffer* nommé **R**.

- Pour démarrer S-Plus sous Windows, consulter l'annexe B du document d'accompagnement.
- Pour quitter, deux options sont disponibles :
 - 1 Taper `q()` à la ligne de commande.
 - 2 Dans Emacs, faire `C-c C-q`. ESS va alors s'occuper de fermer le processus S ainsi que tous les *buffers* associés à ce processus.

Stratégies de travail

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

Il existe principalement deux façons de travailler avec S-Plus et R.

- 1 Le code est virtuel et les objets sont réels.
- 2 Le code est réel et les objets sont virtuels.

Code virtuel, objets réels

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- C'est l'approche qu'encouragent les interfaces graphiques, mais aussi la moins pratique à long terme.
- On entre des expressions directement à la ligne de commande pour les évaluer immédiatement.
- Les objets créés au cours d'une session de travail sont sauvegardés.
- Par contre, le code utilisé pour créer ces objets est perdu lorsque l'on quitte S-Plus ou R, à moins de sauvegarder celui-ci dans des fichiers.

Code réel, objets virtuels

Introduction à
la programmation
en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

- C'est l'approche que nous favoriserons.
- Le travail se fait essentiellement dans des fichiers de script (de simples fichiers de texte) dans lesquels sont sauvegardées les expressions (parfois complexes !) et le code des fonctions personnelles.
- Les objets sont créés au besoin en exécutant le code.

Emacs permet ici de passer efficacement des fichiers de script à l'exécution du code :

- 1 Démarrer un processus S-Plus (`M-x Splus`) ou R (`M-x R`) et spécifier le dossier de travail.
- 2 Ouvrir un fichier de script avec `C-x C-f`. Pour créer un nouveau fichier, ouvrir un fichier n'existant pas.
- 3 Positionner le curseur sur une expression et faire `C-c C-n` pour l'évaluer.
- 4 Le résultat apparaît dans le *buffer* `*S+6*` ou `*R*`.

Emacs permet ici de passer efficacement des fichiers de script à l'exécution du code :

- 1 Démarrer un processus S-Plus (`M-x Splus`) ou R (`M-x R`) et spécifier le dossier de travail.
- 2 Ouvrir un fichier de script avec `C-x C-f`. Pour créer un nouveau fichier, ouvrir un fichier n'existant pas.
- 3 Positionner le curseur sur une expression et faire `C-c C-n` pour l'évaluer.
- 4 Le résultat apparaît dans le *buffer* `*S+6*` ou `*R*`.

Emacs permet ici de passer efficacement des fichiers de script à l'exécution du code :

- 1 Démarrer un processus S-Plus (`M-x Sque`) ou R (`M-x R`) et spécifier le dossier de travail.
- 2 Ouvrir un fichier de script avec `C-x C-f`. Pour créer un nouveau fichier, ouvrir un fichier n'existant pas.
- 3 Positionner le curseur sur une expression et faire `C-c C-n` pour l'évaluer.
- 4 Le résultat apparaît dans le *buffer* `*S+6*` ou `*R*`.

Emacs permet ici de passer efficacement des fichiers de script à l'exécution du code :

- 1 Démarrer un processus S-Plus (`M-x Sque`) ou R (`M-x R`) et spécifier le dossier de travail.
- 2 Ouvrir un fichier de script avec `C-x C-f`. Pour créer un nouveau fichier, ouvrir un fichier n'existant pas.
- 3 Positionner le curseur sur une expression et faire `C-c C-n` pour l'évaluer.
- 4 Le résultat apparaît dans le *buffer* `*S+6*` ou `*R*`.

Gestion des projets ou environnements de travail

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R ont une manière différente, mais tout aussi particulière de sauvegarder les objets créés au cours d'une session de travail.

- Tous deux doivent travailler dans un dossier et non avec des fichiers individuels.
- Dans S-Plus, tout objet créé au cours d'une session de travail est sauvegardé de façon permanente sur le disque dur dans le sous-dossier `__Data` du dossier de travail.
- Dans R, les objets créés sont conservés en mémoire jusqu'à ce que l'on quitte l'application ou que l'on enregistre le travail avec la commande `save.image()`. L'environnement de travail (*workspace*) est alors sauvegardé dans le fichier `.RData` dans le dossier de travail.

Gestion des projets ou environnements de travail

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R ont une manière différente, mais tout aussi particulière de sauvegarder les objets créés au cours d'une session de travail.

- Tous deux doivent travailler dans un dossier et non avec des fichiers individuels.
- Dans S-Plus, tout objet créé au cours d'une session de travail est sauvegardé de façon permanente sur le disque dur dans le sous-dossier `__Data` du dossier de travail.
- Dans R, les objets créés sont conservés en mémoire jusqu'à ce que l'on quitte l'application ou que l'on enregistre le travail avec la commande `save.image()`. L'environnement de travail (*workspace*) est alors sauvegardé dans le fichier `.RData` dans le dossier de travail.

Gestion des projets ou environnements de travail

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R ont une manière différente, mais tout aussi particulière de sauvegarder les objets créés au cours d'une session de travail.

- Tous deux doivent travailler dans un dossier et non avec des fichiers individuels.
- Dans S-Plus, tout objet créé au cours d'une session de travail est sauvegardé de façon permanente sur le disque dur dans le sous-dossier `__Data` du dossier de travail.
- Dans R, les objets créés sont conservés en mémoire jusqu'à ce que l'on quitte l'application ou que l'on enregistre le travail avec la commande `save.image()`. L'environnement de travail (*workspace*) est alors sauvegardé dans le fichier `.RData` dans le dossier de travail.

Gestion des projets ou environnements de travail

Introduction à
la programmation en
S

Vincent
Goulet

Le langage S

Les moteurs S

Documentation

Interfaces

Emacs et ESS

Démarrer et
quitter

Stratégies de
travail

Gestion des
projets

Consulter
l'aide en ligne

S-Plus et R ont une manière différente, mais tout aussi particulière de sauvegarder les objets créés au cours d'une session de travail.

- Tous deux doivent travailler dans un dossier et non avec des fichiers individuels.
- Dans S-Plus, tout objet créé au cours d'une session de travail est sauvegardé de façon permanente sur le disque dur dans le sous-dossier `__Data` du dossier de travail.
- Dans R, les objets créés sont conservés en mémoire jusqu'à ce que l'on quitte l'application ou que l'on enregistre le travail avec la commande `save.image()`. L'environnement de travail (*workspace*) est alors sauvegardé dans le fichier `.RData` dans le dossier de travail.

Le dossier de travail est déterminé au lancement de l'application.

- Avec Emacs et ESS on doit spécifier le dossier de travail à chaque fois que l'on démarre un processus S-Plus ou R.
- Les interfaces graphiques permettent également de spécifier le dossier de travail.

Le dossier de travail est déterminé au lancement de l'application.

- Avec Emacs et ESS on doit spécifier le dossier de travail à chaque fois que l'on démarre un processus S-Plus ou R.
- Les interfaces graphiques permettent également de spécifier le dossier de travail.

Les rubriques d'aide des diverses fonctions disponibles dans S-Plus et R contiennent une foule d'informations ainsi que des exemples d'utilisation. Leur consultation est tout à fait essentielle.

- Pour consulter la rubrique d'aide de la fonction `foo`, on peut entrer à la ligne de commande

```
> ?foo
```

- Dans Emacs, `C-c C-v foo RET` ouvrira la rubrique d'aide de la fonction `foo` dans un nouveau *buffer*.

Les rubriques d'aide des diverses fonctions disponibles dans S-Plus et R contiennent une foule d'informations ainsi que des exemples d'utilisation. Leur consultation est tout à fait essentielle.

- Pour consulter la rubrique d'aide de la fonction `foo`, on peut entrer à la ligne de commande

```
> ?foo
```

- Dans Emacs, `C-c C-v foo RET` ouvrira la rubrique d'aide de la fonction `foo` dans un nouveau *buffer*.

Les rubriques d'aide des diverses fonctions disponibles dans S-Plus et R contiennent une foule d'informations ainsi que des exemples d'utilisation. Leur consultation est tout à fait essentielle.

- Pour consulter la rubrique d'aide de la fonction `foo`, on peut entrer à la ligne de commande

```
> ?foo
```

- Dans Emacs, `C-c C-v foo RET` ouvrira la rubrique d'aide de la fonction `foo` dans un nouveau *buffer*.

Chapitre 2

BASES DU LANGUAGE S

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- Indiçage

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

■ Commandes S

■ Conventions pour les noms d'objets

■ Les objets S

■ Modes et types de données

■ Longueur

■ Attributs

■ L'objet spécial NA

■ L'objet spécial NULL

■ Vecteurs

■ Matrices et tableaux

■ Listes

■ Data frames

■ Indiçage

Affectations et expressions

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Toute commande S est soit une *affectation*, soit une *expression*.

- Normalement, 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- 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
```

Deux symboles d'affectation à éviter

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

■ L'opérateur =

- peut porter à confusion.

■ Le caractère _

- permis dans S-Plus, mais plus dans R depuis la version 1.8.0
- emploi fortement découragé
- rend le code difficile à lire
- dans le mode ESS de Emacs, taper ce caractère génère carrément `_<-_`.

Deux symboles d'affectation à éviter

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

■ L'opérateur =

- peut porter à confusion.

■ Le caractère _

- permis dans S-Plus, mais plus dans R depuis la version 1.8.0
- emploi fortement découragé
- rend le code difficile à lire
- dans le mode ESS de Emacs, taper ce caractère génère carrément `_<-_`.

Astuce

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Pour affecter le résultat d'un calcul dans un objet et en même temps voir ce 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

■ Commandes S

■ Conventions pour les noms d'objets

■ Les objets S

■ Modes et types de données

■ Longueur

■ Attributs

■ L'objet spécial NA

■ L'objet spécial NULL

■ Vecteurs

■ Matrices et tableaux

■ Listes

■ Data frames

■ Indiçage

Caractères permis dans les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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

Caractères permis dans les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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

Caractères permis dans les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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

Caractères permis dans les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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

Règles pour les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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

Règles pour les noms d'objets

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Les noms d'objets ne peuvent commencer par un chiffre.
- Le S est sensible à la casse : `f00`, `F00` et `FOO` 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indice

- Certains noms sont utilisés par le système, aussi vaut-il mieux éviter de les utiliser. En particulier, éviter d'utiliser

`c, q, t, C, D, I, diff, length, mean, pi, range, var.`

- Certains mots sont réservés pour le système et il est interdit de les utiliser comme nom d'objet :

`Inf, NA, NaN, NULL`

`break, else, for, function, if, in, next, repeat,
return, while.`

- Dans S-Plus 6.1 et plus, `T` et `TRUE` (vrai), ainsi que `F` et `FALSE` (faux) sont également des noms réservés.

Noms déjà utilisés et réservés

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indice

- Certains noms sont utilisés par le système, aussi vaut-il mieux éviter de les utiliser. En particulier, éviter d'utiliser

`c, q, t, C, D, I, diff, length, mean, pi, range, var.`

- Certains mots sont réservés pour le système et il est interdit de les utiliser comme nom d'objet :

`Inf, NA, NaN, NULL`

`break, else, for, function, if, in, next, repeat,`
`return, while.`

- Dans S-Plus 6.1 et plus, `T` et `TRUE` (vrai), ainsi que `F` et `FALSE` (faux) sont également des noms réservés.

Noms déjà utilisés et réservés

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indice

- Certains noms sont utilisés par le système, aussi vaut-il mieux éviter de les utiliser. En particulier, éviter d'utiliser

`c, q, t, C, D, I, diff, length, mean, pi, range, var.`

- Certains mots sont réservés pour le système et il est interdit de les utiliser comme nom d'objet :

`Inf, NA, NaN, NULL`

`break, else, for, function, if, in, next, repeat,
return, while.`

- Dans S-Plus 6.1 et plus, `T` et `TRUE` (vrai), ainsi que `F` et `FALSE` (faux) sont également des noms réservés.

TRUE et FALSE dans R

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Dans R, les noms TRUE et FALSE sont également réservés.
- Les variables T et F prennent par défaut les valeurs TRUE et FALSE, respectivement, 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
```

TRUE et FALSE dans R

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Dans R, les noms TRUE et FALSE sont également réservés.
- Les variables T et F prennent par défaut les valeurs TRUE et FALSE, respectivement, 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

■ Commandes S

■ Conventions pour les noms d'objets

■ Les objets S

■ Modes et types de données

■ Longueur

■ Attributs

■ L'objet spécial NA

■ L'objet spécial NULL

■ Vecteurs

■ Matrices et tableaux

■ Listes

■ Data frames

■ Indiçage

Tout est un objet

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Tout dans le langage S est un objet, même les fonctions et les opérateurs.
- Les objets possèdent au minimum un **mode** et une **longueur**.

Mode et longueur

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Le mode d'un objet est obtenu avec la fonction `mode`.

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

```
> mode(v)
```

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

- La longueur d'un objet est obtenue avec la fonction `length`.

```
> length(v)
```

```
[1] 4
```

- Certains objets sont également dotés d'un ou plusieurs **attributs**.

Modes et types de données

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Le mode prescrit ce qu'un objet peut contenir.
- Un objet ne peut donc avoir qu'un seul mode.
- Modes disponibles en S :

<code>numeric</code>	nombres réels
<code>complex</code>	nombres complexes
<code>logical</code>	valeurs booléennes (vrai/faux)
<code>character</code>	chaînes de caractères
<code>function</code>	fonction
<code>list</code>	données quelconques

Longueur

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indice

La longueur d'un objet est égale au nombre d'éléments qu'il contient.

- La longueur d'une chaîne de caractères est toujours 1. Un objet de mode `character` doit contenir plusieurs chaînes de caractères pour que sa longueur soit supérieure à 1.

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

```
> length(v)
```

```
[1] 1
```

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

```
> length(v)
```

```
[1] 9
```

Longueur

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indice

La longueur d'un objet est égale au nombre d'éléments qu'il contient.

- La longueur d'une chaîne de caractères est toujours 1. Un objet de mode `character` doit contenir plusieurs chaînes de caractères pour que sa longueur soit supérieure à 1.

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

```
> length(v)
```

```
[1] 1
```

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

```
> length(v)
```

```
[1] 9
```

Objet vide

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Un objet peut être de longueur 0.
- Doit alors être interprété comme un contenant vide.

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

[1] 0
```

Objet vide

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Un objet peut être de longueur 0.
- Doit alors être interprété comme un contenant vide.

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

```
> length(v)
```

```
[1] 0
```

Attributs

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Éléments d'information additionnels liés à cet objet.
- Attributs les plus fréquemment rencontrés :

<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

L'objet spécial NA

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

NA est fréquemment utilisé pour représenter les données manquantes.

- Son mode est `logical`.
- Toute opération impliquant une donnée NA a comme résultat NA.
- Certaines fonctions (`sum`, `mean`, par exemple), ont par conséquent un argument `na.rm` qui, lorsque `TRUE`, élimine les données manquantes avant de faire un calcul.
- La fonction `is.na` permet de tester si les éléments d'un objet sont NA ou non.

L'objet spécial NA

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

NA est fréquemment utilisé pour représenter les données manquantes.

- Son mode est `logical`.
- Toute opération impliquant une donnée NA a comme résultat NA.
- Certaines fonctions (`sum`, `mean`, par exemple), ont par conséquent un argument `na.rm` qui, lorsque `TRUE`, élimine les données manquantes avant de faire un calcul.
- La fonction `is.na` permet de tester si les éléments d'un objet sont NA ou non.

L'objet spécial NA

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

NA est fréquemment utilisé pour représenter les données manquantes.

- Son mode est `logical`.
- Toute opération impliquant une donnée NA a comme résultat NA.
- Certaines fonctions (`sum`, `mean`, par exemple), ont par conséquent un argument `na.rm` qui, lorsque `TRUE`, élimine les données manquantes avant de faire un calcul.
- La fonction `is.na` permet de tester si les éléments d'un objet sont NA ou non.

L'objet spécial NA

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

NA est fréquemment utilisé pour représenter les données manquantes.

- Son mode est `logical`.
- Toute opération impliquant une donnée NA a comme résultat NA.
- Certaines fonctions (`sum`, `mean`, par exemple), ont par conséquent un argument `na.rm` qui, lorsque `TRUE`, élimine les données manquantes avant de faire un calcul.
- La fonction `is.na` permet de tester si les éléments d'un objet sont NA ou non.

L'objet spécial NA

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

NA est fréquemment utilisé pour représenter les données manquantes.

- Son mode est `logical`.
- Toute opération impliquant une donnée NA a comme résultat NA.
- Certaines fonctions (`sum`, `mean`, par exemple), ont par conséquent un argument `na.rm` qui, lorsque `TRUE`, élimine les données manquantes avant de faire un calcul.
- La fonction `is.na` permet de tester si les éléments d'un objet sont NA ou non.

L'objet spécial NULL

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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 constant vide,
 - NULL est une constante.
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

L'objet spécial NULL

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

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».
- La fonction `is.null` teste si un objet est NULL ou non.

Sommaire

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- Indiçage

En S, tout est un vecteur

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Dans un vecteur simple, tous les éléments doivent être du même mode.
- Il est possible (et souvent souhaitable) de donner une étiquette à chacun des 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
```

Et comment crée-t-on ces vecteurs ?

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Les fonctions de base pour créer des vecteurs sont

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

Indiçage d'un vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Se fait avec `[]`.

- On peut extraire un élément d'un vecteur par

- sa position ou

- son étiquette, si elle existe (auquel cas cette approche est beaucoup plus sûre).

```
> v[3]
```

```
c
```

```
5
```

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

```
c
```

```
5
```

Indiçage d'un vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Se fait avec `[]`.
- On peut extraire un élément d'un vecteur par
 - sa position ou
 - son étiquette, si elle existe (auquel cas cette approche est beaucoup plus sûre).

```
> v[3]
```

```
c
```

```
5
```

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

```
c
```

```
5
```

Indiçage d'un vecteur

Introduction à
la programmation
en

S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Se fait avec `[]`.
- On peut extraire un élément d'un vecteur par
 - sa position ou
 - son étiquette, si elle existe (auquel cas cette approche est beaucoup plus sûre).

```
> v[3]
```

```
c
```

```
5
```

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

```
c
```

```
5
```

Indiçage d'un vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Se fait avec `[]`.
- On peut extraire un élément d'un vecteur par
 - sa position ou
 - son étiquette, si elle existe (auquel cas cette approche est beaucoup plus sûre).

```
> v[3]
```

```
c
```

```
5
```

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

```
c
```

```
5
```


Indiçage d'un vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Se fait avec `[]`.
- On peut extraire un élément d'un vecteur par
 - sa position ou
 - son étiquette, si elle existe (auquel cas cette approche est beaucoup plus sûre).

```
> v[3]
```

```
c
```

```
5
```

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

```
c
```

```
5
```

Sommaire

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- **Matrices et tableaux**
- Listes
- Data frames
- Indiçage

Une matrice est un vecteur ?

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une matrice ou, de façon plus générale, un tableau (*array*) n'est rien d'autre qu'un vecteur doté d'un attribut `dim`.

- À l'interne, une matrice est donc stockée sous forme de vecteur.
- La fonction de base pour créer des matrices est `matrix`.
- La fonction de base pour créer des tableaux est `array`.

Une matrice est un vecteur ?

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une matrice ou, de façon plus générale, un tableau (*array*) n'est rien d'autre qu'un vecteur doté d'un attribut `dim`.

- À l'interne, une matrice est donc stockée sous forme de vecteur.
- La fonction de base pour créer des matrices est `matrix`.
- La fonction de base pour créer des tableaux est `array`.

Une matrice est un vecteur ?

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une matrice ou, de façon plus générale, un tableau (*array*) n'est rien d'autre qu'un vecteur doté d'un attribut `dim`.

- À l'interne, une matrice est donc stockée sous forme de vecteur.
- La fonction de base pour créer des matrices est `matrix`.
- La fonction de base pour créer des tableaux est `array`.

Une matrice est un vecteur ?

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une matrice ou, de façon plus générale, un tableau (*array*) n'est rien d'autre qu'un vecteur doté d'un attribut `dim`.

- À l'interne, une matrice est donc stockée sous forme de vecteur.
- La fonction de base pour créer des matrices est `matrix`.
- La fonction de base pour créer des tableaux est `array`.

Remplissage d'une matrice

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- **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

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

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

Indiçage d'une matrice

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- On extrait les éléments d'une matrice en précisant leurs positions sous la forme (ligne, colonne) dans la matrice, ou encore leurs positions dans le vecteur sous-jacent.

```
> (m <- matrix(c(40, 80, 45, 21, 55, 32),  
+             nrow = 2, ncol = 3))
```

	[,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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- La 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- La 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

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- Matrices et tableaux
- **Listes**
- Data frames
- Indiçage

Un vecteur très général

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une liste est un type de vecteur spécial dont les éléments peuvent être de n'importe quel mode, y compris le mode `list` (ce qui permet d'emboîter des listes).

- La fonction de base pour créer des listes est `list`.
- Généralement préférable de nommer les éléments d'une liste : plus simple et sûr d'extraire les éléments par leur étiquette.

Un vecteur très général

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une liste est un type de vecteur spécial dont les éléments peuvent être de n'importe quel mode, y compris le mode `list` (ce qui permet d'emboîter des listes).

- La fonction de base pour créer des listes est `list`.
- Généralement préférable de nommer les éléments d'une liste : plus simple et sûr d'extraire les éléments par leur étiquette.

Un vecteur très général

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Une liste est un type de vecteur spécial dont les éléments peuvent être de n'importe quel mode, y compris le mode `list` (ce qui permet d'emboîter des listes).

- La fonction de base pour créer des listes est `list`.
- Généralement préférable de nommer les éléments d'une liste : plus simple et sûr d'extraire les éléments par leur étiquette.

Indiçage d'une liste

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'extraction des éléments d'une liste peut se faire de deux façons :

- 1 avec des doubles crochets `[[]]`
- 2 par leur étiquette avec
`nom.liste$etiquette.element.`

Indiçage d'une liste

Introduction à
la programmation
en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'extraction des éléments d'une liste peut se faire de deux façons :

1 avec des doubles crochets `[[]]`

2 par leur étiquette avec
`nom.liste$etiquette.element.`

Indiçage d'une liste

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'extraction des éléments d'une liste peut se faire de deux façons :

- 1 avec des doubles crochets `[[]]`
- 2 par leur étiquette avec
`nom.liste$etiquette.element.`

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- Matrices et tableaux
- Listes
- **Data frames**
- Indiçage

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Liste ou matrice ? Un peu des deux !

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- L'analyse de données — la régression linéaire, par exemple — repose sur les *data frames*.
- Liste de classe `data.frame` dont tous les éléments sont de la même longueur.
- Généralement représenté sous forme d'un tableau à deux dimensions (visuellement similaire à une matrice).
- Plus général qu'une matrice puisque les colonnes peuvent être de modes différents (`numeric`, `complex`, `character` ou `logical`).
- Créé avec la fonction `data.frame` ou `as.data.frame`.
- Moins important lors de l'apprentissage du langage de programmation.

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

- Commandes S
- Conventions pour les noms d'objets
- Les objets S
 - Modes et types de données
 - Longueur
 - Attributs
 - L'objet spécial NA
 - L'objet spécial NULL
- Vecteurs
- Matrices et tableaux
- Listes
- Data frames
- **Indiçage**

Quatre façons d'indicer un vecteur

Introduction à
la programmation en
S

Vincent
Goulet

Commandes
S

Conventions
pour les noms
d'objets

Les objets S

Vecteurs

Matrices et
tableaux

Listes

Data frames

Indiçage

Dans tous les cas, l'indiçage se fait avec des crochets [].

- 1 Avec un vecteur d'entiers positifs. Les éléments se trouvant aux positions correspondant aux entiers sont extraits du vecteur, dans l'ordre. C'est la technique la plus courante.

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

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

- 2 Avec un vecteur d'entiers négatifs. Les éléments se trouvant aux positions correspondant aux entiers négatifs sont alors **éliminés** du vecteur.

```
> 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. Le vecteur d'indiçage doit alors être de la même longueur que le vecteur indicé. Les éléments correspondant à une valeur TRUE sont **extraits** du vecteur, alors que ceux correspondant à 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. Utile pour extraire les éléments d'un vecteur à condition que ceux-ci soient nommés.

```
> x <- c(Rouge = 2, Bleu = 4, Vert = 9,  
+ Jaune = -5)
```

```
> x[c("Bleu", "Jaune")]
```

```
Bleu Jaune
```

```
4      -5
```

Chapitre 3

OPÉRATEURS ET FONCTIONS

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Opérations arithmétiques
- Opérateurs
- Appels de fonctions
 - Exemple
- Quelques fonctions utiles
 - Manipulation de vecteurs
 - Recherche d'éléments dans un vecteur
 - Arrondi
 - Sommaires et statistiques descriptives
 - Sommaires cumulatifs et comparaisons élément par élément
 - Opérations sur les matrices
 - Produit extérieur
- Structures de contrôle
 - Exécution conditionnelle
 - Boucles

Une liste non exhaustive

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Principaux opérateurs arithmétiques, fonctions mathématiques et structures de contrôles offertes par le S.
- Liste loin d'être exhaustive.
- Consulter aussi la section `See Also` des rubriques d'aide.

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

■ Opérations arithmétiques

■ Opérateurs

■ Appels de fonctions

■ Exemple

■ Quelques fonctions utiles

■ Manipulation de vecteurs

■ Recherche d'éléments dans un vecteur

■ Arrondi

■ Sommaires et statistiques descriptives

■ Sommaires cumulatifs et comparaisons élément par élément

■ Opérations sur les matrices

■ Produit extérieur

■ Structures de contrôle

■ Exécution conditionnelle

■ Boucles

L'unité de base est le vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Les opérations sur les vecteurs sont effectuées
élément par élément :

```
> c(1, 2, 3) + c(4, 5, 6)
```

```
[1] 5 7 9
```

```
> 1:3 * 4:6
```

```
[1] 4 10 18
```

Recyclage des vecteurs

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Si les vecteurs impliqués dans une expression arithmétique ne sont pas de la même longueur, les plus courts sont **recyclés**.
- Particulièrement apparent avec les vecteurs de longueur 1 :

```
> 1:10 + 2
```

```
[1] 3 4 5 6 7 8 9 10 11 12
```

```
> 1:10 + rep(2, 10)
```

```
[1] 3 4 5 6 7 8 9 10 11 12
```

Longueur du plus long vecteur multiple de celle des autres vecteurs

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Les vecteurs les plus courts sont recyclés un nombre entier de fois :

```
> 1:10 + 1:5 + c(2, 4)
```

```
[1] 4 8 8 12 12 11 11 15 15 19
```

```
> 1:10 + rep(1:5, 2) + rep(c(2, 4), 5)
```

```
[1] 4 8 8 12 12 11 11 15 15 19
```

Sinon...

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Recyclage un nombre fractionnaire de fois et un avertissement est affiché :

```
> 1:10 + c(2, 4, 6)
```

```
[1] 3 6 9 6 9 12 9 12 15 12
```

Message d'avis :

la longueur de l'objet le plus long n'est
pas un multiple de la longueur de l'objet
le plus court in: 1:10 + c(2, 4, 6)

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Opérations arithmétiques

- **Opérateurs**

- Appels de fonctions

 - Exemple

- Quelques fonctions utiles

 - Manipulation de vecteurs

 - Recherche d'éléments dans un vecteur

 - Arrondi

 - Sommaires et statistiques descriptives

 - Sommaires cumulatifs et comparaisons élément par élément

 - Opérations sur les matrices

 - Produit extérieur

- Structures de contrôle

 - Exécution conditionnelle

 - Boucles

Opérateurs mathématiques et logiques les plus fréquemment employés

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

Ordre décroissant de priorité des opérations.

$^$ ou $**$

puissance

$-$

changement de signe

$*$ /

multiplication, division

$+$ $-$

addition, soustraction

$\% * \% \% \% \% / \%$

produit matriciel, modulo, division
entière

$< <= == >= > !=$

plus petit, plus petit ou égal, égal,
plus grand ou égal, plus grand, dif-
férent de

$!$

négation logique

$\& |$

«et» logique, «ou» logique

Sommaire

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Opérations arithmétiques

- Opérateurs

- Appels de fonctions

 - Exemple

- Quelques fonctions utiles

 - Manipulation de vecteurs

 - Recherche d'éléments dans un vecteur

 - Arrondi

 - Sommaires et statistiques descriptives

 - Sommaires cumulatifs et comparaisons élément par élément

 - Opérations sur les matrices

 - Produit extérieur

- Structures de contrôle

 - Exécution conditionnelle

 - Boucles

Comment spécifier les arguments d'une fonction

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Pas de limite pratique au nombre d'arguments.
- Arguments peuvent être spécifiés dans l'ordre établi dans la définition de la fonction.
- Plus prudent et **fortement recommandé** de spécifier les arguments par leur nom, surtout après les deux ou trois premiers arguments.
- Nécessaire de nommer les arguments s'ils ne sont pas appelés dans l'ordre.
- Certains arguments ont une valeur par défaut qui sera utilisée si l'argument n'est pas spécifié.

Comment spécifier les arguments d'une fonction

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Pas de limite pratique au nombre d'arguments.
- Arguments peuvent être spécifiés dans l'ordre établi dans la définition de la fonction.
- Plus prudent et **fortement recommandé** de spécifier les arguments par leur nom, surtout après les deux ou trois premiers arguments.
- Nécessaire de nommer les arguments s'ils ne sont pas appelés dans l'ordre.
- Certains arguments ont une valeur par défaut qui sera utilisée si l'argument n'est pas spécifié.

Comment spécifier les arguments d'une fonction

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Pas de limite pratique au nombre d'arguments.
- Arguments peuvent être spécifiés dans l'ordre établi dans la définition de la fonction.
- Plus prudent et **fortement recommandé** de spécifier les arguments par leur nom, surtout après les deux ou trois premiers arguments.
- Nécessaire de nommer les arguments s'ils ne sont pas appelés dans l'ordre.
- Certains arguments ont une valeur par défaut qui sera utilisée si l'argument n'est pas spécifié.

Comment spécifier les arguments d'une fonction

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Pas de limite pratique au nombre d'arguments.
- Arguments peuvent être spécifiés dans l'ordre établi dans la définition de la fonction.
- Plus prudent et **fortement recommandé** de spécifier les arguments par leur nom, surtout après les deux ou trois premiers arguments.
- Nécessaire de nommer les arguments s'ils ne sont pas appelés dans l'ordre.
- Certains arguments ont une valeur par défaut qui sera utilisée si l'argument n'est pas spécifié.

Comment spécifier les arguments d'une fonction

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Pas de limite pratique au nombre d'arguments.
- Arguments peuvent être spécifiés dans l'ordre établi dans la définition de la fonction.
- Plus prudent et **fortement recommandé** de spécifier les arguments par leur nom, surtout après les deux ou trois premiers arguments.
- Nécessaire de nommer les arguments s'ils ne sont pas appelés dans l'ordre.
- Certains arguments ont une valeur par défaut qui sera utilisée si l'argument n'est pas spécifié.

Exemple

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

Définition de la fonction `matrix` :

```
matrix(data=NA, nrow=1, ncol=1,  
        byrow=FALSE, dimnames=NULL)
```

- Chaque argument a une valeur par défaut (ce n'est pas toujours le cas).
- Ainsi, un appel à `matrix` sans argument résulte en

```
> matrix()  
      [,1]  
[1,]    NA
```

Exemple

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

Définition de la fonction `matrix` :

```
matrix(data=NA, nrow=1, ncol=1,  
        byrow=FALSE, dimnames=NULL)
```

- Chaque argument a une valeur par défaut (ce n'est pas toujours le cas).
- Ainsi, un appel à `matrix` sans argument résulte en

```
> matrix()
```

```
      [,1]  
[1,]    NA
```

Exemple

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

Définition de la fonction `matrix` :

```
matrix(data=NA, nrow=1, ncol=1,  
        byrow=FALSE, dimnames=NULL)
```

- Chaque argument a une valeur par défaut (ce n'est pas toujours le cas).
- Ainsi, un appel à `matrix` sans argument résulte en

```
> matrix()
```

```
      [,1]  
[1,]    NA
```


- Appel plus élaboré utilisant tous les arguments. Le premier argument est rarement nommé.

```
> matrix(1:6, nrow = 2, ncol = 3,  
+ byrow = TRUE,  
+ dimnames = list(c("Gauche", "Droit"),  
+ c("Rouge", "Vert", "Bleu")))
```

	Rouge	Vert	Bleu
Gauche	1	2	3
Droit	4	5	6

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

■ Opérations arithmétiques

■ Opérateurs

■ Appels de fonctions

■ Exemple

■ Quelques fonctions utiles

■ Manipulation de vecteurs

■ Recherche d'éléments dans un vecteur

■ Arrondi

■ Sommaires et statistiques descriptives

■ Sommaires cumulatifs et comparaisons élément par élément

■ Opérations sur les matrices

■ Produit extérieur

■ Structures de contrôle

■ Exécution conditionnelle

■ Boucles

Système de classement des fonctions

Introduction à
la programmation en
S

S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

■ Diffère entre S-Plus et R.

- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Système de classement des fonctions

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Diffère entre S-Plus et R.
- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Système de classement des fonctions

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Diffère entre S-Plus et R.
- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Système de classement des fonctions

Introduction à
la programmation
en

S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Diffère entre S-Plus et R.
- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Système de classement des fonctions

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Diffère entre S-Plus et R.
- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Système de classement des fonctions

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Diffère entre S-Plus et R.
- Dans S-Plus, les fonctions sont classées dans des **sections** d'une bibliothèque (*library*).
- Dans R, un ensemble de fonctions est appelé un **package**.
- Par défaut, R charge en mémoire quelques packages de la bibliothèque seulement.
- Cela économise l'espace mémoire et accélère le démarrage.
- On charge de nouveaux packages en mémoire avec la fonction `library`.

Manipulation de vecteurs

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

<code>seq</code>	génération de suites de nombres
<code>rep</code>	répétition de valeurs ou de vecteurs
<code>sort</code>	tri en ordre croissant ou décroissant
<code>order</code>	positions dans un vecteur des valeurs en ordre croissant ou décroissant
<code>rank</code>	rang des éléments d'un vecteur en ordre croissant ou décroissant
<code>rev</code>	renverser un vecteur
<code>head</code>	extraction des n premières valeurs (R seulement)
<code>tail</code>	extraction des n dernières valeurs (R seulement)
<code>unique</code>	extraction des éléments différents d'un vecteur

Recherche d'éléments dans un vecteur

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

`which` positions des valeurs TRUE dans un vecteur booléen

`which.min` position du minimum dans un vecteur

`which.max` position du maximum dans un vecteur

`match` position de la première occurrence d'un élément dans un vecteur

`%in%` appartenance d'une ou plusieurs valeurs à un vecteur

Arrondi

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

<code>round</code>	arrondi à un nombre défini de décimales
<code>floor</code>	plus grand entier inférieur ou égal à l'argument
<code>ceiling</code>	plus petit entier supérieur ou égal à l'argument
<code>trunc</code>	troncature vers zéro de l'argument ; différent de <code>floor</code> pour les nombres négatifs

Sommaires et statistiques descriptives

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

<code>sum, prod</code>	somme et produit des éléments d'un vecteur
<code>diff</code>	différences entre les éléments d'un vecteur
<code>mean</code>	moyenne arithmétique et moyenne tronquée
<code>var, sd</code>	variance et écart type (versions sans biais)
<code>min, max</code>	minimum et maximum d'un vecteur
<code>range</code>	vecteur contenant le minimum et le maximum d'un vecteur
<code>median</code>	médiane empirique
<code>quantile</code>	quantiles empiriques
<code>summary</code>	statistiques descriptives d'un échantillon

Sommaires cumulatifs et comparaisons élément par élément

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

`cumsum, cumprod`

somme et produit cumulatif d'un vecteur

`cummin, cummax`

minimum et maximum cumulatif

`pmin, pmax`

minimum et maximum en parallèle,
c'est-à-dire élément par élément
entre deux vecteurs ou plus

Opérations sur les matrices

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

<code>t</code>	transposée
<code>solve</code>	avec un seul argument (une matrice carrée) : inverse d'une matrice ; avec deux arguments (une matrice carrée et un vecteur) : solution du système d'équation $Ax = b$
<code>diag</code>	avec une matrice en argument : diagonale de la matrice ; avec un vecteur en argument : matrice diagonale formée avec le vecteur ; avec un scalaire p en argument : matrice identité $p \times p$
<code>nrow, ncol</code>	nombre de lignes et de colonnes d'une matrice

Opérations sur les matrices (suite)

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

`rowSums, colSums`

sommes par ligne et par colonne,
respectivement, des éléments
d'une matrice ; voir aussi la
fonction `apply`

`rowMeans, colMeans`

moyennes par ligne et par
colonne, respectivement, des
éléments d'une matrice ; voir
aussi la fonction `apply`

`rowVars, colVars`

variance par ligne et par colonne
des éléments d'une matrice
(S-Plus seulement)

Produit extérieur

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

La fonction `outer`, dont la syntaxe est

$$\text{outer}(X, Y, \text{FUN}),$$

applique la fonction `FUN` (`prod` par défaut) entre chacun des éléments de `X` et chacun des éléments de `Y`.

- La dimension du résultat est par conséquent $c(\dim(X), \dim(Y))$.

- Par exemple : le produit extérieur entre deux vecteurs est une matrice contenant tous les produits entre les éléments des deux vecteurs :

```
> outer(c(1, 2, 5), c(2, 3, 6))
```

	[,1]	[,2]	[,3]
[1,]	2	3	6
[2,]	4	6	12
[3,]	10	15	30

- L'opérateur `%o%` est un raccourci de `outer(X, Y, prod)`.

- Par exemple : le produit extérieur entre deux vecteurs est une matrice contenant tous les produits entre les éléments des deux vecteurs :

```
> outer(c(1, 2, 5), c(2, 3, 6))
```

	[,1]	[,2]	[,3]
[1,]	2	3	6
[2,]	4	6	12
[3,]	10	15	30

- L'opérateur `%o%` est un raccourci de `outer(X, Y, prod)`.

Sommaire

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Opérations arithmétiques
- Opérateurs
- Appels de fonctions
 - Exemple
- Quelques fonctions utiles
 - Manipulation de vecteurs
 - Recherche d'éléments dans un vecteur
 - Arrondi
 - Sommaires et statistiques descriptives
 - Sommaires cumulatifs et comparaisons élément par élément
 - Opérations sur les matrices
 - Produit extérieur
- Structures de contrôle
 - Exécution conditionnelle
 - Boucles

Exécution conditionnelle

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

```
if (condition) branche.vrai else  
branche.faux
```

Si *condition* est vraie, *branche.vrai* est exécutée, et *branche.faux* sinon.

Si l'une ou l'autre de *branche.vrai* ou *branche.faux* comporte plus d'une expression, les grouper dans des accolades { }.

```
ifelse(condition, expression.vrai,  
expression.faux)
```

Fonction vectorisée qui remplace chaque élément TRUE du vecteur *condition* par l'élément correspondant de *expression.vrai* et chaque élément FALSE par l'élément correspondant de *expression.faux*.

Exécution conditionnelle

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

```
if (condition) branche.vrai else  
branche.faux
```

Si *condition* est vraie, *branche.vrai* est exécutée, et *branche.faux* sinon.

Si l'une ou l'autre de *branche.vrai* ou *branche.faux* comporte plus d'une expression, les grouper dans des accolades { }.

```
ifelse(condition, expression.vrai,  
expression.faux)
```

Fonction vectorisée qui remplace chaque élément TRUE du vecteur *condition* par l'élément correspondant de *expression.vrai* et chaque élément FALSE par l'élément correspondant de *expression.faux*.

Boucles

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Les boucles sont et **doivent** être utilisées avec parcimonie en S car elles sont généralement inefficaces (particulièrement avec S-Plus).
- Dans la majeure partie des cas, il est possible de vectoriser les calcul pour éviter les boucles explicites.
- Sinon, s'en remettre aux fonctions `apply`, `lapply` et `sapply` pour faire les boucles de manière plus efficace.

Boucles

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Les boucles sont et **doivent** être utilisées avec parcimonie en S car elles sont généralement inefficaces (particulièrement avec S-Plus).
- Dans la majeure partie des cas, il est possible de vectoriser les calcul pour éviter les boucles explicites.
- Sinon, s'en remettre aux fonctions `apply`, `lapply` et `sapply` pour faire les boucles de manière plus efficace.

Boucles

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

- Les boucles sont et **doivent** être utilisées avec parcimonie en S car elles sont généralement inefficaces (particulièrement avec S-Plus).
- Dans la majeure partie des cas, il est possible de vectoriser les calcul pour éviter les boucles explicites.
- Sinon, s'en remettre aux fonctions `apply`, `lapply` et `sapply` pour faire les boucles de manière plus efficace.

Boucles de longueur déterminée

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

```
for (variable in suite) expression
```

Exécuter *expression* successivement pour chaque valeur de *variable* contenue dans *suite*.

Encore ici, on groupera les expressions dans des accolades { }.

À noter que *suite* n'a pas à être composée de nombres consécutifs, ni même par ailleurs de nombres.

Boucles de longueur indéterminée

Introduction à
la programmation
en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

```
while (condition) expression
```

Exécuter *expression* tant que *condition* est vraie.

Si *condition* est fausse lors de l'entrée dans la boucle, celle-ci n'est pas exécutée.

Une boucle `while` n'est par conséquent pas nécessairement toujours exécutée.

```
repeat expression
```

Répéter *expression*. Cette dernière devra comporter un test d'arrêt qui utilisera la commande `break`.

Une boucle `repeat` est toujours exécutée au moins une fois.

Modification du déroulement d'une boucle

Introduction à
la programmation en
S

Vincent
Goulet

Opérations
arithmétiques

Opérateurs

Appels de
fonctions

Quelques
fonctions
utiles

Structures de
contrôle

`break`

Sortie immédiate d'une boucle `for`, `while` ou `repeat`.

`next`

Passage immédiat à la prochaine itération d'une boucle
`for`, `while` ou `repeat`.