Procédure R de WPS

Guide utilisateur et référence

Version: 4.4.2

www.worldprogramming.com

Table des matières

Procédure R 3
Introduction 4
Installation et configuration 4
Installer l'interpréteur R 5
Définir la variable d'environnement *R_HOME* 6
Utiliser R avec WPS 6
Convertir les types de données 7
Utilisation des graphiques R 8
Utiliser des packages R supplémentaires 9
Traitement des macros en langage SAS 10
Exemple 10
Référence de la procédure R 12
PROC R 13
ASSIGN 15
ENDSUBMIT 17
EXECUTE 17
EXPORT 18
IMPORT 19
LOAD 20
SAVE 21
SUBMIT 23

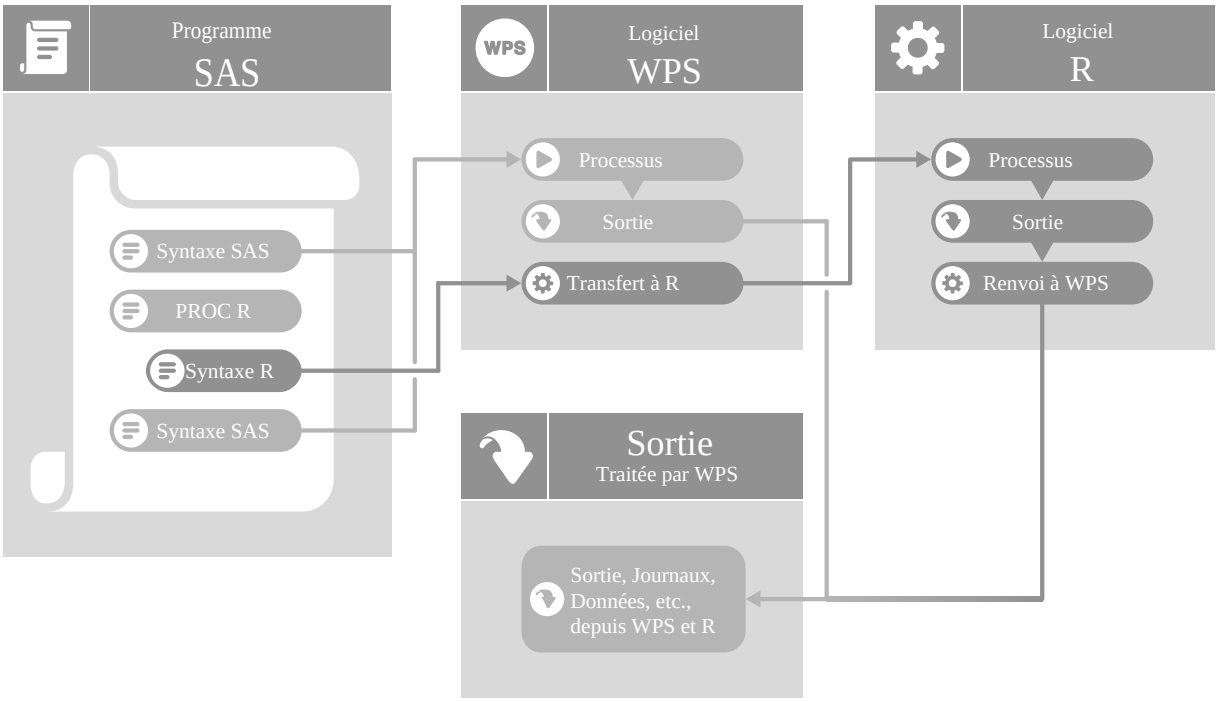
Notices légales 25

Procédure R

La procédure R permet d'inclure dans un programme WPS en langage SAS du code écrit en langage R.

La combinaison de R avec le langage SAS permet d'utiliser la puissance et les performances de SAS pour le gros du traitement et de l'analyse des données, tout en tirant parti des fonctionnalités spécifiques à R.

L'image suivante montre le traitement d'un programme en langage SAS utilisant la procédure R.



Introduction 4
La procédure R permet d'inclure dans un programme en langage SAS du code écrit en langage R.

Installation et configuration 4
Cette section décrit comment configurer l'environnement R pour WPS.

Utiliser R avec WPS 6
L'utilisation de R dans un programme en langage SAS vous permet de faire appel à des fonctionnalités qui ne sont pas disponibles dans WPS.

Exemple 10
Montre comment utiliser un ensemble de données en langage SAS dans la procédure R pour créer un diagramme en nuage de points.

Référence de la procédure R 12
Décrit la syntaxe et les options de `PROC R` et de ses instructions.

Introduction

La procédure R permet d'inclure dans un programme en langage SAS du code écrit en langage R.

En combinant les langages R et SAS, vous pouvez :

- Utiliser le langage SAS pour exécuter la préparation et le traitement de gros volumes de données et transmettre les données traitées à R.
- Utiliser les packages R que vous avez développés auparavant pour l'analyse des données.
- Utiliser des packages d'analyse des données R ou des solutions qui ne sont pas disponibles dans le langage SAS.

Nous conseillons de rédiger des programmes principalement écrits en SAS, et de basculer sur R lorsque des fonctions statistiques spécialisées sont requises.

Les données sont transmises du langage SAS à l'environnement de langage R à l'aide de l'instruction `EXPORT`. Une fois les données transférées, elles sont accessibles aux programmes en R sous la forme d'objets `data.frame`. Une fois le programme en R terminé, il est possible de renvoyer les données à l'environnement en langage SAS de WPS à l'aide de l'instruction `IMPORT`.

Installation et configuration

Cette section décrit comment configurer l'environnement R pour WPS.

Lorsque vous utilisez R avec WPS :

- Assurez-vous que vous utilisez le même type d'installation que pour WPS – vous devez utiliser la version Windows 32 bits ou 64 bits pour les deux logiciels.
- Il n'est pas nécessaire d'installer de nouveaux modules ou d'obtenir une licence spécifique pour utiliser la procédure R.

La procédure R peut être utilisée avec R version 2.15.x et versions ultérieures, et est actuellement prise en charge sur les systèmes Windows, Linux et macOS.

Le logiciel WPS n'est pas fourni avec R. Pour utiliser la procédure R, vous devez installer R séparément.

Après avoir installé R, définissez la variable d'environnement *R_HOME* pour qu'elle désigne le dossier contenant `libr.dll`, sur les plates-formes Windows, ou `libr.so`, pour les plates-formes Unix ou Linux.

Installer l'interpréteur R

Plates-formes Windows

Vous trouverez le programme d'installation à télécharger à l'adresse *https://www.r-project.org/*. Il inclut les versions 32 et 64 bits de R, et peut donc être utilisé avec la version du logiciel WPS installé sur votre système.

Par défaut, l'installation de R enregistre l'emplacement du logiciel dans le Registre Windows, où WPS peut alors trouver la version installée. Il s'agit de la dernière version installée, et vous n'avez à effectuer aucune opération spécifique dans WPS pour la trouver.

Plates-formes Unix ou Linux

WPS requiert la bibliothèque R partagée `libr.so` afin d'interagir avec l'interpréteur R. Cette bibliothèque n'est pas incluse par défaut dans la distribution R binaire pour les plates-formes UNIX. Sur les plates-formes Unix ou Linux, vous devez soit compiler R à partir du code source afin d'inclure la bibliothèque partagée requise, soit installer R à l'aide du système de gestion des packages de votre SE.

Pour compiler R à partir du code source :

1. Un ensemble minimal de bibliothèques doit être préinstallé avant de compiler R à partir du code source. Elles sont équivalentes au package *build-essentials* plus un JDK sur Ubuntu.
2. Téléchargez le code source de R à l'adresse *https://www.r-project.org/* et suivez les instructions fournies avec le téléchargement. Vérifiez que vous utilisez l'option --enable-R-shlib lorsque vous utiliser la commande configure, car c'est elle qui détermine la création de la bibliothèque partagée `libr.so`. Par exemple :

```
./configure --enable-R-shlib --prefix=$HOME/R
```

Pour plus d'informations, voir la documentation de R à l'adresse *https://cran.r-project.org/*.

Plates-formes macOS

Il est possible d'installer la distribution binaire de R depuis le site Web du projet R.

Pour utiliser R avec WPS, il est nécessaire de définir la variable *R_HOME* afin qu'elle pointe vers le répertoire d'installation contenant la bibliothèque partagée `libr.so`. Le paramètre par défaut est : `/Library/Frameworks/Framework/Resources`. En raison de la manière dont les applications sont lancées sur MacOS, il n'est pas possible de définir *R_HOME* dans un script shell de profil.

Pour utiliser une version spécifique de R, vous pouvez modifier le paramètre. Par exemple, `/Library/Frameworks/Framework/Versions/3.0/Resources`

Définir la variable d'environnement *R_HOME*

Afin que WPS trouve la version installée de R, il faut définir la variable d'environnement *R_HOME*.

Sur les plates-formes Windows, la variable d'environnement *R_HOME* doit pointer vers le dossier contenant le fichier `libr.dll`. Sur les plates-formes UNIX ou Linux, la variable d'environnement *R_HOME* doit pointer vers le dossier contenant le fichier `libr.so`.

Si vous exécutez WPS avec R sur une plate-forme Unix ou Linux, il est nécessaire de définir la variable *R_HOME* pour qu'elle pointe vers le dossier contenant `libr.so`.

Si vous avez plusieurs intallations de R, définissez la variable en tant que partie du programme en langage SAS exécuté dans WPS.

Définir *R_HOME* avant de lancer WPS

La variable *R_HOME* peut être définie comme une variable système, et utilisée par toutes les applications de votre dispositif qui exécutent R. Ceci n'est pas nécessaire sur la plate-forme Windows si l'installation par défaut de R est utilisée.

Définir *R_HOME* dans un pogramme en langage SAS

Il est possible de définir la variable *R_HOME* dans un programme en langage SAS à l'aide de l'option système `SET`. Par exemple :

```
OPTIONS SET = R_HOME 'C:\Program Files\R\R-3.5.0';
```

Ceci définit la variable d'environnement *R_HOME* pour la durée de l'exécution du programme en langage SAS.

Utiliser R avec WPS

L'utilisation de R dans un programme en langage SAS vous permet de faire appel à des fonctionnalités qui ne sont pas disponibles dans WPS.

Vous pouvez importer et utiliser les packages R installés à l'aide d'une instruction LIBRARY ; par exemple :

```
PROC R;
  SUBMIT;
    library(datasets)
    data(iris)
    summary(iris)
    plot(iris)
  ENDSUBMIT;
RUN;
```

Ensuite, chaque utilisation de la procédure R dans le programme en langage SAS peut utiliser le même environnement R. Toutes les variables globales ou packages importés utilisés dans une invocation de procédure R sont donc disponibles pour toutes les invocations ultérieures de cette procédure.

Chaque invocation de la procédure R peut inclure plusieurs blocs de code en R imbriqué, et peut utiliser une combinaison de code imbriqué et de programmes en R lancés par l'instruction `EXECUTE`.

Convertir les types de données

Décrit la correspondance entre les formats du langage SAS et les types de données R.

Lorsque vous écrivez des données dans un `data.frame` à l'aide de l'instruction `EXPORT` de la procédure R, les données formatées sont converties dans le type de données R équivalent. WPS comporte de nombreux formats qui affectent la sortie et l'affichage des données. Les formats qui n'affectent que la mise en page des sorties de données, tels que l'ajout de symboles de devise ou l'utilisation de virgules comme séparateurs décimaux, n'ont aucun effet lors de l'écriture de données.

Un `data.frame` est importé à l'aide de l'instruction `IMPORT` de la procédure R. Tout objet qu'il est possible de forcer dans un `data.frame` à l'aide de la fonction `as.data.frame` de R est importé dans l'ensemble de données WPS.

Valeurs logiques

Les valeurs logiques sont converties en valeurs numériques dans l'ensemble de données WPS. La valeur des vecteurs de type logique est convertie comme suit :

Valeur R	Valeur WPS
TRUE	1
FALSE	0

Valeur des entiers

La valeur `NA` du langage R, qui représente l'entier le plus petit (-2147483648), est convertie en valeur manquante dans le langage SAS.

Valeur des réels

Il existe trois valeurs numériques réelles spéciales en langage R :

- `NA` (« pas disponible »). Représente une valeur absente.
- `NaN` (« pas un nombre »). Représente une valeur indéfinie, ou qu'il est impossible d'afficher, telle que le résultat de la division de zéro par zéro.
- `Inf` (« infini »). Représente les valeurs de l'infini positif et négatif. Par exemple, le résultat de la division d'une valeur quelconque par zéro.

Ces valeurs sont converties de leur représentation en langage R en langage SAS comme suit :

Valeur R	Valeur WPS
NA	. (valeur manquante)
NaN	. (valeur manquante)
+Inf	.I
-Inf	.M

Valeurs de date

Lorsque vous importez dans WPS des variables R de type entier ou réel ayant la classe `Date`, le format `DATE9.` leur est appliqué.

Les valeurs `Date` du langage R représentent un nombre de jours à partir de l'époque UNIX, c'est-à-dire le 1er janvier 1970 UTC. Les variables importées sont ajustées pour prendre en compte l'époque du langage SAS, le 1er janvier 1960.

Valeurs date/heure

Lorsque vous importez dans WPS des variables R de type réel ayant la classe `POSIXct`, le format `DATETIME19.` leur est appliqué.

Les valeurs `POSIXct` du langage R représentent un nombre de secondes à partir de l'époque UNIX, c'est-à-dire le 1er janvier 1970 UTC. Les variables importées sont ajustées pour prendre en compte l'époque du langage SAS, le 1er janvier 1960, ainsi que la valeur spécifiée dans l'option `GMTOFFSET` de l'instruction `PROC R`.

Lorsque vous importez dans WPS des variables R de type réel ayant la classe `times`, le format `TIME8.` leur est appliqué.

Valeurs de caractères

Lors de l'importation, WPS évalue une variable de type caractère et lui applique un format correspondant à la longueur de la chaîne la plus longue. Les valeurs de variable de type caractère `NA` (« pas disponible ») sont converties en valeur de caractère manquant (`''`) de langage SAS.

Valeurs de facteur

Dans le langage R, un *facteur* est une forme de variable de type entier, où l'indice des valeurs est une liste de variables catégoriques (la liste est dénommée « niveau » en langage r). Une fois importés dans WPS, ils sont convertis en variables de type caractère dans l'ensemble de données. La longueur de la variable est égale à celle de la chaîne la plus longue dans la liste des niveaux.

Utilisation des graphiques R

Lors du lancement d'une session R, WPS configure R de sorte que les graphiques générés avec le dispositif graphique par défaut sont capturés et inclus dans la sortie ODS HTML standard de la session WPS.

Le programme suivant développe l'exemple précédent afin d'inclure une analyse de régression linéaire simple et les graphiques correspondants.

1. Créez un fichier de programme, et collez et enregistrez le code suivant :

```
data source;
  do x=1 to 10;
    y=ranuni(-1);
    output;
  end;

PROC R;
  export data=source;
  submit;
    model <- lm(source$y ~ source$x)
    print(model)
    par(mfrow=c(2, 2))
    plot(model)
endsubmit;
run;
```

2. Exécutez le programme en cliquant sur l'icône **Exécuter** de la barre d'outils et examinez la sortie HTML.

   La sortie inclut les résultats R imprimés ainsi qu'un graphique généré dans la session R et redirigé vers la sortie WPS.

```
Call:
lm(formula = source$y ~ source$x)
Coefficients:
(Intercept)     source$x
     0.5344       0.0241
```

Utiliser des packages R supplémentaires

Pour utiliser des packages supplémentaires qui ne sont pas inclus dans l'installation de R, nous vous conseillons d'utiliser l'environnement interactif de R pour effectuer l'installation et vérifier le fonctionnement de base des packages. Vous pouvez utiliser les packages installés dans un programme en R à l'aide de la fonction `library()`.

Un environnement R lancé par WPS hérite de la liste des variables d'environnement du processus WPS. Si des logiciels tiers installés et utilisés dans R nécessitent, par exemple, des entrées supplémentaires dans la variable d'environnement *PATH*, il est nécessaire de redémarrer WPS Workbench pour prendre ces modifications en compte.

Traitement des macros en langage SAS

Cette section explique comment utiliser les macros en langage SAS avec des programmes imbriqués en R.

Lors de l'exécution d'un programme en R imbriqué, le code compris entre les instructions `SUBMIT` et `ENDSUBMIT` est transmis tel quel à l'interpréteur R. Le traitement des macros est donc suspendu entre les instructions `SUBMIT` et `ENDSUBMIT` car :

- Le langage R utilise les caractères `&` et `%` dans sas syntaxe. Toute tentative de traitement de macro dans le code source R peut donc causer une interprétation incorrecte de syntaxe R valide en macro SAS.
- Le langage R permet l'utilisation de commentaires en fin de ligne. Ces commentaires peuvent contenir, par exemple, des apostrophes seules ; il ne serait alors pas possible de tokeniser la syntaxe R en utilisant les règles d'interprétation normales du langage SAS.

Exemple

Montre comment utiliser un ensemble de données en langage SAS dans la procédure R pour créer un diagramme en nuage de points.

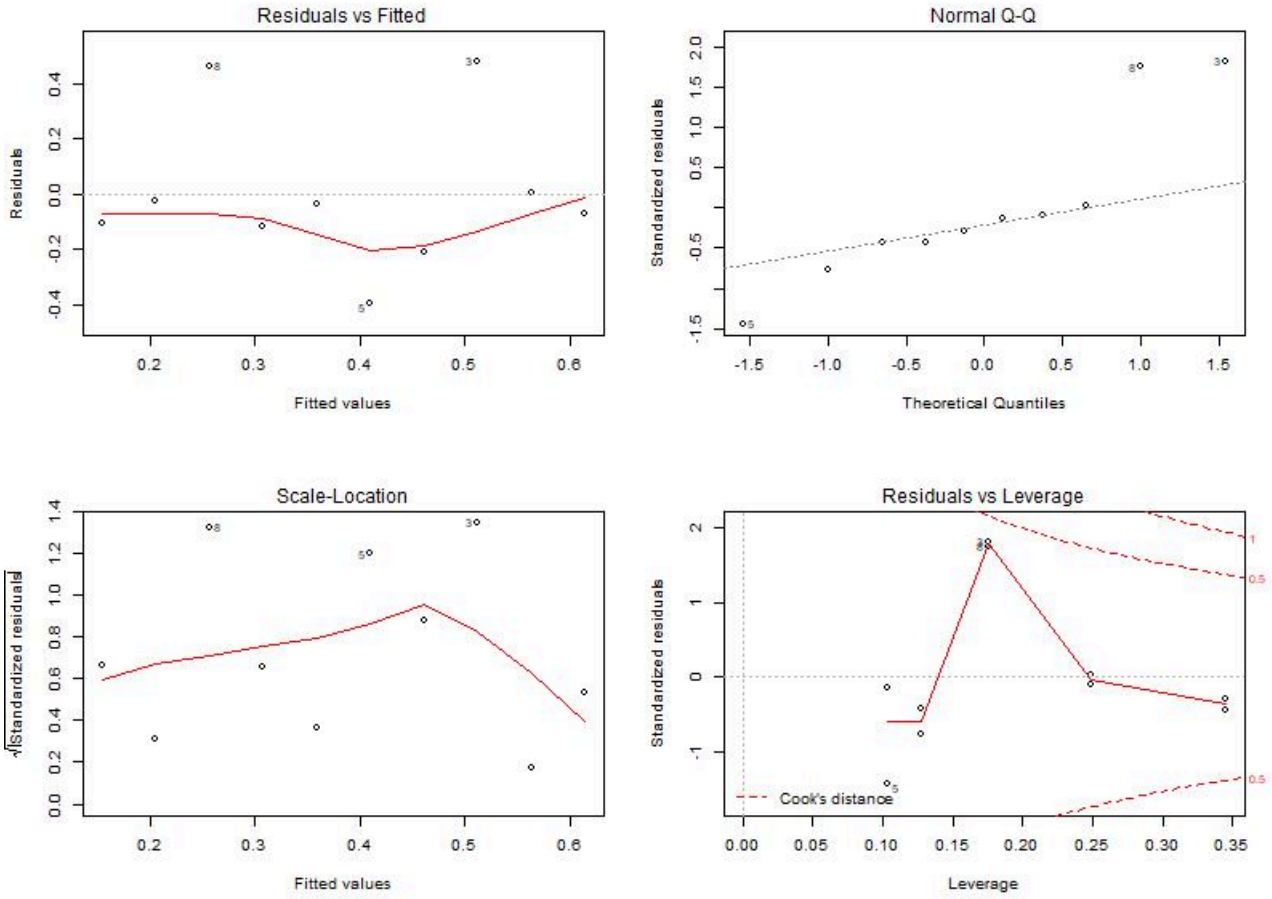
L'exemple suivant crée un ensemble de données dans une étape `DATA` du langage SAS, puis utilise l'instruction `EXPORT` pour passer cet ensemble de données à l'environnement R. L'ensemble de données est converti en un `data.frame` lors de l'exportation, et le data.frame est utilisé pour créer des tracés dans une grille mesurant deux fois la largeur et la hauteur du tracé.

Une destination de sortie fichier PDF est créée à l'aide du système de livraison de sortie (ODS) du langage SAS. Si vous ajoutez PDF aux destinations de sortie, le contenu du data.frame imprimé et le fichier d'image de tracé renvoyé sont inclus dans la sortie PDF. Le PDF est enregistré et vous pouvez consulter la sortie dans une visionneuse de PDF.

```
ODS PDF FILE='scatter_plot.pdf';
DATA SOURCE;
  DO X=1 TO 10;
    Y=RANUNI(-1);
    OUTPUT;
  END;
RUN;

PROC R;
  EXPORT DATA=source;
  SUBMIT;
    str(source)
    print(source)
    model <- lm(source$Y ~ source$X)
    print(model)
    par(mfrow=c(2, 2))
    plot(model)
    x <- (1:10)
  ENDSUBMIT;
  IMPORT R=x;
RUN;
ODS PDF CLOSE;
```

Ceci crée le diagramme de tracé suivant dans la sortie ODS PDF :



Référence de la procédure R

Décrit la syntaxe et les options de `PROC R` et de ses instructions.

PROC R 13
Invoque l'environnement R qui permet d'exécuter des programmes en R imbriqués ou externes.

ASSIGN 15
L'instruction `ASSIGN` permet d'affecter des valeurs de variable de langage SAS à un vecteur R.

ENDSUBMIT 17
Spécifie la fin d'un programme en langage R imbriqué.

EXECUTE 17
Exécute un programme en R stocké dans un fichier distinct.

EXPORT 18
Permet de convertir un ensemble de données pour langage SAS en `data.frame` R et d'y faire référence dans un programme en R.

IMPORT 19
Permet de convertir un `data.frame` pour le langage R en un ensemble de données pour langage SAS et d'y faire référence dans un programme en langage SAS.

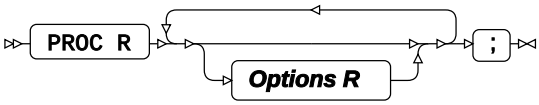
LOAD 20
L'instruction `LOAD` désérialise un *objet R* stocké dans un catalogue de langage SAS.

SAVE 21
Permet de sérialiser des objets R et de les stocker dans un catalogue SAS.

SUBMIT 23
Spécifie la fin d'un programme en langage R imbriqué.

PROC R

Invoque l'environnement R qui permet d'exécuter des programmes en R imbriqués ou externes.



Il est possible de mettre les ensembles de données créés dans WPS à la disposition du programme en R à l'aide de l'instruction `EXPORT`. Inversement, tous les ensembles de données exportés depuis le programme en R peuvent être chargés dans WPS à l'aide de l'instruction `IMPORT`.

Les programmes en R peuvent être inclus au sein des programmes WPS dans une procédure , ou bien figurer dans des fichiers distincts.

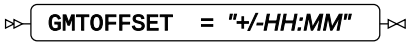
- Pour exécuter un programme en R imbriqué au sein d'un programme WPS, utilisez les instructions `SUBMIT` et `ENDSUBMIT`.
- Pour les programmes en R stockés dans des fichiers externes, utilisez l'instruction `EXECUTE`.

Pour quitter l'environnement R, il suffit d'utiliser une instruction `RUN`.

Options

Les options suivantes sont disponibles :

GMTOFFSET
Spécifie le décalage appliqué à l'heure UTC lors du déplacement d'un ensemble de données entre les langages SAS et R pour prendre en compte le fuseau horaire en cours.



Les valeurs de date et d'heure en langage R représentent le temps universel (UTC) associé à un fuseau horaire. En langage SAS, les valeurs de date et d'heure ne sont pas associés à un fuseau horaire. La valeur GMTOFFSET spécifiée est appliquée lors de l'utilisation des instructions ASSIGN, EXPORT ou IMPORT.

KEEP

Spécifie que l'environnement R en cours persiste une fois la procédure terminée.

Si cette option est spécifiée, l'environnement R en cours reste actif à la fin de la procédure en cours, et est utilisé lors de l'invocation suivante de la procédure R dans le même programme. Si cette invocation ne spécifie pas KEEP, l'environnement est fermé à la fin de la procédure.

Le comportement par défaut consiste à fermer l'environnement R à la fin de la procédure. Lorsque vous spécifiez KEEP, l'environnement R en cours est conservé, notamment les modules chargés lors de l'exécution d'un programme en R, et est utilisé à l'invocation suivante de PROC R.

Vous pouvez spécifier l'option système RKEEP afin d'utiliser le même environnement R tout au long de l'exécution du programme en langage SAS.

LIB

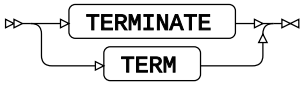
Spécifie l'emplacement de la bibliothèque par défaut pour l'étape de procédure. L'emplacement par défaut est la bibliothèque WORK.

L'emplacement LIB est utilisé :

- si *nom-bibliothèque* n'est pas spécifié dans le chemin de l'option DATA de l'instruction EXPORT lors de l'exportation d'un ensemble de données ;
- si *nom-bibliothèque* n'est pas spécifié dans le chemin de l'option DATA de l'instruction IMPORT lors de l'importation d'un ensemble de données ;
- si *nom-bibliothèque* n'est pas spécifié dans le chemin de l'option CATALOG de l'instruction SAVE lors de l'enregistrement d'un objet R dans un catalogue en langage SAS ;
- Si *nom-bibliothèque* n'est pas spécifié dans le chemin de l'option CATALOG de l'instruction LOAD lors du chargement d'un objet R dans un catalogue en langage SAS ;

TERMINATE

Spécifie qu'il faut fermer l'environnement R en cours une fois la procédure terminée.



Si vous spécifiez TERMINATE, l'environnement R en cours s'arrête, même si l'option système RKEEP a été spécifiée.

TIMESASCHRON

Spécifie si les valeurs d'heure sont représentées en R à l'aide de la classe `chron`.

Par défaut, les valeurs d'heure sont stockées dans le type `POSIXct` R, représentant un nombre de secondes depuis minuit. Lorsque vous spécifiez `TIMESASCHRON`, les valeurs d'heure sont stockées dans R en tant que types `chron.times`.

Pour utiliser l'option, vous devez inclure le package `chron` dans votre environnement R en utilisant l'instruction R `library()`.

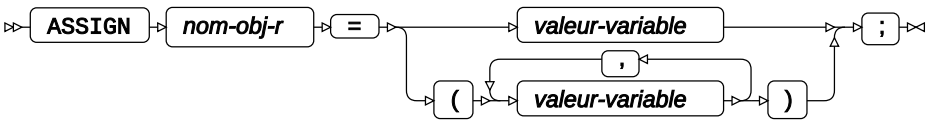
Exemple

L'exemple suivant montre comment utiliser `PROC R` pour rechercher la version de l'interpréteur R utilisé avec WPS. Les informations de version sont écrites dans la sortie ODS.

```
PROC R;
  SUBMIT;
    print(R.version)
  ENDSUBMIT;
RUN;
```

ASSIGN

L'instruction `ASSIGN` permet d'affecter des valeurs de variable de langage SAS à un vecteur R.



L'instruction `ASSIGN` permet de passer des paramètres à un programme en R, et il est possible d'utiliser le *nom-obj-r* spécifié dans un programme imbriqué ou un programme exécuté à l'aide de la commande `EXECUTE`. La casse est prise en compte pour la valeur *nom-obj-r* et, à la différence des variables du langage SAS, vous devez utiliser la même casse dans le programme en R que dans la définition `ASSIGN` pour y faire référence.

Il est possible de générer des variables en utilisant le développement ou l'exécution de variable de macro en langage SAS. Cela permet de prétraiter les variables à l'aide du langage SAS et de passer le résultat à un programme en R.

nom-obj-r

Spécifie une ou des valeurs de variable à passer à un programme en R. La valeur *nom-obj-r* spécifie le nom de la variable utilisé pour faire référence à *valeur-variable* dans le programme en R.

Il est possible de définir *nom-obj-r* en utilisant un nom littéral en langage SAS (`'nom-obj-r'N`) pour créer un objet R qui ne serait pas valide en langage SAS. Par exemple, pour affecter une valeur `Peter` à un objet R `employee.firstname`, l'instruction serait :

```
ASSIGN 'employee.firstname'N = 'Peter';
```

Il est possible d'affecter plusieurs *valeur-variable* à un même objet en langage R. Dans ce cas, toutes les valeurs doivent être du même type et se trouver dans une liste séparée par des virgules et entre parenthèses.

Passer plusieurs variables à R

Dans cet exemple, plusieurs variables de vecteur sont passées d'un programme en langage SAS à un programme en R qui convertit les vecteurs en data.frame.

```
PROC R;
  ASSIGN Nu = (1, 2, 3, 4, 5);
  ASSIGN Ch = ('Cyan', 'Magenta', 'Yellow', 'Black', 'Green');
  SUBMIT;
    DFrame <- data.frame(Nu, Ch)
    print (DFrame)
  ENDSUBMIT;
RUN;
```

Le résultat est le suivant :

```
  Nu       Ch
   1     Cyan
   2  Magenta
   3   Yellow
   4    Black
   5    Green
```

Affecter une variable de macro en langage SAS à un objet R

Dans cet exemple, la variable de macro en langage SAS *PARM* est passée à un programme en R pour déterminer la taille de l'échantillon dans un échantillon généré aléatoirement.

```
%LET PARM=15;
PROC R;
  ASSIGN parm=&PARM;
  SUBMIT;
    x<-sample(1:3, parm, replace=TRUE)
    print(x);
  ENDSUBMIT;
RUN;
```

Le résultat est le suivant :

```
[1] 1 1 3 3 3 2 1 1 3 1 1 3 3 1 1
```

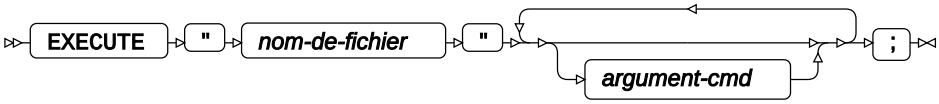
ENDSUBMIT

Spécifie la fin d'un programme en langage R imbriqué.

L'instruction `ENDSUBMIT` doit figurer en début de ligne après le programme en langage R.

EXECUTE

Exécute un programme en R stocké dans un fichier distinct.



L'instruction `EXECUTE` est l'alias de l'instruction `SUBMIT`. Elle permet de placer le code en R dans un fichier distinct afin de pouvoir l'exécuter depuis WPS comme depuis un environnement R interactif.

nom-de-fichier

Chaîne entre guillemets contenant le chemin d'accès au fichier de programme en R. *nom-de-fichier* peut être un nom de chemin absolu ou relatif.

Lorsque vous utilisez le Workbench pour exécuter un programme en langage R, si un chemin d'accès relatif est spécifié, la racine du chemin d'accès est l'espace de travail. Par exemple, pour exécuter un fichier nommé `math.r` depuis un projet nommé `calcul`, le chemin relatif est `/calcul/math.r`.

argument-cmd

Spécifie un argument de ligne de commande passé au programme en R.

Exemple d'exécution d'un programme en R stocké dans un fichier

Dans cet exemple, un programme en R stocké dans un fichier externe `model.r` est exécuté dans la procédure R. Contenu du fichier source `model.r` :

```
model <- lm(source$Y ~ source$X)
print(model)
par(mfrow=c(2, 2))
plot(model)
```

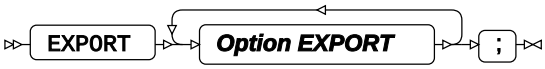
Le programme suivant crée un ensemble de données. L'ensemble de données est passé au programme en R à l'aide de l'instruction `EXPORT` avant l'exécution du programme en utilisant le fichier `model.r` :

```
DATA SOURCE;
  DO X=1 TO 10;
    Y=RANUNI(-1);
    OUTPUT;
  END;

PROC R;
  EXPORT DATA=source;
  EXECUTE "model.r";
RUN;
```

EXPORT

Permet de convertir un ensemble de données pour langage SAS en `data.frame` R et d'y faire référence dans un programme en R.

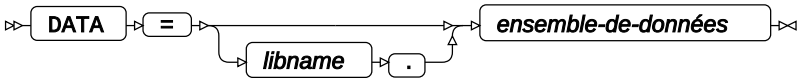


Options d'exportation

Les options suivantes sont disponibles avec l'instruction `EXPORT`

DATA

Spécifie le nom de l'ensemble de données WPS.



L'emplacement de la bibliothèque emplacement peut être spécifié soit en tant que *nom de bibliothèque* de l'option `DATA`, ou en tant qu'option `LIB` de l'instruction `PROC R` :

- Si vous spécifiez le *nom de bibliothèque*, cet emplacement est toujours utilisé.
- Si l'option `LIB` de l'instruction `PROC R` est spécifiée, et que le *nom de bibliothèque* n'est pas spécifié, l'emplacement de l'option `LIB` est utilisé.
- Si vous n'avez spécifié ni le *nom de bibliothèque*, ni l'option `LIB` de l'instruction `PROC R`, l'emplacement `WORK` est utilisé.

R

Spécifie le nom du `data.frame` pandas tel qu'il figure dans l'environnement R.

Si cette option n'est pas spécifiée, la valeur par défaut de *dataframe-r* est le nom de l'*ensemble de données* spécifié dans l'option `DATA`. Si vous utilisez la valeur *ensemble-de-données* par défaut dans un programme en R imbriqué, le nom de la variable doit utilisez la même casse que dans l'option `DATA`.

Exemple d'exportation des données de WPS vers R

L'exemple suivant crée un ensemble de données contenant deux colonnes numériques. L'ensembles de données est exporté vers l'environnement R, et le contenu du `data.frame` est imprimé dans la sortie ODS.

```
DATA SOURCE;
  DO X=1 TO 10;
    Y=RANUNI(-1);
    OUTPUT;
  END;

PROC R;
  EXPORT DATA=SOURCE;
  SUBMIT;
    str(source)
  ENDSUBMIT;
RUN;
```

Le résultat est le suivant :

```
'data.frame': 10 obs. of  2 variables:
 $ x: num  1 2 3 4 5 6 7 8 9 10
 $ y: num  0.371 0.924 0.59 0.434 0.962 ...
```

IMPORT

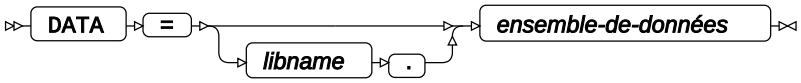
Permet de convertir un `data.frame` pour le langage R en un ensemble de données pour langage SAS et d'y faire référence dans un programme en langage SAS.

Options d'importation

Les options suivantes sont disponibles avec l'instruction `IMPORT`

DATA

Spécifie l'emplacement et le nom de l'ensemble de données dans l'environnement de langage SAS de WPS Analytics.



L'emplacement de la bibliothèque emplacement peut être spécifié soit en tant que *nom de bibliothèque* de l'option `DATA`, ou en tant qu'option `LIB` de l'instruction `PROC R` :

- Si vous spécifiez le *nom de bibliothèque*, cet emplacement est toujours utilisé.
- Si l'option `LIB` de l'instruction `PROC R` est spécifiée, et que le *nom de bibliothèque* n'est pas spécifié, l'emplacement de l'option `LIB` est utilisé.
- Si vous n'avez spécifié ni le *nom de bibliothèque*, ni l'option `LIB` de l'instruction `PROC R`, l'emplacement `WORK` est utilisé.

Si cette option n'est pas spécifiée, la valeur par défaut d'*ensemble-de-données* est le nom du *dataframe-r* spécifié dans l'option `R`.

R

Spécifie le nom du `data.frame` pandas tel qu'il figure dans l'environnement R. Doit être spécifié.

La casse de la variable *dataframe-r* est prise en compte, et la casse utilisée pour le `data.frame` dans le programme en R doit correspondre.

LOAD

L'instruction `LOAD` désérialise un *objet R* stocké dans un catalogue de langage SAS.

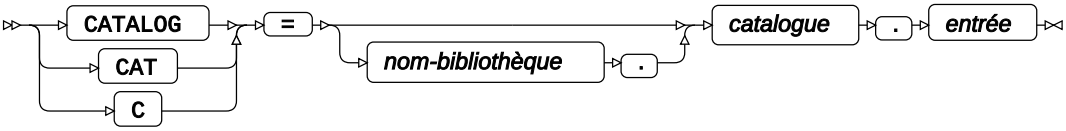
Les instructions `SAVE` et `LOAD` permettent de sérialiser un objet R et de le stocker dans un catalogue SAS, puis de le désérialiser dans une session WPS. L'instruction `SAVE` sérialise un objet R et le stocke dans une entrée de catalogue. L'instruction `LOAD` désérialise un objet R depuis un catalogue.

Options LOAD

Les options suivantes sont disponibles avec l'instruction `LOAD`

CATALOG

Spécifie depuis quel catalogue charger l'objet R stocké.



Un catalogue SAS est défini par :

nom-bibliothèque

Spécifie le nom de la bibliothèque où est stocké le catalogue. L'emplacement de la bibliothèque peut être spécifié soit en tant que *nom-bibliothèque*, soit en tant qu'option `LIB` de l'instruction `PROC R` :

- Si vous spécifiez le *nom de bibliothèque*, cet emplacement est toujours utilisé.
- Si l'option `LIB` de l'instruction PROC R est spécifiée, et que *nom-bibliothèque* n'est pas spécifié, l'emplacement de l'option `LIB` est utilisé.
- Si vous n'avez spécifié ni *nom-bibliothèque*, ni l'option `LIB` de l'instruction `PROC R`, l'emplacement `WORK` est utilisé.

catalogue

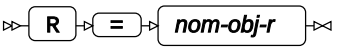
Spécifie le nom du catalogue.

entrée

Spécifie le nom de l'objet R dans le catalogue.

R

Spécifie le nom de la variable pour l'objet R chargé tel qu'il est utilisé dans le programme en langage R.



Comme le langage R prend la casse en compte, *nom-obj-r* doit utiliser la même casse que le nom de la variable R. Il est possible de spécifier le nom en utilisant une syntaxe de nom littéral (par exemple, `"r.object.name"N`) si le nom de l'objet r ne suit pas les règles normales des identifiants en langage SAS.

Exemple d'utilisation de l'instruction `LOAD`

```
proc r;
  load cat=catalog.entry r='target.object'n;
run;
```

SAVE

Permet de sérialiser des objets R et de les stocker dans un catalogue SAS.

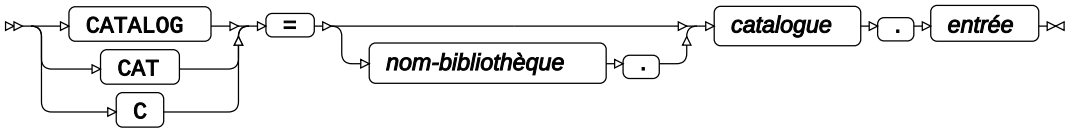
Les instructions `SAVE` et `LOAD` permettent de sérialiser un objet R et de le stocker dans un catalogue SAS, puis de le désérialiser dans une session WPS. L'instruction `SAVE` sérialise un objet R et le stocke dans une entrée de catalogue. L'instruction `LOAD` désérialise un objet R depuis un catalogue.

Options SAVE

Les options suivantes sont disponibles avec l'instruction `SAVE`.

CATALOG

Spécifie l'emplacement où enregistrer l'objet R.



L'emplacement de la bibliothèque peut être spécifié soit en tant que *nom-bibliothèque*, soit en tant qu'option `LIB` de l'instruction `PROC R` :

nom-bibliothèque

Spécifie le nom de la bibliothèque où est stocké le catalogue. L'emplacement de la bibliothèque peut être spécifié soit en tant que *nom-bibliothèque*, soit en tant qu'option `LIB` de l'instruction `PROC R` :

- Si vous spécifiez le *nom de bibliothèque*, cet emplacement est toujours utilisé.
- Si l'option `LIB` de l'instruction `PROC R` est spécifiée, et que le *nom de bibliothèque* n'est pas spécifié, l'emplacement de l'option `LIB` est utilisé.
- Si vous n'avez spécifié ni *nom-bibliothèque*, ni l'option `LIB` de l'instruction `PROC R`, l'emplacement `WORK` est utilisé.

catalogue

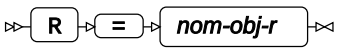
Spécifie le nom du catalogue.

entrée

Spécifie le nom de l'objet R dans le catalogue.

R

Spécifie le nom de l'objet R à enregistrer dans le catalogue. Doit être spécifié.



Comme le langage R prend la casse en compte, la casse de *nom-obj-r* doit correspondre à celle du nom de l'objet dans le programme en R. Il est possible de spécifier le nom en utilisant une syntaxe de nom littéral (par exemple, `"r.object.name"N`) si le nom de l'objet r ne suit pas les règles normales des identifiants en langage SAS.

DESCRIPTION

Sp écifie une chaîne de caractères de description enregistrée avec l'entrée de catalogue.

La description est affichée dans la sortie de l'instruction `PROC CATALOG`.

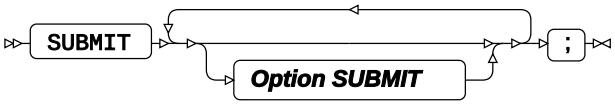
L'entrée de catalogue est de type `ROBJECT`.

Exemple d'enregistrement d'un objet R dans un catalogue WPS

```
proc r;
  save cat=catalog.entry r='source.object'n;
run;
```

SUBMIT

Spécifie la fin d'un programme en langage R imbriqué.



Le terme « programme imbriqué » désigne une partie de la procédure `R` dans un programme en langage SAS. L'instruction `SUBMIT` signale le début du programme, et l'instruction `ENDSUBMIT` signale sa fin.

Le programme en langage R doit commencer sur une nouvelle ligne après l'instruction `SUBMIT`, et l'instruction `ENDSUBMIT` doit figurer seule sur sa ligne.

Un même environnement de procédure R peut contenir plusieurs programmes en R imbriqués. Chaque programme en R est exécuté lorsqu'il se présente. Les variables définies dans un programme imbriqué peuvent être réutilisées dans les programmes imbriqués suivants du même environnement de procédure R.

Options SUBMIT

Il est possible d'utiliser l'option suivante avec l'instruction `SUBMIT`.

symbole-r

Permet de remplacer un symbole dans le programme en R par une chaîne de caractères.

Avant d'être transmis à l'environnement R, WPS pré-traite les programmes en R imbriqués afin de remplacer tout*symbole-r* défini dans le programme par le contenu de *valeur-remplacement*. Si *valeur-remplacement* est une chaîne de caractères, elle doit être placée entre guillemets. Si *valeur-remplacement* est une variables de macro SAS, elle doit être précédée d'une esperluète (&), mais les guillemets ne sont pas nécessaires.

La casse est prise en compte pour la valeur *symbole-r* et vous devez utiliser la même casse dans le programme en R que dans l'instruction `SUBMIT`.

```
PROC R;
SUBMIT greeting = 'Hello World';
r.welcome <- "&greeting."
print (r.welcome)
ENDSUBMIT;
RUN;
```

La syntaxe de remplacement d'un *symbole-r* défini est identique à celle utilisée pour le remplacement de variable de macro en langage SAS. Le nom du *symbole-r* défini est précédé d'une esperluète (&) dans le code imbriqué.

Instruction SUBMIT – Exemple simple

L'exemple suivant montre comment incorporer un programme en langage R dans un programme en langage SAS :

```
PROC R;
SUBMIT;
x <- (1:10)
print(x)
ENDSUBMIT;
RUN;
```

La sortie est la suivante :

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

Instruction SUBMIT – Utiliser une variable de macro dans un programme en R

L'exemple suivant montre comment transmettre une variable d'un programme en langage SAS à un programme en langage R. La variable *welcome* est définie à l'aide de la macro en langage SAS `%LET`. La variable de macro spécifiée est affectée à la variable *symbole-r* de l'instruction `SUBMIT`. Le programme imbriqué en langage R fait alors référence à *symbole-r* :

```
%LET welcome = 'Hello World';
PROC R;
SUBMIT greeting = &welcome;
r.welcome <- "&greeting"
print (toupper(r.welcome))
ENDSUBMIT;
RUN;
```

La sortie est la suivante :

```
[1] "HELLO WORLD"
```

Notices légales

Marques

WPS et World Programming sont des marques commerciales ou des marques déposées de World Programming Limited dans l'Union européenne et dans d'autres pays. Le sigle (r) ou ® indique l'enregistrement au niveau de l'Union européenne (« marque communautaire »).

SAS et tous les autres noms de produits et de services de SAS Institute Inc. sont des marques déposées ou des marques commerciales de SAS Institute Inc. aux États-Unis et dans d'autres pays. ® indique que la marque est déposée aux États-Unis.

Toutes les autres marques commerciales sont la propriété de leurs détenteurs respectifs.

Notices générales

World Programming Limited n'est associé d'aucune manière à SAS Institute Inc.

WPS n'est pas le système SAS.

Les expressions « SAS » et « langage SAS » utilisées dans ce document font référence au langage de programmation SAS qui est souvent désigné par ces termes.

Les expressions « programme », « programme SAS » et « programme en langage SAS » utilisées dans ce document font référence aux programmes écrits en langage SAS. Ils peuvent également être appelés « scripts », « scripts SAS » ou « scripts en langage SAS ».

Les expressions « IML » et « langage IML », « syntaxe IML » et « Interactive Matrix Language » utilisées dans ce document font référence au langage de programmation informatique qui est souvent désigné par ces termes.

WPS inclut du logiciel développé par des tiers. Vous trouverez plus d'informations dans le fichier THANKS ou acknowledgements-fr.txt inclus dans l'installation de WPS.