



# *Procedura R di WPS*

## *guida dell'utente e riferimento*

Versione: 4.4.3

(c) 2022 World Programming, an Altair Company

[www.worldprogramming.com](http://www.worldprogramming.com)

# Indice

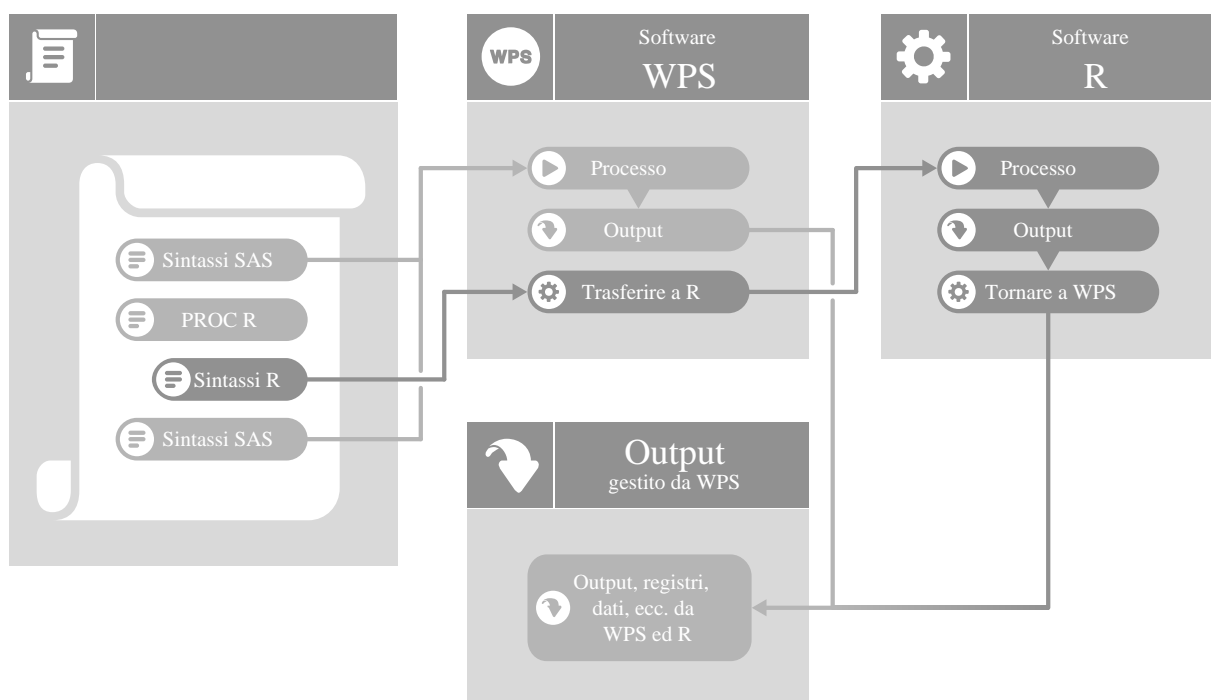
<b>Procedura R.....</b>	<b>3</b>
Introduzione.....	4
Installazione e configurazione.....	4
Installazione dell'interprete di R.....	5
Impostazione della variabile di ambiente <i>R_HOME</i> .....	6
Uso di R con WPS.....	6
Conversione del tipo di dati.....	7
Uso della grafica di R.....	8
Uso di ulteriori pacchetti di R.....	9
Elaborazione di macro nel linguaggio SAS.....	9
Esempio.....	10
Riferimento alla procedura R.....	11
PROC R.....	12
ASSIGN.....	14
ENDSUBMIT.....	16
EXECUTE.....	16
EXPORT.....	17
IMPORT.....	18
LOAD.....	19
SAVE.....	20
SUBMIT.....	22
<b>Avvisi legali.....</b>	<b>24</b>

# Procedura R

La procedura R consente ai programmi di WPS scritti in linguaggio SAS di includere il codice scritto in linguaggio R.

La combinazione del linguaggio R e del linguaggio SAS consente di scrivere la maggior parte di una soluzione di elaborazione e analisi dei dati nel linguaggio SAS di livello industriale e ad alte prestazioni, sfruttando al contempo le funzionalità presenti nel linguaggio R.

L'immagine seguente dimostra come viene elaborato un programma in linguaggio SAS che utilizza la procedura R.



Introduzione <a href="#">↗</a> .....	4
La procedura R consente ai programmi in linguaggio SAS di includere il codice scritto in linguaggio R.	
Installazione e configurazione <a href="#">↗</a> .....	4
Impostazione dell'ambiente di R per WPS.	
Uso di R con WPS <a href="#">↗</a> .....	6
L'uso di R in un programma in linguaggio SAS permette di utilizzare funzioni che potrebbero non essere disponibili in WPS.	

Esempio <a href="#">↗</a> .....	10
Dimostra come utilizzare un set di dati in linguaggio SAS nella procedura R per creare un diagramma del grafico a dispersione.	
Riferimento alla procedura R <a href="#">↗</a> .....	11
Descrive la sintassi e le opzioni per PROC R e le sue istruzioni contenute.	

## Introduzione

La procedura R consente ai programmi in linguaggio SAS di includere il codice scritto in linguaggio R.

Combinando il linguaggio R e SAS, è possibile:

- Utilizzare il linguaggio SAS per eseguire la procedura in blocco e la preparazione dei dati, e trasferire i dati elaborati a R.
- Utilizzare i pacchetti R sviluppati in precedenza per l'analisi dei dati.
- Utilizzare pacchetti o soluzioni di analisi dei dati R che non potrebbero essere disponibili nel linguaggio SAS.

Si consiglia di codificare i programmi principalmente in linguaggio di SAS, utilizzando R laddove sono necessarie statistiche specialistiche.

I dati vengono trasferiti tra gli ambienti del linguaggio SAS e del linguaggio R mediante l'istruzione `EXPORT`. Una volta trasferiti i dati, vengono messi a disposizione come oggetto `frame.dati` a un programma R. Al completamento del programma R, se necessario, è possibile riportare i dati all'ambiente del linguaggio SAS tramite l'istruzione `IMPORT`.

## Installazione e configurazione

Impostazione dell'ambiente di R per WPS.

Quando si usa R con WPS:

- L'interprete di R a 32 bit è necessario per la versione di WPS a 32 bit e l'interprete di R a 64 bit è necessario per la versione di WPS a 64 bit.
- Non è necessario installare altri moduli e non vi sono requisiti di licenza specifici per usare la procedura R.

La procedura R si può utilizzare con la versione 2.15.x di R e versioni successive, ed è attualmente supportata su sistemi Microsoft Windows, Linux e macOS.

Il software WPS non viene fornito con una copia di R. Per utilizzare la procedura R, è necessaria un'installazione separata di R.

Dopo l'installazione di R, impostare la variabile di ambiente `R_HOME` in modo che punti alla cartella contenente `libr.dll` su piattaforme Windows o `libr.so` su piattaforme UNIX o Linux.

## Installazione dell'interprete di R

### Piattaforme Windows

È possibile scaricare il pacchetto del programma di installazione per Windows in <https://www.r-project.org/> e include versioni di R a 32 e a 64 bit, quindi lo stesso programma di installazione funziona per le versioni del software WPS a 32 e 64 bit.

Per impostazione predefinita, l'installazione di R salva il percorso di installazione nel registro di sistema di Windows, che il luogo in cui WPS cerca di trovare la versione attualmente installata. Questa è la versione di R installata più recentemente e non è necessaria nessuna configurazione specifica di WPS per trovarla.

### Piattaforme UNIX o Linux

WPS richiede la raccolta R condivisa `libr.so` per interagire con l'interprete di R. Questa raccolta non è inclusa per impostazione predefinita nella distribuzione binaria di R per le piattaforme UNIX. Su piattaforme UNIX o Linux, è necessario compilare R dall'origine in modo che includa la raccolta condivisa richiesta oppure installare R tramite il sistema di gestione dei pacchetti di sistemi.

Per compilare dal codice sorgente:

1. È necessario un minimo insieme di raccolte preinstallate prima di poter compilare R dal codice sorgente. Questi sono equivalenti al pacchetto `build-essentials` più un JDK su Ubuntu.
2. Scaricare il codice sorgente R da <https://www.r-project.org/> e seguite le istruzioni incluse nel download. Assicurarsi di utilizzare l'opzione `--enable-R-shlib` quando si esegue la configurazione per compilare la raccolta condivisa `libr.so`, ad esempio:

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

Per maggiori informazioni, consultare la documentazione di R all'indirizzo <https://cran.r-project.org/>.

### Piattaforme macOS

È possibile installare direttamente la distribuzione binaria di R dal sito Web del progetto R.

Per usare R con WPS, è necessario impostare la variabile `R_HOME` in modo che punti alla directory di installazione contenente la raccolta condivisa `libr.so`. L'impostazione predefinita è: `/Library/Frameworks/Framework/Resources`. A causa del modo in cui le applicazioni vengono avviate in macOS, non è possibile impostare `R_HOME` in uno script del profilo shell.

Per utilizzare una versione specifica di R, è possibile modificare adeguatamente l'impostazione, ad esempio: `/Library/Frameworks/Framework/Versions/3.0/Resources`

## Impostazione della variabile di ambiente *R\_HOME*

Per trovare la versione installata di R, è necessario impostare la variabile di ambiente *R\_HOME*.

Nelle piattaforme Windows, la variabile di ambiente *R\_HOME* deve puntare alla cartella contenente il file `libr.dll`. Nelle piattaforme UNIX o Linux, la variabile di ambiente *R\_HOME* deve puntare alla cartella contenente il file `libr.so`.

Se si esegue WPS con R su una piattaforma Unix o Linux, impostare la variabile *R\_HOME* in modo che punti alla cartella contenente `libr.so`.

Se si dispone di più installazioni di R, impostare la variabile *R\_HOME* come parte del programma in linguaggio SAS che si esegue in WPS.

### Impostare *R\_HOME* prima di eseguire WPS

La variabile *R\_HOME* può essere definita come variabile di sistema, e usata da tutte le applicazioni sul dispositivo che esegue R. Non è necessario impostare la variabile sulla piattaforma Windows se viene usata la posizione di installazione predefinita di R.

### Impostare *R\_HOME* in un programma in linguaggio SAS

È possibile impostare la variabile *R\_HOME* in un programma in linguaggio SAS mediante l'opzione di sistema `SET`, ad esempio:

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

Imposterà la variabile di ambiente *R\_HOME* per la durata dell'esecuzione del programma in linguaggio SAS.

## Uso di R con WPS

L'uso di R in un programma in linguaggio SAS permette di utilizzare funzioni che potrebbero non essere disponibili in WPS.

È possibile importare e usare i pacchetti R installati all'interno del codice R nella riga usando l'istruzione `library`, per esempio:

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

Ogni uso successivo della procedura R in un programma in linguaggio SAS può utilizzare lo stesso ambiente di R. Ciò significa che le variabili globali o i pacchetti importati utilizzati in una invocazione della procedura R sono disponibili a tutte le invocazioni successive della procedura R.

Ogni invocazione della procedura R può includere più blocchi di codice in linguaggio R nella riga e utilizzare una combinazione di codice in linguaggio R nella riga e programmi R eseguiti mediante l'istruzione `EXECUTE`.

## Conversione del tipo di dati

Descrive la corrispondenza tra i formati del linguaggio SAS e i tipi di dati di R.

Quando si scrivono dati su un `frame.dati` con l'istruzione `EXPORT` della procedura R, i dati formattati vengono convertiti in un tipo di dati di R equivalente. WPS ha molti formati che influenzano l'output e la visualizzazione dei dati. I formati che influenzano solo il layout dell'output dei dati, come l'aggiunta di simboli di valuta o separatori di virgola, non hanno alcun effetto durante la scrittura dei dati.

Un `frame.dati` viene importato tramite l'istruzione `IMPORT` della procedura R. Qualsiasi oggetto che può essere forzato in un `frame.dati` usando la funzione di R `as.data.frame` viene importata nel set di dati di WPS.

### Valori logici

I valori logici sono convertiti in valori numerici nel set di dati di WPS. I valori dei vettori di tipo logico vengono convertiti come riportato di seguito:

Valore di R	Valore di WPS
TRUE	1
FALSE	0

### Valori dei numeri interi

Il valore di `NA` del linguaggio R, rappresentato come il minimo valore del numero intero (-2147483648) viene convertito in un valore mancante del linguaggio SAS.

### Valori dei numeri reali

Esistono tre specifici valori numerici reali nel linguaggio R:

- `NA` (Non disponibile). Rappresenta un valore assente.
- `NaN` (Non un numero). Rappresenta un valore indefinito, o un valore che non può essere visualizzato, per esempio il risultato di zero diviso per zero.
- `Inf` (Infinito). Rappresenta valori infiniti positivi e negativi. Per esempio il risultato del tentativo di dividere qualsiasi valore per zero.

Questi valori vengono convertiti dalla rappresentazione in linguaggio R alla rappresentazione in linguaggio SAS come segue:

Valore di R	Valore di WPS
NA	. (valore mancante)
NaN	. (valore mancante)
+Inf	.I
-Inf	.M

### Valori di data

Le variabili intere o reali nel linguaggio R che hanno una classe `Date` vengono formattate come `DATE9` . quando vengono importate in WPS.

I valori di `Date` del linguaggio R rappresentano un conteggio di giorni dal periodo Unix del 1° gennaio 1970 UTC. Le variabili importate vengono regolate in modo da tener conto del periodo del linguaggio SAS del 1° gennaio 1960.

### Valori di data/ora

Le variabili reali nel linguaggio R che hanno una classe `POSIXct` vengono formattate come `DATETIME19` . quando vengono importate in WPS.

I valori di `POSIXct` del linguaggio R rappresentano un conteggio di secondo dal periodo Unix del 1° gennaio 1970 UTC. Le variabili importate vengono regolate in modo da tener conto del periodo del linguaggio SAS del 1° gennaio 1960 e usano inoltre il valore specificato nell'opzione `GMTOFFSET` dell'istruzione `PROC R`.

Le variabili reali nel linguaggio R che hanno una classe `times` vengono formattate come `TIME8` . quando vengono importate in WPS.

### Valori del carattere

All'importazione, WPS analizza una variabile di carattere e assegna un formato che è la lunghezza della stringa più lunga. I singoli valori in una variabile di carattere che sono `NA` (Non disponibile) vengono convertiti nel valore di carattere mancante del linguaggio SAS ( ' ' ).

### Valori dei fattori

Un *fattore* del linguaggio R è una forma di variabile intera, in cui i valori indicizzano un elenco di variabili categoriche (l'elenco è conosciuto come livello nel linguaggio R). Se importati in WPS, questi vengono convertiti in variabili di caratteri nel set di dati. Alla variabile viene assegnata una lunghezza uguale alla stringa più lunga nell'elenco dei livelli.

## Uso della grafica di R

All'avvio di una sessione di R, WPS configura R in modo tale che i grafici generati con il dispositivo di grafica predefinito vengono acquisiti e inclusi nell'output ODS HTML standard della sessione di WPS.

Il seguente programma amplia l'esempio precedente fino ad includere l'analisi e la grafica per le regressioni lineari semplici.



1. Creare un nuovo file di programma, quindi incollare il seguente codice e salvare il file:

```
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. Eseguire il programma facendo clic sull'icona della barra degli strumenti **Esegui** ed esaminare l'output HTML.

L'output include i risultati di R stampati unitamente ad un grafico semplice generato dall'interno della sessione di R e indirizzato all'output di WPS.

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

## Uso di ulteriori pacchetti di R

Per usare pacchetti aggiuntivi che non sono inclusi nell'installazione di R, si consiglia di installare e controllare il funzionamento di base di questi pacchetti nell'ambiente interattivo di R. È possibile utilizzare i pacchetti installati in un programma R tramite la funzione `library()`.

Un ambiente di R avviato da WPS eredita le variabili di ambiente dal processo di WPS. Se viene installato software di terzi per l'uso in R che richiede, per esempio, voci aggiuntive nella variabile d'ambiente `PATH`, è necessario riavviare Workbench per registrare le modifiche.

## Elaborazione di macro nel linguaggio SAS

Uso di macro del linguaggio SAS con programmi R nella riga.

Quando viene eseguito un programma R nella riga, il codice tra le istruzioni `SUBMIT` e `ENDSUBMIT` viene trasferito testualmente all'interprete di R. L'elaborazione di macro viene pertanto sospesa tra le istruzioni `SUBMIT` ed `ENDSUBMIT` perché:

- Il linguaggio R utilizza i caratteri & e % come parte della sintassi. Il tentativo di elaborare le macro del codice sorgente di R potrebbe produrre una sintassi di R valida, interpretata erroneamente come istruzioni di macro del linguaggio SAS.
- Il linguaggio R consente commenti in stile fine riga. Il contenuto potrebbe contenere, ad esempio, apostrofi senza corrispondenza; la tokenizzazione della sintassi di R che usa le normali regole di analisi del linguaggio SAS non poteva quindi verificarsi.

## Esempio

Dimostra come utilizzare un set di dati in linguaggio SAS nella procedura R per creare un diagramma del grafico a dispersione.

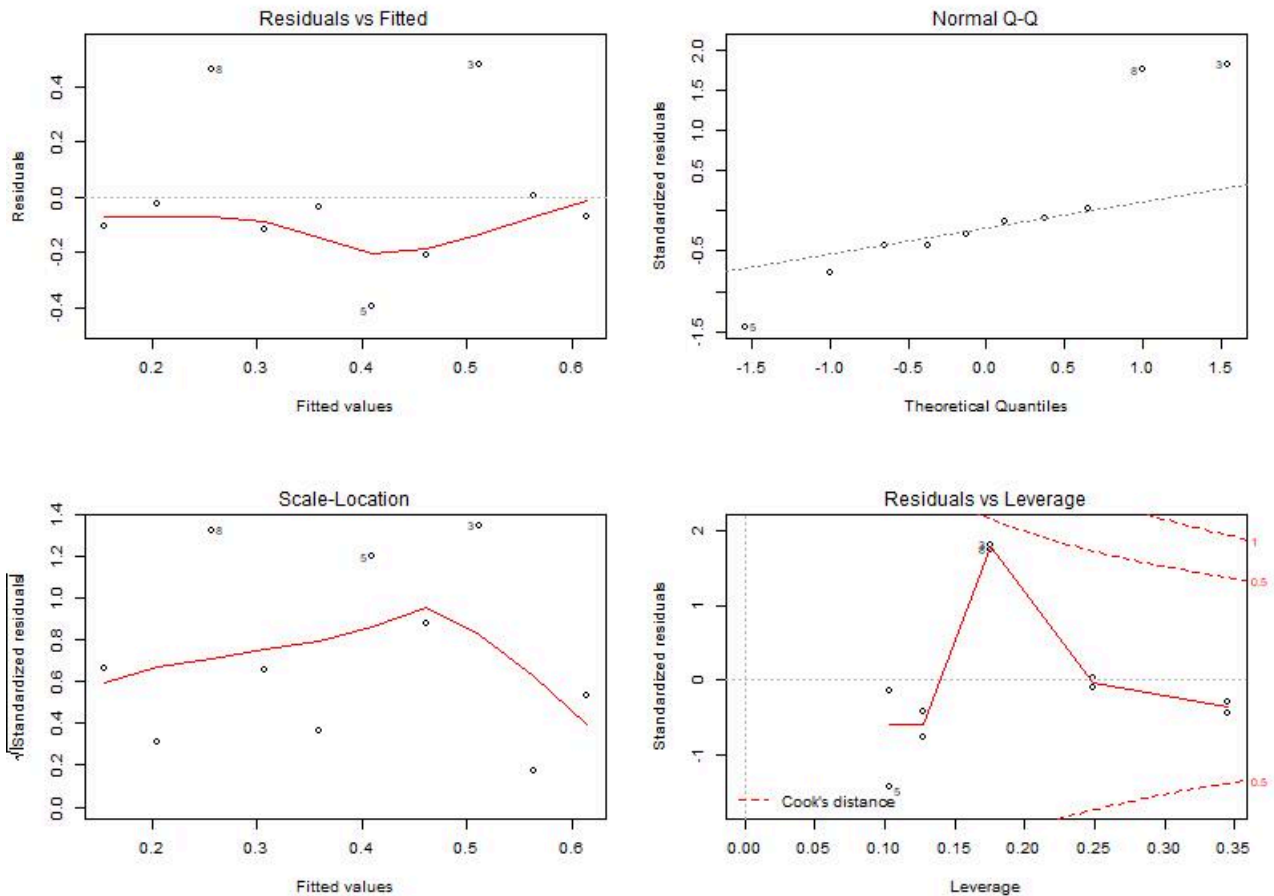
L'esempio seguente crea un set di dati in un `DATA` step in linguaggio SAS, quindi usa l'istruzione `EXPORT` per trasferire il set di dati all'ambiente di R. Il set di dati viene convertito in un `frame.dati` durante l'esportazione e il `frame.dati` viene utilizzato per creare tracciati in una griglia larga due tracciati e profonda due tracciati.

Una destinazione del file PDF di output viene creata con il sistema di consegna dell'output SAS (ODS). L'aggiunta del PDF alle destinazioni di output include il contenuto del `frame.dati` stampato e il file di immagine del tracciato restituito nell'output del PDF. Il PDF viene salvato ed è possibile visualizzare l'output in un visualizzatore di 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;
```

Questo crea il seguente tracciato nell'output del PDF di ODS:



## Riferimento alla procedura R

Descrive la sintassi e le opzioni per PROC R e le sue istruzioni contenute.

PROC R <a href="#">↗</a> .....	12
Richiama l'ambiente di R che consente l'esecuzione di programmi in linguaggio R nella riga o esterni.	
ASSIGN <a href="#">↗</a> .....	14
È possibile utilizzare l'istruzione ASSIGN per assegnare i valori delle variabili del linguaggio SAS a un vettore R.	
ENDSUBMIT <a href="#">↗</a> .....	16
Specifica la fine di un programma in linguaggio R nella riga.	
EXECUTE <a href="#">↗</a> .....	16
Esegue un programma R archiviato in un file separato.	

**EXPORT** ..... 17  
Consente di convertire un set di dati in linguaggio SAS in un `frame.dati` di R e di farne riferimento in un programma R.

**IMPORT** ..... 18  
Consente la conversione di un `frame.dati` del linguaggio R in un set di dati del linguaggio SAS e il riferimento in un programma in linguaggio SAS.

**LOAD** ..... 19  
L'istruzione `LOAD` deserializza un *oggetto di R* archiviato in un catalogo del linguaggio SAS.

**SAVE** ..... 20  
Consente agli oggetti di R di essere serializzati e archiviati in un catalogo del linguaggio SAS.

**SUBMIT** ..... 22  
Specifica l'inizio di un programma in linguaggio R nella riga.

## PROC R

Richiama l'ambiente di R che consente l'esecuzione di programmi in linguaggio R nella riga o esterni.



I set di dati creati in WPS si possono mettere a disposizione del programma R tramite l'istruzione `EXPORT` e un set di dati importato dal programma R in WPS tramite l'istruzione `IMPORT`.

È possibile scrivere un programma R nella riga, nella procedura R, oppure eseguirlo da un file separato:

- Per eseguire un programma R nella riga, utilizzare le istruzioni `SUBMIT` ed `ENDSUBMIT`.
- Per eseguire un programma R archiviato in un file esterno, utilizzare l'istruzione `EXECUTE`.

Si può uscire dall'ambiente di R con un'istruzione `RUN`.

## Opzioni

Le seguenti opzioni sono disponibili.

### GMTOFFSET

Specifica l'offset su UTC applicato quando si passano set di dati dal linguaggio SAS agli ambienti del linguaggio R per tener conto del fuso orario corrente.

```
GMTOFFSET = "+/-HH:MM"
```

I valori di data e ora nel linguaggio R sono rappresentati in UTC (Universal Time Coordinated) con un fuso orario associato. Nel linguaggio SAS, i valori di data e ora non hanno un fuso orario implicito. Il GMTOFFSET specificato viene applicato quando si usano le istruzioni ASSIGN, EXPORT o IMPORT.

## KEEP

Specifica che l'ambiente di R attuale non viene terminato quando si chiude la procedura.

» **KEEP** «

Se specificato, l'attuale ambiente di R resta attivo quando la procedura attuale viene chiusa e l'ambiente viene utilizzato nella successiva invocazione della procedura di R dello stesso programma. Se quell'invocazione non specifica **KEEP**, l'ambiente viene terminato alla chiusura della procedura.

Il comportamento predefinito è quello di terminare l'ambiente di R al termine della procedura. Specificando **KEEP**, si mantiene l'attuale ambiente di R, inclusi i moduli caricati durante l'esecuzione di un programma R, da utilizzare alla prossima invocazione di PROC R.

È possibile specificare l'opzione di sistema **RKEEP** per utilizzare lo stesso ambiente di R per la durata dell'esecuzione del programma in linguaggio SAS.

## LIB

Specifica il percorso della raccolta predefinito per il passo della procedura. Il percorso predefinito è la biblioteca **WORK**.

» **LIB = *raccolta-predefinita*** «

Il percorso **LIB** viene utilizzato:

- Quando si esporta un set di dati e il *nome-raccolta* non è specificato come parte del percorso per l'opzione **DATA** dell'istruzione **EXPORT**.
- Quando si importa un set di dati e il *nome-raccolta* non è specificato come parte del percorso per l'opzione **DATA** dell'istruzione **IMPORT**.
- Quando si salva un oggetto di R in un catalogo del linguaggio SAS e il *nome-raccolta* non è specificato come parte del percorso per l'opzione **CATALOG** dell'istruzione **SAVE**.
- Quando si carica un oggetto di R in un catalogo del linguaggio SAS e il *nome-raccolta* non è specificato come parte del percorso per l'opzione **CATALOG** dell'istruzione **LOAD**.

## TERMINATE

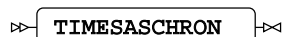
Specifica che l'ambiente di R non viene terminato quando si chiude la procedura.

» **TERMINATE** «  
» **TERM** «

Specificando **TERMINATE**, si arresta l'attuale ambiente di R anche se è stata specificata l'opzione di sistema **RKEEP**.

## TIMESASCHRON

Specifica se i valori dell'ora sono rappresentati in R con la classe `chron`.



Per impostazione predefinita, i valori di tempo sono archiviati nel tipo R `POSIXct`, che rappresenta un conteggio di secondi dalla mezzanotte. Quando è specificato `TIMESASCHRON`, i valori di tempo vengono archiviati in R come tipi `chron.times`.

Per usare l'opzione `TIMESASCHRON`, è necessario includere il pacchetto `chron` nell'ambiente di R tramite l'istruzione di R `library()`.

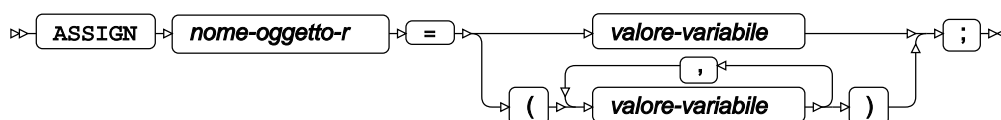
## Esempio

L'esempio seguente dimostra come usare `PROC R` per trovare la versione dell'interprete di R usato con WPS. Le informazioni sulla versione sono scritte nell'output di ODS.

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

## ASSIGN

È possibile utilizzare l'istruzione `ASSIGN` per assegnare i valori delle variabili del linguaggio SAS a un vettore R.



L'istruzione `ASSIGN` è usata per trasferire parametri a un programma R e il *nome-oggetto-r* specificato può essere usato in un programma nella riga o in un programma eseguito tramite il comando `EXECUTE`. Il *nome-oggetto-r* distingue tra maiuscole e minuscole, e a differenza delle variabili del linguaggio SAS, nel programma R si deve fare riferimento alle stesse maiuscole/minuscole della definizione `ASSIGN`.

È possibile generare le variabili tramite l'espansione o l'esecuzione di variabili della macro del linguaggio SAS. Questo consente di pre-elaborare le variabili usando il linguaggio SAS e trasferire il risultato a un programma R.

### *nome-oggetto-r*

Specifica uno o più valori di variabili da trasferire a un programma R. Il *nome-oggetto-r* specifica il nome della variabile a cui fa riferimento il *valore-variabile* nel programma R.

Il *nome-oggetto-r* può essere definito tramite un valore letterale di nome del linguaggio SAS ('oggettoR.nome'N) per creare un oggetto R che non sarebbe valido nel linguaggio SAS. Per esempio, per assegnare un valore Peter a un oggetto di R dipendente.nome, l'istruzione ASSIGN sarebbe:

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

È possibile assegnare più *valori-variabile* a un singolo oggetto del linguaggio R. In questo caso, tutti i valori devono essere dello stesso tipo e in un elenco separato da virgole tra parentesi.

## Trasferimento di più variabili a R

In questo esempio, più variabili vettoriali vengono trasferite da un programma in linguaggio SAS a un programma R che converte i vettori in un frame di dati.

```
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;
```

Ciò produce il seguente output:

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

## Assegnazione di una variabile della macro del linguaggio SAS a un oggetto di R

In questo esempio, la variabile della macro del linguaggio SAS *PARM* viene trasferita a un programma R per determinare la dimensione del campione in un campione generato casualmente.

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

Ciò produce il seguente output:

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

## ENDSUBMIT

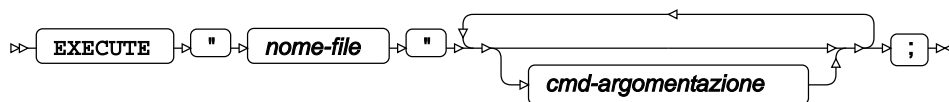
Specifica la fine di un programma in linguaggio R nella riga.



L'istruzione `ENDSUBMIT` deve essere inserita all'inizio di una nuova riga dopo il programma in linguaggio R.

## EXECUTE

Esegue un programma R archiviato in un file separato.



L'istruzione `EXECUTE` è un'alternativa all'uso dell'istruzione `SUBMIT`. Consente di inserire il codice R in un file separato e consente di eseguire lo stesso programma sia in WPS che in un ambiente di R interattivo.

### ***nome-file***

Una stringa tra virgolette contenente il percorso del file del programma R. Il *nome file* può essere un percorso assoluto o un percorso relativo.

Quando si usa Workbench per eseguire un programma in linguaggio R, se viene specificato un percorso relativo, la radice del percorso è l'area di lavoro. Ad esempio, per eseguire un file denominato `math.r` da un progetto denominato `calculate`, il percorso relativo è `calculate/math.r`.

### ***cmd-argomentazione***

Specifica un'argomentazione della riga di comando trasferita al programma R.

## Un esempio di esecuzione di un programma R archiviato in un file

In questo esempio, un programma R archiviato in un file esterno `model.r` viene eseguito nella procedura R. Il contenuto del file di origine `model.r`:

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



Il seguente programma crea un set di dati. Il set di dati viene trasferito al programma R tramite l'istruzione `EXPORT` prima che il programma venga eseguito tramite il file `model.r`:

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

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

## EXPORT

Consente di convertire un set di dati in linguaggio SAS in un `frame.dati` di R e di farne riferimento in un programma R.

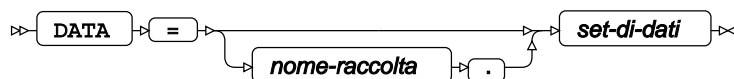


### Opzioni di esportazione

Le seguenti opzioni sono disponibili con l'istruzione `EXPORT`.

#### DATA

Specifica il nome del set di dati di WPS.



Il percorso della raccolta si può specificare mediante *nome-raccolta* nell'opzione `DATA` o l'opzione `LIB` dell'istruzione `PROC R`:

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC R` è specificata e *nome-raccolta* non è specificato, viene utilizzato il percorso nell'opzione `LIB`.
- Se non vengono specificati né *nome-raccolta*, né l'opzione `LIB` sull'istruzione `PROC R`, viene utilizzata la posizione `WORK`.

#### R

Specifica il nome del `frame.dati` usato nell'ambiente di R.



Se questa opzione non è specificata, il valore predefinito di *frame-dati-r* è il nome del *set di dati* specificato nell'opzione *DATA*. Se si utilizza il nome predefinito del *set di dati* in un programma in linguaggio R nella riga, il nome della variabile deve corrispondere alle maiuscole/minuscole utilizzate nell'opzione *DATA*.

## Un esempio di esportazione dei dati da WPS a R

Il seguente esempio crea un set di dati contenente due colonne numeriche. Il set di dati viene esportato nell'ambiente R e il contenuto del `frame.dati` viene stampato nell'output di ODS.

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

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

Ciò produce il seguente output:

```
'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

Consente la conversione di un `frame.dati` del linguaggio R in un set di dati del linguaggio SAS e il riferimento in un programma in linguaggio SAS.

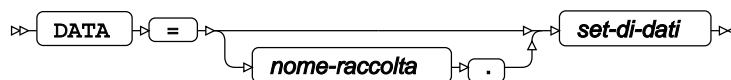


### Opzioni di importazione

Le seguenti opzioni sono disponibili con l'istruzione `IMPORT`.

#### DATA

Specifica il percorso e il nome del set di dati utilizzato nell'ambiente del linguaggio SAS di WPS Analytics.



Il percorso della raccolta si può specificare mediante *nome-raccolta* nell'opzione `DATA` o l'opzione `LIB` dell'istruzione `PROC R`:

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC R` è specificata e *nome-raccolta* non è specificato, viene utilizzato il percorso nell'opzione `LIB`.
- Se non vengono specificati né *nome-raccolta*, né l'opzione `LIB` sull'istruzione `PROC R`, viene utilizzata la posizione `WORK`.

Se questa opzione non è specificata, il valore predefinito del *set-di-dati* è il nome del *frame-dati-r* specificato nell'opzione `R`.

## R

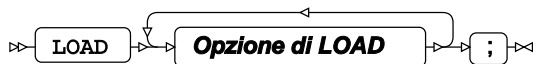
Specifica il nome del `frame.dati` usato nell'ambiente di R. È necessario specificarlo.



*nome-frame-r* distingue tra maiuscole e minuscole, e deve corrispondere alle maiuscole/minuscole utilizzate per il `frame.dati` importato nel programma R.

## LOAD

L'istruzione `LOAD` deserializza un *oggetto di R* archiviato in un catalogo del linguaggio SAS.



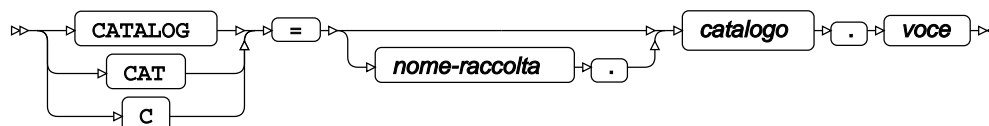
Le istruzioni `SAVE` e `LOAD` consentono di serializzare e archiviare un oggetto di R in un catalogo del linguaggio SAS e di deserializzarlo successivamente in una sessione di WPS. L'istruzione `SAVE` serializza un oggetto di R e lo archivia in una voce di un catalogo. L'istruzione `LOAD` deserializza un oggetto di R da un catalogo.

## Opzioni di caricamento

Le seguenti opzioni sono disponibili con l'istruzione `LOAD`.

### CATALOG

Specifica il catalogo da cui viene caricato l'oggetto di R archiviato.



Un catalogo del linguaggio SAS è definito tramite:

**nome-raccolta**

Specifica il nome della raccolta in cui viene archiviato il catalogo. Il percorso della raccolta si può specificare mediante *nome-raccolta* o l'opzione `LIB` dell'istruzione `PROC R`.

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC R` è specificata e *nome-raccolta* non è specificato, viene utilizzato il percorso nell'opzione `LIB`.
- Se non vengono specificati né *nome-raccolta*, né l'opzione `LIB` sull'istruzione `PROC R`, viene utilizzata la posizione `WORK`.

**catalogo**

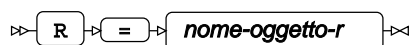
Specifica il nome del catalogo.

**voce**

Specifica il nome dell'oggetto di R nel catalogo.

**R**

Specifica il nome della variabile per l'oggetto di R caricato come usato nel programma del linguaggio R.



Poiché R distingue tra maiuscole e minuscole, il *nome-oggetto-r* deve corrispondere alle maiuscole/minuscole usate del nome della variabile di R. Il nome può essere specificato tramite la sintassi del valore letterale del nome (per esempio "r.oggetto.nome" `N`) se il nome dell'oggetto di R non segue le normali regole per gli identificatori nel linguaggio SAS.

## Un esempio dell'uso dell'istruzione `LOAD`

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

## SAVE

Consente agli oggetti di R di essere serializzati e archiviati in un catalogo del linguaggio SAS.



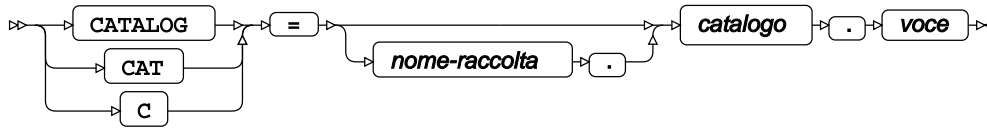
Le istruzioni `SAVE` e `LOAD` consentono di serializzare e archiviare un oggetto di R in un catalogo del linguaggio SAS e di deserializzarlo successivamente in una sessione di WPS. L'istruzione `SAVE` serializza un oggetto di R e lo archivia in una voce di un catalogo. L'istruzione `LOAD` deserializza un oggetto di R da un catalogo.

## Opzioni di salvataggio

Le seguenti opzioni sono disponibili con l'istruzione `SAVE`.

### CATALOG

Specifica il percorso in cui l'oggetto di R verrà salvato.



Il percorso della raccolta si può specificare mediante il *nome-raccolta* o l'opzione `LIB` dell'istruzione `PROC R`:

#### *nome-raccolta*

Specifica il nome della raccolta in cui viene archiviato il catalogo. Il percorso della raccolta si può specificare mediante *nome-raccolta* o l'opzione `LIB` dell'istruzione `PROC R`.

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC R` è specificata e *nome-raccolta* non è specificato, viene utilizzato il percorso nell'opzione `LIB`.
- Se non vengono specificati né *nome-raccolta*, né l'opzione `LIB` sull'istruzione `PROC R`, viene utilizzata la posizione `WORK`.

#### *catalogo*

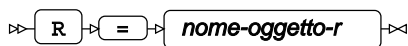
Specifica il nome del catalogo.

#### *voce*

Specifica il nome dell'oggetto di R nel catalogo.

### R

Specifica il nome dell'oggetto di R da salvare nel catalogo. È necessario specificarlo.



Poiché `R` distingue tra maiuscole e minuscole, le maiuscole/minuscole del *nome-oggetto-r* devono corrispondere al nome dell'oggetto nel programma R. Il nome può essere specificato tramite la sintassi del valore letterale del nome (per esempio `"r.oggetto.nome"N`) se il nome dell'oggetto di R non segue le normali regole per gli identificatori nel linguaggio SAS.

### DESCRIPTION

Specifica una stringa descrittiva salvata con la voce del catalogo.



La descrizione viene visualizzata nell'output dall'istruzione `PROC CATALOG`.

La voce del catalogo avrà un tipo di `OBJECT`.

## Un esempio di salvataggio di un oggetto di R in un catalogo WPS

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

## SUBMIT

Specifica l'inizio di un programma in linguaggio R nella riga.



Un programma in linguaggio R nella riga è definito come parte della procedura R in un programma in linguaggio SAS. L'istruzione `SUBMIT` contrassegna l'inizio del programma e l'istruzione `ENDSUBMIT` contrassegna la fine.

Un programma in linguaggio R deve iniziare su una nuova riga dopo l'istruzione `SUBMIT` e l'istruzione `ENDSUBMIT` deve apparire separatamente all'inizio di una riga.

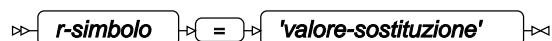
Più programmi nella riga del linguaggio R possono esistere nell'ambiente di una singola procedura R. Ogni programma in linguaggio R viene eseguito così come viene trovato. Le variabili definite in un programma in linguaggio R possono essere utilizzate nei successivi programmi nella riga nell'ambiente della stessa procedura R.

## Opzioni di SUBMIT

È possibile utilizzare la seguente opzione con l'istruzione `SUBMIT`.

### *r-simbolo*

Abilita la sostituzione di un simbolo nel programma R con una stringa.



Prima di essere trasferito all'ambiente di R, un programma R nella riga viene pre-elaborato per sostituire qualsiasi *r-simbolo* definito nel programma con il contenuto del *valore-sostituzione*. Se il *valore-sostituzione* è una stringa, deve essere racchiuso tra virgolette. Se il *valore-sostituzione* è una variabile della macro del linguaggio SAS, deve essere preceduta da una e commerciale (&), ma le virgolette non sono necessarie.

Il nome di *r-simbolo* distingue tra maiuscole e minuscole e se ne deve far riferimento nel programma R usando le stesse maiuscole/minuscole della definizione dell'istruzione `SUBMIT`.

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

La sintassi per la sostituzione di un *r-simbolo* definito è identica a quella usata per la sostituzione di una variabile della macro del linguaggio SAS, il nome di *r-simbolo* definito è preceduto da una commerciale (&) nel codice incorporato.

## Istruzione `SUBMIT` – esempio di base

L'esempio seguente dimostra come incorporare un programma in linguaggio R in un programma in linguaggio SAS:

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

Che produce il seguente output:

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

## Istruzione `SUBMIT` – uso di una variabile della macro in un programma R

Il seguente esempio dimostra come trasferire una variabile da un programma in linguaggio SAS a un programma in linguaggio R. La variabile *welcome* viene definita tramite la macro `%LET` del linguaggio SAS. La variabile della macro specificata viene assegnata al *r-simbolo* dell'istruzione `SUBMIT`. Si fa riferimento al *r-simbolo* nel programma in linguaggio R incorporato:

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

Che produce il seguente output:

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

# Avvisi legali

(c) 2022 World Programming, an Altair Company

Le presenti informazioni sono riservate e soggette a diritto d'autore. Non è possibile riprodurre o trasmettere nessuna parte di questa pubblicazione, in qualsiasi forma o con qualsiasi mezzo, elettronico o meccanico, inclusa la fotocopiatura, la registrazione o eventuali sistemi di archiviazione e recupero dati.

## Marchi commerciali

WPS e World Programming sono marchi registrati o marchi commerciali di World Programming Limited nell'Unione europea e altri paesi. (r) o ® indica un marchio comunitario.

SAS e tutti gli altri nomi di prodotti o servizi di SAS Institute Inc. sono marchi registrati o marchi commerciali di SAS Institute Inc. negli Stati Uniti e in altri paesi. ® indica la registrazione negli USA.

Tutti gli altri marchi commerciali sono proprietà dei rispettivi titolari.

## Avvisi generali

World Programming Limited non è associata in alcun modo a SAS Institute.

WPS non è SAS System.

Le frasi "SAS", "linguaggio SAS" e "linguaggio di SAS" utilizzate in questo documento si usano in riferimento al linguaggio di programmazione spesso denominato in uno dei suddetti modi.

Le frasi "programma", "programma SAS" e "programma in linguaggio SAS" utilizzate in questo documento si riferiscono a programmi scritti in linguaggio SAS. Questi possono anche essere denominati "script", "script SAS" o "script in linguaggio SAS".

Le frasi "IML", "linguaggio IML", "sintassi IML", "Interactive Matrix Language" e "linguaggio di IML" utilizzate in questo documento si usano in riferimento al linguaggio di programmazione spesso denominato in uno dei suddetti modi.

WPS include software sviluppato da terzi. È possibile trovare maggiori informazioni nel file THANKS o acknowledgments.txt inclusi nell'installazione di WPS.