

Procedura Python di WPS

guida dell'utente e riferimento

Versione: 4.4.4

(c) 2022 World Programming, an Altair Company

www.worldprogramming.com

Indice

Procedura PYTHON.....	3
Introduzione.....	3
Installazione e configurazione.....	4
Uso di Python con WPS.....	5
Conversione del tipo di dati.....	6
Importare i moduli di Python personalizzati.....	8
Uso della grafica creata da Python.....	10
Esempio.....	11
Riferimento alla procedura Python.....	12
PROC PYTHON.....	13
EXECUTE.....	14
EXPORT.....	16
IMPORT.....	17
SUBMIT.....	18
ENDSUBMIT.....	19
Avvisi legali.....	20

Procedura PYTHON

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

La combinazione del linguaggio Python 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, utilizzando allo stesso tempo le funzionalità presenti nel linguaggio Python.

Introduzione ↗	3
La procedura Python consente ai programmi in linguaggio SAS di includere il codice scritto in linguaggio Python.	
Installazione e configurazione ↗	4
Impostazione dell'ambiente di Python per WPS.	
Uso di Python con WPS ↗	5
L'uso di Python in un programma in linguaggio SAS consente di utilizzare pacchetti di Python specialistici, quali <code>Scikit-learn</code> o <code>Tensorflow</code> .	
Esempio ↗	11
Dimostra come utilizzare un set di dati in linguaggio SAS in <code>PROC PYTHON</code> per creare un diagramma del grafico a dispersione.	
Riferimento alla procedura Python ↗	12
Descrive la sintassi e le opzioni per <code>PROC PYTHON</code> e le sue istruzioni contenute.	

Introduzione

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

Combinando il linguaggio Python e il linguaggio SAS, si può:

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

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

Installazione e configurazione

Impostazione dell'ambiente di Python per WPS.

Quando si usa Python con WPS:

- L'interprete di Python a 32 bit è necessario per la versione di WPS a 32 bit e l'interprete di Python a 64 bit è necessario per la versione di WPS a 64 bit.
- È necessario installare il pacchetto `pandas` con l'interprete Python. Questo si può controllare tramite l'utilità `pip` eseguendo `pip list` nella riga di comando.

La distribuzione di WPS non include l'interprete Python o il pacchetto `pandas`. Se Python non è installato, è possibile ottenere una copia dell'interprete Python da <https://www.python.org> o installare un ambiente di Python in pacchetto che include i moduli necessari.

La procedura `PYTHON` si può utilizzare con la versione 3.5.0 di Python e versioni successive, ed è attualmente supportata su Microsoft Windows, sui sistemi basati su Linux e macOS.

La procedura non è attualmente supportata in mainframe IBM.

Variabili di ambiente di Python

È necessario impostare la variabile di ambiente di `PYTHONHOME` in modo che WPS trovi e utilizzi l'interprete Python. Tale variabile deve fare riferimento alla cartella in cui si trova la raccolta principale di Python, ad esempio, `python35.dll` in Microsoft Windows.

Flussi di output ed errore standard

Il flusso di output standard Python (`sys.stdout`) e il flusso di errori standard (`sys.stderr`) vengono reindirizzati all'output WPS quando viene eseguita la procedura:

- `sys.stderr` viene reindirizzato al file di registro WPS.
- `sys.stdout` viene reindirizzato al file di elenco WPS quando WPS viene eseguito sulla riga di comando o a tutte le destinazioni di output ODS specificate quando viene utilizzato Workbench.

Se si utilizza la funzione `print()` di Python per trasferire l'output al file di elenco di WPS, il numero di caratteri che vengono visualizzati nell'elenco è determinato dall'opzione di sistema `LINESIZE` di WPS. L'opzione `LINESIZE` può servire per stampare le stringhe fino a 256 caratteri; l'uso della funzione `print()` dovrebbe quindi essere per informazioni limitate come le istruzioni di registro.

Se si desidera restituire grandi volumi di contenuti generati da Python in WPS, creare un DataFrame contenente il contenuto richiesto e importare il DataFrame in WPS mediante l'istruzione `IMPORT`.

L'esempio seguente crea un elenco di funzioni disponibili nel pacchetto panda di Python. L'elenco viene stampato mediante `print (fnList)` e anche convertito in un `DataFrame` e importato in WPS.

```
PROC PYTHON;
SUBMIT;
import inspect
fnPandas = inspect.getmembers(pandas, inspect.isfunction)
fnList = [fn[0] for fn in fnPandas]
print (fnList)
fnTable = pandas.DataFrame({'function': fnList})
ENDSUBMIT;
IMPORT DATA=PANDAFN PYTHON=fnTable;
RUN;
```

L'output stampato tronca l'elenco di funzioni:

```
['Expr', 'Term', 'bdate_range', 'concat', 'crosstab',
 'cut', 'date_range', 'eval', 'factorize',
```

Quando vengono convertiti in un `DataFrame` e importati come set di dati WPS, vengono elencate tutte le funzioni disponibili.

Uso di Python con WPS

L'uso di Python in un programma in linguaggio SAS consente di utilizzare pacchetti di Python specialistici, quali `Scikit-learn` o `Tensorflow`.

La prima volta in cui si richiama la procedura Python in un programma in linguaggio SAS, WPS importa automaticamente i pacchetti `pandas` e `numpy`. È possibile accedere alla funzionalità nei pacchetti `pandas` e `numpy` in un programma in linguaggio Python nella riga, scritto tra le istruzioni `SUBMIT` e `ENDSUBMIT`, facendo riferimento al pacchetto completo, alla classe o al nome della funzione, per esempio:

```
PROC PYTHON;
SUBMIT;
content = pandas.read_csv('file:///C:/project/sourcedata/wps.csv')
...
ENDSUBMIT;
RUN;
```

In alternativa, puoi utilizzare l'istruzione `import ... as ...` per creare l'alias del nome del pacchetto `pandas` o `numpy`, ad esempio:

```
PROC PYTHON;
SUBMIT;
import pandas as pd
content = pd.read_csv('file:///C:/project/sourcedata/wps.csv')
...
ENDSUBMIT;
RUN;
```

È possibile importare ed utilizzare altri pacchetti di Python all'interno del codice Python in linea, ad esempio:

```
PROC PYTHON;
  EXPORT DATA=source;
  SUBMIT;
import statsmodels.formula.api as lm
result = lm.ols(formula='x ~ y + z', data=source).fit()
...
  ENDSUBMIT;
RUN;
```

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

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

Conversione del tipo di dati

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

Questa sezione descrive la corrispondenza tra i formati di WPS e i tipi di dati panda di Python. WPS ha molti formati che influenzano l'output e la visualizzazione dei dati. Quando si scrivono dati su un pandas DataFrame con l'istruzione EXPORT della procedura Python, i dati formattati vengono convertiti in un tipo di dati equivalente e panda o numpy.

Molti formati influenzano solo il layout dell'output dei dati, come l'aggiunta di simboli di valuta o separatori di virgola, e questi formati non hanno alcun effetto durante la scrittura dei dati.

Dati non formattati in linguaggio SAS in Python

I dati non formattati vengono convertiti in un tipo di pandas DataFrame come segue:

Formato WPS	Tipo di dati Python	Note
Numero non formattato	float64	
Stringa non formattata	object	La lunghezza massima dell'oggetto per una variabile non è nota come parte dei metadati del set di dati.

Dati formattati in linguaggio SAS in Python – numeri

I formati numerici di base vengono convertiti in un tipo di pandas DataFrame come segue:

Formato WPS	Tipo di dati Python	Note
<i>w.d</i>	float64	
BEST. e BEST <i>w.</i>	float64	
FLOAT <i>w.d</i>	float64	

Dati formattati in linguaggio SAS in Python – stringhe

I formati di tipo carattere di base vengono convertiti in un tipo di pandas DataFrame come segue:

Formato WPS	Tipo di dati Python	Note
\$ <i>w.</i> \$CHAR <i>w.</i> \$F <i>w.</i>	object	La lunghezza massima dell'oggetto per una variabile non è nota come parte dei metadati del set di dati.

Dati formattati in linguaggio SAS in Python – date e ore

I formati di tipo ora, data/ora e ora vengono convertiti in un tipo di pandas DataFrame come segue:

Formato WPS	Tipo di dati Python	Note
DATE <i>w.</i>	datetime64 [ns]	Tipo di data/ora Numpy.
DDMMYY <i>w.</i> e tutte le varianti (come DDMMYYB <i>w.</i> , MMDDYY <i>w.</i> e YYMM <i>w.</i>)	datetime64 [ns]	Tipo di data/ora Numpy.
DTDATE <i>w.</i> e tutte le varianti (come DTMONYY <i>w.</i> e DTWKDATX <i>w.</i>)	datetime64 [ns]	Tipo di data/ora Numpy.
TIME <i>w.</i> , HOUR <i>w.</i> , HHMM <i>w.</i> e tutti i formati di ora simili.	float64	
JULIAN <i>w.</i> e tutti i formati di data simili.	datetime64 [ns]	Tipo di data/ora Numpy.

Tipi di dati panda Python al set di dati del linguaggio SAS

I set di dati del linguaggio SAS contengono solo dati numerici e di carattere. I formati possono essere applicati ai dati nel set di dati per rappresentare in modo più ravvicinato i dati di origine da Python.

L'importazione di un pandas DataFrame mediante l'istruzione `IMPORT` converte i dati come segue:

Tipo di dati Pandas	Formato WPS	Note
Oggetto	Carattere	La lunghezza massima dell'oggetto viene calcolata prima dell'importazione.
int64	Numerico	
bool	Numerico	<i>TRUE</i> viene convertito in 1; <i>False</i> viene convertito in zero (0).
Float64	Numerico	
datetime64[ns]	Numerico	Formattato come DATETIME19.

Non è possibile rappresentare alcuni valori in Python nello stesso modo in cui vengono rappresentati in un set di dati del linguaggio SAS. La seguente tabella mostra come vengono convertiti questi valori:

Valore dei dati Pandas	Valore di WPS	Note
null string (' ')	' '	Valore di carattere mancante.
True	1	Valore numerico
False	0 (zero)	Valore numerico
NaN Definito con <code>float('nan')</code>	.	Valore numerico mancante
Infinito Definito con <code>float('inf')</code> Definito con <code>float('-inf')</code>	.	Valore numerico mancante

Importare i moduli di Python personalizzati

Come usare i propri pacchetti e moduli in Python.

Per usare pacchetti personalizzati, devono essere accessibili in Python. I percorsi verso ubicazioni contenenti pacchetti personalizzati sono specificati dalla variabile `sys.path` di Python. La variabile `sys.path` costruisce un elenco di percorsi da cercare usando le variabili di ambiente di `PYTHONHOME` e `PYTHONPATH`, oppure è possibile specificare i percorsi modificando il valore della variabile durante l'esecuzione di un programma.

- `sys.path` contiene un elenco di cartelle in cui è stata eseguita la ricerca per i pacchetti. La variabile è costruita dai valori in `PYTHONHOME` e `PYTHONPATH`.
- `PYTHONHOME` specifica il percorso dell'interprete Python e delle raccolte standard, inclusi i pacchetti installati in *pacchetti-del-sito* usando un gestore di pacchetti come `PIP`. È necessario specificare la variabile per consentire a WPS di interagire con Python.
- `PYTHON` specifica un elenco di cartelle per la ricerca dei pacchetti. Questo elenco è anteposto all'elenco di ricerca definito in `sys.path`.

Il primo elemento in `sys.path` è la directory contenente il programma Python o una stringa vuota, interpretata da Python come directory corrente. Se viene eseguito da WPS, `sys.path[0]` contiene il primo percorso di ricerca specificato in `PYTHONPATH` o `PYTHONHOME`. Se si programmano i pacchetti di riferimenti nella directory corrente, è necessario modificare `sys.path` quando `PROC PYTHON` è in esecuzione. Consultare [Modifica della variabile `sys.path`](#) (pag. 9)

Impostare `PYTHON` prima di eseguire WPS

È possibile definire la variabile `PYTHON` come variabile di sistema e utilizzarla con tutte le applicazioni presenti nel proprio dispositivo che esegue Python.

Se si dispone di più installazioni di Python, ad esempio Python 2 e Python 3, l'impostazione di `PYTHONPATH` mediante una variabile di sistema potrebbe far sì che il programma tenti di importare la versione errata di un pacchetto. In queste circostanze, è necessario impostare la variabile come parte del programma in linguaggio SAS eseguito in WPS.

Impostare `PYTHONPATH` in un programma in linguaggio SAS

Il `PYTHONPATH` può essere impostato in un programma in linguaggio SAS mediante l'opzione di sistema `SET`. Ad esempio, per utilizzare i pacchetti archiviati nella cartella `C:\temp\python`, è possibile aggiungere quanto segue ad un programma prima dell'istruzione `PROC PYTHON`:

```
OPTIONS SET = PYTHONPATH 'C:\temp\python';
```

Se `PYTHONHOME` è specificato come `C:\python3`, il contenuto della variabile `sys.path` di Python è:

```
['C:\\temp\\python', 'C:\\python3\\python37.zip',  
'C:\\python3\\DLLs', 'C:\\python3\\lib', 'C:\\python3',  
'C:\\python3\\lib\\site-packages']
```

Modifica della variabile `sys.path`

La variabile `sys.path` di Python si può modificare programmaticamente aggiungendo quanto segue a un programma in linguaggio Python:

```
import os, sys  
sys.path.append(os.getcwd())
```

Se un programma in linguaggio Python eseguito mediante l'istruzione `EXECUTE` include altri file Python, aggiungere il precedente come programma nella riga prima del programma eseguito per consentire a WPS di individuare i file inclusi:

```
PROC PYTHON;  
  SUBMIT;  
import os, sys  
sys.path.append(os.getcwd())  
  ENDSUBMIT;  
  EXECUTE 'programs/dataCollect.py';  
RUN;
```

Uso della grafica creata da Python

I grafici generati mediante la funzionalità in Python vengono acquisiti e possono essere inclusi nell'output ODS standard della sessione di WPS.

WPS crea una variabile temporanea, *wpsgloc* che punta a una cartella temporanea utilizzata per archiviare i grafici per l'inclusione all'output ODS. La variabile deve essere aggiunta al nome del file di immagine ogni volta che si crea un'immagine. È possibile farlo usando, ad esempio, la funzione di Python `os.path.join()`:

```
PROC PYTHON;
SUBMIT;
import os
import matplotlib.pyplot as plt
...
plt.savefig(os.path.join(wpsgloc, 'my_image.png'))
ENDSUBMIT;
RUN;
```

Non è possibile modificare la variabile *wpsgloc* nel programma in linguaggio SAS ed è:

- Se si esegue in Workbench, il percorso `WORK`, ad esempio:

```
C:\Users\user-id\AppData\Local\Temp\WPS Temporary Data
```

- La directory di lavoro quando WPS Analytics viene eseguito sulla riga di comando.

```
ODS PDF FILE='scatter_plot.pdf';
PROC PYTHON;
  EXPORT DATA=STATS PYTHON=df;
  SUBMIT;
import os
import matplotlib.pyplot as plt
df.plot(kind='scatter', x='i', y='j')
plt.savefig(os.path.join(wpsgloc, 'scatter.png'))
  ENDSUBMIT;
RUN;
ODS PDF CLOSE;
```

wpsgloc è un percorso temporaneo per le immagini grafiche utilizzate in Workbench.

Se si esegue sulla riga di comando, i grafici vengono creati nella directory corrente. Si fa riferimento ai grafici nell'output HTML che vengono copiati in output PDF.

Esempio

Dimostra come utilizzare un set di dati in linguaggio SAS in PROC PYTHON 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 passare il set di dati all'ambiente di Python. Il set di dati viene convertito in un pandas DataFrame come parte dell'esportazione e il DataFrame viene utilizzato per creare un grafico a dispersione con Matplotlib.

Una destinazione del file PDF di output viene creata con il sistema di consegna dell'output SAS (ODS). L'aggiunta di PDF alle destinazioni di output include il file di immagine del grafico a dispersione restituito nell'output PDF. Il PDF viene salvato ed è possibile visualizzare l'output in un visualizzatore di PDF.

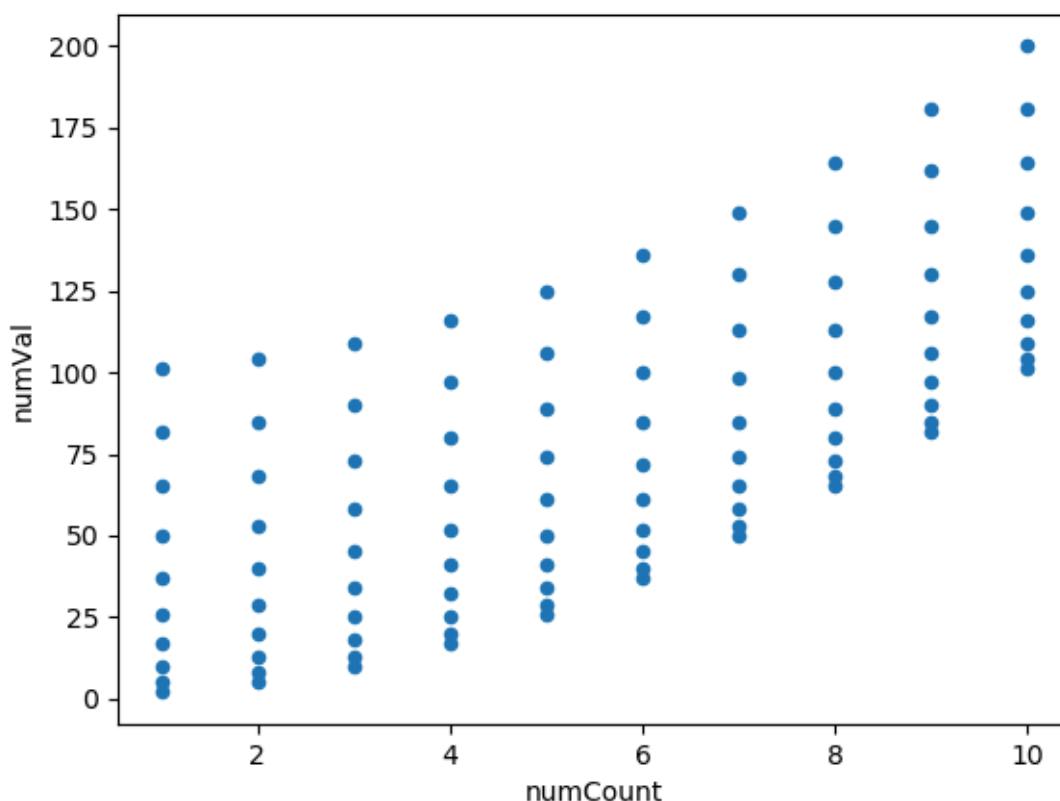
Questo esempio richiede i seguenti pacchetti di Python:

- SciPy
- Matplotlib

```
ODS PDF FILE='scatter_plot.pdf';
DATA stats (drop = count);
  DO count=1 TO 10;
    DO numCount=1 TO 10;
      numVal = (numCount*numCount)+(count*count);
      OUTPUT;
    END;
  END;
RUN;

PROC PYTHON;
  EXPORT DATA=STATS PYTHON=df;
  SUBMIT;
import os
import matplotlib.pyplot as plt
df.plot(kind='scatter', x='numCount', y='numVal')
plt.savefig(os.path.join(wpsgloc, 'scatter.png'))
  ENDSUBMIT;
RUN;
ODS PDF CLOSE;
```

Questo crea il seguente grafico a dispersione nell'output PDF di ODS:



Riferimento alla procedura Python

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

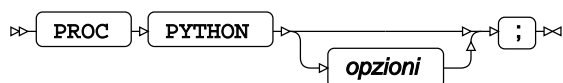
PROC PYTHON ↗	13
Richiama l'ambiente di Python che consente l'esecuzione di programmi in linguaggio Python nella riga o esterni.	
EXECUTE ↗	14
Esegue un programma Python archiviato in un file separato.	
EXPORT ↗	16
Converte un set di dati in linguaggio SAS in un pandas DataFrame.	
IMPORT ↗	17
Consente la conversione di un pandas DataFrame in un set di dati del linguaggio SAS e il riferimento in un programma in linguaggio SAS.	

SUBMIT [↗](#)..... 18
 Specifica l'inizio di un programma in linguaggio Python nella riga.

ENDSUBMIT [↗](#)..... 19
 Specifica la fine di un programma in linguaggio Python nella riga.

PROC PYTHON

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



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

È possibile scrivere un programma Python nella riga della procedura `PYTHON` oppure eseguirlo da un file separato:

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

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

Quando si richiama l'ambiente di Python, i pacchetti `pandas` e `numpy` vengono caricati automaticamente. Questi pacchetti si possono utilizzare citando il nome completo del pacchetto nel codice o usando l'istruzione `import ... as ...` per creare l'alias del nome del pacchetto.

Opzioni

Le seguenti opzioni sono disponibili con l'istruzione `PROC PYTHON`.

KEEP

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



Se specificato, l'attuale ambiente di Python resta attivo quando la procedura attuale viene chiusa e l'ambiente viene utilizzato nella successiva invocazione della procedura Python nello 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 Python al termine della procedura. Specificando `KEEP`, si mantiene l'attuale ambiente di Python, inclusi i moduli caricati durante l'esecuzione di un programma Python, da utilizzare alla prossima invocazione della procedura `PYTHON`.

È possibile specificare l'opzione di sistema `PYTHONKEEP` per utilizzare lo stesso ambiente di Python per la durata dell'esecuzione del programma in linguaggio SAS.

LIB

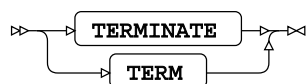
Specifica il percorso della raccolta per i set di dati utilizzati nella procedura `PYTHON`. Il percorso predefinito è la biblioteca `WORK`.



Il percorso `riferimento-raccolta` viene utilizzato quando `nome-raccolta` non è specificato come parte del percorso per l'opzione `DATA` delle istruzioni `EXPORT` o `IMPORT`.

TERMINATE

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



Specificando `TERMINATE`, si arresta l'attuale ambiente di Python anche se è stata specificata l'opzione di sistema `PYTHONKEEP`. Una successiva invocazione di `PROC PYTHON` nello stesso programma ha solo pacchetti `pandas` e `numpy` predefiniti caricati.

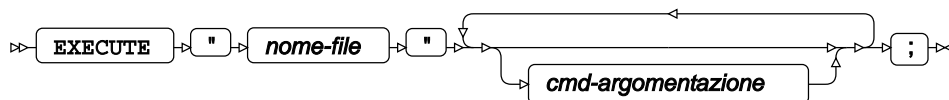
Esempio

L'esempio seguente mostra come richiamare la procedura `PYTHON` per stampare `hello world` nell'output ODS.

```
PROC PYTHON;
SUBMIT;
print ('Hello World')
ENDSUBMIT;
RUN;
```

EXECUTE

Esegue un programma Python archiviato in un file separato.



L'istruzione `EXECUTE` è un'alternativa all'uso dell'istruzione `SUBMIT`. Consente l'inserimento del codice Python in un file separato da eseguire in WPS Analytics.

Opzioni di esecuzione

Le seguenti opzioni sono disponibili con l'istruzione EXECUTE.

nome-file

Una stringa, tra virgolette, contenente il percorso del file del programma Python. *nome-file* può essere un nome di percorso assoluto o relativo.

In Workbench, il percorso per i nomi di percorsi dei file relativi è l'area di lavoro. Ad esempio, per eseguire un file denominato `myProgram.py` da un progetto denominato `python`, il percorso relativo è `/python/myProgram.py`.

cmd-argomentazione

Specifica un'argomentazione della riga di comando trasferita al programma Python. Le argomentazioni sono accessibili nella programmazione da `sys.argv`.

Tutte le argomentazioni della riga di comando vengono trasferite al programma Python come stringhe. Le argomentazioni numeriche devono essere racchiuse tra virgolette e il programma Python deve convertire le argomentazioni nel tipo numerico richiesto.

Esempio di base

In questo esempio, un programma Python archiviato in un file esterno viene eseguito nella procedura PYTHON. Si fa riferimento al file mediante il percorso assoluto:

```
PROC PYTHON;  
  EXECUTE 'C:\temp\printDS.py';  
RUN;
```

Esempio – specificare un'argomentazione a un programma Python

In questo esempio, viene eseguito un programma Python esterno `multiple.py` che restituisce il quadrato di un valore specificato. Il valore viene specificato in una variabile.

```
import sys  
  
def multiply (value):  
    return value*value  
  
print(multiply(int(sys.argv[0])))
```

`multiple.py` viene eseguito dalla procedura PYTHON in un programma in linguaggio SAS e il valore richiesto da moltiplicare viene trasferito al programma Python come argomentazione all'istruzione EXECUTE. L'argomentazione numerica `6` viene trasferita come stringa. Il programma Python usa la funzione `int()` integrata per restituire un numero intero prima di calcolare il quadrato.

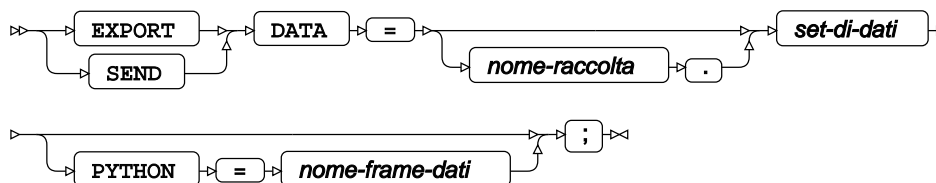
```
PROC PYTHON;  
  EXECUTE 'C:/temp/multiple.py' '6';  
RUN;
```

Ciò produce il seguente output ODS:

```
36
```

EXPORT

Converte un set di dati in linguaggio SAS in un pandas `DataFrame`.



La preparazione del set di dati dovrebbe essere completata prima di esportare i dati in Python. Ciò consente di utilizzare la capacità di elaborazione del set di dati del linguaggio SAS per creare un set di dati di esportazione contenente solo i dati richiesti per il programma in linguaggio Python.

Opzioni di esportazione

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

DATA

Specifica il percorso e il nome del set di dati del set di dati in linguaggio SAS da convertire.

La posizione della libreria si può specificare mediante *nome-raccolta* nell'opzione `DATA` o l'opzione `LIB` dell'istruzione `PROC PYTHON`.

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC PYTHON` 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 PYTHON`, viene utilizzata la posizione `WORK`.

PYTHON

Specifica il nome del pandas `DataFrame` così come viene usato nel programma in linguaggio Python.

Il nome del pandas `DataFrame` nel codice in linguaggio Python fa distinzione tra maiuscole e minuscole, e il *nome-frame-dati* specificato deve corrispondere al caso utilizzato del nome della variabile Python.

Se questa opzione non è specificata, il valore predefinito di *nome-frame-dati* 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 Python nella riga, il nome della variabile deve corrispondere alle maiuscole/minuscole utilizzate nell'opzione `DATA`.

Esempio – esporta un set di dati in linguaggio SAS in un pandas DataFrame.

L'esempio seguente crea un set di dati in un DATA step in linguaggio SAS. Il set di dati viene quindi esportato in un pandas DataFrame e i tipi di colonna vengono stampati.

```
DATA TESTDATA;
INFILE CARDS DLM='#';
INPUT num char $;
CARDS;
1 # Hello
2 # World
;
RUN;

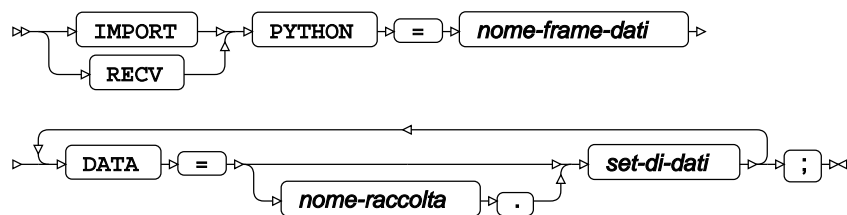
PROC PYTHON;
EXPORT DATA=TESTDATA PYTHON=dframe;
SUBMIT;
print (dframe)
ENDSUBMIT;
RUN;
```

Che scrive quanto segue nell'output di ODS:

	num	char
0	1.0	Hello
1	2.0	World

IMPORT

Consente la conversione di un pandas DataFrame 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 PYTHON`.

- Se viene specificato *nome-raccolta*, quel percorso viene sempre usato.
- Se l'opzione `LIB` dell'istruzione `PROC PYTHON` 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 PYTHON`, viene utilizzata la posizione `WORK`.

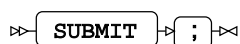
PYTHON

Specifica il nome del pandas `DataFrame` così come viene usato nell'ambiente Python. È necessario specificarlo.

nome-frame-dati distingue tra maiuscole e minuscole, e deve corrispondere alle maiuscole/minuscole utilizzate per il pandas `DataFrame` importato nel programma Python.

SUBMIT

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



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

Il programma in linguaggio Python deve iniziare su una nuova riga dopo l'istruzione `SUBMIT` e l'istruzione `ENDSUBMIT` deve apparire separatamente all'inizio di una riga. La prima riga di codice del programma Python non deve iniziare con uno spazio ed eventuali istruzioni successive devono seguire i normali requisiti di Python per il rientro, ad esempio:

```
PROC PYTHON;
  SUBMIT;
  fruits = ['apple', 'banana', 'cherry', 'damson', 'elderberry', 'fig']
  for fruit in fruits:
    print(fruit)
  ENDSUBMIT;
RUN;
```

In una singola procedura `PYTHON`, possono esistere più programmi in linguaggio Python nella riga. Ogni programma in linguaggio Python viene eseguito così come viene trovato. Le variabili definite in un programma in linguaggio nella riga possono essere utilizzate nei successivi programmi nella riga nella stessa procedura Python.

ENDSUBMIT

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

» `ENDSUBMIT` ; «

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

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.