How can I write a Python program to rename variables in SPSS?

Authored by

stats writer

June 30, 2024

Python is a programming language that can be used to manipulate data in SPSS, a statistical software. To rename variables in SPSS using Python, one can use the "RenameVariables" function from the "spss" library. This function allows the user to specify the old and new variable names, and the program will automatically rename them. This can be useful for organizing and cleaning up data sets in SPSS. By utilizing Python's capabilities, users can efficiently and effectively rename variables in SPSS without having to manually change each one.

How can I write a Python program to rename variables in SPSS? | SPSS FAQ

NOTE: This page was created using SPSS version 18.0.2.

Variables can be easily renamed using the SPSS command rename.
However, if you would like to rename many variables in your dataset by simply
adding a suffix to the variable name, this can be easily done with a simple
Python program. Before running the examples below, you will need to
install Python and the SPSS Python plug-in. These can be obtained from the
SPSS website.

Example 1: Defining a function

We will start by defining a very simple function using

**Python. This
function, which we call myprint, prints the string that you provide.**

```
begin program.
def myprint(mystring):
print mystring

myprint("Hello, World!")
myprint("d:dataelemapi2.sav")
end program.
```

**We start all Python programs with the SPSS command begin program, and
we end the Python program with the SPSS command end program.
Because these are SPSS commands, they end with a period. On the second line,
we define the function myprint, and the argument for that function is
called mystring. Next, we use the Python print command to
print mystring. Note that the print command must be indented. We then make two calls to myprint. First, we**

give
the argument "Hello, World!", and in the second call we give a path and data
file name. Let's see the output from this program.

```
Hello, World!
d:dataelemapi2.sav
```

Example 2: Renaming variables using Python

In the next example, we will rename several variables by adding a suffix (in
this example, the suffix will be _new) to
the current variable name. We will use the elemapi2.sav dataset in this example and in the following examples.

```
begin program.
import spss, spssaux
spssaux.OpenDataFile('d:dataelemapi2.sav')
vdict=spssaux.VariableDict()
mylist=vdict.range(start="grad_sch", end="enroll")
nvars = len(mylist)

for i in range(nvars):
```

```
myvar = mylist
mynewvar = myvar+"_new"
spss.Submit(r"""
rename variables ( %s = %s) .
""" %(myvar, mynewvar))
end program.
```

As before, we start with the SPSS command begin program. Next we import
two modules, spss and spssaux. We will use functions from
these libraries in the program. On the third line, we open the data file.
In this example, the data file is called elemapi2.sav, and it is located
in a directory called 'D:data'. On the fourth line, we put the names of
the variables in the dataset into a Python variable called vdict.
Next, we create a Python variable called mylist that contains only the
names of the variables from our start variable (which is grad_sch in this
example) to our end variable (which is enroll in this

example). On
the next line, we create a variable called nvars which contains the
number of variables in mylist.

In the next part of the program, we use a loop to rename each of the
variables contained in myvar. On the fourth line of the loop, we
use the spss.Submit function so that we can use the SPSS rename
variables command to rename the variables. We specify that the SPSS
command should be read by Python as a raw string by using starting the string
with r""" (the r means "raw"; we could have used R). The
%s is used
to represent a Python variable to SPSS. The first %s refers to the
Python variable myvar, and the second %s refers to the Python
variable mynewvar, as indicated in the parentheses after the close of the

triple quotes. Please note that the indenting in the loop is necessary.

Example 3: Renaming variables with a Python function

In this example, we will write a Python function called renamefun to
rename the variables. This function will have four arguments. The
location and name of the dataset will be given in datapath; the first
variable to be renamed will be given in mystart; the last variable to be
renamed will be given in myend; the suffix to be used to rename the
variables will be given in suffix. Each of these arguments must be
given in quotes.

```
begin program.
import spss, spssaux
def renamefun(datapath, mystart, myend, suffix):
spssaux.OpenDataFile(datapath)
vdict=spssaux.VariableDict()
mylist=vdict.range(start=mystart, end=myend)
```

```
nvars = len(mylist)
for i in range(nvars):
myvar = mylist
mynewvar = myvar+suffix
spss.Submit(r"""
rename variables ( %s = %s) .
""" %(myvar, mynewvar))

renamefun("d:dataelemapi2.sav", "grad_sch", "enroll",
"_new")
end program.
```

Much of the code above looks similar to that used in the previous example.
We use the spssaux.OpenDataFile() function to open the data file with the
variables that we want to rename. As before, we use the spssaux.VariableDict() function to obtain the names of the variables in the
data file, then we select only the variables between the start variable and the
end variable, and put those names into a Python variable called mylist.
The loop used in this example is almost identical to the

**loop used in the
previous example. Finally, we use our new function and then end the
program.**

```
def renamefun(datapath, mystart, myend, suffix):
spssaux.OpenDataFile(datapath)
vdict=spssaux.VariableDict()
mylist=vdict.range(start=mystart, end=myend)
nvars = len(mylist)
for i in range(nvars):
myvar = mylist
mynewvar = myvar+suffix
spss.Submit(r"""
rename variables ( %s = %s) .
""" %(myvar, mynewvar))
```

Example 4: Using the Python function

**You can save the syntax above as a Python function called renamefun.py in your Python
directory. Once you have saved the file as a Python file (with the .py extension), you
can use the SPSS program below to call the function. Because renamefun**

is just a function (rather than a complete module), you need to use the from
option on the import statement.

```
begin program.
from renamefun import renamefun
import spss, spssaux
renamefun("D:dataelemapi2.sav", "grad_sch", "enroll", "_myvar")
end program.
```

For more information