

Productive Laziness with Python Programmability in SPSS

Albert-Jan Roskam, PhD.
Statistics Netherlands (CBS)

ar.roskam@cbs.nl

ASSESS – York UK - October 2015



Python: What? Why? (1)

- General purpose language (unlike R)
- Easy to learn, but not a toy language
- “Readability counts”
- Developed in 90s by Van Rossum (Amsterdam)
- Monty Python’s Flying Circus
- Free, incl. lots of additional libraries
- Python is everywhere, incl. in SPSS!
- Python 2 and 3



Python: What? Why? (2)

- Python will help you become more productive
- Less code to maintain
- Less boring, repetitive work
- Do things that cannot be done using traditional syntax (or macro language)
- If you know Python (R, SQL, SPSS, ...) you increase your market value!



Some language characteristics

- Case-sensitive
- Indentation mandatory (unlike e.g. SPSS)
 - Easier to follow the program logic
- Many datatypes (str, int, float, list, tuple, etc.)
 - SPSS just has string and float
- Starts counting at zero (unlike SPSS, R)
- "Batteries included" philosophy



Python and SPSS

- External mode: SPSS in a Python program
- Internal mode: Python program in SPSS

Task – *Python module*

- Job automation/generalization - *spss*
- Access to metadata - *spssaux*
- Access to case data – *spssdata, spss*
- Access to output data – *SpssClient, spssaux*



Python and SPSS

- A very simple program:

```
BEGIN PROGRAM PYTHON.
```

```
print "Hello world"
```

```
END PROGRAM.
```

- Woah, that was useless!
- But: "print" is a useful diagnostic command
- Just like: `help()`, `type()`, `repr()`, `len()`, `dir()`
 - e.g.: `len("abc")` # returns 3



"Run only with SPSS v22"

```
BEGIN PROGRAM PYTHON.  
import spss  
#help(spss)  
version = spss.version.version  
if version.startswith("22"):  
    cmd = "TITLE 'Version %s'"  
    spss.Submit(cmd % version)  
else:  
    spss.Submit("ECHO 'sorry, need v22'")  
END PROGRAM.
```



SPSS Macro vs Python function

* Now add some things that are hard or impossible with macros.

```
BEGIN PROGRAM PYTHON.
```

```
import glob, spss
```

```
def process_weeks(filename):
```

```
    spss.Submit("GET FILE = '%s'." % filename)
```

```
    for week in range(1, 52+1):
```

```
        if week % 2 == 0: ### modulus: only process even weeks ###
```

```
            spss.Submit("TITLE '** file: %s | week: %s'." % (filename, week))
```

```
            spss.Submit("TEMPORARY.")
```

```
            spss.Submit("SELECT IF (week EQ %s)." % week)
```

```
            spss.Submit("SUMMARIZE var /CELLS=MEDIAN.")
```

```
### run 'process_weeks' function on *all* .sav files in temp ###
```

```
import glob
```

```
# for filename in glob.glob(r"c:\cbstemp\user\somefile_*.sav"):
```

```
    process_weeks(filename)
```

```
END PROGRAM.
```



Access meta data

```
BEGIN PROGRAM PYTHON.  
import spss, spssaux  
for v in spssaux.VariableDict():  
    vname = v.VariableName  
    spss.Submit("TITLE 'Results of variable \"%s\".'" % vname)  
    if v.VariableLevel == "scale":  
        spss.Submit("SUMMARIZE %s /CELLS=MEAN." % vname)  
    else:  
        spss.Submit("FREQUENCIES %s" % vname)  
END PROGRAM.
```



Some useful libraries

Library	What it does	Standard library?
os, os.path	Operating system stuff	Yes
sys	Python interpreter stuff	Yes
(date)time	Time/date, formatting, etc	Yes
glob	List files that match a pattern	Yes
re	Regular expressions	Yes
subprocess	Run external programs	Yes
csv	Read/write .csv files	Yes
cElementtree	Read/write .xls files	Yes
pyodbc	Read/write from/to databases	Yes
openpyxl	Read/write .xlsx	No
xlrd, xlwt	Read/write .xls	No
savReaderWriter	Read/write .sav, .zsav	No
numpy	Numerical Python	No
pandas	R-like DataFrames, etc.	No



Resources

Python in general

- <https://docs.python.org/2/tutorial/>
- <http://alan-g.me.uk/tutor/>
- <http://www.tutorialspoint.com/python>
- <http://www.python-excel.org>
- Many more (free) tutorials on the internet

Python and SPSS

- SPSS Programming and Data management
<http://tinyurl.com/spss-laziness> Tip: Choose the edition that matches your SPSS version!
- Python Scripting Guide for IBM SPSS Statistics:
<http://tinyurl.com/spss-laziness2>



Mailing lists and forums

Python in general

- Python tutor
<https://mail.python.org/mailman/listinfo/tutor>
- Python main (high volume!)
<https://mail.python.org/mailman/listinfo/python-list>

Python and SPSS

- IBM Predictive Analytics (formerly devWorks)
<https://developer.ibm.com/predictiveanalytics/forums/>
- SPSS-L <https://listserv.uga.edu/cgi-bin/wa?A0=SPSSX-L> (not very Python-oriented)

