Increasing computational speed by combining Stata and Python

Mathias Sinning

ANU Crawford School of Public Policy

10 February 2022

PyStata

- Stata provides two ways for Python and Stata to interact:
  - In Stata 16 or higher, Python can be used as part of a running Stata session (Python integration)
    - You can embed and execute Python code interactively or in do-files and ado-files
  - In Stata 17 or higher, Stata can be used in a standalone Python environment (Python package: **pystata**)

Application: Kendall's τ (Kendall, 1938)

- For any two pairs of ranks (x_i, y_i) and (x_j, y_j) of one variable pair $(\textit{varname}_1, \textit{varname}_2)$, $1 \leq i, j \leq n$, where n is the number of observations, define them as concordant if

$$(x_i - x_j)(y_i - y_j) > 0$$

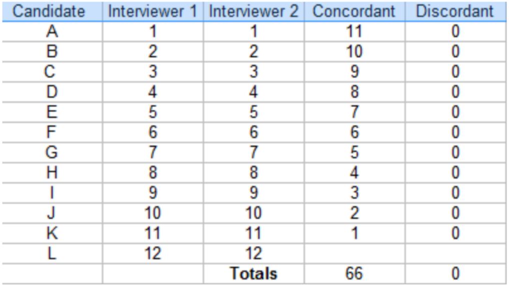
  and discordant if this product is less than zero.

- Kendall's score S is defined as $C - D$, where C (D) is the number of concordant (discordant) pairs. Let $N = n(n-1)/2$ be the total number of pairs, so τ_a is given by

$$\tau_a = S/N.$$

- Formula can be adjusted to account for ties $\rightarrow \tau_b$

Example: Perfect positive rank correlation



Candidate	Interviewer 1	Interviewer 2	Concordant	Discordant
A	1	1	11	0
B	2	2	10	0
C	3	3	9	0
D	4	4	8	0
E	5	5	7	0
F	6	6	6	0
G	7	7	5	0
H	8	8	4	0
I	9	9	3	0
J	10	10	2	0
K	11	11	1	0
L	12	12		
		Totals	66	0

Sorting algorithms

- Different sorting algorithms have different running times

  Example: Look for a word in a dictionary

- Algorithm 1
  - Start at the beginning and go through word by word $\rightarrow O(n)$
  - Look for n words in a dictionary $\rightarrow O(n^2)$

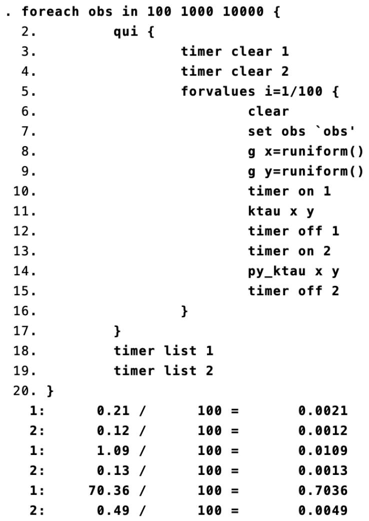
- Algorithm 2
  1. Open the dictionary in the middle and check the first word
  2. Decide whether to look in the right or the left half
  3. Divide the remainder in half again, and repeat step 2 until you find the word you are looking for
     $\rightarrow O(\log_2 n)$

  - Look for n words in a dictionary $\rightarrow O(n \log_2 n)$

Example: ktau.ado vs. py_ktau.ado

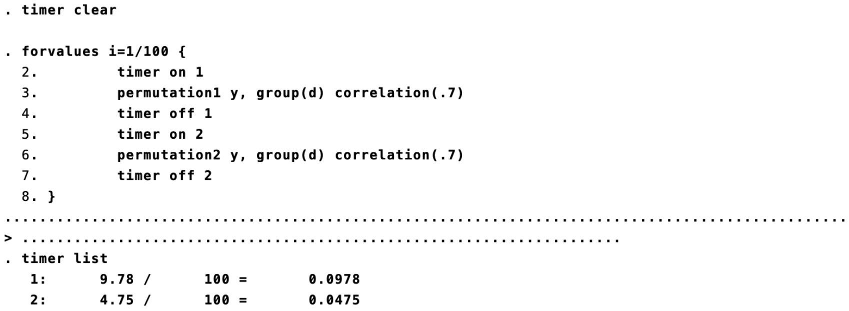
```
program py_ktau, rclass
        version 16.1
        syntax varlist(min=2 max=2) [if] [in]
        marksample touse
        local var1: word 1 of `varlist'
        local var2: word 2 of `varlist'
        python: calctau("`var1'", "`var2'", "`touse'")
        display as txt " Kendall's tau: " as res r(tau)
        local tau=r(tau)
        return scalar tau_b=`tau'
end

version 16.1
python:
from sfi import Data, Scalar
from scipy.stats import kendalltau
def calctau(var1, var2, touse):
        x = Data.get(var1, None, touse)
        y = Data.get(var2, None, touse)
        Scalar.setValue("r(tau)", kendalltau(x,y)[0])
end
```

Example: `ktau.ado` vs. `py_ktau.ado`



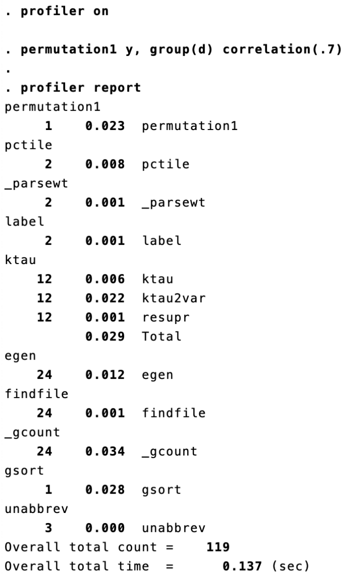
```
. foreach obs in 100 1000 10000 {
  2.         qui {
  3.                 timer clear 1
  4.                 timer clear 2
  5.                 forvalues i=1/100 {
  6.                         clear
  7.                         set obs `obs'
  8.                         g x=runiform()
  9.                         g y=runiform()
 10.                         timer on 1
 11.                         ktau x y
 12.                         timer off 1
 13.                         timer on 2
 14.                         py_ktau x y
 15.                         timer off 2
 16.                 }
 17.         }
 18.         timer list 1
 19.         timer list 2
 20. }
   1:      0.21 /      100 =       0.0021
   2:      0.12 /      100 =       0.0012
   1:      1.09 /      100 =       0.0109
   2:      0.13 /      100 =       0.0013
   1:     70.36 /      100 =       0.7036
   2:      0.49 /      100 =       0.0049
```

Example: ktau.ado vs. py_ktau.ado



```
. timer clear

. forvalues i=1/100 {
  2.         timer on 1
  3.         permutation1 y, group(d) correlation(.7)
  4.         timer off 1
  5.         timer on 2
  6.         permutation2 y, group(d) correlation(.7)
  7.         timer off 2
  8. }
...................................................................................................
> ......................................................................
. timer list
   1:      9.78 /      100 =       0.0978
   2:      4.75 /      100 =       0.0475
```

profiler



```
. profiler on

. permutation1 y, group(d) correlation(.7)
.
. profiler report
permutation1
     1    0.023  permutation1
pctile
     2    0.008  pctile
_parsewt
     2    0.001  _parsewt
label
     2    0.001  label
ktau
    12    0.006  ktau
    12    0.022  ktau2var
    12    0.001  resupr
          0.029  Total
egen
    24    0.012  egen
findfile
    24    0.001  findfile
_gcount
    24    0.034  _gcount
gsort
     1    0.028  gsort
unabbrev
     3    0.000  unabbrev
Overall total count =    119
Overall total time  =      0.137 (sec)
```

profiler

```
. profiler clear

. profiler on

. permutation2 y, group(d) correlation(.7)
.
. profiler report
permutation2
     1    0.028  permutation2
pctile
     2    0.009  pctile
_parsewt
     2    0.001  _parsewt
label
     2    0.001  label
py_ktau
    24    0.014  py_ktau
gsort
     1    0.027  gsort
unabbrev
     3    0.000  unabbrev
Overall total count =     35
Overall total time  =      0.080 (sec)
```

Setting a seed

- Many commands require a seed to ensure replicability
- It is easy to control seeds in Stata and to pass them on to Python
- `numpy.random.seed(seed)` can be used in Python to set a seed