Effective Programming Practices for Economists

Basic Python

Tracebacks

Janoś Gabler and Hans-Martin von Gaudecker

Contents

- What are Exceptions and Tracebacks?
- The Anatomy of a Python Traceback
- Reading tracebacks
- Asking for help

Motivation

- We sometimes told you that you cannot do certain things:
  - Example: Can only use hashable objects as dictionary keys
- Now we will discuss what happens if you do it anyways
  - Exception: What class of error?
  - Traceback: Detailed report that helps you to localize the error
- Pro tip: Read the traceback!

Example Traceback

```
>>> d = {"a" : 1}
>>> d[[1, 2, 3]] = "b"
>>> d["c"] = 3


-----------------------------------------------------------
TypeError 1                    Traceback (most recent call last)
/home/janos/some_file.py line 2
  1 d = {"a" : 1}
2
----> 2 d[[1, 2, 3]] = "b"
  3 d["c"] = 3

TypeError: unhashable type: 'list' 3
```

- The code on the left has a problem
- Traceback tells us everything we need:
  1. What type of Exception occurred:
     `TypeError`
  2. Where did it occur: In line 2 of
     `some_file.py`
  3. What happened exactly *(used an unhashable type where we must not)*
- Tracebacks can get very long! Read from bottom to top.
- Always look for these three things!

Common sources of errors

- `ValueError` : Called a function with something invalid

- `KeyError` : Typo in a variable name or a dictionary key

- `TypeError` : Called a function with something that has the wrong type

- `ImportError` : Typo in an import

How not to ask for help

- "I wanted to do the exercise but it does not work"
- "Python does not work on my computer"
- "My code does not work, here is a screenshot"
- Asking via DM on Zulip instead of the course stream

What to keep in mind

- We do not remember what task 3 in exercise 5 is
- We like to see that you tried on your own
- We like to see that you tried to reduce the amount we have to read
- We love well formatted, self-contained examples

A better way (for a hypothetical task)

In the task where we should use Python to calculate the output value of a Cobb-Douglas production function (assignment 1, exercise 2) the following line:

```
cobb_douglas(labor, capital, alpha)
```

gives me a type error:

```
TypeError: unsupported operand type(s) for ** or pow(): 'tuple' and 'float'
```

I don't understand the error because I'm just passing in numbers.

A better way (continued)

Here is a minimal example to reproduce the error:

```python
labor = 2.5,
capital = 4.5
alpha = 0.33

def cobb_douglas(labor, capital, alpha):
    return labor**alpha * capital**(1 - alpha)


cobb_douglas(labor, capital, alpha)
```

I attach the entire traceback as `txt` file ...