Applied Data Analytics

Getting help / Python basics

Python tracebacks

Hans-Martin von Gaudecker and Aapo Stenhammar

Contents

- What are Exceptions and Tracebacks?

- The Anatomy of a Python Traceback

- Reading tracebacks

Motivation

- We pretty much told you what you need to do so far

    - Example: `pd.Series([1, 2, 3], index=["a", "b", "c"])`

- Now we will discuss what happens if you do not follow instructions precisely

    - Exception: What class of error?

    - Traceback: Detailed report that helps you to localize the error

- Pro tip: Read the traceback!

Example Traceback

```
pd.Series(
    [1, 2, 3],
    labels=["a", "b", "c"]
)
```

Create a series with labels

```
1   pd.Series([1, 2, 3], labels=["a", "b", "c"])          Evaluated ●

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 pd.Series([1, 2, 3], labels=["a", "b", "c"])

TypeError: Series.__init__() got an unexpected keyword argument 'labels'
```

- 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 1 of our cell.
  3. What happened exactly *(tried to use a non-existent keyword argument)*
- 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 an index label or a dictionary key

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

- **`ImportError`** : Typo in an import