UCLA College | Social Sciences Economics

Day 5: EDA/ File Operations, HW1, Exception and Error Handling

Paul Schumacher, MSc Quantitative Economics

Announcements

Attendance

Midterm Feedback

https://docs.google.com/forms/d/e/1FAIpQLSc_ICA_EEFPMCyHVCyRyN11QpToozAONpM4fMM3MW4J3PBJjg/viewform?usp=sf_link

HW1 due Tomorrow at 11pm

HW1

Basic Task/ Classes:

- Create a class object called "TimeValue" which computes the PV, FV, Annuity or Perpetuity of an investment.
- Your code should prompt the user to enter their desired investment computation (one of the 4 mentioned above),
- and then to enter the required values for their choice.

Advanced Task/ Exception and Error Handling:

- Make sure you consider issues that may come up,
  - such as, e.g., if the user enters the interest rate in years, but the compounding periods (in your formula) are expected to be in months, the rates are in percent rather decimals, and so on.
  - You will need to decide what actions to take in such cases (e.g., in your code you can convert the rates to the same units).

Presentation/ Submission:

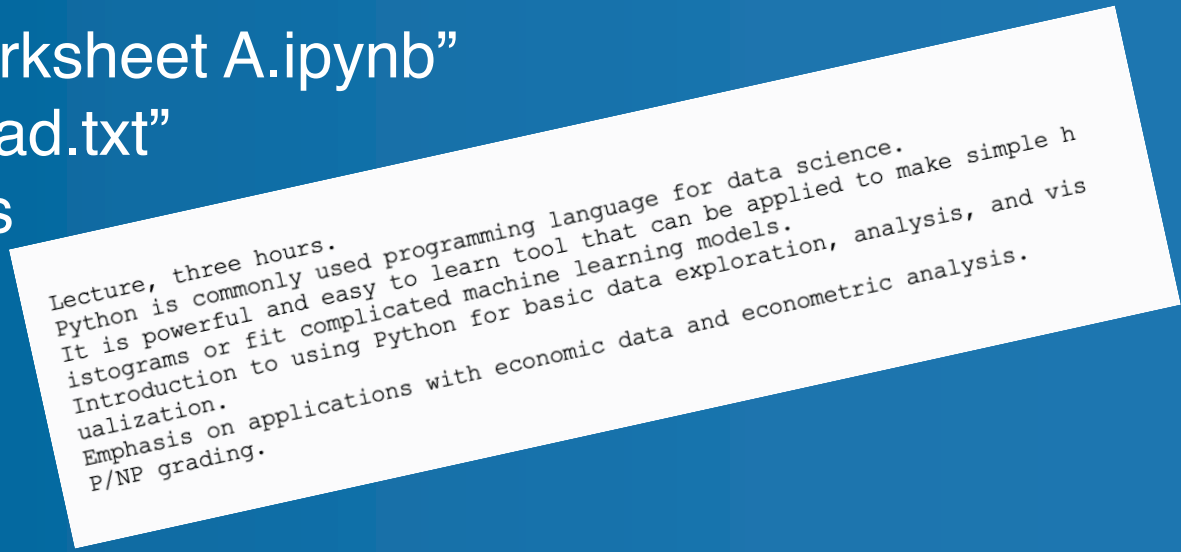
- For the ???*presentation*???, you will use your notebook to showcase your work. Deadline July 26, 11PM (PST).
- You will need to upload your notebook through the Canvas link
- You *can* work in groups. Only one upload per group, but each group member's name must be listed
- explain the decisions you made when **handling any issues related to user inputs** and demonstrate how it works
- Your presentation should be short, ~5 minutes

Quick Intro – 3 Students

1. Name, School, Hobbies, …
2. What is your favorite social media platform and why?
3. Team Barbie or Team Oppenheimer or both? Why?
4. What do you think is nowadays the biggest “threat” to humanity if there is one?
5. What is your favorite food?
6. What do you wish future self?

Class exercise: Exploratory Data Analysis (EDA), File Operation

- “PLF Day 5 Worksheet A.ipynb”
- Use text file ”read.txt”
- Breakout rooms
- Time: 15 min



Exception and Error Handling

- Useful for code section where we might anticipate and error…

```
f = open("raed.txt")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[22], line 1
----> 1 f = open("raed.txt")

File ~/anaconda3/lib/python3.10/site-packages/IPython/core/interactiveshell.py:284, in _modified_open(file, *args, **kwargs)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are doing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'raed.txt'
```

```
try:
    f = open("raed.txt")
except Exception: #catches every error/ exception causes by the input
    print("Sorry. This file does not exist!")
#else:
 #   pass
#finally:
 #   pass
```

```
Sorry. This file does not exist!
```

raed.txt ≠ read.txt

Exception and Error Handling

- Useful for code section where we might anticipate and error…

```
try:
    f = open("read.txt")
    a = b #another error
except Exception: #catches every error/ exception causes by the input
    print("Sorry. This file does not exist!")
#else:
 #   pass
#finally:
 #   pass
```

```
Sorry. This file does not exist!
```

Same error message as before even though the error is caused by something else…

```
a = b
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[33], line 1
----> 1 a = b

NameError: name 'b' is not defined
```

Taking care of only first error:

```
try:
    f = open("read.txt")
    a = b #another error
except FileNotFoundError: #catches a specific error
    print("Sorry. This file does not exist!")
#else:
 #   pass
#finally:
 #   pass
```

```
---------------------------------------------------------------------------
--
NameError                                 Traceback (most recent call las
t)
Cell In[34], line 3
      1 try:
      2     f = open("read.txt")
----> 3     a = b #another error
      4 except FileNotFoundError: #catches a specific error
      5     print("Sorry. This file does not exist!")

NameError: name 'b' is not defined
```

Exception and Error Handling

- Useful for code section where we might anticipate and error…

```
try:
    f = open("read.txt")
    a = b #another error
except FileNotFoundError: #catches a specific error
    print("Sorry. This file does not exist!")
except Exception: #for the remaining errors
    print("Sorry. Something went wrong.")
#else:
 #   pass
#finally:
 #   pass
```

```
Sorry. Something went wrong.
```

Add a second Exception
Order matters: Specific *before* general

```
try:
    f = open("read.txt")
    a = b #another error
except FileNotFoundError as e: #catches a specific error
    print(e)
except Exception as e: #for the remaining errors
    print(e)
#else:
 #   pass
#finally:
 #   pass
```

```
name 'b' is not defined
```

Display Error message only

Exception and Error Handling

- Useful for code section where we might anticipate and error…

else

```
try:
    f = open("read.txt")
except FileNotFoundError as e: #catches a specific error
    print(e)
except Exception as e: #for the remaining errors
    print(e)
else: #since we do NOT have an exception, we can run the code
    print(f.read())
    f.close()
#finally:
 #   pass
```

```
Lecture, three hours.
Python is commonly used programming language for data science.
It is powerful and easy to learn tool that can be applied to make simple
histograms or fit complicated machine learning models.
Introduction to using Python for basic data exploration, analysis, and vi
sualization.
Emphasis on applications with economic data and econometric analysis.
P/NP grading.
```

Runs if no errors

finally

```
try:
    f = open("re4734984ad.txt")
except FileNotFoundError as e: #catches a specific error
    print(e)
except Exception as e: #for the remaining errors
    print(e)
else: #since we do NOT have an exception, we can run the code
    print(f.read())
    f.close()
finally: #runs no matter what happens
    print('Executing Finally...')
```

```
[Errno 2] No such file or directory: 're4734984ad.txt'
Executing Finally...
```

Runs if even with errors

Exception and Error Handling

- Useful for code section where we might anticipate an error…

```
try:
    f = open("read.txt")
    if f.name == "read.txt":
        raise Exception
except FileNotFoundError as e: #catches a specific error
    print(e)
except Exception as e: #for the remaining errors
    print("Error")
else: #since we do NOT have an exception, we can run the code
    print(f.read())
    f.close()
finally: #runs no matter what happens
    print('Executing Finally...')
```

```
Error
Executing Finally...
```

Manually cause error

https://www.youtube.com/watch?v=NIWwJbo-9_8&ab_channel=CoreySchafer

Class exercise: Exception and Error Handling

- “PLF Day 5 Worksheet B.ipynb”
- Breakout rooms
- Time: 20 min

Any Questions?