

# PLF Session 5 EDA/ File Operations

## UCLA, Econ 10P: Introduction to Python for Economists

1. Install the Python packages **wooldridge** which contains 111 data sets often used in econometrics and **NumPy** which is a library supporting mathematical functions for large, multi-dimensional arrays and matrices.
2. Open the data set 'wage1' from wooldridge!
3. Display the...
  - number of rows and columns
  - first 5 elements/lines of the data set
  - last 5 elements/lines of the data set
  - a general overview of the some statistical measures
  - all the columns in the data set

## File Operations

### Reading a .txt File

4. Open the .txt file "read.txt". Read each sentence in a single line.

### Writing a new .txt File

5. Create and open the .txt file "text.txt". Write "Hello World!" and close the file. Check you work through reading the newly created file.

# PLF Session 5 - Exception and Error Handling

UCLA, Econ 10P: Introduction to Python for Economists

```
In [22]: f = open("raed.txt")
```

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

File ~/anaconda3/lib/python3.10/site-packages/IPython/core/interactiveshe
ll.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 a
re 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'
```

```
In [59]: try:
        f = open("raed.txt")
    except Exception: #for the remaining errors
        print("Sorry. This file does not exist!.")
    #else:
        # pass
    #finally:
        # pass
```

Sorry. This file does not exist!.

```
In [35]: 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.

```
In [38]: 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

```
In [39]: 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 visualization.

Emphasis on applications with economic data and econometric analysis.

P/NP grading.

```
In [42]: 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: #runs no matter what happens
        print('Executing Finally...')
```

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 visualization.

Emphasis on applications with economic data and econometric analysis.

P/NP grading.

Executing Finally...

```
In [44]: 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...

1. Create a **basic** function that first multiplies the input by 3 and then divides this result by 2. Once you are done input "1" as an integer and then "one" as a string. The goal is to get an output in the first case and an error in the second one.
2. Include an if statement in the function to specify the Error and raise an exception for "TypeError". If a TypeError occurs display the output: "This function is designed to work only with floats and ints. "

3. Create a try and except statement for the same function but show the error message only. Use "six" as an input.
4. Integrate the try and except statement in your function. Run the function with "1" and "one"