

• 2025-03-21 UM Scientific Computing - Python

Instructors:

Tim Norris - tnorris@miami.edu

Erica Newcome - exn297@miami.edu

Rimpi Khurana - rxk519@miami.edu

Schedule and Curriculum

Thursday March 21st

- AM: the Unix Shell: <https://swcarpentry.github.io/shell-novice/>

- PM: Introduction to Python: <https://swcarpentry.github.io/python-novice-inflammation/>

Friday March 22nd

- AM: Version Control with Git: <https://swcarpentry.github.io/git-novice/>

- PM: More Advanced Python: <https://swcarpentry.github.io/python-novice-inflammation/>

Tim's Repo of the workshop:

https://github.com/tibbbsen/analyze_patients

Day 1: Sign-in

- Jenna Courtade, University of Miami Libraries, Research Data & Open Scholarship (United States)
- Silvia Fuentes - University of Miami Alumni (Business Analytics '19) - from Spain
- Jamie Cooper - University of Miami - Miami Project - From Liverpool, England
- Clay Tobin - University of Miami (Civic & Community Engagement) - from Puerto Rico
- Eddie Gomez - egg25@miami.edu (College of Interactive Media - Ecuador, Cuba)
- Victoria Munoz - vcm47@miami.edu (Chemistry department) - Parents: USA and Cuba
- Mascha Korsch - UM, Department of Informatics and Health Data Sciences - Germany
- Shannon Shaughnessy, University of Miami Department of Psychology - from Flint, Michigan, USA
- Elena Mishkovsky - exm5149@med.miami.edu, University of Miami Dept of Neurology
- Velarie Ansu - University of Miami, Dept of Psychiatry. Ghana
- Ana Bolaños - University of Miami, Department of Neurology, axb2916@miami.edu
- Maddyel Sanchez:09:21 University of Miami, Department of Pathology, mxs2135@miami.edu, Cuba
- Yasmani Rodriguez, Hussman Institut for Human Genomics, Cuba, rxxy207@miami.edu
- Paul Sarzosa, Data Analyst, pxs708@miami.edu
- Anthony (Angel) Morataya, University of Miami, Department of Public Health,

- Ngoc Huong Giang TRAN, University of Miami, Department of Medicine, nxt538@miami.edu
- Brittany Larsen, University of Miami, Department of Psychiatry and Behavioral Sciences, from South Dakota, bal184@miami.edu
- Sunjoo Cho, UM Department of Neurology, s.cho6@miami.edu
- Aidan Wells, UM Department of Neurology, axw1195@miami.edu
- Claudia Blandino, UM Miami Project, cxb917@miami.edu
- Anastasiya Drandarov, University of Miami Department of Orthopaedics, Russia, axd1361@miami.edu
-

Bash Notes

`ls` # list directory contents
`pwd` # print working directory
`man` # manual page (or `<command> --help`)
`clear` # clear the shell screen
`cd` # change directory
`nano` # text editor
`mkdir` # make directory
`mv` # move or rename
`cp` # copy
`wc` # word count
`cat` # list contents of file (really concatenate, but with one file it simply prints contents of file)

redirection and pipes:
`>` # redirect output to a file

directory notes:
`ls .` # list contents of self
`ls ..` # list contents of parent

Use Case Thoughts

- Using `wc` or checking the "head" of a file can help you preview large datasets.

Python Notes

Day 2: Sign-in

- Anthony (Angel) Morataya, University of Miami, Department of Public Health

- Velarie Ansu, UM
- Clay Tobin - University of Miami (Civic & Community Engagement) - from Puerto Rico
- Brittany Larsen, University of Miami, Department of Psychiatry and Behavioral Sciences, from South Dakota, bal184@miami.edu
- Claudia Blandino, UM Miami Project, cxb917@miami.edu
- Silvia Fuentes - University of Miami Alumni (Business Analytics '19) - from Spain
- Maddyel Sanchez:09:21 University of Miami, Department of Pathology, mxs2135@miami.edu, Cuba
- Mascha Korsch - UM, Department of Informatics and Health Data Sciences - Germany
- Jenna Courtade, University of Miami Libraries, Research Data & Open Scholarship (United States)
- Aidan Wells, UM Department of Neurology, axw1195@miami.edu

Python Code with Functions

```
import numpy as np
import matplotlib.pyplot as plt
import glob
```

```
input_path, output_path, extension = 'data/', 'output/', 'png'
```

```
files = glob.glob(input_path+'inflammation-*.csv')
files.sort()
```

```
# run simple checks
```

```
def detect_problems(patient_data):
    """Perform basic tests on one file for patient data"""
    if np.amax(patient_data, axis=0)[0] == 0 and np.amax(patient_data, axis=0)[20] == 20:
        print('suspicious maxima')
    elif np.sum(np.amin(patient_data, axis=0)) == 0:
        print('suspicious minima')
    else:
        print('passed')
```

```
# plot mean, max, and min
```

```
def plot_graphs(file_name, patient_data):
    """Plot line graph of patient inflammation data for one file"""
    # set up figure
    fig = plt.figure(figsize=(10.0, 3.0))

    # set up plots
    axes1 = fig.add_subplot(1, 3, 1)
    axes2 = fig.add_subplot(1, 3, 2)
    axes3 = fig.add_subplot(1, 3, 3)
```

```
# plot the mean
```

```
axes1.set_ylabel('average')
```

```
axes1.plot(np.mean(patient_data, axis=0))
```

```
# plot the max
```

```
axes2.set_ylabel('max')
```

```
axes2.plot(np.amax(patient_data, axis=0))
```

```
# plot the min
```

```
axes3.set_ylabel('min')
```

```
axes3.plot(np.amin(patient_data, axis=0))
```

```
fig.tight_layout()
```

```
plt.savefig(output_path + file_name[5:-3] + extension)
```

```
plt.show()
```

```
# loop through files, check for error and plot graphs
```

```
for file in files[0:3]:
```

```
# get the data
```

```
print(file)
```

```
data = np.loadtxt(fname=file, delimiter=',')
```

```
# detect problems and create graph
```

```
detect_problems(data)
```

```
plot_graphs(file,data)
```