



Course Syllabus

FEF210 – DATA ANALYSIS WITH PYTHON

Instructor: Dr. Mustafa Murat ARAT

E-mail: muratarat@hacettepe.edu.tr

Office: Room Number 12, Department of Statistics, Hacettepe University, Beytepe Campus

Office Hours: Wednesdays / 08:40-10:30

Course Days and Hours: Wednesdays / 10:40 – 12: 30

Course Location: LAB1 at Department of Statistics, Hacettepe University, Beytepe Campus

Course Credits: 5 AKTS

Course Webpage: https://mmuratarat.github.io/turkish/fef210_2425fall/

Softwares: Python / Jupyter Environment / PyCharm / Visual Studio Code / Google Colab (<https://colab.research.google.com>), Amazon Sagemaker Studio Lab (<https://studiolab.sagemaker.aws>)

Course Objective: Teaching the fundamental concepts of programming at an advanced level using the Python programming language and its libraries.

Course Resources:

1. Michael H. Goldwasser, Michael T. Goodrich, and Roberto Tamassia (2013). Data Structures and Algorithms in Python. Wiley. ISBN: 978-1-118-29027-9
2. Eric Matthes. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming (2nd. ed.). No Starch Press. ISBN: 978-1593279288
3. Al Sweigart. (2019). Automate the Boring Stuff with Python: Practical Programming for Total Beginners (2nd. ed.). No Starch Press. ISBN: 978-1593279929
4. Python Official Documentation - <https://docs.python.org/3/>
5. StackOverFlow - <https://stackoverflow.com>
6. Google Python Style Code - <https://google.github.io/styleguide/pyguide.html>

Learning Outcomes:

1. Can produce solutions to real business problems by writing computer programs and scripts from scratch.
2. Can perform all kinds of operations on Python quickly and efficiently by using the appropriate data structure and algorithm.
3. Can perform all kinds of vector and mathematical operations using the NumPy library.
4. Using the Pandas library can read different data formats into the Python environment and perform all kinds of data transformation operations.
5. Can perform analysis of simple textual data.
6. Can work effectively in research teams with interdisciplinary interaction
7. Can define classes using Python programming language

Accommodations: Any student requiring special accommodations for any reason should contact the instructor as soon as possible.

Attendance Policy: All students attend a minimum of 70% of class hours during their course. If you come after the instructor (when the door is closed) or leave any class for more than 5 minutes you will be marked



absent for that class. All absences from class are counted as official absences. If you have a special excuse, please write to instructor not too late.

Class Policy: You must be in class before the lecture starts. You are expected to do your best to be in class on time. You are not allowed to enter the classroom after the lecture starts. You must turn off all your mobile phones. During the lecture, you must avoid all activities that are better performed elsewhere. Besides, do not hesitate to ask any questions related to the lecture during the courses!

General Comments: (1) If you are having problems with the course, come and discuss the situation with me as soon as possible. (2) If you are having problems with **ME**, come and discuss the situation with me as soon as possible. (3) If the pace of the lectures is too fast or slow, let me know. I am not always aware of it, no matter how obvious it may be to you. (4) If you do not understand a line of code, first run it, then create a small example and see what it does! (5) If you have a question, please first use Google and search and try to understand by yourself. If it does not help, just let me know! (6) English is important while programming. Try to learn the language!

Cheating & Plagiarism: You are responsible for knowing the University policies on cheating and plagiarism. Not giving credit to a person for their intellectual work and passing it off as your own is stealing. If you have any questions or concerns about whether your behavior could be interpreted as plagiarism, please ask me before you submit the work.

Course Outline (tentative):

WEEK 1 (SEPTEMBER 25th, 2024)

General Introduction

WEEK 2 (OCTOBER 2nd, 2024)

1. Python Basics
 - 1.1. Why Python?
 - 1.2. Installing and Installing Python
 - 1.3. Python 2 and Python 3
 - 1.4. Which Python do I use?
 - 1.5. JupyterLab, Anaconda, Pycharm and Visual Studio Code Installation
 - 1.6. Installing or Updating Python Packages
 - 1.7. Python Communities and Finding Answers to Your Problems: Stackoverflow.com and Stackexchange.com
 - 1.8. Introduction to Basic Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn and statsmodels
 - 1.9. Zen of Python
 - 1.10. PEP8 – Python Styling Guide
 - 1.11. Indentations
 - 1.12. Commenting
 - 1.13. print function
 - 1.14. f-strings

WEEK 3 (OCTOBER 9th, 2024)

2. Primitive Data Types



- 2.1. Numeric Data Type
- 2.2. String Data Type
- 2.3. Boolean Data Type
- 2.4. Type Conversion
- 2.5. None Data Type

WEEK 4 (OCTOBER 16th, 2024)

I am out of town for business.

WEEK 5 (OCTOBER 23rd, 2024)

- 3. Functions
 - 3.1. docstring
 - 3.2. Return Statement
 - 3.3. Function Arguments
 - 3.3.1. Required arguments
 - 3.3.2. Positional arguments
 - 3.3.3. Keyword arguments
 - 3.3.4. Default arguments
 - 3.4. Anonymous Functions (Lambda function)
 - 3.5. args and kwargs arguments

WEEK 6 (OCTOBER 30th, 2024)

- 4. Continuing Functions
 - 4.1. Namespaces
 - 4.2. Scope of Variables
 - 4.2.1. Local Scope
 - 4.2.2. Global Scope
 - 4.2.3. Inclusive Scope
 - 4.2.4. Built-in Scope
 - 4.2.5. LEGB Rule

WEEK 7 (NOVEMBER 6th, 2024)

- 5. Control Flow
 - 5.1. if, elif and else structure
 - 5.1.1. simple if
 - 5.1.2. if-else
 - 5.1.3. nested if
 - 5.1.4. if-elif-else
 - 5.1.5. ternary operator
- 6. Loops
 - 6.1. for loops
 - 6.2.1. break statement
 - 6.2.2. continue statement
 - 6.2.3. pass statement
 - 6.2. while loops



WEEK 8 (NOVEMBER 13rd, 2024)

7. Primitive and Non-Primitive Data Structures

7.1. Non-Primitive Data Structures

7.1.1. Tuples and Tuple Methods

7.1.2. Lists and List Methods

7.1.3. Built-in Sequence Functions (enumerate, sorted, zip, reversed, filter, map)

7.1.4. Dictionaries and Dictionary Methods

7.1.5. Sets and Set Methods

7.2. List, Set, Dict Comprehensions

WEEK 9 (NOVEMBER 20th, 2024)

Midterm

WEEK 10 (NOVEMBER 27th, 2024)

9. Object Oriented Programming (OOP)

9.1. Procedural and Modular Programming

9.2. Create a class

9.3. Building an object

9.4. Attributes/Methods

9.5. Class Attributes / Instance Attributes

9.6. Accessing Attributes: getattr(), setattr(), delattr()

9.7. Class Methods / Instance Methods / Static Methods

WEEK 11 (DECEMBER 4th, 2024)

10. Continuing Object-Oriented Programming

10.1. Encapsulation

10.2. Polymorphism

10.3. Public Variables / Private Variables / Protected Variables

10.4. Dunder Methods (Special Methods)

10.5. Method Resolution Order (MRO)

10.6. Inheritance

10.7. super() function

WEEK 12 (DECEMBER 11st, 2024)

11. Errors and Exceptions in Python

12. Virtual Environments

13. Basic String Manipulations

WEEK 13 (DECEMBER 18th, 2024)

14. A Detailed Look at the NumPy Library

WEEK 14 (DECEMBER 25th, 2024)

15. A Detailed Look at the pandas Library