

CSC 5741 (2020/21)

Data Mining and Warehousing

Lecture 2: Python for Data Mining and Machine Learning

Lighton Phiri

Department of Library & Information Science
University of Zambia

<http://lis.unza.zm/~lightonphiri>

Updates on Activities—March 29, 2021 (1/2)

- Trial paper reading discussion moved to next week
 - Review paper and aligned with grading rubric
- Trial Talk moved to next week
- Invited Speakers to be confirmed

Automatic Classification of Digital Objects for Improved Metadata Quality of Electronic Theses and Dissertations in Institutional Repositories

Lighton Phiri

Department of Library and Information Science,
University of Zambia,
Lusaka, Zambia
E-mail: lighton.phiri@unza.zm

Abstract: Higher Education Institutions typically employ Institutional Repositories (IRs) in order to curate and make available Electronic Theses and Dissertations (ETDs). While most of these IRs are implemented with self-archiving functionalities, self-archiving practices are still a challenge. This arguably leads to inconsistencies in the tagging of digital objects with descriptive metadata, potentially compromising searching and browsing of scholarly research output in IRs. This paper proposes an approach to automatically classify ETDs in IRs, using supervised machine learning techniques, by extracting features from the minimum possible input expected from document authors: the ETD manuscript. The experimental results demonstrate the feasibility of automatically classifying IR ETDs. Additionally, ensuring that repository digital objects are appropriately structured, Automatic classification of research objects has the obvious benefit of improving the searching and browsing of content in IRs and further presents opportunities for the implementation of third-party tools and extensions that could potentially result in effective self-archiving strategies.

March 29, 2021

CSC 5741 (2020/21) L03 - 2

Updates on Activities—March 29, 2021 (2/2)

This paper focuses on outlining the gap between men and women in the field of science, technology, engineering, and mathematics (STEM). Many studies have been carried out to demonstrate the gap using surveys. However, very little of these surveys targeted a population of women either by university degree or nationality. This meant that most research was done on men. In this study, the gender gap in computer science was examined. A bibliometric approach was chosen to detect the gender gap in order to cater for as many gender researchers as possible.

The study focused on researchers that actively do research and publish their findings regardless of their degree, employment level, nationality, age, or origin. As a case study, the gender gap in the scientific field of technology and Research Centre 89 (Tecno) was examined. The gender gap in researchers who submitted manuscripts to their results in proceedings of international conferences over the last six years was used.

DATA WAS EXTRACTED FROM THE CONFERENCE PROCEEDINGS USING A DATA SCRAPER. The script extracted the authors' first and last names, and the conference name and year. The conference years were limited to 2010–2019. A total of 17,833 records were extracted. Of these, 242 were removed because authors used abbreviations for the first name.

The results were then classified using a name recognition software called NameSor Applied Ontology. To access the cultural and ethnic origin of the names, the names were classified by origin from the National Origin API. The 17,781 records were classified into 71 ethnic classes. Classification of gender was done using the NameGender API. The results showed that 90.1% of the names were classified as male and only 9.9% were classified as female. 24.9% of the names were considered to be gender neutral. The reason for this was that the script had to work with other factors led to the removal of all Asian names from the dataset. This increased the percentage of male and female authors to 87.5 and 11.3 respectively.

lighon.phiri@unza.zm: (25%) Accuracy
(25%) Coverage
(10%) Arguments
(20%) Presentation and Layout
(5%) Personal Reflection

> Paper readings are a nice way of identifying gaps: one of the ways of identifying gaps is contextualising research with existing literature. [1] How would this be adapted to our environment? Can we take advantage of the approach or modified version of it?
> No mention of the role of Socio??
> Virtually no personal reflection
> Virtually no arguments or presentation presented, save for the last statement, which is ify

lighon.phiri@unza.zm: * Research encompasses much more than publishing! Do you think the problem statement and indeed the title is indicative of 'Gaps in Computer Publishing'? If so, what gaps should have been tagged 'Gap in CS Publishing'?

lighon.phiri@unza.zm: * Whenever a study draws comparisons with existing literature, you want to draw comparisons.
> Are there any shortcomings with the proposed approach after comparing with existing literature? Which approach is more reliable and credible? In what instances would the proposed approach be desirable?

permissions on an item

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 4

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 5

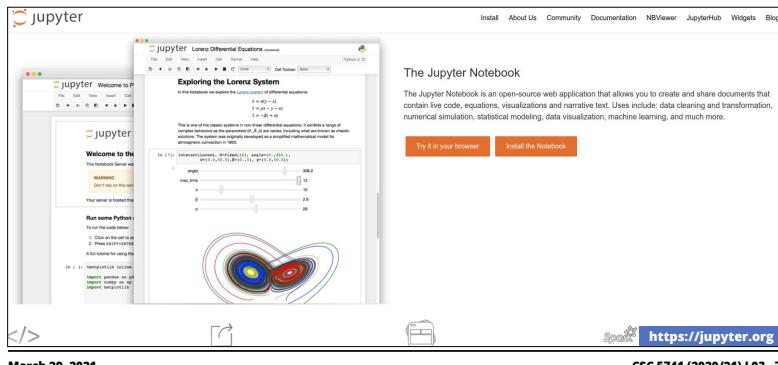
Lecture Series Outline

- Part I: Jupyter Notebooks
 - Jupyter Notebooks Interface
 - Textual Content
 - Live Code
 - Visualisations
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 6

About Jupyter Notebooks (1/2)



The screenshot shows the Jupyter Notebook interface. On the left, there's a sidebar with various links and a 'Welcome to the Jupyter Notebook' message. The main area displays a Jupyter notebook cell containing code and output related to the Lorenz system. The output includes a plot of the Lorenz attractor and some numerical results. Below the notebook, there are icons for file operations like copy, paste, and save.

March 29, 2021

CSC 5741 (2020/21) L03 - 7

About Jupyter Notebooks (2/2)



The screenshot shows a Jupyter Notebook cell. The code cell contains the following Python code:

```
1 \title{CSC 5741: Lecture #03---Python for Machine Learning}
2 \author{(Lighton Phiri)\<lighton.phiri@unza.zm>}
3 \date{March 6 2020}
4 \maketitle
```

Below the code cell is a 'Table of Contents' section with a single item: 'Introduction'.

- A notebook is a web application that contains descriptive textual content, live code, equations and visualizations.
 - While we shall predominantly use the Python kernel, additional kernels for other languages like R can be installed.

March 29, 2021

CSC 5741 (2020/21) L03 - 8

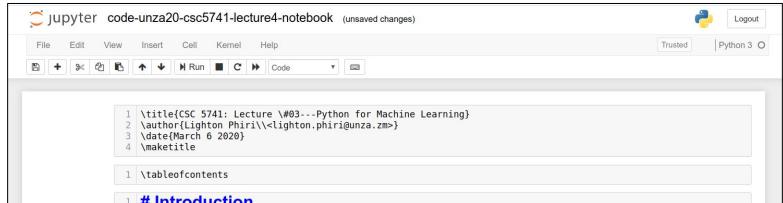
About Jupyter Notebooks: Installation

- Installation instructions are available online (<https://jupyter.org/install>)



A screenshot of a Jupyter Notebook interface. The title bar says "jupyter code-unza20-multilabel_subject_classification Last Checkpoint: 03/06/2021 (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Help. The toolbar has icons for New, Open, Save, Run, Cell, Kernel, Help, Code, and Cell Type. The code cell contains Python code for setting a title and importing libraries. The output cell shows the rendered text "# Introduction". The bottom status bar says "March 29, 2021" and "CSC 5741 (2020/21) L03 - 9".

About Jupyter Notebooks: UI



A screenshot of a Jupyter Notebook interface showing the UI. The title bar says "jupyter code-unza20-csc5741-lecture4-notebook (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Help. The toolbar has icons for New, Open, Save, Run, Cell, Kernel, Help, Code, and Cell Type. The code cell contains Python code for setting a title and importing libraries. The output cell shows the rendered text "# Introduction". The bottom status bar says "March 29, 2021" and "CSC 5741 (2020/21) L03 - 10".

- Text and live code is specified in cells and menubar and/or toolbar are used to execute cell contents
- Output appears immediately below the input cell

About Jupyter Notebooks: Text (1/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash course introduction to Python 3
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make references to online documentation for the various tools.
9 Additionally, you can exploit tools like [Zed Offline Documentation Browser](https://zeddocs.org) to
10 download and search through offline documentation. You are also encouraged to look up and explore other
11 libraries, especially as you work towards the Mini Projects.
```

- Textual content is primarily specified using the markup language "Markdown"
 - You essentially specify the structure of your text, similar to HTML

March 29, 2021

CSC 5741 (2020/21) L03 - 11

About Jupyter Notebooks: Text (2/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks---Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash course introduction to Python 3
6 3. Core Python Modules---Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make references to online documentation for the various tools.
9 Additionally, you can exploit tools like [Zed Offline Documentation Browser](https://zeddocs.org) to
10 download and search through offline documentation. You are also encouraged to look up and explore other
11 libraries, especially as you work towards the Mini Projects.
```

- Common markup: Headings
 - # h1
 - ## h2
 - ### h3
 - #### h4

March 29, 2021

CSC 5741 (2020/21) L03 - 12

About Jupyter Notebooks: Text (3/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks--Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash course introduction to Python 3
6 3. Core Python Modules--Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
Additionally, you can exploit tools like [Zeal Offline Documentation Browser](https://zealdocs.org) to
download and search through offline documentation. You are also encouraged to look up and explore other
libraries, especially as you work towards the Mini Projects.
```

- Common markup: Lists—Unordered
 - * Jupyter Notebooks
 - * Python 3
 - * Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 13

About Jupyter Notebooks: Text (4/4)

```
1 # Introduction
2
3 During these "hands-on" activities, we will explore and experiment the following:
4 1. Jupyter Notebooks--Quick walkthrough of Jupyter Notebooks
5 2. Python 3---Crash course introduction to Python 3
6 3. Core Python Modules--Quick walkthrough of some core Python modules that will be used in the course.
7
8 In all instances, you are encouraged to make reference to online documentation for the various tools.
Additionally, you can exploit tools like [Zeal Offline Documentation Browser](https://zealdocs.org) to
download and search through offline documentation. You are also encouraged to look up and explore other
libraries, especially as you work towards the Mini Projects.
```

- Common markup: Lists—Unordered
 - 1. Jupyter Notebooks
 - 2. Python 3
 - 3. Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 14

About Jupyter Notebooks: Code (1/2)

```
In [3]: 1 # 1. Draw a line plot showing the trends of the Lusaka BNB between November 2016 and April 2018
2 # We will plot BNB as a function of months
3
4 # Format input data points
5 var_bnb_months = ['Nov 16','Dec 16','Jan 17','Feb 17','Mar 17','Apr 17','May 17','June 17','July 17','Aug
6 17','Sep 17','Oct 17','Nov 17','Dec 17','Jan 18','Feb 18','Mar 18','Apr 18']
6 var_bnb_values =
[5085.14,4976.67,4935.46,4918.76,5017.89,4973.03,4952.69,4958.52,4859.35,4928.37,4883.57,4869.47,4924.54,4957.
7 47,5229.14,5385.42,5574.81,5433.04]
8 plt.style.use("ggplot") # Use the visually appealing ggplot R theme
9 plt.plot(var_bnb_months,var_bnb_values,color="red") # plot BNB months vs BNB values
```

- Python code, shell commands and magics are the most common type of code
 - Python code is specified in its raw form in the cells

March 29, 2021

CSC 5741 (2020/21) L03 - 15

About Jupyter Notebooks: Code (2/2)

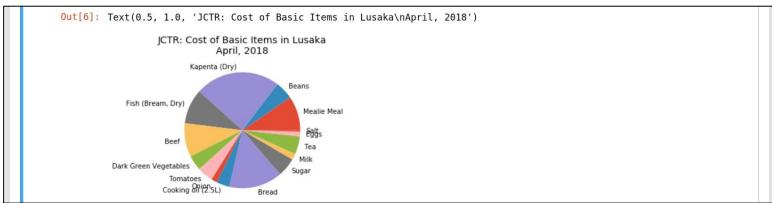
```
In [3]: 1 # 1. Draw a line plot showing the trends of the Lusaka BNB between November 2016 and April 2018
2 # We will plot BNB as a function of months
3
4 # Format input data points
5 var_bnb_months = ['Nov 16','Dec 16','Jan 17','Feb 17','Mar 17','Apr 17','May 17','June 17','July 17','Aug
6 17','Sep 17','Oct 17','Nov 17','Dec 17','Jan 18','Feb 18','Mar 18','Apr 18']
6 var_bnb_values =
[5085.14,4976.67,4935.46,4918.76,5017.89,4973.03,4952.69,4958.52,4859.35,4928.37,4883.57,4869.47,4924.54,4957.
7 47,5229.14,5385.42,5574.81,5433.04]
8 plt.style.use("ggplot") # Use the visually appealing ggplot R theme
9 plt.plot(var_bnb_months,var_bnb_values,color="red") # plot BNB months vs BNB values
```

- Python code, shell commands and magics are the most common type of code
 - Shell commands are prefixed with "!"
 - Cell magics are prefixed with "%"
 - Available magics specified with "%lsmagic"

March 29, 2021

CSC 5741 (2020/21) L03 - 16

About Jupyter Notebooks: Visualisations



- Visualisations can be generated using plotting libraries like matplotlib or using HTML via the "%html" magic

March 29, 2021

CSC 5741 (2020/21) L03 - 17

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
 - Google Colab Interface
- Part III: Getting Started With Python
- Part IV: Core Python Libraries

March 29, 2021

CSC 5741 (2020/21) L03 - 18

Google Colab Interface (1/5)

- Google Colaboratory is a cloud-based alternative to Jupyter Notebook

File Edit View Insert Runtime Tools Help Last saved at 8:09 PM

Code snippets

Filter code snippets

Adding form fields →

Camera Capture →

Cross-output communication →

display.Javascript to execute JavaScript →

Downloading files or importing data →

Adding form fields

Form example

Forms support multiple types of fields with type checking including strings, date

https://colab.research.google.com

March 29, 2021

CSC 5741 (2020/21) L03 - 19

Google Colab Interface (2/5)

- Google Colaboratory is a cloud-based alternative to Jupyter Notebook

File Edit View Insert Runtime Tools Help Last saved at 8:09 PM

Code snippets

Filter code snippets

Adding form fields →

Camera Capture →

Cross-output communication →

display.Javascript to execute JavaScript →

Downloading files or importing data →

Adding form fields

Form example

Forms support multiple types of fields with type checking including strings, date

https://colab.research.google.com

March 29, 2021

CSC 5741 (2020/21) L03 - 20

Google Colab Interface (3/5)

- Google Colaboratory is a cloud-based alternative to Jupyter Notebook

A screenshot of the Google Colab interface. The sidebar on the left contains sections like 'Code snippets' (with a red box around it), 'Adding form fields', 'Camera Capture', 'Cross-output communication', 'display Javascript to execute Java...', 'Downloading files or importing dat...', 'Adding form fields' (another red box), 'Forms example', and 'Forms support multiple types of fields with type-checking including orders, date'. The main area shows a code cell with the URL 'https://colab.research.google.com'. At the bottom, the date 'March 29, 2021' and page 'CSC 5741 (2020/21) L03 - 21' are visible.

Google Colab Interface (4/5)

- Google Colaboratory is a cloud-based alternative to Jupyter Notebook

A screenshot of the Google Colab interface, identical to the one above but with a red box highlighting the entire sidebar area. The sidebar includes 'Code snippets', 'Adding form fields', 'Camera Capture', 'Cross-output communication', 'display Javascript to execute Java...', 'Downloading files or importing dat...', 'Adding form fields', 'Forms example', and 'Forms support multiple types of fields with type-checking including orders, date'. The main area shows a code cell with the URL 'https://colab.research.google.com'. At the bottom, the date 'March 29, 2021' and page 'CSC 5741 (2020/21) L03 - 22' are visible.

Google Colab Interface (5/5)

- Google Colaboratory is a cloud-based alternative to Jupyter Notebook

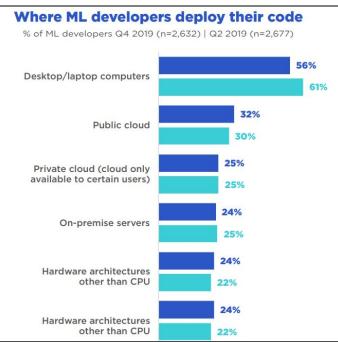
A screenshot of the Google Colab interface. The sidebar on the left contains sections like 'Code snippets' (with a red box around it), 'Adding form fields', 'Camera Capture', 'Cross-output communication', 'display Javascript to execute Java...', 'Downloading files or importing dat...', 'Adding form fields' (another red box), 'Forms example', and 'Forms support multiple types of fields with type-checking including orders, date'. The main area shows a code cell with the URL 'https://colab.research.google.com'. At the bottom, the date 'March 29, 2021' and page 'CSC 5741 (2020/21) L03 - 23' are visible.

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
 - Introduction
 - Installation and Setup
 - Basics
 - Data Structures
 - Flow Control
 - Functions and Modules
- Part IV: Core Python Libraries

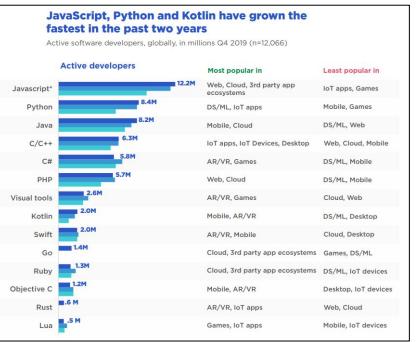
March 29, 2021 CSC 5741 (2020/21) L03 - 24

Motivation



March 29, 2021

CSC 5741 (2020/21) L03 - 25



Getting Started With Python (1/3)

```
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
```

March 29, 2021

CSC 5741 (2020/21) L03 - 26

Getting Started With Python (2/3)

- Python is an interpreted language
- Python is a scripting language
- Python is a general purpose language
- Python is an Object Oriented language
- [...]
- [...]
- We recommend using Python 3

March 29, 2021

CSC 5741 (2020/21) L03 - 27

Getting Started With Python (3/3)

- Python statements can be executed directly from the interpreter
- Python scripts can be executed as shell commands

March 29, 2021

CSC 5741 (2020/21) L03 - 28

Installation and Setup

- [...]

March 29, 2021

CSC 5741 (2020/21) L03 - 29

Installation and Setup (1/3)

- Download and install the latest version of Python 3**
 - Installers also available on course Web page, in the resources directory
- Download and install the latest version of pip**

The screenshot shows the Python.org homepage. It features a search bar and navigation links for About, Downloads, Documentation, Community, Success Stories, News, and Events. A prominent section displays a code snippet demonstrating list comprehensions and the use of the `upper()` method. Below this, there's a brief description of lists and an "Learn More" link. At the bottom, there's a large blue button with the URL <https://www.python.org/downloads>.

March 29, 2021

CSC 5741 (2020/21) L03 - 30

Installation and Setup (2/3)

- Any text editor will be sufficient for scripting.**
 - Vim, Notepad [...]
- On IDEs**
 - There are plenty of IDEs to choose from
 - In the recent past, I have worked with Wing 101 and Kate

The screenshot shows a Stack Overflow search results page for the query "What IDE to use for Python? [closed]". The results table includes columns for ID, Name, Cross Platform, Commercial Free, Auto Code Completion, Integrated Python Debugging, and Bracket Matching. The table lists several IDEs, with Wing 101 and Kate being mentioned. The URL <https://stackoverflow.com/q/81584/664424> is visible at the bottom.

March 29, 2021

CSC 5741 (2020/21) L03 - 31

Installation and Setup (3/3)

The screenshot shows the Visual Studio Code interface with the Python extension installed. The sidebar shows the Python extension icon. The main panel displays the Python extension settings, including a brief description and a "Quick start" guide with three steps: installing Python, opening a file, and starting coding. The status bar at the bottom shows the terminal command: `lightrunner@lightrunner-Lenovo-Deepad-320-15IKB ~$ cd /Projects/2020/2015-m13n30-csc5741/code/python_crash_course`. The URL <https://code.visualstudio.com> is visible at the bottom right.

March 29, 2021

CSC 5741 (2020/21) L03 - 32

Installation and Setup (3/3)

The screenshot shows the Visual Studio Code interface with the Jupyter Notebooks extension installed. The sidebar on the left has a tree view with 'OPEN FOLDERS' and 'Jupyter Notebooks' selected. The main workspace shows the 'Introduction' notebook from the 'code-uniz20-csc5741-lecture-notebook.ipynb' file. The content of the notebook is visible, including the introduction text and some code cells. At the bottom right of the notebook area, there is a link: <https://code.visualstudio.com>.

March 29, 2021

CSC 5741 (2020/21) L03 - 33

Basics

- No need to specify data types on variable declaration
- Indentation is important

March 29, 2021

CSC 5741 (2020/21) L03 - 34

Identifiers (1/2)

- Python is case-sensitive, meaning uppercase and lowercase are considered as different
 - age is different from AGE
 - favourite_course is different from Favourite_Course
- Variable names, like other identifiers, follow rules
 - can use letters, numbers or underscores
 - can't use other punctuation
 - can't start with a number
 - can't use Python keywords (reserved words)
- The assignment operator in Python is the equals sign =
 - `>>> age = 19`

March 29, 2021

CSC 5741 (2020/21) L03 - 35

Identifiers (2/2)

- Python keywords (reserved words) can't be used when naming identifiers
- `>>> import keyword`
- `>>> keyword.kwlist`
- `['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']`

March 29, 2021

CSC 5741 (2020/21) L03 - 36

Comments (1/2)

- Comments are useful in explaining your code, and are ignored by the Python interpreter
- Single line comments are simply indicated with a hash # character
- Everything to the right of the hash is ignored
 - >>> course_code = "csc5741" # creates a variable course_code

March 29, 2021

CSC 5741 (2020/21) L03 - 37

Comments (2/2)

- Multiple line comments are specified between sets of three quotes, ''' or """

```
''' Author: Mwangala Sikota  
Course: CSC 5741  
Lecture #04 '''
```

```
"" " Author: Mumbi Mumbi  
Course: CSC 5741  
Lecture #04 " ""
```

March 29, 2021

CSC 5741 (2020/21) L03 - 38

Data Types (1/3)

- Variables don't require explicit type declaration in Python, as in other programming languages
 - >>> x = 5
- There are a few basic data types in Python

Integers int
Float float
String str
Boolean bool

March 29, 2021

CSC 5741 (2020/21) L03 - 39

Data Types (2/3)

- Integer, whole numbers**
 - >>> i = 23
- Float, floating point numbers**
 - full stop indicates decimal point
 - >>> d = 2.345
- String, piece of text**
 - enclosed in single (') or double quotes (")
 - >>> x = 'CSC 5741'
 - >>> y = "CSC 5741"

March 29, 2021

CSC 5741 (2020/21) L03 - 40

Data Types (3/3)

- Boolean, true or false
 - values True and False, start with capital letter
 - 0, "", [], {}, None are considered False, everything else is True
 - `>>> weekday = True`

March 29, 2021

CSC 5741 (2020/21) L03 - 41

Functions (1/3)

- Functions are used to perform simple operations, sometimes on values
- Functions are called with round brackets ()
 - `function_name()`
- Functions can be passed certain values, which are referred to as parameters (or arguments) separated by commas
 - `function_name(parameter)`
 - `function_name(parameter1, parameter2, ...)`

March 29, 2021

CSC 5741 (2020/21) L03 - 42

Functions (2/3)

- Python has many built-in functions, here are some:
 - `print()` function prints information to the screen
 - `input()` function gets information from the user
 - `type()` function returns data type of variable or value
 - `>>> x = 3`
 - `>>> type(x)`
 - `<class 'int'>`

March 29, 2021

CSC 5741 (2020/21) L03 - 43

Functions (3/3)

```
def csc5741(x, y='Y', z='Z'):
    print(x + ' ' + y)
    return 0
csc5741('Xxxx', 'Yyyyy')
```

- All arguments are named
- Naming useful for optional arguments
- Return is optional

March 29, 2021

CSC 5741 (2020/21) L03 - 44

Data Structures

- Tuple
 - var = (1, 2, 3, 4, 5)
- List
 - var = [1, 2, 3, 4, 5]
- Dictionary
 - var = {"one":1, "two":2, "three":3, "four":4, "five":5}
- Set
 - var = {1, 2, 3, 4, 5}

March 29, 2021

CSC 5741 (2020/21) L03 - 45

Loops

```
for i in [1,2,3]:  
    print(i)
```

```
while i < 5:  
    i += 1  
    print(i)
```

- No curly braces or "end for"
- Structure is derived from level of indentation
- One statement per line
- No semicolons required

March 29, 2021

CSC 5741 (2020/21) L03 - 46

Modules (1/2)

- Modules facilitate extensibility and reusability
- Modules are collections of functions adding functionality to Python
- Modules can be imported using import keyword
 - Once modules are imported, their functions can be accessed by using the module name
 - The help() function displays what is contained in a module

```
from math import sqrt  
import math
```

March 29, 2021

CSC 5741 (2020/21) L03 - 47

Modules (2/2)

- Single functions can be imported using the from statement
 - >>> from math import sqrt
- When using the from statement functions can be accessed without the module name
 - >>> sqrt(16)
- Everything from the module can be imported using an asterisk with the from statement
 - >>> from math import *

March 29, 2021

CSC 5741 (2020/21) L03 - 48

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries
 - Datasets
 - matplotlib
 - pandas
 - Scikit-learn

March 29, 2021

CSC 5741 (2020/21) L03 - 49

Datasets

- See Jupyter Notebook “2020/21 CSC 5741: Lecture #04 Notebook—Python for Machine Learning” (<http://bit.ly/2Q2T2Lw>)

March 29, 2021

CSC 5741 (2020/21) L03 - 50

Matplotlib (1/7)

- The matplotlib library is best installed using pip, as with all libraries or using apt-get, if on Mac or Linux
 - pip3 install matplotlib
 - sudo apt-get install python-matplotlib
- Test installation by importing a library module

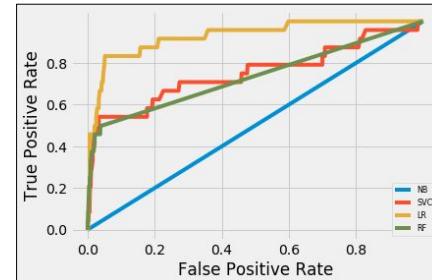
```
[lightonphir@lightonphir-Lenovo-d320-15IKB:~]$ pip3 install matplotlib
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: numpy >= 1.10.0 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: cycler >= 0.10 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: pytz >= 2018.4 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: six >= 1.5 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: kiwisolver >= 1.0.1 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: numpy >= 1.10.0 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: six >= 1.5 in /usr/local/lib/python3.6/site-packages
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from matplotlib)
[lightonphir@lightonphir-Lenovo-d320-15IKB:~]$
```

March 29, 2021

CSC 5741 (2020/21) L03 - 51

Matplotlib (2/7)

- Basic elements of a plot
 - Plot title
 - Axis labels
 - Legend



March 29, 2021

CSC 5741 (2020/21) L03 - 52

Matplotlib (3/7)

- **Creating plots is a four-step process**

- 1) Import matplotlib
- 2) Draw the plot
- 3) Specify plot aesthetics
- 4) Render plot

March 29, 2021

CSC 5741 (2020/21) L03 - 53

Matplotlib (4/7)

- **Creating plots is a four-step process**

- 1) Import matplotlib
import matplotlib.pyplot as plt
- 2) Draw the plot
- 3) Specify plot aesthetics
- 4) Render plot

March 29, 2021

CSC 5741 (2020/21) L03 - 54

Matplotlib (5/7)

- **Creating plots is a four-step process**

- 1) Import matplotlib
import matplotlib.pyplot as plt
- 2) Draw the plot
plt.plot([...])
plt.hist([...])
- 3) Specify plot aesthetics
- 4) Render plot

- **Illustration**

- Simple plots
- Plots using pandas
dataframe

March 29, 2021

CSC 5741 (2020/21) L03 - 55

Matplotlib (6/7)

- **Creating plots is a four-step process**

- 1) Import matplotlib
import matplotlib.pyplot as plt
- 2) Draw the plot
plt.plot([...])
plt.hist([...])
- 3) Specify plot aesthetics
plt.xlabel("...")
plt.ylabel("...")
- 4) Render plot

- **Illustration**

- Simple plots
- Plots using pandas
dataframe

March 29, 2021

CSC 5741 (2020/21) L03 - 56

Matplotlib (7/7)

- Creating plots is a four-step process
 - 1) Import matplotlib
 `import matplotlib.pyplot as plt`
 - 2) Draw the plot
 `plt.plot([...])`
 `plt.hist([...])`
 - 3) Specify plot aesthetics
 `plt.xlabel("[..."); plt.ylabel("[...")`
 `plt.legend()`
 - 4) Render plot
 `plt.show()`
- Illustration
 - Simple plots
 - Plots using pandas dataframe

March 29, 2021

CSC 5741 (2020/21) L03 - 57

Matplotlib—Exercises

- See Jupyter Notebook “2020/21 CSC 5741: Lecture #04 Notebook—Python for Machine Learning” (<http://bit.ly/2Q2T2Lw>)

March 29, 2021

CSC 5741 (2020/21) L03 - 58

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries
 - Datasets
 - matplotlib
 - pandas
 - Scikit-learn

March 29, 2021

CSC 5741 (2020/21) L03 - 59

Pandas (1/9)

- Why use pandas instead a spreadsheet for data analysis
 - Efficiency as data scales
 - Very user-friendly
 - Dataframe similar to spreadsheet

March 29, 2021

CSC 5741 (2020/21) L03 - 60

Pandas (2/9)

- Why use pandas instead a spreadsheet for data analysis
 - Efficiency as data scales
 - Very user-friendly
 - Dataframe similar to spreadsheet

March 29, 2021

CSC 5741 (2020/21) L03 - 61

Pandas (3/9)

- Pandas DataFrame
 - Two dimensional labeled data structure
 - DataFrame can be viewed as a representation of a Spreadsheet worksheet

	StudentID	Gender	Minor	LastName	... PassedTest3	
0	2017013156@student.unza.zm	M	Geography	Anayawa	...	NO
1	2017012891@student.unza.zm	M	Civic	Banda	...	NO
2	2017012962@student.unza.zm	M	Languages	Banda	...	NO
3	2017012963@student.unza.zm	M	Civic	Bwalya	...	NO
4	2017008514@student.unza.zm	M	History	Bwalya	...	NO
5	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
6	2017012923@student.unza.zm	M	Civic	Chatabela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012964@student.unza.zm	M	Mathematics	Chitete	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012965@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012990@student.unza.zm	M	Mathematics	Hamamba	...	NO
14	2017012912@student.unza.zm	M	Civic	Imakando	...	NO
15	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
16	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
17	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
18	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
19	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

March 29, 2021

CSC 5741 (2020/21) L03 - 62

Pandas (4/9)

- Pandas series
 - One dimensional labeled array that can hold any data type.
 - Similar to column in Spreadsheet applications

	StudentID	Gender	Minor	LastName	... PassedTest3	
0	2017013156@student.unza.zm	M	Geography	Anayawa	...	NO
1	2017012891@student.unza.zm	M	Civic	Banda	...	NO
2	2017012962@student.unza.zm	M	Languages	Banda	...	NO
3	2017012963@student.unza.zm	M	Civic	Bwalya	...	NO
4	2017008514@student.unza.zm	M	History	Bwalya	...	NO
5	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
6	2017012923@student.unza.zm	M	Civic	Chatabela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012964@student.unza.zm	M	Mathematics	Chitete	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012965@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012990@student.unza.zm	M	Mathematics	Hamamba	...	NO
14	2017012912@student.unza.zm	M	Civic	Imakando	...	NO
15	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
16	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
17	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
18	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
19	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

March 29, 2021

CSC 5741 (2020/21) L03 - 63

Pandas (5/9)

- Columns
 - Ellipse indicate more columns. Structure of data frame indicated on last line of output

	StudentID	Gender	Minor	LastName	... PassedTest3	
0	2017013156@student.unza.zm	M	Geography	Anayawa	...	NO
1	2017012891@student.unza.zm	M	Civic	Banda	...	NO
2	2017012962@student.unza.zm	M	Languages	Banda	...	NO
3	2017012963@student.unza.zm	M	Civic	Bwalya	...	NO
4	2017008514@student.unza.zm	M	History	Bwalya	...	NO
5	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
6	2017012923@student.unza.zm	M	Civic	Chatabela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012964@student.unza.zm	M	Mathematics	Chitete	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012965@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012990@student.unza.zm	M	Mathematics	Hamamba	...	NO
14	2017012912@student.unza.zm	M	Civic	Imakando	...	NO
15	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
16	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
17	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
18	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
19	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

March 29, 2021

CSC 5741 (2020/21) L03 - 64

Pandas (6/9)

- Index

- Automatically generated, but can be changed
- Uniquely identifies rows in the DataFrame

	StudentID	Gender	Minor	LastName	... PassedTest3	
0	2017013156@student.unza.zm	M	Geography	Anayava	...	NO
1	2017012891@student.unza.zm	M	Civic	Banda	...	NO
2	2017012892@student.unza.zm	M	Languages	Banda	...	NO
3	2017012893@student.unza.zm	M	Civic	Bwalya	...	NO
4	2017080514@student.unza.zm	M	History	Bwalya	...	NO
5	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
6	2017012923@student.unza.zm	M	Civic	Chabela	...	YES
7	2017012938@student.unza.zm	M	History	Chakulya	...	NO
8	2017012939@student.unza.zm	M	Mathematics	Chabula	...	NO
9	2017008343@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012966@student.unza.zm	F	Languages	Chilumba	...	NO
11	2017012967@student.unza.zm	F	History	Chisha	...	NO
12	2017012939@student.unza.zm	F	History	Gondwe	...	NO
13	2017012940@student.unza.zm	M	Mathematics	Hamaambo	...	NO
14	2017012941@student.unza.zm	F	Civic	Imakando	...	NO
15	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
16	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
17	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
18	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
19	2017004431@student.unza.zm	M	Mathematics	Kamsanga	...	YES

March 29, 2021

CSC 5741 (2020/21) L03 - 55

Pandas (7/9)

- Data

	StudentID	Gender	Minor	LastName	... PassedTest3	
0	2017012891@student.unza.zm	M	Civic	Banda	...	NO
1	2017012912@student.unza.zm	M	Languages	Banda	...	NO
2	2017012913@student.unza.zm	M	Civic	Bwalya	...	NO
3	2017080514@student.unza.zm	M	History	Bwalya	...	NO
4	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
5	2017012923@student.unza.zm	M	Civic	Chabela	...	YES
6	2017012983@student.unza.zm	M	History	Chakulya	...	NO
7	2017012984@student.unza.zm	M	Mathematics	Chabula	...	NO
8	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
9	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
10	2017012961@student.unza.zm	F	Languages	Chisha	...	NO
11	2017012965@student.unza.zm	F	History	Gondwe	...	NO
12	2017012939@student.unza.zm	F	History	Hamaambo	...	NO
13	2017012940@student.unza.zm	M	Mathematics	Imakando	...	NO
14	2017012941@student.unza.zm	F	Civic	Indakando	...	NO
15	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
16	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
17	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
18	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
19	2017004431@student.unza.zm	M	Mathematics	Kamsanga	...	NO

March 29, 2021

CSC 5741 (2020/21) L03 - 66

Pandas (8/9)

- Some common operations

- Reading data files
 - df.read_csv([...])
 - df.read_html([...])
 - df.read_json([...])
 - df.read_*
- Inspecting dataframes
 - df.head([...])
 - df.tail([...])
 - df.columns
 - df[...]

March 29, 2021

CSC 5741 (2020/21) L03 - 67

Pandas (9/9)

- Some common operations

- Converting to different file formats
 - df.to_csv([...])
 - df.to_excel([...])
 - df.to_sql([...])
 - df.to_*
- Renaming columns
 - df.rename(columns={...})
- Aggregating data
 - df.groupby(['...']).mean()
 - df.groupby(['...']).max()

March 29, 2021

CSC 5741 (2020/21) L03 - 68

Pandas—Exercise

- See Jupyter Notebook “2020/21 CSC 5741: Lecture #04 Notebook—Python for Machine Learning” (<http://bit.ly/2Q2T2Lw>)

March 29, 2021

CSC 5741 (2020/21) L03 - 69

Lecture Series Outline

- Part I: Jupyter Notebooks
- Part II: Google Colab
- Part III: Getting Started With Python
- Part IV: Core Python Libraries
 - Datasets
 - matplotlib
 - pandas
 - Scikit-learn

March 29, 2021

CSC 5741 (2020/21) L03 - 70

Scikit-learn

- Scikit-learn
 - Ensure that the module is installed by using the import statement

```
lightonphirl@lightonphirl-Lenovo-ideapad-320-15IKB:~$ pip3 install sklearn
Collecting sklearn
  Downloading https://files.pythonhosted.org/packages/1e/7a/dbb3be0ce9bd5c8b7e3d
sklearn-0.0.tar.gz
Collecting scikit-learn (from sklearn)
  Downloading https://files.pythonhosted.org/packages/5e/82/c0de5839d613b82bdd0
scikit_learn-0.26.3-cp36-cp36m-manylinux1_x86_64.whl (5.4MB)
    0% |████████████████████████████████| 20KB 55KB/s eta 0:01:38
```

```
lightonphirl@lightonphirl-Lenovo-ideapad-320-15IKB:~$ python3
Python 3.6.7 (default, Oct 22 2016, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sklearn
>>> dir(sklearn)
['__SKLEARN_SETUP__', '__all__', '__builtins__', '__cached__', '__check_build',
 '__doc__', '__file__', '__name__', '__package__', '__path__', '__spec__', '__version__',
 '__config__', 'base', 'clone', 'externals', 'get_config', 'logger', 'logging', 're', 'set_config',
 'setup_module', 'show_versions']
>>>
```

March 29, 2021

CSC 5741 (2020/21) L03 - 71

scikit-learn—Exercises

- See Jupyter Notebook “2020/21 CSC 5741: Lecture #04 Notebook—Python for Machine Learning” (<http://bit.ly/2Q2T2Lw>)

March 29, 2021

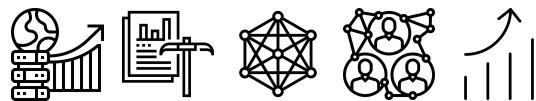
CSC 5741 (2020/21) L03 - 72

Bibliography

- [1] A Byte of Python
<https://python.swaroopch.com>
- [2] Python 3.4 Programming Tutorials
<https://www.youtube.com/playlist?list=PL6gx4Cwl9DGAcbMi1sH6oAMk4JHw91mC>
- [3] Python for Beginners | Python.org
<https://www.python.org/about/gettingstarted>
- [4] Pyplot tutorial – Matplotlib 3.0.3 documentation
<https://matplotlib.org/tutorials/introductory/pyplot.html>
- [5] 10 Minutes to pandas – pandas 0.22.0 documentation
<https://pandas.pydata.org/pandas-docs/version/0.22/10min.html>

March 29, 2021

CSC 5741 (2020/21) L03 - 73



CSC 5741 (2020/21)

Data Mining and Warehousing

Lecture 2: Python for Data Mining and Machine Learning

Lighton Phiri
Department of Library & Information Science
University of Zambia
<http://lis.unza.zm/~lightonphiri>