

CSC 5741

Lecture 2: Python for Data Mining and Machine Learning

Lighton Phiri <lighton.phiri@unza.zm>

Department of Library and Information Science
University of Zambia

Announcements—March 26, 2019

- **CSC 5741 Moodle site up and running**
 - All assignments will be available and submitted via The Moodle
- **Mini Projects**
 - Open date: March 29, 2019
 - Due date: May 17, 2019
- **Paper reading #01**
 - Open date: March 29, 2019
 - Due date: April 5, 2019
 - Paper suggestion rotation—Two every week
- **Invited Academic Talk**
 - April 2, 2019—Lillian Mzyece

Announcements—April 2, 2019 (1/3)

- No talk today—April 2, 2019—as none of the invited guests are able to show up
 - I could give another talk if people wish :P
- Mulizwa will come through to give a talk on April 9, 2019—next week
 - Biography & abstract will be sent
 - Please show up on time and ask a lot of questions

CSC 5741 Invited Talks Slots
by Lighton Phiri • 4 days ago • Print

University of Zambia
All times displayed in Africa/Lusaka

Table Calendar

	Apr 2 TUE	Apr 9 TUE	Apr 16 TUE	Apr 23 TUE	Apr 30 TUE	May 7 TUE	May 14 TUE
5 participants	✓ 0/1	✓ 1/1	✓ 1/1	✓ 1/1	✓ 0/1	✓ 1/1	✓ 1/1
Enter your name	●	●	●	●	●	●	●
Lillian Mzyece						✓	
Andreya Kumwenda				✓			
Friday C. Chazanga		✓					
Soft Mulizwa	✓						
Francis Chulu						✓	

<https://doodle.com/poll/bmv7b5yqq5nbdu9n>

Announcements—April 2, 2019 (2/3)

- Paper reading summaries are due on April 8, 2019
 - Please follow the reading assignment specifications

CSC 5741: Data Mining and Warehousing ›
Paper Reading Summary Assignment #01 | Open Date: April 1, 2019
1 post by 1 author

 me (Lighton Phiri change)

★ Write a short summary of the "Data-driven intervention-level prediction modeling for academic performance" publication [1, 2] by Mgala and Mbogho.
* The summary should be no more than 250 words and must be submitted as a single PDF document via The Moodle or via the course Team Drive.
* The filename for you PDF document should be STUDENTID_paper_reading_summary1.pdf: replace STUDENTID with your STUDENT ID.
* Use A4 page site, single line spacing, 12pt Serif font and 2.54cm margins.

The deadline for submitting the summary is April 8, 2019 at 23H59 GMT+2. No late hand-ins, printed or handwritten submissions will be accepted.

[1] <https://doi.org/10.1145/2737856.2738012>
[2] See attachment: a2-mgala.pdf

Best wishes.

--
Lighton Phiri, PhD
Department of Library and Information Science
University of Zambia
Lusaka, Zambia
Email: lighton.phiri@unza.zm
Web: <http://lis.unza.zm/~lightonphiri>

Attachments (1)


a2-mgala.pdf

245 KB [View](#) [Download](#)

<https://groups.google.com/a/unza.zm/d/forum/csc5741>

Announcements—April 2, 2019 (3/3)

- Please make your selections as soon as possible
 - You might want to make selections to work in packs
- Errata
 - Distribution of marks for presentation was less than 100%
 - Page limit for technical report is four (4) pages not six (6) pages

CSC 5741 Mini Project (2018/19)
by Lighton Phiri • a day ago • Print

ter te	2 (b) NETD: Classification universities based on ETD output	2 (c) NETD: Cluster analysis of ETDs by subject area	3 (a) YouTube: Recommend YouTube Videos to undergraduate students	3 (b) YouTube: Classification of YouTube comments to undergraduate students	3 (c) YouTube: Classification of random YouTube videos to undergraduate students	4 (a) Facebook: Classification of posts on popular 'Zambian' Facebook pages
	✓ 1/1	✓ 0/1	✓ 0/1	✓ 1/1	✓ 0/1	✓ 1/1
	●	●	●	●	●	●
				✓		
	✓					✓

<https://doodle.com/poll/534szgg6aw56yxtz>

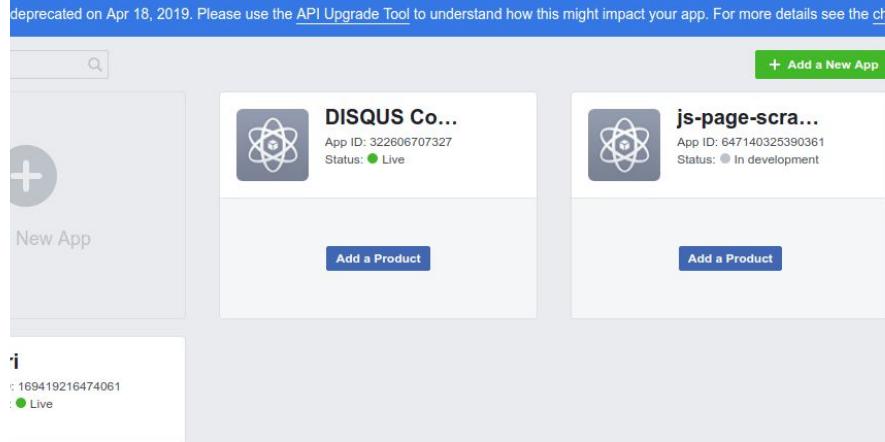
Todos | Showcases—April 2, 2019 (1/6)

```
</metadata>
▼<about>
  ▼<provenance xmlns="http://www.openarchives.org/OAI/2.0/provenance" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi
    http://www.openarchives.org/OAI/2.0/provenance.xsd">
    ▼<originDescription harvestDate="2014-02-04T04:15:33Z" altered="false">
      <baseURL>http://etd.uovs.ac.za/cgi-bin/NDLTDO/UFS/oai.pl</baseURL>
      <identifier>oai:etd.uovs.ac.za:etd-07172013-155725</identifier>
      <datestamp>2013-07-17</datestamp>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNamespace>
    </originDescription>
  </provenance>
</about>
</record>
▼<resumptionToken completeListSize="171424" cursor="0">
  2014-02-04T16:15:33Z!2037-01-01T00:00:00Z!!oai_dc!1000!171424!oai:union.ndltd.org:ufs/oai:etd.uovs.ac.za:etd-07172013-155725
</resumptionToken>
</OAI-PMH>
```

<http://www.netd.ac.za>

- Extraction of records of NETD and NDLTD Union Catalog is best done using the OAI-PMH protocol
 - Use the 'resumptionToken' to loop through batches of records

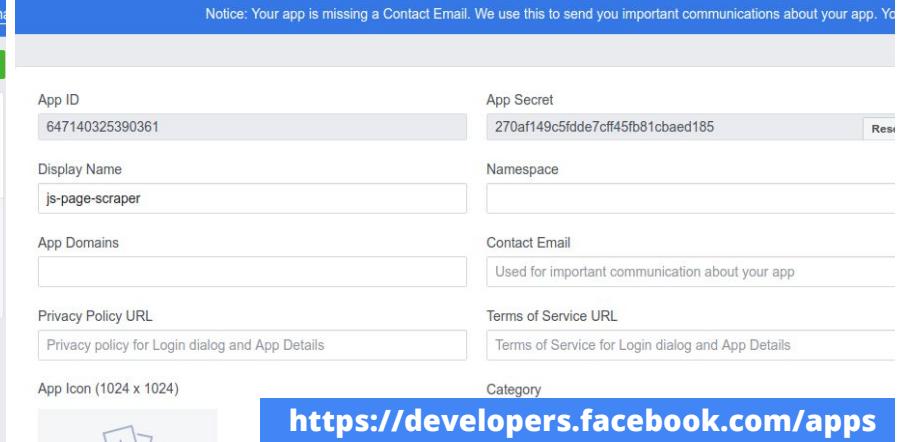
Todos | Showcases—April 2, 2019 (2/6)



The screenshot shows the Facebook Developers App Dashboard. On the left, there's a sidebar with icons for 'New App' and '169419216474061'. The main area displays two app cards:

- DISQUS Co...**: App ID: 322606707327, Status: Live. Includes a 'New App' button.
- js-page-scraper...**: App ID: 647140325390361, Status: In development. Includes a 'New App' button.

A green button at the top right says '+ Add a New App'.



The screenshot shows the 'App Settings' page for the 'js-page-scraper' app. It has a notice about missing contact email. The form fields are as follows:

App ID	647140325390361	App Secret	270af149c5fdde7cff45fb81cbaed185
Display Name	js-page-scraper	Namespace	
App Domains		Contact Email	Used for important communication about your app
Privacy Policy URL	Privacy policy for Login dialog and App Details	Terms of Service URL	Terms of Service for Login dialog and App Details
App Icon (1024 x 1024)		Category	https://developers.facebook.com/apps

- Extraction of data from Facebook is best done using Facebook's Graph API
 - You will need to get access tokens for this to work

Todos | Showcases—April 2, 2019 (3/6)

[lightonphiri / code-fb-page-scrapers](#) Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

A simple JavaScript-based script for extracting information from Facebook pages. Edit

Manage topics

7 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

 lightonphiri	Git sync transaction	Latest commit 84390da on Jan 29, 2015
 js	Git sync transaction	4 years ago
 .gitignore	Initial commit	4 years ago
 README.md	Initial commit	4 years ago
 index.html	Added templates and boilerplate code	

<https://github.com/lightonphiri/code-fb-page-scrapers>

Todos | Showcases—April 2, 2019 (4/6)

- YouTube data is accessible via the YouTube Data API
 - Those working with comments will need to both the 'Comments' and 'CommentThreads' resources

The screenshot shows the YouTube Data API documentation for the 'CommentThreads' resource. The page has a red header with the YouTube logo, 'Data API', and navigation links for 'HOME', 'GUIDES', 'REFERENCE', 'SAMPLES', and 'SUPPORT'. A search bar and 'ALL PRODUCTS' dropdown are also in the header. The main content area has a sidebar with links like 'Overview', 'Activities', 'Captions', etc., under 'Comments'. The main content area is titled 'CommentThreads' and describes it as a resource for YouTube comment threads. It includes sections for 'Methods' (with 'list', 'insert', 'update' methods), 'Resource representation' (with a JSON example), and a 'Comments' section. A blue banner at the bottom contains the URL <https://developers.google.com/youtube/v3>.

Todos | Showcases—April 2, 2019 (5/6)

- Google Scholar does not provide an API, so you are going to have to scrap for data
 - Python BeautifulSoup library might be useful
 - Some projects on GitHub and Bitbucket might be useful when extracting data from Google Scholar

The screenshot shows a GitHub repository page for `ckreibich/scholar.py`. The repository has 49 issues, 44 pull requests, 0 projects, and 0 wiki pages. It has 5 contributors and was last updated on Feb 1, 2017. The README.md file describes the module as a parser for Google Scholar's output, capable of extracting content excerpts. The scholar.py file is also described as less aggressive in its requests. The repository URL is <https://github.com/ckreibich/scholar.py>.

A parser for Google Scholar, written in Python

47 commits 1 branch 0 releases 5 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

ckreibich Less aggressive use of num= requests; version bump to 2.11 Latest commit 7e6efb4 on Feb 1, 2017

README.md Ability to extract content excerpts as reported in search results. 4 years ago

scholar.py Less aggressive use of num= requests; version bump to 2.11 2 years ago

README.md

scholar.py

scholar.py is a Python module that implements a querier and parser for Google Scholar's output. Its classes can be used independently, but it can also be invoked as a command-line tool.

The script used to live at <http://icir.org/christian/scholar.html>, and I've moved it here so I can more easily manage the various patches and suggestions I'm receiving for scholar.py. Thanks guys, for all your interest! If you'd like to get in touch, email me at christian@icir.org or ping me on [Twitter](#).

Cheers,
Christian

Features

<https://github.com/ckreibich/scholar.py>

Todos | Showcases—April 2, 2019 (6/6)

The screenshot shows the Microsoft Academic homepage. At the top left is the "Microsoft Academic" logo. On the right are two user icons: a double quote mark and a "Sign up / Sign in" link. Below the header is a dark blue banner with the text "Research more, search less". A search bar with the placeholder "Search any topic, author, journal, etc. or any combination of these" is centered. To the right of the search bar is a magnifying glass icon. On the far right, there's a sidebar with various academic metrics: 213,973,556 Papers, 256,683,013 Authors, 663,508 Topics, 4,374 Journals, and 25,496 Books. A red box highlights the bottom navigation bar, which contains links for "Home", "About", "Contact", "Help", and "Privacy". The footer features the text "Unleash the Power of Research" and the URL "http://www.netd.ac.za".

- **Read up on how to use Microsoft Academic Graph API to extract data from Microsoft Academic Search**

Announcements—April 9, 2019 (1/3)

- **Paper reading suggestions account towards participation**
 - Simple formula will be used to aggregate scores from other activities, e.g. talks.
- **Hint: Look for papers aligned to the problem you are solving in the Mini Project**

No.	First Name	Lastname
1	Chola	Paul Modest
2	Daka	John Chrispin
3	Lamaswala	Inonge
4	Mubanga	Mubanga
5	Mukuma	Nonde
6	Mulenga	David
7	Mumbi	Memory
8	Mutende	Kaumba
9	Nongola	Justin
10	Nyambe	Teddy
11	Phiri	Jonathan
12	Sampa	Anthny Wila
13	Shamane	Tasha

[https://groups.google.com/a/unza.zm/forum
/?hl=en#!forum/csc5741](https://groups.google.com/a/unza.zm/forum/?hl=en#!forum/csc5741)

Announcements—April 9, 2019 (2/3)

- Mini Project questions involving Facebook's Graph API can use WordPress REST API
- Anyone managed with Graph API
 - WordPress REST API route: restrict model implementation to popular sites like LusakaTimes, ZambianWatchDog & Mwebantu.

The screenshot shows the REST API Handbook interface. At the top, there's a search bar and a navigation menu with links to 'CHAPTERS', 'REST API Handbook', 'Reference' (which is currently selected), and other sections like 'Posts', 'Post Revisions', 'Categories', etc. Below the menu, under 'TOPICS', there's a list of endpoints: 'Posts', 'Post Revisions', 'Categories', 'Tags', 'Pages', 'Comments', 'Taxonomies', 'Media', 'Users', 'Post Types', 'Post Statuses', and 'Settings'. Under each endpoint, there are 'Arguments', 'Definition', and 'Example Request' links. At the bottom, there's a section titled 'Schema #' with a description: 'The schema defines all the fields that exist for a post object.' A table lists two fields: 'date' (with a definition of 'The date the object was published, in the site's timezone.' and context links for 'view', 'edit', and 'embed') and 'date_gmt' (with a definition of 'The date the object was published, as GMT.').

<https://developer.wordpress.org/rest-api>

Announcements—April 9, 2019 (3/3)

```
id: 229497,
date: "2019-04-09T07:33:43",
date_gmt: "2019-04-09T05:33:43",
guid: {
  rendered: "https://www.lusakatimes.com/?p=229497"
},
modified: "2019-04-09T07:33:43",
modified_gmt: "2019-04-09T05:33:43",
slug: "faz-div-1-wrap-kansanshi-top-zone-2",
status: "publish",
type: "post",
link: "https://www.lusakatimes.com/2019/04/09/faz-div-1-wrap-kansanshi-top-zone-2/",
title: {
  rendered: "FAZ DIV 1 WRAP: Kansanshi top Zone 2"
},
content: {
  rendered: "<p>Kansanshi Dynamos have won the first half of the 2019 FAZ Division Zone 2 season with 36 points after a 2-1 win over Zesco Luapula in the Week 15 m  
a three point gap half way into the season.</p> <p>The Solwezi side benefited from Zesco's own goal to triumph in their latest match at home in Solwezi.</p> <p>G  
home in Ndola.</p> <p>Gomes remain stuck on 33 points from 15 matches played while Konkola are six points behind in third place.</p> <p>Fourth placed Indeni and  
are leading with a one point gap after overcoming Lundazi United 1-0 to amass 33 points while National Assembly are second on the table.</p> <p>Zambeef are top  
placed Kabwe Youth are three points behind.</p> <p>Zone 4 leaders Young Green Eagles have 31 points, two above second placed Zesco Shockers after 15 matches play  
Nkwazi 2-0 Riflemen</p> <p>Chipata City Council 1-0 Petauke United</p> <p>Lundazi United 0-1 Young Green Buffaloes</p> <p>Happy Hearts 0-2 Police College</p> <p>  
Rangers 2-2 Kafue Celtic</p> <p>Paramilitary 1-1 Lusaka City Council</p> <p>National Assembly 2-1 Romeki FC</p> <p>ZONE TWO</p> <p>Ndola United 3-1 Roan United</  
0-0 Indeni</p> <p>Nchanga Rangers 2-0 FOM FM</p> <p>Gomes 0-1 Konkola Blades</p> <p>ZNS Luamfumu 2-0 Trident</p> <p>Kansanshi Dynamos 1-0 Zesco Luapula</p> <p>Ka  
<p>Intersport Youth 1-2 Riverside United</p> <p>Kateshi Coffee Bullets 0-0 Kabwe Youth Soccer Academy</p> <p>Malalo Police 0-1 Zambeef FC</p> <p>Mungwi Hotspur  
Eagles</p> <p>Tazara Express 2-0 Tazara Rangers</p> <p>Real Nakonde 3-1 Mpulungu Harbour</p> <p>Kabwe Rangers 2-0 Chindwin Sentries</p> <p>ZONE FOUR</p> <p>Mazab  
Medics</p> <p>Sinazongwe United 1-0 Zesco Shockers</p> <p>Choma football Stars 0-2 Livingstone Pirates</p> <p>Kalomo Jetters 1-0 New Monze Swallows</p> <p>Young  
Stars</p> <p>Blue Arrows 0-0 Katima Border Stars</p> ",
  protected: false
},
excerpt: {
  rendered: "<p>Kansanshi Dynamos have won the first half of the 2019 FAZ Division Zone 2 season with 36 points after a 2-1 win over Zesco Luapula in the Week 15 m  
three point gap half way into the season. The Solwezi side benefited from Zesco's [&hellip;];</p> ",
  protected: false
},
  https://www.lusakatimes.com/wp-json/wp/v2/posts

```

Lecture Series Outline

- Part I: Getting Started With Python
- Part II: pandas, matplotlib and scikit-learn
- Part III: Academic Talk [Trial]
- Part IV: Paper Reading [Trial]
- Part V: About Next Week

Lecture Series Outline

- **Part I: Getting Started With Python**
 - Introduction
 - Installation and Setup
 - Basics
 - Data Structures
 - Flow Control
 - Functions and Modules
- **Part II: pandas, matplotlib and scikit-learn**
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

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.
```

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

Getting Started With Python (3/3)

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

Installation and Setup

- [...]

Installation and Setup (1/2)

- 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 with the "Docs" menu item highlighted in yellow. The main content area displays a code snippet demonstrating Python list comprehensions and the enumerate function.

```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types
Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)

Installation and Setup (2/2)

- 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 question "What IDE to use for Python? [closed]". The question has 1028 answers and was last edited on Nov 11 '14 at 1:57. Below the question, there is a "Results" section with a table comparing various IDEs based on several criteria. The table includes columns for Cross Platform, Commercial/Free, Auto Code Completion, Integrated Python Debugging, and Bracket Matching. The table lists eight IDEs: Atom, BlackAdder, BlueFish, ConTEXT, DABO, Dr.Python, and DreamPie.

		Cross Platform	Commercial/Free	Auto Code Completion	Integrated Python Debugging	Bracket Matching
1		Y	F		Y	Y
2	Atom	Y	C			
3	BlackAdder	Y	C			
4	BlueFish	L				
5	ConTEXT	W	C			
6	DABO	Y				
7	Dr.Python		F		Y	
8	DreamPie	F				

<https://stackoverflow.com/q/81584/664424>

Basics

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

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

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']`

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`

Comments (2/2)

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

```
''' Author: Paul Chola  
Course: CSC 5741  
Lecture #02 '''
```

```
""" Author: Tasha Shamane  
Course: CSC 5741  
Lecture #02 """
```

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`

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"`

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**

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, ...)`

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'>**

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

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}`

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

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
```

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 *`

Lecture Series Outline

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
 - matplotlib
 - pandas
 - scikit-learn
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

Matplotlib (1/7)

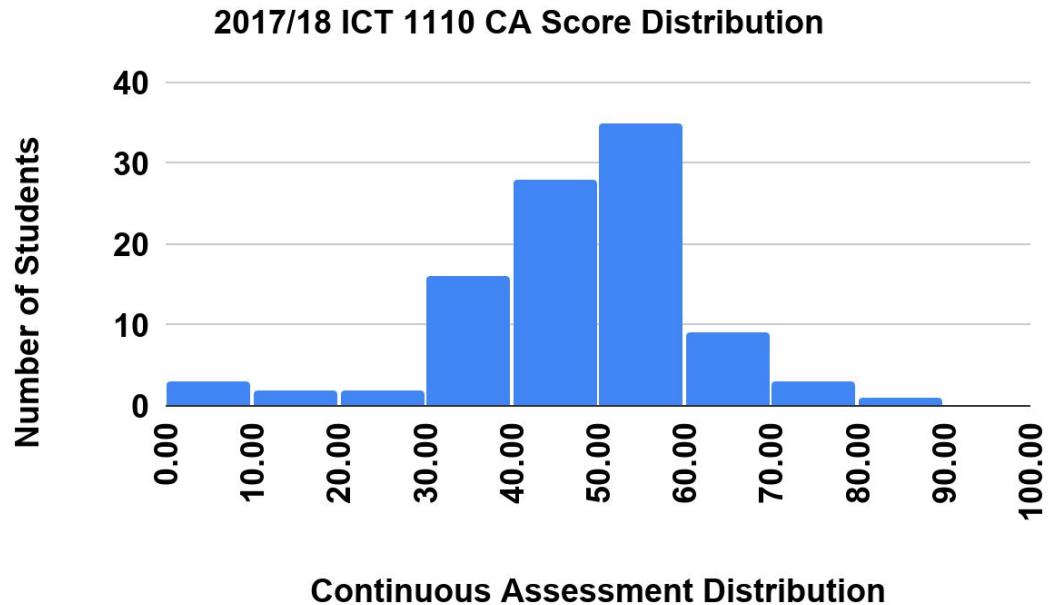
- The matplotlib library is best installed using pip, as with all libraries or using apt-get, if on Mac or Linux
 - pip3 install matplotlib
 - sudo apt-get install python-matplotlib
- Test installation by importing a library module

```
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ pip3 install matplotlib
Requirement already satisfied: matplotlib in ./local/lib/python3.6/site-packages
Requirement already satisfied: python-dateutil>=2.1 in ./local/lib/python3.6/site-
Requirement already satisfied: cycler>=0.10 in ./local/lib/python3.6/site-pac
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in ./lo
matplotlib) (2.3.1)
Requirement already satisfied: kiwisolver>=1.0.1 in ./local/lib/python3.6/site-
Requirement already satisfied: numpy>=1.10.0 in ./local/lib/python3.6/site-pac
Requirement already satisfied: six>=1.5 in ./local/lib/python3.6/site-packages
1.12.0)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (fr
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ █
```

```
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ python3
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 matplotlib
>>> dir(matplotlib)
['_MatplotlibDeprecationWarning', 'MutableMapping', 'Parameter', 'Path', 'RcParams', 'URL_REGEX',
'PENDIX', '__bibTeX__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
'__path__', '__spec__', '__version__', '__version__numpy__', '__warningregistry__', '__add_data_doc',
'__or_data__', '__create_tmp_config_dir', '__deprecated_ignore_map', '__deprecated_map', '__deprecated_rema
tials_fmt', '__get_config_or_cache_dir', '__get_data_path', '__get_xdg_cache_dir', '__get_xdg_config_o
log', '__logged_cached', '__open_file_or_url', '__parse_commandline', '__preprocess_data', '__rc_params',
'__set_logger_verbose_level', '__verbose_msg', '__version__', '__atexit__', 'cbook', 'checkdep_dvipng',
'checkdep_tknscape', 'checkdep_pdftops', 'checkdep_ps_distiller', 'checkdep_usetex', 'colors', 'comp
lib', 'cycler', 'dateutil', 'dedent', 'defaultParams', 'default_test_modules', 'distutils', 'font
ools', 'get_backend', 'get_cachedir', 'get_configdir', 'get_data_path', 'get_home', 'get_label', 'g
importlib', 'inspect', 'interactive', 'io', 'is_interactive', 'is_url', 'locale', 'logging', 'mat
plotlib', 'numpy', 'os', 'pprint', 'pyparsing', 'rc', 'rcParams', 'rcParamsDefault', 'rcParamsOrigi
nale', 'rc_file_defaults', 'rc_params', 'rc_params_from_file', 'rcdefaults', 'rcsetup', 're', 'san
', 'stat', 'subprocess', 'sys', 'tempfile', 'test', 'tk_window_focus', 'urllib', 'use', 'validate_
arnings']
```

Matplotlib (2/7)

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



Matplotlib (3/7)

- **Creating plots is a four-step process**
 - 1) Import matplotlib
 - 2) Draw the plot
 - 3) Specify plot aesthetics
 - 4) Render plot

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

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

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

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

Matplotlib—Exercise

- JCTR regularly compiles Basic Needs Basket statistics for major towns in The Republic of Zambia. Using the Lusaka April 2018 BNB dataset (dataset available on <https://goo.gl/1zSigy>):
 - Draw a line plot showing the trends of the Lusaka BNB between November 2016 and April 2018
 - Draw a bar plot showing BNB costs across Zambia in April 2018
 - Draw a pie chart showing the cost of basic food items in Lusaka for the month of April 2018

Lecture Series Outline

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
 - matplotlib
 - pandas
 - scikit-learn
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

Pandas (1/9)

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

Pandas (2/9)

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

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	2017008915@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	Chaibela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012934@student.unza.zm	M	Mathematics	Chibale	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012966@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012999@student.unza.zm	M	Mathematics	Hamaamba	...	NO
14	2017001325@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	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

+ Shell Shell No. 2 Shell No. 4 Shell No. 3

lecture-02 : python3 - Drop-Down Terminal

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	2017008915@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	Chaibela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012934@student.unza.zm	M	Mathematics	Chibale	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012966@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012999@student.unza.zm	M	Mathematics	Hamaamba	...	NO
14	2017001325@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	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

+ Shell Shell No. 2 Shell No. 4 Shell No. 3

lecture-02 : python3 - Drop-Down Terminal

Pandas (5/9)

- **Columns**

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

	StudentID	Gender	Minor	LastName	...	PassedTest3
0	2017015158@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	2017008915@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	Chaibela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012934@student.unza.zm	M	Mathematics	Chibale	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012966@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012999@student.unza.zm	M	Mathematics	Hamaamba	...	NO
14	2017001325@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	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

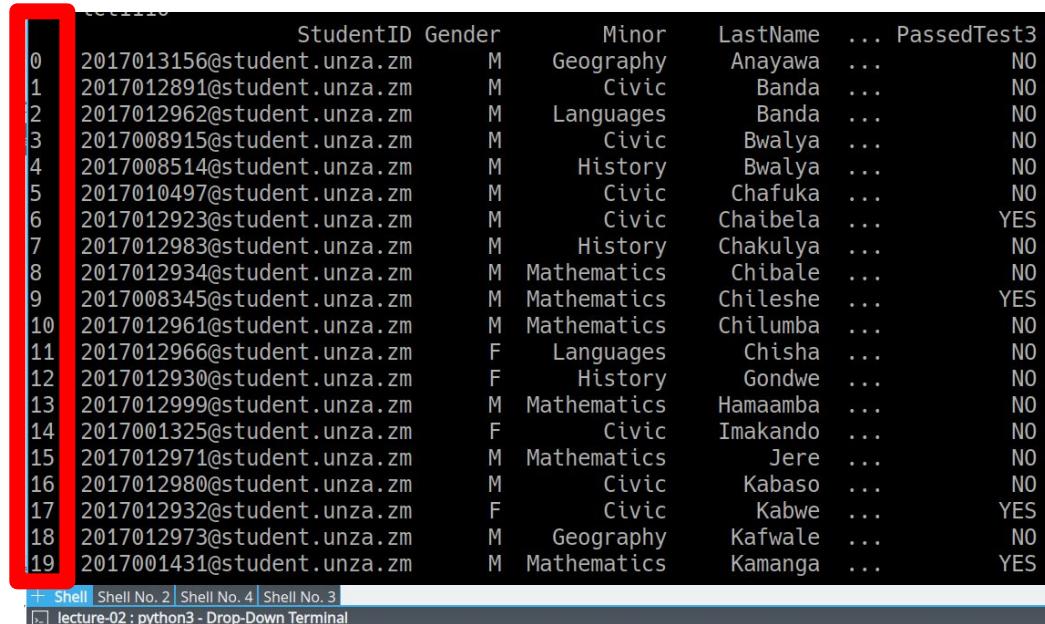
+ Shell Shell No. 2 Shell No. 4 Shell No. 3

lecture-02 : python3 - Drop-Down Terminal

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	Anayawa	...	NO
1	2017012891@student.unza.zm	M	Civic	Banda	...	NO
2	2017012962@student.unza.zm	M	Languages	Banda	...	NO
3	2017008915@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	Chaibela	...	YES
7	2017012983@student.unza.zm	M	History	Chakulya	...	NO
8	2017012934@student.unza.zm	M	Mathematics	Chibale	...	NO
9	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
10	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
11	2017012966@student.unza.zm	F	Languages	Chisha	...	NO
12	2017012930@student.unza.zm	F	History	Gondwe	...	NO
13	2017012999@student.unza.zm	M	Mathematics	Hamaamba	...	NO
14	2017001325@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	2017001431@student.unza.zm	M	Mathematics	Kamanga	...	YES

+ Shell Shell No. 2 Shell No. 4 Shell No. 3

lecture-02 : python3 - Drop-Down Terminal

Pandas (7/9)

- Data

	StudentID	Gender	Minor	LastName	...	PassedTest3
0	2017012891@student.unza.zm	M	Civic	Banda	...	NO
1	2017012962@student.unza.zm	M	Languages	Banda	...	NO
2	2017008915@student.unza.zm	M	Civic	Bwalya	...	NO
3	2017008514@student.unza.zm	M	History	Bwalya	...	NO
4	2017010497@student.unza.zm	M	Civic	Chafuka	...	NO
5	2017012923@student.unza.zm	M	Civic	Chaibela	...	YES
6	2017012983@student.unza.zm	M	History	Chakulya	...	NO
7	2017012934@student.unza.zm	M	Mathematics	Chibale	...	NO
8	2017008345@student.unza.zm	M	Mathematics	Chileshe	...	YES
9	2017012961@student.unza.zm	M	Mathematics	Chilumba	...	NO
10	2017012966@student.unza.zm	F	Languages	Chisha	...	NO
11	2017012930@student.unza.zm	F	History	Gondwe	...	NO
12	2017012999@student.unza.zm	M	Mathematics	Hamaamba	...	NO
13	2017001325@student.unza.zm	F	Civic	Imakando	...	NO
14	2017012971@student.unza.zm	M	Mathematics	Jere	...	NO
15	2017012980@student.unza.zm	M	Civic	Kabaso	...	NO
16	2017012932@student.unza.zm	F	Civic	Kabwe	...	YES
17	2017012973@student.unza.zm	M	Geography	Kafwale	...	NO
18						
19						

+ Shell Shell No. 2 Shell No. 4 Shell No. 3

lecture-02 : python3 - Drop-Down Terminal

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['...']`

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()`

Pandas—Exercise

- Using pandas and the 2018/19 ICT 1110 assessment scores dataset (dataset available on <https://goo.gl/wC1H7Q>):
 - Print out the details (using a List) of the student who got the highest CA score
 - Print out the average CA scores for each of the different Minors
 - Print out the average CA scores for each gender
 - Export a summary table, to HTML, of mean CA scores by student Minors
 - Export, to CSV, a dataset consisting of StudentID, Minor and Total CA

Lecture Series Outline

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
 - matplotlib
 - pandas
 - scikit-learn
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

Scikit-learn

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
 - matplotlib
 - pandas
 - scikit-learn
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

Scikit-learn

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

```
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ pip3 install sklearn
Collecting sklearn
  Downloading https://files.pythonhosted.org/packages/1e/7a/dbb3be0ce9bd5c8b7e3d8
sklearn-0.0.tar.gz
Collecting scikit-learn (from sklearn)
  Downloading https://files.pythonhosted.org/packages/5e/82/c0de5839d613b82bdd08
scikit_learn-0.20.3-cp36-cp36m-manylinux1_x86_64.whl (5.4MB)
    0% ||                               | 20KB 55kB/s eta 0:01:38
```

```
lightonphiri@lightonphiri-Lenovo-ideapad-320-15IKB:~$ python3
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 sklearn
>>> dir(sklearn)
['__SKLEARN_SETUP__', '__all__', '__builtins__', '__cached__', '__check_build__', '__doc__', '__name__', '__package__', '__path__', '__spec__', '__version__', '__config__', 'base', 'clone', 'externals', 'get_config', 'logger', 'logging', 're', 'set_config', 'setup_module', 'showWarnings']
>>>
>>>
```

Lecture Series Outline

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
- **Part III: Academic Talk [Trial]**
 - Towards Increased Online Visibility of Research in Zambia
- **Part IV: Paper Reading [Trial]**
- **Part V: About Next Week**

Lecture Series Outline

- **Part I: Getting Started With Python**
- **Part II: pandas, matplotlib and scikit-learn**
- **Part III: Academic Talk [Trial]**
- **Part IV: Paper Reading [Trial]**
 - L. Phiri (2018) “Research Visibility in the Global South: Towards Increased Online Visibility of Scholarly Research Output in Zambia”
- **Part V: About Next Week**

Research Visibility in the Global South: Towards Increased Online Visibility of Research in Zambia

Lighton Phiri <lighton.phiri@unza.zm>

**Department of Library and Information Science
University of Zambia**

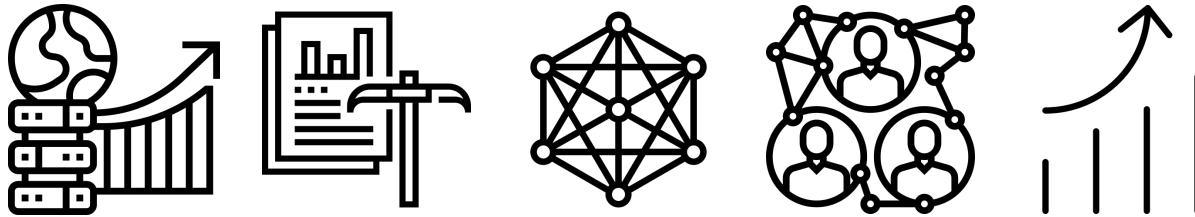
Paper Reading Session [Trial]

- [1] L. Phiri (2018) "Research Visibility in the Global South: Towards Increased Online Visibility of Scholarly Research Output in Zambia"

http://lis.unza.zm/~lightonphiri/papers/paper-icict18-online_visibility.pdf

Bibliography

- [1] **Python for Beginners | Python.org**
<https://www.python.org/about/gettingstarted>
- [2] **Pyplot tutorial – Matplotlib 3.0.3 documentation**
<https://matplotlib.org/tutorials/introductory/pyplot.html>
- [3] **10 Minutes to pandas – pandas 0.22.0 documentation**
<https://pandas.pydata.org/pandas-docs/version/0.22/10min.html>



CSC 5741

Lecture 2: Python for Data Mining and Machine Learning

Lighton Phiri <lighton.phiri@unza.zm>

Department of Library and Information Science
University of Zambia