Python for Data Analysis and Visualization

Instructor: Claudia Carroll

Session 1

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Arts & Sciences at Washington University in St. Louis
Signature Initiative

Goals for the Workshop

- Develop further fluency in the Python programming language
- Learn how to access and perform basic analysis of data contained in tabular form using the Pandas library
- Practice creating basic visualizations of file-based data using Python

Workshop Plan

Class 1	Accessing and Exploring Tabular Data
Class 2	Data Cleaning and Aggregation
Class 3	Static Data Visualizations
Class 4	Plotly: Interactive Data Visualizations

Lesson Structure

1. Review of previous class's material
2. Brief lecture on new concepts/material
3. Demo and Exercise (x2)

*Each class will have accompanying homework exercises, which you are strongly encourage to complete between class sessions.

Today's Lesson Plan

1. Review of Python Data Formats
2. Demo: Accessing Data with Pandas
3. Exercise 1: Practicing open and reading data files

Questions?

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Lecture: Crash Course on Python Basics

Data Types

Strings	"Heuston, we have a problem"
Integers	35
Floats	35.6
Booleans	True/False

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Data Collection Types

List: ["apple", 12, "computer science", "apple", 13.2] • Order is saved • Can be rearranged after list is defined • Can contain duplicates • Elements can be added or removed • Indicated by square brackets	**Set**: {"orange", "house", 102} □ Order is not saved (unordered) □ Cannot contain duplicates □ Cannot add or remove elements once defined □ Indicated by curly brackets
Tuple : ("apple", 29, 32) □ Order is saved □ Order cannot be rearranged after tuple is defined □ Can contain duplicates □ Elements cannot be added or removed □ Indicated by parentheses	**Dictionary:** {"name": "Anne", "age": 19, "major": "communications"} □ Stores data in key/value pairs □ Order is saved (as of Python 3.7) □ Duplicate values permitted, but not duplicate keys within one item □ Elements can be added and removed □ Indicated by curly brackets and colons

Identify Data Types

```
1    x = 42
2    y = "Hello, World!"
3    z = [1, 2, 3]
4    w = {"name": "Alice", "age": 30}
5    b = ["ringo", "paul", "george", "john"]
6
```

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Indices

- An index is the **position of a subset of data** within a data type (e.g. letters in a string, or elements in a list)
- In python, the index starts with **0**
- Elements can be access by index with the following syntax: **list[i]**
- **Example:**

```
>>> fruits = ["apples", "oranges", "melons"]
>>> print(fruits[1])
"oranges"
```

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Boolean Statements

- A **Boolean statement** is a statement that is either True or False
- Boolean statements are primarily used to filter data or methods based on certain conditions
- Boolean statements are usually produced using **Boolean operators**

```
>>> age = 15
>>> print(age < 12)
False
>>> print (age > 12)
True
```

Arithmetic Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (divide and return the remainder)
**	Exponential
//	Floor division (divide and round down to the nearest whole number)

Comparison Operators

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Great than or equal to
<=	Less than or equal to

Conditionals

- Used to *filter* data for further operations

```
number = 0

if number > 0:
 print('Positive number’)
elif number <0:
 print('Negative number’)
else:
 print('Zero') print('This statement is always executed')
```

For Loops

- Used to *iterate* over a sequence of data (string, list, dictionary etc.)
- Can be *nested*

Syntax:

```
fruits = ["apple", "banana", "cherry"]

for x in fruits:
  for letter in x:
    print(letter)
```

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Functions

- Blocks of pre-written code
- Created by user or included in Python libraries
- Must be *defined* before being *called*
- Data can be passed to a function as arguments

```
foods = [“eggs”, “bread”, “milk”, “cookies”]

>>> def long_strings(x):
      for food in foods:
        if len(food) > 5:
          print(food)
      return

>>> print(long_string(foods))
cookies
```

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

What is Pandas?

- The most common Python *library* for basic data manipulation
- Built on the NumPy library
- Often used in conjunction with additional libraries for advanced data analysis and visualization, such as matplotlib or SciPy
- Particularly suited to working with *tabular* data (csv, tsv, excel etc.)
- Reads tabular data as a *dataframe*
- A dataframe is a 2-dimensional array in python

Uses of Pandas?

- Reading data stored in CSV files (other file formats can be read as well)
- Slicing and subsetting data
- Dealing with missing data
- Inserting and deleting columns from data structures
- Aggregating data
- Joining of datasets (after they have been loaded into Dataframes)

Class Materials

https://github.com/ClaudiaECarroll/python_data

1. Download all three CSV files from class_1 to your computer

2. Upload those three files to Google Colab

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Demo: Intro to Pandas

Exercise 2:

1. Write the code to print each country in the data file gdp_africa.csv and that country's mean gdp between 1952 and 1982.

2. Now write the code to print each year (as represented by the column headings) in gdp_africa.csv and the mean GDP in Africa that year. How does your code differ from part one?

Exercise 2 Part 1: Solution

Write the code to print each country in the data file gdp_africa.csv and that country's mean gdp between 1952 and 1982.

```
df_africa = pd.read_csv
("/content/drive/MyDrive/workshop_data/gdp_africa.csv", index_col=0)


countries_africa = df_africa.index.values

for x in countries_africa:
        y = df_africa.loc[x, "1952": "1982"].mean()
        print(x, y)
```

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Exercise 2 Part 2: Solution

Now write the code to print each year (as represented by the column headings) in gdp_africa.csv and the mean GDP in Africa that year. How does your code differ from part one?

```
years_africa = df_africa.columns.values

for x in years_africa:
y = df_africa[x].mean()
print(x, y)
```

Homework!

1. Complete in-class exercises
2. Complete Class One Homework Exercises

Materials and homework can be access via my GitHub repo for this class:

https://github.com/ClaudiaECarroll/python_data

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)