Lab 1: Getting the Feel of Data Analysis!

Nischal Regmi
Everest Engineering College

2022

1 Introduction

This lab will focus on introducing you to the art of data analysis. We will be using the Python language for data mining. Students will get acquainted with basic Python constructs, learn how to load/save data files and displaying some basic types of plots. Although Python is a general purpose object oriented programming language, in our course, we will be mostly writing scripts without using user defined functions and classes.

There are many IDEs for Python. I prefer Sypder as it has a nice interface for visualizing and editing data tables, thus making it friendly for data analysis related tasks. In addition, Spyder comes as a single package containing both the IDE and Python compilers. This makes the process of installation easy.

I won't be teaching the Python programming language in this course. Fortunately, Python has a simple and elegant syntax that makes it very easy to learn. This is one of the reasons for its popularity. In addition, there is a very large collection of Python tutorial available online. I will also provide example codes that could be useful. However, we do not require much of programming. A knowledge of using Python libraries will be sufficient.

The official website for the Python language is `python.org`. To have a look at the Python Language Reference, visit: `https://docs.python.org/3/reference/`

In Python, we use several libraries like NumPy and Pandas. Reference manuals for such libraries are maintained by their respective creators.

2 Python Basics

Execute following examples to get an idea about Python language basics.

Example 1 Illustrating basic syntax of Python

```
# A very basic Python script
# Comment begins with hash

number1  = 23 #Unlike C++/Java, no line end-marker here
number2 = 17

#let's read another number from user
number3 = int(input('Enter an integer '))
# In the above line, we asked the user to enter an integer.
# By default whatever the user enters is string,
# so, we explicityly converted it to integer.
# But we would rarely ask the user to inupt anything in
# data mining modules.
# Now lets find the largest number.
# Python does not use brackets for control structures
# like if-else and function.
# Indentation (tab) plays the role of brackets.
maximum = number1
if number2 > maximum: #note the colon at the end
        maximum = number2
if number3 > maximum:
        maximum = number3

#now print the maximum
print('Maximum number is ', maximum)
```

Example 2 Lists in Python.

Lists in Python are flexible as they can store any data type and manipulated dynamic. Similar to C/C++, list index starts from 0.

```
#define a list and sort it
num1 = [17, 23, 10, 20] #initialized list
num1.sort()
print('Sorted List: ')
print(num1)

#define empty list, read 4 numbers, then sort
num2 = []
for i in range(0,4):
        num2.append(int(input('Enter number: ')))
num2.sort()
print('Sorted List: ')
print(num2)
```

Example 3 This example introduces arrays and NumPy.

Arrays and related functions are implemented in Python's standard library NumPy (Numerical Python). This example illustrates how to import the numpy library and use it. Syntax and usage of any Python library is similar to this example.

NumPy arrays can be manipulated much faster using the functions provided in this library. However, for convenience, we will use loops to manipulate arrays. Similar to C/C++ and Python lists, NumPy arrays also begin with index value 0.

```
#import numpy library into the variabe np
import numpy as np

#declare numpy array of 5 elements
x = np.array([1,23,12,-4,8])
#another numpy array of 5 elements
y = np.array([34,21,200,83,0])


# add two arrays element by element using
# a for-loop
```

```
s = np.empty(shape=(5)) #an array of size 5
for i in range(0,4):
        s[i] = x[i]+y[i]
print(s)

#But as far as possible, don't use loops with
# numpy arrays. There are several built-in functions
# that execute faster
p = np.empty(shape=(5)) # an array of length 5
p = np.add(x,y)
print(p)

#we can declare a 2D arrays as this
A = np.array([[34,43, 45], [9, 3, 1], [12, 47, 8]]) #3 by 3 array
B = np.array([[-1,-4, 56], [213, 2, 6], [19, 10, 1]])
C = np.empty(shape=(3,3)) #an array of 3 rows and 3 cols
C = np.add(A,B)
print(C)
```

Example 4 Data frames in Pandas

Modern programming languages like Python have constructs called data frames, which equivalents of database tables. A data frame is a table with table headings, and different columns my contain data of different types.

The Pandas library for Python provides data frames and various function for their manipulations . This example shows how to load data frame from CSV files, and demonstrates basic plotting function.

```
import pandas as pd

#read data frame from a csv file
df = pd.read_csv('nepalMacro.csv')

#now we will do some plotting
# this requires us to import plotting functions
# from the library matplotlib
#import matplotlib.pyplot as plt

# plot Year in x-axis, all others in the y
df.plot(x='Year')

# plot Year versus Human capital
df.plot(x='Year',y='HumanCapital')

# plot year in the x-axis, popularion and employment in y-axis
df.plot(x='Year',y=['Population','Employment'])
```

Example 5 Cross-tabs using pandas

Cross-tabs are used to infer relationships between categorical variables. In addition, we can assess the relationship between a categorical variable with a numeric, by *discretizing* the latter variable. Observe the following example carefully.

```
import pandas as pd
df = pd.read_csv(''districtLevelData.csv")
table1 = pd.crosstab(index = df['district'],columns=df['Ecological belt'])
med = df['internet'].median()
internet = pd.Series('Low',range(75)) #initialize a series
for i in range(0,75):
        if(df['internet'][i] > med):
                internet[i] = 'High'
table2 = pd.crosstab(index = internet,columns=df['Development Region'])
```

3 Assignments

For these questions, apart from referring to the examples given in this lab sheet, you will also need to search the internet for different functions available in the Pandas and NumPy library.

1. **Understanding skewness**: You must have read about skewness in your statistic courses. Using the data in the file 'DistrictLevelData.csv', answer the following questions.
   (a) What are the mean, median and mode of the numeric variables given?
   (b) What are the skewness parameters for the numeric variables?
   (c) Plot the histograms for each of the numeric variables.
   (d) Based on the skewness parameters and the histogram, which of the variables you think is the most skewed?
   (e) What could be the socio-political reason that some fo the variables are more skewed that the others?
   (f) Is there any way to describe how skewed are the categorical variables in the dataset?
2. **Correlation analysis**: Using the data in the file 'DistrictLevelData.csv', answer the following.
   (a) Observe the correlation between the numeric variable in the given dataset. Which of the variables do you think is linked with internet access?
   (b) Which of the categorical variables is more related to internet access? To answer this, you should first discretize internet access into three levels using the technique suggested in Example 5.
3. **Extra Problem – Statistical significance**: You identified the best correlate of internet access in problem 2, for both numerical and categorical cases. Do you think the relationships you are statistically significant? For the numerical case (problem 2a), you should observe the p-value of the correlation coefficients. For the categorical case (problem 2b), you need to perform the chi-squared tests. These function are available in the scipy package.
4. **Extra Problem – Visualization**: For the numeric variables in the file 'DistrictLevelData.csv', use appropriate Python tools to
   (a) Visualize the mean and spread of the variables using box plots.
   (b) Visualize the correlation between variables using scatter plots.
   (c) Variables do have different range of values; some have higher values, like income, and some have lower, like age. How would you display variables with different range of values in a single box plot?