Welcome to Pandas Python Library for Beginners in Data Science. This is a project-based course which should take approximately 1 hour to finish. Before diving into the project, please take a look at the course objectives and structure:

**Course Objectives**

In this course, we are going to focus on these learning objectives:

1. Three  methods of creating a series.

2. Two methods of creating data frames.

3.  Importing/exporting different types of data files and viewing rows.

4. Get a summary of the data & view column names and data types.

5. Calculate mean & cumulative sum. Determine minimum & maximum values.

6. String operations such as converting to uppercase letters , lowercase letters, swap case, finding the length of a string, splitting strings and detecting unique values.

7. Repeating strings.

By the end of this course, you will master the basics of Pandas**.**

**Course Structure**

This course is divided into 3 parts:

1. Course Overview: This introductory reading material.
2. Pandas Python Library for Beginners in Data Science**:** This is the hands on project that we will work on in Rhyme.
3. Graded Quiz: This is the final assignment that you need to pass in order to finish the course successfully.

**Project Structure**

The hands on project on Pandas Python Library for Beginners in Data Science is divided into the following tasks:

Task 1: Three  methods of creating a series.

Task 2: Two methods of creating data frames.

Task 3:  Importing/exporting different types of data files and viewing rows.

Task 4: Get a summary of the data & view column names and data types.

Task 5: Calculate mean & cumulative sum. Determine minimum & maximum values.

Task 6: String operations such as converting to uppercase letters , lowercase letters, swap case, finding the length of a string, splitting strings and detecting unique values.

Task 7: Repeating strings and conclusion to project.

**Project Summary**

Throughout this guided project on "Pandas Python Library for Beginners in Data Science" , you have learned:

1. Quantitative data is also called numeric data. Qualitative data is also called text data.

**Understanding Data Structures in Pandas**

1. The three methods of creating a pandas series manually are by:

* Passing a numpy array:

i) Create the array using the np.array function from the numpy library.

ii)Create a pandas series using the pd.Series() function and pass the array to it.

* Passing a list:

i) Create a list using square brackets.

ii) Create a pandas series using the pd.Series() function and pass the list to it.

* Passing a dictionary:

i) Create a pandas series using the pd.Series function.

ii) Create a dictionary within the brackets.

3. In Python, a square bracket is used to create lists.

4. Python only recognizes a value as a string if it is within quotation marks. So do not forget to add them.

5. An example syntax for a dictionary is:

s1 = pd.Series(

{ 'A' : 1,

'3' : ' Python Programming ',

})

6. The two ways of creating a pandas data frame manually are:

* Passing an array:

i) Create an empty data frame using pd.DataFrame()

ii) Create an array using the numpy function np.array() or by generating a random array using the numpy

function np.random.randn()

* Passing a dictionary:

i) Create an empty data frame using pd.DataFrame()

ii) Create a dictionary within the brackets. (the syntax has been mentioned above.)

**Importing/Exporting Data**

7. You an read in the following file types using pandas:

* csv file using read\_csv()
* json files using read\_json()
* urls using read\_csv( "your-link-here" )
* xls/xlsx files using pd.read\_excel()

8. Reading data directly from the url works only for websites which store data in a tabular format. So in the case of github, make sure to view the data in ‘raw’ mode and use that link.

9. To view the rows of a data frame use the .head() function. The number you put in the brackets determine how many rows of data should be displayed.

10. You can export your file, using pandas, to the following formats:

* csv file using .to\_csv()
* json file using .to\_json()
* xls/xlsx files using to\_excel()

**Gain Insight into the Data**

11. The first thing a data scientist does is understand the data by generating a summary. There are two ways to get a summary of the dataset:

* Quick summary using the function .info() which displays:

i) Total number of entries

ii) Total number of columns

iii) Non-null count per column

iv) Data type

v) Memory usage

* Detailed summary using the function .describe() which displays:

i) Count which is the  number of non-NA/null observations.

ii) Mean of all the values in each column.

iii) Std which is the standard deviation of all the observations in each column.

iv) Minimum value of each column

v) 25th quartile

vi) 50th quartile

vii) 75 quartile

viii) Maximum value of each column

12. The summary information helps us to determine how to handle the data. For example if the dataset has outliers, we will need to minimize them.

13. We use the pandas function .columns() to see the names of the columns. Usually this step helps us identify features that will be useful for our analysis.

14. We determine the datatype of the values using the .dtypes function. This helps us with transformation of data from one type to another which is more appropriate for our analysis.

**Data Preprocessing/Cleaning:**

15. Sometimes a dataset might have duplicate values. This can interfere with our analysis. So we handle them by:

* Detecting duplicates using the function .duplicated()
* Dropping the duplicates using the function .drop\_duplicates()

16. During data cleaning, we might find some missing values. The easiest way to handle them is by:

* Detecting missing values using the function .isnull().
* A more efficient way to print just the number of missing values per column is to print only the sum of missing values using the .sum()
* Then dropping the rows which contain the missing vales.

17. While dropping the entire row with the missing value is the easiest method to handle missing values, it is not always the best method. This is because you are loosing a lot of valuable data by dropping the entire row. So be sure to only drop rows with missing data when it does not have a negative impact on the analysis.

18. We calculate the mean of an entire column using the aptly named function .mean()

19. Calculate the cumulative sum of a data frame by:

* Using the pandas function .apply()
* Then, within the brackets, using the np.cumsum

20. Find the maximum value of each column using the function .max()

21. Find the minimum value of each column using the function .min()

22. To convert a string to lowercase use the pandas function .str.lower()

23. To convert a string to uppercase use the pandas function .str.upper()

24. Nan is the abbreviation for Not a Number.

25. You can swap the capitalization of the letters in a string. This means all capital letters will become lower case and vice versa. The function to do this is str.swapcase(). Note that even the letters in the word NaN are swapped.

26. To find the number of characters in a string we use the function .str.len(). Remember that spaces are counted as individual characters.

27. To split a string, which is a sentence, into words we use the function .str.split()

28. The function .unique() prints the unique values in the series in the form of an array.

29. To repeat a string by a specified number of times:

* First create a list with the number of times you want each string to be repeated.
* Then pass this list to the function str.repeat()