# INFO 208 Quiz #2 Lab Exploration Component

Your Name: Venkata Anil Kumar Thota

The following Lab items should be carried out by you using your regular Operating System (Windows, macOS, or perhaps Linux) and a browser such as Chrome, Firefox, or Safari.

We will **not** be using your **Hortonworks HDP Sandbox** VM for this Quiz. But, remember, you were asked to keep your HDP Sandbox VM for later use — both in this course and for continuing education beyond this course. You will probably be using the Hortonworks HDP Sandbox for the Final Exam Practical

The purpose of this **Lab Exploration Component of Quiz #2** is to give you more practice with Python 3 & Jupyter and then to ***introduce*** you to the Julia language — remember that this is one of the languages as the basis for Jupyter.

**Open up the area after each question / section** in this MS Word file and type in (or paste) your answer(s) and your comments. Mail the resulting file to me at [glen.mules@sjsu.edu](mailto:glen.mules@sjsu.edu)

Your email **Subject** line should be:

**INFO 208 Quiz #2**

And the attached MS Word file sent by email should be named:

**INFO\_208\_Quiz2-*YourLastName*.docx**

The file naming is important as it causes my email software to sort your *Quiz Response Email* into a directory for me so that I don’t lose amongst my other daily email.

You will be able to upload your Practical / Lab Exploration Component into Quiz #2 on Canvas as the last item, if you have time.

**First Lab.**  **Using an existing Python 3 Jupyter Notebook.** You are not expected to be an expert in Python 3, but, hopefully, you have an introductory background.

**Steps:**

\_\_1. Go to the Gallery of interesting Jupyter Notebooks at <https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks#programming-and-computer-science> and select **two notebooks** which ***do not have*** “Introductory” in their title or description. Your choices should be beyond the mere introductory level.

My suggestion is that you select your notebooks from an area of your own personal experience, your undergraduate major or minor, or your work/business background, e.g.:

* Programming and Computer Science
* Statistics, Machine Learning and Data Science
* Mathematics, Physics, Chemistry, Biology
* Earth Science and Geo-Spatial data
* Social data
* Psychology and Neuroscience
* Machine Learning, Statistics and Probability
* Physics, Chemistry and Biology
* Economics and Finance

List here your chosen notebooks:

* py\_exploratory\_comp\_2\_sol.ipynb (http://nbviewer.jupyter.org/github/mbakker7/exploratory\_computing\_with\_python/blob/master/notebook2\_arrays/py\_exploratory\_comp\_2\_sol.ipynb)
* ETL\_with\_Python.ipynb (https://github.com/dimgold/ETL\_with\_Python/blob/master/ETL\_with\_Python.ipynb)

2. Work your way through these two Python-based Notebooks and cut-and-paste some snippets here to illustrate that you worked them and achieved completion of the work in the notebooks themselves.

**First NoteBook(**py\_exploratory\_comp\_2\_sol.ipynb):

Exploratory Computing with Python: -

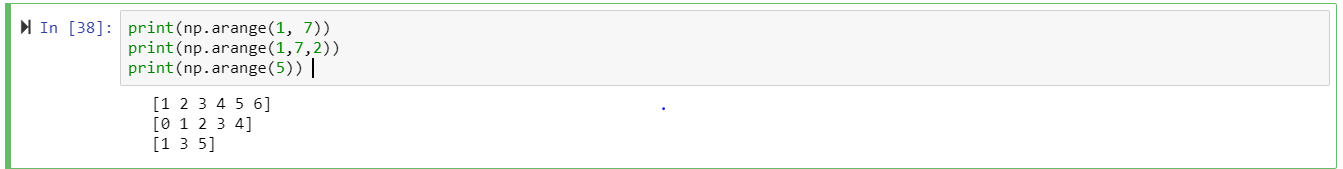
* Imported numpy package as “np” and matplotlib plotting package as “plt”.



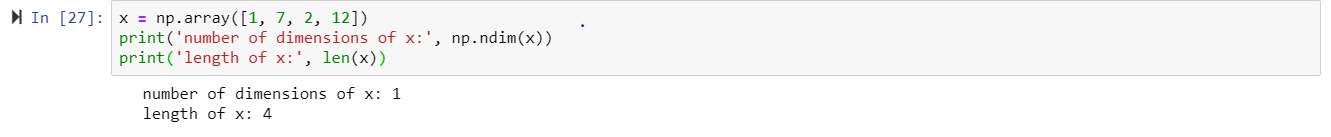
* Created array using array function available in numpy Package. It also provides upcasting option which is an added advantage.



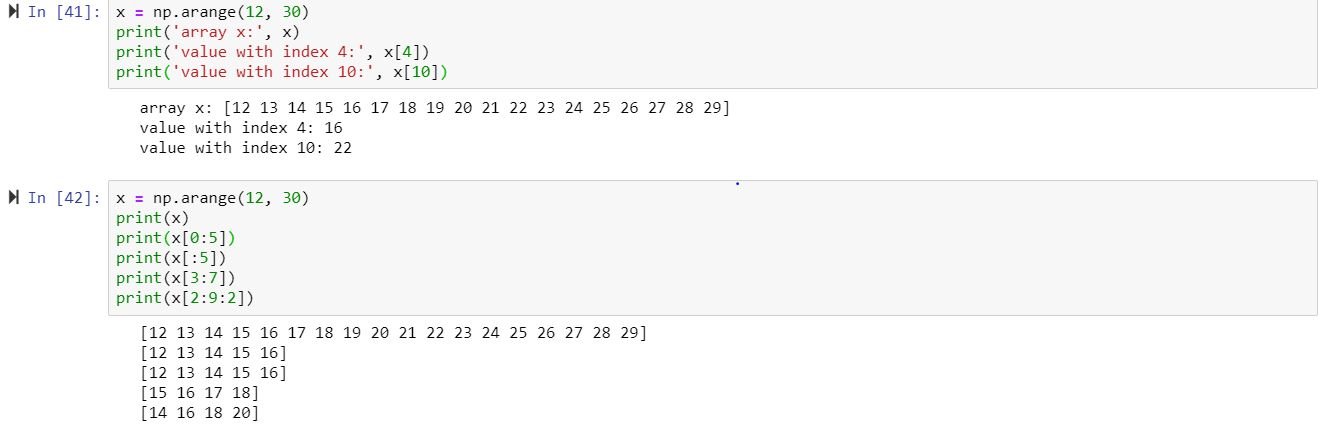
* Explored about “arange” function and tried different ways of using it. In the below Screenshot function with two arguments, print the values in the form of array, the first value is the first argument given in the function and last value of the array is the value excluding the second argument (second argument -1). If we add one more argument it is the representation of taking the step before it reaches end. Specifying one argument(n), it displays array values starting from 0 to n-1.

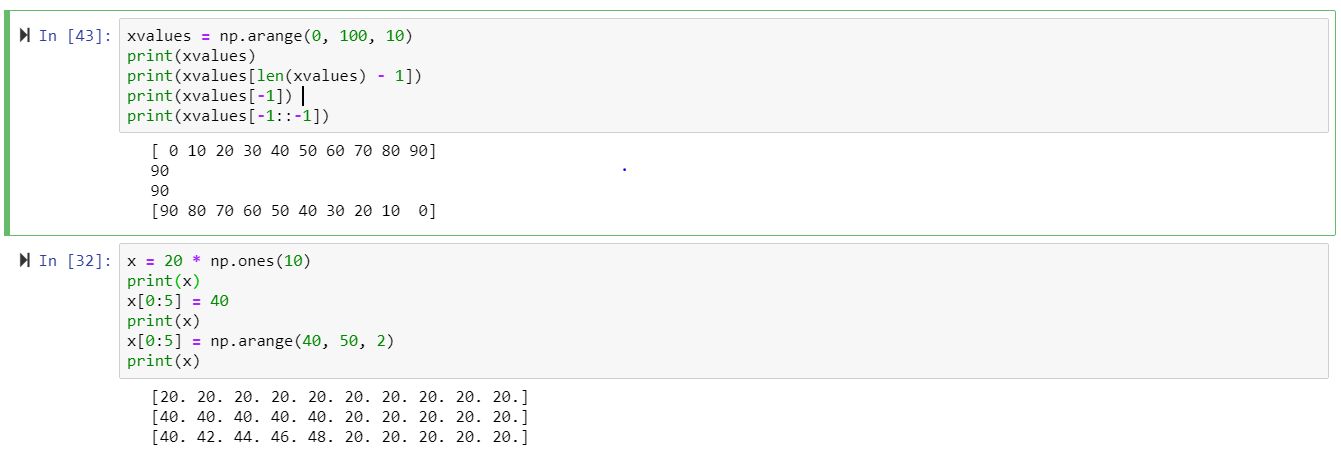


* “ndim” function gives the dimension of the array, “len” function displays the length of the given array.



* Several ways of retrieving values from the array by giving index as the input to the function.

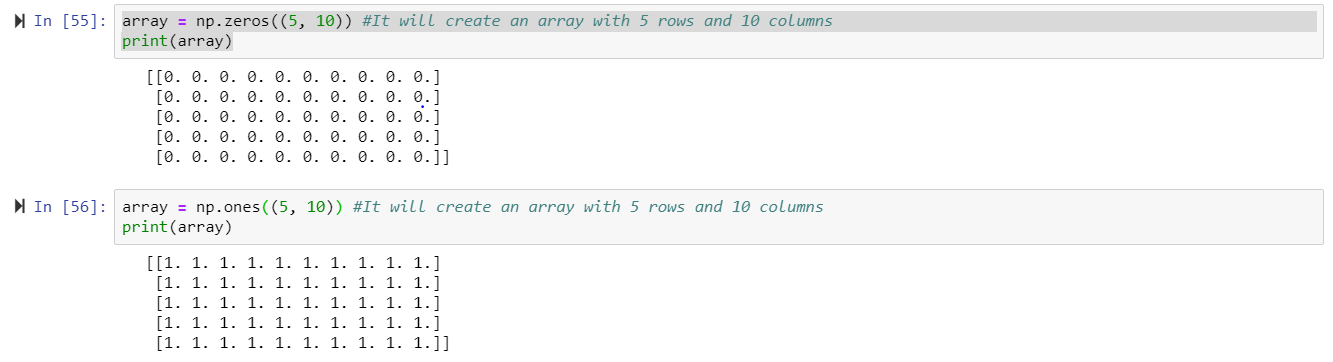




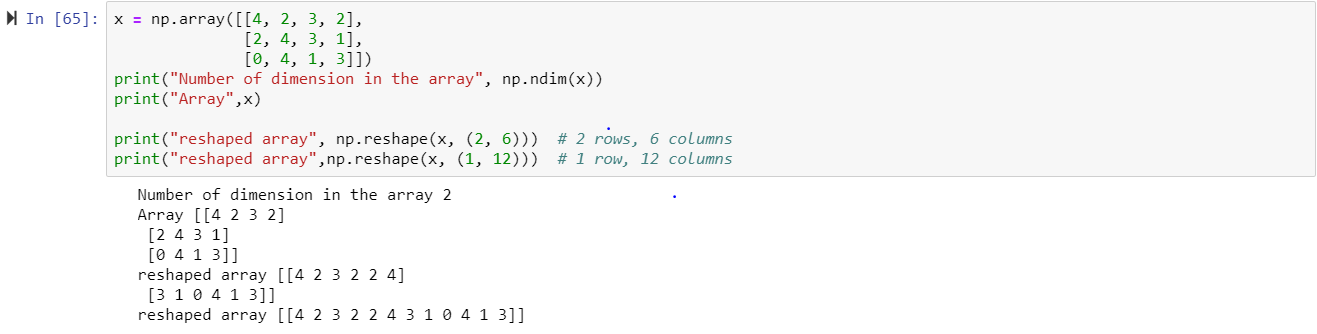
* Created list and tuple datatypes, and tried several operations using them. We can modify list whereas tuple once defined cannot be modified.



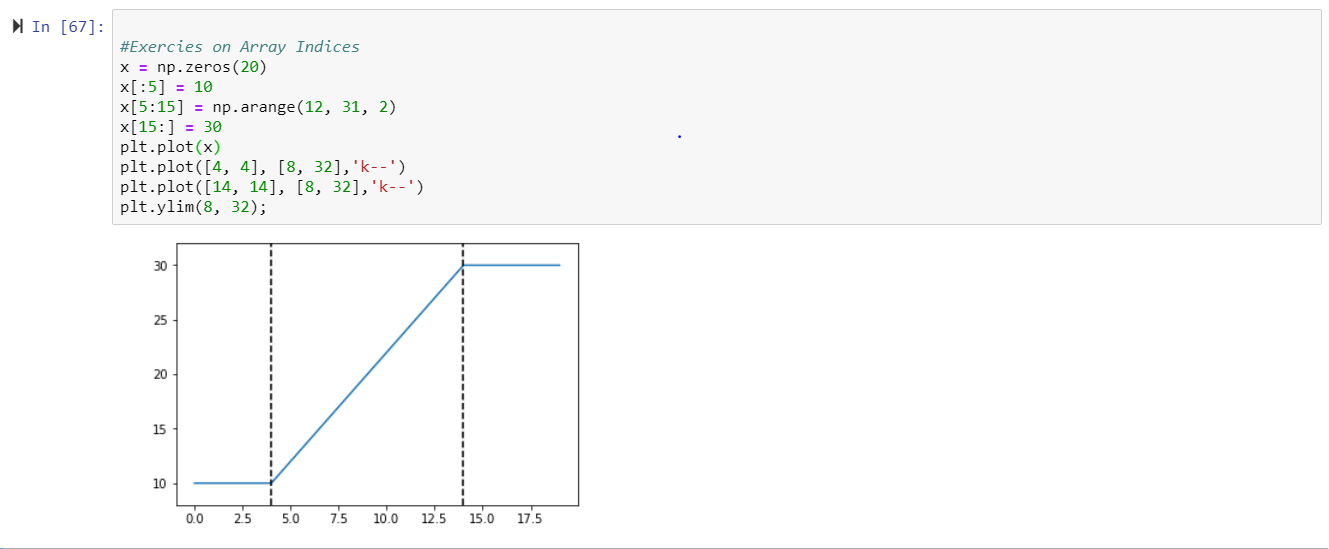
* Numpy package has special function , so that the array will display all the values with zero and one.

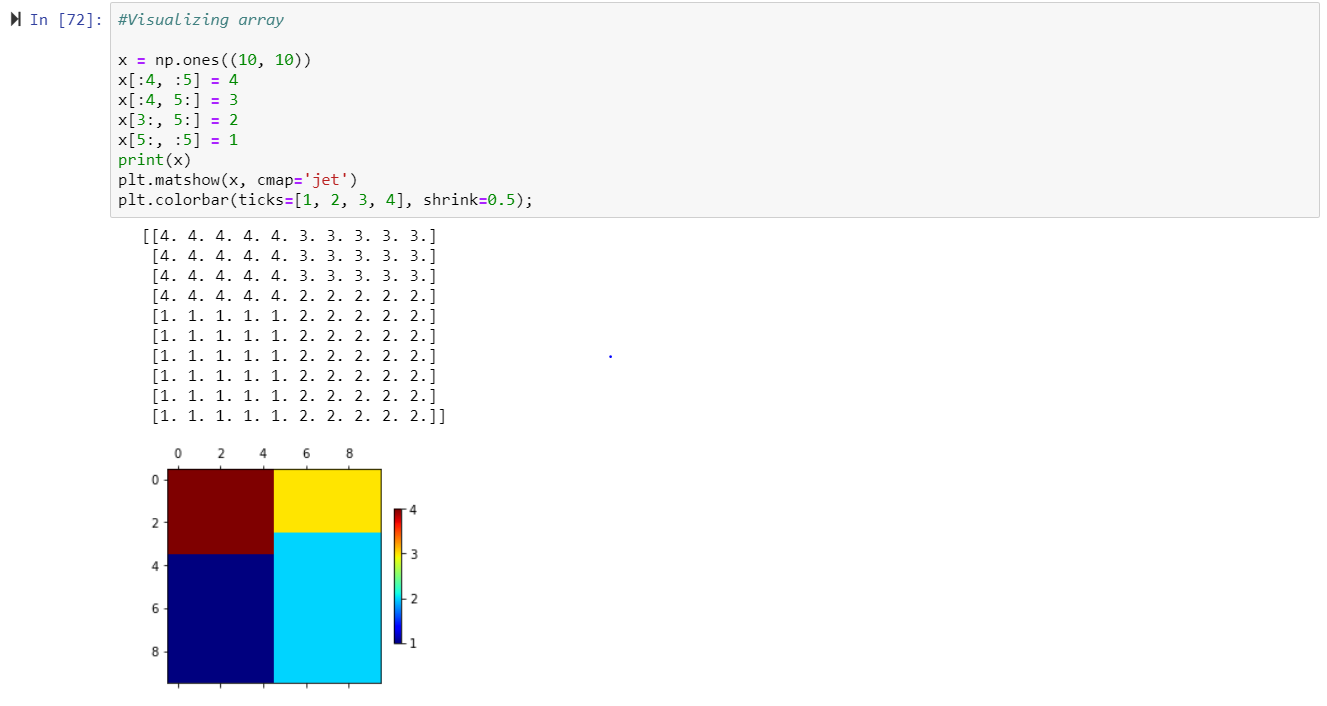


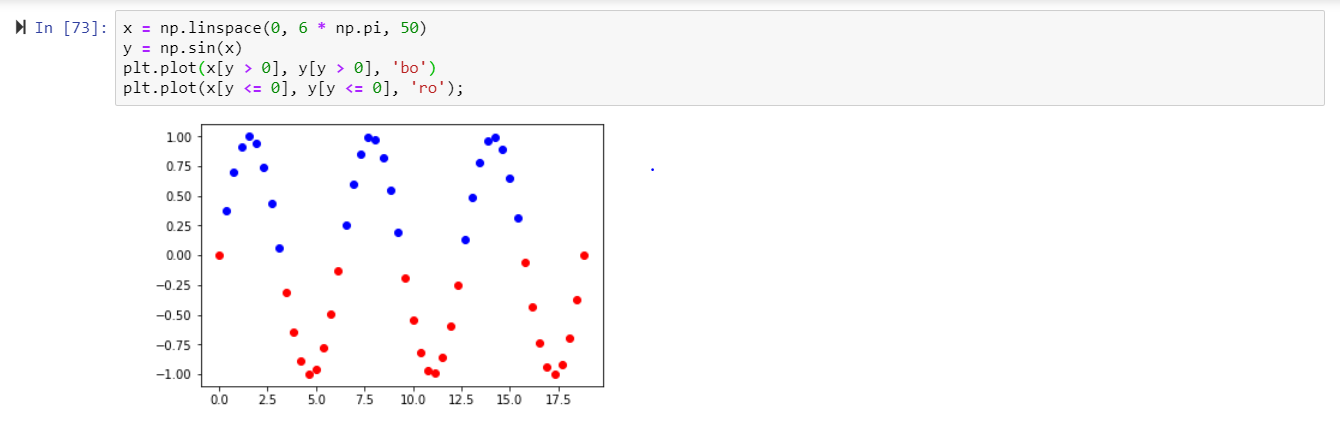
* Created two-dimensional array. We can also reshape the array using the “Reshape” function.

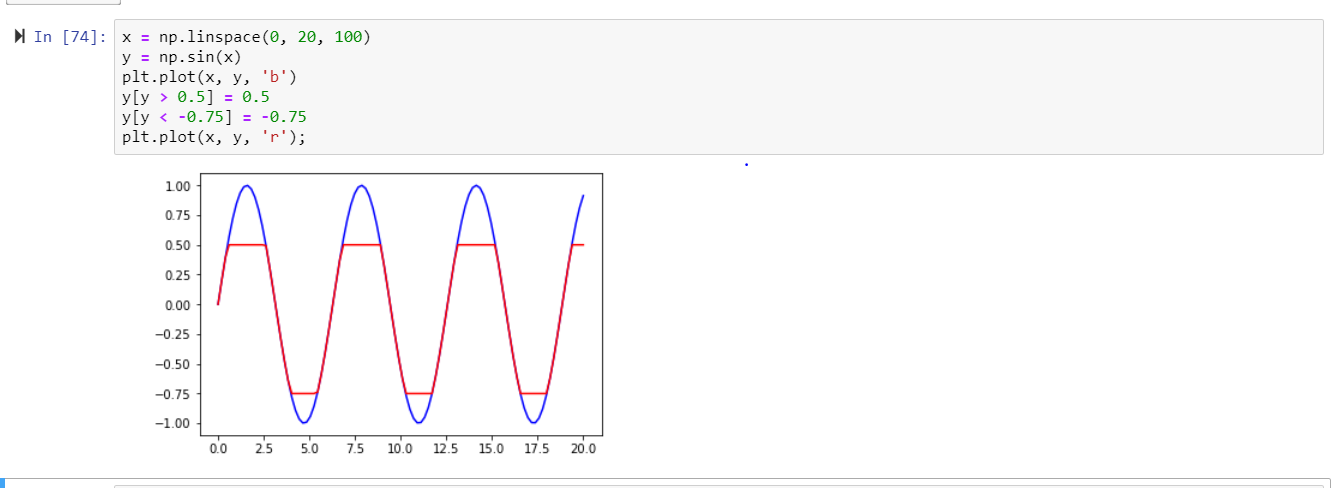


* Completed several exercises on Arrays, indices, visualizing 2-dimensional arrays,conditions on arrays. Below are the screenshots.





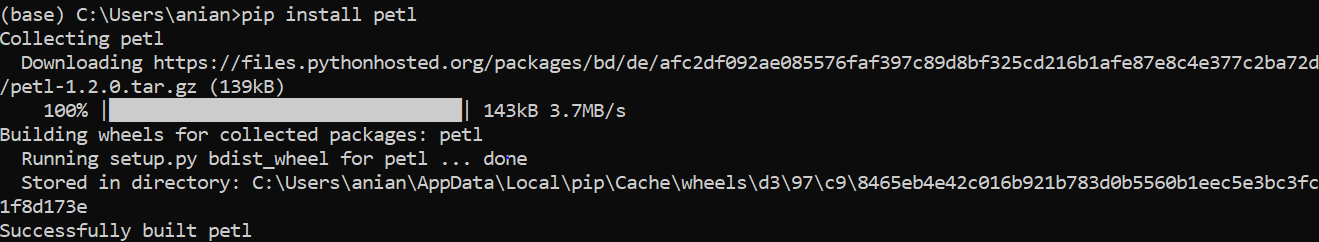




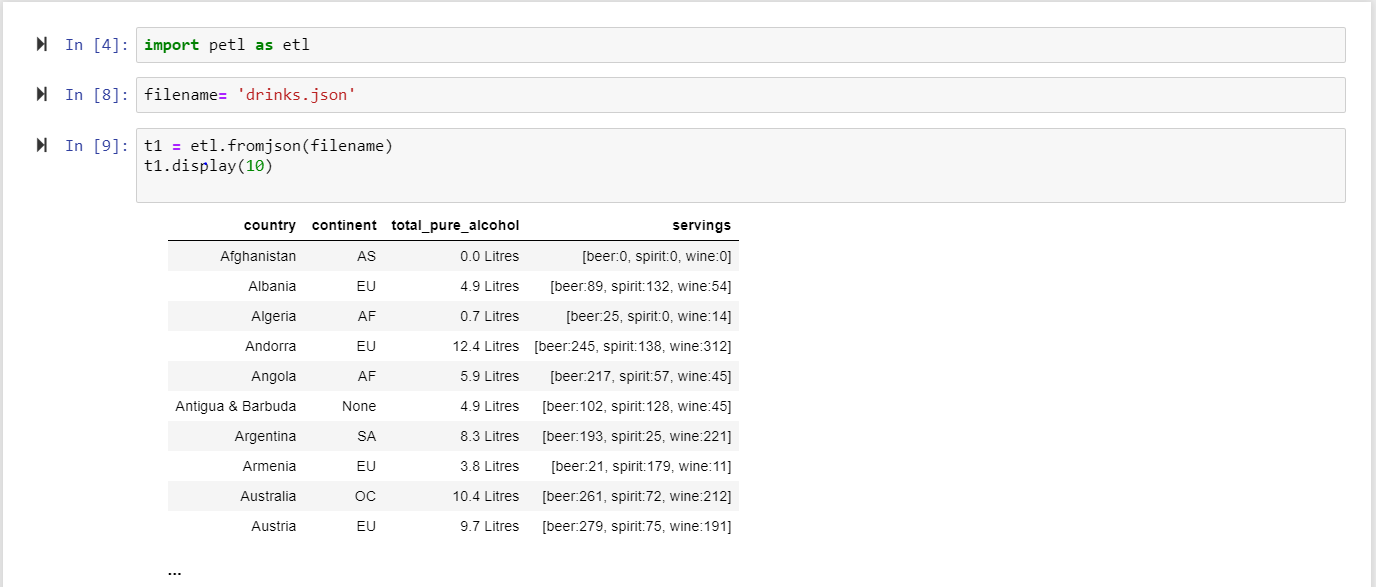
**Second NoteBook:**

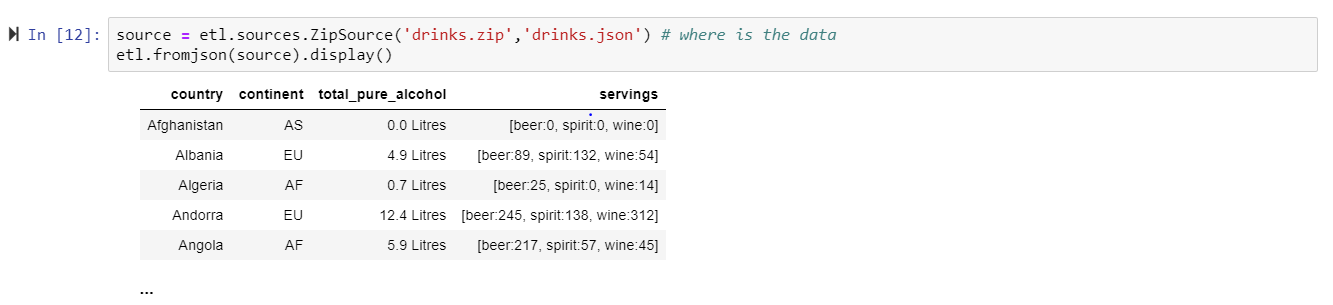
ETL with Python(ETL\_with\_Python.ipynb): -

* Installed petl package for running for performing ETL process on the Data.

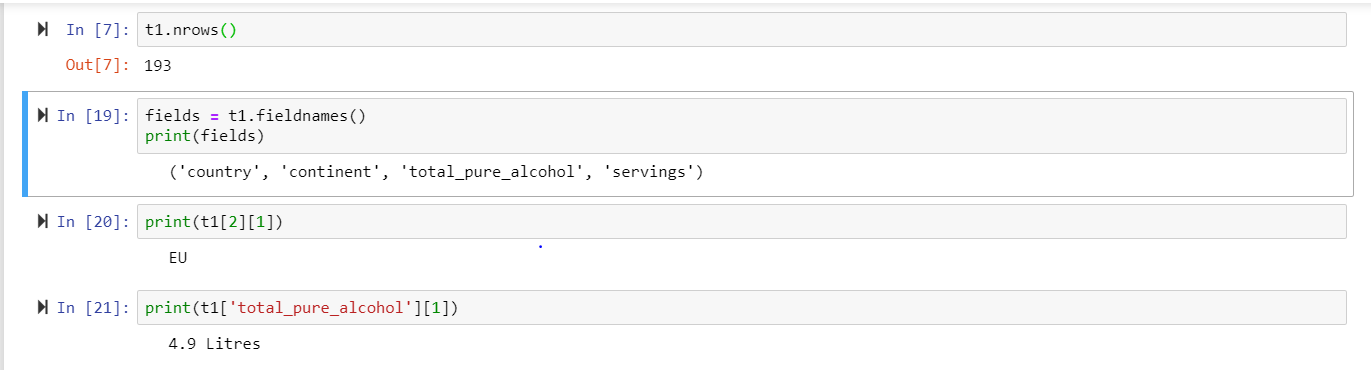


* Imported data filesusing “petl” which are in form of Json and zip file.

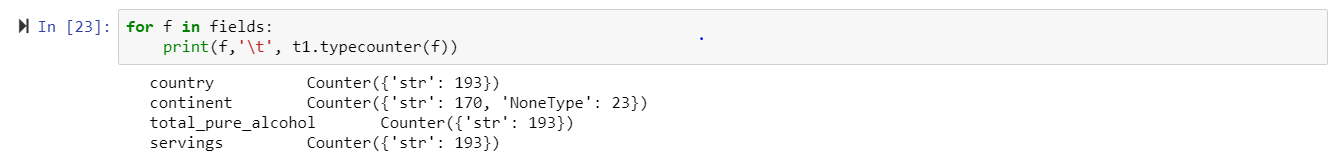




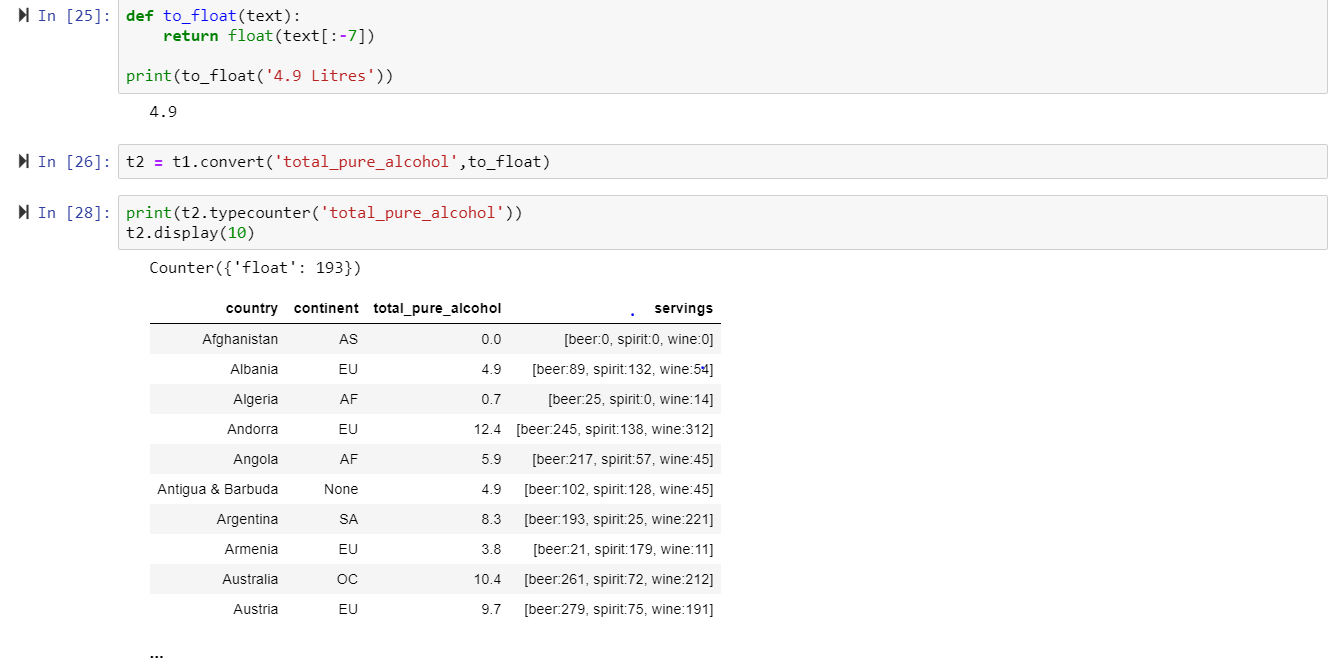
* Exploring Data by finding Number of rows available, Fieldnames, Checking Specific Values etc…



* Using the typecounter() finding the Datatype of the fields and Null values in the Columns. Using this information, we will start our Transformation Phase, in which we need to do some operations on Data.



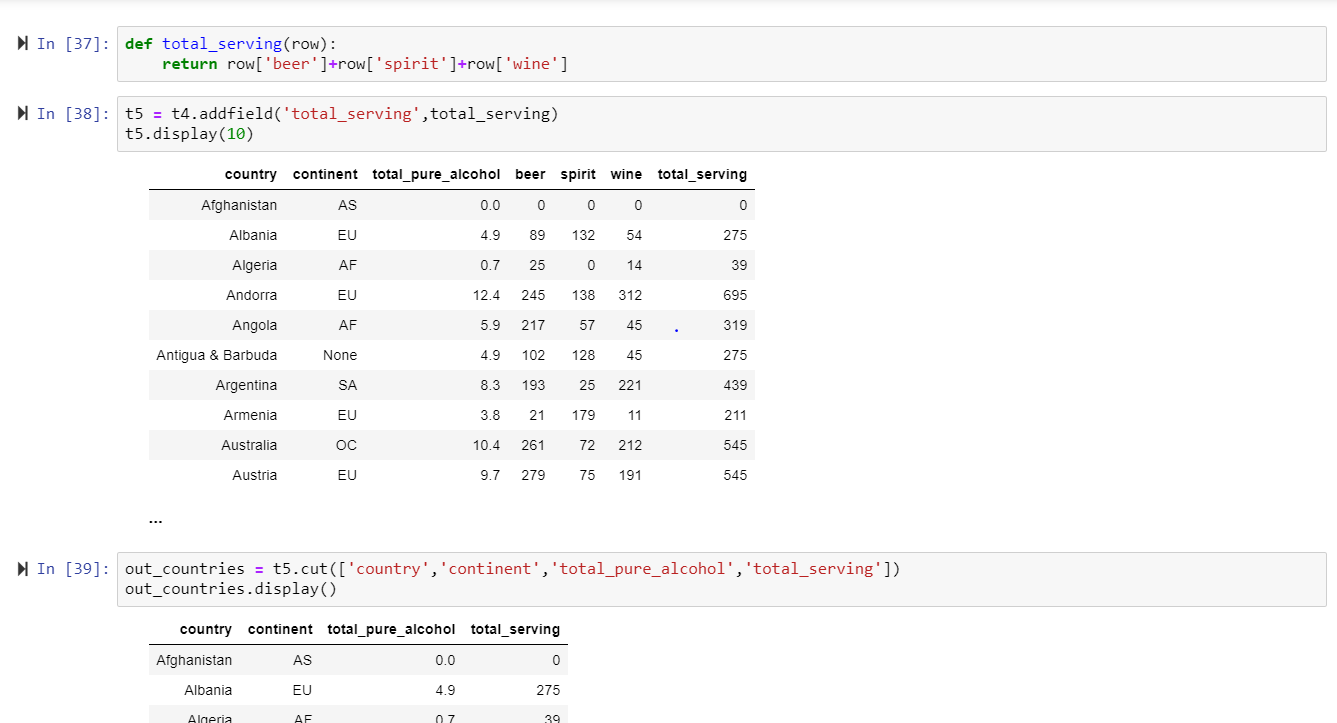
* Converting column of data type “String” to “Float” by using Functions.



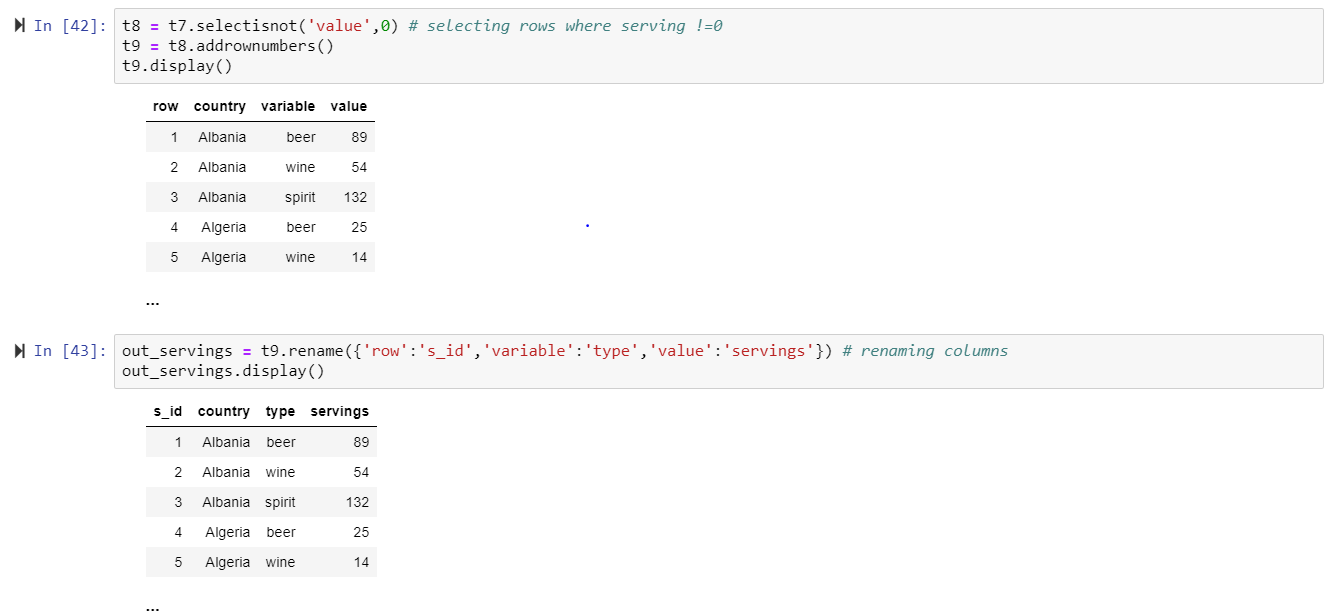
* Converting servings column from “String” to “Dictionary” and unpacking servings column in the way that “beer” “spirit” “wine” will be displayed as separate column’s.

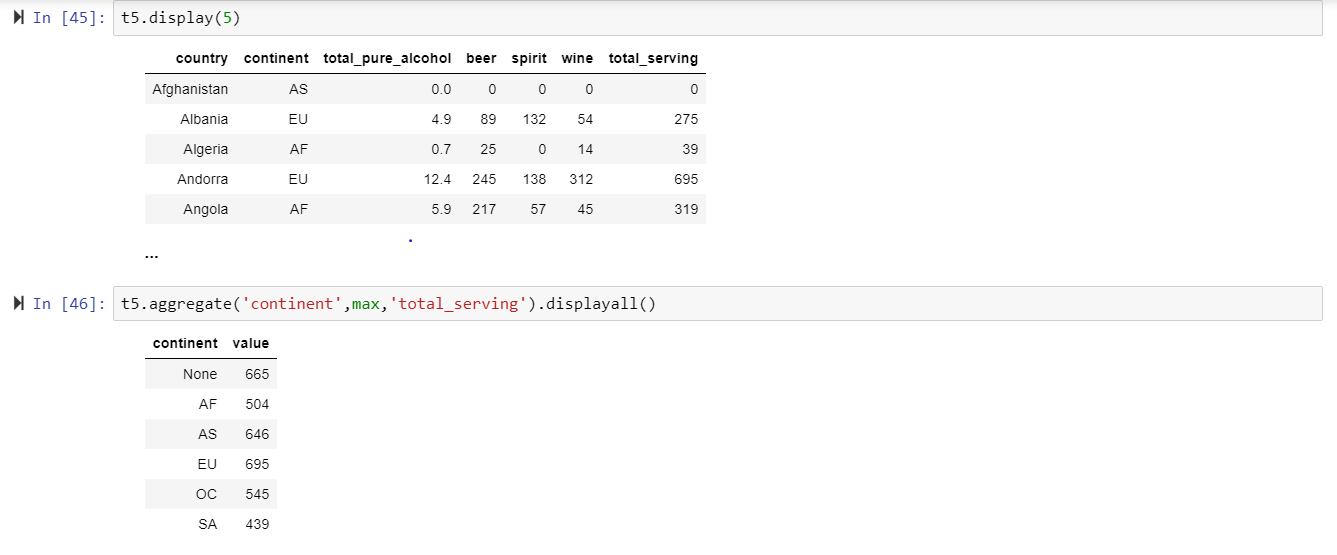


* Added additional column to display the total serving. Performed different operations like slicing needed columns, removing Null values ,adding Row Numbers so the Data can be in the convenient format for the End-user.



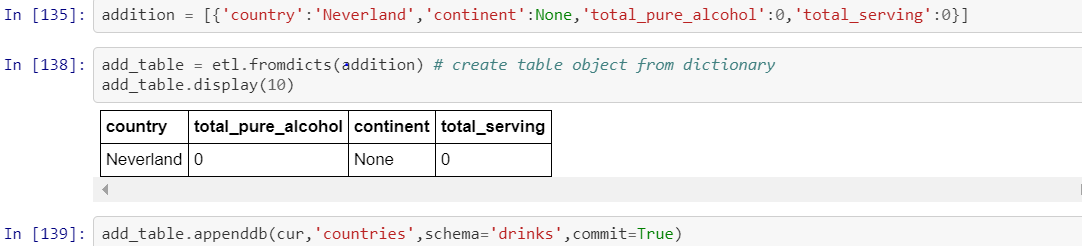






* Final stage is the Loading stage. In this I created schema and added data to the available tables so that the data can be found in Mysql Server.





3. Comment here on **what** you learned both about the topic chosen and the Jupyter Notebook approach to documenting work (***at least a half dozen sentences***):

**First NoteBook(**py\_exploratory\_comp\_2\_sol.ipynb):

Exploratory Computing with Python: -

* Numpy stands for “Numerical Python”. It is a library which contains multidimensional array objects and a collection of functions to process those arrays. Numpy array is a grid of values of the same type and is indexed by a tuple.Numpy is used generally with Scipy and Matplotlib packages. Matplotlib is a plotting library for python. Pyplot() is used to plot 2D data. We can also create subplots and images using Matplotlib. Clearly understood the difference between array and matrices which can be created by Numpy. Array objects can have more than 2 dimensions whereas matrix will have exactly two dimensions. We should be careful while performing operations on array and matrix for instance, in array \* refers to element wise multiplication and matrix refers to matrix multiplication. By completing this notebook got a picture of difference between list and tuple. One of the main characteristics of a list is that it is ordered. The order of the elements in a list is an intrinsic property of that list and does not change, unless the list itself is modified. The same is true of tuples, except they can’t be modified.
* Jupyter Notebook denote documents that contain both code and rich text elements, such as figures, links, equations, because of the mix of code and text elements, these documents are the ideal place to bring together an analysis description and its results as well as they can be executed perform the data analysis in real time.
* Jupyter Notebook allows to visualize data at any point in the notebook. This is extremely useful because we can view the state of the as each step of your logic executes. This capability reinforces the use of Jupyter Notebook in a prototyping workflow when we are attempting to confirm that your workflow is doing what it needs to do at each step of the way. Hence, Jupyter Notebook provides most convenient way for implementing array using Numpy.

**Second NoteBook:**

ETL with Python(ETL\_with\_Python.ipynb): -

* ETL stands for Extract Transform Load. It is a tool that extract data from the source mainly Relational Database systems, transform data by applying some techniques and loads the data in to the Data warehouse. Important thing which is to be highlighted is the data loaded in the DW in form of Fact tables and Dimension. The First step in ETL is to pull out the data from the multiple desired sources and make it available for further Processing. During Extraction it is difficult to decide what data is required for the User hence more Data is extracted than needed. Transformation stage includes transforming data into suitable format that can be easily loaded into a Data Warehouse System. Transformation involves calculations, joins and defining primary and foreign keys on the Data. It also includes data correction, cleansing and removing incorrect data. The last step in the process is loading cleansed and transformed data for further analysis.
* For performing above operations there are few tools called as ETL tools. For example, PowerCenter Informatica, Cognos Data Manager, CloverETL, Open Text Integration Center.
* Petl package is designed for convenience and ease of use, especially when working interactively with data that are unfamiliar, heterogeneous and/or of mixed quality.
* petl transformation pipelines make minimal use of system memory and can scale to millions of rows if speed is not a priority. For very large datasets packages like pandas,pytables.
* Implemented ETL operations on drinks dataset in my local machine. Imported datasets from local machine, which comes under extracting phase in ETL. Explored dataset using different commands so that it will be easy while transforming Data. In the transformation Phase, performed different operations on the data, removed data errors, joining data from two different data sources, aggregating, sorting and many advanced validation rules.
* As a server-client application, the Jupyter Notebook App allows to edit and run notebooks via a web browser. The application can be executed on a PC without Internet access or it can be installed on a remote server, where it can access it through the Internet. Its two main components are the kernels and a dashboard.
* A kernel is a program that runs and introspects the user’s code. The Jupyter Notebook App has a kernel for Python code, but there are also kernels available for other programming languages.
* The dashboard of the application not only shows the notebook documents that we have made and can reopen but can also be used to manage the kernels, and check which ones are running and shut them down if necessary.

**Second Lab.** No prior knowledge of programming in Julia is expected. The intent is to introduce you to the Julia language.

Some features of Julia:

* Language syntax is similar to Python.
* But whereas Python is interpreted, Julia is compiled down to CPU executable statements that run often as fast as C / C++.
* “Looks like Python, feels like Lisp, runs like C”

**Steps for the Second Lab:**

\_\_1. Do a Google search on the topic “Julia Language” and read several of the articles including the Wikipedia article.

List here which articles you read:

* https://julialang.org/learning/
* https://www.juliabloggers.com/
* http://faculty.uml.edu/hung\_phan/others/ntjulia.pdf

\_\_2. Look through the accompanying file **Intro\_Julia\_tutorial\_slides\_ODSC\_West.pdf** to get an overview of the main reasons for another computer language and why Julia could be that language for you.

\_\_3. Bookmark the following pages for use as references:

* Introducing Julia  
  https://en.wikibooks.org/wiki/Introducing\_Julia
* Introducing Julia/print  
  <https://en.wikibooks.org/wiki/Introducing_Julia/print#Introduction>

\_\_4. For this lab you will be working with Julia notebooks *remotely* using a browser. You do **not** need to install Julia on your local system (**nor** in your Sandbox VM).

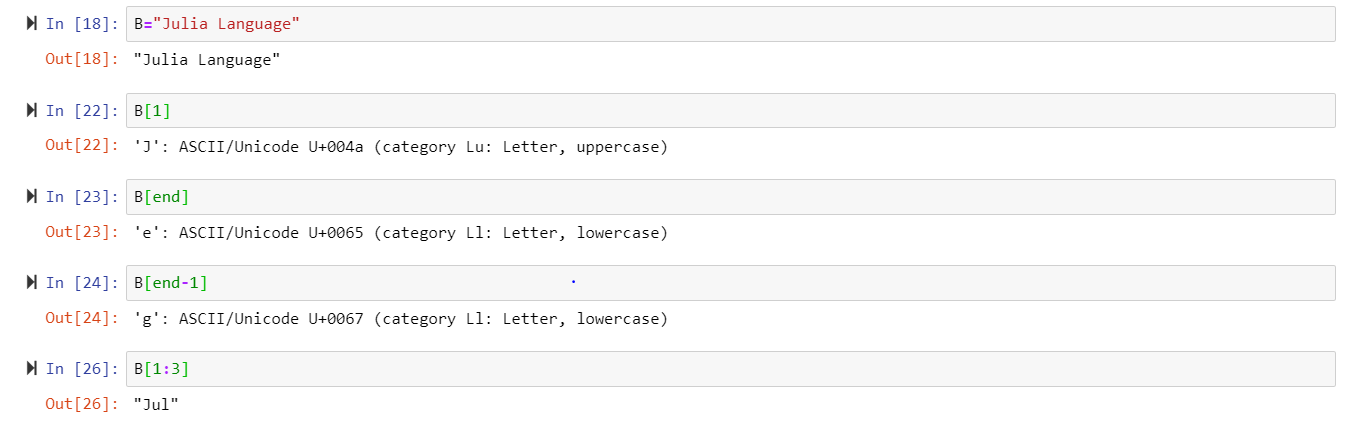
Create an account at JuliaBox (<https://juliabox.com/>) using your sjsu.edu email account and then work your way through *several* of the “Getting started” Julia Notebooks: e.g., 00, 01, …, 09, …

\_\_5. Document here your work (using cut & paste or similar): e.g., 😺 + 😞 == 😀  
 *Expand this space as necessary*.

* Completed 10 Tutorials in juliabox. Apart from above mentioned reference I also went through YouTube video tutorials of Julia. Below I attached some screenshots and my understanding about Julia and its importance.
* Println is used to display output. It is like printf function in C language. Assigned different values and text to different variables. Using typeof() found the datatype of the variable , it also displays bits in the processor which refers to the size of operating System.



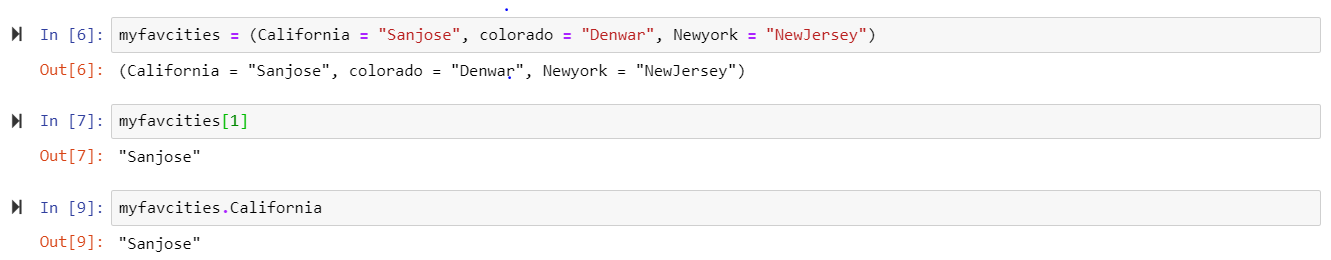
* String is stored as a sequence of characters in the memory, in Julia index starts with “1”. In no other language index will start from “1”.



* Created Tuple data structure and retrieved value from the Tuple.



* Worked with Named Tuples.



* Created Dictionaries, retrieved Values from the dictionary and used pop function to delete the Values.

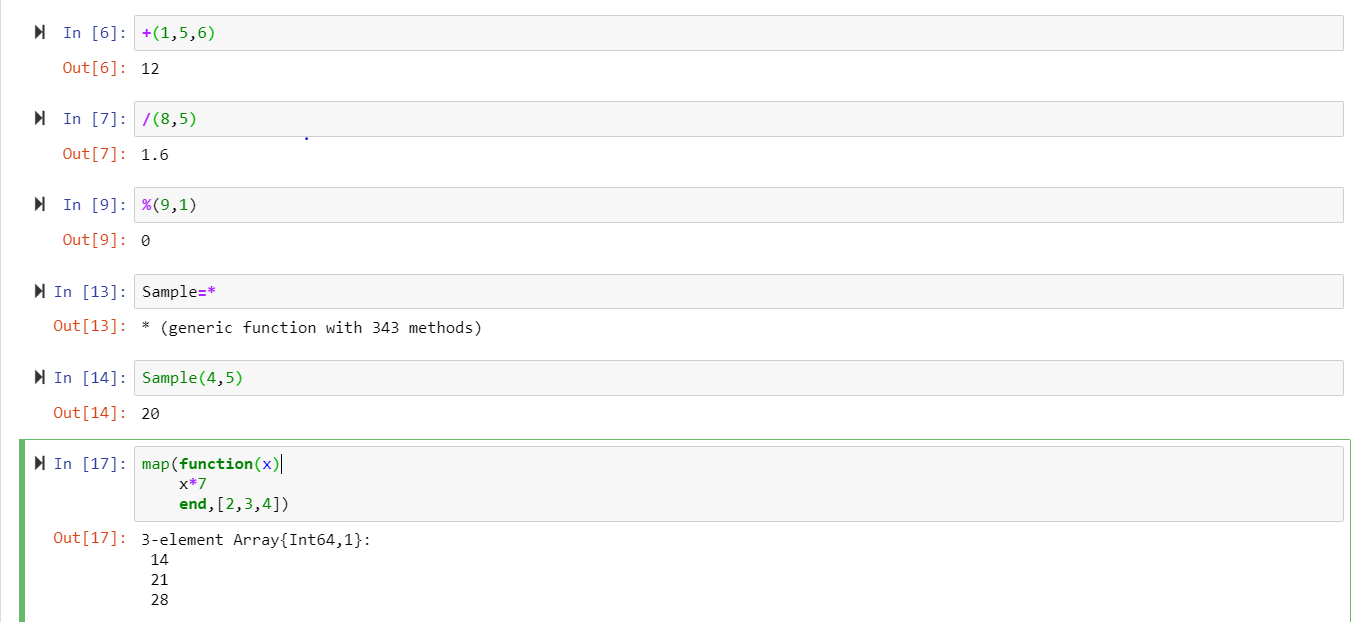


* Explored While loop and for loop and used them in different Cases.

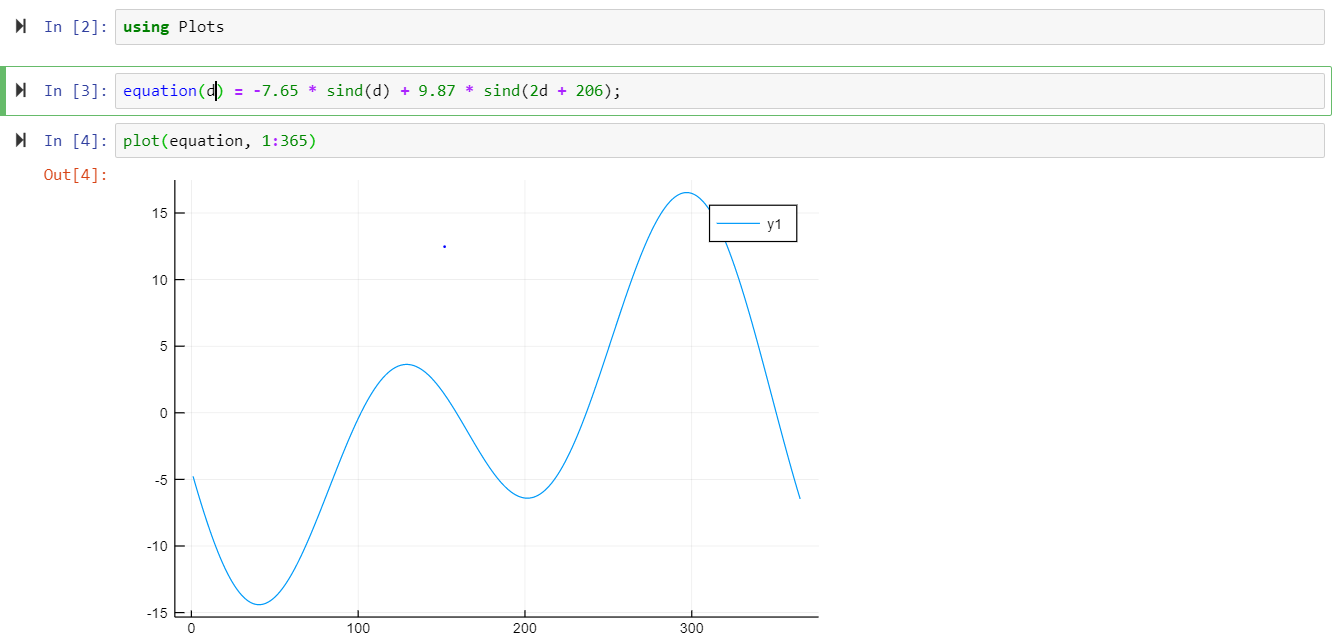




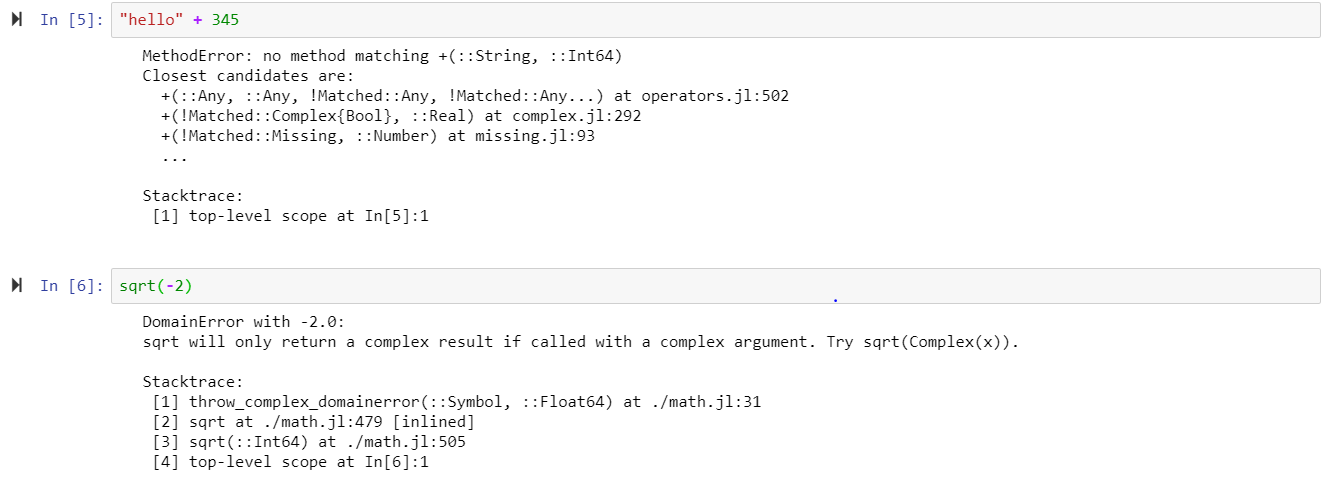
* Operators are functions in Julia. Functions are first class objects in Julia which means you can assign function to a variable. We can also function as an argument to another Function.

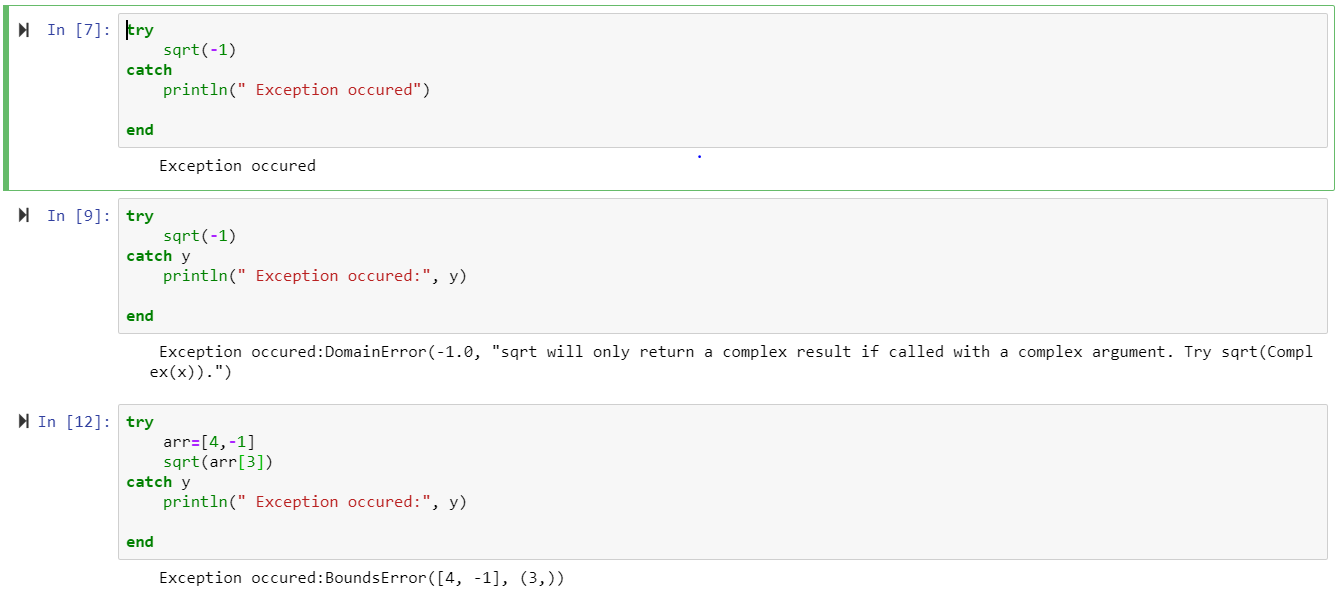


* Reviewed about the usage of Julia in plotting the functions.

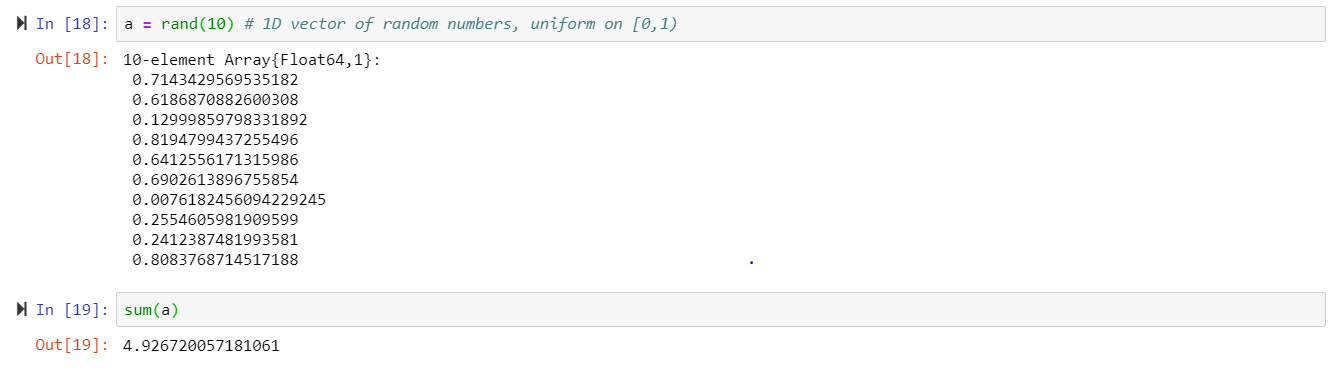


* Exceptions are errors detected at the time of program execution. Exceptions terminate the execution of the code. The code that you might suspect might throw an error we should include in try catch block. Tried different type of exceptions.





* Generate some random numbers and found the sum of those using Julia function and compared it with “C” function. Here I wrote Julia function that calls “C” function.





6.Comment here on **what** you learned both about the Julia language and the Jupyter Notebook approach to working with Julia (***at least a half dozen sentences***):

* Julia is a high-level, high- performance dynamic programming language for Computing. Julia is Just-In-Time compiled using the LLVM compiler framework. At its best, Julia can approach the speed of Julia. Julia syntax is similar to python, but it is more expressive and powerful. Julia programs can generate other Julia programs, and even modify their own code which also termed as Metaprogramming. The most amazing feature of Julia is its parallelism. In Julia, one can parallelize directly from command line by calling the desired script with a given number of cores. Additionally, it is possible to send tasks to different threads directly from code. When you call a function in Julia it will look up in a table at runtime which concrete function it should call based on the types of its arguments. An additional feature of Julia is speed. Julia complies codes on the fly, reaching an incredible velocity. Julia combines the method of interpreting languages, such as Python, that transform the code into bytecode on the fly, with compiled languages like C, that compile their code into machine code and then running the resulted executable. While Julia offers awesome features, it also comes with some pitfalls. First, Julia hashes its dictionaries differently to Python, which currently can prove slower. Second, while not necessary a disadvantage, in Julia Language arrays are 1-indexed, along with Fortran array ordering. Thus, depending on the code one uses, it might not be out of the box compatible with Python or C.
* Julia's got many benefits, though it's new. I believe that in the next few years scientists, research facilities, mathematicians, physicists and anyone who does scientific calculations will move to Julia. For Julia we need a platform for running Julia code along with figures and graphics. Jupyter is the best tool for this purpose because it will provide neatly formatted tables and figures. Notebook has the ability to re-run individual snippets, in a notebook you can edit those snippets before re-running them. Hence, notebook is the best approach for Julia.

**I hope that you were able to add a smile to your life with Julia:** 😺 + 😞 **==** 😀