Learning Objectives

In the second module of this course, you will learn how to import **Data Frames** and select information from a **Data Frame**.

Learning Objectives:

1. Importing **csv** files and loading a **Data Frame**
2. Visualizing a **Data Frame**

Modules In Python

We've imported the built-in statistics package, then used that package to perform some basic descriptive statistics. There are many other modules and packages we can use.

To clarify, a package is a collection of Python modules, while a module is a single Python file.

For the purposes of data science, a module can contain executable statements as well as function definitions. These statements are intended to initialize the module. They are executed only the first time the module name is encountered in an import statement.

Modules we will use as we tackle data science:
* **Statistics**
* **Matplotlib**
* **Pandas**

It is customary but not required to place all import statements at the beginning of a module. This way if you or someone else is reading your code they can instantly know which modules will be used.

```
import matplotlib
import statistics
import pandas
```

Some of the module names, tend to be long. In order to make our lives easier, Python lets us shorten the module name when referring to the module later in our code. For example, below you can see that we use the key word `as` when importing `statistics` in order to shorten it to `sta`

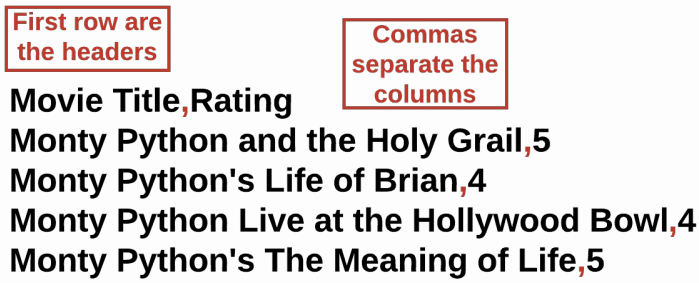
```
import matplotlib
import statistics as sta
import pandas as pd

x=[1,2,3,4,5,6,4]
print(sta.mode(x))
print(sta.mean(x))
print(sta.median(x))
```

CSV and Data Frames

CSV Files

Python can work with files types beyond text files. Comma Separated Value (CSV) files are an example of a commonly used file format for storing data. CSV files are similar to a spreadsheet in that data is stored in rows and columns. Each row of data is on a separate line in the file, and commas are used to indicate a new column. Here is an example of a CSV file.



images/monty-python-csv

There are many modules that can help with reading CSV. The Pandas module mentioned on our last page helps with the presentation of our CSV files. The Pandas module lets us access, read, modify and search data. We can use the code below to load and display our data frame.

A ***Data Frame*** is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it as a spreadsheet or SQL table. It is generally the most commonly used pandas object. A CSV file is returned as a two-dimensional data structure with labeled axes, which allows us to interpret it as a data frame.

Use the following to read a CSV file.

```
import pandas as pd
import matplotlib
#loading Data Frame
data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')

#Display our data frame
print(data)
```

A CSV can be represented as a set of lists. Below we have a list of lists stored under the variable `People_Info`. Then we use the pandas library to create a Data Frame where as argument we use the `People_Info` and use a new list `columns` to provide the column names. Replace the code in the window with the code below.

```
import pandas as pd
import matplotlib

People_Info = [['Jon','squarepants',21],['bobby','Brown',23],
        ['boby','Lee',42],['Jill','Star',28],
        ['SquiQ','tentacles',32]]


df = pd.DataFrame (People_Info,columns=
        ['First_Name','Last_Name','Age'])
print (df)
```

Analyzing our Data Frame

In order to get a brief overview, of our data we use the `.info` method. This method prints information about a Data Frame including the index `data type` and column, non-null values and memory usage.

```
import pandas as pd
import matplotlib
#loading Data Frame
data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')

#Display our data frame
print(data)
print() #this is to add empty line so our results are more
        legible
#Displays information about our data frame
print(data.info())
```

When working with a larger data sets, it is easier to simply print the first couple rows of our data set. For that, we can use `.head()` which prints the first 5 lines of our Data Frame.

```
import pandas as pd
import matplotlib
#loading Data Frame
data=
        pd.read_csv('/home/codio/workspace/csv/monty_python_movie
        s.csv')

#Display our data frame
print(data)
print()
#Displays information about our data frame
print(data.info())
print()
# head prints the first 5 lines of our data Frame
print(data.head())
```

You can also supply a number inside the parentheses to specify exactly how many lines of the data frame you want. The default is 5.

Try:

```
data.head(1)
data.head(3)
data.head(10)
```