Week 0 - Introduction

A quick introduction to git and python.

Please run through this tutorial on how git functions. Further reading on git exists here.

For an introduction to python programming, please follow the tutorials here and here.

We will cover certain aspects of python programming and specific packages throughout the course. However, at the end of these tutorials, you should be comfortable with the exercises below.

```
In [ ]:  import numpy as np
         from sklearn import datasets
         import matplotlib.pyplot as plt
```

(1) Write a function that calculates and stores the first 30 numbers of the Lucas series in a numpy array.

Then write two assertions as tests of your function.

```
In [ ]:  # Function
         def lucas_series_generator(n):

             if n == 1:
                 return np.array([2])
             elif n == 2:
                 return np.array([2, 1])

             ls = np.zeros(n, dtype=int)

             ls[0] = 2
             ls[1] = 1

             for i in range(2, n):
                 ls[i] = ls[i-1] + ls[i-2]

             return ls

         # Storing and printing results
         ls_30 = lucas_series_generator(30)
         print(ls_30)

         # Assertions
         assert len(ls_30) == 30
         assert np.array_equal(ls_30[:5], np.array([2, 1, 3, 4, 7]))
```

```
[      2       1       3       4       7      11      18      29      47
      76     123     199     322     521     843    1364    2207    3571
    5778    9349   15127   24476   39603   64079  103682  167761  271443
  439204  710647 1149851]
```
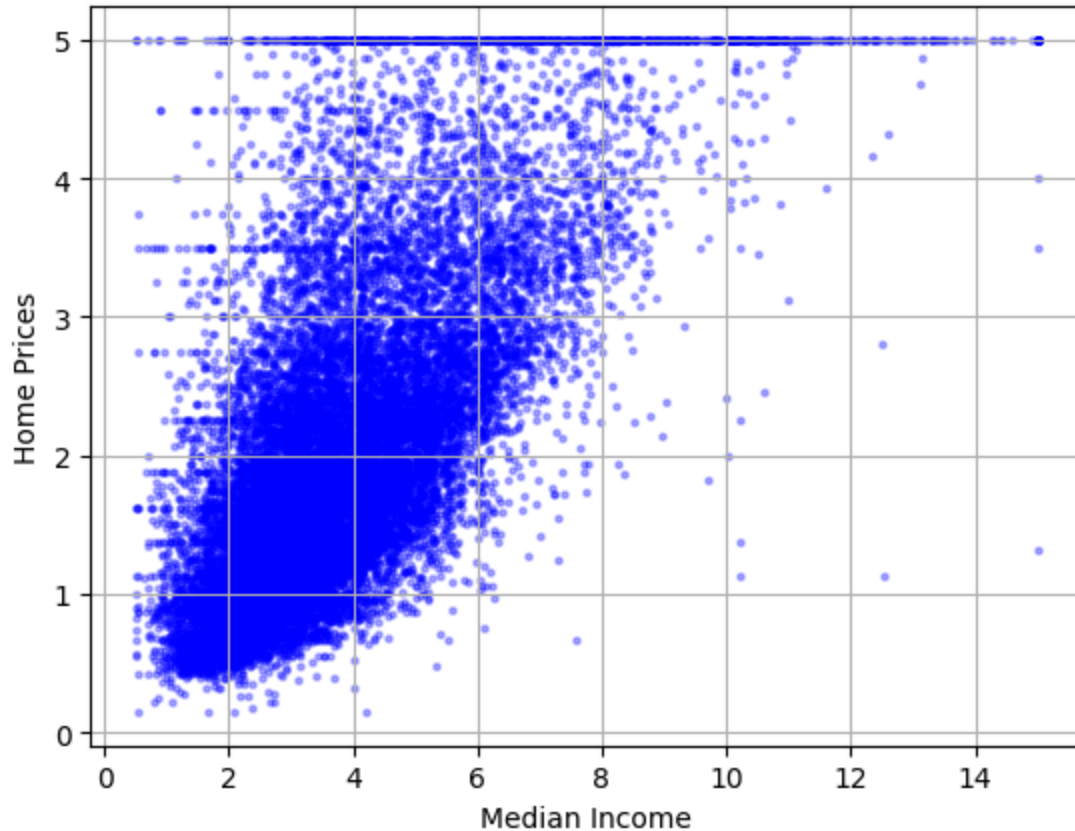
(2) Plot the data that is imported below, labeling the X and Y axes.

```
In [ ]: ca_housing = datasets.fetch_california_housing()

        y = ca_housing.target
        x = ca_housing.data[:, 0]
        ylabel = "Home Prices"
        xlabel = "Median Income"

        plt.scatter(x, y, alpha=0.3, color='blue', s=5)
        plt.xlabel(xlabel)
        plt.ylabel(ylabel)
        plt.grid()
```


(3) Write a function that satisfies the assertions below.

```
In [ ]: # Function here
        def func(n1,n2):
            num = n1*n2
            return num

        assert(func(0, 20) == 0)
        assert(func(1, 10) == 10)
        assert(func(2, 5) == 10)
        assert(func(3, 2) == 6)
        assert(func(4, 1) == 4)
```

(4) Commit these files to git with a descriptive message.