Homework-0

January 30, 2017

1 Homework 0

You will not have to turn this homework in. The goal of this homework is just to practice the skills learned in Lecture 1. These skills will be essential for the rest of this course.

Before starting to work on this homework, **make a new folder** named `hw0-submission` in your private repo and move `Homework-0.ipynb` into that folder.

Let's make sure that you placed everything in the correct folder

```
In [ ]: import os

        if os.path.basename(os.path.realpath('.')) != "hw0-submission":
            print("The name of the current directory should be 'hw0-submission'")
        else:
            print("You did everything correctly. Congrats!")
```

Now, assuming that everything is in their correct place, you should `git add` all the new directories/files and make your initial `git commit`.

1.1 This is a Markdown cell

Insert and edit your own markdown cell <u>below</u>. Write your favorite quote.

```
In [ ]: # This is a code cell. Try executing it. Do not clear your output.
        print("hi")
```

Write a function that takes as argument a string and returns its 3-letter suffix. If the string is shorter than 3 letters, return the whole string.

```
In [ ]: def suffix_3(string):
            """
            Returns the 3-letter suffix.

            Parameters:
                string: str

            Returns:
                suffix: str
                If `string` is more than 3-letters long, `suffix` is its 3-letter suffix. Otherwise, `s
            """
```

```
In [ ]: print(suffix_3("Hi!"))
        print(suffix_3("Superb"))
        print(suffix_3("Yo"))
```

When you finish with the above function, don't forget to `add` and `commit` to your local git repository.

Let's print the versions of some packages that we will be using in the future. If any of these packages are not present, `conda install` them. Do not clear your output.

```
In [ ]: import numpy
        print('numpy:', numpy.__version__)

        import scipy
        print('scipy:', scipy.__version__)

        import matplotlib
        print('matplotlib:', matplotlib.__version__)

        import sklearn
        print('scikit-learn:', sklearn.__version__)

        import pandas
        print('pandas:', pandas.__version__)
```

Write a function that takes as argument a 2D list representing a matrix **M** and returns a boolean indicating whether **M** is an orthogonal matrix.

Keep in mind the possibility of roundoff error in floating point computations.

To test if two matrices are equal, you may find functions in the `numpy.testing` package to be useful.

```
In [ ]: import numpy

        def is_matrix_orthogonal(M):
            """
            Returns True if M is an orthogonal matrix.

            Parameters:
                M: 2D list

            Returns:
                is_orthogonal: boolean
            """
```

```
In [ ]: print(is_matrix_orthogonal([[1, 0], [0, -1]]))
        print(is_matrix_orthogonal([[-1.2396, -4.3801, -1.7737], [-1.3121, -2.9193, -4.5496], [-2.1143,
        print(is_matrix_orthogonal([[-0.8185, 0.5740, -0.0249], [-0.2510, -0.3183, 0.9142], [-0.5168, -0
```

When you finish with the above function, don't forget to `add` and `commit` to your local git repository.

1.2 Submitting

Now, you can push your changes to Github.