assignment02

September 23, 2018

1 Assignment02 : Taylor Approximation in Python

2 Name : Joon-Oh Oh

3 Student ID: 20146136

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import random
        from scipy.misc import derivative
```

4 Define a 2-dimentional function, $f(x) = x^2$

```
In [2]: def givenFunction(x):
            f = x ** 2
            return f
```

5 Create a function for getting first derivative of any given function.

```
In [3]: def firstDerivative(x):
            return derivative(givenFunction, x, dx = 1e-4)
```

6 Set the domain for the given function.

```
In [4]: x = np.arange(-15, 15, 0.1)
```

7 Randomly pick 3 points from the defined domain.

```
In [5]: points = np.empty(3)
        print("Three picked points: ")
        for i in range(0, 3):
            points[i] = random.choice(x)
            print(points[i])
```

```
Three picked points:
-5.500000000000034
-7.300000000000027
7.199999999999921
```

8 Define the alfine function after taken taylor approximation.

```
In [6]: def taylorAprroximation(x, xPoint, yPoint):
            f = firstDerivative(xPoint) * (x - xPoint) + yPoint
            return f
```

9 Compute the result graphs.

```
In [7]: f = givenFunction(x)
        aF = np.empty((3,x.size))
        for i in range(0, 3):
            aF[i] = taylorAprroximation(x, points[i], givenFunction(points[i]))
```

10 Plot selected 3 points, 3 taylor approximations, and the original function.

```
In [8]: plt.figure(1)
        plt.plot(x, f, 'b', label="original function")
        plt.xlim(-20, 20)
        for i in range(0, 3):
            plt.plot(points[i], givenFunction(points[i]), 'o', label="picked point %d" %i)
        for i in range(0, 3):
            plt.plot(x, aF[i], 'r', label="taylor approximation at x = %d" % points[i])
        plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
        plt.show()
```