assignment02

September 29, 2018

1 This script demonstrates the second order Taylor expansion of a given function

1.0.1 Name : Dohyun Kim

1.0.2 Student ID : 2018120193

1.0.3 Git : https://github.com/ppooiiuuyh/datamining_assignments/tree/master/assignment02

2 import packages for plotting graphs and manipulating data:

```
In [49]: import numpy as np
         import matplotlib.pyplot as plt
```

3 define my function: $f(x) = \cos(x) \cdot x$

```
In [50]: def myFunction(x):
             f = np.cos(x) * x
             return f
```

4 define the derivative of my function:

$first : f'(x) = -\sin(x) \cdot x + \cos(x)$
$second : f''(x) = -\cos(x) \cdot x - 2\sin(x)$

```
In [57]: def myDerivFunc(x):
             Df = - np.sin(x) * x + np.cos(x)
             return Df

         def myDerivFunc_2(x):
             Df2 = -np.cos(x)*x -np.sin(x) + -np.sin(x)
             return Df2
```

5 define second order Taylor expansion

```
In [58]: def second_order_taylor(x,a,func, first_der, second_der):
             sot = func(a) + (x-a)*first_der(a) + ((x-a)**2)*second_der(a)/2
             return sot
```

6 define the domain of the function: $x = [-15 : 0.1 : 15]$

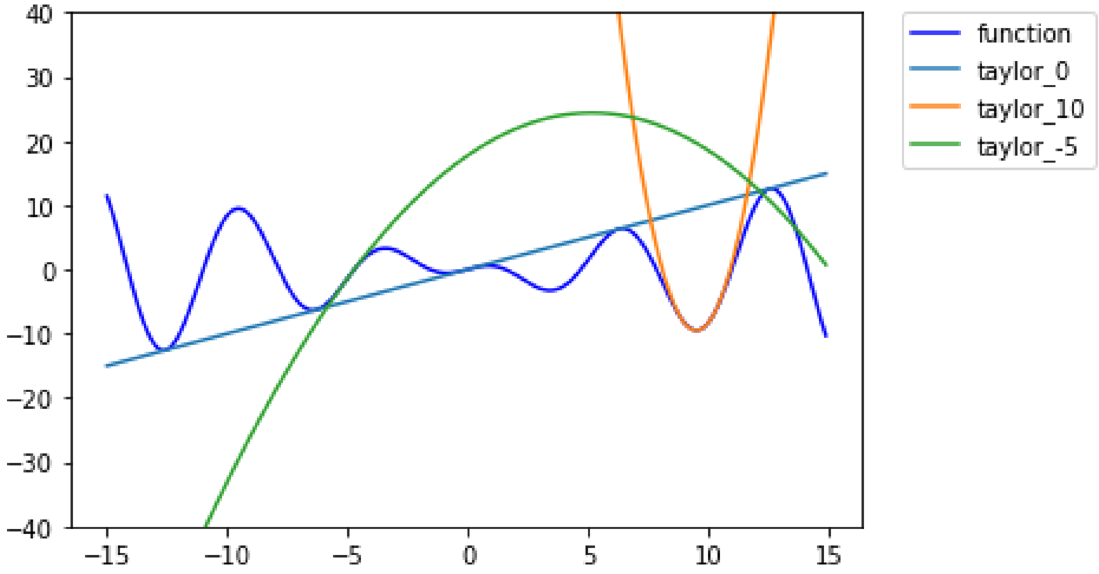
```
In [59]: x = np.arange(-15, 15, 0.1)
         A = [0,10,-5]
```

7 compute the graph

```
In [60]: f = myFunction(x)
         sots = []
         for a in A:
             sots.append(second_order_taylor(x,a,myFunction,myDerivFunc,myDerivFunc_2))
```

8 plot the graphs for the function and its derivative

```
In [61]: plt.figure(1)
         plt.ylim(-40,40)
         plt.plot(x, f, 'b', label="function")
         #plt.plot(x, Df, 'r', label="derivative")
         for e,a in enumerate(A) :
             plt.plot(x, sots[e], label="taylor_"+str(a))
         plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
         plt.show()
```


function
taylor_0
taylor_10
taylor_-5