CSC431 Final Project - Due Date: March 17

1 Part 1

1. Generate random $[x, y, \delta y]$ data points using the following code

```
data = [(x,random.gauss(1,0.01)*exp(log(2)+log(id)*x/400),20)
        for x in range(100,201)]
```

where `id` is your student id.
2. Plot the data.
3. The data can be modeled with the following function

$$y = 2e^{bx}$$

You must determine the value of b in three different ways:

- Analitycally, as function of your student id.
- Using the linear least square fitting function discussed in class. Notice the fitting function above is not linear in b therefore you need to perform a transformation of the data before you can apply linear least squares.
- By minimizing χ^2 using an optimization algorithm (for example Newton).

For each of the above methods:

- Explain the method in detail
- Show your steps
- Show your result
- A plot of your fitting function superimposed to the data

Do you expect the three methods to agree? Why? If you do not expect them to agree exactely, To what precision should they agree?

1.1 Part 2

Here is a typical implementation a the Black-Scholes pricing formula:

```
def cdf(z):
    if z > 6.0:
        I = 1.0
    elif z < -6.0:
        I = 0.0
    else:
        a = fabs(z)
        b1 = 0.31938153
        b2 = -0.356563782
        b3 = 1.781477937
        b4 = -1.821255978
        b5 = 1.330274429
        p1 = 0.2316419
        c1 = 0.398923
        t = 1.0/(1.0+a*p1)
        I = 1.0 - c1*exp(-z*z/2)*((((b5*t+b4)*t+b3)*t+b2)*t+b1)*t
        if z<0:
            I = 1.0-I
    return I

def priceBS(S, X, r, sigma, t):
    sqrt_t = sqrt(t)
    d1 = (log(S/X)+r*t)/(sigma*sqrt_t)+0.5*sigma*sqrt_t
    d2 = d1 - sigma*sqrt_t)
    c = S*cdf(d1)-X*exp(-r*t)*cdf(d2)
    return c
```

where `cdf` is an an auxiliary funciton (it approximates the cumulative distribution function for the gaussian distribution) and `priceBS` is the actual pricing algorithm. The inputs are:

- `S`: underlying stock price
- `X`: strike price
- `r`: daily expected rate of return for the stock

- `sigma`: daily volatility for the stock
- `t`: time to expiration in days (can be smaller then 1).

The function returns the Black-Scholes price for a European call option.

S and X are dollar amounts. Will the result be more accurate or less accurate if S and X are expressed in cents? What if they are expressed in thousands of dollars?

Will the result be more accurate or less accurate if time is expressed in years instead of days (and accordingly r,σ are yearly return and volatility)?

Plot the output for $X = 10.0, r = 0.001, \sigma = 0.03, t = 90.0$ as function of S in range $[0, 20]$.

Look up the defintions of the greeks (delta, theta, and gamma) and write a program that computes them numerically using the above formula. `https://www.investopedia.com/terms/g/greeks.asp`. Plot the values of the greeks as function of X, S, σ and t in separate plots.

Deliverables

You must deliver a 6 pages paper in PDF (Word documents will not be accepted) solving both problems described above. For exach problem, the paper should include an introduction explaining the problem in your own language. Imagine the reader of your paper your boss at work and not somboby like me who alreday understands the problem. The paper should include one section for each of the three methods described above. The paper should include 4 plots as requested above. The paper should have a conclusion answering the above questions.

For each problem the paper should have an appending listing all your code. The code must be properly formatted/indented.

Be aware that this is an individual project. The data should be different for every student and different papers will be checked for similarities. Excessive similarities will be reported for plagiarism.