QUESTION 1a

Write a 500-word explanation of bitcoin stock-flow-model and make an argument for why it is wrong?

ANSWER

The Stock-to-flow model (S2F), popularized by a pseudonymous Dutch Institutional Investor who operates under the Twitter account "PlanB", has been widely praised and is the leading valuation model for bitcoin proponents.

PlanB's paper "Modelling Bitcoin Value with Scarcity" states that certain precious metals have maintained a monetary role throughout history because of their unforgeable costliness and low rate of supply. For example, gold which is valuable because new supply (minded gold) is insignificant to the current supply and because it is impossible to replicate the vast stores of gold around the globe. He then put forward that this same logic applies to bitcoin, which becomes more valuable as new supply is reduced every four years, ultimately culminating in a supply, which PlanB defines as "Scarcity" can be quantified using a metric called STOCK-TO-FLOW (S2F), which is the ratio between current supply and new supply. The Stock-to-Flow model attempts to value BTC in a way similar to other scarce assets like gold and silver, it looks at historical values of BTC and projects where it might go over time.

The model suggest that investors can forecast the future USD market capitalization. This has helped give credence to those $100,000 bitcoins projections.

From the theoretical view point, the model is based on the rather strong assertion that the USD market capitalization of a monetary good (e.g gold and silver) is derived directly from their rate of new supply. No evidence or research is provided to support this idea, other than the singular data points selected to chart gold and silver's market capitalization against bitcoin's trajectory.

The stock- to – flow model looks at historical values of BTC and projects where it might go over time.

Bitcoin's stock-to-flow model gives you the impression that gold stock-to-flow is nearly constant. The reality is that gold's stock-to-flow ratio is constantly fluctuating.

Gold's S2F does not drive its price; plan B observed that because bitcoin's mining flow gets cut in half every four years, bitcoin's S2F ratio leaps every four years. This halving events boost bitcoin's stock-toflow ratio dramatically. Contracting supply is fueling the BTC's dramatic price rise.

Intuitively the Stock-to-Flow model makes sense once you conduct a simple thought experiment. Imagine if the flow of gold suddenly slowed to a trickle. Instead of 3,000 tons of gold being mined annually, only 3 tons of gold each year; I predict that the price would skyrocket. That's because investors, national banks, jewelry makers, phone makers (0.034grams of gold in each phone) and other industries that use gold would have to fight over a (nearly) fixed supply.

Gold analyst and bugs only cite gold's stock-to-flow to help explain why gold has monetary value, but they don't use it to predict gold's price.

Economics 101 teaches that the price of anything is driven by supply and demand. If the supply is only known then the price cannot be predicted.
The stock-to-flow model will be accurate provided that the demand continues to grow exponentially as it has for the last 10 years.

QUESTION 1b

Yara Inc. is listed on the NYSE with a stock price of $40- the company is not known to pay dividends. We need to price a call option with a strike price of $45 maturing in 4 months. The continuously-compounded risk-free rate is 3% per year, the mean return on the stock is 7% per year and the standard deviation of the stock return is 40% per year. What is the Black-Scholes call price?

SOLUTION

Given:

Stock Price (p_0) = $40

Strike Price (x) = $45

Time to maturity (t) days = 0.33355

Annual Risk-free rate (r) = 3%

Volatility (σ) = 40%

Using;

$$V_c = P_0 N d_1 - \frac{X}{e^{Krf(t)}} N d_2$$

Where;

$$d_1 = \frac{[In\left[\frac{P_0}{X}\right]+(K_{rf}+0.5\sigma^2)t]}{\sigma\sqrt{t}}$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

$$d_1 = \frac{[In\left[\frac{40}{45}\right]+(0.03+0.5(0.40)^2).33355\,]}{0.40\sqrt{0.33355}}$$

$$= \frac{-0.117783+(0.11)(0.33355)}{0.231015}$$

$$= \frac{-0.117783+0.036690}{0.231015}$$

$$= -0.35102$$

$$d_2 = d_1 - \sigma\sqrt{t}$$

$$d_2 = -0.35102 - (0.40)\sqrt{0.33355}$$

$$= -0.315102 - 0.231015$$

$$= -0.582035$$

$$N(d_1) = 0.363169$$

$$N(d_2) = 0.280957$$

$$V_c = 40(0.363169) - \frac{45}{e^{0.03(0.33355)}}(0.274252)$$

$$= 14.52676 - 12.51718$$

$$= 2.009$$

$V_c =$ \$2.01

QUESTION 2a

Why is it a bad idea to use recursion method to find the Fibonacci of a number?

Recursion by definition is “when a thing is defined in terms of itself”. With respect to a programming function, recursion happens when a function calls itself within its own definition. It calls itself over and over again until a base condition is met that breaks the loop.

Running a space complexity analysis will be a tricky thing to do because of a lot of things are happening behind the scenes of a recursive function and the reason for the poor performance is heavy push and pop of the stack memory in each recursive call.

The iteration method is a faster approach to finding the Fibonacci of a number as it stores the first two values of our Fibonacci number in two variables and using “current number” to store our Fibonacci number. Storing these values prevents us from constantly using memory space in the stack.

QUESTION 2b

Write a function that takes in a Proth Number and uses Proth’s theorem to determine if said number is prime? You can write this in any programming language but C/C++/Go lang are preferred.

ANSWER

```
//C++ program to check a Proth number and prime.

#Include <iostream>

Using namespace std;

// Utility function to check power of two.

bool  IsPowerOfTwo (int n)

{

return (n && ! (n & (n-1)));

}

// Function to check if the given number is a proth prime number or not, using n=k*2^n+1

bool isProthNumber (int n)

{

int k=5;

while (k < (n/k)) {

// check if k divides n or not

if  (n % k == 0) {

// check if n/k is power of 2 or not

if (isPowerOfTwo (n/k))

return true;

}
```

```
// updateto next odd number
K=k+2
}
// if we get here it means there exists no value of k such that k is a odd number and n/k is a poer of 2 greater than k.
return false;
}
//Driver code
int main( )
{
// get n
int n = 3;
int k = 5
//check n for proth number
if (isProthNumber (n-1))
cout << “YES”;
else
cout << “NO”;
return 0;
}
```

OUTPUT: YES

Question 3

Over all real numbers, find the minimum value of a positive real number, y such that

$$y = \sqrt{(x+6)^2+25} + \sqrt{(x-6)^2+121}$$

SOLUTION

$\frac{dy}{dx} = 1\frac{2(x+6)}{2\sqrt{(x+6)^2+25}} + 1\frac{2(x-6)}{2\sqrt{(x-6)^2+121}}$

$\text{At minimum } \frac{dy}{dx} = 0$

$0 = \frac{x+6}{\sqrt{(x+6)^2+25}} + \frac{(x-6)}{\sqrt{(x-6)^2+121}}$

$\frac{(6-x)}{\sqrt{(x-6)^2+121}} = \frac{(x+6)}{\sqrt{(x+6)^2+25}}$

Where;

$(6-x) = -(x-6)$

Squaring both sides

$$\frac{(-(x-6))2}{(x-6)^2+121} = \frac{(x+6)2}{(x+6)^2+25}$$

$$\frac{(x-6)^2}{(x-6)^2+121} = \frac{(x+6)^2}{(x+6)^2+25}$$

$$(x-6)^2(x+6)^2 + 25(x-6)^2 = (x-6)^2(x+6)^2 + 121(x+6)^2$$

$$25(x-6)^2 = 121(x+6)^2$$

$$\frac{25}{121} = [\frac{x+6}{x-6}]^2$$

$$\frac{x+6}{x-6} = \frac{5}{11}$$

$$11x + 66 = 5x - 30$$

$$6x = -96$$

$$x = -16$$

Now at minimum point, $x = -16$

$y = \sqrt{(x+6)^2+25} + \sqrt{(x-6)^2+121}$

At,

$$x = -16$$

$$y = \sqrt{(-10)^2 + 25} + \sqrt{(-22)^2 + 121}$$

$$y = \sqrt{125} + \sqrt{605}$$

$$= \sqrt{25 * 5} + \sqrt{121 * 5}$$

$$= 5\sqrt{5} + 11\sqrt{5}$$

$$y = 16\sqrt{5}$$

That is the minimum value is $16\sqrt{5}$.