**Why Bancor Protocol is able to stabilize the price of smart token?**
**i.e., why the price of smart token fluctuates like figure below?**
(12/23 Final Bancor Protocol.pdf)

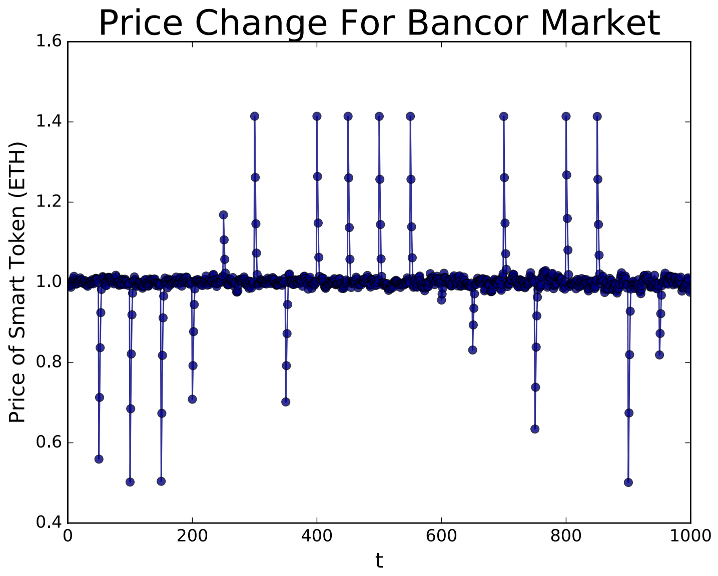


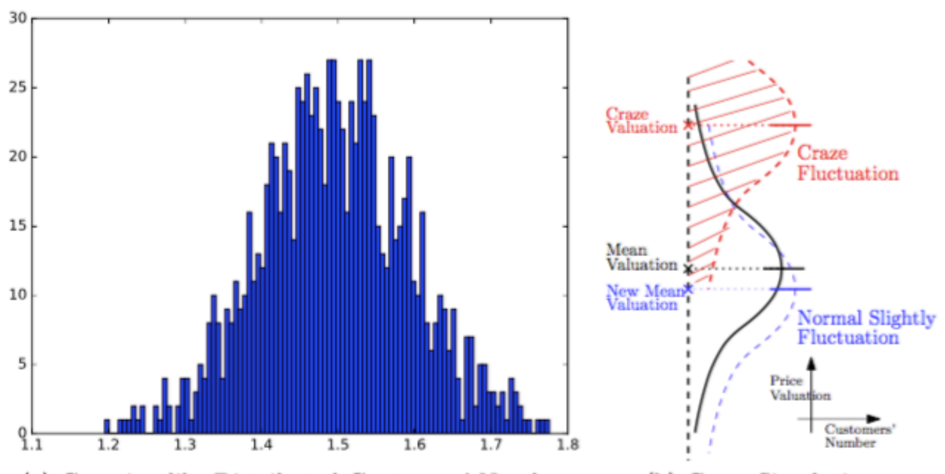
Figure 1: The x-axis denotes to the number of time step in simulator, and the y-axis presents the price of smart token in Bancor market, e.g. in the beginning, 1 smart token is valued at 1.0 ETH.

**Ans.**
notations:
**T**: interval between time epochs.
**R**: the bouncing range of mean valuation, when time epochs comes.
**N**: customer number
**sig**: the sigma in Gaussian function. Smaller the sigma is, closer valuations are.
**Ps**: the price of smart token in the Bancor market

What simulator does is:
1. Customers in Bancor simulator follow the all-in policy -- they use all of their smart tokens or reserve tokens for sale or purchase.
2. In normal cases, customers launch their valuation based on the current price of smart token **(Psc)** as mean valuation. In time epoch, customers randomly choose a valuation between range from **Psc/R** to **Psc*R.**

Figure 2 helps to better understand this policy. Below codes show how we implement this policy in programming:

```
Psc = MyBancorMarket.getCurrentPrice()
if (j > 0) and (j % R == 0):
    # Choose new valuation between (Psc / R, Psc * R)
    if bool(random.getrandbits(1)):
        valuation_mu = random.uniform(Psc / R, Psc)
    else:
        valuation_mu = random.uniform(Psc, Psc * R)
else:
    valuation_mu = currentMarketPrice
```



(a) Gaussian-like Distributed Customers' Number (b) Craze Simulating

Figure 2: This figure shows how customers make valuation. In Figure 2(b), black lined valuation curve represents the original states. The blue lined curve means the new valuation of made in normal cases, using the new slightly changed **Ps** as mean valuation; while red lined curve shows the new valuation distribution made by using a bounced value (**Ps/R** ~ **Ps*R**) as mean valuation, when time epoch comes.

Now let us use a simple example to illustrate why the price of smart token is stable in simulating.
Initialization in code:

```
# Smart Token Initialization
KennyCoin = Smartcoin(name='Kenny',reservetokenName='ETH',initCRR=0.5,
                      initPrice=1,initIssueNum=800000)
# Bancor Market Initialization
MyBancorMarket = BancorMarket(smartToken = KennyCoin)
# Customer Initialization
custList = []
custNum = 2000
for i in range(custNum):
  Joe = Customer(smartToken = KennyCoin, market = MyBancorMarket,
                 tokenBalance = 200,
                 reserveBalance = 200)
  custList.append(Joe)
```

Thus in the original state, our Bancor Market has properties below, and we also record all customers' status:

Market:

Price of Kenny Coin: 1.0 ETH
Smart Token Supply: 800000

All Customers:

Reserve Balance: 400000 # 200 * 2000
Smart Token Balance: 400000

Assuming the next time step of original state is time epoch, and all customers are going to buy Kenny Coin. (Extreme Case)
Then, due to all-in policy, the status of Market and all customers is changed as below:

Market:

Price of Kenny Coin: 1.414 ETH
Smart Token Supply: 1131371

All Customers:

Reserve Balance: 0
Smart Token Balance: 731371

As you can see, the price of smart token is increased.
Then, in the next time step we generate valuation "normally" -- we use 1.414 ETH now as mean valuation. Therefore, there might be half customers who want to buy and half who want to sell. However, as those who want to buy do not have any reserve token, there are actually only sellers in the market whose number takes over about the half of all customers' number. In this situation, the new status of Market and Customers is listed as below:

Market:

Price of Kenny Coin: 0.957 ETH
Smart Token Supply: 765686

All Customers:

Reserve Balance: ≈433578
Smart Token Balance: ≈365685 # 731371 /2

Again, if the next time step we generate valuation "normally", we get:

Market:

Price of Kenny Coin: ≈0.979 ETH
Smart Token Supply: ≈783148

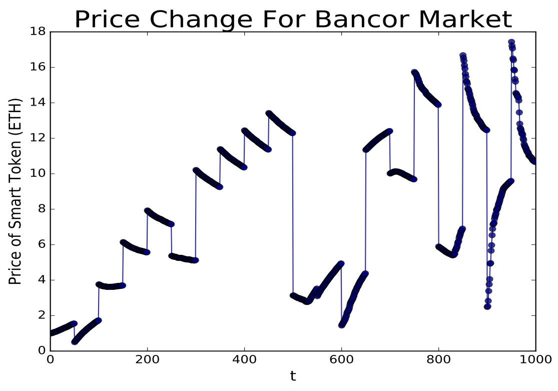
All Customers:

Reserve Balance: ≈ 383146
Smart Token Balance: ≈ 416674

It can be viewed that the Price of Kenny Coin now goes back and is close to 1.0 ETH. Therefore, the reason why in our simulator the Price of smart token can always go back to average after its bouncing, is that **under all-in policy, almost half of customers cannot launch transaction orders after time epoch.**

**Potential Sol:**
We can add customers' reserve or smart token balance if they are run out.
However, how much money should we add becomes a tricky problem. Also, we cannot add too many smart tokens to customers as smart tokens cannot be unlimitedly added (the total sum should <= Smart Token Supply).

I also made experiments after I added 200 reserve to customers once they run out, the result is presented as below:



This result actually meets our expectation.

In my opinion, the Bancor Market does have its ability to stabilize the price of smart token (even in potential solution's part, the price of smart token slowly tries to recover to its original status).
In fact, after careful examination of my codes, I believe, at least in this part, there might be no or very few bugs. **The reason why the price of smart token is stable is that our all-in policy exaggerates the recovering ability of Bancor**.