Appendix

Part of Python Code:

```python
# CVaR
mu_h = BitcoinMean['A'][i+5]
sig = BitcoinSD['A'][i+5]
alpha = 0.01
sig_h = sig * np.sqrt(h / windowA)
lev = 100 * (1 - alpha)
CVaR_n_A = alpha ** -1 * norm.pdf(norm.ppf(alpha)) * sig_h - mu_h

# Strategy
def Strategy(i, delta=0.01):
    # delta = 0.0001
    Sigma = np.mat([[GoldRisk[i], BitcoinRisk[i]], [BitcoinRisk[i], GoldRisk[i]]])
    Omega = np.matmul((delta * np.linalg.inv(Sigma)),
                      np.mat([[GoldMeanRes[i]], [BitcoinMeanRes[i]]]))
    if Omega[0] * Omega[1] > 0:
        if Omega[0] < 0 and Omega[1] < 0:
            Omega[0] = 0
            Omega[1] = 0
        else:
            temp = (Omega[0] + Omega[1])
            Omega[0] = Omega[0] / temp
            Omega[1] = Omega[1] / temp
    else:
        if Omega[0] < 0:
            Omega[0] = 0
            Omega[1] = 1
        if Omega[1] < 0:
            Omega[1] = 0
            Omega[0] = 1

    return Omega

 for i in range(len(GoldRisk)-1):
    # print("i:", i, "\n")
    Omega = Strategy(i, 0.0005)
    print(Omega)
    # 0 为 gold， 1 为 bitcoin

    if Omega[0] == lastOmega[0] and Omega[1] == lastOmega[1]:
        lastOmega = Omega
        # print("Indication: 1\n")

    elif Cash > 0:
        trade = Omega
        GoldAmount = float(Cash) * trade[0] / GoldPrice[window + i - 1] * (1 - alphaGold / 10000)
        BitcoinAmount = float(Cash) * trade[1] / BitcoinPrice[window + i - 1] * (1 - alphaBitcoin / 10000)
        lastOmega = Omega
        Cash = 0
        # print("Indication: 2\n")

    elif Cash == 0:
        if Omega[0] == 0 and Omega[1] == 0:
            Cash = GoldAmount * GoldPrice[window + i - 1] * (1 - alphaGold / 10000) + BitcoinAmount * BitcoinPrice[window + i - 1] * (1 - alphaBitcoin / 10000)
            GoldAmount = 0
            BitcoinAmount = 0
        else:
            OmegaDiff = Omega - lastOmega

            if float(OmegaDiff[0]) < 0: # gold
                Cash = GoldAmount * abs(float(OmegaDiff[0])) * GoldPrice[window + i - 1] *(1 - alphaGold / 10000)
                GoldAmount = GoldAmount - GoldAmount * abs(float(OmegaDiff[0]))
                BitcoinAmount = BitcoinAmount + Cash / BitcoinPrice[window + i - 1] * (1 - alphaBitcoin / 10000)
                Cash = 0
                # print("Indication: 3\n")

            elif float(OmegaDiff[1]) < 0:
                Cash = BitcoinAmount * abs(float(OmegaDiff[1]))* BitcoinPrice[window + i - 1] * (1 - alphaBitcoin / 10000)
                BitcoinAmount = BitcoinAmount - BitcoinAmount * abs(float(OmegaDiff[1]))
                GoldAmount = GoldAmount + Cash / GoldPrice[window + i - 1] * (1 - alphaGold / 10000)
                Cash = 0
                # print("Indication: 4\n")

    lastOmega = Omega
    value = float(Cash) + GoldAmount * GoldPrice[window+i] + BitcoinAmount * BitcoinPrice[window+i]

    Value.append(float(value))
    print(i, value, "\n")

```

Part of Python Code:

```
for (i in BayesWindow:length(Gold)){
  tempData= c(Gold[(i-BayesWindow):i])
  tempData = zoo(tempData, index(tempData))
  ss = AddSemilocalLinearTrend(list(), tempData)
  model = bsts(tempData, state.specification = ss, niter = 500)
  Res = predict(model)
  Mean=Res$mean
  sd = Res$mean-Res$interval[1]

  GoldBMean = c(GoldBMean, Mean)
  RelativeGoldBMean = c(RelativeGoldBMean, (Mean-tempData[BayesWindow])/tempData[BayesWindow])
  GoldBSD = c(GoldBSD, sd)

  cat(i, "\n")
  cat("\n")
  cat("\n")
}
```