B2_황재훈

Jaehun Hwang

1/4/2021

P4.Implementation

- Following code tries the stock level $X \in \{11, 12, 13, 14, 15\}$

```
import numpy as np

for X in range(11,16):
    MC_N = 1000
    D = np.random.choice(list(range(11,16)), MC_N, replace=True)
    sales_rev = 2 * np.minimum(D, X)
    salvage_rev = 0.5 * np.maximum(np.subtract(X,D), 0)
    material_cost = 1 * X
    profit = sales_rev + salvage_rev - material_cost
    print("X: ", X, ", expected profit: ", np.mean(profit))
```

```
## X:  11 , expected profit:  11.0
## X:  12 , expected profit:  11.703
## X:  13 , expected profit:  12.07
## X:  14 , expected profit:  12.221
## X:  15 , expected profit:  12.0225
```

P5.Uniform

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

try_X = np.arange(20, 40.001, 0.01)
# print(try_X)

result = {"try_X": [], "exp_profits": []}
# exp_profits = None

for X in try_X:
  MC_N = 10000
  D = np.random.uniform(20, 40, size=MC_N)
  sales_rev = 2 * np.minimum(D, X)
  salvage_rev = 0.5 * np.maximum(np.subtract(X,D), 0)
  material_cost = 1 * X
  exp_profit = np.mean(sales_rev + salvage_rev - material_cost)
  result["try_X"].append(X)
  result["exp_profits"].append(exp_profit)

result_frame = pd.DataFrame(result)

print(result_frame.head(10))



plt.figure()
result_frame.plot.scatter(x="try_X", y="exp_profits")
plt.savefig("B1_P6.png", bbox_inches="tight")
```

```
##    try_X  exp_profits
## 0  20.00    20.000000
## 1  20.01    20.009998
## 2  20.02    20.019988
## 3  20.03    20.029963
## 4  20.04    20.039976
## 5  20.05    20.049951
## 6  20.06    20.059812
## 7  20.07    20.069854
## 8  20.08    20.079738
## 9  20.09    20.089733
```

Output:

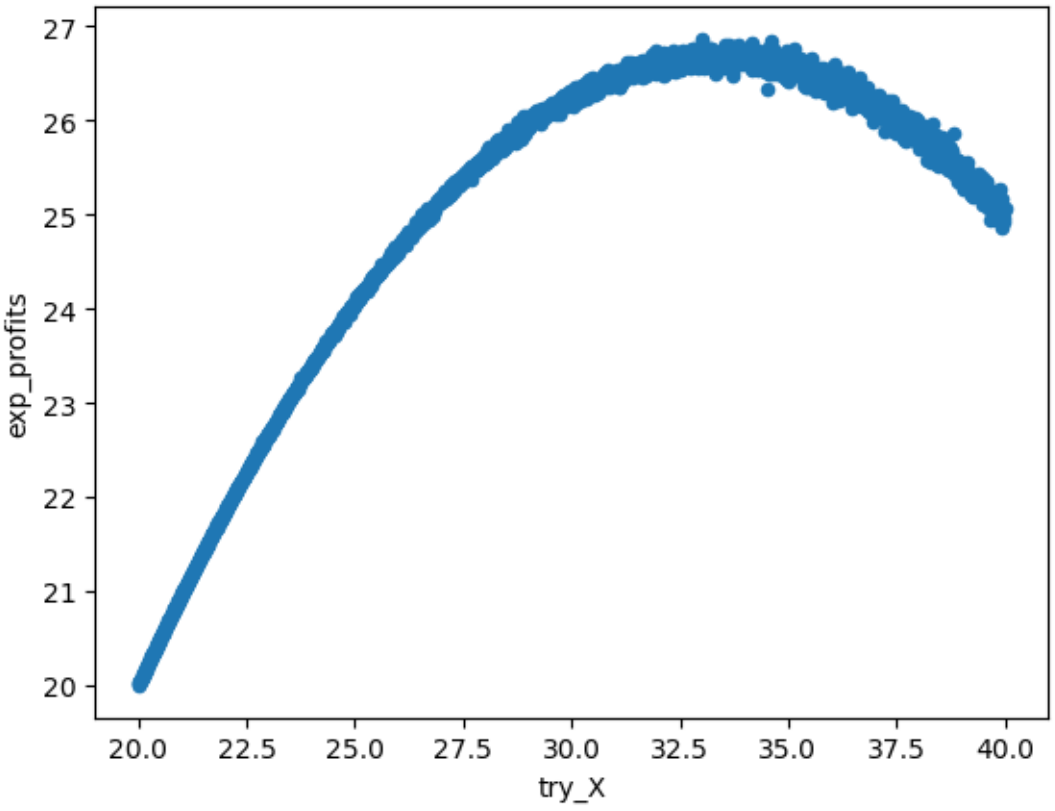

Figure 1: output