B2_Exercises

Kwon do yun

2021-01-04

Implementation (p. 4)

```
import numpy as np

for X in range(11,16):
    D=np.random.randint(11,high=16)
    sales_rev = 2*np.minimum(D,X)
    salvage_rev = 0.5*np.maximum(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 :  12.0
## X:  13 , Expected profit :  13.0
## X:  14 , Expected profit :  14.0
## X:  15 , Expected profit :  13.5
```

```
import numpy as np

for X in range(11,16):
    D = np.random.uniform(11,15,X)
    sales_rev = 2*np.minimum(D,X)
    salvage_rev = 0.5*np.maximum(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 :  11.684179013421174
## X:  13 , Expected profit :  12.762699986920952
```

```
## X:  14 , Expected profit :  11.954237693019374
## X:  15 , Expected profit :  12.066315669784666
```

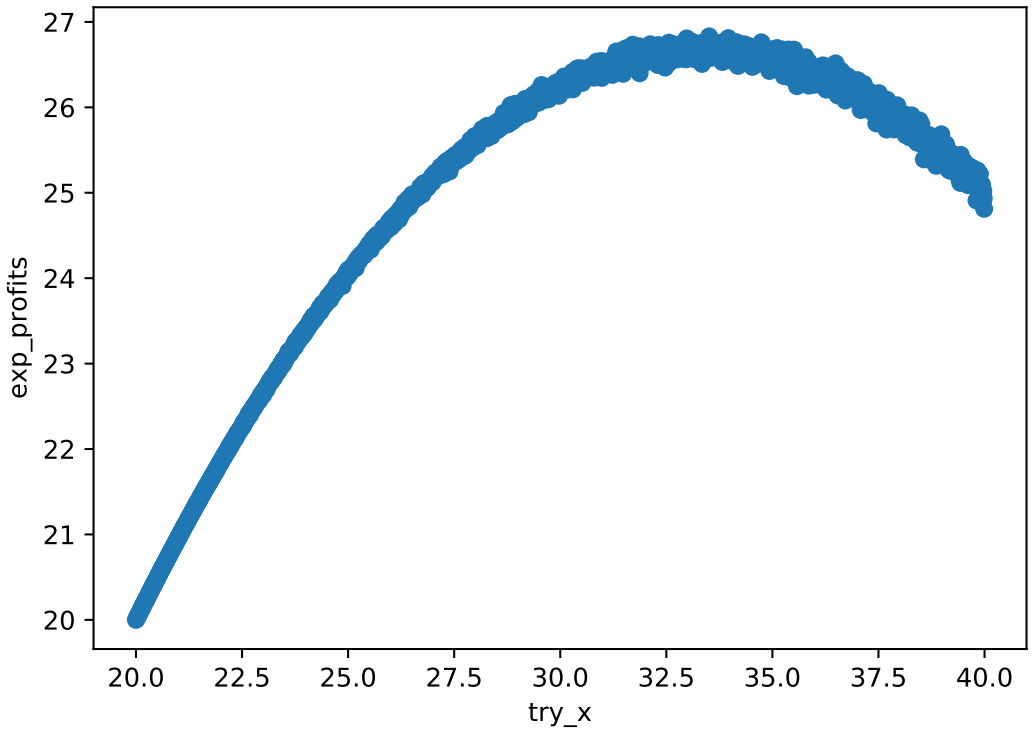
Continuous distribution - grid search approach (p. 5)

```
import numpy as np
import pandas as pd

try_X = np.arange(20,40,0.01)
exp_profits=np.array([])

for X in try_X:
    MC_N = 10000
    D = np.random.rand(MC_N,1)*20+20
    sales_rev= 2*np.minimum(D,X)
    salvage_rev = 0.5*np.maximum(X-D,0)
    material_cost = 1 *X
    exp_profit= np.mean(sales_rev + salvage_rev - material_cost)
    exp_profits=np.append(exp_profits,exp_profit)

results = pd.DataFrame({'try_X':try_X,'exp_profits':exp_profits})
results
```

```
##       try_X  exp_profits
## 0     20.00    20.000000
## 1     20.01    20.009994
## 2     20.02    20.019978
## 3     20.03    20.029982
## 4     20.04    20.039947
## ...     ...          ...
## 1995  39.95    25.002477
## 1996  39.96    24.896690
## 1997  39.97    25.025202
## 1998  39.98    24.935484
## 1999  39.99    24.812516
##
## [2000 rows x 2 columns]
```

plot (p. 6)

```
import matplotlib.pyplot as plt

plt.plot(try_X,exp_profits,'o')
plt.xlabel("try_x")
plt.ylabel("exp_profits")
plt.show()
```



(p. 8)

```
idx=np.where(exp_profits==np.max(exp_profits))

print(try_X[idx])
```

```
## [33.51]
```

```
print(exp_profits[idx])
```

```
## [26.83031059]
```