B2_total

Son Min Sang

2021-01-11

차 례

Implementation (page 4)

```
import numpy as np

for X in range(11,16):
    MC_N=10000
    D=np.random.choice(np.arange(11,16),MC_N,replace=True) # random discrete uniform

    sales_rev=2*np.minimum(D,X) # vector level minimum
    salvage_rev=0.5*np.maximum(X-D,0) # vector level maximum
    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:  11.70135
## X:  13  expected profit:  12.1186
## X:  14  expected profit:  12.20345
## X:  15  expected profit:  12.02865
```

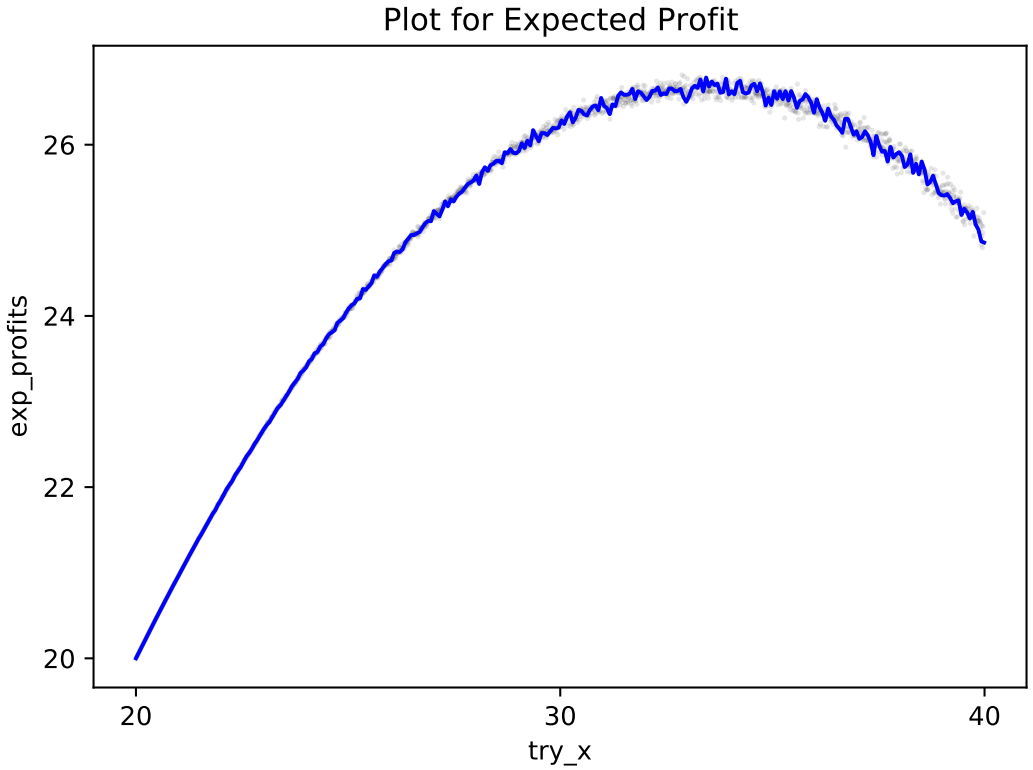
Continuous distribution - grid search approach (page 5)

```
import numpy as np
import pandas as pd

try_X = np.arange(20,40,step=0.01)
exp_profits=np.array([])
for X in try_X:
    MC_N = 10000
    D=np.random.uniform(20,40,size=MC_N)
    sales_rev= 2*np.minimum(D,X) # vector level minimum
    salvage_rev = 0.5*np.maximum(X-D,0) # vector level minimum
    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    20.019989
## 3     20.03    20.029963
## 4     20.04    20.039981
## ...     ...          ...
## 1995  39.95    24.797075
## 1996  39.96    24.972881
## 1997  39.97    25.043909
## 1998  39.98    25.205681
## 1999  39.99    24.856851
##
## [2000 rows x 2 columns]
```

Plot (page 6)

```
from scipy.interpolate import make_interp_spline, BSpline
import matplotlib.pyplot as plt

plt.plot(try_X,exp_profits,'o', color ='black', ms=1,alpha=0.1)
plt.xlabel("try_x")
plt.ylabel("exp_profits")
plt.title("Plot for Expected Profit")
plt.rc('font', size=25)
x_new = np.linspace(results['try_X'].min(),results['try_X'].max(),300)
spline = make_interp_spline(results['try_X'], results['exp_profits'], k=3)
power_smooth = spline(x_new)
plt.plot(x_new,power_smooth , color = 'blue')
plt.show()
```



page 8

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

```
print(*try_X[idx]) #this is optimal quantity
```

```
## 32.87000000000201
```

```
print(*exp_profits[idx]) #this is expected optimal profit
```

```
## 26.81550148593958
```