

# Lecture B2.Newsvendor2

Baek, Jong min

2021-01-05

## 차 례

Implementation(Page.4) . . . . .	2
Continuous distrivution - grid search approach(Page.5) . . . . .	3
Visualization1(Page.6) . . . . .	4
Visualizaion2(Page.7) . . . . .	5
Page.8 . . . . .	5

## Implementation(Page.4)

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

```
# replace = True(Restoration extraction)

for X in range(11,16):
    MC_N = 10000
    D = np.random.choice(range(11,16),MC_N,replace=True)
    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 : {}, expected profit : {}'.format(X,np.mean(profit)))

## x : 11, expected profit : 11.0
## x : 12, expected profit : 11.69985
## x : 13, expected profit : 12.11395
## x : 14, expected profit : 12.20585
## x : 15, expected profit : 12.01185
```

## Continuous distribution - grid search approach(Page.5)

```
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(low=20,high=40,size=MC_N)

    sales_rev = 2*np.minimum(D,X)
    salvage_rev = 0.5*np.maximum(X-D,0)
    material_cost = 1*X

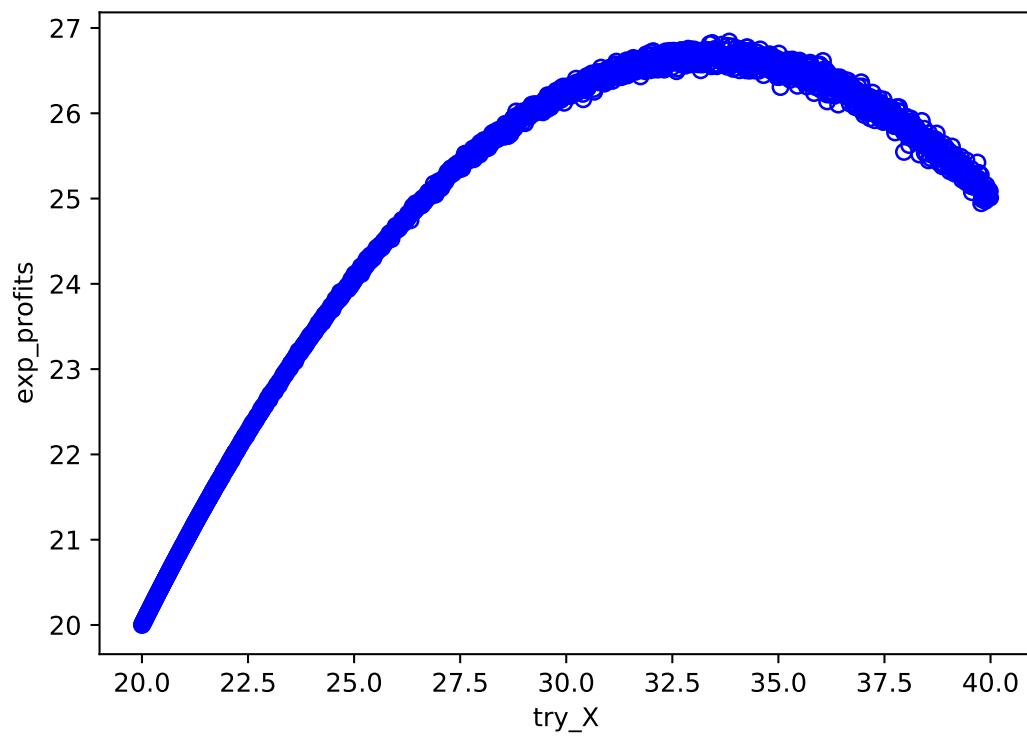
    exp_profit = np.mean(sales_rev + salvage_rev - material_cost)
    exp_profits = np.append(exp_profits,exp_profit)

results = pd.DataFrame(try_X,exp_profits)
results
```

```
##           0
## 20.000000  20.00
## 20.009994  20.01
## 20.019988  20.02
## 20.029959  20.03
## 20.039945  20.04
## ...      ...
## 25.023176  39.95
## 25.029370  39.96
## 25.079082  39.97
## 25.089585  39.98
## 25.011247  39.99
##
## [2000 rows x 1 columns]
```

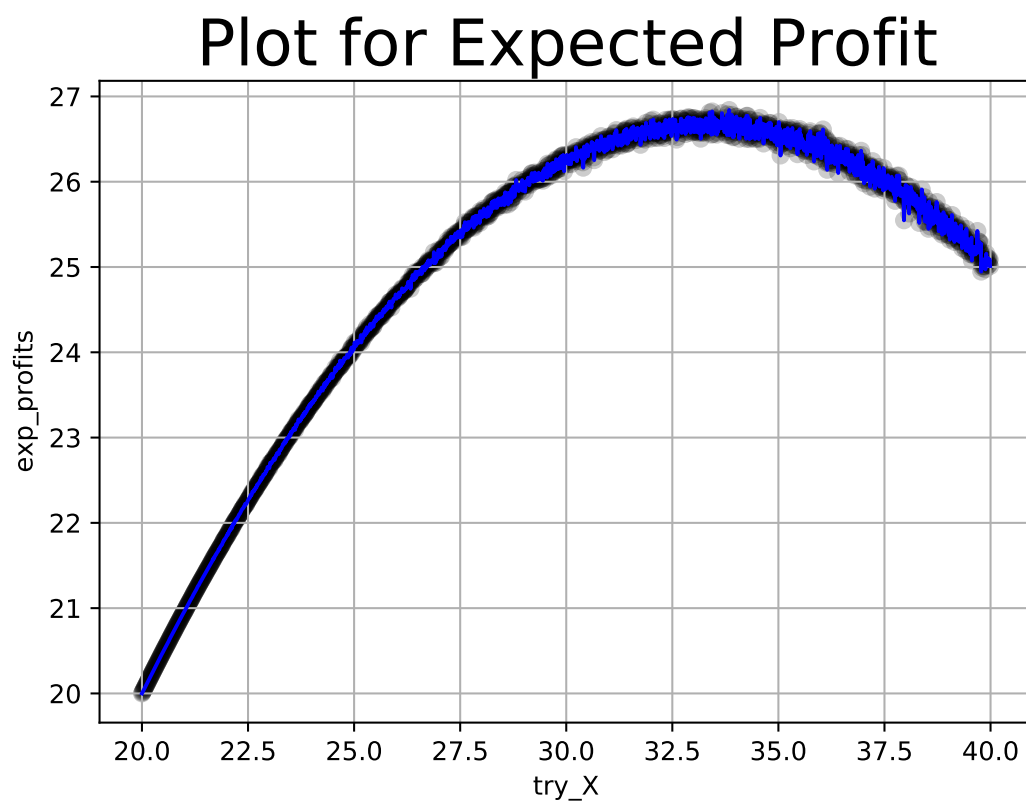
## Visualization1(Page.6)

```
plt.scatter(try_X,exp_profits,facecolors='none',edgecolors='b')  
plt.xlabel('try_X')  
plt.ylabel('exp_profits')  
plt.show()
```



## Visualizaion2(Page.7)

```
from scipy.interpolate import make_interp_spline, BSpline
plt.scatter(try_X,exp_profits,alpha=0.2,c='black')
plt.plot(try_X,exp_profits,c='b')
plt.grid(True)
plt.xlabel('try_X')
plt.ylabel('exp_profits')
plt.title('Plot for Expected Profit',fontsize=25)
plt.show()
```



## Page.8

```
idx = np.where(exp_profits == np.max(exp_profits))
try_X[idx]
```

```
## array([33.84])
```

```
exp_profits[idx]
```

```
## array([26.84089911])
```

B2.Rmd

```
"Hello"
```

```
## [1] "Hello"
```