B2_python_Jeong

Jeong, wonryeol

1/3/2021

#Implementation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
for X in range(11,15):
  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:  11.69235
## X:  13  expected profit:  12.0826
## X:  14  expected profit:  12.2087
```

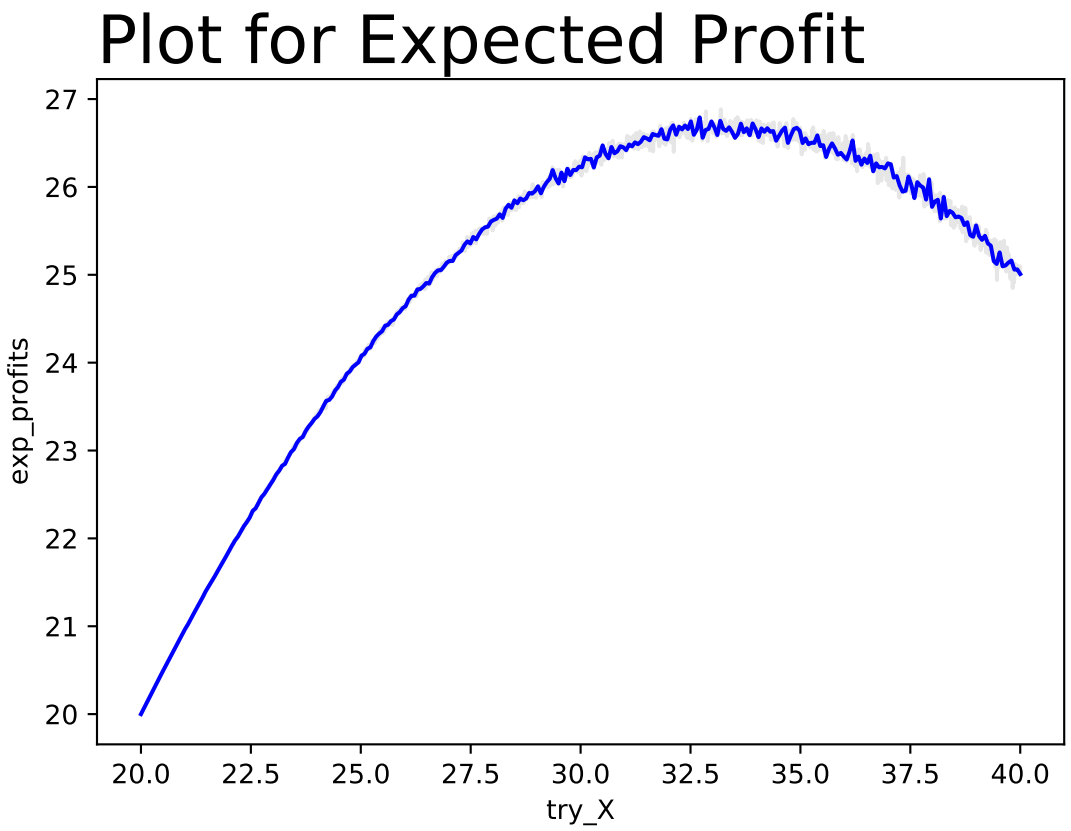
#Continuous distribution - grid search approach

```
try_X=np.arange(20,40.01,step=0.01)
exp_profits = np.array([])

for X in try_X:
  MC_N = 10000
  D = np.random.uniform(20,40,MC_N)

  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

  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    20.009997
## 2     20.02    20.019990
## 3     20.03    20.029967
## 4     20.04    20.039971
## ...     ...          ...
## 1996  39.96    25.056518
## 1997  39.97    25.052954
## 1998  39.98    25.042849
## 1999  39.99    25.095247
## 2000  40.00    25.009139
##
## [2001 rows x 2 columns]
```

```
from scipy.interpolate import make_interp_spline, BSpline


plt.plot(results['try_X'],results['exp_profits'],color='black',alpha = 0.1)

plt.title('Plot for Expected Profit', loc='left',fontsize=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) #BSpline object
power_smooth = spline(x_new)


plt.plot(x_new,power_smooth , color = 'blue')
plt.xlabel('try_X')
```

```
plt.ylabel('exp_profits')
plt.show()
```



append

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

print(try_X[idx])
```

```
## [33.19]
```

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

```
## [26.88341901]
```