Derivatives_Python_Codes

January 31, 2021

[114]:
```python
# import packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
```

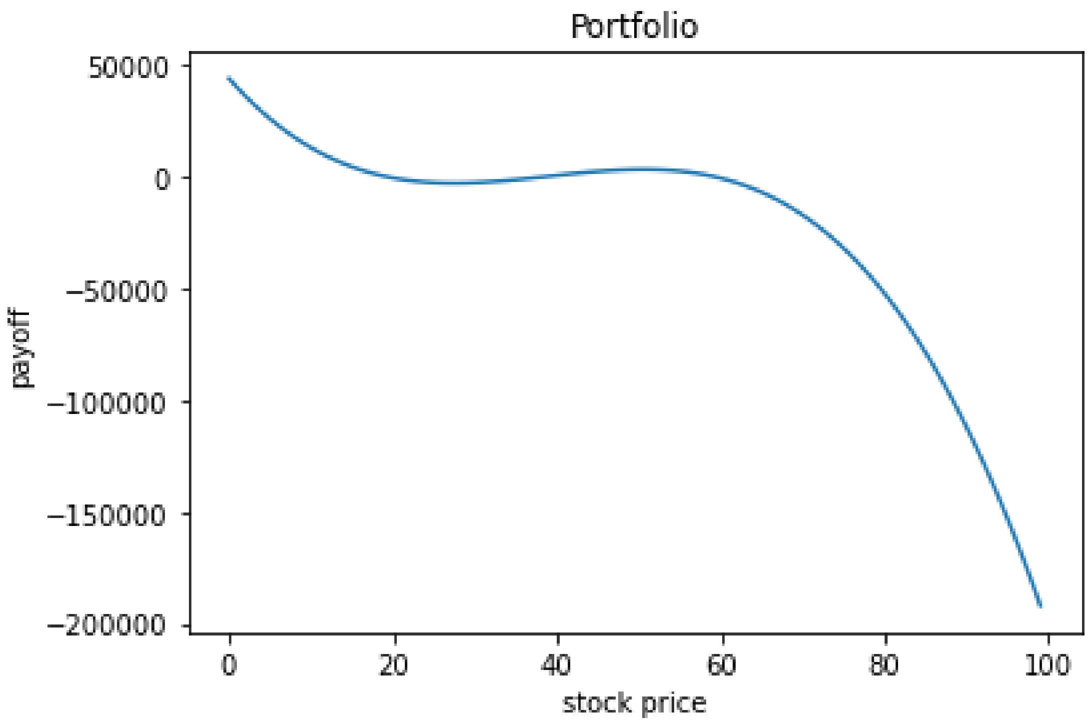
[115]:
```python
# stock is the possible stock price space
stock = np.arange(1,101)
```

[116]:
```python
# Then call and put are the possible call and put payments space
call = np.maximum(stock - np.transpose([5*np.arange(1,21)]), 0)
put = np.maximum(-stock + np.transpose([5*np.arange(1,21)]), 0)
```

[118]:
```python
# import portfolio data
# use your own file path to replace
file_path = r"C://Users/26529/Desktop/XiaoyuDu world/6 Georgia tech QCF/1␣
↪Gatech Courses/Semester 4/7 Derivatives/MATLAB_Python/portfolio_surgery_code/
↪Piecewise Linear Approximation Data.xlsx"
portfolio_dataframe = pd.read_excel(file_path)
portfolio_dataframe
```

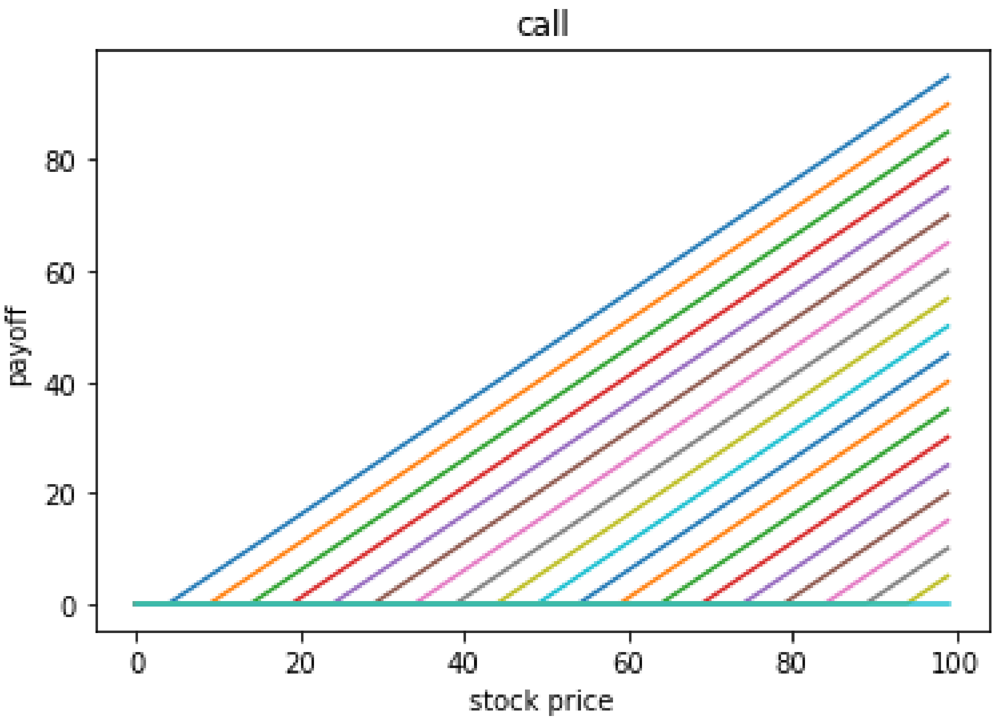
[118]:
```
    S&P  Portfolio 1  Portfolio 2
0     1        43719   263.223140
1     2        39672   197.222222
2     3        35853   145.857988
3     4        32256   105.102041
4     5        28875    72.222222
..  …            …           …
95   96      -153216  -145.550018
96   97      -162393  -145.632806
97   98      -171912  -145.713306
98   99      -181779  -145.791600
99  100      -192000  -145.867769

[100 rows x 3 columns]
```

[137]:
```
plt.plot(portfolio_dataframe['Portfolio 1'])
plt.title('Portfolio')
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.show()
```

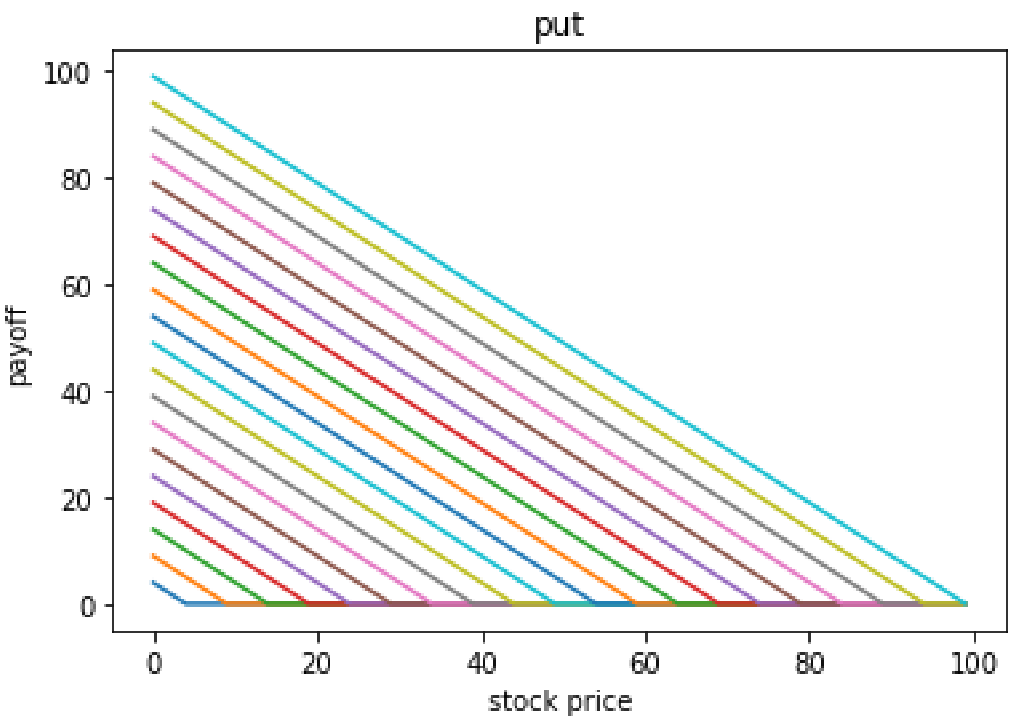


[120]:
```
# plot call and put
call, put = np.transpose(call), np.transpose(put)
```

[122]:
```
plt.plot(call)
plt.title('call')
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.show()
```

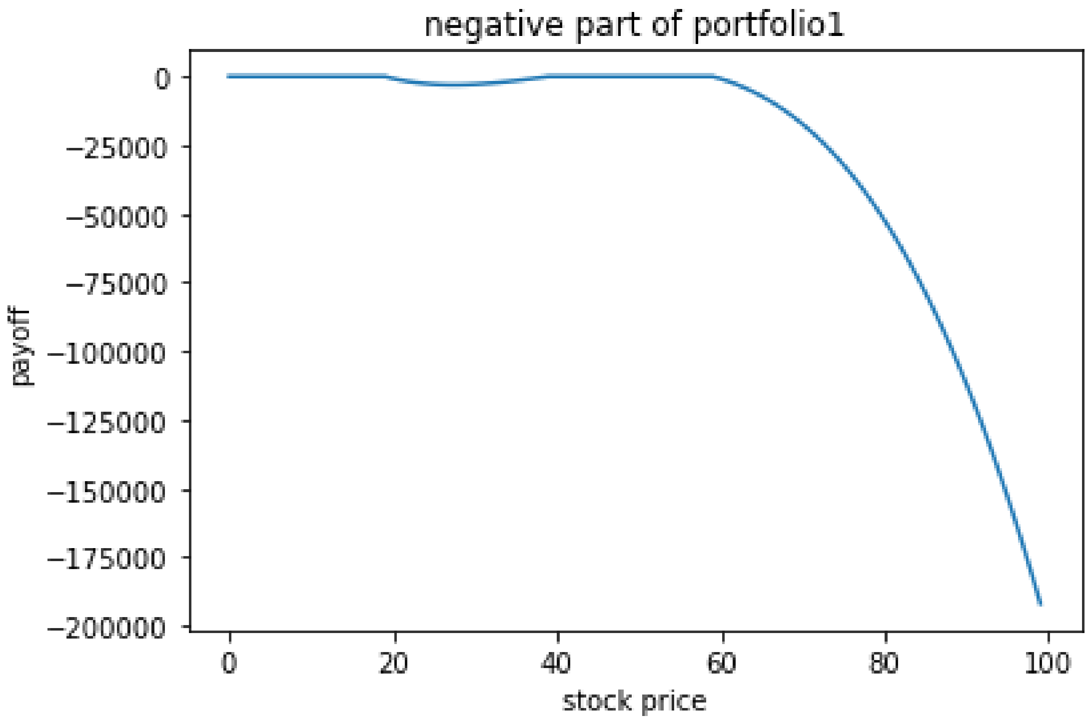


[123]:
```
plt.plot(put)
plt.title('put')
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.show()
```



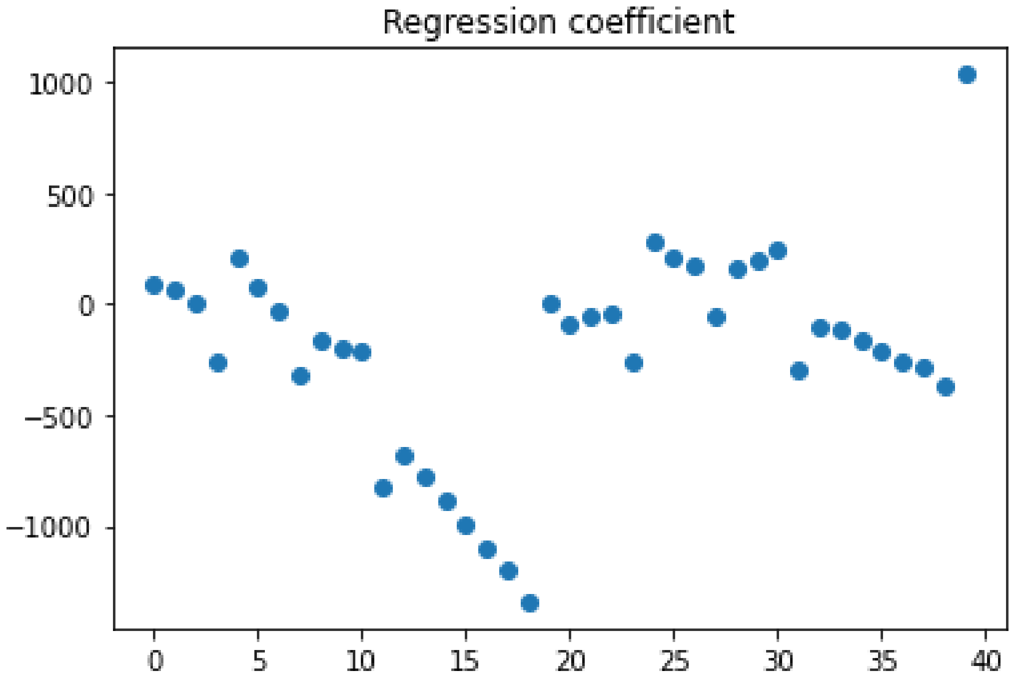
[139]:
```
# x is the independent variable, axis = 1 means concat
x = np.concatenate((call, put), axis=1)
```

[140]:
```
negative_portfolio1 = np.minimum(portfolio['Portfolio 1'], 0)
plt.plot(negative_portfolio1)
plt.title("negative part of portfolio1")
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.show()
```



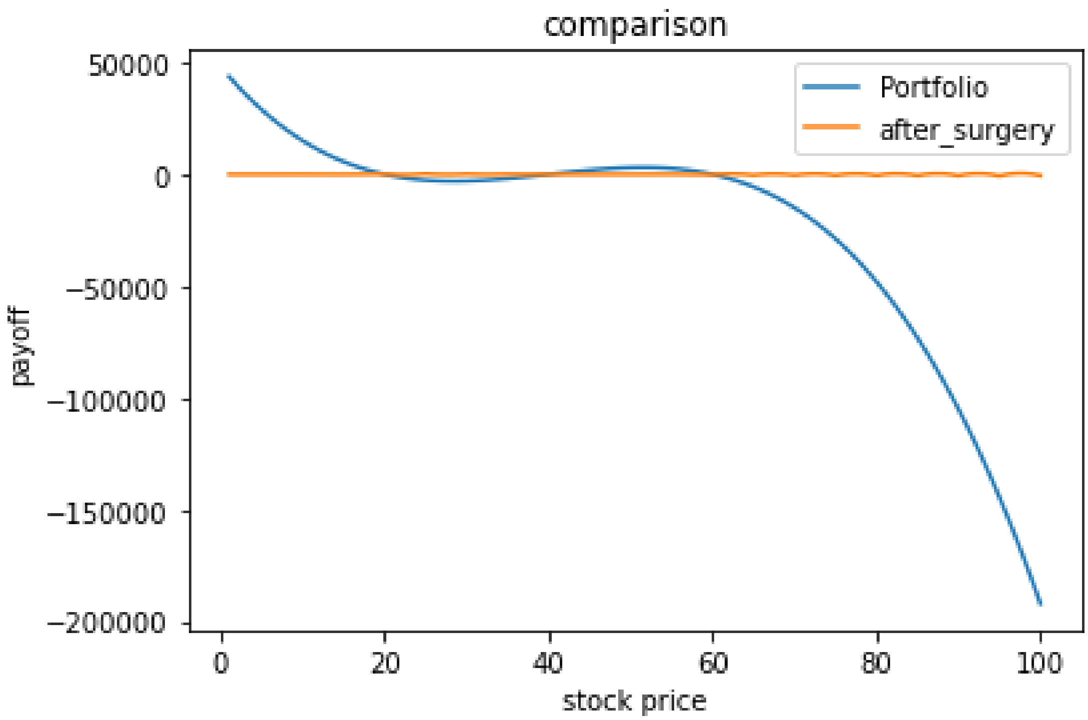
[141]:
```
model = LinearRegression(fit_intercept=False)
beta = model.fit(x, negative_portfolio1.values).coef_
plt.scatter(list(range(len(beta))), beta)
plt.title('Regression coefficient')
```

[141]: Text(0.5, 1.0, 'Regression coefficient')

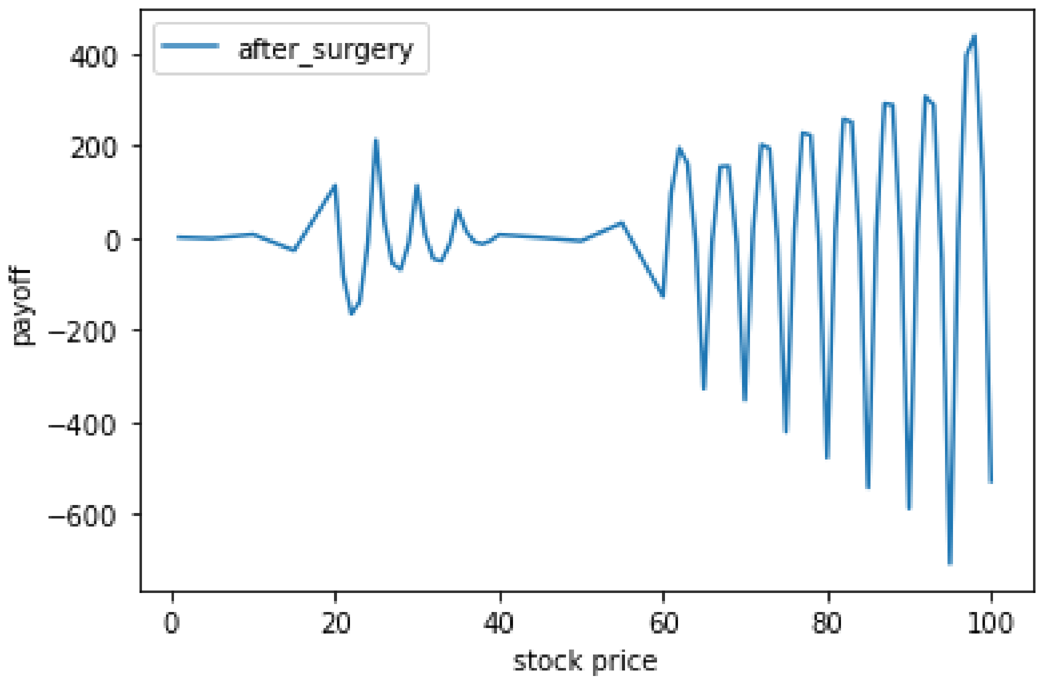


[142]:
```
aftersurgery = negative_portfolio1 - np.dot(x,beta)
```

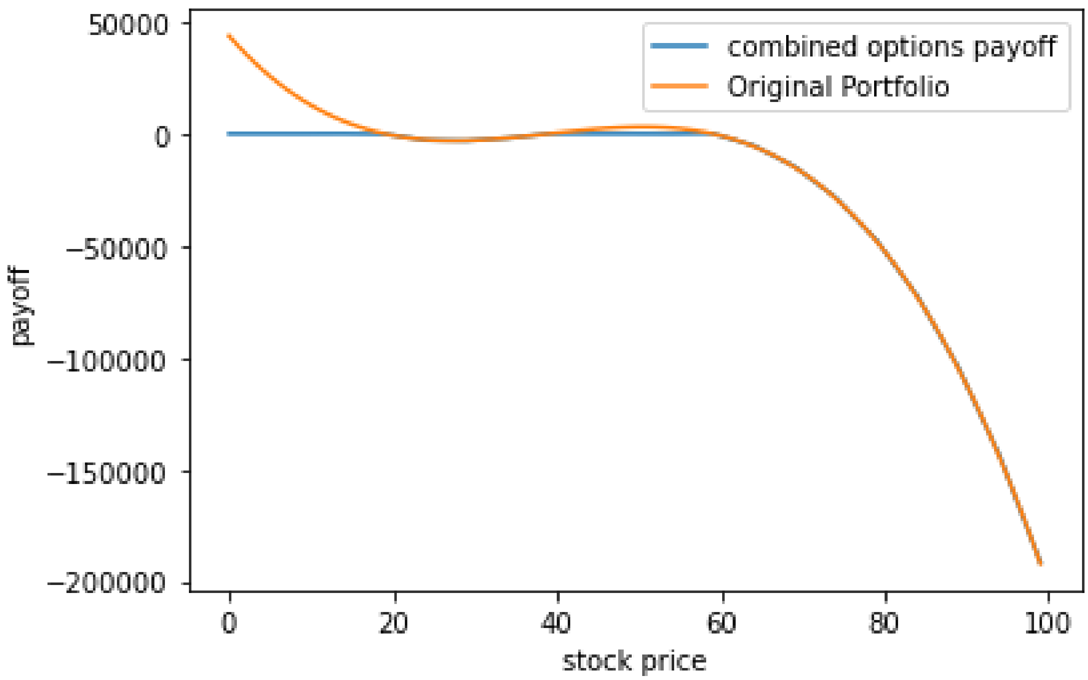
[144]:
```
plt.plot(stock,portfolio['Portfolio 1'])
plt.plot(stock, aftersurgery)
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.legend(["Portfolio","after_surgery"])
plt.title("comparison")
plt.show()
```



[145]:
```
plt.plot(stock, aftersurgery)
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.legend(["after_surgery"])
plt.show()
```



[146]:
```
option_payoff = np.dot(x,beta)
plt.xlabel('stock price')
plt.ylabel('payoff')
plt.plot(option_payoff)
plt.plot(portfolio_dataframe['Portfolio 1'])
plt.legend(["combined options payoff","Original Portfolio"])
plt.show()
```



50000
0
−50000
−100000
−150000
−200000
payoff
0
20
40
60
80
100
stock price
combined options payoff
Original Portfolio