y9i2qzz5o

March 27, 2025

[1]:
```
# UPS vs NPS Calculator

import pandas as pd
import numpy as np
```

[2]:
```
pay_matrix_df = pd.read_csv("CPC7.csv")
promotion_criteria_df = pd.read_csv("Simplified_Minimum_SL_Promotion.csv")
```

[3]:
```
promotion_criteria_df
```

[3]:
```
   From   To  Min Time (Years)
0     1    2                 3
1     2    3                 3
2     3    4                 5
3     4    5                 5
4     5    6                 6
5     6    7                 5
6     7    8                 2
7     8    9                 2
8     9   10                 2
9    10   11                 5
10   11   12                 5
11   12   13                 5
12   13  13                 2
13   13   14                 3
14  13    14                2
15   14   15                 3
16   15   16                 1
17   16   17                 1
```

[4]:
```
pay_matrix_df.drop(columns = pay_matrix_df.columns[0], inplace = True)
pay_matrix_df.index.name = "Pay Level"
```

[5]:
```
start_Pay_Level = 10
```

[6]:
```
pay_matrix_df
```

```
[6]:            1      2      3      4      5       6       7       8       9  \
Pay Level
0           18000  19900  21700  25500  29200   35400   44900   47600   53100
1           18500  20500  22400  26300  30100   36500   46200   49000   54700
2           19100  21100  23100  27100  31000   37600   47600   50500   56300
3           19700  21700  23800  27900  31900   38700   49000   52000   58000
4           20300  22400  24500  28700  32900   39900   50500   53600   59700
5           20900  23100  25200  29600  33900   41100   52000   55200   61500
6           21500  23800  26000  30500  34900   42300   53600   56900   63300
7           22100  24500  26800  31400  35900   43600   55200   58600   65200
8           22800  25200  27600  32300  37000   44900   56900   60400   67200
9           23500  26000  28400  33300  38100   46200   58600   62200   69200
10          24200  26800  29300  34300  39200   47600   60400   64100   71300
11          24900  27600  30200  35300  40400   49000   62200   66000   73400
12          25600  28400  31100  36400  41600   50500   64100   68000   75600
13          26400  29300  32000  37500  42800   52000   66000   70000   77900
14          27200  30200  33000  38600  44100   53600   68000   72100   80200
15          28000  31100  34000  39800  45400   55200   70000   74300   82600
16          28800  32000  35000  41000  46800   56900   72100   76500   85100
17          29700  33000  36100  42200  48200   58600   74300   78800   87700
18          30600  34000  37200  43500  49600   60400   76500   81200   90300
19          31500  35000  38300  44800  51100   62200   78800   83600   93000
20          32400  36100  39400  46100  52600   64100   81200   86100   95800
21          33400  37200  40600  47500  54200   66000   83600   88700   98700
22          34400  38300  41800  48900  55800   68000   86100   91400  101700
23          35400  39400  43100  50400  57500   70000   88700   94100  104800
24          36500  40600  44400  51900  59200   72100   91400   96900  107900
25          37600  41800  45700  53500  61000   74300   94100   99800  111100
26          38700  43100  47100  55100  62800   76500   96900  102800  114400
27          39900  44400  48500  56800  64700   78800   99800  105900  117800
28          41100  45700  50000  58500  66600   81200  102800  109100  121300
29          42300  47100  51500  60300  68600   83600  105900  112400  124900
30          43600  48500  53000  62100  70700   86100  109100  115800  128600
31          44900  50000  54600  64000  72800   88700  112400  119300  132500
32          46200  51500  56200  65900  75000   91400  115800  122900  136500
33          47600  53000  57900  67900  77300   94100  119300  126600  140600
34          49000  54600  59600  69900  79600   96900  122900  130400  144800
35          50500  56200  61400  72000  82000   99800  126600  134300  149100
36          52000  57900  63200  74200  84500  102800  130400  138300  153600
37          53600  59600  65100  76400  87000  105900  134300  142400  158200
38          55200  61400  67100  78700  89600  109100  138300  146700  162900
39          56900  63200  69100  81100  92300  112400  142400  151100  167800

               10       11       12        13       13A        14        15  \
Pay Level
0           56100  67700.0  78800.0  123100.0  131100.0  144200.0  182200.0
1           57800  69700.0  81200.0  126800.0  135000.0  148500.0  187700.0
```

```
2           59500   71800.0   83600.0  130600.0  139100.0  153000.0  193300.0
3           61300   74000.0   86100.0  134500.0  143300.0  157600.0  199100.0
4           63100   76200.0   88700.0  138500.0  147600.0  162300.0  205100.0
5           65000   78500.0   91400.0  142700.0  152000.0  167200.0  211300.0
6           67000   80900.0   94100.0  147000.0  156600.0  172200.0  217600.0
7           69000   83300.0   96900.0  151400.0  161300.0  177400.0  224100.0
8           71100   85800.0   99800.0  155900.0  166100.0  182700.0       NaN
9           73200   88400.0  102800.0  160600.0  171100.0  188200.0       NaN
10          75400   91100.0  105900.0  165400.0  176200.0  193800.0       NaN
11          77700   93800.0  109100.0  170400.0  181500.0  199600.0       NaN
12          80000   96600.0  112400.0  175500.0  186900.0  205600.0       NaN
13          82400   99500.0  115800.0  180800.0  192500.0  211800.0       NaN
14          84900  102500.0  119300.0  186200.0  198300.0  218200.0       NaN
15          87400  105600.0  122900.0  191800.0  204200.0       NaN       NaN
16          90000  108800.0  126600.0  197600.0  210300.0       NaN       NaN
17          92700  112100.0  130400.0  203500.0  216600.0       NaN       NaN
18          95500  115500.0  134300.0  209600.0       NaN       NaN       NaN
19          98400  119000.0  138300.0  215900.0       NaN       NaN       NaN
20         101400  122600.0  142400.0       NaN       NaN       NaN       NaN
21         104400  126300.0  146700.0       NaN       NaN       NaN       NaN
22         107500  130100.0  151100.0       NaN       NaN       NaN       NaN
23         110700  134000.0  155600.0       NaN       NaN       NaN       NaN
24         114000  138000.0  160300.0       NaN       NaN       NaN       NaN
25         117400  142100.0  165100.0       NaN       NaN       NaN       NaN
26         120900  146400.0  170100.0       NaN       NaN       NaN       NaN
27         124500  150800.0  175200.0       NaN       NaN       NaN       NaN
28         128200  155300.0  180500.0       NaN       NaN       NaN       NaN
29         132000  160000.0  185900.0       NaN       NaN       NaN       NaN
30         136000  164800.0  191500.0       NaN       NaN       NaN       NaN
31         140100  169700.0  197200.0       NaN       NaN       NaN       NaN
32         144300  174800.0  203100.0       NaN       NaN       NaN       NaN
33         148600  180000.0  209200.0       NaN       NaN       NaN       NaN
34         153100  185400.0       NaN       NaN       NaN       NaN       NaN
35         157700  191000.0       NaN       NaN       NaN       NaN       NaN
36         162400  196700.0       NaN       NaN       NaN       NaN       NaN
37         167300  202600.0       NaN       NaN       NaN       NaN       NaN
38         172300  208700.0       NaN       NaN       NaN       NaN       NaN
39         177500       NaN       NaN       NaN       NaN       NaN       NaN

                 16        17        18
Pay Level
0          205400.0  225000.0  250000.0
1          211600.0       NaN       NaN
2          217900.0       NaN       NaN
3          224400.0       NaN       NaN
4               NaN       NaN       NaN
5               NaN       NaN       NaN
```

```
6               NaN       NaN       NaN
7               NaN       NaN       NaN
8               NaN       NaN       NaN
9               NaN       NaN       NaN
10              NaN       NaN       NaN
11              NaN       NaN       NaN
12              NaN       NaN       NaN
13              NaN       NaN       NaN
14              NaN       NaN       NaN
15              NaN       NaN       NaN
16              NaN       NaN       NaN
17              NaN       NaN       NaN
18              NaN       NaN       NaN
19              NaN       NaN       NaN
20              NaN       NaN       NaN
21              NaN       NaN       NaN
22              NaN       NaN       NaN
23              NaN       NaN       NaN
24              NaN       NaN       NaN
25              NaN       NaN       NaN
26              NaN       NaN       NaN
27              NaN       NaN       NaN
28              NaN       NaN       NaN
29              NaN       NaN       NaN
30              NaN       NaN       NaN
31              NaN       NaN       NaN
32              NaN       NaN       NaN
33              NaN       NaN       NaN
34              NaN       NaN       NaN
35              NaN       NaN       NaN
36              NaN       NaN       NaN
37              NaN       NaN       NaN
38              NaN       NaN       NaN
39              NaN       NaN       NaN
```

[7]:

```
# Preprocessing

mil_service = 0  # this tracks if the employee is military person or not

if not mil_service:
    pay_matrix_df.drop(columns = ["13A"], inplace = True)
    promotion_criteria_df.drop([12, 14], inplace = True)
    #promotion_criteria_df.reset_index(drop=True,inplace = True)
    promotion_criteria_df.set_index('From', inplace=True)
```

[8]:

```
# Assumption: Promotion is applied as soon as they accrue, till Level 14 only
# Assumption: Fitment factor is kept on conservative side of 1.7 for each␣
 ↪prospective Pay Commission

def determine_basic_pay(service_length, start_level, pay_matrix,␣
 ↪promotion_criteria):
    current_level_str = promotion_criteria.index[start_level-1]
    current_level_num = start_level
    total_years_completed = 0
    years_spent_each_level = 0
    current_salary = pay_matrix[current_level_str].iloc[years_spent_each_level]
    num_decades_completed = 0 # for Pay Commission effect, assuming that␣
 ↪payment is reset 1.7 times the last pay matrix


    while total_years_completed < service_length:
        # Check if eligible for promotion
        num_decades_completed = int(total_years_completed/10)
        fitment_factor = (1.7)**num_decades_completed
        if (current_level_num < 14) : # change this number above which you␣
 ↪would have no promotion (<16)
            if years_spent_each_level == promotion_criteria["Min Time␣
 ↪(Years)"][current_level_str]:
                current_level_num += 1
                current_level_str = list(promotion_criteria.
 ↪index)[current_level_num-1]
                result = pay_matrix[pay_matrix[current_level_str] >␣
 ↪current_salary][current_level_str]
                if len(list(result))>0:
                    current_salary = fitment_factor*list(result)[0]
                else:
                    current_salary =␣
 ↪fitment_factor*pay_matrix[current_level_str][0]
                years_spent_each_level = 0
        else:
            # Move down by cell for annual increment if there's still a cell␣
 ↪down
            if total_years_completed + 1 < len(pay_matrix[current_level].
 ↪dropna()):
                current_salary = fitment_factor*pay_matrix[current_level].
 ↪iloc[years_completed+1]

        years_spent_each_level+= 1
        total_years_completed += 1

    return current_salary, current_level_str
```

[9]:

```
service_years = 33
start_pay_level = 10
da_rate = 5 # 5% annually

sal_df = pd.DataFrame(columns=["Year", "Level", "Basic Pay", "DA"]) #

for i in range(service_years):
    current_salary, current_level = determine_basic_pay(i, start_pay_level,␣
 ↪pay_matrix_df, promotion_criteria_df)
    temp = pd.DataFrame({"Year":[i], "Level": [current_level] , "Basic Pay":␣
 ↪[current_salary], "DA":[(i%11)*da_rate]})
    sal_df = pd.concat([sal_df,temp])
    print("Year: ", i, "Current Basic Pay:", current_salary, "Current Level:",␣
 ↪current_level, "DA Rate (%):", (i%11)*da_rate)
```

```
Year:  0 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 0
Year:  1 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 5
Year:  2 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 10
Year:  3 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 15
Year:  4 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 20
Year:  5 Current Basic Pay: 56100 Current Level: 10 DA Rate (%): 25
Year:  6 Current Basic Pay: 67700.0 Current Level: 11 DA Rate (%): 30
Year:  7 Current Basic Pay: 67700.0 Current Level: 11 DA Rate (%): 35
Year:  8 Current Basic Pay: 67700.0 Current Level: 11 DA Rate (%): 40
Year:  9 Current Basic Pay: 67700.0 Current Level: 11 DA Rate (%): 45
Year:  10 Current Basic Pay: 67700.0 Current Level: 11 DA Rate (%): 50
Year:  11 Current Basic Pay: 133960.0 Current Level: 12 DA Rate (%): 0
Year:  12 Current Basic Pay: 133960.0 Current Level: 12 DA Rate (%): 5
Year:  13 Current Basic Pay: 133960.0 Current Level: 12 DA Rate (%): 10
Year:  14 Current Basic Pay: 133960.0 Current Level: 12 DA Rate (%): 15
Year:  15 Current Basic Pay: 133960.0 Current Level: 12 DA Rate (%): 20
Year:  16 Current Basic Pay: 228650.0 Current Level: 13 DA Rate (%): 25
Year:  17 Current Basic Pay: 228650.0 Current Level: 13 DA Rate (%): 30
Year:  18 Current Basic Pay: 228650.0 Current Level: 13 DA Rate (%): 35
Year:  19 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 40
Year:  20 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 45
Year:  21 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 50
Year:  22 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 0
Year:  23 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 5
Year:  24 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 10
Year:  25 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 15
Year:  26 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 20
Year:  27 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 25
Year:  28 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 30
Year:  29 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 35
Year:  30 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 40
Year:  31 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 45
```

```
Year:  32 Current Basic Pay: 245140.0 Current Level: 14 DA Rate (%): 50
```

[10]:

```
sal_df
```

[10]:

```
   Year Level Basic Pay  DA
0     0    10     56100   0
0     1    10     56100   5
0     2    10     56100  10
0     3    10     56100  15
0     4    10     56100  20
0     5    10     56100  25
0     6    11   67700.0  30
0     7    11   67700.0  35
0     8    11   67700.0  40
0     9    11   67700.0  45
0    10    11   67700.0  50
0    11    12  133960.0   0
0    12    12  133960.0   5
0    13    12  133960.0  10
0    14    12  133960.0  15
0    15    12  133960.0  20
0    16    13  228650.0  25
0    17    13  228650.0  30
0    18    13  228650.0  35
0    19    14  245140.0  40
0    20    14  245140.0  45
0    21    14  245140.0  50
0    22    14  245140.0   0
0    23    14  245140.0   5
0    24    14  245140.0  10
0    25    14  245140.0  15
0    26    14  245140.0  20
0    27    14  245140.0  25
0    28    14  245140.0  30
0    29    14  245140.0  35
0    30    14  245140.0  40
0    31    14  245140.0  45
0    32    14  245140.0  50
```

[11]:

```
sal_df["NPS"] = (sal_df["Basic Pay"]*12*(1+sal_df["DA"]/100))*(0.1+0.14)
```

[12]:

```
sal_df["UPS"] = (sal_df["Basic Pay"]*12*(1+sal_df["DA"]/100))*(0.1+0.1)
```

[13]:

```
sal_df["NPS_val_at_Ret"] = sal_df["NPS"]*(1.1)**(30-sal_df["Year"])
```

[14]:

```
sal_df["UPS_val_at_Ret"] = sal_df["UPS"]*(1.1)**(30-sal_df["Year"])
```

```
[15]: sal_df["NPS_val_at_Ret"].sum()
```

```
[15]: 66726055.95629886
```

```
[16]: sal_df["UPS_val_at_Ret"].sum()
```

```
[16]: 55605046.63024907
```

```
[17]: sal_df.astype(float).round(0)
```

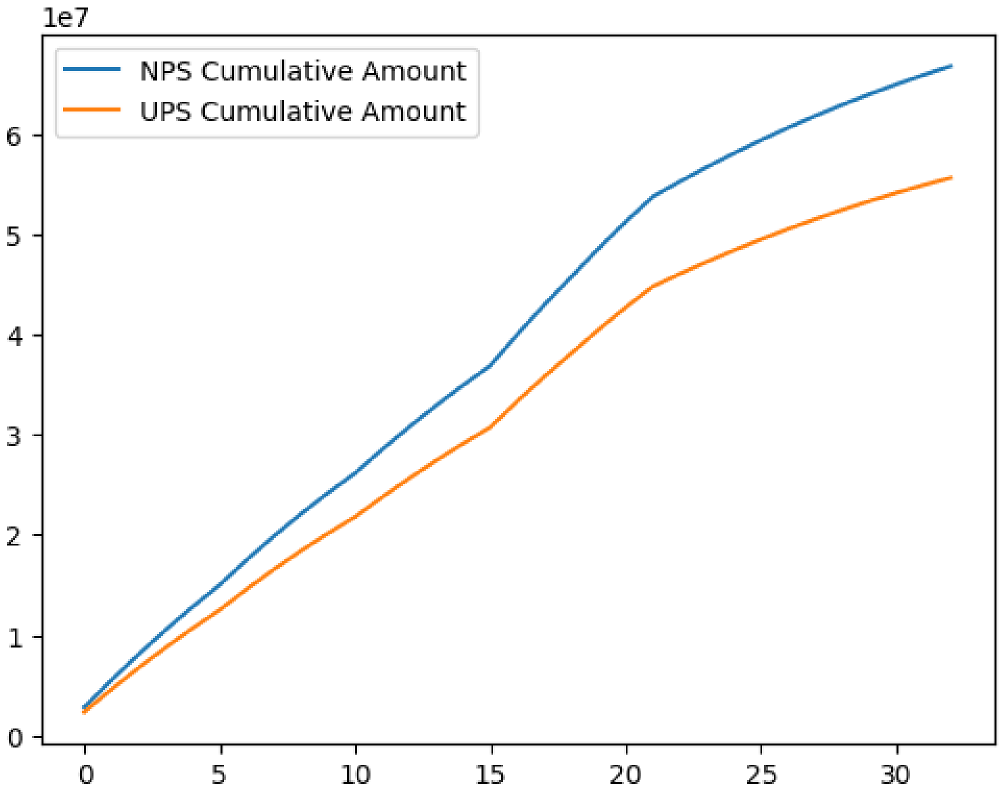
```
[17]:    Year  Level  Basic Pay    DA        NPS       UPS  NPS_val_at_Ret  \
      0   0.0   10.0    56100.0   0.0   161568.0  134640.0       2819265.0
      0   1.0   10.0    56100.0   5.0   169646.0  141372.0       2691117.0
      0   2.0   10.0    56100.0  10.0   177725.0  148104.0       2562968.0
      0   3.0   10.0    56100.0  15.0   185803.0  154836.0       2435879.0
      0   4.0   10.0    56100.0  20.0   193882.0  161568.0       2310715.0
      0   5.0   10.0    56100.0  25.0   201960.0  168300.0       2188177.0
      0   6.0   11.0    67700.0  30.0   253469.0  211224.0       2496600.0
      0   7.0   11.0    67700.0  35.0   263218.0  219348.0       2356930.0
      0   8.0   11.0    67700.0  40.0   272966.0  227472.0       2222022.0
      0   9.0   11.0    67700.0  45.0   282715.0  235596.0       2092163.0
      0  10.0   11.0    67700.0  50.0   292464.0  243720.0       1967552.0
      0  11.0   12.0   133960.0   0.0   385805.0  321504.0       2359547.0
      0  12.0   12.0   133960.0   5.0   405095.0  337579.0       2252295.0
      0  13.0   12.0   133960.0  10.0   424385.0  353654.0       2145043.0
      0  14.0   12.0   133960.0  15.0   443676.0  369730.0       2038677.0
      0  15.0   12.0   133960.0  20.0   462966.0  385805.0       1933923.0
      0  16.0   13.0   228650.0  25.0   823140.0  685950.0       3125873.0
      0  17.0   13.0   228650.0  30.0   856066.0  713388.0       2955371.0
      0  18.0   13.0   228650.0  35.0   888991.0  740826.0       2790035.0
      0  19.0   14.0   245140.0  40.0   988404.0  823670.0       2820033.0
      0  20.0   14.0   245140.0  45.0  1023705.0  853087.0       2655226.0
      0  21.0   14.0   245140.0  50.0  1059005.0  882504.0       2497078.0
      0  22.0   14.0   245140.0   0.0   706003.0  588336.0       1513381.0
      0  23.0   14.0   245140.0   5.0   741303.0  617753.0       1444591.0
      0  24.0   14.0   245140.0  10.0   776604.0  647170.0       1375801.0
      0  25.0   14.0   245140.0  15.0   811904.0  676586.0       1307579.0
      0  26.0   14.0   245140.0  20.0   847204.0  706003.0       1240391.0
      0  27.0   14.0   245140.0  25.0   882504.0  735420.0       1174613.0
      0  28.0   14.0   245140.0  30.0   917804.0  764837.0       1110543.0
      0  29.0   14.0   245140.0  35.0   953104.0  794254.0       1048415.0
      0  30.0   14.0   245140.0  40.0   988404.0  823670.0        988404.0
      0  31.0   14.0   245140.0  45.0  1023705.0  853087.0        930641.0
      0  32.0   14.0   245140.0  50.0  1059005.0  882504.0        875211.0

         UPS_val_at_Ret
      0       2349388.0
```

```
0    2242597.0
0    2135807.0
0    2029899.0
0    1925596.0
0    1823481.0
0    2080500.0
0    1964108.0
0    1851685.0
0    1743469.0
0    1639626.0
0    1966289.0
0    1876912.0
0    1787536.0
0    1698898.0
0    1611602.0
0    2604894.0
0    2462809.0
0    2325029.0
0    2350028.0
0    2212688.0
0    2080898.0
0    1261150.0
0    1203825.0
0    1146500.0
0    1089649.0
0    1033659.0
0     978844.0
0     925453.0
0     873679.0
0     823670.0
0     775534.0
0     729342.0
```

[18]:

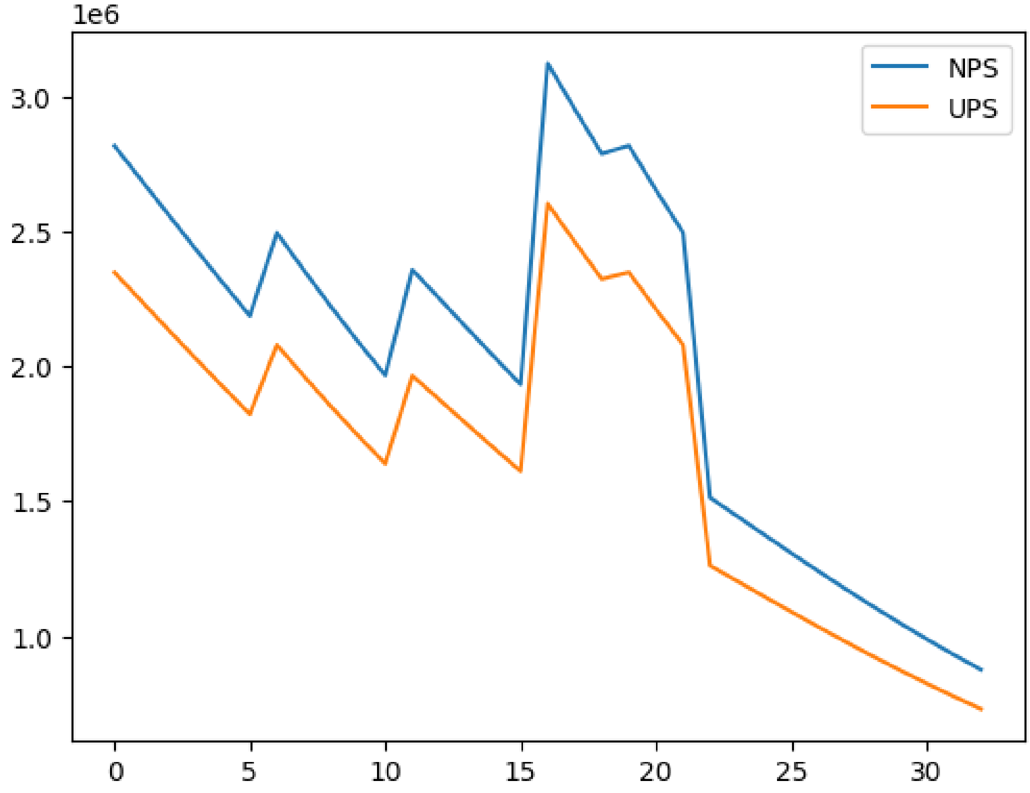
```python
import matplotlib.pyplot as plt
plt.plot(sal_df["Year"], np.cumsum(sal_df["NPS_val_at_Ret"]), label="NPS␣
  ↪Cumulative Amount")
plt.plot(sal_df["Year"], np.cumsum(sal_df["UPS_val_at_Ret"]), label="UPS␣
  ↪Cumulative Amount")
plt.legend()
plt.show()
```



[19]:

```
import matplotlib.pyplot as plt

plt.plot(sal_df["Year"], sal_df["NPS_val_at_Ret"], label="NPS")
plt.plot(sal_df["Year"], sal_df["UPS_val_at_Ret"], label="UPS")
plt.legend()
plt.show()
```



[20]:

```
# Now lets compare the pension available under both the options
# A1: 60% of the corpus is withdrawn at the retirement under NPS
# A2:

# UPS Calculations
P = sal_df["Basic Pay"].iloc[-1] #average of Basic Pay for last twelve months
Q = service_years*12 # months of qualifying service

if Q > 300:
    Q = 300
FWP = 0 # Final Withdrawal Percentage

Assured_Payout = (P/2)*(Q/300)
Admissible_Payout = Assured_Payout* (1)*(1-FWP)

Dearness_Relief= sal_df["DA"].iloc[-1]
Total_Monthly_Pension = Admissible_Payout*(1+Dearness_Relief/100)
UPS_Lumpsum_Amount = FWP*sal_df["UPS_val_at_Ret"].sum() + int(service_years/
  ↪2)*(P/10)*(1+Dearness_Relief/100)
```

[21]:

```python
def NPV(monthly_amount, years, discount_rate):
    value = 0
    for i in range(years*12):
        value += monthly_amount/(1+discount_rate/12)**i
    return value
```

[22]:

```python
Assured_Payout
```

[22]:

```
122570.0
```

[23]:

```python
UPS_Lumpsum_Amount
```

[23]:

```
588336.0
```

[24]:

```python
Admissible_Payout
```

[24]:

```
122570.0
```

[25]:

```python
# Total NPV, assuming 20 years of pension after retirement
UPS_Total_NPV = UPS_Lumpsum_Amount + NPV(Admissible_Payout, 20, 0.1)
UPS_Total_NPV
```

[25]:

```
13395449.425646387
```

[26]:

```python
# NPS Calculations
NPS_Lumpsum = 0.6*sal_df["NPS_val_at_Ret"].sum()
Annuity = 0.4*sal_df["NPS_val_at_Ret"].sum()

Annuity_Rate = 6 # percent per annum
Monthly_pension = (Annuity*Annuity_Rate/100)/12
```

[27]:

```python
NPS_Lumpsum
```

[27]:

```
40035633.573779315
```

[28]:

```python
Monthly_pension
```

[28]:

```
133452.1119125977
```

[29]:

```python
# Total NPV, assuming 20 years of pension after retirement
NPS_Total_NPV = NPS_Lumpsum + NPV(Monthly_pension, 20, 0.1)
NPS_Total_NPV
```

[29]:

```
53979798.819407895
```

[30]:

```python
NPS_Total_NPV - UPS_Total_NPV
```

```
[30]: 40584349.39376151
```

```
[ ]:
```

```
[ ]:
```