Pathuri Sai Harish (SR # 21747)

D/L Method: Run Production Functions

Methodology

The code aims to use the Duckworth-Lewis (D/L) method to compute the resources remaining in a cricket match given the overs-remaining and wickets-in-hand. The code first loads the cricket match data from a CSV file [1] using Pandas. It then applies some cleaning to the data to include only completed games, not interrupted ones. This means the games that were played 50 overs in the first innings or less than 50 overs with 0 wickets remaining are included. The cleaned data contains 59201 rows and is then stored in a separate data frame. The code then defines the D/L model using the parameters Z0 and L (a total of 11 parameters). In the D/L system, the Z0 and L values are estimated through an optimization algorithm that minimizes the normalized sum of squared errors between the predicted and actual runs. The class DLModel contains methods to define the Z-function, compute the error function (normalized per wickets_remaining, overs, and runs), and optimize the error function to obtain the optimal values of Z0s and L subject to constraints as suggested in the question. The class DataUtil contains utility methods to extract the match total runs and plot the graphs. The code finally plots two graphs, the average runs obtainable through the D/L method, and the resources remaining through the DL method using the optimal values of Z0s and L.

Results

Experimented with various optimization methods provided by the scipy.optimize.minimize() module [2]. The normalized squared error for each optimization method is as follows:

Optimization Algorithm	Normalized Square Error
SLSQP	**4989.935425213922**
trust-constr	4994.192160361856
COBYLA	5077.119336230687

Table 1. Optimization Algorithms vs Normalized Square Error

Thus the SLSQP is used for getting the optimal Z0s and L.

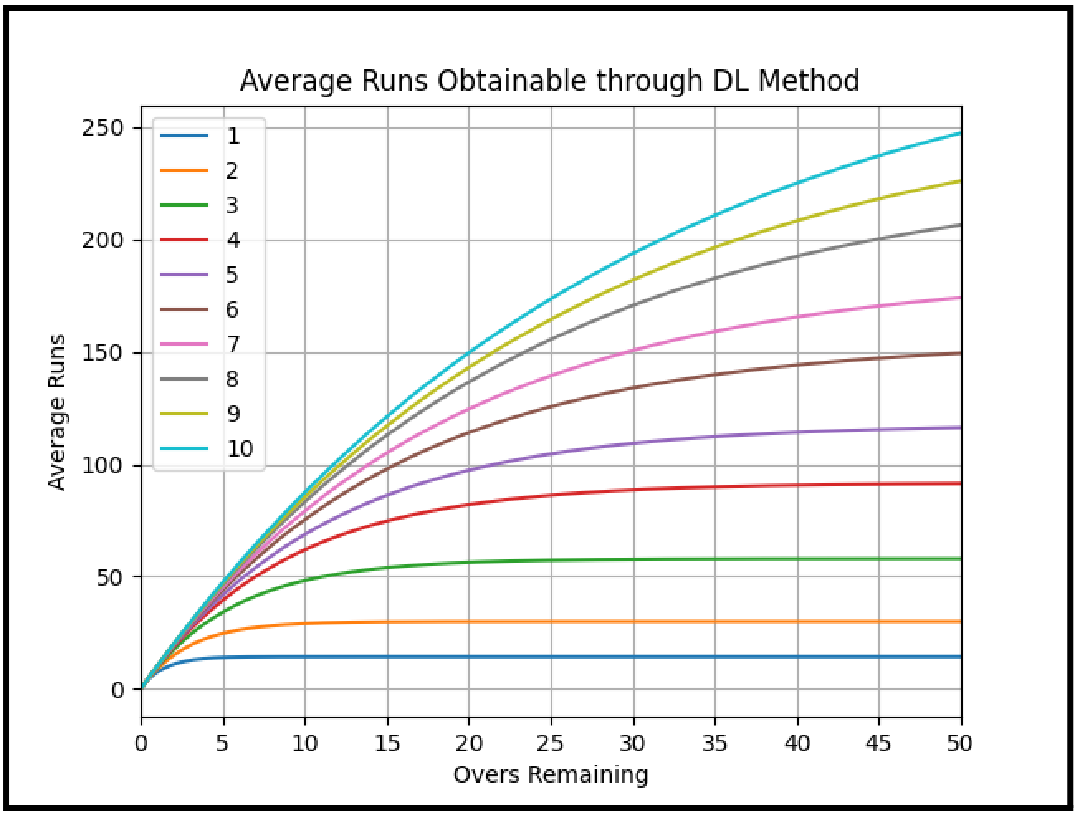
The Z0s for wickets_remaining 1 through 10 are as follows (Trimmed to three decimals):

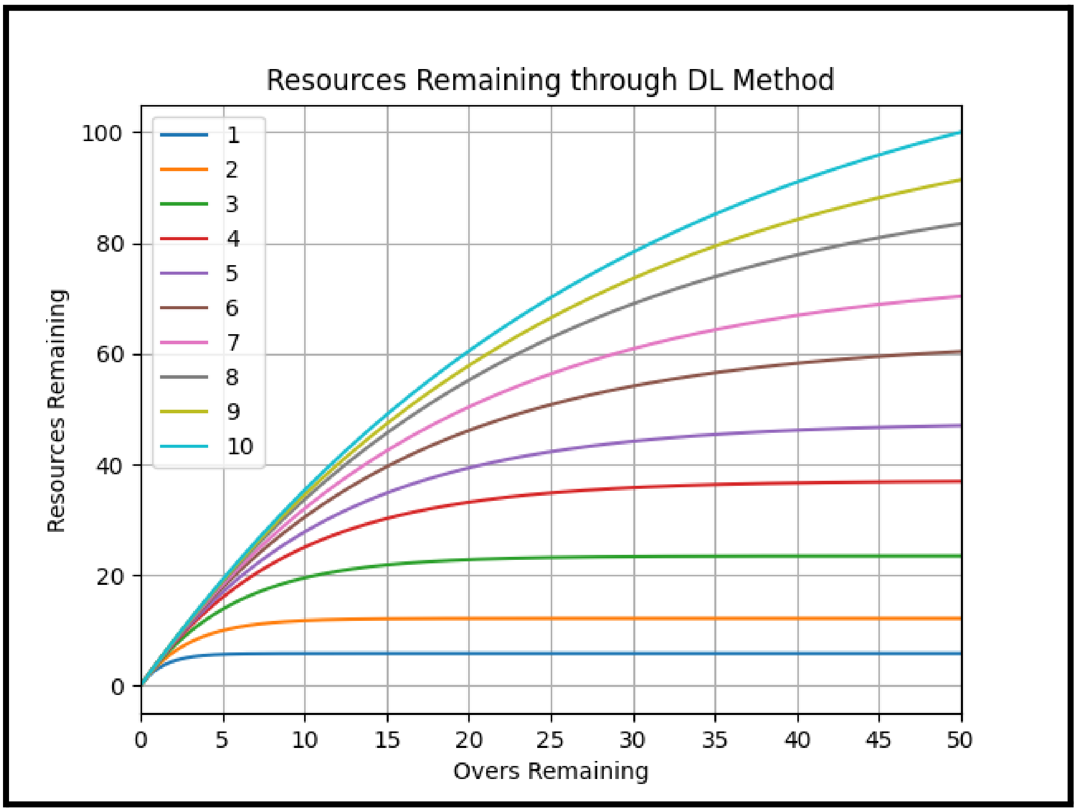
Z0(1)	Z0(2)	Z0(3)	Z0(4)	Z0(5)	Z0(6)	Z0(7)	Z0(8)	Z0(9)	Z0(10)
14.499	30.124	57.986	91.638	117.687	154.846	185.492	231.228	263.007	302.085

Table 2. Z0s for wickets_remaining 1 to 10

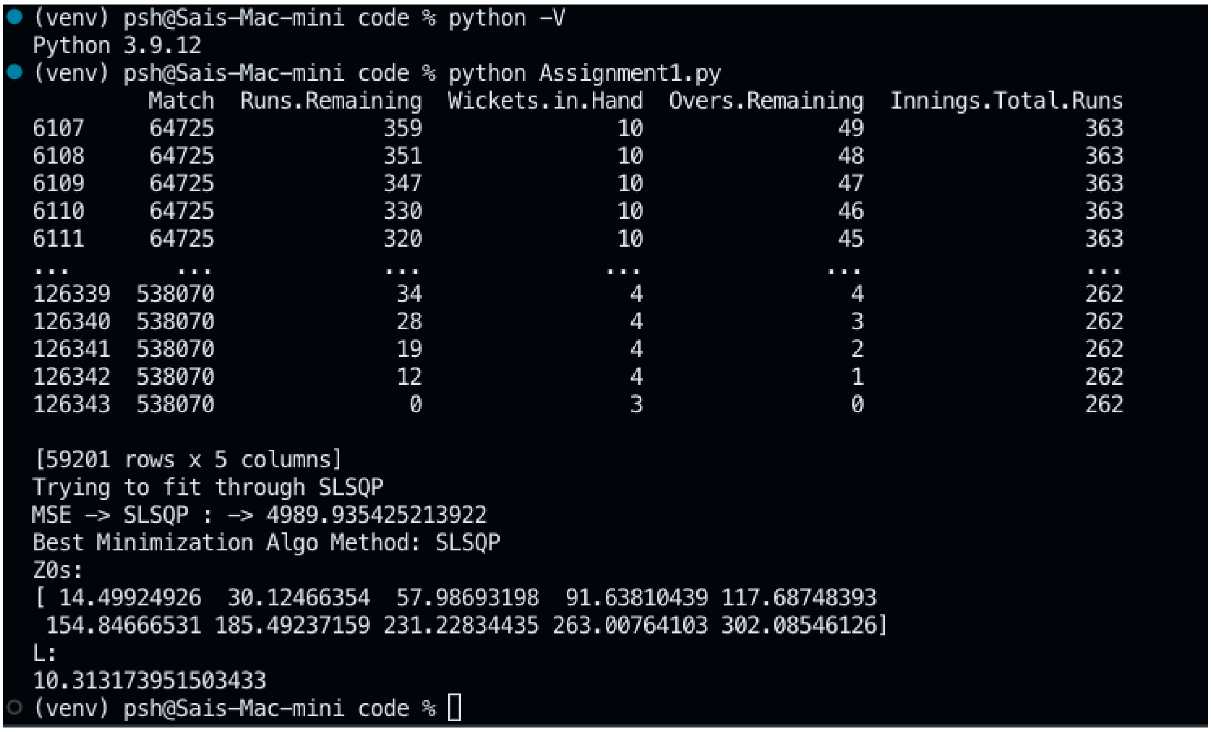
The L value is **10.313.**

Below are the plots for the average runs and resources remaining percentage through SLSQP:





Running Assignment1.py on my system gives the following output:



```
(venv) psh@Sais-Mac-mini code % python -V
Python 3.9.12
(venv) psh@Sais-Mac-mini code % python Assignment1.py
         Match  Runs.Remaining  Wickets.in.Hand  Overs.Remaining  Innings.Total.Runs
6107     64725             359               10               49                 363
6108     64725             351               10               48                 363
6109     64725             347               10               47                 363
6110     64725             330               10               46                 363
6111     64725             320               10               45                 363
...        ...             ...              ...              ...                 ...
126339  538070              34                4                4                 262
126340  538070              28                4                3                 262
126341  538070              19                4                2                 262
126342  538070              12                4                1                 262
126343  538070               0                3                0                 262

[59201 rows x 5 columns]
Trying to fit through SLSQP
MSE -> SLSQP : -> 4989.935425213922
Best Minimization Algo Method: SLSQP
Z0s:
[ 14.49924926  30.12466354  57.98693198  91.63810439 117.68748393
 154.84666531 185.49237159 231.22834435 263.00764103 302.08546126]
L:
10.313173951503433
(venv) psh@Sais-Mac-mini code % ▯
```

References

1. *https://ece.iisc.ac.in/~rajeshs/E0259/04_cricket_1999to2011.csv*
2. *https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html*