VG441 Midterm

Anna Li
Student ID: 518370910048

Problem 1(Forecasting)

(a) Scatter plot the demands against time (Figure 1).

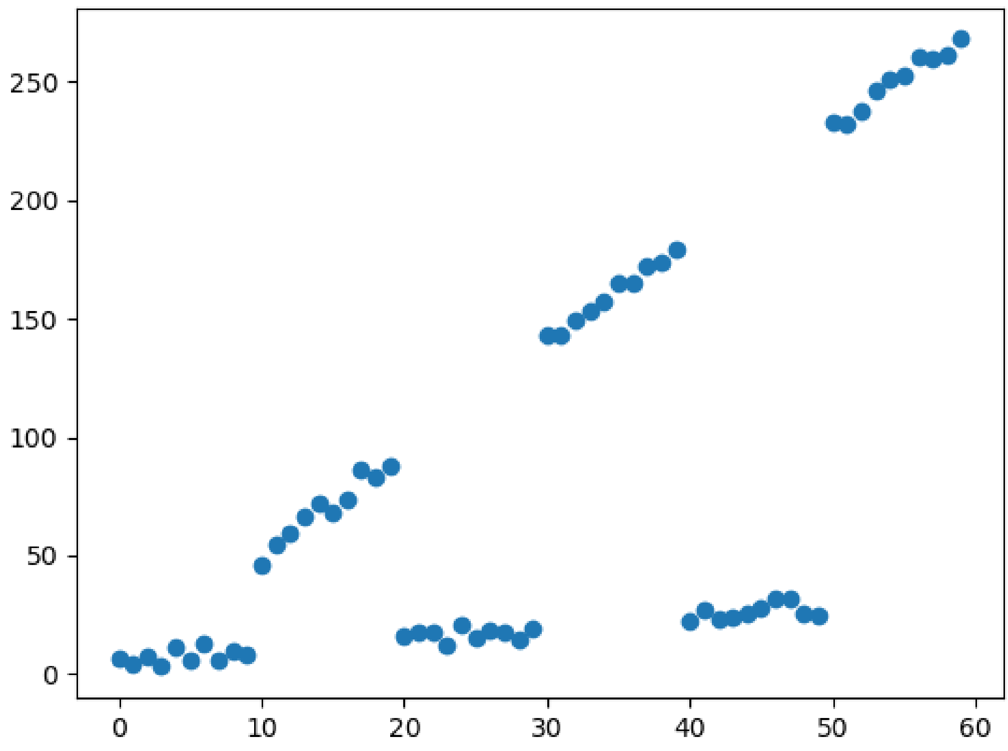

Fig. 1: demands of time

Figure 1 is the demands against time

(b) Run a simple regression and plot your results on top of scatter plot (Figure 2).

First, judging from the plot we got, we should use ARIMA model to forecast the data and get:

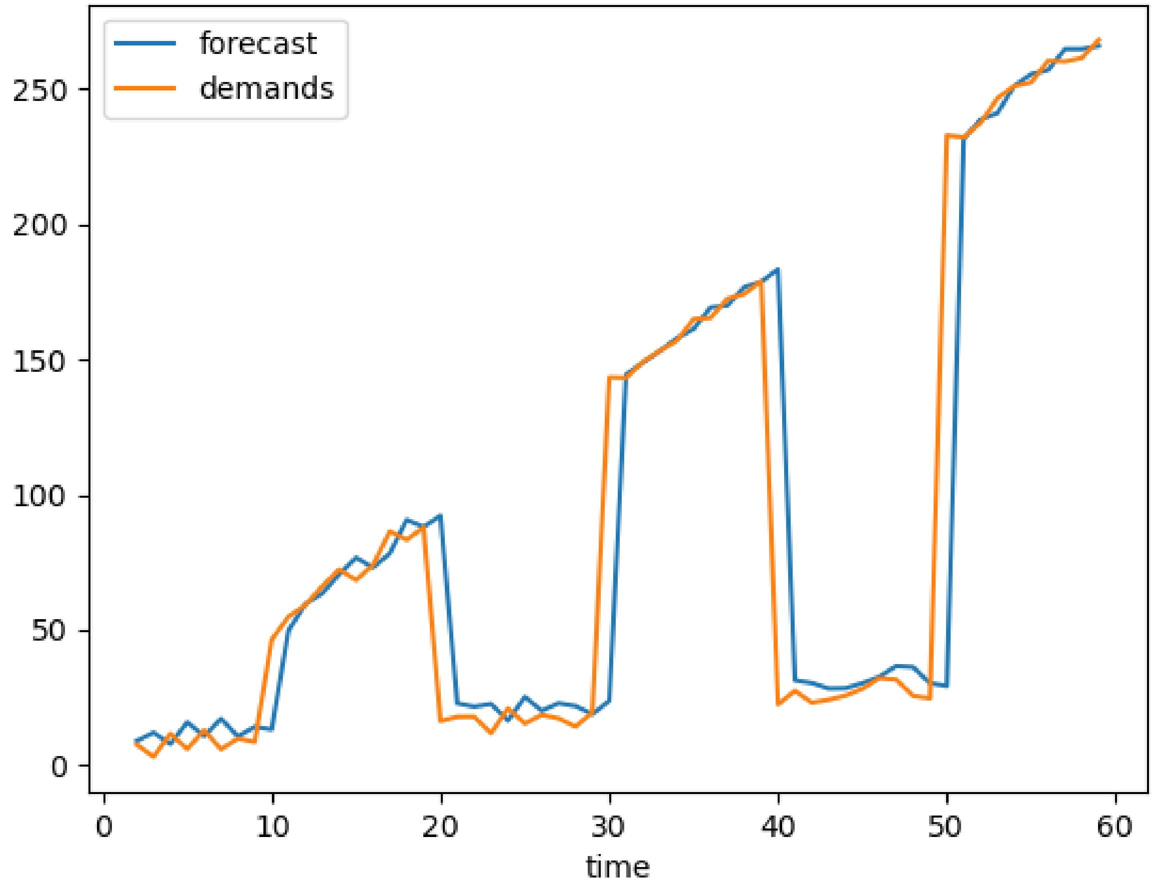

Fig. 2: ARIMA model of demands

(c) Run gradient boosting method with different number of trees:

1. params = 'n_estimators': 1, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

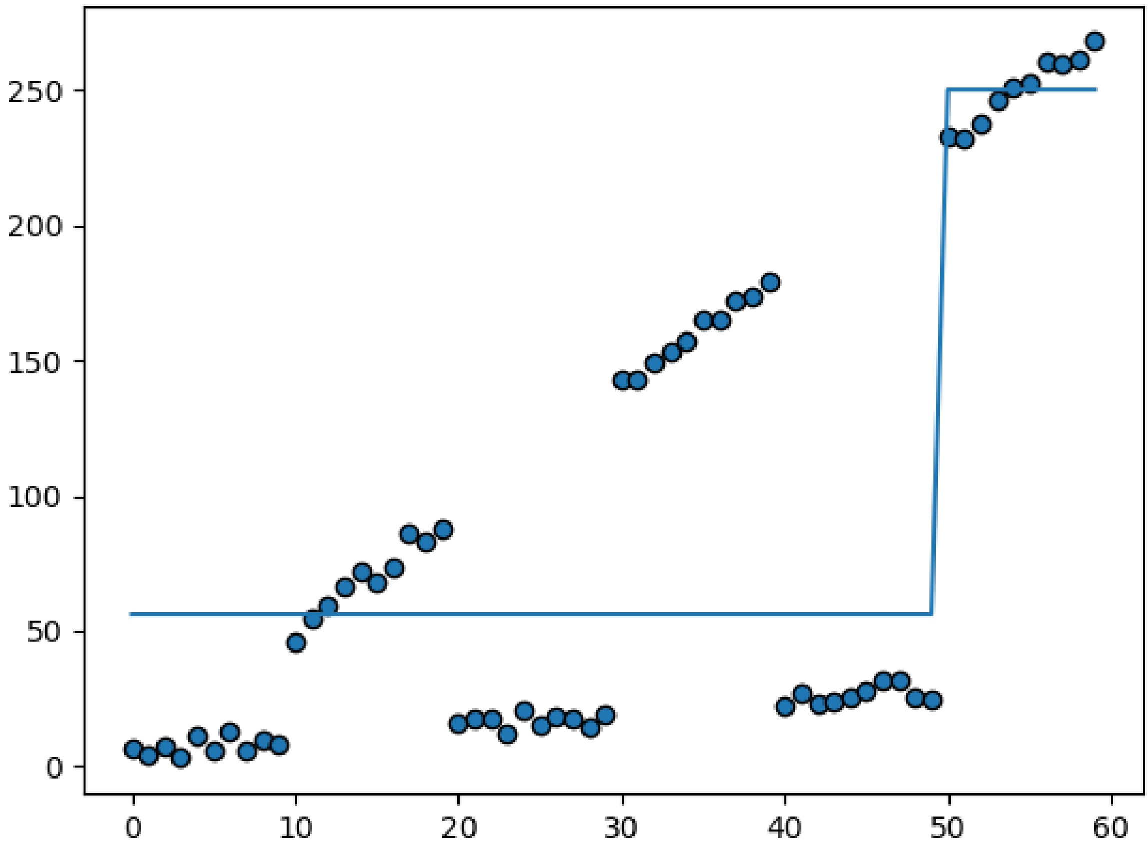

Fig. 3: results of param:1,1,1,'ls'

Just as Fig.4 shows, and R2 sq: 0.6959704096715094

2. params = 'n_estimators': 2, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

Just as Fig.4 shows, and R2 sq: 0.7290384207566345

3. params = 'n_estimators': 5, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

Just as Fig.5 shows, and R2 sq: 0.8860240765658012

4. params = 'n_estimators': 10, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

Just as Fig.6 shows, and R2 sq: 0.9751642096479143

5. params = 'n_estimators': 20, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

Just as Fig.7 shows, and R2 sq: 0.9875429979290639

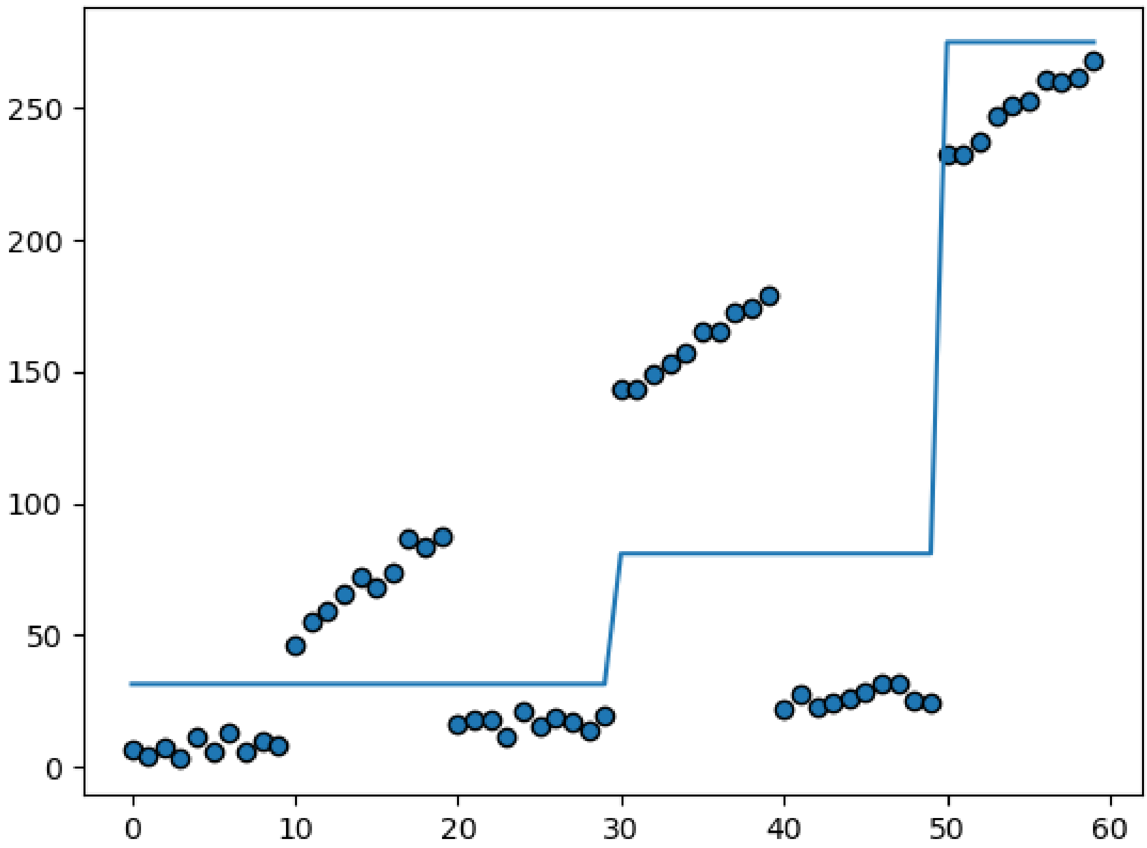

Fig. 4: results of param:2,1,1,'ls'

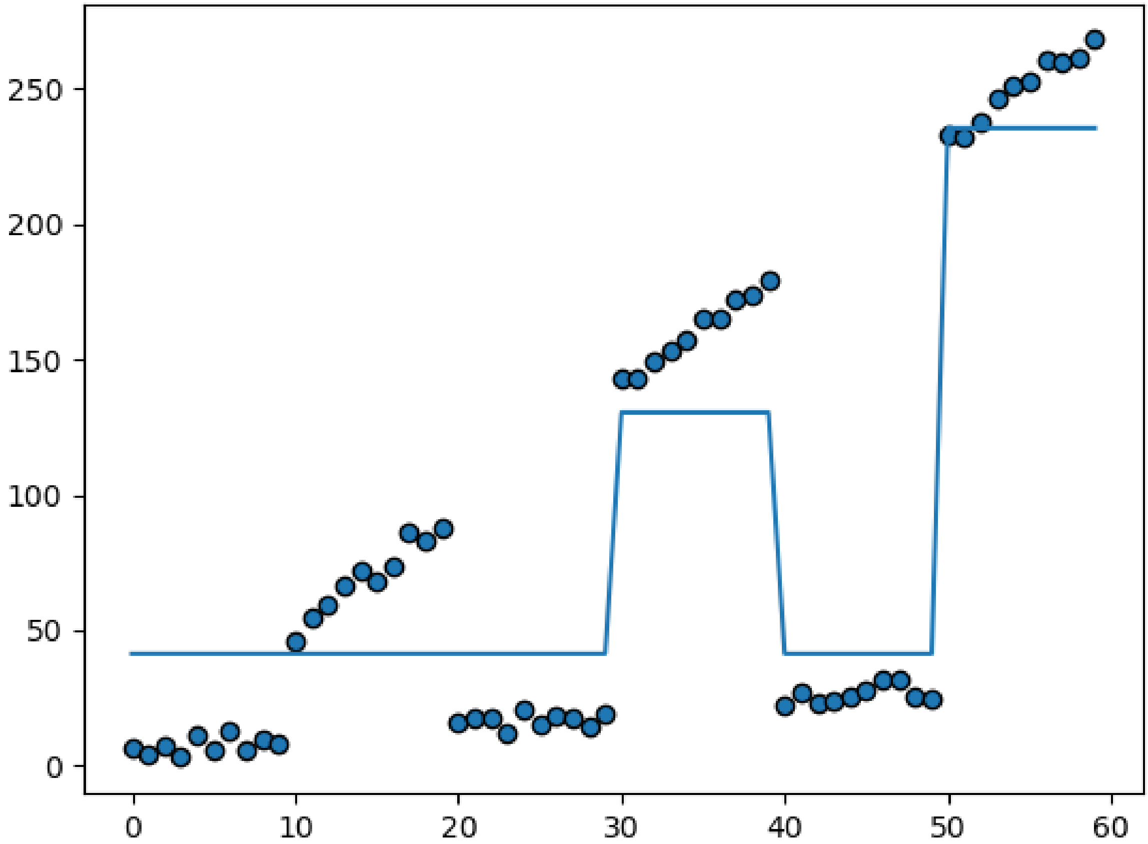

Fig. 5: results of param:5,1,1,'ls'

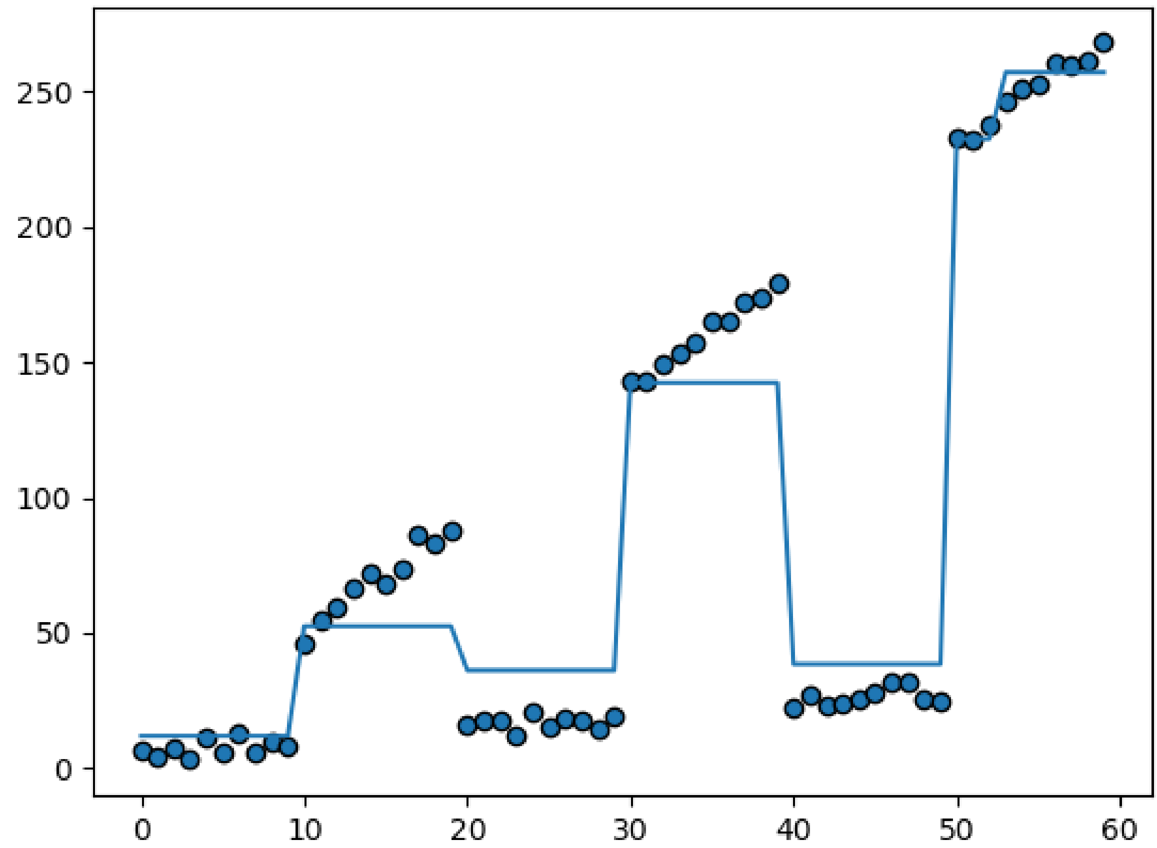

Fig. 6: results of param:10,1,1,'ls'

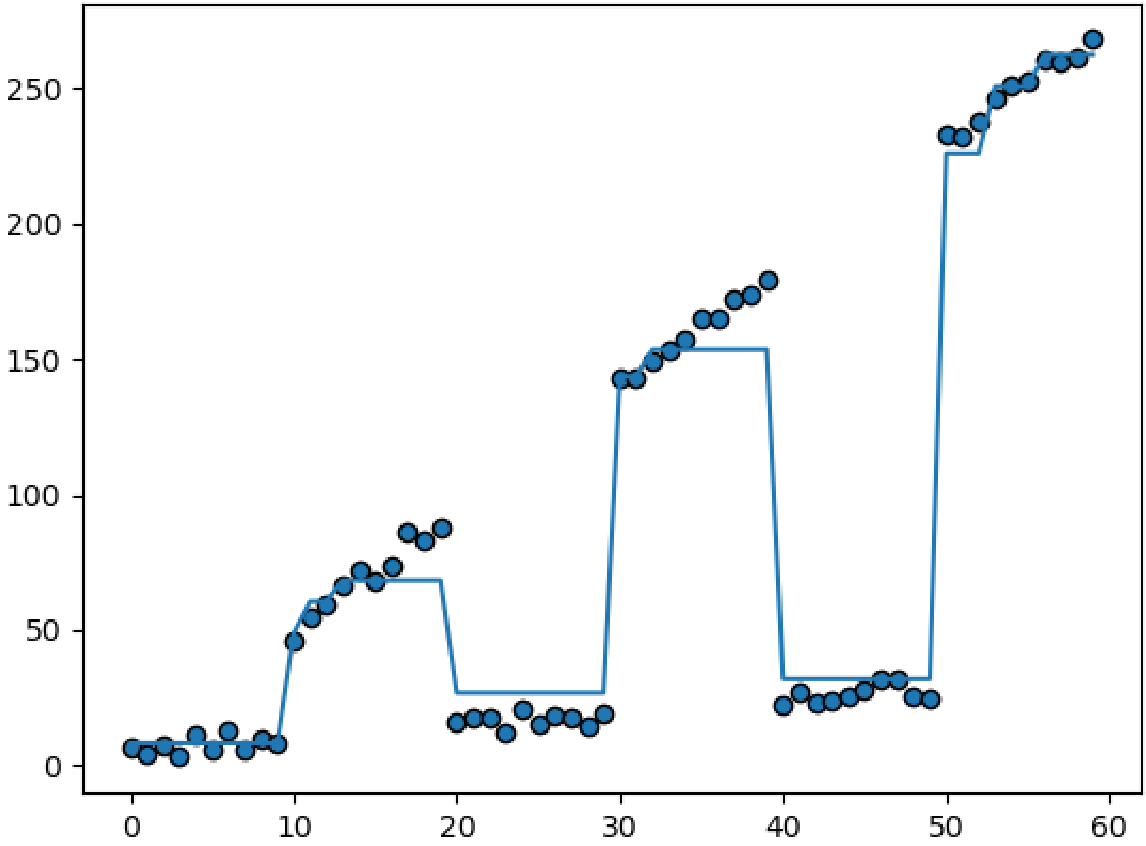

Fig. 7: results of param:20,1,1,'ls'

5. params = 'n_estimators': 50, 'max_depth': 1, 'learning_rate': 1, 'loss': 'ls'

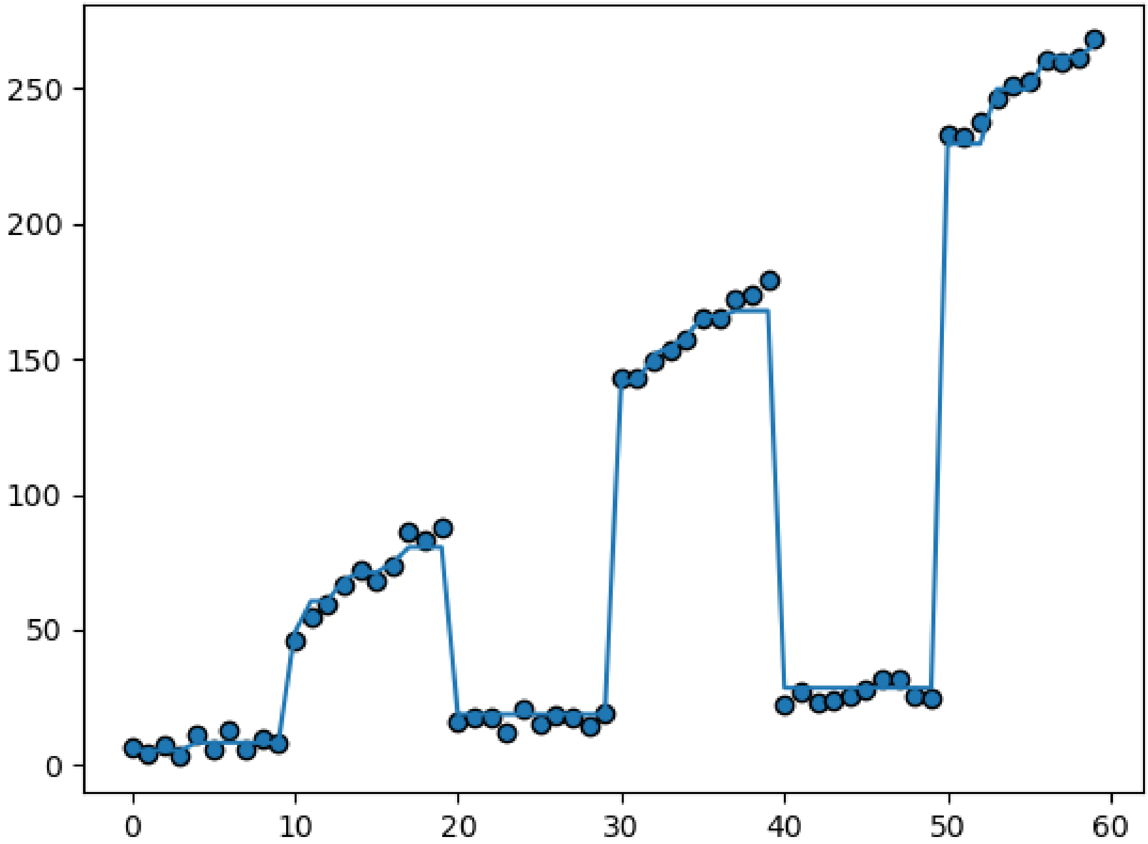

Fig. 8: results of param:50,1,1,'ls'

Just as Fig.8 shows, and R2 sq: 0.99817724226017594

Problem 2 (Quantity-Discount Model)

(a) What is the optimal ordering strategy?

From the question,

$$K = 50\$ \text{ /order} \quad h = 200/12\$ \text{ /(unit* month)} \quad \lambda = 50 \text{units/month} \quad c = \begin{cases} 520 & , x < 12 \\ 510 & , 12 \leq x \leq 64 \\ 495 & , 65 \leq x \leq 128 \\ 485 & , x > 128 \end{cases} \tag{1}$$

Therefore, we use the All-unit Discount: For this structure, we could generate the $g(Q)$, and get that:

$$\begin{aligned} g_0(Q) &= 520 * 50 + 50 * 50/Q + 200/24 * Q \\ g_1(Q) &= 510 * 50 + 50 * 50/Q + 200/24 * Q \\ g_2(Q) &= 495 * 50 + 50 * 50/Q + 200/24 * Q \\ g_3(Q) &= 485 * 50 + 50 * 50/Q + 200/24 * Q \end{aligned} \tag{2}$$

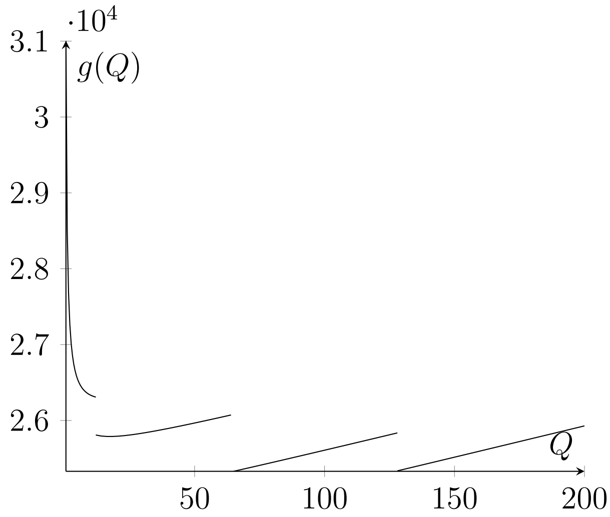

Fig. 9: Total cost for all-units quantity discount structure

And we could draw the graph like: and the Q_j^* is:

$$\begin{array}{rcl} Q_0^* & = & 17.32 \\ Q_1^* & = & 17.32 \\ Q_2^* & = & 17.32 \\ Q_3^* & = & 17.32 \end{array} \tag{3}$$

Among these value, Q_1^* is feasible, which is:

$$g_1(17) = 25788.7$$

Then we caculare the cost of breakpoints to the left of $Q_1^*$and get:

$$g_2(65) = 25330.13$$

Therefore, the optimal orfer quantity is $Q = 65$, which incurs a total monthly cost of 25330.13$

(b)The supplier has offered to be a drop shipper, i.e., they will ship directly to the customer. In exchange, they will increase the unit price to $520 per computer, but not charge the ordering costs and all inventory will be held at the supplier. From a purely financial standpoint, should Zeus take them up on the offer?

According to the question, we could get this results:

$$\lambda = 50\text{units/month} \quad c = 520\$/unit \tag{4}$$

Therefore, we can list the equation that

$$\text{Average Cycle cost} = 26000\$/month \tag{5}$$

Therefore, Zeus should not take them up on the offer.

Problem 3 (Wagner-Whitin Model)

(a) Use dynamic programming to solve the problem (on paper by hands).

First, we read from question and get the equation that:

$$K = 1000 \quad h = 1.2 \tag{6}$$

Then we apply the dynamic programming and get:

$$
\begin{aligned}
\theta_8 &= 0 \\
\theta_7 &= 1000 + 1.2(0 \cdot d_7) + \theta_8 \\
&= 1000[\text{s(7)=8}] \\
\theta_6 &= \min\{1000 + 1.2(0 \cdot d_6) + \theta_7, 1000 + 1.2(0 \cdot d_6 + 1 \cdot d_7) + \theta_8\} \\
&= \min\{2000, 1348\} \\
&= 1348[\text{s(6)=8}] \\
\theta_5 &= \min\{1000 + 1.2(0 \cdot d_5) + \theta_6, 1000 + 1.2(0 \cdot d_5 + 1 \cdot d_6) + \theta_7, \\
&\quad 1000 + 1.2(0 \cdot d_5 + 1 \cdot d_6 + 2 \cdot d_7) + \theta_8\} \\
&= \min\{2348, 2252, 1948\} \\
&= 1948[\text{s(5)=8}] \\
\theta_4 &= \min\{1000 + 1.2(0 \cdot d_4) + \theta_5, 1000 + 1.2(0 \cdot d_4 + 1 \cdot d_5) + \theta_6, \\
&\quad 1000 + 1.2(0 \cdot d_4 + 1 \cdot d_5 + 2 \cdot d_6) + \theta_7, \\
&\quad 1000 + 1.2(0 \cdot d_4 + 1 \cdot d_5 + 2 \cdot d_6 + 3 \cdot d_7) + \theta_8\} \\
&= \min\{2948, 2552, 2708, 2752\} \\
&= 2552[\text{s(4)=6}] \\
\theta_3 &= \min\{1000 + 1.2(0 \cdot d_3) + \theta_4, 1000 + 1.2(0 \cdot d_3 + 1 \cdot d_4) + \theta_5, \\
&\quad 1000 + 1.2(0 \cdot d_3 + 1 \cdot d_4 + 2 \cdot d_5) + \theta_6, \\
&\quad 1000 + 1.2(0 \cdot d_3 + 1 \cdot d_4 + 2 \cdot d_5 + 3 \cdot d_6) + \theta_7, \\
&\quad 1000 + 1.2(0 \cdot d_3 + 1 \cdot d_4 + 2 \cdot d_5 + 3 \cdot d_6 + 4 \cdot d_7) + \theta_8\} \\
&= \min\{3552, 3056, 2864, 3272, 3664\} \\
&= 2864[\text{s(3)=6}] \\
\theta_2 &= \min\{1000 + 1.2(0 \cdot d_2) + \theta_3, 1000 + 1.2(0 \cdot d_2 + 1 \cdot d_3) + \theta_4, \\
&\quad 1000 + 1.2(0 \cdot d_2 + 1 \cdot d_3 + 2 \cdot d_4) + \theta_5, \\
&\quad 1000 + 1.2(0 \cdot d_2 + 1 \cdot d_3 + 2 \cdot d_4 + 3 \cdot d_5) + \theta_6, \\
&\quad 1000 + 1.2(0 \cdot d_2 + 1 \cdot d_3 + 2 \cdot d_4 + 3 \cdot d_5 + 4 \cdot d_6) + \theta_7, \\
&\quad 1000 + 1.2(0 \cdot d_2 + 1 \cdot d_3 + 2 \cdot d_4 + 3 \cdot d_5 + 4 \cdot d_6 + 5 \cdot d_7) + \theta_8\} \\
&= \min\{3864, 3678, 3290, 3302, 3962, 4702\} \\
&= 3290[\text{s(2)=5}] \\
\theta_1 &= \min\{1000 + 1.2(0 \cdot d_1) + \theta_2, 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2) + \theta_3, \\
&\quad 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3) + \theta_4, \\
&\quad 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3 + 3 \cdot d_4) + \theta_5, \\
&\quad 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3 + 3 \cdot d_4 + 4 \cdot d_5) + \theta_6, \\
&\quad 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3 + 3 \cdot d_4 + 4 \cdot d_5 + 5 \cdot d_6) + \theta_7, \\
&\quad 1000 + 1.2(0 \cdot d_1 + 1 \cdot d_2 + 2 \cdot d_3 + 3 \cdot d_4 + 4 \cdot d_5 + 5 \cdot d_6 + 6 \cdot d_7) + \theta_8\} \\
&= \{4290, 4050, 3990, 3710, 3926, 4838, 5926\} \\
&= 3710\text{s(1)=5}
\end{aligned} \tag{7}
$$

Therefore, we could get that the best choice is:
Order 570 on Sunday, order 670 on Thursday

(b) Formulate as a shortest path problem and draw the corresponding diagram with nodes, edges, and edge costs. Solve using Dijkstra's algorithm (on paper by hands).

1 2 3 4 5 6 7 8

And we list the edge and edge cost as:

edge	edge cost
1-2	1000
1-3	1186
1-4	1438
1-5	1762
1-6	2578
1-7	3838
1-8	5926
2-3	1000
2-4	1126
2-5	1342
2-6	1954
2-7	2962
2-8	4702
3-4	1000
3-5	1108
3-6	1516
3-7	2272
3-8	3664
4-5	1000
4-6	1204
4-7	1708
4-8	2752
5-6	1000
5-7	1252
5-8	1948
6-7	1000
6-8	1348
7-8	1000

And by applying the Dikstra's algorithm, we could conclude that:

Number of steps	X	A[s]	B[s]
1	2,3,4,5,6,7,8	0	$\emptyset$
2	3,4,5,6,7,8	A[2]=1000	B[2]={1-2}
3	4,5,6,7,8	A[3]=1186	B[4]={1-3}
4	5,6,7,8	A[4]=1438	B[4]={1-4}
5	6,7,8	A[5]=1762	B[5]={1-5}
6	7,8	A[6]=2578	B[6]={1-6}
7	8	A[7]=3014	B[7]={1-5-7}
8	$\emptyset$	A[8]=3710	B[8]={1-5-8}

(c) Formulate the problem as a MILP on paper and solve it in Python.

we set variable as:
$d = [220, 155, 105, 90, 170, 210, 290]$
$T, K, h = 7, 1000, 1.2$ We want to minimize: $K * y[t] + h * x[t]$
and the constraints are:
$q[t] \leq M * y[t]$
$x[t] = x[t-1] + q[t] - d[t]$
$x[0] = q[0] - d[0]$ Therefore, we solve this in python and get the results that:

```
Thread count: 4 physical cores, 8 logical processors, using
Optimize a model with 14 rows, 21 columns and 34 nonzeros
Model fingerprint: 0x75f00a53
Variable types: 14 continuous, 7 integer (7 binary)
Coefficient statistics:
Matrix range     [1e+00, 1e+06]
Objective range  [1e+00, 1e+03]
Bounds range     [1e+00, 1e+00]
RHS range        [9e+01, 3e+02]
Presolve removed 3 rows and 4 columns
Presolve time: 0.00s
Presolved: 11 rows, 17 columns, 27 nonzeros
Variable types: 11 continuous, 6 integer (6 binary)

Root relaxation: objective 1.949020e+03, 7 iterations, 0.00

Nodes    |    Current Node    |     Objective Bounds     |     Work
Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

0     0 1949.02000    0    4          - 1949.02000     -     -    0s
H    0     0                    5948.0000000 1949.02000  67.2%     -    0s
H    0     0                    3926.0000000 1949.02000  50.4%     -    0s
0     0 3204.90092    0    3 3926.00000 3204.90092  18.4%     -    0s
H    0     0                    3710.0000000 3204.90092  13.6%     -    0s

Cutting planes:
Implied bound: 7
MIR: 2
Flow cover: 4

Explored 1 nodes (15 simplex iterations) in 0.00 seconds
```

```
Thread count was 8 (of 8 available processors)

Solution count 3: 3710 3926 5948

Optimal solution found (tolerance 1.00e-04)
Best objective 3.709999999983e+03, best bound 3.70999999998

Variable            X
-------------------------
order_quantity[0]          570
order_quantity[4]          670
inventory_level[0]          350
inventory_level[1]          195
inventory_level[2]           90
inventory_level[4]          500
inventory_level[5]          290
if_order[0]             1
if_order[4]             1
```

Problem 4 (Linear Programming Duality)

(a) Please write down duality of the following linear programming problem.

$$
\begin{aligned}
min \quad & 24y_1 + 60y_2 \\
s.t. \quad & 3y_1 + y_2 \geq 6 \\
& 2y_1 + 2y_2 \leq 14 \\
& y_1 + 4y_2 = 13 \\
& y_1 \geq 0 \\
& y_2 \leq 0
\end{aligned}
\tag{8}
$$

(b) Bipartite graph is a special graph. Its vertices are divided into two separate sets and edges only exist between those two sets.

suppose we have n nodes in I and m nodes in J,

$$
\begin{aligned}
min \quad & \sum_{i:(i,j)\in E} y_i + \sum_{j:(i,j)\in E} y_{n+j} \\
s.t. \quad & y_i + y_j \geq 1 \text{ for (i,j)} \in E \\
& y_k \geq 0 \text{ for } k \in A
\end{aligned}
\tag{9}
$$