2024 PRODUCTION PLANNING	Date	Presenter
	2024. 05. 31.	금강산

생산계획 팀프로젝트

중간 발표

고려대학교 산업경영공학부

2020170818 금강산

2020170837 최원준

2021170866 이예일

고려대학교
KOREA UNIVERSITY

TABLE OF CONTENTS

01. 문제 정의

02. 프로그래밍 특성

03. 결과 분석

04. 프로젝트 진행 소감

PROBLEM STATEMENT

문제 정의

기업 선정

- 혁신적인 우주 탐사 선두 주자 SPACEX
- 로켓이라는 상품의 특성상 MTO 생산 전략 사용
- MPS : 로켓 자재 및 부품의 생산 계획 수립
- MRP: MPS의 세부적 관리

생산 품목

- 페어링
- 중간부
- 엔진
- 완성체인 로켓을 만드는 것이 아닌, Space X의 기술력으로 외부에 판매하는 상황으로 가정하고 각각의 품목으로 접근

문제 정의

SOP(Chase Demand)

Space X SOP Chase Demand		History		Month											
		11	12	1	2	3	4	5	6	7	8	9	10	11	12
Sale	Forecast	50	45	45	45	45	50	50	50	60	60	60	70	70	60
	Actual	30	20												
Operation	Plan	50	45	45	45	45	50	50	50	60	60	60	70	70	60
	Employees	1136	1023	1125	1125	1023	1190	1136	1250	1364	1364	1579	1750	1591	1500
	Working Days	22	22	20	20	22	21	22	20	22	22	19	20	22	20
	Actual	50	55												
Inventory	Plan	30	30	35	35	35	35	35	35	35	35	35	35	35	35
	Actual	25	35												
	Days of supply	27.5	15.5	15.1	17.1	15.5	17.1	12.3	12.3	12.8	11.1	10	11	10	10

Employee productivity : 2units/day/worker

문제 정의

MPS(페어링)

Space X MPS Item : 페어링	On Hand	Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Forecast		10	10	10	15	15	15	15	20	20	20	15	15
Orders		5	2	4									
Projected available balance	12												
Available – to - promise													
MPS													

Lot size = 50, Safety Stock = 0

문제 정의

MPS(중간부)

Space X MPS Item : 중간부	On Hand	Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Forecast		20	20	20	20	25	20	15	20	20	20	15	15
Orders		15	9	12									
Projected available balance	15												
Available – to - promise													
MPS													

Lot size = 50, Safety Stock = 0

문제 정의

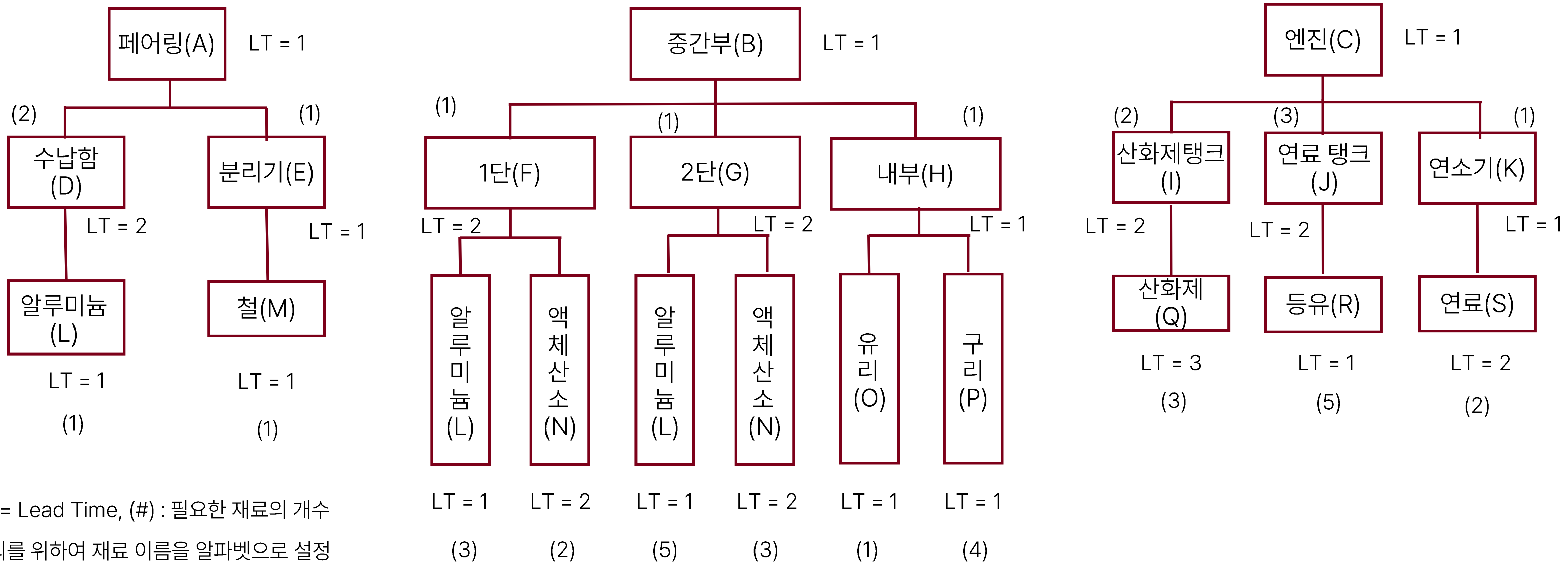
MPS(엔진)

Space X MPS Item : 엔진	On Hand	Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Forecast		25	25	25	25	30	30	30	20	20	20	15	15
Orders		15	8	7	6	4							
Projected available balance	20												
Available – to - promise													
MPS													

Lot size = 50, Safety Stock = 0

문제 정의

BOM



페어링(A) LT = 1
(2) 수납함(D) LT = 2
알루미늄(L) LT = 1 (1)
(1) 분리기(E) LT = 1
철(M) LT = 1 (1)

중간부(B) LT = 1
(1) 1단(F) LT = 2
알루미늄(L) LT = 1 (3)
액체산소(N) LT = 2 (2)
(1) 2단(G) LT = 2
알루미늄(L) LT = 1 (5)
액체산소(N) LT = 2 (3)
(1) 내부(H) LT = 1
유리(O) LT = 1 (1)
구리(P) LT = 1 (4)

엔진(C) LT = 1
(2) 산화제탱크(I) LT = 2
산화제(Q) LT = 3 (3)
(3) 연료 탱크(J) LT = 2
등유(R) LT = 1 (5)
(1) 연소기(K) LT = 1
연료(S) LT = 2 (2)

LT = Lead Time, (#) : 필요한 재료의 개수

편의를 위하여 재료 이름을 알파벳으로 설정

- Lot for Lot Strategy → Lot Size의 경우 부품별 정의
- Safety Stock의 경우 0으로 설정

PROBLEM STATEMENT

문제 정의

MRP

Part : A		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	20										
Projected available balance	10												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : B		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		45	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : C		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : D		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		0	20										
Projected available balance	30												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

PROBLEM STATEMENT

문제 정의

MRP

Part : E		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	15										
Projected available balance	50												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : F		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		70	60										
Projected available balance	30												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

Part : G		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	30										
Projected available balance	50												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

Part : H		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		40	30										
Projected available balance	40												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

PROBLEM STATEMENT

문제 정의

MRP

Part : I		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

Part : J		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		25	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

Part : K		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		10	10										
Projected available balance	20												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : L		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		10	10										
Projected available balance	20												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

PROBLEM STATEMENT

문제 정의

MRP

Part : M		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		80	60	0	0	0	60						
Projected available balance	50												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : N		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

Part : O		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		40	20										
Projected available balance	30												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : P		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		0	20										
Projected available balance	30												
Planned order release													

Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

문제 정의

MRP

Part : Q		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	40										
Projected available balance	10												
Planned order release													

Q = Lot-for-Lot, LT = 3, Safety Stock(SS) = 0

Part : R		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	15										
Projected available balance	30												
Planned order release													

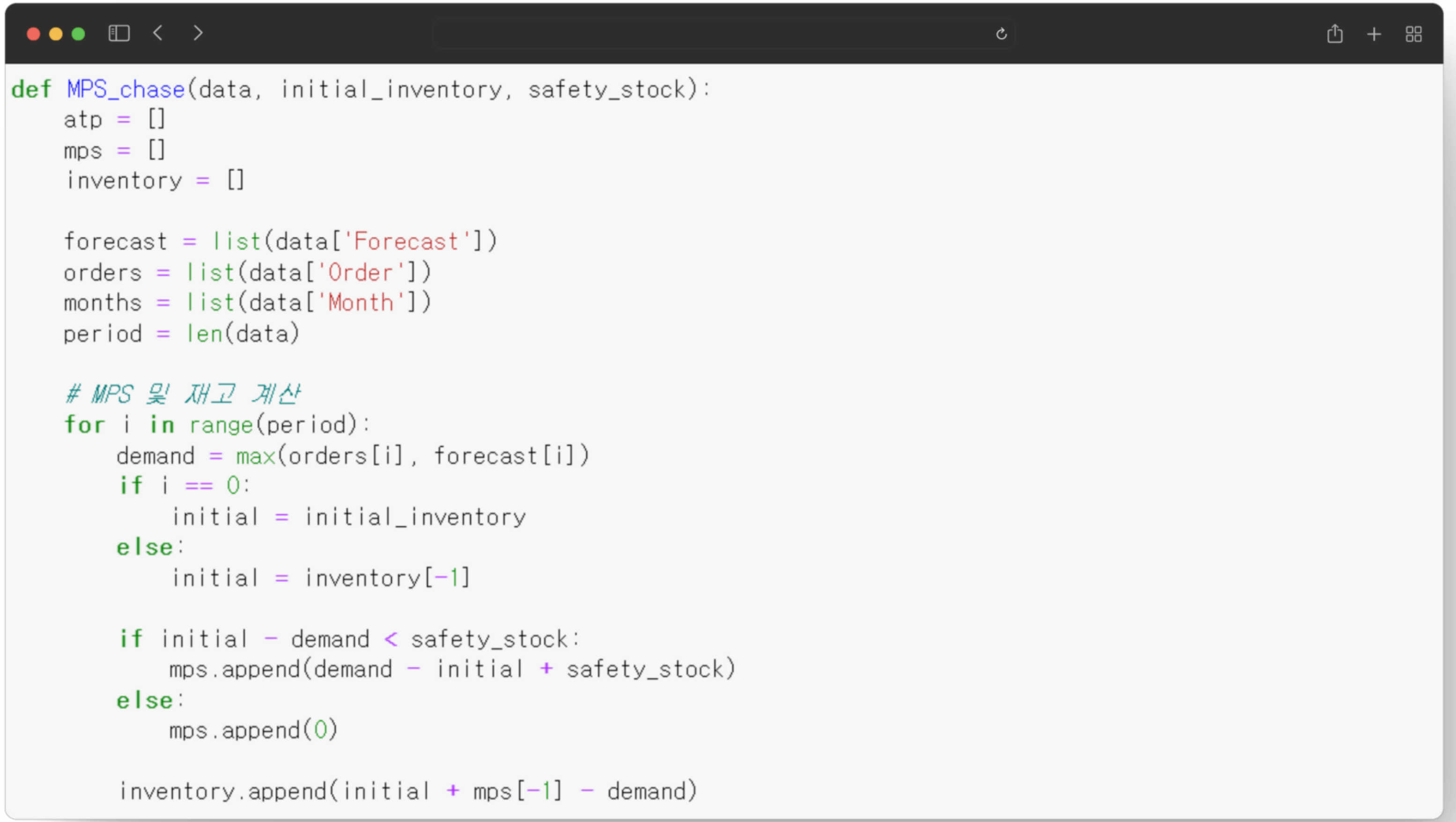
Q = Lot-for-Lot, LT = 1, Safety Stock(SS) = 0

Part : S		Month											
		1	2	3	4	5	6	7	8	9	10	11	12
Gross Requirement													
Scheduled receipts		50	10										
Projected available balance	15												
Planned order release													

Q = Lot-for-Lot, LT = 2, Safety Stock(SS) = 0

프로그래밍

MPS

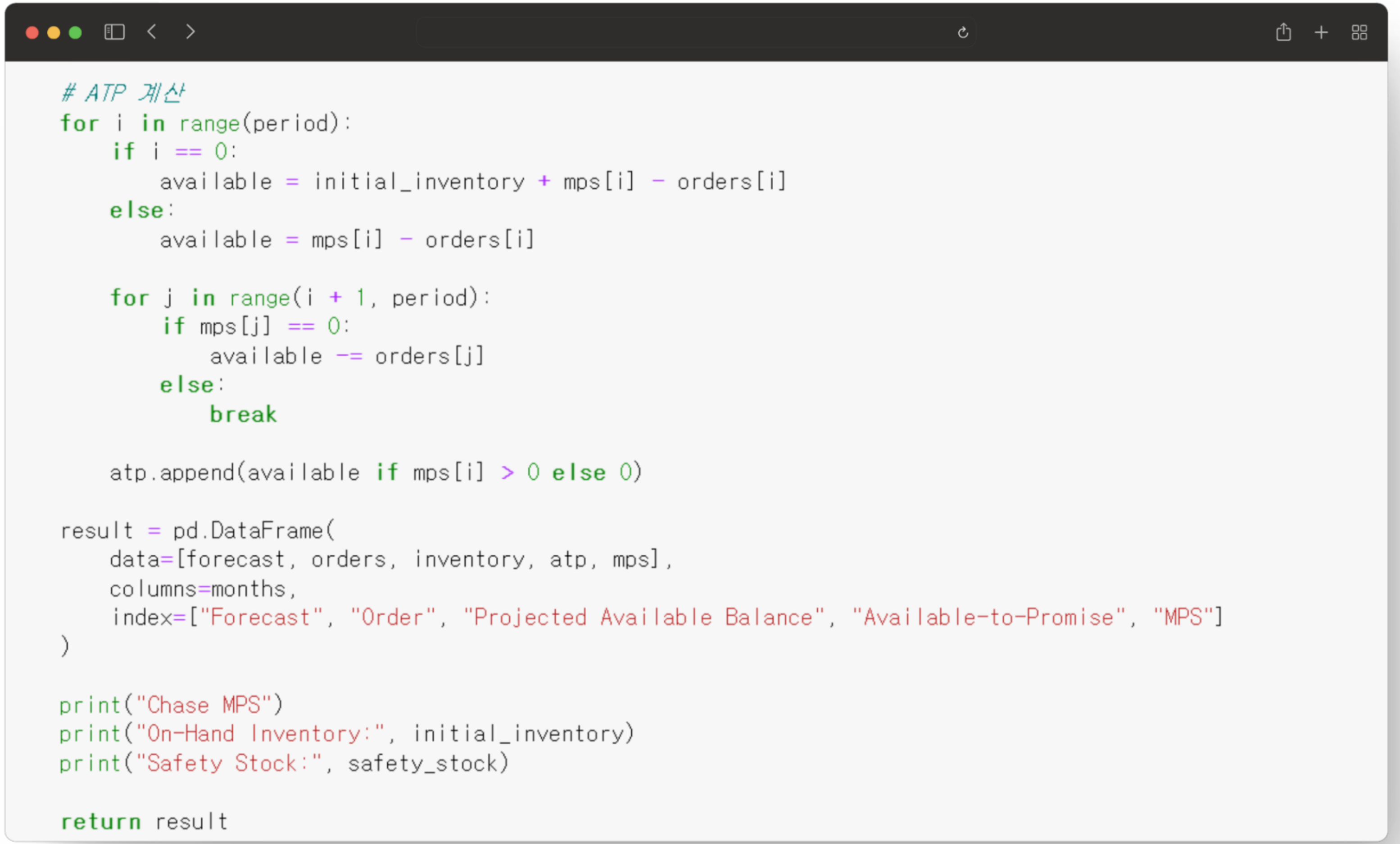


```python
def MPS_chase(data, initial_inventory, safety_stock):
    atp = []
    mps = []
    inventory = []

    forecast = list(data['Forecast'])
    orders = list(data['Order'])
    months = list(data['Month'])
    period = len(data)

    # MPS 및 재고 계산
    for i in range(period):
        demand = max(orders[i], forecast[i])
        if i == 0:
            initial = initial_inventory
        else:
            initial = inventory[-1]

        if initial - demand < safety_stock:
            mps.append(demand - initial + safety_stock)
        else:
            mps.append(0)

        inventory.append(initial + mps[-1] - demand)
```

12개월 동안의 Forecast와 Order Data를 csv 파일을 이용하여 입력하고 초기 재고, 안전 재고 값을 지정하면 Chasing 방식을 통해 MPS를 계산하는 코드

프로그래밍

MPS

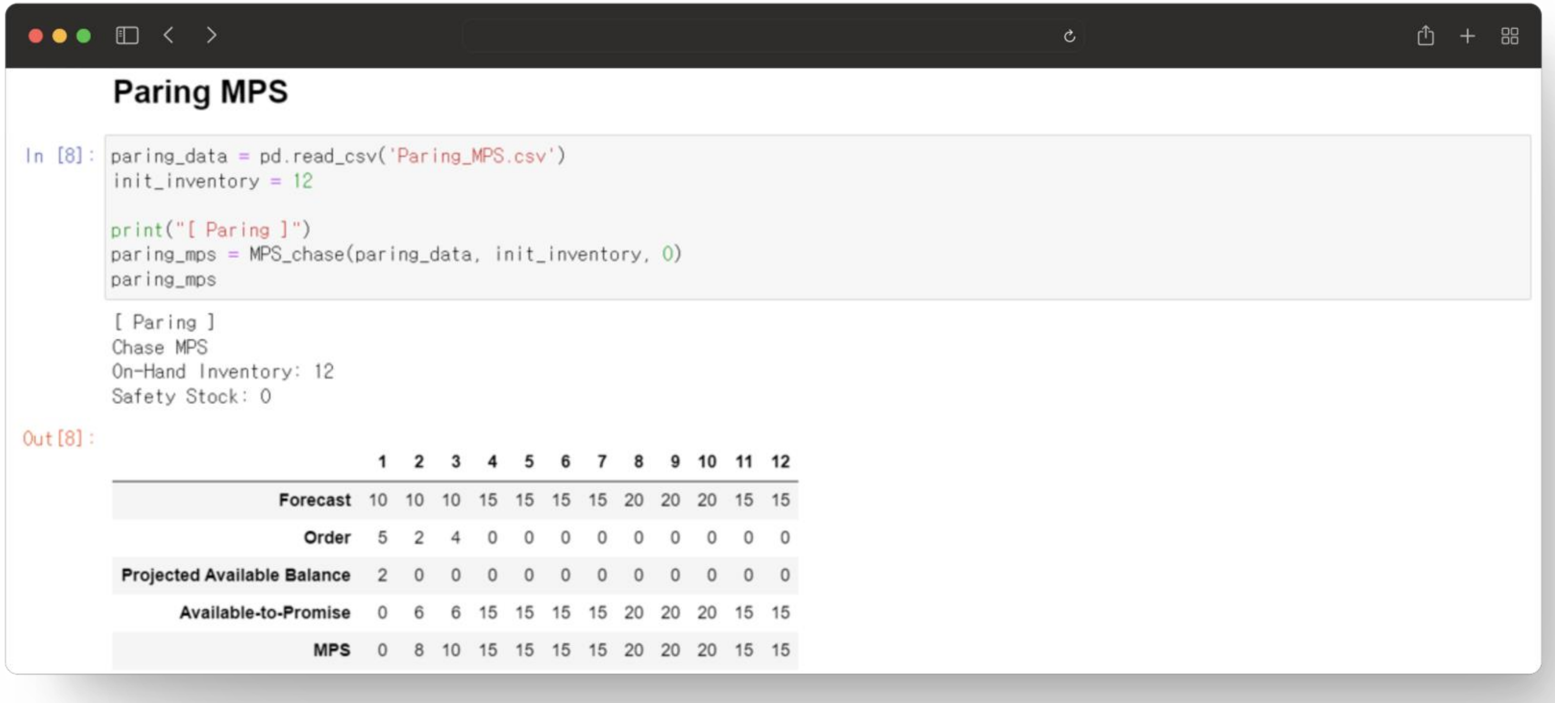


```
# ATP 계산
for i in range(period):
    if i == 0:
        available = initial_inventory + mps[i] - orders[i]
    else:
        available = mps[i] - orders[i]

    for j in range(i + 1, period):
        if mps[j] == 0:
            available -= orders[j]
        else:
            break

    atp.append(available if mps[i] > 0 else 0)

result = pd.DataFrame(
    data=[forecast, orders, inventory, atp, mps],
    columns=months,
    index=["Forecast", "Order", "Projected Available Balance", "Available-to-Promise", "MPS"]
)

print("Chase MPS")
print("On-Hand Inventory:", initial_inventory)
print("Safety Stock:", safety_stock)

return result
```

ATP(Available – to – promise) 값을 계산하는 코드로, Forecast, Order, PAB(Projected Available Balance), ATP, MPS 값을 계산한 뒤 Dataframe 형식으로 반환

프로그래밍

MPS



Paring MPS

In [8]:

```
paring_data = pd.read_csv('Paring_MPS.csv')
init_inventory = 12

print("[ Paring ]")
paring_mps = MPS_chase(paring_data, init_inventory, 0)
paring_mps
```

```
[ Paring ]
Chase MPS
On-Hand Inventory: 12
Safety Stock: 0
```

Out[8]:

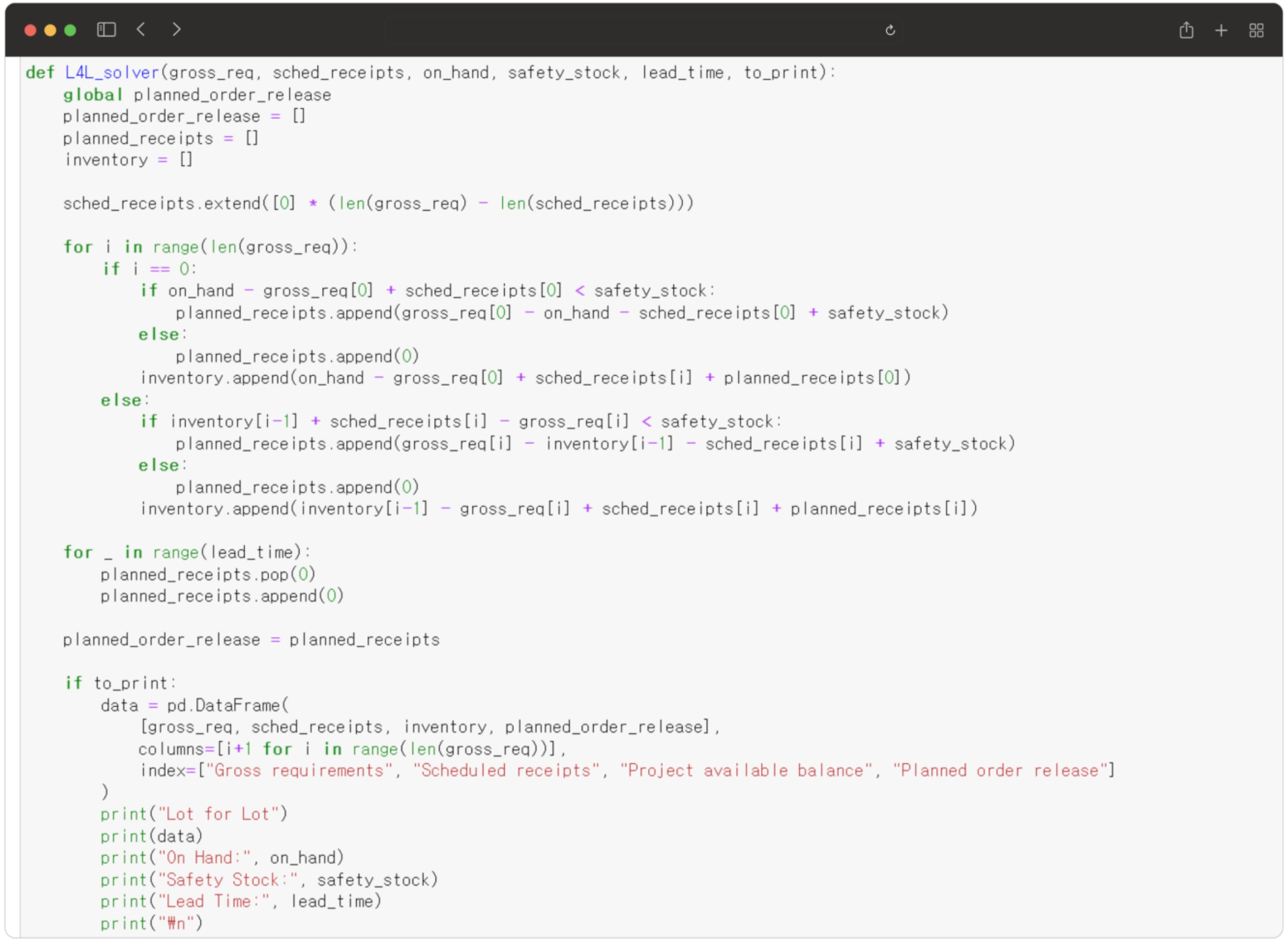
	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	10	10	10	15	15	15	15	20	20	20	15	15
Order	5	2	4	0	0	0	0	0	0	0	0	0
Projected Available Balance	2	0	0	0	0	0	0	0	0	0	0	0
Available-to-Promise	0	6	6	15	15	15	15	20	20	20	15	15
MPS	0	8	10	15	15	15	15	20	20	20	15	15

Csv 파일에 포함된 12개월 간의 Forecast, Order 정보를 가져오고, 입력한 초기 재고에 맞게 MPS를 전개하는 코드

페어링, 중간부, 엔진 세가지 품목에 대하여 모두 동일한 방식으로 적용

프로그래밍

Lot – for - Lot

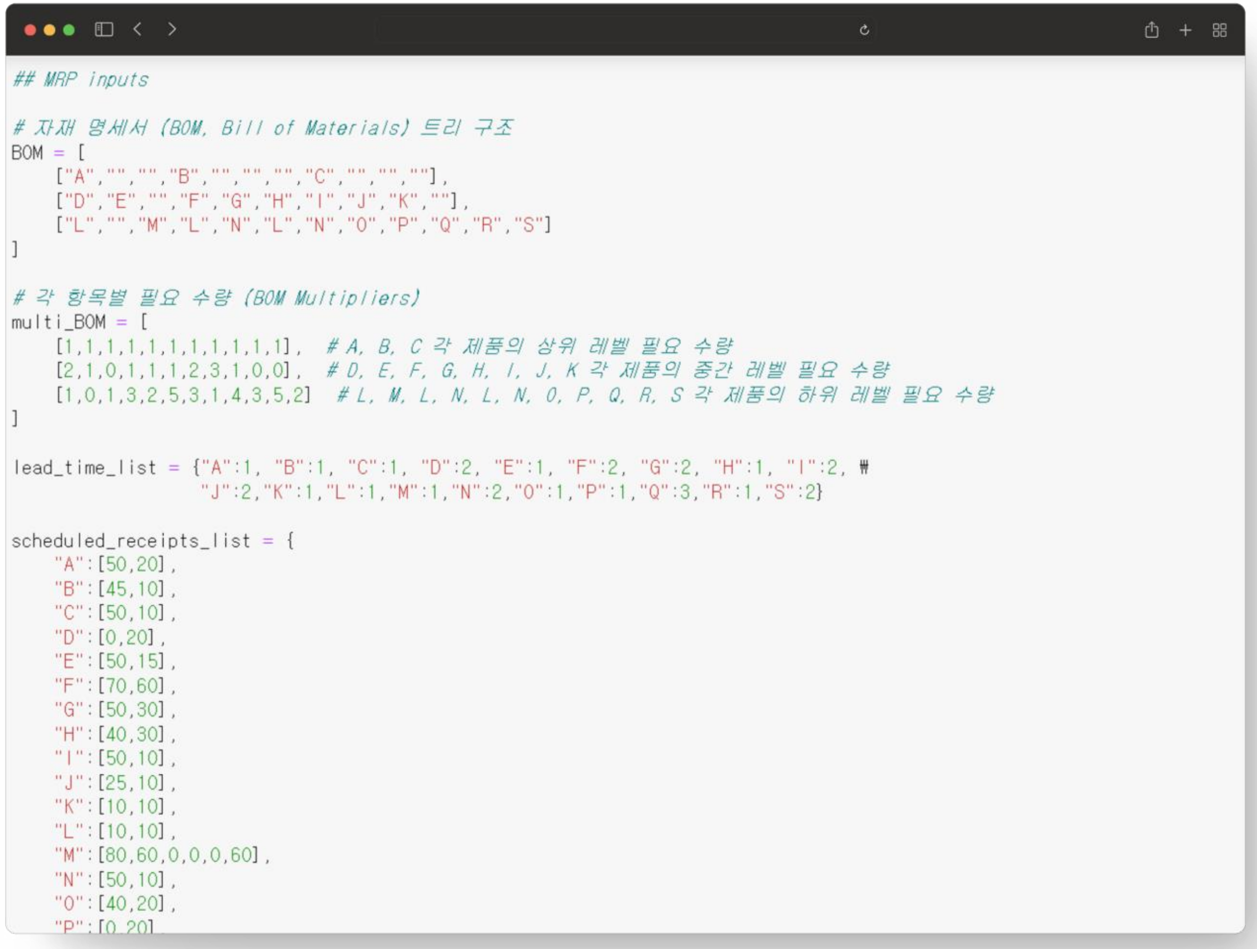


```
def L4L_solver(gross_req, sched_receipts, on_hand, safety_stock, lead_time, to_print):
    global planned_order_release
    planned_order_release = []
    planned_receipts = []
    inventory = []

    sched_receipts.extend([0] * (len(gross_req) - len(sched_receipts)))

    for i in range(len(gross_req)):
        if i == 0:
            if on_hand - gross_req[0] + sched_receipts[0] < safety_stock:
                planned_receipts.append(gross_req[0] - on_hand - sched_receipts[0] + safety_stock)
            else:
                planned_receipts.append(0)
            inventory.append(on_hand - gross_req[0] + sched_receipts[i] + planned_receipts[0])
        else:
            if inventory[i-1] + sched_receipts[i] - gross_req[i] < safety_stock:
                planned_receipts.append(gross_req[i] - inventory[i-1] - sched_receipts[i] + safety_stock)
            else:
                planned_receipts.append(0)
            inventory.append(inventory[i-1] - gross_req[i] + sched_receipts[i] + planned_receipts[i])

    for _ in range(lead_time):
        planned_receipts.pop(0)
        planned_receipts.append(0)

    planned_order_release = planned_receipts

    if to_print:
        data = pd.DataFrame(
            [gross_req, sched_receipts, inventory, planned_order_release],
            columns=[i+1 for i in range(len(gross_req))],
            index=["Gross requirements", "Scheduled receipts", "Project available balance", "Planned order release"]
        )
        print("Lot for Lot")
        print(data)
        print("On Hand:", on_hand)
        print("Safety Stock:", safety_stock)
        print("Lead Time:", lead_time)
        print("₩n")
```

Lot – for – Lot 생산 전략에 대한 함수

프로그래밍

MRP

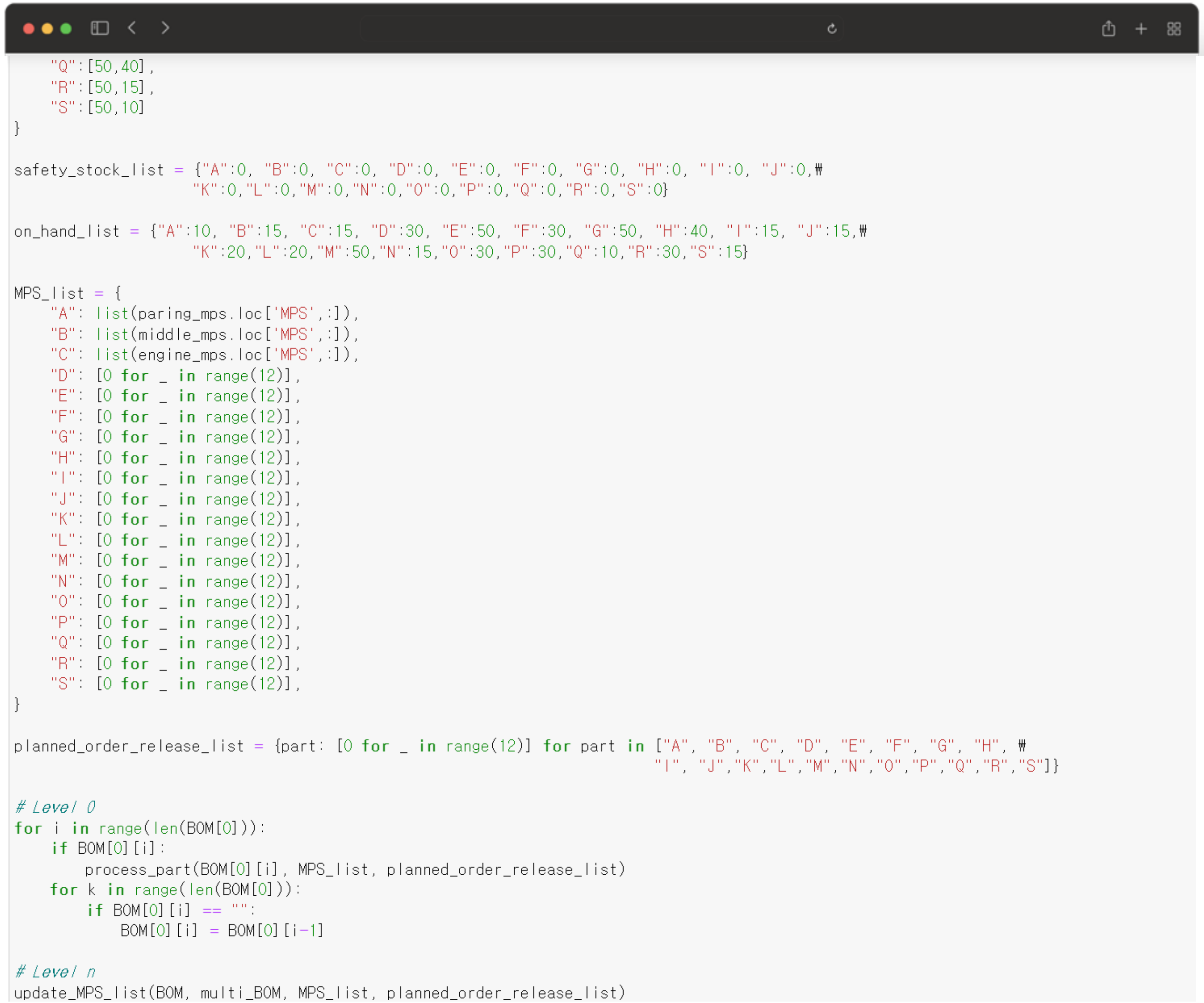


```
## MRP inputs

# 자재 명세서 (BOM, Bill of Materials) 트리 구조
BOM = [
    ["A","","","B","","","","C","","",""],
    ["D","E","","F","G","H","I","J","K",""],
    ["L","","M","L","N","L","N","O","P","Q","R","S"]
]

# 각 항목별 필요 수량 (BOM Multipliers)
multi_BOM = [
    [1,1,1,1,1,1,1,1,1,1,1],  # A, B, C 각 제품의 상위 레벨 필요 수량
    [2,1,0,1,1,1,2,3,1,0,0],  # D, E, F, G, H, I, J, K 각 제품의 중간 레벨 필요 수량
    [1,0,1,3,2,5,3,1,4,3,5,2]  # L, M, L, N, L, N, O, P, Q, R, S 각 제품의 하위 레벨 필요 수량
]

lead_time_list = {"A":1, "B":1, "C":1, "D":2, "E":1, "F":2, "G":2, "H":1, "I":2, ₩
                  "J":2,"K":1,"L":1,"M":1,"N":2,"O":1,"P":1,"Q":3,"R":1,"S":2}

scheduled_receipts_list = {
    "A":[50,20],
    "B":[45,10],
    "C":[50,10],
    "D":[0,20],
    "E":[50,15],
    "F":[70,60],
    "G":[50,30],
    "H":[40,30],
    "I":[50,10],
    "J":[25,10],
    "K":[10,10],
    "L":[10,10],
    "M":[80,60,0,0,0,60],
    "N":[50,10],
    "O":[40,20],
    "P":[0,20]
```

BOM, 부품별 Lead Time과 Safety Stock, 현재 재고와 MPS 값을 입력하면 부품 별 MRP를 전개해서 출력하는 코드

프로그래밍

MRP

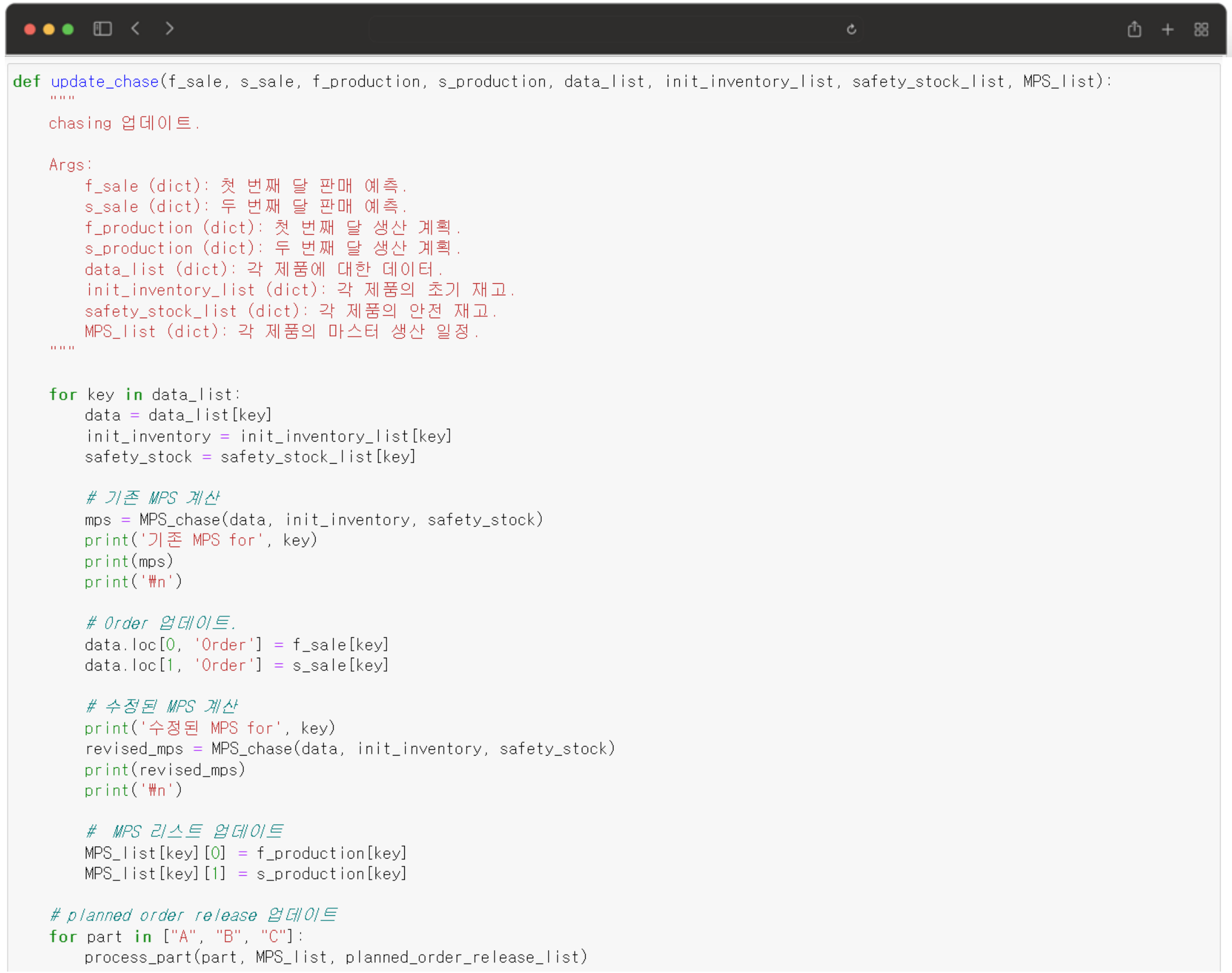


```
    "Q":[50,40],
    "R":[50,15],
    "S":[50,10]
}

safety_stock_list = {"A":0, "B":0, "C":0, "D":0, "E":0, "F":0, "G":0, "H":0, "I":0, "J":0,₩
                     "K":0,"L":0,"M":0,"N":0,"O":0,"P":0,"Q":0,"R":0,"S":0}

on_hand_list = {"A":10, "B":15, "C":15, "D":30, "E":50, "F":30, "G":50, "H":40, "I":15, "J":15,₩
                     "K":20,"L":20,"M":50,"N":15,"O":30,"P":30,"Q":10,"R":30,"S":15}

MPS_list = {
    "A": list(paring_mps.loc['MPS',:]),
    "B": list(middle_mps.loc['MPS',:]),
    "C": list(engine_mps.loc['MPS',:]),
    "D": [0 for _ in range(12)],
    "E": [0 for _ in range(12)],
    "F": [0 for _ in range(12)],
    "G": [0 for _ in range(12)],
    "H": [0 for _ in range(12)],
    "I": [0 for _ in range(12)],
    "J": [0 for _ in range(12)],
    "K": [0 for _ in range(12)],
    "L": [0 for _ in range(12)],
    "M": [0 for _ in range(12)],
    "N": [0 for _ in range(12)],
    "O": [0 for _ in range(12)],
    "P": [0 for _ in range(12)],
    "Q": [0 for _ in range(12)],
    "R": [0 for _ in range(12)],
    "S": [0 for _ in range(12)],
}

planned_order_release_list = {part: [0 for _ in range(12)] for part in ["A", "B", "C", "D", "E", "F", "G", "H", ₩
                                                                        "I", "J","K","L","M","N","O","P","Q","R","S"]}

# Level 0
for i in range(len(BOM[0])):
    if BOM[0][i]:
        process_part(BOM[0][i], MPS_list, planned_order_release_list)
    for k in range(len(BOM[0])):
        if BOM[0][i] == "":
            BOM[0][i] = BOM[0][i-1]

# Level n
update_MPS_list(BOM, multi_BOM, MPS_list, planned_order_release_list)
```

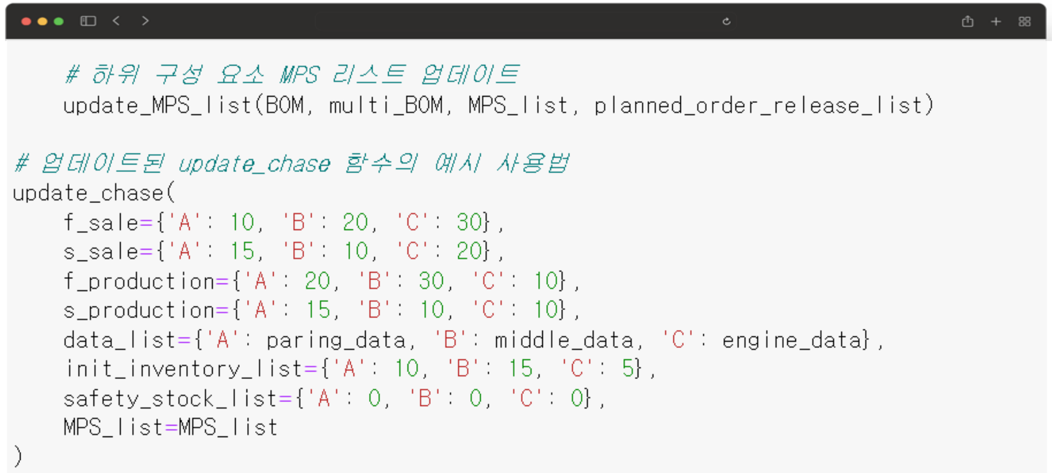
BOM, 부품별 Lead Time과 Safety Stock, 현재 재고와 MPS 값을 입력하면 부품 별 MRP를 전개해서 출력하는 코드

프로그래밍

Update Program



```python
def update_chase(f_sale, s_sale, f_production, s_production, data_list, init_inventory_list, safety_stock_list, MPS_list):
    """
    chasing 업데이트.

    Args:
        f_sale (dict): 첫 번째 달 판매 예측.
        s_sale (dict): 두 번째 달 판매 예측.
        f_production (dict): 첫 번째 달 생산 계획.
        s_production (dict): 두 번째 달 생산 계획.
        data_list (dict): 각 제품에 대한 데이터.
        init_inventory_list (dict): 각 제품의 초기 재고.
        safety_stock_list (dict): 각 제품의 안전 재고.
        MPS_list (dict): 각 제품의 마스터 생산 일정.
    """

    for key in data_list:
        data = data_list[key]
        init_inventory = init_inventory_list[key]
        safety_stock = safety_stock_list[key]

        # 기존 MPS 계산
        mps = MPS_chase(data, init_inventory, safety_stock)
        print('기존 MPS for', key)
        print(mps)
        print('₩n')

        # Order 업데이트.
        data.loc[0, 'Order'] = f_sale[key]
        data.loc[1, 'Order'] = s_sale[key]

        # 수정된 MPS 계산
        print('수정된 MPS for', key)
        revised_mps = MPS_chase(data, init_inventory, safety_stock)
        print(revised_mps)
        print('₩n')

        #  MPS 리스트 업데이트
        MPS_list[key][0] = f_production[key]
        MPS_list[key][1] = s_production[key]

    # planned order release 업데이트
    for part in ["A", "B", "C"]:
        process_part(part, MPS_list, planned_order_release_list)
```



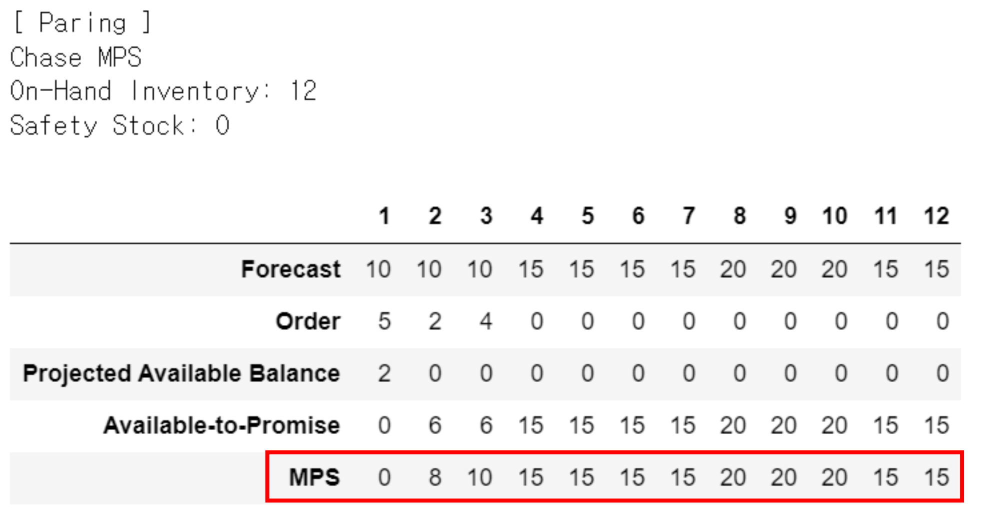
```python
    # 하위 구성 요소 MPS 리스트 업데이트
    update_MPS_list(BOM, multi_BOM, MPS_list, planned_order_release_list)

# 업데이트된 update_chase 함수의 예시 사용법
update_chase(
    f_sale={'A': 10, 'B': 20, 'C': 30},
    s_sale={'A': 15, 'B': 10, 'C': 20},
    f_production={'A': 20, 'B': 30, 'C': 10},
    s_production={'A': 15, 'B': 10, 'C': 10},
    data_list={'A': paring_data, 'B': middle_data, 'C': engine_data},
    init_inventory_list={'A': 10, 'B': 15, 'C': 5},
    safety_stock_list={'A': 0, 'B': 0, 'C': 0},
    MPS_list=MPS_list
)
```

1st, 2nd month 판매, 생산 실적 주어졌을 때 MPS와 MRP 계획 Update 하는 코드

제품 [페어링], [중간부], [엔진]의 1st, 2nd month 판매, 생산 실적 및 기존 Forecast, Order가 포함된 csv 파일, 초기 재고와 Safety Stock, MPS를 입력하여 수정된 MPS와 MRP 전개

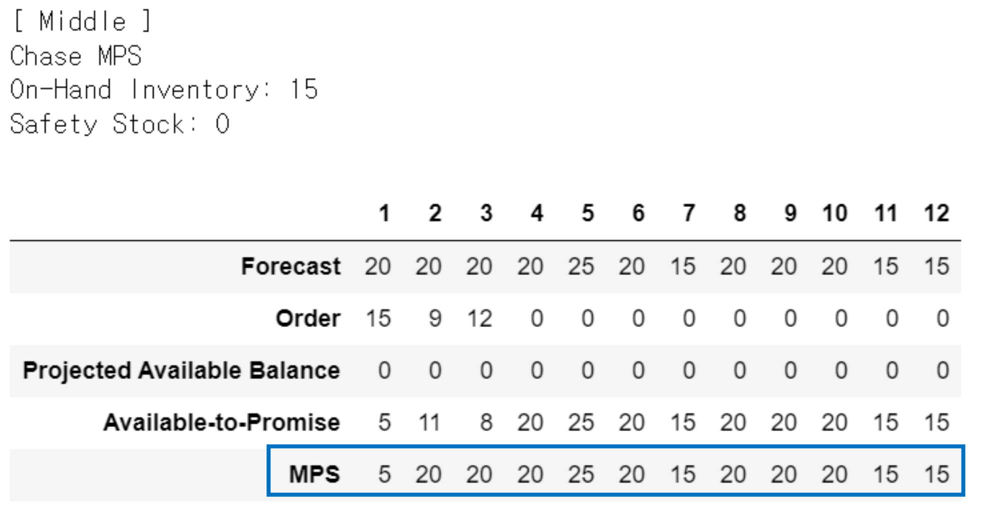
R E S U L T   A N A L Y S I S

결과 분석



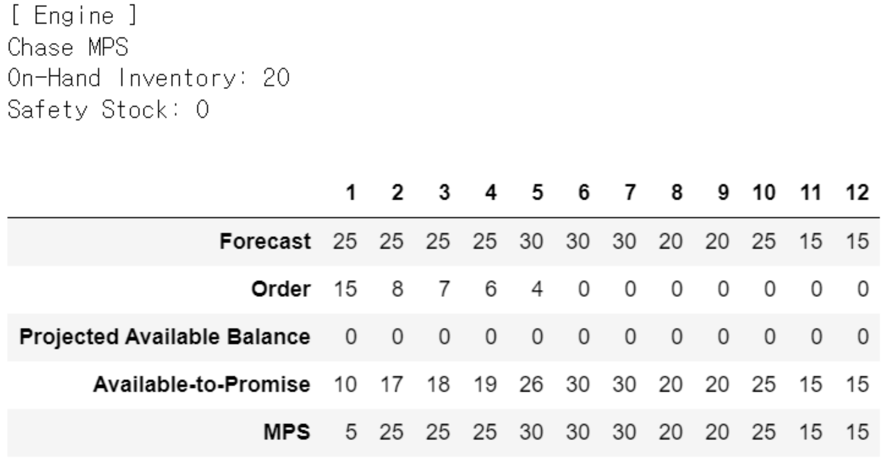
```
[ Paring ]
Chase MPS
On-Hand Inventory: 12
Safety Stock: 0
```

	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	10	10	10	15	15	15	15	20	20	20	15	15
Order	5	2	4	0	0	0	0	0	0	0	0	0
Projected Available Balance	2	0	0	0	0	0	0	0	0	0	0	0
Available-to-Promise	0	6	6	15	15	15	15	20	20	20	15	15
MPS	0	8	10	15	15	15	15	20	20	20	15	15



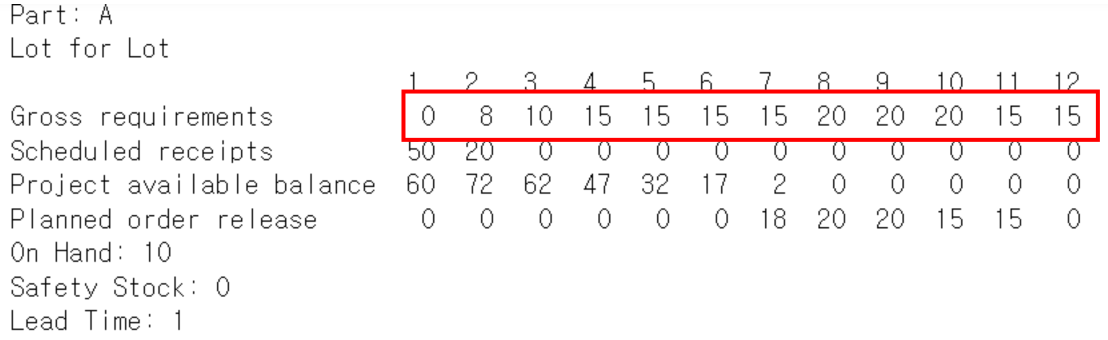
```
[ Middle ]
Chase MPS
On-Hand Inventory: 15
Safety Stock: 0
```

	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	20	20	20	20	25	20	15	20	20	20	15	15
Order	15	9	12	0	0	0	0	0	0	0	0	0
Projected Available Balance	0	0	0	0	0	0	0	0	0	0	0	0
Available-to-Promise	5	11	8	20	25	20	15	20	20	20	15	15
MPS	5	20	20	20	25	20	15	20	20	20	15	15

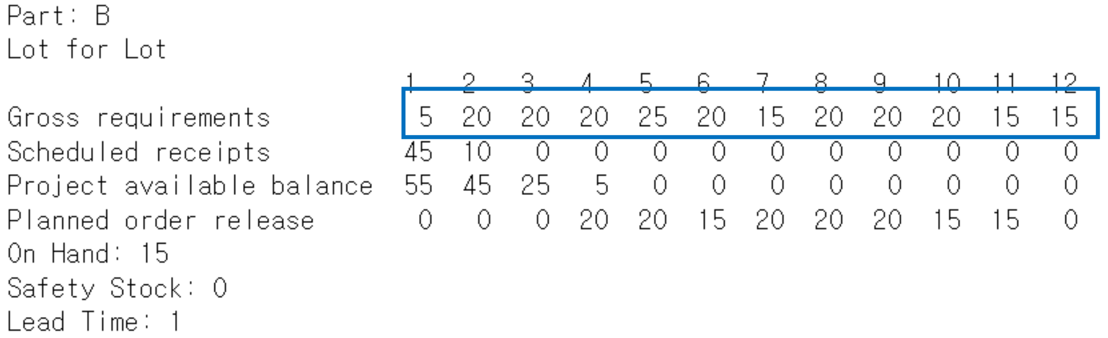


```
[ Engine ]
Chase MPS
On-Hand Inventory: 20
Safety Stock: 0
```

	1	2	3	4	5	6	7	8	9	10	11	12
Forecast	25	25	25	25	30	30	30	20	20	25	15	15
Order	15	8	7	6	4	0	0	0	0	0	0	0
Projected Available Balance	0	0	0	0	0	0	0	0	0	0	0	0
Available-to-Promise	10	17	18	19	26	30	30	20	20	25	15	15
MPS	5	25	25	25	30	30	30	20	20	25	15	15

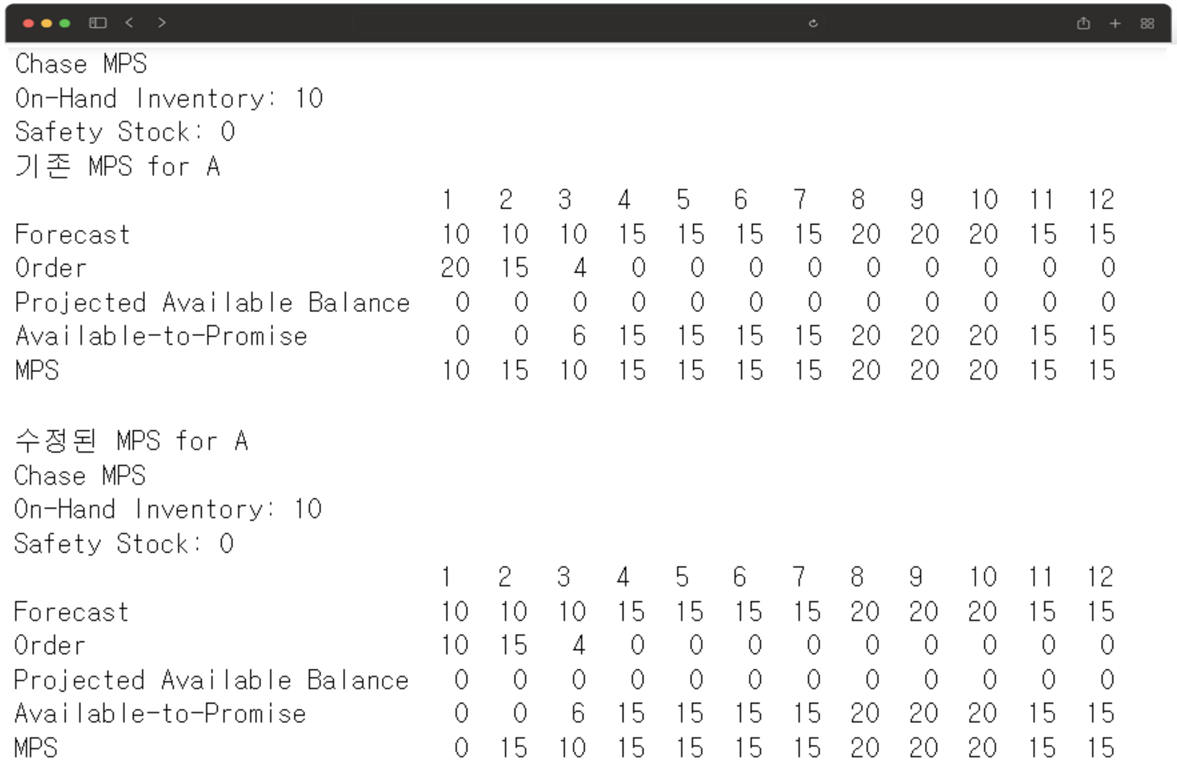


```
Part: A
Lot for Lot
                           1    2    3    4    5    6    7    8    9   10   11   12
Gross requirements         0    8   10   15   15   15   15   20   20   20   15   15
Scheduled receipts        50   20    0    0    0    0    0    0    0    0    0    0
Project available balance 60   72   62   47   32   17    2    0    0    0    0    0
Planned order release      0    0    0    0    0    0   18   20   20   15   15    0
On Hand: 10
Safety Stock: 0
Lead Time: 1
```

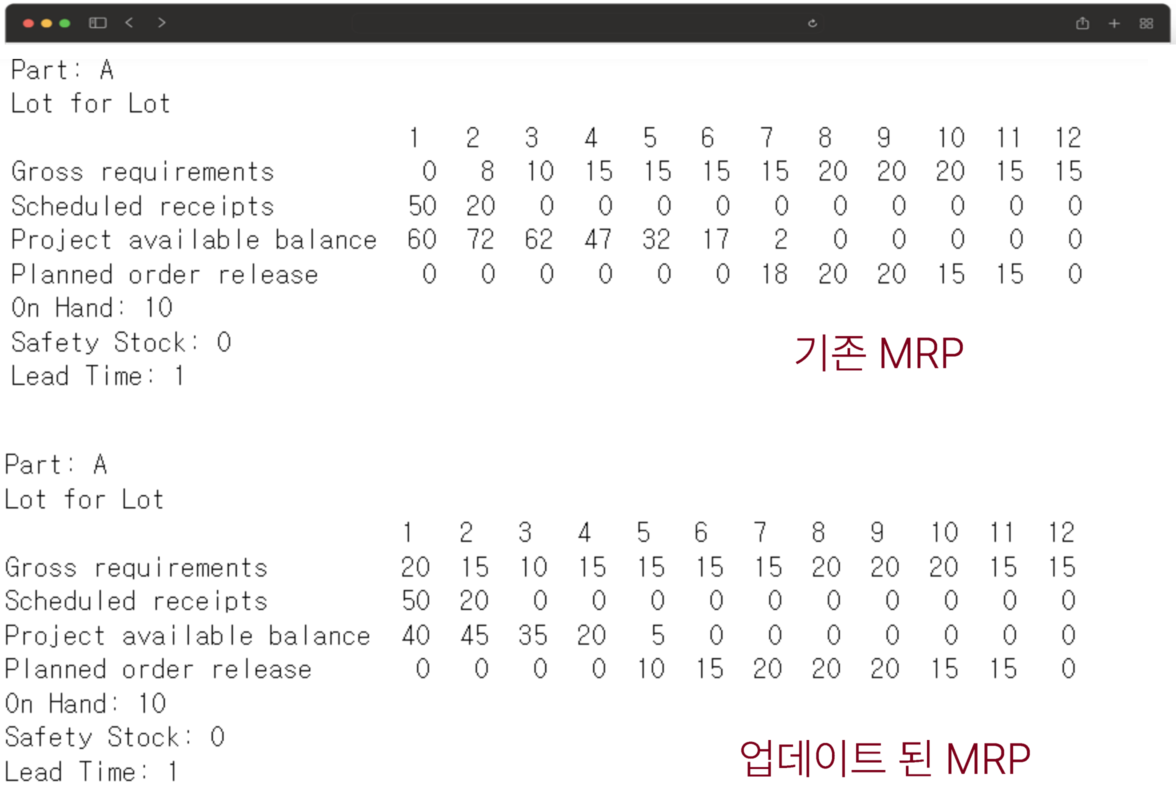


```
Part: B
Lot for Lot
                           1    2    3    4    5    6    7    8    9   10   11   12
Gross requirements         5   20   20   20   25   20   15   20   20   20   15   15
Scheduled receipts        45   10    0    0    0    0    0    0    0    0    0    0
Project available balance 55   45   25    5    0    0    0    0    0    0    0    0
Planned order release      0    0    0   20   20   15   20   20   20   15   15    0
On Hand: 15
Safety Stock: 0
Lead Time: 1
```

결과 분석



```
Chase MPS
On-Hand Inventory: 10
Safety Stock: 0
기존 MPS for A
                              1   2   3   4   5   6   7   8   9  10  11  12
Forecast                     10  10  10  15  15  15  15  20  20  20  15  15
Order                        20  15   4   0   0   0   0   0   0   0   0   0
Projected Available Balance   0   0   0   0   0   0   0   0   0   0   0   0
Available-to-Promise          0   0   6  15  15  15  15  20  20  20  15  15
MPS                          10  15  10  15  15  15  15  20  20  20  15  15

수정된 MPS for A
Chase MPS
On-Hand Inventory: 10
Safety Stock: 0
                              1   2   3   4   5   6   7   8   9  10  11  12
Forecast                     10  10  10  15  15  15  15  20  20  20  15  15
Order                        10  15   4   0   0   0   0   0   0   0   0   0
Projected Available Balance   0   0   0   0   0   0   0   0   0   0   0   0
Available-to-Promise          0   0   6  15  15  15  15  20  20  20  15  15
MPS                           0  15  10  15  15  15  15  20  20  20  15  15
```



```
Part: A
Lot for Lot
                            1   2   3   4   5   6   7   8   9  10  11  12
Gross requirements          0   8  10  15  15  15  15  20  20  20  15  15
Scheduled receipts         50  20   0   0   0   0   0   0   0   0   0   0
Project available balance  60  72  62  47  32  17   2   0   0   0   0   0
Planned order release       0   0   0   0   0   0  18  20  20  15  15   0
On Hand: 10
Safety Stock: 0
Lead Time: 1
```

기존 MRP

```
Part: A
Lot for Lot
                            1   2   3   4   5   6   7   8   9  10  11  12
Gross requirements         20  15  10  15  15  15  15  20  20  20  15  15
Scheduled receipts         50  20   0   0   0   0   0   0   0   0   0   0
Project available balance  40  45  35  20   5   0   0   0   0   0   0   0
Planned order release       0   0   0   0  10  15  20  20  20  15  15   0
On Hand: 10
Safety Stock: 0
Lead Time: 1
```

업데이트 된 MRP

[페어링 품목의 기존/수정 MPS 및 MRP]

Chasing 생산 전략의 MPS Update 시 수정된 MPS, MRP 결과의 일부

CONCLUSION

프로젝트 진행 소감

실제 생산 환경에서 SOP, MPS, MRP 값들의 사용 방안에 대해 직접 다양한 변수를 설정해가며 수행하여 방법론에 대한 이해도 향상

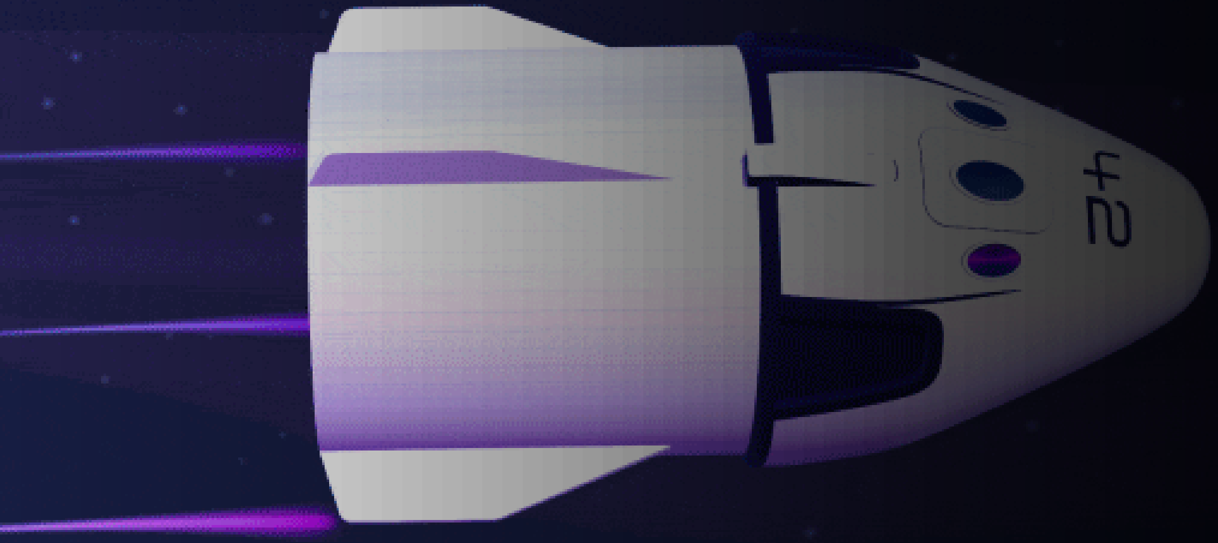
MTO 환경을 가정하고 프로젝트를 수행함으로써 ATO, MTS 등의 생산 특징에 대한 이해도 향상

코드를 통해 방법론들을 직접 구현하면서 개념들간 유기적 연결 관계를 시각적으로 확인 가능

코드를 통해 관련된 계산들을 효율적으로 수행 가능, 복잡한 BOM을 가진 품목에 대해서도 계산 가능

Chasing 방법으로 생산 방식을 통일한 점, MTO 환경에서만 연구를 진행한 점 등을 바탕으로 추후 Batch 방식, MTS 환경에서의 변화 양상 또한 확인해보고 싶음

2024 PRODUCTION PLANNING	Date	Presenter
	2024. 05. 31.	금강산



생산계획 팀프로젝트

중간 발표

고려대학교 산업경영공학부

2020170818 금강산

2020170837 최원준

2021170866 이예일

고려대학교
KOREA UNIVERSITY