MPS 함수 설명

이 함수는 MPS(Master Production Schedule)를 계산하기 위한 알고리즘을 구현한 것으로 보입니다. MPS는 제조업에서 생산 계획을 수립하는데 사용되는 중요한 개념입니다.

함수는 세 개의 매개변수를 입력으로 받습니다:

* data: 데이터 프레임 형태로 제공되는 생산 계획과 예측 데이터입니다.
* lot\_size: 생산을 위한 최소 생산량인 로트 사이즈입니다.
* safety\_stock: 안전 재고로 유지되어야 하는 재고량입니다.

함수는 주어진 데이터에 대해 MPS와 ATP(Available to Promise) 값을 계산하여 데이터 프레임에 추가합니다. 계산은 두 개의 반복문을 통해 이루어집니다.

첫 번째 반복문은 데이터의 길이만큼 반복하며, 각 시점에서의 생산 계획과 예측 값을 고려하여 사용 가능한 재고량과 MPS를 계산합니다. 각 시점에서의 사용 가능한 재고량은 이전 시점의 사용 가능한 재고량에서 주문량과 예측값 중 더 큰 값을 뺀 것입니다. 이후, 사용 가능한 재고량이 안전 재고량보다 작은 경우 MPS를 로트 사이즈로 설정하고, 그렇지 않은 경우 MPS를 0으로 설정합니다. 마지막으로, 사용 가능한 재고량은 이전 시점의 사용 가능한 재고량에서 주문량과 예측값 중 더 큰 값을 뺀 뒤, MPS를 더한 값으로 업데이트합니다.

두 번째 반복문은 데이터의 두 번째부터 끝까지 반복하며, ATP 값을 계산합니다. ATP는 Available to Promise의 약자로, 해당 시점에서 사용 가능한 재고량을 나타냅니다. 첫 번째 시점의 경우, MPS 값이 0인 경우 ATP는 이전 시점의 사용 가능한 재고량에서 주문량을 뺀 값입니다. 그렇지 않은 경우, 현재 시점부터 MPS 값이 0이 나오기 전까지의 주문량을 누적하여 ATP를 계산합니다. 이후, 두 번째 시점 이후의 경우 MPS 값이 0인 경우 ATP는 0이며, 그렇지 않은 경우 현재 시점부터 MPS 값이 0이 나오기 전까지의 주문량을 누적하여 ATP를 계산합니다.

마지막으로, 함수는 계산된 MPS와 ATP 값을 포함하는 데이터 프레임을 반환합니다.

Batch MPS 함수 설명

이 함수는 MPS(Master Production Schedule)를 계산하는 알고리즘을 구현한 것입니다. MPS는 제조업에서 생산 계획을 수립하는 데 사용되는 중요한 개념입니다.

함수는 세 개의 매개변수를 입력으로 받습니다:

* data: 데이터 프레임 형태로 제공되는 생산 계획과 예측 데이터입니다. 이 데이터는 "Available(ending)", "Orders", "Forecast" 열로 구성되어야 합니다.
* safety\_stock: 안전 재고로 유지되어야 하는 재고량입니다.
* lot\_size (선택적 매개변수): 최소 생산량인 로트 사이즈입니다. 기본값은 0이며, 0인 경우 데이터 내 최대 예측값이 로트 사이즈로 설정됩니다.

함수는 입력 데이터 프레임 **data**를 채우기 위해 **fillna(0)**을 사용하여 누락된 값들을 0으로 채웁니다.

그런 다음, 로트 사이즈가 0인 경우, 데이터 내 최대 예측값과 주문량 중 더 큰 값을 **lot\_size**로 설정합니다.

첫 번째 반복문은 데이터의 길이만큼 반복하며, 각 시점에서의 생산 계획과 예측 값을 고려하여 사용 가능한 재고량과 MPS를 계산합니다. 각 시점에서의 사용 가능한 재고량은 이전 시점의 사용 가능한 재고량에서 주문량과 예측값 중 더 큰 값을 뺀 것입니다. 이후, 사용 가능한 재고량이 안전 재고량보다 작은 경우 MPS를 로트 사이즈로 설정하고, 그렇지 않은 경우 MPS를 0으로 설정합니다. 마지막으로, 사용 가능한 재고량은 이전 시점의 사용 가능한 재고량에서 주문량과 예측값 중 더 큰 값을 뺀 뒤, MPS를 더한 값으로 업데이트합니다.

두 번째 반복문은 데이터의 두 번째부터 끝까지 반복하며, ATP(Available to Promise) 값을 계산합니다. ATP는 해당 시점에서 사용 가능한 재고량을 나타냅니다. 첫 번째 시점의 경우, 주문량이 0인 경우 ATP는 초기 사용 가능한 재고량을 나타냅니다. 그렇지 않은 경우, 현재 시점부터 MPS 값이 0이 나오기 전까지의 주문량을 누적하여 ATP를 계산합니다. 이후, 두 번째 시점 이후의 경우 MPS 값이 0인 경우 ATP는 0이며, 그렇지 않은 경우 현재 시점부터 MPS 값이 0이 나오기 전까지의 주문량을 누적하여 ATP를 계산합니다.

마지막으로, 함수는 계산된 MPS와 ATP 값을 포함하는 데이터 프레임을 반환합니다. 입력 데이터 프레임 **data**는 함수 내에서 변경되어 업데이트된 값으로 채워집니다.

Chasing MPS 함수 설명

이 함수는 Chasing MPS(Chase MPS)를 계산하는 알고리즘을 구현한 것입니다. Chasing MPS는 제조업에서 생산 계획을 수립하는 데 사용되는 또 다른 방법입니다.

함수는 **data**라는 데이터 프레임을 입력으로 받습니다. 이 데이터 프레임은 "week", "Forecast", "Orders", "Available(ending)" 열로 구성되어야 합니다.

함수는 먼저 MPS 결과를 저장할 새로운 데이터 프레임인 **mps\_df**를 생성합니다. **mps\_df**는 "week", "Forecast", "Order", "Available(ending)", "ATP", "MPS" 열로 구성됩니다. **week**, **Forecast**, **Order**, **Available(ending)** 열은 입력 데이터 프레임 **data**에서 가져온 값을 그대로 할당합니다.

그런 다음, 1부터 12까지의 반복문을 실행합니다. 각 반복에서 다음 작업을 수행합니다:

1. **Available(ending)** 열을 업데이트합니다. 이전 주차의 **Available(ending)** 값에서 현재 주차의 예측값과 주문량 중 더 큰 값을 뺀 결과를 할당합니다. 만약 이 값이 음수이면 0으로 설정합니다.
2. **MPS** 열을 업데이트합니다. 현재 주차의 예측값에서 이전 주차의 **Available(ending)** 값을 뺀 결과를 할당합니다. 이 값이 음수이면 0으로 설정합니다.
3. **ATP** 열을 업데이트합니다. 현재 주차의 **MPS** 값에서 주문량을 뺀 결과를 할당합니다. 이 값이 음수이면 0으로 설정합니다.

위 작업을 모든 주차에 대해 반복하면서 **mps\_df**가 업데이트되고, 최종적으로 계산된 Chasing MPS 결과를 담은 **mps\_df**를 반환합니다.

즉, 이 함수는 입력 데이터를 기반으로 Chasing MPS를 계산하고, 각 주차별로 **Forecast**, **Order**, **Available(ending)**, **ATP**, **MPS** 값을 계산하여 저장한 데이터 프레임을 반환합니다.

Batch v:s Chasing (MPS) 설명

MPS 계산과 Chasing MPS 계산은 생산 계획을 수립하는 데 사용되는 두 가지 다른 방법입니다. 함수들의 기능 측면에서는 다음과 같은 차이가 있습니다:

MPS 계산:

* MPS(Master Production Schedule)는 예측값과 주문량을 기반으로 생산 계획을 수립하는 방법입니다.
* MPS를 계산하는 알고리즘은 현재 예측값과 주문량을 고려하여 생산 가능한 재고량을 계산하고, 이를 바탕으로 생산 계획인 MPS를 결정합니다.
* MPS 계산은 공급과 수요를 균형시키는 목표를 가지고 있으며, 안전 재고량 등의 요소를 고려하여 생산 계획을 수립합니다.

Chasing MPS 계산:

* Chasing MPS는 예측값과 주문량에 따라 생산 계획을 유동적으로 조정하는 방법입니다.
* Chasing MPS는 주문량이나 예측값에 따라 실시간으로 생산 계획을 조정하여 생산과 주문량을 일치시킵니다.
* Chasing MPS는 주문에 따라 재고를 조절하고 재고 부족이나 과잉 생산을 최소화하기 위해 사용됩니다.
* Chasing MPS 계산은 현재 주차의 예측값과 주문량을 고려하여 생산 가능한 재고량을 계산하고, 이를 바탕으로 MPS를 결정합니다. 이후 ATP(Available to Promise)와 MPS 값의 차이를 통해 생산 계획을 조정합니다.

따라서 MPS 계산은 예측값과 주문량을 기반으로 고정된 MPS를 계산하며, Chasing MPS 계산은 예측값과 주문량에 따라 생산 계획을 유동적으로 조정하는 방법입니다.

MRP 함수 설명

mrp\_abc 함수는 다른 함수들에서 생성된 MPS(Master Production Schedule) 데이터프레임을 입력으로 받아, MRP(Material Requirements Planning)을 계산하는 함수입니다. 아래는 각 함수의 기능에 대한 설명입니다:

1. mrp\_abc 함수:
   * 입력: MPS (df\_a, 10, 120)
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MPS의 MPS 열로 설정합니다.
     + 각 행을 반복하면서 Gross Requirement가 0이 아닌 경우 다음을 수행합니다:
       - 현재 인덱스(idx)가 lead\_time보다 작거나 같으면 Scheduled Receipts 열에 Gross Requirement 값을 할당합니다.
       - 그렇지 않은 경우 Planned Receipts 및 Planned Order Release 열에 Gross Requirement 값을 할당합니다.
     + 계산된 MRP 데이터프레임을 반환합니다.
2. mrp\_dj 함수:
   * 입력: MRP\_a (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_a의 Planned Order Release 열로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다:
       - Gross Requirement가 0이면 이전 행의 PIB 값을 현재 행에 할당합니다.
       - Gross Requirement가 0이 아닌 경우 다음을 수행합니다:
         * 이전 행의 PIB 값에서 Gross Requirement을 뺀 값이 safety\_stock보다 크거나 같으면 현재 행의 PIB 값을 이전 행의 PIB에서 Gross Requirement을 뺀 값으로 할당합니다.
         * 그렇지 않은 경우:

현재 인덱스(idx)가 lead\_time보다 작거나 같으면 Scheduled Receipts 열에 (Gross Requirement - 현재 행의 PIB) 값을 할당합니다.

batch가 0이 아니면 현재 행의 PIB 값을 (이전 행의 PIB + batch - Gross Requirement)로 할당하고, Planned Receipts 열에 batch 값을 할당합니다.

batch가 0이면 현재 행의 PIB 값을 0으로 할당합니다.

Net Requirements, Planned Receipts, Planned Order Release 등의 열을 계산합니다.

* + - 계산된 MRP 데이터프레임을 반환합니다.

1. mrp\_e 함수:
   * 입력: MRP\_a, MRP\_b (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_a의 Planned Order Release \* 2 + MRP\_b의 Planned Order Release \* 2로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다 (나머지 함수와 유사).
     + 계산된 MRP 데이터프레임을 반환합니다.
2. mrp\_f 함수:
   * 입력: MRP\_d, MRP\_h (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_d의 Planned Order Release \* 3 + MRP\_h의 Planned Order Release \* 2로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다 (나머지 함수와 유사).
     + 계산된 MRP 데이터프레임을 반환합니다.
3. mrp\_g 함수:
   * 입력: MRP\_d, MRP\_c (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_d의 Planned Order Release + MRP\_c의 Planned Order Release \* 2로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다 (나머지 함수와 유사).
     + 계산된 MRP 데이터프레임을 반환합니다.
4. mrp\_h 함수:
   * 입력: MRP\_b, MRP\_c (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_b의 Planned Order Release + MRP\_c의 Planned Order Release \* 2로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다 (나머지 함수와 유사).
     + 계산된 MRP 데이터프레임을 반환합니다.
5. mrp\_i 함수:
   * 입력: MRP\_h (이전 함수의 MRP 결과), inventory, safety\_stock, batch
   * 출력: MRP 데이터프레임
   * 기능:
     + MRP 데이터프레임을 초기화하고, NaN 값을 0으로 채웁니다.
     + Gross Requirement 열을 MRP\_h의 Planned Order Release \* 3으로 설정합니다.
     + PIB(Present Inventory Balance) 열의 첫 번째 값에 초기 재고(inventory)를 할당합니다.
     + 각 행을 반복하면서 Gross Requirement를 처리합니다 (나머지 함수와 유사).
     + 계산된 MRP 데이터프레임을 반환합니다.

이러한 함수들은 주어진 MRP 데이터를 처리하고, Gross Requirement, PIB, Net Requirements, Planned Receipts 및 Planned Order Release와 같은 열을 계산하여 MRP 데이터프레임을 반환합니다. 이를 통해 제조 공정에서 필요한 자재의 수요와 납품 일정을 관리할 수 있습니다.

MRP 출력 코드 설명

주어진 코드는 자재 요구 계획(Material Requirements Planning, MRP) 계산을 수행하기 위한 일련의 함수들을 생성하려고 하는 것으로 보입니다. MRP는 생산 계획과 재고 관리 시스템으로, 생산 계획을 충족시키기 위해 필요한 자재와 구성품의 수량을 결정하는 데 도움을 줍니다.

제공된 코드를 기반으로 보면, 이미 다음과 같은 함수들을 정의한 것으로 보입니다:

MPS: 이 함수는 입력 데이터를 기반으로 자재 생산 일정(Material Production Schedule, MPS)을 계산합니다. mrp\_abc: 이 함수는 A, B, C 항목에 대한 MRP 계산을 수행하며, 리드 타임(발주 대기 시간)을 고려합니다. mrp\_dj: 이 함수는 D와 J 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려합니다. mrp\_e: 이 함수는 E 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려하고, A와 B 항목의 데이터를 사용합니다. mrp\_f: 이 함수는 F 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려하고, D와 H 항목의 데이터를 사용합니다. mrp\_g: 이 함수는 G 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려하고, D와 C 항목의 데이터를 사용합니다. mrp\_h: 이 함수는 H 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려하고, B와 C 항목의 데이터를 사용합니다. mrp\_i: 이 함수는 I 항목에 대한 MRP 계산을 수행하며, 재고, 안전 재고 및 배치 크기를 고려하고, H 항목의 데이터를 사용합니다. print\_all: 이 함수는 입력 파일과 구성 매개변수를 기반으로 모든 항목에 대한 MPS 및 MRP 테이블을 출력합니다.

이 함수들을 사용하기 위해서는 필요한 입력 데이터와 구성 파일을 제공해야 합니다. 코드는 지정된 경로 디렉토리에 data\_input.xlsx, a\_input.xlsx, b\_input.xlsx, c\_input.xlsx 및 d\_input.xlsx라는 이름의 입력 파일이 존재한다고 가정합니다.

원하는 경로와 함께 print\_all 함수를 호출하여 MRP 계산을 실행하고 결과를 표시할 수 있습니다. 코드를 실행하기 전에 지정된 형식의 필요한 입력 파일을 준비해야 합니다.