

# 《ERP 与供应链》课程作业

刘铭宸

软件工程 2003 班

U202010783

2023 年 3 月 31 日

# 目录

<b>1 问题一</b>	<b>2</b>
1.1 ER 模型 . . . . .	2
1.2 提取数据的 sql 语句 . . . . .	6
<b>2 问题二</b>	<b>8</b>
2.1 数据分析 . . . . .	8
2.1.1 拟合曲线 . . . . .	8
2.1.2 概率分布 . . . . .	8
2.1.3 统计规律 . . . . .	9
2.2 聚类分析 . . . . .	9
2.2.1 k-means . . . . .	10
2.2.2 DBSCAN . . . . .	10
2.3 库存管理的改进计划 . . . . .	11
<b>3 致谢</b>	<b>12</b>
<b>4 参考资料</b>	<b>12</b>
<b>5 附录</b>	<b>12</b>
5.1 k-means 聚类分析代码 . . . . .	12
5.2 DBSCAN 聚类分析代码 . . . . .	13

# 1 问题一

以 FS 食品公司为应用背景，开发一个库存管理系统，管理 FS 公司的原材料库存和产成品库存。在如下功能设计（图 1）和 ER 图（图 2）设计的基础上，详细定义 ER 模型，并进一步从中提取能够支持库存量动态分析的数据，写出提取数据的 sql 语句。

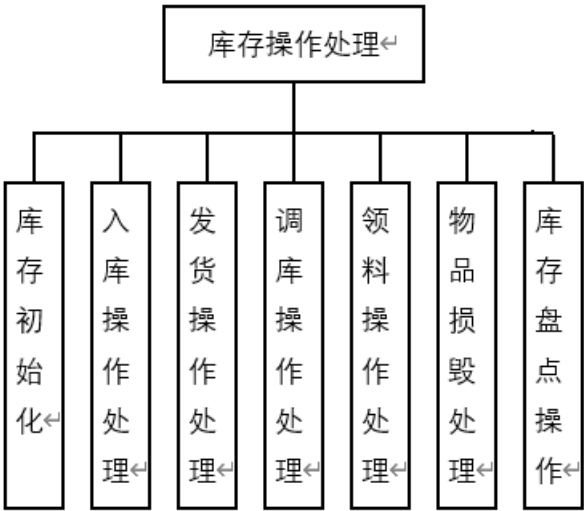


图 1: 库存管理系统功能设计

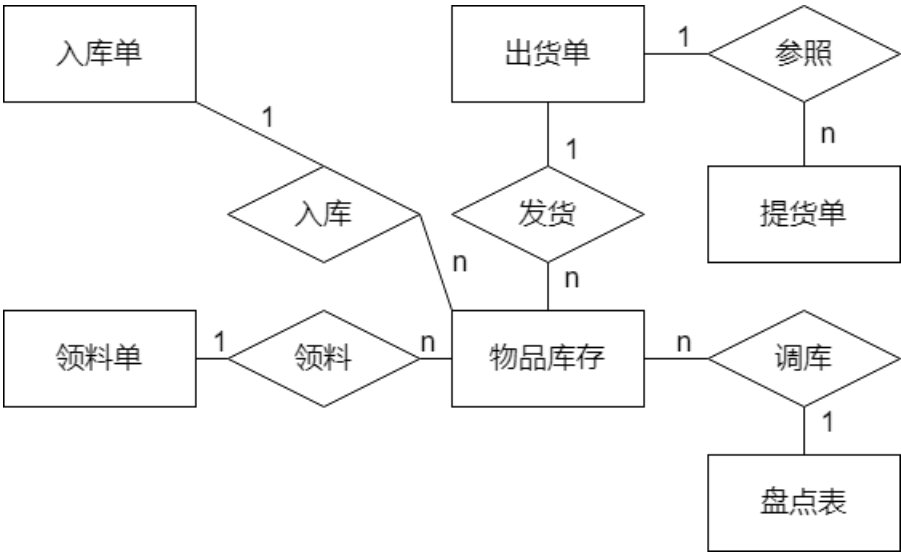


图 2: 库存管理系统 ER 图

## 1.1 ER 模型

根据题目对库存管理系统的功能设计要求并参照 ERP 设计图表，对该库存管理系统的 ER 图进行了扩充，结果如图 3 所示。

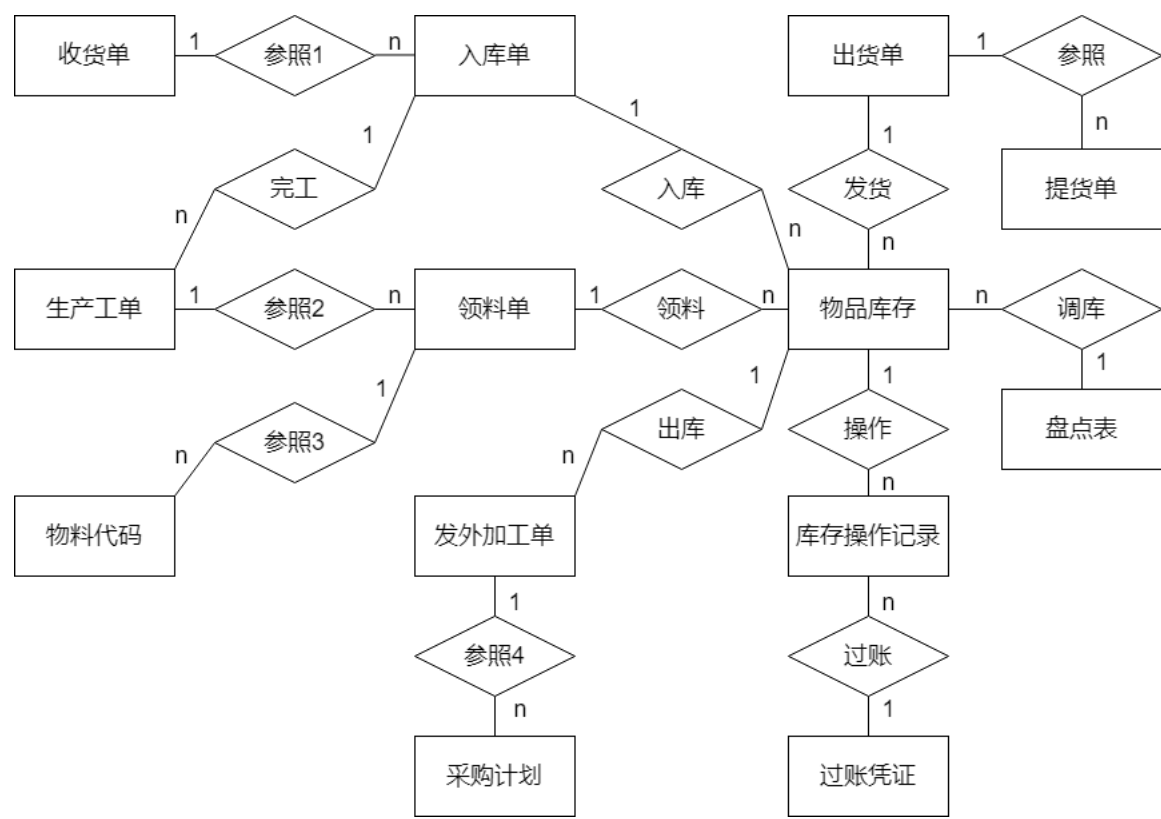


图 3: 详细 ER 图

对 ER 模型的详细定义如下列表所示。

物品库存表 (Inventory)				
Field	Length	Type	Description	PrimKey/ForeKey
ItemID	10	VARCHAR	物品编号	PrimKey
ItemName	50	VARCHAR	物品名称	
Category	20	VARCHAR	物品分类	
Quantity	10	INT	库存数量	
Unit	10	VARCHAR	单位	
LastUpdate	-	DATETIME	最后更新时间	

盘点表 (InventoryCheck)				
Field	Length	Type	Description	PrimKey/ForeKey
CheckID	10	VARCHAR	盘点编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
CheckDate	-	DATE	盘点日期	
Quantity	10	INT	盘点数量	
Operator	20	VARCHAR	操作员	

入库单 (InboundOrder)				
Field	Length	Type	Description	PrimKey/ForeKey
OrderID	10	VARCHAR	入库单编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
InboundDate	-	DATE	入库日期	
Quantity	10	INT	入库数量	
Operator	20	VARCHAR	操作员	

领料单 (PickList)				
Field	Length	Type	Description	PrimKey/ForeKey
PickID	10	VARCHAR	领料单编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
PickDate	-	DATE	领料日期	
Quantity	10	INT	领料数量	
Operator	20	VARCHAR	操作员	

出货单 (OutboundOrder)				
Field	Length	Type	Description	PrimKey/ForeKey
OrderID	10	VARCHAR	出货单编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
OutboundDate	-	DATE	出货日期	
Quantity	10	INT	出货数量	
Operator	20	VARCHAR	操作员	

提货单 (PickUpOrder)				
Field	Length	Type	Description	PrimKey/ForeKey
PickUpID	10	VARCHAR	提货单编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
PickUpDate	-	DATE	提货日期	
Quantity	10	INT	提货数量	
Customer	50	VARCHAR	客户名称	
Operator	20	VARCHAR	操作员	

生产工单表 (ProductionOrder)				
Field	Length	Type	Description	PrimKey/ForeKey
ProductionOrderID	10	VARCHAR	生产工单编号	PrimKey
ItemID	10	VARCHAR	物品编号	ForeKey
StartDate	-	DATE	开始生产日期	
EndDate	-	DATE	结束生产日期	
Quantity	10	INT	生产数量	
Status	20	VARCHAR	生产状态	
Operator	20	VARCHAR	操作员	

## 1.2 提取数据的 sql 语句

为了支持库存量的动态分析，我们可以提取各个表中与库存变动相关的数据。以下是提取数据的 SQL 语句：

查询当前库存总量：

```
1 SELECT ItemID , ItemName , Category , Quantity , Unit , LastUpdate
2 FROM Inventory ;
```

查询某个时间段内的入库数量：

```
1 SELECT ItemID , SUM(Quantity) AS InboundTotal
2 FROM InboundOrder
3 WHERE InboundDate BETWEEN '开始日期' AND '结束日期'
4 GROUP BY ItemID ;
```

查询某个时间段内的领料数量：

```
1 SELECT ItemID , SUM(Quantity) AS PickTotal
2 FROM PickList
3 WHERE PickDate BETWEEN '开始日期' AND '结束日期'
4 GROUP BY ItemID ;
```

查询某个时间段内的出货数量：

```
1 SELECT ItemID , SUM(Quantity) AS OutboundTotal
2 FROM OutboundOrder
3 WHERE OutboundDate BETWEEN '开始日期' AND '结束日期'
4 GROUP BY ItemID ;
```

查询某个时间段内库存变动情况（合并入库、领料、出货、提货数量）：

```
1 WITH InboundTotal AS (
2     SELECT ItemID , SUM(Quantity) AS InQty
3     FROM InboundOrder
4     WHERE InboundDate BETWEEN '开始日期' AND '结束日期'
5     GROUP BY ItemID
6 ),
7 PickTotal AS (
8     SELECT ItemID , SUM(Quantity) AS PickQty
9     FROM PickList
10    WHERE PickDate BETWEEN '开始日期' AND '结束日期'
11    GROUP BY ItemID
```

```
12     ),
13     OutboundTotal AS (
14         SELECT ItemID, SUM(Quantity) AS OutQty
15         FROM OutboundOrder
16         WHERE OutboundDate BETWEEN '开始日期' AND '结束日期'
17         GROUP BY ItemID
18     ),
19     PickUpTotal AS (
20         SELECT ItemID, SUM(Quantity) AS PickUpQty
21         FROM PickUpOrder
22         WHERE PickUpDate BETWEEN '开始日期' AND '结束日期'
23         GROUP BY ItemID
24     )
25 SELECT i.ItemID, i.ItemName, i.Category, i.Unit,
26        COALESCE(it.InQty, 0) AS InboundQty,
27        COALESCE(pt.PickQty, 0) AS PickQty,
28        COALESCE(ot.OutQty, 0) AS OutboundQty,
29        COALESCE(pu.PickUpQty, 0) AS PickUpQty
30 FROM Inventory i
31 LEFT JOIN InboundTotal it ON i.ItemID = it.ItemID
32 LEFT JOIN PickTotal pt ON i.ItemID = pt.ItemID
33 LEFT JOIN OutboundTotal ot ON i.ItemID = ot.ItemID
34 LEFT JOIN PickUpTotal pu ON i.ItemID = pu.ItemID;
```



## 2 问题二

以附件文件“库存数据.xlsx” excel 表格文档数据为 ABCDEF 共 6 种物品的每小时库存量值，进行统计分析和数据挖掘，分析这 6 种物品的库存量变化规律，结合库存管理的原理，提出对这 6 种物品进行库存管理的改进计划。

### 2.1 数据分析

#### 2.1.1 拟合曲线

首先对六种物品的库存量随时间变化趋势进行了分析，如图 4 所示。

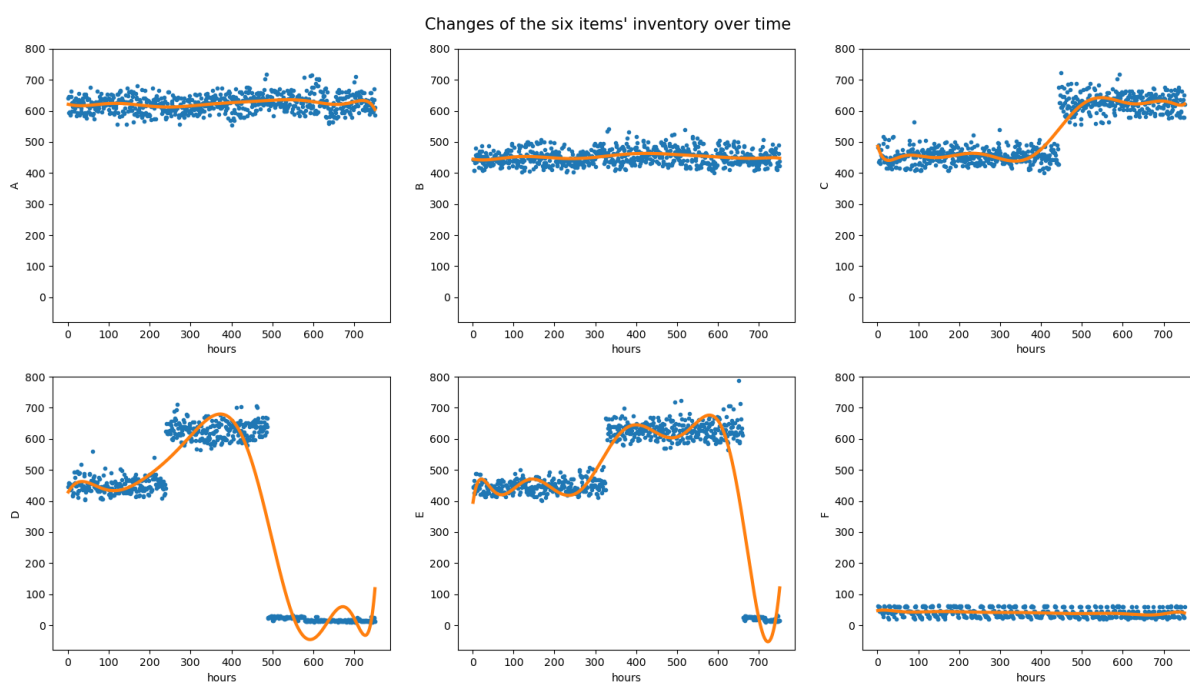


图 4: 六种物品的库存量随时间变化趋势

可以看出，物品 A、物品 B 和物品 F 在这 751 个小时的库存量变化较少，曲线较为平缓，其中物品 A 和 B 的库存量一直维持在较高水平，物品 F 的库存量一直较少；物品 D 和物品 E 的库存量存在较大的波动，随时间先增加后大量减少；物品 C 的库存量变化相对较小。

#### 2.1.2 概率分布

六种物品的库存量的分布直方图与密度曲线如图 5 所示。

可以看到，物品 A 和物品 B 的库存量比较符合正态分布，说明其变化量相对均匀，起伏较小；而物品 C、D、E 的分布较为分散，说明在这一时间段中库存量的变化率相对较大。

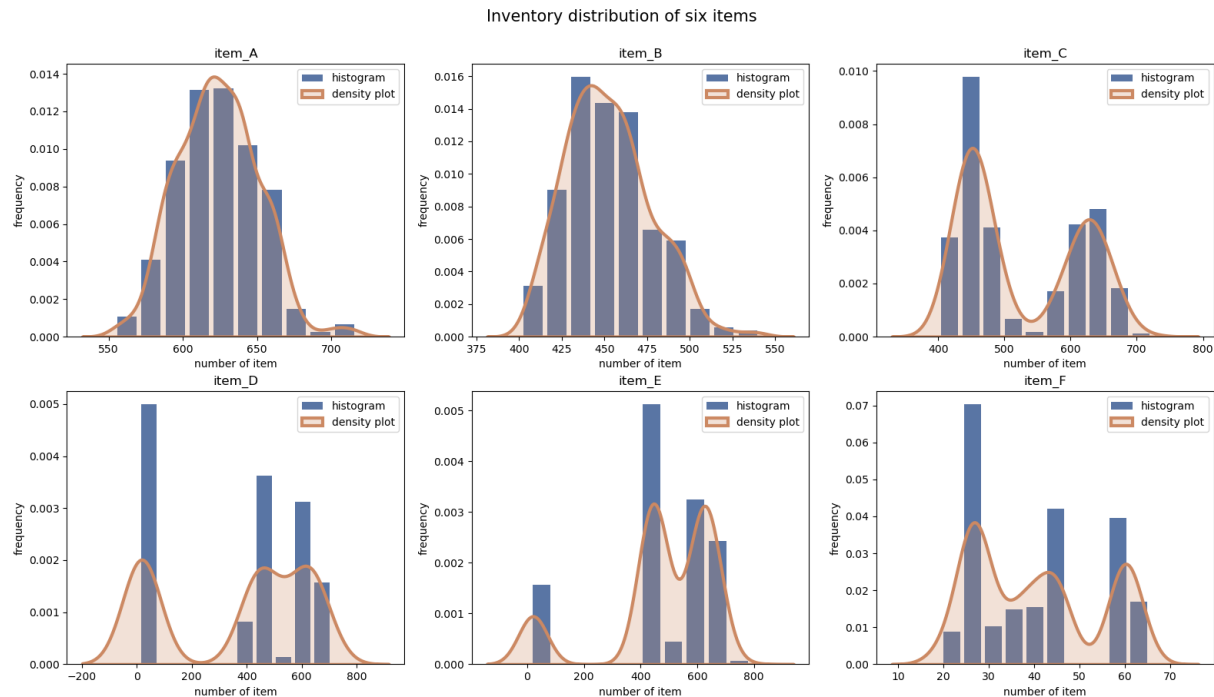


图 5: 六种物品的库存量的分布直方图与密度曲线

2.1.3 统计规律

六种物品的库存量的统计数据如表 2 所示。

	A	B	C	D	E	F
mean	623.66	452.20	524.86	357.83	476.76	40.37
std	27.25	24.78	89.25	260.20	191.03	13.68
min	554	401	401	9.5	11	19.5
max	717	541	722	710	788	65.3

表 1: 库存量统计表

上表列出了这六种物品库存量的平均值、标准差、最小值和最大值，从中可以更加直观地看出其库存量随时间的变化分布。物品 A、B 变化较小，物品 C、D、E 变化较大，物品 F 虽然变化很小，但其库存量一直维持在较低水平。

2.2 聚类分析

ABC 库存控制法是根据库存物品的价格来划分物品的重要程度，以分别采取不同的管理措施。但由于在库存数据中仅提供了六种库存物品的库存量随时间（小时）的变化情况，故采用聚类方法以 751 个小时对应的库存量的值为根据对这六种物品进行分类。

### 2.2.1 k-means

首先对库存数据进行预处理，去除掉几个较为明显的脏数据，如删除表格中库存量属性不是数字的分量；将读入的表格数据进行转置，使每种物品的在 751 个小时中的库存量成为一个数据点（751 维的向量）。接着人为选择聚类数量，在对原始数据进行初步分析后（见 2.1），可以发现按照其库存量随时间分布曲线的相似度大致可以分为 3 类，故将聚类数量设为 3。

之后使用 k-means 方法对这 6 个 751 维的向量进行聚类，选择 k-means++ 作为初始化中心点的方法，最大迭代次数设置为 300，并用生成的 k-means 模型对数据进行聚类。

最后，由于数据维数过高（751 维），所以使用主成分分析（PCA）的方法对数据进行降维以便于可视化。PCA 将数据降为两维，第一个主成分（即第一维）是数据中最大方差的方向，而第二个主成分（第二维）是与第一个主成分正交且具有次大方差的方向。这样，PCA 可以保留原始数据中尽可能多的信息，同时减少数据的维度。聚类结果如图 6 所示。

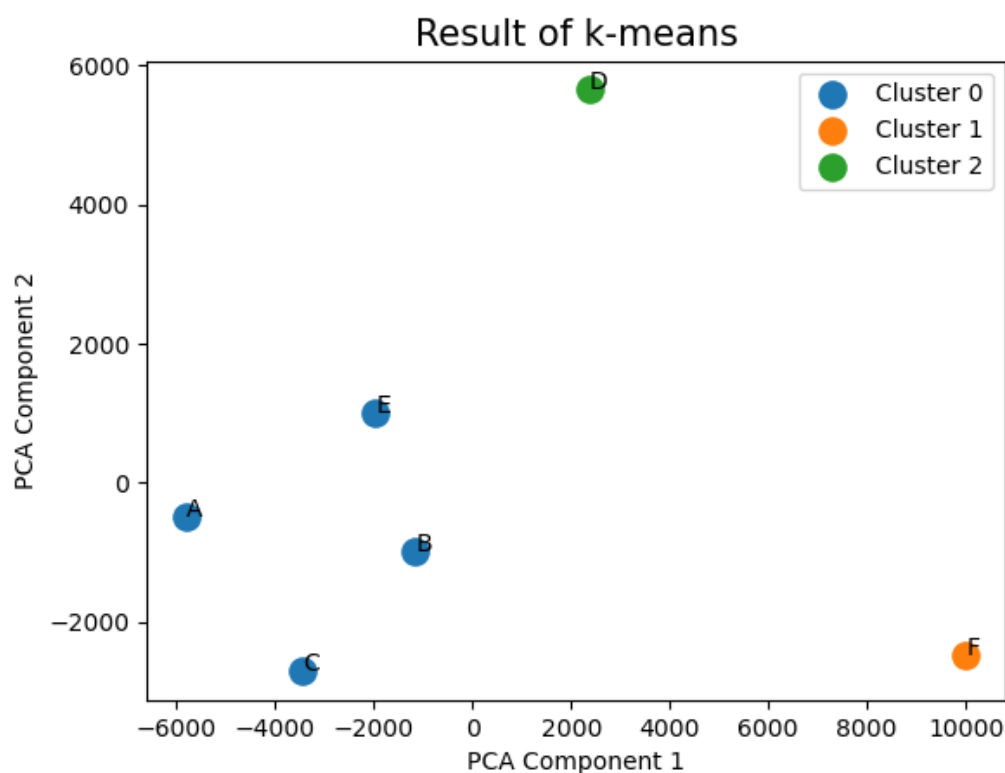


图 6: k-means 聚类结果

如图所示，k-means 聚类方法将这六种物品分为了 3 类，分别为  $\{A, B, C, E\}$ 、 $\{D\}$  和  $\{F\}$ 。

### 2.2.2 DBSCAN

为了验证聚类方法对物品分类的可靠性，我使用了 DBSCAN 聚类方法（Density-Based Spatial Clustering of Applications with Noise）来解决这一问题。与 k-means 不同，DBSCAN 是一种基于密度的聚

类算法，可以找到任意形状的聚类，并识别噪声数据点。DBSCAN 不需要预先设定聚类数量，但需要设置密度半径和最小数据点数作为参数。

计算过程与 k-means 类似，设定合适的密度半径使这六个物品被聚类为三类。结果如图 7 所示。

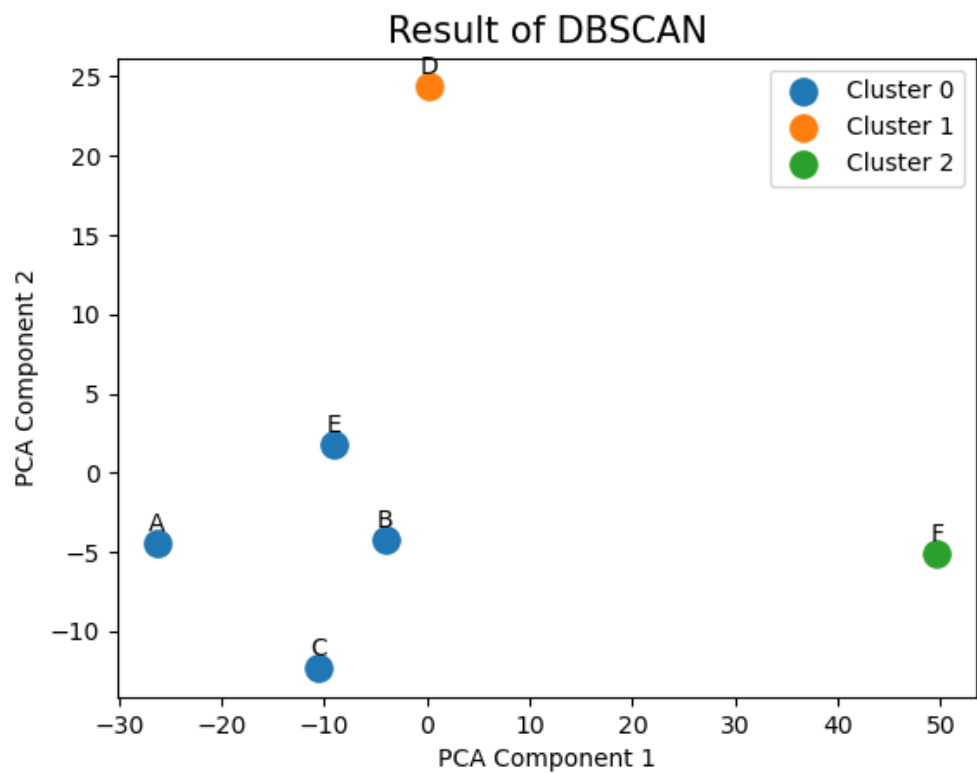


图 7: DBSCAN 聚类结果

如图所示，DBSCAN 聚类方法将这六种物品分为了与 k-means 方法同样的三类，证明聚类分析得到的结果可靠性较高。

### 2.3 库存管理的改进计划

在库存管理中，安全库存和储备库存是两个重要概念。安全库存是为了应对供应链不确定性而设置的额外库存，以防止缺货。储备库存是在预期需求高峰或供应链中断期间使用的额外库存。结合六种物品的库存统计数据的分析，我们可以提出以下针对各物品的库存管理改进计划：

**物品 A：**A 物品的库存波动相对较小。可以将安全库存设置为一个较小的值（例如，标准差的 1 倍）。另外，可以适当减少最大库存量以降低库存成本。

**物品 B：**B 物品的库存波动也较小。与 A 物品类似，可以将安全库存设置为标准差的 1 倍，并适当调整最大库存量以降低库存成本。

**物品 C：**C 物品的库存波动较大。为防止缺货，建议将安全库存设置为标准差的 1.5 倍。同时，需要密切关注库存水平，以应对可能的需求波动。

**物品 D:** D 物品的库存波动非常大。为确保供应稳定,建议将安全库存设置为标准差的 2 倍。此外,需要加强对供应链的监控,以减少库存波动并确保及时供应。

**物品 E:** E 物品的库存波动同样很大。建议将安全库存设置为标准差的 2 倍,并密切关注库存水平。另外,与供应商合作优化供应链管理以减少库存波动。

**物品 F:** F 物品的库存波动较小,但相对于其平均库存量,波动仍然显著。建议将安全库存设置为标准差的 1.5 倍。另外,考虑与供应商协商更频繁的供货安排,以减少库存波动并确保库存充足。

在库存管理中,根据物品的库存波动和供应链特点,我们可以调整安全库存和储备库存水平。对于波动较大的物品(如 C、D 和 E),需要增加安全库存以防止缺货。同时,密切关注库存水平,并与供应商合作优化供应链管理以减少库存波动。对于波动较小的物品(如 A、B 和 F),可以将安全库存设置为较低的水平,降低库存成本。另外,可以考虑与供应商协商更频繁的供货安排,以进一步减少库存波动。

### 3 致谢

感谢方少红老师的教学,希望方老师的课能越办越好!

### 4 参考资料

[1] 数据集:“库存数据.xlsx”。

[2] 罗鸿.ERP 原理·设计·实施(第五版)[M].北京:电子工业出版社,2020.

### 5 附录

#### 5.1 k-means 聚类分析代码

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from sklearn.cluster import KMeans
5 from sklearn.decomposition import PCA
6
7 data = pd.read_excel('库存数据.xlsx')
8 # 提取物品的库存量数据,并转置,使每种物品的库存量成为一个数据点
9 items_data = data.iloc[:, 1:].T
10 # 选择聚类数量
11 num_clusters = 3
12 # 初始化 KMeans 模型
13 kmeans = KMeans(n_clusters=num_clusters, random_state=0)
```

```

14 # 对数据进行聚类
15 kmeans.fit(items_data)
16 # 得到聚类结果
17 labels = kmeans.labels_
18 # 使用主成分分析 (PCA) 降维以便于可视化
19 pca = PCA(n_components=2)
20 items_data_pca = pca.fit_transform(items_data)
21 # 创建一个可视化图
22 fig, ax = plt.subplots()
23 # 为每个聚类绘制散点图
24 for cluster_id in range(num_clusters):
25     cluster_data = items_data_pca[labels == cluster_id]
26     ax.scatter(cluster_data[:, 0], cluster_data[:, 1], label=f'Cluster_{cluster_id}', s=120)
27 # 设置图例和轴标签
28 ax.legend()
29 ax.set_xlabel('PCA_Component_1')
30 ax.set_ylabel('PCA_Component_2')
31 # 为每个数据点添加物品标签
32 item_labels = ['A', 'B', 'C', 'D', 'E', 'F']
33 for i, item_label in enumerate(item_labels):
34     ax.annotate(item_label, (items_data_pca[i, 0], items_data_pca[i, 1]))
35 plt.title("Result_of_k-means", fontsize=15)
36 # 显示图形
37 plt.show()

```

## 5.2 DBSCAN 聚类分析代码

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from sklearn.decomposition import PCA
5 from sklearn.cluster import DBSCAN
6 from sklearn.preprocessing import StandardScaler
7
8 data = pd.read_excel('库存数据.xlsx')

```

```

9  # 提取物品的库存量数据，并转置，使每种物品的库存量成为一个数据点
10 items_data = data.iloc[:, 1:].T
11 # 对数据进行标准化
12 scaler = StandardScaler()
13 scaled_data = scaler.fit_transform(items_data)
14 # 初始化 DBSCAN 模型
15 # 调整 eps 和 min_samples 参数以获得合适的聚类结果
16 dbscan = DBSCAN(eps=25, min_samples=1)
17 # 对数据进行聚类
18 dbscan.fit(scaled_data)
19 # 得到聚类结果
20 labels = dbscan.labels_
21 # 使用 PCA 将数据降至二维
22 pca = PCA(n_components=2)
23 reduced_data = pca.fit_transform(scaled_data)
24 # 创建一个可视化图
25 fig, ax = plt.subplots()
26 # 获取聚类数量（排除噪声点）
27 num_clusters = len(set(labels)) - (1 if -1 in labels else 0)
28 # 为每个聚类绘制散点图并添加物品标签
29 item_labels = ['A', 'B', 'C', 'D', 'E', 'F']
30 for cluster_id in range(num_clusters):
31     cluster_data = reduced_data[labels == cluster_id]
32     ax.scatter(cluster_data[:, 0], cluster_data[:, 1], label=f'Cluster_{cluster_id}', s=120)
33     for point, label in zip(cluster_data, np.array(item_labels)[labels == cluster_id]):
34         ax.annotate(label, (point[0], point[1]), textcoords="offsetpoints", xytext=(0, 5), ha='center')
35 # 如果存在噪声点，将其绘制到图上并添加物品标签
36 if -1 in labels:
37     noise_data = reduced_data[labels == -1]
38     ax.scatter(noise_data[:, 0], noise_data[:, 1], label='Noise')
39     for point, label in zip(noise_data, np.array(item_labels)[labels == -1]):
40         ax.annotate(label, (point[0], point[1]), textcoords="offsetpoints", xytext=(0, 5), ha='center')

```

```
41 # 设置图例和轴标签
42 ax.legend()
43 ax.set_xlabel('PCA_Component_1')
44 ax.set_ylabel('PCA_Component_2')
45 # 显示图形
46 plt.title("Result_of_DBSCAN", fontsize=15)
47 plt.show()
```