

```
In [1]: # 导入所有需要的library和package

import pandas as pd
import numpy as np
import math
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.pylab import mpl

from mpl_toolkits.mplot3d import Axes3D

from plotly.subplots import make_subplots
import plotly.graph_objects as go
from plotly.offline import plot

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

plt.style.use('_mpl-gallery')
mpl.rcParams['font.sans-serif'] = ['SimHei'] #显示中文
mpl.rcParams['axes.unicode_minus'] = False #显示负号
```

导入需要分析的数据表

```
In [2]: # 从总表中导入“三方仓明细汇总” 工作表

df_三方仓明细汇总 = pd.read_excel("自有仓及三方仓库库存库龄数据-2023年6月 (1).xlsx", "三方仓明细汇总")

# 导入仓储费报价
df_仓储费 = pd.read_excel("自有仓及三方仓库库存库龄数据-2023年6月 (1).xlsx", "仓储费报价")

# 导入海外仓订单操作费与仓储费

df_海外仓操作费 = pd.read_excel("运德供应链仓储服务报价.xlsx", "03-海外仓订单操作费与仓储费")

# 导入国内仓订单操作费与仓储费

df_国内仓操作费 = pd.read_excel("运德供应链仓储服务报价.xlsx", "01-国内仓订单操作费与仓储费")
```

```
# 导入头程费用
```

```
df_头程费用 = pd.read_excel("海运结算价-8月上（公布版）.xlsx", '海运结算价8月上-结果')
```

Data Cleaning and Selecting

```
In [3]: # 选择含有深圳的row
df_头程费用 = df_头程费用[(df_头程费用['加权报价'].str.contains('深圳')) & (df_头程费用['Unnamed: 2'].str.contains('普货'))]
```

```
In [4]: ### 清理海外仓数据表 ###
```

```
# col: 只要线路，折合RMB
# 去除nah
```

```
df_头程费用 = df_头程费用[['加权报价', '折合RMB']].dropna()
```

```
In [5]: # rename the columns
df_头程费用.rename(columns = {'加权报价': '线路'}, inplace = True)
```

```
In [6]: ### 清理海外仓数据表 ###
```

```
# save col 2 - 13
# save row 6-8
```

```
df = df_海外仓操作费
df = df.iloc[6:15, 1:13]
df.drop(index = [9,10,11,12], inplace = True)
```

```
# define desired new col names
col_name = ['费用项目', '0-0.5', '0-1', '1-2', '2-5', '5-10', '10-15', '15-20', '20-40', '40-60', '续重5KG', '最高收费']
```

```
# Rename all columns using a for loop
for i, col in enumerate(df.columns):
    df.rename(columns={col: col_name[i]}, inplace=True)
```

```
df_海外仓操作费 = df
```

```
In [7]: ### 清理国内仓数据表 ###
```

```
# 选取需要的行列
```

```

df = df_国内仓操作费
df = df.iloc[18:24, 1:8]
df = df.drop(index = [21, 20, 23])

# define desired new col names
col_name = ['费用项目', '0-2', '2-5', '5-10', '10-15', '续重1KG', '最高收费']

# Rename all columns using a for loop
for i, col in enumerate(df.columns):
    df.rename(columns={col: col_name[i]}, inplace=True)

df_国内仓操作费 = df

```

In [8]: ### 清理仓储费数据表 ###

```

#选取需要的subset
df_仓储费 = df_仓储费.iloc[6:, 0:13]

#重命名column，将第一行内容作为列名
df_仓储费.columns = df_仓储费.iloc[0]

# 不需要第一行
df_仓储费 = df_仓储费.iloc[1:]

# 去掉第一列的空值的行，然后重置index
df_仓储费 = df_仓储费.dropna(subset = ['辅助']).reset_index(drop = True)

# 去掉不需要的列
df_仓储费 = df_仓储费.drop(columns = ['项目', '货型'])

```

In [9]: ### 清理三方仓数据表 ###

```

# 筛选需要用的列组成新的数据表

col = ['仓库', '细分仓', '货型', 'SKU编码', '库龄(天)', '库龄段', '币种', '总体积', 'SKU重量', '在库库存']
df_三方仓明细汇总 = df_三方仓明细汇总[col].dropna().reset_index(drop = True)

# 合并“细分仓”和“货型”
df = df_三方仓明细汇总

# 使用 apply() 函数将两个列合并为一列
df['辅助'] = df.apply(lambda row: row['仓库'] + row['货型'], axis=1)

```

```
# 已经有了辅助列了，所以drop原本的'细分仓'和'货型'
df = df.drop(columns = ['仓库', '货型'])

# 将新家的'辅助列'移到df的第一列
last_col = df.pop('辅助')
df.insert(0, '辅助', last_col)

df_new_三方仓明细 = df

# 合并两个表，关键列为“辅助”
merged_df = pd.merge(df_new_三方仓明细, df_仓储费, on='辅助', how = 'left')
```

将df_海外仓操作费的USD单位转换为人民币

```
In [10]: df = df_海外仓操作费

for i in range(1, df.shape[0]):
    for j in range(1, 12):
        df.iloc[i, j] = df.iloc[i, j] * 6.9240

# 改项目名称
df.iloc[1, 0] = '美国仓入库操作费 (CNY/单件)'
df.iloc[2, 0] = '美国仓出库操作费 (CNY/单件)'
df.iloc[3, 0] = '欧洲仓入库操作费 (CNY/单件)'
df.iloc[4, 0] = '欧洲仓出库操作费 (CNY/单件)'

df_海外仓操作费 = df
```

```
In [11]: # 查看各个表
display(df_海外仓操作费, df_国内仓操作费, merged_df.head(5), df_头程费用)
```

	费用项目	0-0.5	0-1	1-2	2-5	5-10	10-15	15-20	20-40	40-60	续重5KG	最高收费
6	费用项目	0 < X≤0.5KG	0 < X≤1KG	1 < X≤2KG	2 < X≤5KG	5 < X≤10KG	10 < X≤15KG	15 < X≤20KG	20 < X≤40KG	40 < X≤60KG	续重5KG	最高收费
7	美国仓入库操作费 (CNY/单件)	0.921661	0.921661	0.986574	1.285171	1.882366	2.678626	3.275821	4.1544	6.924	0.79626	13.848
8	美国仓出库操作费 (CNY/单件)	4.107766	4.453966	4.726598	7.324397	9.185654	10.499483	13.784056	20.772	48.468	9.853718	152.328
13	欧洲仓入库操作费 (CNY/单件)	0.599372	0.599372	0.599372	0.762973	1.171973	1.683224	2.124421	2.816821	3.509221	0.6924	11.7708
14	欧洲仓出库操作费 (CNY/单件)	2.82216	2.82216	2.82216	5.092114	5.65449	6.441816	8.466369	17.464384	155.944384	8.998015	138.48
	费用项目	0-2	2-5	5-10	10-15	续重1KG	最高收费					
18	费用项目	0 < X≤2 kg	2 < X≤5 kg	5 < X≤10 kg	10 < X≤15 kg	续重1 kg	最高收费					
19	深圳入库费用 (CNY/PCS)	0.336	0.504	1.1025	1.5225	0.168	13.6185					
22	深圳出库费用 (CNY/PCS)	0.504	0.9345	1.5225	1.9425	0.189	15.5505					

	辅助	细分仓	SKU编码	库龄 (天)	库龄 段	币种	总体积	SKU 重量	在 库 库 存	0- 30 天	31- 60 天	61-90 天	91- 120天	121- 180天	181- 270天	271- 360天	360天 以上	旺季附 加费 \n10- 12月	单位
0	易达云美国仓T240	美东12仓	AMA005571_B_US	2	0-30天	USD	26.993064	34.0	168	0	0	0.3177	0.4236	0.75895	1.50025	1.74735	3.2476	0.05295	USD/立方/天
1	易达云美国仓T240	美东12仓	AMA005571_B_US_PP	15	0-30天	USD	40.526640	26.0	301	0	0	0.3177	0.4236	0.75895	1.50025	1.74735	3.2476	0.05295	USD/立方/天
2	易达云美国仓T240	美东12仓	AMA005571_BL_US	2	0-30天	USD	6.732000	26.0	50	0	0	0.3177	0.4236	0.75895	1.50025	1.74735	3.2476	0.05295	USD/立方/天
3	易达云美国仓T240	美东12仓	AMA005571_S_US	2	0-30天	USD	32.616619	33.7	203	0	0	0.3177	0.4236	0.75895	1.50025	1.74735	3.2476	0.05295	USD/立方/天
4	易达云美国仓T240	美东12仓	AMA005571_S_US_PP	15	0-30天	USD	20.330640	26.0	151	0	0	0.3177	0.4236	0.75895	1.50025	1.74735	3.2476	0.05295	USD/立方/天

	线路	折合RMB
1	深圳-美西	28013.0
2	深圳-美东	36118.0
5	深圳-美西快线	29089.0
9	深圳-捷克	47236.0
11	深圳-英国	33649.0
19	深圳-美南	35989.0

分析思路

* __情况__ *

merged_df 记录了库龄数据（库龄，库龄段，总体积，货值等等）和仓储费数据（不同库龄段的仓储费，币种等等）。

df_海外仓操作费 记录了美国仓入库，出库的操作费数据。

df_国内仓操作费 记录了深圳出库，入库的操作费数据。

补充说明：头程费用美东单柜36000人民币，美西单柜25000人民币，单柜按照67方计算

* __目的__ *

哪些库存需要报废处理，用什么指标来判断，相关的费用是什么？通过对库龄的分析，及时处理滞留的库存，计算不同处理方式的费用，减少公司物流在仓储费这一部分的成本。

* __解决问题思路__ *

我们将一个月作为计算的基本时间单位，对比一个月仓储费，以及出库费用，国内发货的入库费用，找出需要报废的指标。

首先我们将“三方仓明细汇总”和“仓储费报价”两个表导入。我们需要计算每一个SKU编码的一个月累计仓储费，所以将两个表联合，方便计算。

(1) 定义一个将所有货币按照汇率转换成人民币的公式

```
In [12]: # 定制transfer_currency公式，将所有非人民币的仓储费转换为人民币
```

```
def transfer_currency(df, col_1, col_2, col_3):  
  
    # col_1: 记录货币种类的列  
    # col_2: 转换后的列  
    # col_3: 原本需要被转换的列  
  
    for i in range(df.shape[0]):  
  
        if df.loc[i, col_1] == 'USD':  
            df.loc[i, col_2] = df.loc[i, col_3] * 6.9240
```

```

elif df.loc[i, col_1] == 'GBP':
    df.loc[i, col_2] = df.loc[i, col_3] * 8.6504

elif df.loc[i, col_1] == 'EUR':
    df.loc[i, col_2] = df.loc[i, col_3] * 7.6361

elif df.loc[i, col_1] == 'CAD':
    df.loc[i, col_2] = df.loc[i, col_3] * 5.0893

elif df.loc[i, col_1] == 'JPY':
    df.loc[i, col_2] = df.loc[i, col_3] * 0.0517

else:
    df.loc[i, col_2] = df.loc[i, col_3]

return df

```

In [13]: # 设计function计算一个月的仓储费

```

def calc_monthly_fee(df, col_1, col_2, col_3, col_4, list_1, list_2):
    # df = merged_df
    # col_1 = 库龄段
    # col_2 = 库龄(天)
    # col_3 = 月仓储
    # col_4 = 总体积
    # list_1 = 库龄段
    # list_2 = 分段天数

    for j in range(df.shape[0]):
        for i in range(len(list_1)-1):

            if (df.loc[j, col_1] == list_1[i]) & ((df.loc[j, col_2] + 30) <= list_2[i+1]):
                df.loc[j, col_3] = df.loc[j, col_4] * (df.loc[j, list_1[i]] * 30)

            if (df.loc[j, col_1] == list_1[i]) & ((df.loc[j, col_2] + 30) > list_2[i+1]):
                df.loc[j, col_3] = df.loc[j, col_4] * ((df.loc[j, list_1[i]] * (list_2[i] - df.loc[j, col_2])) + (df.loc[j, list_1[i+1]] * (df.loc[j, col_2] - df.loc[j, list_1[i]])))

            if df.loc[j, col_1] == '360天以上':
                df.loc[j, col_3] = df.loc[j, col_4] * df.loc[j, '360天以上'] * 30

```



```
return df
```

```
In [14]: # APPLY FUNCTION
库龄段 = ['0-30天', '31-60天', '61-90天', '91-120天', '121-180天', '181-270天', '271-360天', '360天以上']
分段天数 = [0, 30, 60, 90, 120, 180, 270, 360]

df_月仓储 = calc_monthly_fee(merged_df, '库龄段', '库龄(天)', '月仓储', '总体积', 库龄段, 分段天数)
```

```
In [15]: # 将月仓储费用单位转换为人民币

df_月仓储 = transfer_currency(df_月仓储, '币种', '月仓储CNY', '月仓储')

# 选取需要的列
df_月仓储 = df_月仓储[['辅助', '细分仓', 'SKU编码', 'SKU重量', '在库库存', '月仓储CNY', '总体积']]
```

```
In [16]: display(df_月仓储.head(5), df_月仓储.tail(5), df_月仓储.describe())
```

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积
0	易达云美国仓T240	美东12仓	AMA005571_B_US	34.0	168	0.0	26.993064
1	易达云美国仓T240	美东12仓	AMA005571_B_US_PP	26.0	301	0.0	40.526640
2	易达云美国仓T240	美东12仓	AMA005571_BL_US	26.0	50	0.0	6.732000
3	易达云美国仓T240	美东12仓	AMA005571_S_US	33.7	203	0.0	32.616619
4	易达云美国仓T240	美东12仓	AMA005571_S_US_PP	26.0	151	0.0	20.330640

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积
1516	4PX英国仓T50	4PX英国仓	SYB000635_GY_1	0.338	160	229.732479	0.582400
1517	4PX英国仓T50	4PX英国仓	SYB000635_GY_2	0.095	198	77.907475	0.197505
1518	4PX英国仓T50	4PX英国仓	SYG000028	1.310	64	54.984323	0.139392
1519	4PX英国仓T50	4PX英国仓	TGG000001_W	0.150	300	42.761090	0.253500
1520	4PX英国仓T50	4PX英国仓	TGG000001_W	0.150	45	14.999275	0.038025

	SKU重量	在库库存	月仓储CNY	总体积
count	1521.000000	1521.000000	1521.000000	1521.000000
mean	3.806268	22.510191	146.404343	1.364100
std	9.690661	60.679302	618.278124	5.825936
min	0.001000	0.000000	0.000000	0.000000
25%	0.067000	2.000000	0.141280	0.001008
50%	0.377000	4.000000	0.769934	0.005290
75%	1.880000	10.000000	5.729136	0.042632
max	67.500000	801.000000	9617.755668	67.709520

计算美国擦仓入库，出库，深圳出库费用

```
In [17]: df = df_月仓储
# 选取美国仓，以及非美国仓数据作为新的dataframe
美国仓 = df[df['辅助'].str.contains('美国')]
非美国仓 = df[~df['辅助'].str.contains('美国')]
非美国仓 = 非美国仓.reset_index().drop(columns = 'index')

display(美国仓.head(5),非美国仓)
```

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积
0	易达云美国仓T240	美东12仓	AMA005571_B_US	34.0	168	0.0	26.993064
1	易达云美国仓T240	美东12仓	AMA005571_B_US_PP	26.0	301	0.0	40.526640
2	易达云美国仓T240	美东12仓	AMA005571_BL_US	26.0	50	0.0	6.732000
3	易达云美国仓T240	美东12仓	AMA005571_S_US	33.7	203	0.0	32.616619
4	易达云美国仓T240	美东12仓	AMA005571_S_US_PP	26.0	151	0.0	20.330640

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积
0	51英国仓M	51英国2仓	AM001907_PJ24	3.000	1	2.324968	0.010540
1	51英国仓M	51英国2仓	AM001907_PJ39	0.047	10	1.667624	0.007560
2	51英国仓M	51英国2仓	AM001907_PJ40	0.048	10	1.667624	0.007560
3	51英国仓XL	51英国2仓	AM001907_UK	27.000	1	48.509333	0.219912
4	51英国仓M	51英国2仓	AM002071_ARG_S	0.200	1	0.159859	0.002310
...
1433	4PX英国仓T50	4PX英国仓	SYB000635_GY_1	0.338	160	229.732479	0.582400
1434	4PX英国仓T50	4PX英国仓	SYB000635_GY_2	0.095	198	77.907475	0.197505
1435	4PX英国仓T50	4PX英国仓	SYG000028	1.310	64	54.984323	0.139392
1436	4PX英国仓T50	4PX英国仓	TGG000001_W	0.150	300	42.761090	0.253500
1437	4PX英国仓T50	4PX英国仓	TGG000001_W	0.150	45	14.999275	0.038025

1438 rows × 7 columns

```
In [18]: df_国内仓操作费 = df_国内仓操作费.iloc[1:3,:]  
df_美国仓操作费 = df_海外仓操作费.iloc[1:3,:]  
df_非美国仓操作费 = df_海外仓操作费.iloc[3,:]  
display(df_美国仓操作费, df_非美国仓操作费, df_国内仓操作费)
```

	费用项目	0-0.5	0-1	1-2	2-5	5-10	10-15	15-20	20-40	40-60	续重5KG	最高收费
7	美国仓入库操作费 (CNY/单件)	0.921661	0.921661	0.986574	1.285171	1.882366	2.678626	3.275821	4.1544	6.924	0.79626	13.848
8	美国仓出库操作费 (CNY/单件)	4.107766	4.453966	4.726598	7.324397	9.185654	10.499483	13.784056	20.772	48.468	9.853718	152.328

	费用项目	0-0.5	0-1	1-2	2-5	5-10	10-15	15-20	20-40	40-60	续重5KG	最高收费
13	欧洲仓入库操作费 (CNY/单件)	0.599372	0.599372	0.599372	0.762973	1.171973	1.683224	2.124421	2.816821	3.509221	0.6924	11.7708
14	欧洲仓出库操作费 (CNY/单件)	2.82216	2.82216	2.82216	5.092114	5.65449	6.441816	8.466369	17.464384	155.944384	8.998015	138.48

	费用项目	0-2	2-5	5-10	10-15	续重1KG	最高收费
19	深圳入库费用 (CNY/PCS)	0.336	0.504	1.1025	1.5225	0.168	13.6185
22	深圳出库费用 (CNY/PCS)	0.504	0.9345	1.5225	1.9425	0.189	15.5505

设计公式计算,计算国内仓和海外仓的出库入库操作费用

```
In [19]: def calc_出入库费用(df, df_2):

# df : the original dataframe which needs to be calculated
# df_2: the daraframe which has the fee information

for i in range(df.shape[0]):
    if (0 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 1):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,1]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,1]

    elif (1 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 2):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,3]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,3]

    elif (2 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 5):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,4]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,4]

    elif (5 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 10):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,5]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,5]

    elif (10 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 15):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,6]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,6]

    elif (15 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 20):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,7]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,7]

    elif (20 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 40):
        df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,8]
        df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,8]
```

```

elif (40 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 60):
    df.loc[i, '入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,9]
    df.loc[i, '出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,9]

elif (60 < df.loc[i, 'SKU重量']):
    df.loc[i, '入库费用'] = ((int(df.loc[i, 'SKU重量'] - 60) / 5) * df_2.iloc[0,10] + df_2.iloc[0,9]) * df.loc[i, '在库库存']
    df.loc[i, '出库费用'] = ((int(df.loc[i, 'SKU重量'] - 60) / 5) * df_2.iloc[1,10] + df_2.iloc[1,9]) * df.loc[i, '在库库存']

elif (((int(df.loc[i, 'SKU重量'] - 60) / 5) * df_2.iloc[0,10] + df_2.iloc[0,9]) > df_2.iloc[0,10]):
    df.loc[i, '入库费用'] = df_2.iloc[1,10] * df.loc[i, '在库库存']

elif (((int(df.loc[i, 'SKU重量'] - 60) / 5) * df_2.iloc[1,10] + df_2.iloc[1,9]) > df_2.iloc[1,10]):
    df.loc[i, '出库费用'] = df_2.iloc[1,10] * df.loc[i, '在库库存']

return df

```

In [20]: def calc_国内仓出入库费用(df, df_2):

```

# df : the original dataframe which needs to be calculated
# df_2: the dataframe which has the fee information

for i in range(df.shape[0]):
    if (0 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 2):
        df.loc[i, '国内仓入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,1]
        df.loc[i, '国内仓出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,1]

    elif (2 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 5):
        df.loc[i, '国内仓入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,2]
        df.loc[i, '国内仓出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,2]

    elif (5 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 10):
        df.loc[i, '国内仓入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,3]
        df.loc[i, '国内仓出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,3]

    elif (10 < df.loc[i, 'SKU重量']) & (df.loc[i, 'SKU重量'] <= 15):
        df.loc[i, '国内仓入库费用'] = df.loc[i, '在库库存'] * df_2.iloc[0,4]
        df.loc[i, '国内仓出库费用'] = df.loc[i, '在库库存'] * df_2.iloc[1,4]

    elif (15 < df.loc[i, 'SKU重量']):
        # 费用超出了最高收费的情况
        if (((int(df.loc[i, 'SKU重量'] - 15)) * df_2.iloc[0,5] + df_2.iloc[0,4]) > df_2.iloc[0,5]):
            # 按照最高收费来算

```

```
df.loc[i, '国内仓入库费用'] = df_2.iloc[0,5] * df.loc[i, '在库库存']

if (((int(df.loc[i, 'SKU重量'] - 15)) * df_2.iloc[1,5] + df_2.iloc[1,4]) > df_2.iloc[1,5]):
    df.loc[i, '国内仓出库费用'] = df_2.iloc[1,5] * df.loc[i, '在库库存']
else:
    df.loc[i, '国内仓入库费用'] = ((int(df.loc[i, 'SKU重量'] - 15)) * df_2.iloc[0,5] + df_2.iloc[0,5]) * df.loc[i, '在库库存']
    df.loc[i, '国内仓出库费用'] = ((int(df.loc[i, 'SKU重量'] - 15)) * df_2.iloc[1,5] + df_2.iloc[1,5]) * df.loc[i, '在库库存']

return df
```

```
In [27]: 美国仓 = calc_出入库费用(美国仓,df_美国仓操作费)
非美国仓 = calc_出入库费用(非美国仓,df_非美国仓操作费)
display(美国仓.head(5), 非美国仓.head(5))
```

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
0	易达云美国仓T240	美东12仓	AMA005571_B_US	34.0	168	0.0	26.993064	697.9392	3489.696	28.224	31.752
1	易达云美国仓T240	美东12仓	AMA005571_B_US_PP	26.0	301	0.0	40.526640	1250.4744	6252.372	50.568	56.889
2	易达云美国仓T240	美东12仓	AMA005571_BL_US	26.0	50	0.0	6.732000	207.7200	1038.600	8.400	9.450
3	易达云美国仓T240	美东12仓	AMA005571_S_US	33.7	203	0.0	32.616619	843.3432	4216.716	34.104	38.367
4	易达云美国仓T240	美东12仓	AMA005571_S_US_PP	26.0	151	0.0	20.330640	627.3144	3136.572	25.368	28.539

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
0	51英国仓M	51英国2仓	AM001907_PJ24	3.000	1	2.324968	0.010540	0.762973	5.092114	0.504	0.9345
1	51英国仓M	51英国2仓	AM001907_PJ39	0.047	10	1.667624	0.007560	5.993723	28.221601	3.360	5.0400
2	51英国仓M	51英国2仓	AM001907_PJ40	0.048	10	1.667624	0.007560	5.993723	28.221601	3.360	5.0400
3	51英国仓XL	51英国2仓	AM001907_UK	27.000	1	48.509333	0.219912	2.816821	17.464384	0.168	0.1890
4	51英国仓M	51英国2仓	AM002071_ARG_S	0.200	1	0.159859	0.002310	0.599372	2.822160	0.336	0.5040

入库出库数据

```
In [26]: 美国仓 = calc_国内仓出入库费用(美国仓, df_国内仓操作费)
非美国仓 = calc_国内仓出入库费用(非美国仓, df_国内仓操作费)
display(美国仓.head(5), 非美国仓.head(5))
```

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
0	易达云美国仓T240	美东12仓	AMA005571_B_US	34.0	168	0.0	26.993064	697.9392	3489.696	28.224	31.752
1	易达云美国仓T240	美东12仓	AMA005571_B_US_PP	26.0	301	0.0	40.526640	1250.4744	6252.372	50.568	56.889
2	易达云美国仓T240	美东12仓	AMA005571_BL_US	26.0	50	0.0	6.732000	207.7200	1038.600	8.400	9.450
3	易达云美国仓T240	美东12仓	AMA005571_S_US	33.7	203	0.0	32.616619	843.3432	4216.716	34.104	38.367
4	易达云美国仓T240	美东12仓	AMA005571_S_US_PP	26.0	151	0.0	20.330640	627.3144	3136.572	25.368	28.539

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
0	51英国仓M	51英国2仓	AM001907_PJ24	3.000	1	2.324968	0.010540	0.762973	5.092114	0.504	0.9345
1	51英国仓M	51英国2仓	AM001907_PJ39	0.047	10	1.667624	0.007560	5.993723	28.221601	3.360	5.0400
2	51英国仓M	51英国2仓	AM001907_PJ40	0.048	10	1.667624	0.007560	5.993723	28.221601	3.360	5.0400
3	51英国仓XL	51英国2仓	AM001907_UK	27.000	1	48.509333	0.219912	2.816821	17.464384	0.168	0.1890
4	51英国仓M	51英国2仓	AM002071_ARG_S	0.200	1	0.159859	0.002310	0.599372	2.822160	0.336	0.5040

直接报废

立即报废指标： 出库费用 < 月仓储费 如果当下一个月的仓储费已经大于出库的费用了，那么继续存在仓库只会产生更多的仓储费。这个时候这批货物应该选择及时出库，报废处理。

```
In [23]: df美国仓_报废 = 美国仓[美国仓['月仓储CNY'] > 美国仓['出库费用']]
df非美国仓_报废 = 非美国仓[非美国仓['月仓储CNY'] > 非美国仓['出库费用']]
frames = [df美国仓_报废, df非美国仓_报废]
df_报废 = pd.concat(frames)
df_报废
```

Out[23]:

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
33	西邮美国仓 T240	西邮美西四仓	AMA005571_B_US	34.000	90	2739.419636	14.460570	373.896000	1869.480000	15.1200	17.0100
34	西邮美国仓 T240	西邮美西四仓	AMA005571_S_US	33.700	168	5113.583320	26.993064	697.939200	3489.696000	28.2240	31.7520
35	西邮美国仓 T240	西邮美西四仓	AMA005689_US_S	30.000	57	2559.254838	13.509534	236.800800	1184.004000	9.5760	10.7730
36	西邮美国仓 TM	西邮美西五仓 大件仓	AMA005570_US_B	56.200	3	191.492519	1.117428	20.772000	145.404000	0.5040	0.5670
39	西邮美国仓 T240	西邮美西五仓 大件仓	AMA005875_US_BL	35.000	90	4808.003443	32.832000	373.896000	1869.480000	15.1200	17.0100
...
1424	4PX英国仓 T120	4PX英国仓	OSA004930_BL	6.336	200	2694.544237	6.831000	234.394646	1130.897959	220.5000	304.5000
1425	4PX英国仓 T120	4PX英国仓	OSA004930_BL	6.336	68	916.145041	2.322540	79.694179	384.505306	74.9700	103.5300
1426	4PX英国仓 T120	4PX英国仓	OSA004930_G	6.280	40	538.908847	1.366200	46.878929	226.179592	44.1000	60.9000
1427	4PX英国仓 T120	4PX英国仓	OSA004930_P	6.275	6	80.836327	0.204930	7.031839	33.926939	6.6150	9.1350
1428	4PX英国仓 T120	4PX英国仓	OSA004930_P	6.275	17	229.036260	0.580635	19.923545	96.126327	18.7425	25.8825

243 rows × 11 columns

暂时不报废

不需要立即报废的，对比国内出库，美国入库，如果月仓储大于这些费用，就进一步计算海运费用。

月仓储费用对比出库入库费用，如果月仓储小于这些费用， 不需要从国内发货。

In [29]: # 美国仓：月仓储小于出库费用，并且月仓储大于美国入库费用以及国内出入库费用

```
df_a = 美国仓[(美国仓['月仓储CNY'] <= 美国仓['出库费用']) & (  
    美国仓['月仓储CNY'] > (美国仓['入库费用'] + 美国仓['国内仓入库费用'] + 美国仓['国内仓出库费用']))]  
df_a.head(10)
```

Out[29]:

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
17	易达云美国仓TM	美西18仓	AMA005900_US_B	54.0	56	1332.949340	16.230900	387.744000	2714.208000	9.408	10.584
18	易达云美国仓TM	美西18仓	AMA005900_US_S	54.0	40	952.106672	11.593500	276.960000	1938.720000	6.720	7.560
37	西邮美国仓T240	西邮美西五仓大件仓	AMA005690_US_B	39.5	42	828.916215	8.867880	174.484800	872.424000	7.056	7.938
38	西邮美国仓T240	西邮美西五仓大件仓	AMA005690_US_BK	39.5	45	888.124516	9.501300	186.948000	934.740000	7.560	8.505
45	西邮美国仓TM	西邮美西五仓大件仓	SPC000036_B_P	54.0	58	1092.184141	11.307448	401.592000	2811.144000	9.744	10.962
46	西邮美国仓TM	西邮美西五仓大件仓	SPC000036_BL_P	67.5	40	1036.545712	10.731420	321.550560	2490.528180	6.720	7.560
47	西邮美国仓TM	西邮美西五仓大件仓	SPC000036_GR_P	67.2	56	1451.163996	15.023988	450.170784	3486.739452	9.408	10.584
48	西邮美国仓TM	西邮美西五仓大件仓	SPC000036_W_P	67.2	70	1813.954995	18.779985	562.713480	4358.424315	11.760	13.230
49	西邮美国仓T120	西邮美东一仓	HZG000045_W	21.8	340	3775.906907	23.304960	1412.496000	7062.480000	57.120	64.260
50	西邮美国仓T240	西邮美东二仓	AMA005417	33.6	175	2916.491621	31.201100	727.020000	3635.100000	29.400	33.075

In [28]: # 非美国仓：月仓储小于出库费用，并且月仓储大于美国入库费用以及国内出入库费用

```
df_b = 非美国仓[(非美国仓['月仓储CNY'] <= 非美国仓['出库费用']) & (  
    非美国仓['月仓储CNY'] > (非美国仓['入库费用'] + 非美国仓['国内仓入库费用'] + 非美国仓['国内仓出库费用']))]  
df_b.head(10)
```

Out[28]:

	辅助	细分仓	SKU编码	SKU重量	在库库存	月仓储CNY	总体积	入库费用	出库费用	国内仓入库费用	国内仓出库费用
0	51英国仓M	51英国2仓	AM001907_PJ24	3.00	1	2.324968	0.010540	0.762973	5.092114	0.504	0.9345
41	51英国仓L	51英国2仓	AM003672_W	26.00	1	8.098800	0.101214	2.816821	17.464384	0.168	0.1890
42	51英国仓L	51英国2仓	AM003965_31	1.92	35	91.534035	0.414960	20.978030	98.775603	11.760	17.6400
63	51英国仓L	51英国2仓	AMA005499_P	2.31	1	4.573525	0.016445	0.762973	5.092114	0.504	0.9345
64	51英国仓L	51英国2仓	AMA005499_P	2.31	35	126.963326	0.575575	26.704039	178.223985	17.640	32.7075
66	51英国仓L	51英国2仓	AMA005669_P_S	2.05	5	15.485081	0.070200	3.814863	25.460569	2.520	4.6725
72	51英国仓M	51英国2仓	AMA005686_PJ2	0.50	3	7.941067	0.036000	1.798117	8.466480	1.008	1.5120
73	51英国仓L	51英国2仓	AMA005686_PJ22	3.50	1	2.501436	0.011340	0.762973	5.092114	0.504	0.9345
74	51英国仓M	51英国2仓	AMA005687_PJ2	0.50	2	5.294045	0.024000	1.198745	5.644320	0.672	1.0080
79	51英国仓L	51英国2仓	AMA005700_W	19.60	1	5.457805	0.068209	2.124421	8.466369	0.168	0.1890