

## ABC分析\_0716

```
# パッケージ読み込み
import numpy as np
import pandas as pd

from statistics import mean
from sklearn.metrics import mean_absolute_error

import matplotlib.pyplot as plt
# グラフのスタイルとサイズ
plt.style.use('ggplot')
plt.rcParams['figure.figsize'] = [12, 9]

#画像を読み込んだものを、CSVファイル化
# Extracted data 画像を読み込んだもの
data = {
    'ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21],
    '部品名': ['P050-A00', 'P050-B00', 'P065-A00', 'P065-B00', 'P075-A00', 'P075-B00',
    'P100-A00', 'P100-A01', 'P100-A02', 'P100-B00',
    'P100-B01', 'P150-A00', 'P150-A01', 'P150-B00', 'P150-B01', 'P150-B02',
    'P200-A00', 'P200-A01', 'P200-A02', 'P200-B00', 'P200-B01'],
    'Total': [92, 8, 8, 6, 169, 20, 5644, 348, 0, 949, 86, 4311, 1679, 357, 711, 193,
    2685, 388, 0, 365, 165]
}

# Create DataFrame
df = pd.DataFrame(data)

# Save to CSV データをCSVファイルにする
df.to_csv('Warehouse_data.csv')

#ABC分析の実施
# Recalculate the ABC analysis correctly by sorting the Total values in descending
order

# Sort the DataFrame by 'Total' in descending order
df_sorted = df.sort_values(by='Total', ascending=False).reset_index(drop=True)

# Calculate the cumulative sum and cumulative percentage
df_sorted['Cumulative_Sum'] = df_sorted['Total'].cumsum()
df_sorted['Cumulative_Percentage'] = 100 * df_sorted['Cumulative_Sum'] /
df_sorted['Total'].sum()

# Define ABC categories based on cumulative percentage
df_sorted['ABC_Category'] = 'C'
df_sorted.loc[df_sorted['Cumulative_Percentage'] <= 70, 'ABC_Category'] = 'A'
df_sorted.loc[(df_sorted['Cumulative_Percentage'] > 70) &
(df_sorted['Cumulative_Percentage'] <= 90), 'ABC_Category'] = 'B'

#部品ごとの集計グラフ
# Prepare data for plotting
df_sorted = df.sort_values(by='Total', ascending=False)
category_counts = df['ABC_Category'].value_counts().sort_index()

# Plot Total vs ID
plt.figure(figsize=(12, 6))
plt.bar(df_sorted['部品名'], df_sorted['Total'], color='skyblue')
plt.xlabel('部品名')
plt.ylabel('Total')
plt.title('Total Values for Each Part')
plt.xticks(rotation=90)
plt.tight_layout()
plt.savefig('total_values_bar_chart.png')
```

```

plt.close()

#
# Corrected ABC analysis result
abc_analysis_corrected = pd.DataFrame({
    'ID': [7, 12, 17, 13, 10, 15, 8, 14, 20, 1, 11, 16, 21, 18, 5, 6, 2, 4, 3, 19, 9],
    '部品名': ['P100-A00', 'P150-A00', 'P200-A00', 'P150-A01', 'P100-B00', 'P150-B01',
    'P100-A01', 'P150-B00', 'P200-B00', 'P050-A00', 'P100-B01', 'P150-B02', 'P200-B01',
    'P200-A01', 'P075-A00', 'P075-B00', 'P050-B00', 'P065-B00', 'P065-A00', 'P200-A02',
    'P100-A02'],
    'Total': [5644, 4311, 2685, 1679, 949, 711, 348, 357, 365, 92, 86, 193, 165, 388,
    169, 20, 8, 6, 8, 0, 0],
    'Cumulative Sum': [5644, 9955, 12640, 14319, 15268, 15979, 16327, 16684, 17049,
    17141, 17227, 17420, 17585, 17973, 18142, 18162, 18170, 18176, 18184, 18184, 18184],
    'Cumulative_Percentage': [31.05, 54.78, 69.51, 78.71, 83.93, 87.88, 89.78, 91.70,
    93.68, 94.18, 94.66, 95.77, 96.69, 98.82, 99.61, 99.72, 99.76, 99.80, 99.84, 99.84,
    99.84],
    'ABC_Category': ['A', 'A', 'A', 'B', 'B', 'B', 'B', 'C', 'C', 'C', 'C', 'C', 'C',
    'C', 'C', 'C', 'C', 'C', 'C', 'C']
})

# Calculate ABC category distribution
category_counts_corrected =
abc_analysis_corrected['ABC_Category'].value_counts().sort_index()

# Plot ABC Category Distribution
plt.figure(figsize=(8, 6))
category_counts_corrected.plot(kind='bar', color=['red', 'yellow', 'green'])
plt.xlabel('ABC Category')
plt.ylabel('Number of Items')
plt.title('Distribution of ABC Categories (Corrected)')
plt.tight_layout()
plt.savefig('distribution_abc.png')
plt.close()

# Plot Cumulative Percentage Line Plot
plt.figure(figsize=(12, 6))
plt.plot(abc_analysis_corrected['部品名'],
abc_analysis_corrected['Cumulative_Percentage'], marker='o', color='b')
plt.axhline(y=70, color='r', linestyle='--', label='70% Threshold')
plt.axhline(y=90, color='g', linestyle='--', label='90% Threshold')
plt.xlabel('部品名')
plt.ylabel('Cumulative Percentage')
plt.title('Cumulative Percentage of Total Values (Corrected)')
plt.legend()
plt.xticks(rotation=90)
plt.tight_layout()
plt.savefig('cumulative_percentage.png')
plt.close()

```