In [3]:

```
# 1. 导入依赖
import pandas as pd
import numpy as np
import json
import os
from pathlib import Path

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
plt.rcParams['font.sans-serif'] = ['STHeiti', 'SimHei', 'Microsoft YaHei', '
plt.rcParams['axes.unicode_minus'] = False
import matplotlib.pyplot as plt

# 让图像内嵌在 notebook 里显示
%matplotlib inline
```

In [4]:

```
# 2. 读取月度特征数据
monthly_path = "data/sh000016_monthly_features.csv"  # 如果路径不同，在这里改
df_month = pd.read_csv(monthly_path)


df_month["month"] = pd.to_datetime(df_month["month"])
df_month["month"] = df_month["month"].dt.to_period("M").dt.to_timestamp()
df_month.head()
```

Out[4]:

	month	close_month_end	ret_month	vol_month_sum	vol_month_chg	vol_20_ann
0	2015-01-01	2405.38	-0.068249	240759567500	-0.372974	
1	2015-02-01	2474.59	0.028773	113685256600	-0.527806	
2	2015-03-01	2754.66	0.113178	257993309000	1.269365	
3	2015-04-01	3250.49	0.179997	368238254400	0.427317	
4	2015-05-01	3111.33	-0.042812	259537919800	-0.295190	

In [5]:

```
# 3. 处理日期列并设为索引
date_col = "month"   # 如果你的列叫 'month' 就改成 "month"
df_month[date_col] = pd.to_datetime(df_month[date_col])
df_month = df_month.sort_values(date_col).set_index(date_col)

df_month.head()
```

Out[5]:

	close_month_end	ret_month	vol_month_sum	vol_month_chg	vol_20_annual_
month					
2015-01-01	2405.38	-0.068249	240759567500	-0.372974	
2015-02-01	2474.59	0.028773	113685256600	-0.527806	
2015-03-01	2754.66	0.113178	257993309000	1.269365	
2015-04-01	3250.49	0.179997	368238254400	0.427317	
2015-05-01	3111.33	-0.042812	259537919800	-0.295190	

In [6]:

```python
# 3. 指定用于 Regime 聚类的特征列
feature_cols = [
    "ret_month",
    "vol_month_sum",
    "vol_20_annual_month_end"
]

# 检查列是否存在
missing = [c for c in feature_cols if c not in df_month.columns]
if missing:
    print("缺失列：", missing)
else:
    print("聚类特征列：", feature_cols)

# 4. 取出特征矩阵并标准化
X_raw = df_month[feature_cols].copy().dropna()
scaler = StandardScaler()
X = scaler.fit_transform(X_raw)

X.shape
```

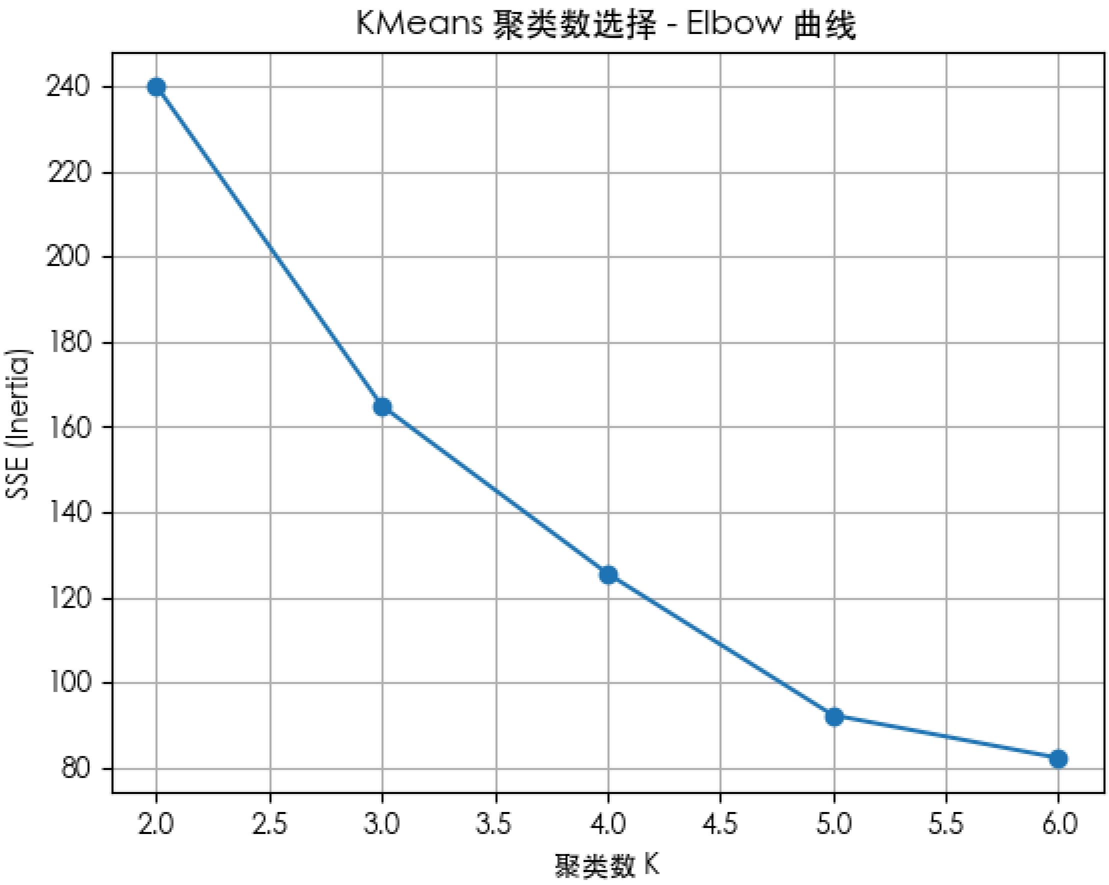
```
聚类特征列： ['ret_month', 'vol_month_sum', 'vol_20_annual_month_end']
```

Out[6]:

```
(120, 3)
```

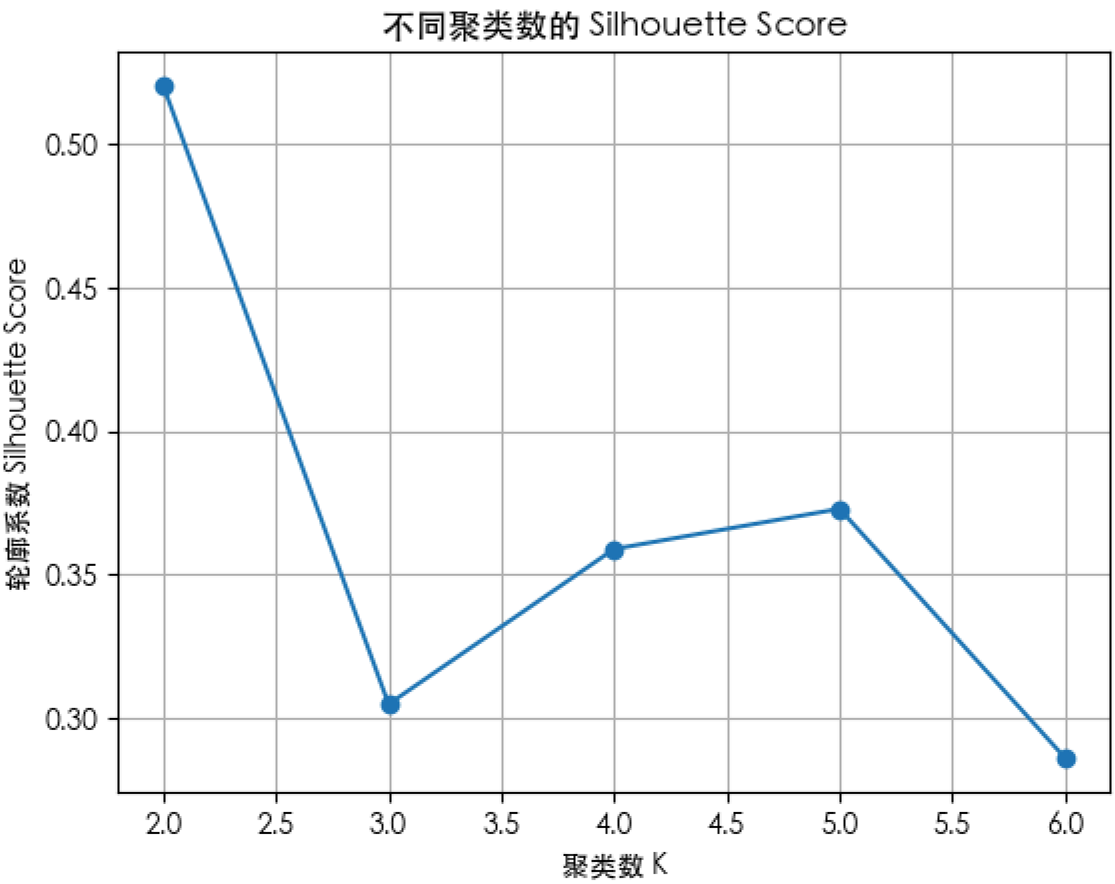
In [7]:

```python
# 5. Elbow 曲线 (SSE)
sse = []
K_range = range(2, 7)

for k in K_range:
    km = KMeans(n_clusters=k, random_state=42, n_init="auto")
    km.fit(X)
    sse.append(km.inertia_)

plt.figure()
plt.plot(list(K_range), sse, marker="o")
plt.xlabel("聚类数 K")
plt.ylabel("SSE (Inertia)")
plt.title("KMeans 聚类数选择 - Elbow 曲线")
plt.grid(True)
plt.show()
```



In [8]:

```
# 6. Silhouette Score
sil_scores = []
for k in K_range:
    km = KMeans(n_clusters=k, random_state=42, n_init="auto")
    labels = km.fit_predict(X)
    sil = silhouette_score(X, labels)
    sil_scores.append(sil)

plt.figure()
plt.plot(list(K_range), sil_scores, marker="o")
plt.xlabel("聚类数 K")
plt.ylabel("轮廓系数 Silhouette Score")
plt.title("不同聚类数的 Silhouette Score")
plt.grid(True)
plt.show()

list(zip(K_range, sil_scores))
```



Out[8]:

```
[(2, np.float64(0.5203540742408097)),
 (3, np.float64(0.304812888705872)),
 (4, np.float64(0.3588605817272719)),
 (5, np.float64(0.3728757902352798)),
 (6, np.float64(0.2861553907304862))]
```

In [9]:

```
# 7. KMeans 聚类
K = 3  # 如果你觉得 4 更好，就改成 4
kmeans = KMeans(n_clusters=K, random_state=42, n_init="auto")
cluster_labels = kmeans.fit_predict(X)

df_cluster = df_month.loc[X_raw.index].copy()
df_cluster["cluster"] = cluster_labels

df_cluster.head()
```

Out[9]:

	close_month_end	ret_month	vol_month_sum	vol_month_chg	vol_20_annual_
month					
2015-01-01	2405.38	-0.068249	240759567500	-0.372974	
2015-02-01	2474.59	0.028773	113685256600	-0.527806	
2015-03-01	2754.66	0.113178	257993309000	1.269365	
2015-04-01	3250.49	0.179997	368238254400	0.427317	
2015-05-01	3111.33	-0.042812	259537919800	-0.295190	

In [10]:

```
# 8. 不同 cluster 的特征均值（很重要）
cluster_profile = df_cluster.groupby("cluster")[feature_cols].mean()
cluster_profile
```

Out[10]:

	ret_month	vol_month_sum	vol_20_annual_month_end
cluster			
0	0.052686	7.565731e+10	0.175145
1	-0.030444	6.459630e+10	0.178653
2	-0.019967	2.975105e+11	0.453852

In [11]:

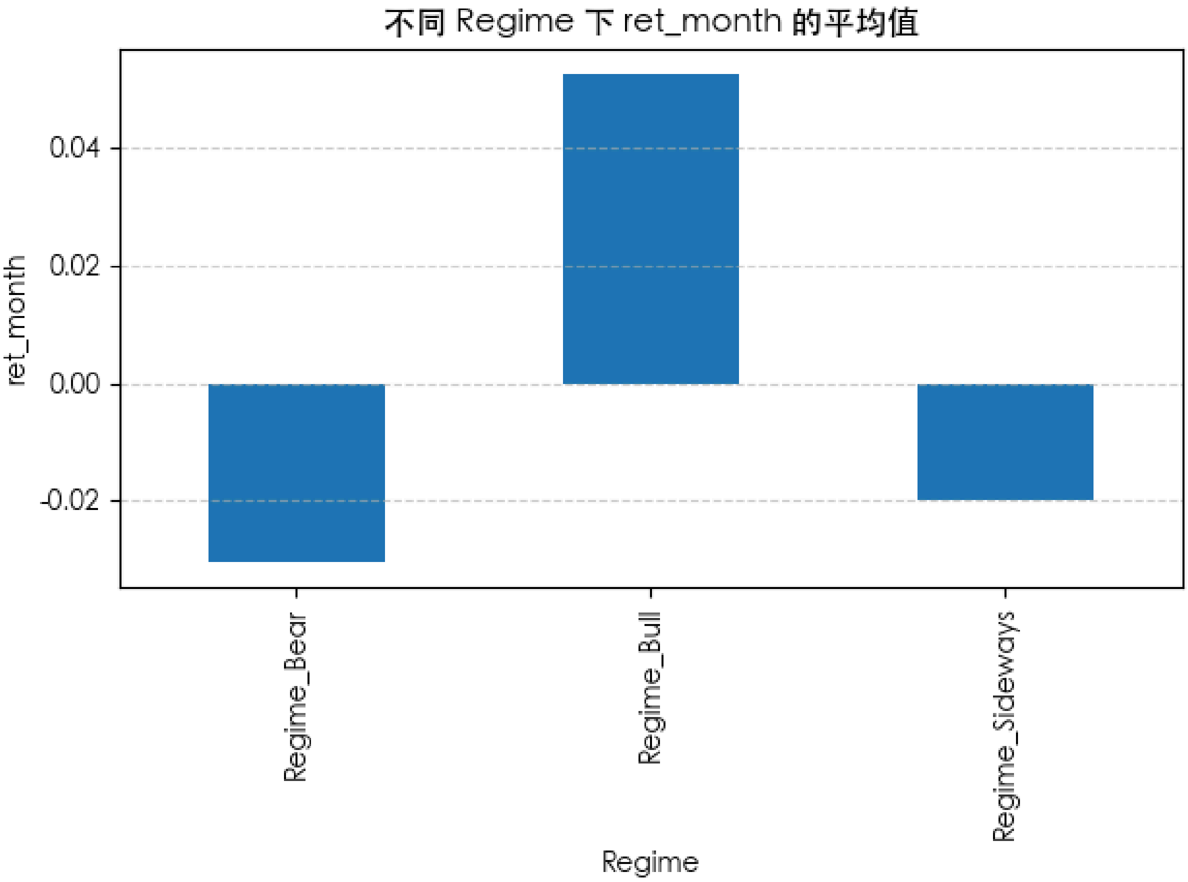
```
# 9. 根据 cluster_profile 的结果手动映射 Regime 名称
cluster_to_regime = {
    0: "Regime_Bull",       # 比如 0 类：上涨+低波动
    1: "Regime_Bear",       # 比如 1 类：下跌+高波动
    2: "Regime_Sideways",   # 比如 2 类：震荡
}

df_cluster["regime"] = df_cluster["cluster"].map(cluster_to_regime)

df_cluster[["cluster", "regime"]].head()
```

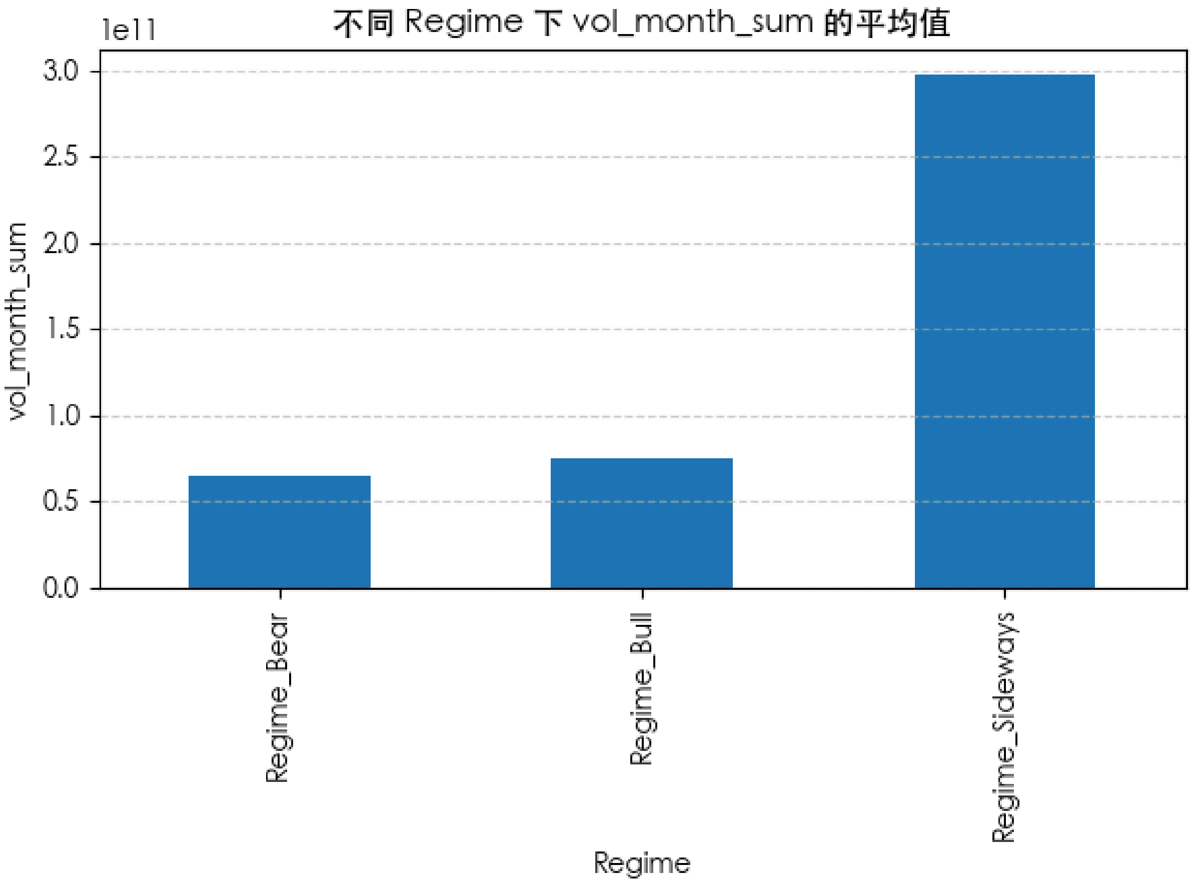
Out[11]:

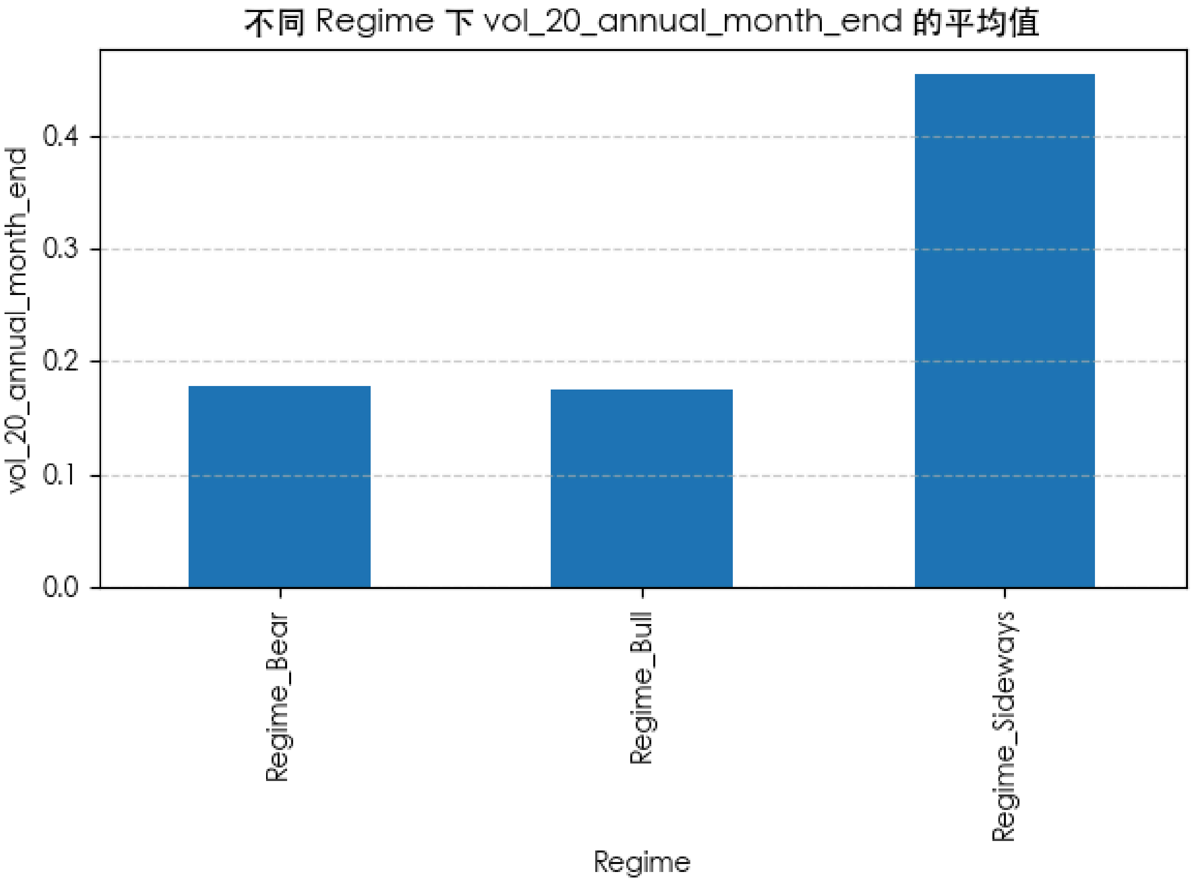
	cluster	regime
month		
2015-01-01	2	Regime_Sideways
2015-02-01	0	Regime_Bull
2015-03-01	2	Regime_Sideways
2015-04-01	2	Regime_Sideways
2015-05-01	2	Regime_Sideways

In [12]:

```
# 10. Regime 画像（每个特征一个柱状图）
regime_profile = df_cluster.groupby("regime")[feature_cols].mean()

for col in feature_cols:
    plt.figure()
    regime_profile[col].plot(kind="bar")
    plt.title(f"不同 Regime 下 {col} 的平均值")
    plt.xlabel("Regime")
    plt.ylabel(col)
    plt.grid(axis="y", linestyle="--", alpha=0.6)
    plt.tight_layout()
    plt.show()
```

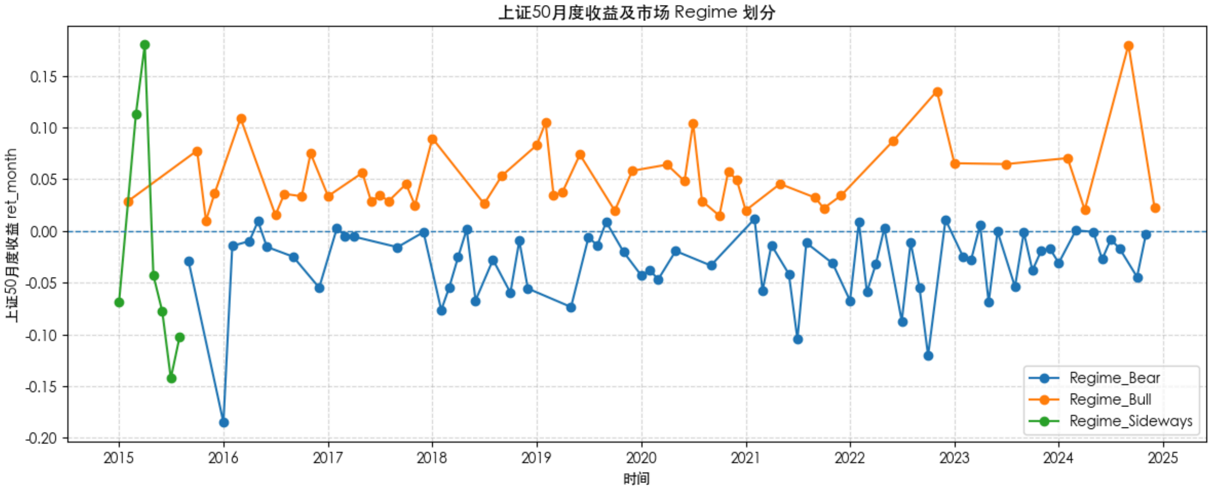



不同 Regime 下 vol_month_sum 的平均值
1e11
vol_month_sum
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Regime_Bear
Regime_Bull
Regime_Sideways
Regime


不同 Regime 下 vol_20_annual_month_end 的平均值
vol_20_annual_month_end
0.4
0.3
0.2
0.1
0.0
Regime_Bear
Regime_Bull
Regime_Sideways
Regime

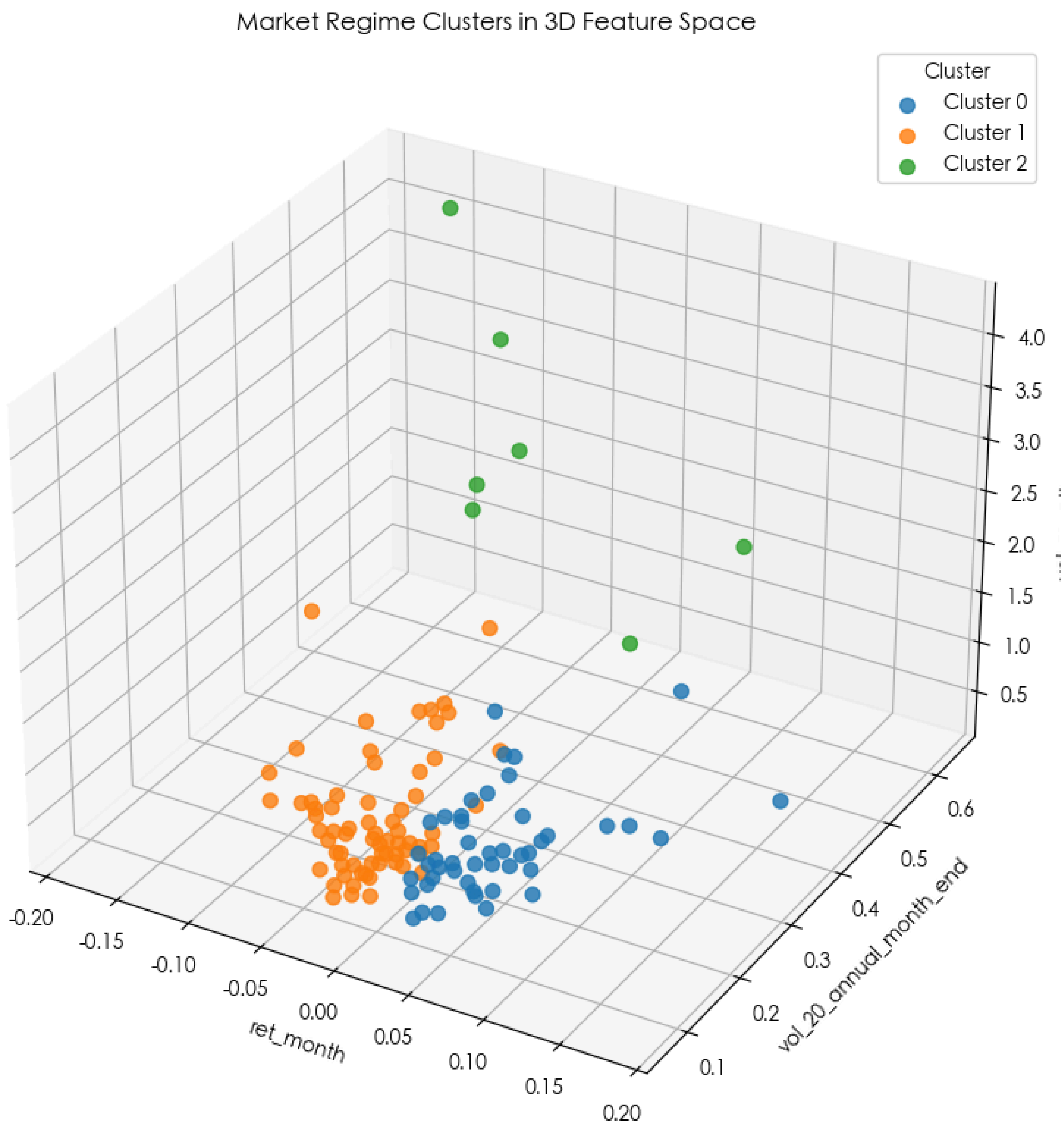
In [13]:

```
# 11. 在时间轴上按 Regime 着色显示月度收益
plt.figure(figsize=(12, 5))

for regime_name, sub_df in df_cluster.groupby("regime"):
    plt.plot(sub_df.index, sub_df["ret_month"], marker="o", linestyle="-", l

plt.axhline(0, linestyle="--", linewidth=1)
plt.xlabel("时间")
plt.ylabel("上证50月度收益 ret_month")
plt.title("上证50月度收益及市场 Regime 划分")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)
plt.tight_layout()
plt.show()
```

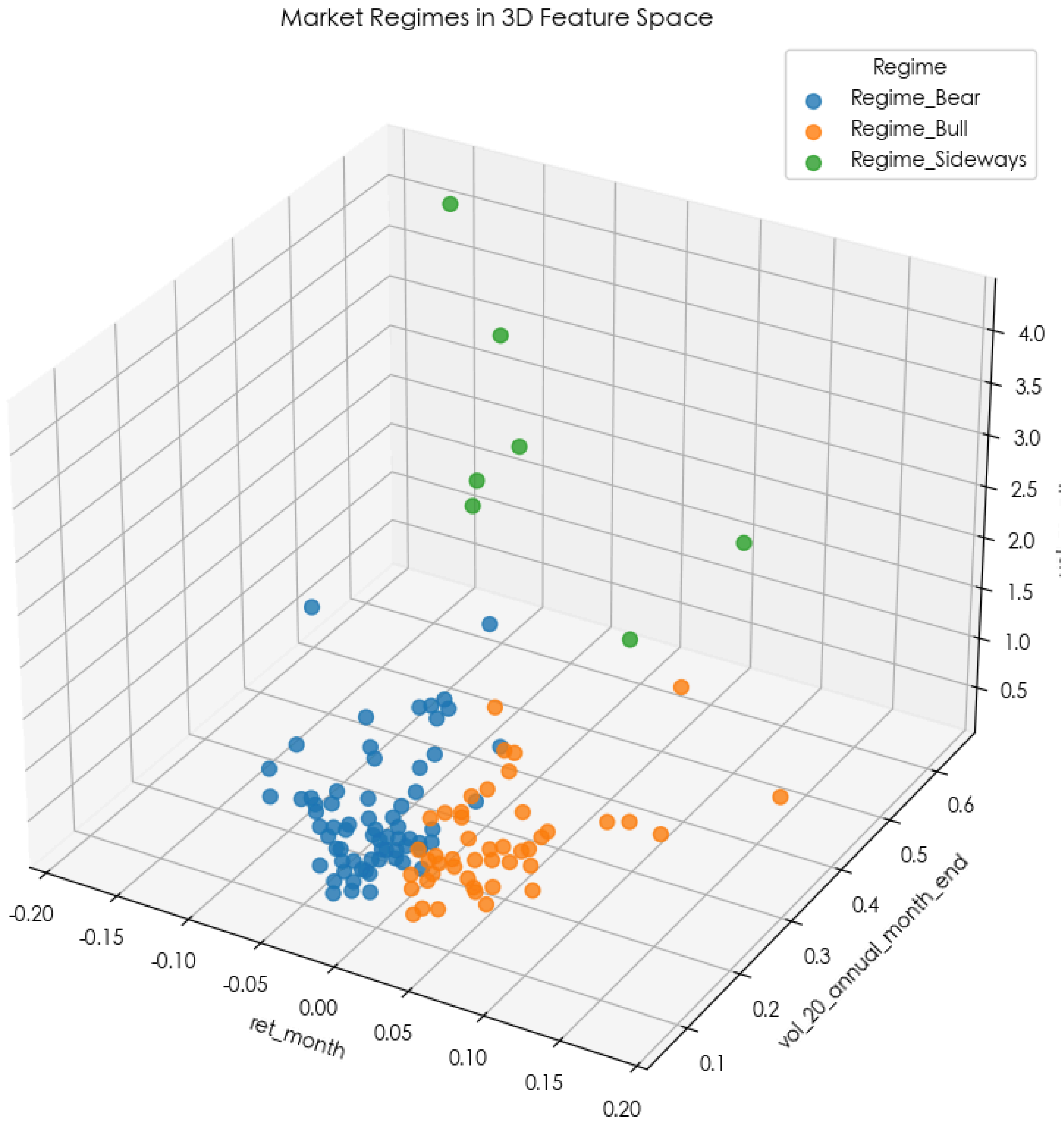


In [14]:

```
from mpl_toolkits.mplot3d import Axes3D  # 仅为激活 3D 支持
import matplotlib.pyplot as plt

# 选择要画的三个维度
x_col = "ret_month"
y_col = "vol_20_annual_month_end"
z_col = "vol_month_sum"

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection="3d")

for c, sub in df_cluster.groupby("cluster"):
    ax.scatter(
        sub[x_col],
        sub[y_col],
        sub[z_col],
        s=50,
        label=f"Cluster {c}",
        alpha=0.8,
    )

ax.set_xlabel(x_col)
ax.set_ylabel(y_col)
ax.set_zlabel(z_col)
ax.set_title("Market Regime Clusters in 3D Feature Space")
ax.legend(title="Cluster")
plt.tight_layout()
plt.show()
```



In [15]:
```
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection="3d")

for r, sub in df_cluster.groupby("regime"):
    ax.scatter(
        sub[x_col],
        sub[y_col],
        sub[z_col],
        s=50,
        label=r,
        alpha=0.8,
    )

ax.set_xlabel(x_col)
ax.set_ylabel(y_col)
ax.set_zlabel(z_col)
ax.set_title("Market Regimes in 3D Feature Space")
ax.legend(title="Regime")
plt.tight_layout()
plt.show()
```



In [16]:
```
df_cluster.head()
```

Out[16]:

	close_month_end	ret_month	vol_month_sum	vol_month_chg	vol_20_annual_
month					
2015-01-01	2405.38	-0.068249	240759567500	-0.372974	
2015-02-01	2474.59	0.028773	113685256600	-0.527806	
2015-03-01	2754.66	0.113178	257993309000	1.269365	
2015-04-01	3250.49	0.179997	368238254400	0.427317	
2015-05-01	3111.33	-0.042812	259537919800	-0.295190	

In [17]:

```python
df_cluster.to_csv("data/cluster_info")
```

In [18]:

```python
import pandas as pd

# 读取两个文件
df_factor = pd.read_csv("data/factor_longshort.csv")
df_cluster = pd.read_csv("data/cluster_info")

# 1. 处理日期格式
df_factor["日期"] = pd.to_datetime(df_factor["日期"])
df_cluster["month"] = pd.to_datetime(df_cluster["month"])

# 2. 为了 merge 统一字段名，把 df_factor 的“日期”重命名成 month
df_factor = df_factor.rename(columns={"日期": "month"})

# 3. 合并因子收益 + regime 信息
df_merged = pd.merge(
    df_factor,
    df_cluster[["month", "regime", "cluster"]],
    on="month",
    how="left"
)

# 4. 查看结果
df_merged.head(), df_merged.tail(), df_merged["regime"].value_counts()
```

Out[18]:

```
(       month  MOM20  MOM120   RSI     PB    PE   DIV   ROE  PROFIT_GR   VO
L  \
 0 2015-01-01   0.00    0.00  0.00   0.00  0.00  0.00  0.00       0.00  0.0
0
 1 2015-02-01  -4.22   -4.22 -7.76  -4.65 -7.35 -4.72 -8.90       1.88  3.0
5
 2 2015-03-01   1.84    1.84  3.44  -2.68 -4.02 -6.98 -0.27       9.71 -5.8
5
 3 2015-04-01  -7.83   -7.83 -3.86   2.96 -5.44 -2.13 -9.53      -7.40  8.3
5
 4 2015-05-01  -9.61   -9.61 -2.20 -10.82 -8.75 -8.05 -1.88      -6.20 -1.8
9

    BETA           regime  cluster
 0  0.00  Regime_Sideways        2
 1  2.06      Regime_Bull        0
 2 -0.30  Regime_Sideways        2
 3  1.45  Regime_Sideways        2
 4  8.29  Regime_Sideways        2  ,
          month  MOM20  MOM120   RSI     PB     PE    DIV    ROE  PROFIT_GR
\
 115 2024-08-01  -2.42   -2.42 -0.85  -0.32   2.82   7.71   2.65      -3.12
 116 2024-09-01  -5.67   -5.67  6.73 -10.52 -13.61  -8.65   0.66       0.34
 117 2024-10-01   4.76    4.76  1.72  -1.96 -13.81 -12.00 -17.04     -14.21
 118 2024-11-01   1.08    1.08 -0.73   1.19  -0.55  -2.54  -4.16      -3.70
 119 2024-12-01   2.68    2.68 -2.88   4.91   8.07   6.38  -0.74       5.75

        VOL   BETA       regime  cluster
 115   2.52   4.74  Regime_Bear        1
 116 -15.37 -23.44  Regime_Bull        0
 117  -9.54 -10.56  Regime_Bear        1
 118  -1.82  -2.99  Regime_Bear        1
 119   8.17   8.87  Regime_Bull        0  ,
 regime
 Regime_Bear        67
 Regime_Bull        46
 Regime_Sideways     7
 Name: count, dtype: int64)
```

In [19]:

```
df_merged.head()
```

Out[19]:

	month	MOM20	MOM120	RSI	PB	PE	DIV	ROE	PROFIT_GR	VOL	BI
0	2015-01-01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	(
1	2015-02-01	-4.22	-4.22	-7.76	-4.65	-7.35	-4.72	-8.90	1.88	3.05	2
2	2015-03-01	1.84	1.84	3.44	-2.68	-4.02	-6.98	-0.27	9.71	-5.85	-(
3	2015-04-01	-7.83	-7.83	-3.86	2.96	-5.44	-2.13	-9.53	-7.40	8.35	1
4	2015-05-01	-9.61	-9.61	-2.20	-10.82	-8.75	-8.05	-1.88	-6.20	-1.89	8

```
In [20]: df_merged.to_csv("data/final_factor_longshort.csv")
```

```
In [21]: !jupyter nbconvert --to webpdf --allow-chromium-download "cluster.ipynb"
```

```
[NbConvertApp] Converting notebook cluster.ipynb to webpdf
[NbConvertApp] WARNING | Alternative text is missing on 8 image(s).
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 811695 bytes to cluster.pdf
```

```
In [ ]:
```