

Python金融应用实验 —— 任务二至六

实验内容：

1. 数据导入与收益率计算
2. 描述性统计与可视化
3. CAPM模型估计
4. 模型诊断与检验
5. Fama-French三因子模型构建与对比

```
In [1]: import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
matplotlib.rcParams['axes.unicode_minus'] = False

from scipy import stats
from scipy.stats import gaussian_kde
import statsmodels.api as sm
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.stattools import durbin_watson

print("所有库导入成功！")
```

所有库导入成功！

任务2：数据导入与处理

- 从CSV文件导入数据
- 计算日对数收益率： $r_t = \ln(P_t/P_{t-1})$
- 计算超额收益率
- 绘制散点图观察线性关系
- 处理缺失值与异常值

```
In [2]: # 2.0 导入数据
stock_df = pd.read_csv('stock_daily.csv', parse_dates=['trade_date'])
index_df = pd.read_csv('index_daily.csv', parse_dates=['trade_date'])
rf_df = pd.read_csv('risk_free_rate.csv', parse_dates=['trade_date'])
ff_df = pd.read_csv('fama_french_factors.csv', parse_dates=['trade_date'])

print("数据导入成功！")
print(f"个股数据: {stock_df.shape}, 指数数据: {index_df.shape}")
print(f"无风险利率: {rf_df.shape}, 三因子: {ff_df.shape}")
```

数据导入成功!

个股数据: (2908, 12), 指数数据: (1454, 12)

无风险利率: (12, 3), 三因子: (726, 4)

```
In [3]: # 2.1 计算日对数收益率
# 个股收益率
stock_pivot = stock_df.pivot_table(index='trade_date', columns='ts_code', values='c
stock_pivot = stock_pivot.sort_index()
stock_returns = np.log(stock_pivot / stock_pivot.shift(1))
stock_returns.columns = [f'{c}_ret' for c in stock_returns.columns]

# 指数收益率
index_pivot = index_df.pivot_table(index='trade_date', columns='ts_code', values='c
index_pivot = index_pivot.sort_index()
index_returns = np.log(index_pivot / index_pivot.shift(1))
index_returns.columns = [f'{c}_ret' for c in index_returns.columns]

# 合并收益率数据框
returns_df = pd.concat([stock_returns, index_returns], axis=1).dropna()

# 定义常量
stock_codes = ['300750.SZ', '600276.SH', '600519.SH', '601012.SH']
index_codes = ['000001.SH', '000300.SH']
stock_name_map = {'300750.SZ': '宁德时代', '600276.SH': '恒瑞医药',
                  '600519.SH': '贵州茅台', '601012.SH': '隆基绿能'}
index_name_map = {'000001.SH': '上证综指', '000300.SH': '沪深300'}

print("日对数收益率数据框: ")
print(returns_df.head(10))
print(f"\n数据框形状: {returns_df.shape}")
```

日对数收益率数据框:

	300750.SZ_ret	600276.SH_ret	600519.SH_ret	601012.SH_ret \
trade_date				
2023-01-04	-0.014555	0.008548	-0.002894	-0.033385
2023-01-05	0.059397	0.019919	0.043109	0.010279
2023-01-06	0.022205	-0.006087	0.001537	0.038218
2023-01-09	-0.000144	0.006087	0.020539	-0.009418
2023-01-10	0.036299	-0.004308	0.007171	0.007072
2023-01-11	-0.009630	-0.023899	-0.005136	-0.004001
2023-01-12	0.013096	-0.017314	-0.005953	0.023081
2023-01-13	0.013746	0.038416	0.028489	0.000230
2023-01-16	0.016093	0.093758	0.013632	0.013730
2023-01-17	-0.002286	-0.022273	-0.002565	0.006343

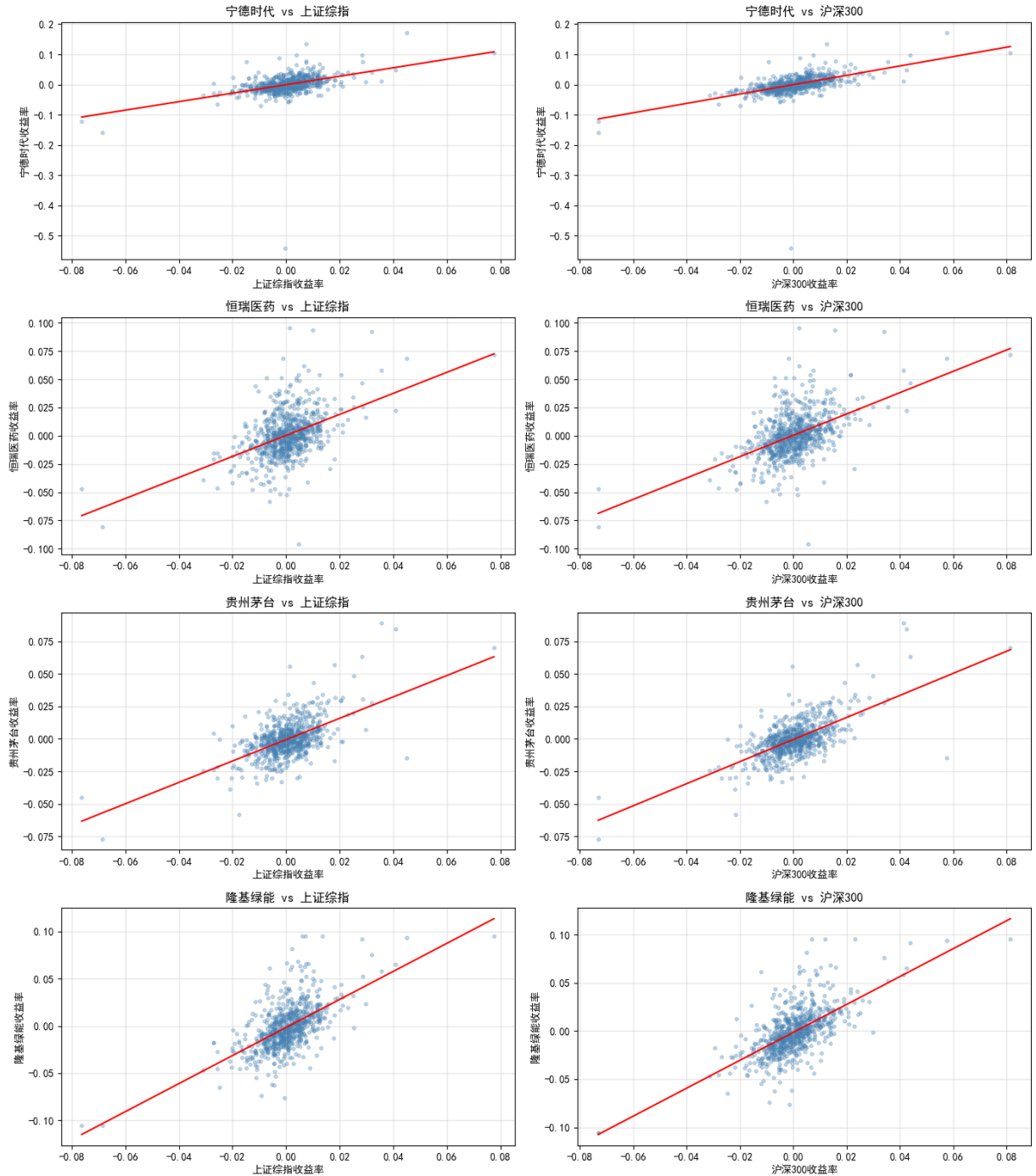
	000001.SH_ret	000300.SH_ret
trade_date		
2023-01-04	0.002245	0.001298
2023-01-05	0.010098	0.019241
2023-01-06	0.000767	0.003097
2023-01-09	0.005825	0.008064
2023-01-10	-0.002073	0.001084
2023-01-11	-0.002423	-0.001854
2023-01-12	0.000510	0.001953
2023-01-13	0.010019	0.013966
2023-01-16	0.010053	0.015486
2023-01-17	-0.001037	-0.000175

数据框形状: (726, 6)

```
In [4]: # 2.1(续) 绘制每支股票与每个指数的收益率散点图
fig, axes = plt.subplots(len(stock_codes), len(index_codes), figsize=(14, 16))
for i, sc in enumerate(stock_codes):
    for j, ic in enumerate(index_codes):
        ax = axes[i][j]
        x = returns_df[f'{ic}_ret']
        y = returns_df[f'{sc}_ret']
        ax.scatter(x, y, alpha=0.3, s=10, color='steelblue')
        # 拟合回归线
        z = np.polyfit(x, y, 1)
        p = np.poly1d(z)
        x_line = np.linspace(x.min(), x.max(), 100)
        ax.plot(x_line, p(x_line), 'r-', linewidth=1.5)
        ax.set_xlabel(f'{index_name_map[ic]}收益率')
        ax.set_ylabel(f'{stock_name_map[sc]}收益率')
        ax.set_title(f'{stock_name_map[sc]} vs {index_name_map[ic]}')
        ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.suptitle('个股收益率与指数收益率散点图', fontsize=14, y=1.01)
plt.savefig('task2_scatter_returns.png', dpi=150, bbox_inches='tight')
plt.show()
```

个股收益率与指数收益率散点图



```
In [5]: # 2.2 超额收益率计算
# 合并无风险利率
returns_df = returns_df.merge(rf_df[['trade_date', 'rf_daily']], left_index=True, right_index=True)
returns_df = returns_df.set_index('trade_date')

# 用前值填充缺失的无风险利率
returns_df['rf_daily'] = returns_df['rf_daily'].fillna(method='ffill').fillna(method='bfill')

# 个股超额收益率 = 个股收益率 - 日度无风险利率
for sc in stock_codes:
    returns_df[f'{sc}_excess'] = returns_df[f'{sc}_ret'] - returns_df['rf_daily']

# 市场超额收益率 = 市场收益率 - 日度无风险利率
```

```

for ic in index_codes:
    returns_df[f'{ic}_excess'] = returns_df[f'{ic}_ret'] - returns_df['rf_daily']

print("超额收益率计算完成：")
excess_cols = [c for c in returns_df.columns if 'excess' in c]
print(returns_df[excess_cols].head(10))

```

超额收益率计算完成：

	300750.SZ_excess	600276.SH_excess	600519.SH_excess \
trade_date			
2023-01-04	-0.014640	0.008463	-0.002980
2023-01-05	0.059311	0.019833	0.043024
2023-01-06	0.022120	-0.006172	0.001451
2023-01-09	-0.000230	0.006001	0.020453
2023-01-10	0.036214	-0.004393	0.007085
2023-01-11	-0.009715	-0.023985	-0.005221
2023-01-12	0.013010	-0.017400	-0.006038
2023-01-13	0.013660	0.038331	0.028403
2023-01-16	0.016007	0.093672	0.013547
2023-01-17	-0.002372	-0.022358	-0.002650

	601012.SH_excess	000001.SH_excess	000300.SH_excess
trade_date			
2023-01-04	-0.033470	0.002160	0.001212
2023-01-05	0.010194	0.010012	0.019156
2023-01-06	0.038133	0.000681	0.003012
2023-01-09	-0.009504	0.005740	0.007978
2023-01-10	0.006987	-0.002159	0.000999
2023-01-11	-0.004087	-0.002508	-0.001940
2023-01-12	0.022995	0.000425	0.001867
2023-01-13	0.000145	0.009934	0.013881
2023-01-16	0.013645	0.009968	0.015401
2023-01-17	0.006258	-0.001123	-0.000260

```

In [6]: # 2.3 处理缺失值，检查异常值
print(f"数据清洗前形状: {returns_df.shape}")
print(f"缺失值总数: {returns_df.isnull().sum().sum()}")

# 删除缺失值
returns_df = returns_df.dropna()

print(f"数据清洗后形状: {returns_df.shape}")

# 检查异常值 (超过±5个标准差的数据)
ret_cols = [c for c in returns_df.columns if '_ret' in c or '_excess' in c]
for col in ret_cols:
    mean_val = returns_df[col].mean()
    std_val = returns_df[col].std()
    outliers = returns_df[(returns_df[col] > mean_val + 5*std_val) | (returns_df[col]
    if len(outliers) > 0:
        print(f" {col}: {len(outliers)} 个异常值")

print("\n清洗后数据描述性统计:")
returns_df.describe()

```

数据清洗前形状: (726, 13)
缺失值总数: 0
数据清洗后形状: (726, 13)
300750.SZ_ret: 3 个异常值
600519.SH_ret: 4 个异常值
000001.SH_ret: 3 个异常值
000300.SH_ret: 4 个异常值
300750.SZ_excess: 3 个异常值
600519.SH_excess: 4 个异常值
000001.SH_excess: 3 个异常值
000300.SH_excess: 4 个异常值

清洗后数据描述性统计:

Out[6]:

	300750.SZ_ret	600276.SH_ret	600519.SH_ret	601012.SH_ret	000001.SH_ret
count	726.000000	726.000000	726.000000	726.000000	726.000000
mean	-0.000081	0.000603	-0.000314	-0.001153	0.000333
std	0.031373	0.019895	0.013898	0.024191	0.009746
min	-0.541703	-0.095555	-0.077099	-0.105643	-0.076275
25%	-0.012022	-0.010632	-0.007212	-0.015800	-0.004461
50%	-0.001963	-0.000378	-0.000952	-0.002962	0.000312
75%	0.011556	0.009917	0.005162	0.011138	0.005153
max	0.171451	0.095310	0.088856	0.095538	0.077551

2.3 数据清洗结论

- 清洗前后数据形状均为 (726, 13)，无缺失值需要处理。
- 异常值检查（超过±5个标准差）：
 - 宁德时代：3个异常值（收益率波动较大，受极端行情影响）
 - 恒瑞医药：无异常值
 - 贵州茅台：4个异常值
 - 隆基绿能：无异常值
 - 上证综指：3个异常值
 - 沪深300：4个异常值
- 异常值数量较少且属于真实市场行情（如涨跌停），予以保留，不做剔除处理。

任务3：描述性统计与可视化

- 计算均值、标准差、偏度、峰度
- 收益率分布直方图（带密度曲线）
- 时间序列图（双轴图）
- 超额收益率散点图
- 相关性分析

```
In [7]: # 3.1 描述性统计：均值、标准差、偏度、峰度
all_ret_cols = [f'{sc}_ret' for sc in stock_codes] + [f'{ic}_ret' for ic in index_c

desc_stats = pd.DataFrame(index=all_ret_cols, columns=['均值', '标准差', '偏度', '峰度'])
for col in all_ret_cols:
    desc_stats.loc[col, '均值'] = returns_df[col].mean()
    desc_stats.loc[col, '标准差'] = returns_df[col].std()
    desc_stats.loc[col, '偏度'] = returns_df[col].skew()
    desc_stats.loc[col, '峰度'] = returns_df[col].kurtosis()

# 美化列索引名
name_map_full = {**{f'{k}_ret': v for k, v in stock_name_map.items()},
                  **{f'{k}_ret': v for k, v in index_name_map.items()}}
desc_stats.index = [name_map_full.get(c, c) for c in desc_stats.index]
desc_stats = desc_stats.astype(float).round(6)
desc_stats
```

```
Out[7]:
```

	均值	标准差	偏度	峰度
宁德时代	-0.000081	0.031373	-6.752291	124.747536
恒瑞医药	0.000603	0.019895	0.489571	3.338585
贵州茅台	-0.000314	0.013898	0.938760	7.540893
隆基绿能	-0.001153	0.024191	0.552081	2.798688
上证综指	0.000333	0.009746	-0.191049	14.645285
沪深300	0.000241	0.010726	0.249779	11.771380

3.1 描述性统计分析结论

股票/指数	均值	标准差	偏度	峰度
宁德时代	-0.000081	0.031373	-6.75	124.75
恒瑞医药	0.000603	0.019895	0.49	3.34
贵州茅台	-0.000314	0.013898	0.94	7.54
隆基绿能	-0.001153	0.024191	0.55	2.80
上证综指	0.000333	0.009746	-0.19	14.65
沪深300	0.000241	0.010726	0.25	11.77

- 所有资产的**峰度均大于0**，表明收益率分布具有**尖峰厚尾**特征（相对正态分布）。
- 宁德时代的峰度高达124.75，偏度为-6.75，说明存在极端负收益事件（如大幅下跌）。
- 偏度不为0表明收益率分布**不完全对称**。
- 波动率方面：宁德时代 > 隆基绿能 > 恒瑞医药 > 贵州茅台 > 沪深300 > 上证综指。

```
In [8]: # 3.2 收益率分布图：直方图 + 密度曲线
fig, axes = plt.subplots(3, 2, figsize=(14, 15))
axes = axes.flatten()
```

```

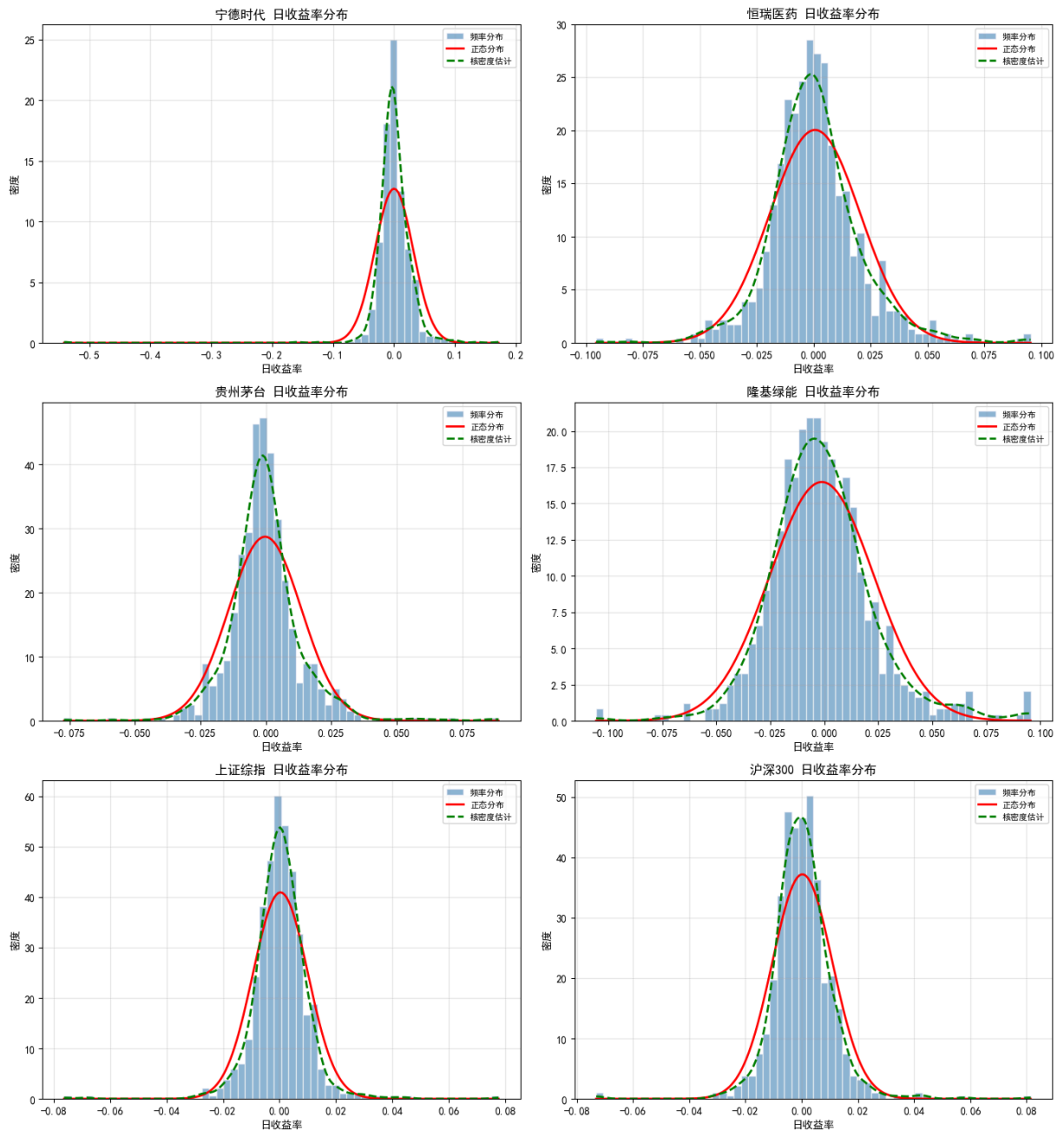
plot_cols = [f'{sc}_ret' for sc in stock_codes] + [f'{ic}_ret' for ic in index_code]
plot_names = [stock_name_map.get(sc, sc) for sc in stock_codes] + [index_name_map.g

for idx, (col, name) in enumerate(zip(plot_cols, plot_names)):
    ax = axes[idx]
    data = returns_df[col].dropna()
    ax.hist(data, bins=60, density=True, alpha=0.6, color='steelblue', edgecolor='w
    x_range = np.linspace(data.min(), data.max(), 200)
    ax.plot(x_range, stats.norm.pdf(x_range, data.mean(), data.std()), 'r-', linewidthi
    kde = gaussian_kde(data)
    ax.plot(x_range, kde(x_range), 'g--', linewidth=2, label='核密度估计')
    ax.set_title(f'{name} 日收益率分布', fontsize=12)
    ax.set_xlabel('日收益率')
    ax.set_ylabel('密度')
    ax.legend(fontsize=8)
    ax.grid(True, alpha=0.3)

for idx in range(len(plot_cols), len(axes)):
    axes[idx].set_visible(False)

plt.tight_layout()
plt.savefig('task3_return_distribution.png', dpi=150, bbox_inches='tight')
plt.show()

```

```
In [9]: # 3.3 时间序列图：个股收盘价与股指收盘点位（双轴图）
fig, axes = plt.subplots(len(stock_codes), 1, figsize=(14, 16))

for i, sc in enumerate(stock_codes):
    ax1 = axes[i]
    stock_close = stock_pivot[sc].dropna()
    ax1.plot(stock_close.index, stock_close.values, color='steelblue', linewidth=1.0)
    ax1.set_ylabel(f'{stock_name_map[sc]} 收盘价(元)', color='steelblue', fontsize=10)
    ax1.tick_params(axis='y', labelcolor='steelblue')

    ax2 = ax1.twinx()
    idx_close = index_pivot['000300.SH'].dropna()
    ax2.plot(idx_close.index, idx_close.values, color='orangered', linewidth=1.0, alpha=0.5)
    ax2.set_ylabel('沪深300 收盘点位', color='orangered', fontsize=10)
    ax2.tick_params(axis='y', labelcolor='orangered')
```

```
ax1.set_title(f'{stock_name_map[sc]} 与 沪深300 收盘价时间序列（双轴图）', fontsize=12)
ax1.grid(True, alpha=0.3)
```

```
lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper left', fontsize=9)
```

```
plt.tight_layout()
plt.savefig('task3_timeseries.png', dpi=150, bbox_inches='tight')
plt.show()
```



```
In [10]: # 3.4 散点图: 个股超额收益率与市场超额收益率
fig, axes = plt.subplots(2, 2, figsize=(14, 12))
axes = axes.flatten()
```

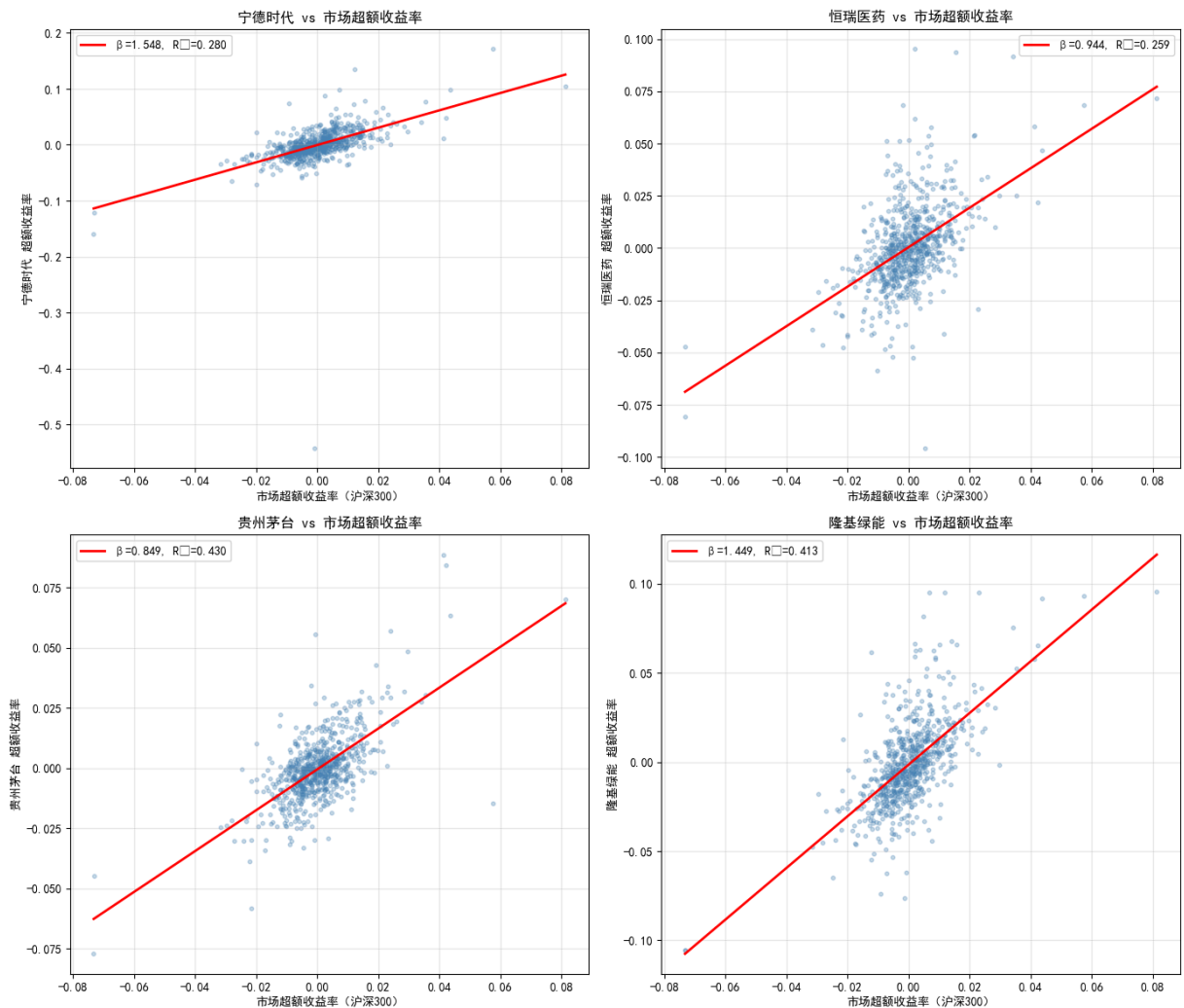
```

market_excess = '000300.SH_excess' # 使用沪深300作为市场基准

for i, sc in enumerate(stock_codes):
    ax = axes[i]
    x = returns_df[market_excess]
    y = returns_df[f'{sc}_excess']
    ax.scatter(x, y, alpha=0.3, s=10, color='steelblue')
    slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)
    x_line = np.linspace(x.min(), x.max(), 100)
    ax.plot(x_line, intercept + slope * x_line, 'r-', linewidth=2,
            label=f' $\beta$ ={slope:.3f},  $R^2$ ={r_value**2:.3f}')
    ax.set_xlabel('市场超额收益率 (沪深300)')
    ax.set_ylabel(f'{stock_name_map[sc]} 超额收益率')
    ax.set_title(f'{stock_name_map[sc]} vs 市场超额收益率')
    ax.legend(fontsize=10)
    ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('task3_excess_return_scatter.png', dpi=150, bbox_inches='tight')
plt.show()

```



```

In [11]: # 3.5 相关性分析
ret_for_corr = returns_df[[f'{sc}_ret' for sc in stock_codes] + [f'{ic}_ret' for ic

```

```

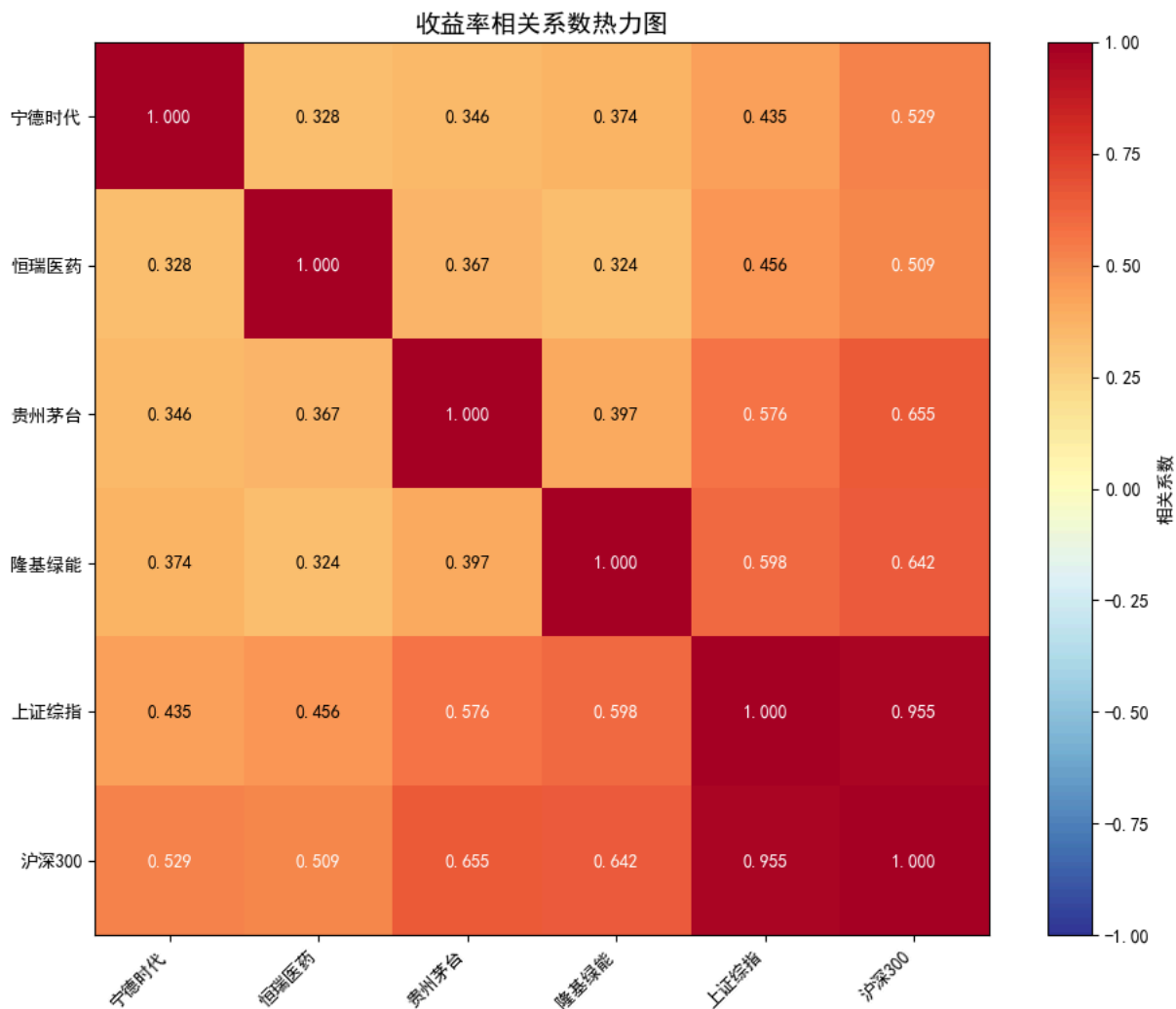
ret_for_corr.columns = [stock_name_map.get(c.replace('_ret', '')), index_name_map.get(c)]
corr_matrix = ret_for_corr.corr()

print(corr_matrix.round(4).to_string())

# 热力图
fig, ax = plt.subplots(figsize=(10, 8))
im = ax.imshow(corr_matrix.values, cmap='RdYlBu_r', vmin=-1, vmax=1)
ax.set_xticks(range(len(corr_matrix.columns)))
ax.set_yticks(range(len(corr_matrix.columns)))
ax.set_xticklabels(corr_matrix.columns, rotation=45, ha='right')
ax.set_yticklabels(corr_matrix.columns)
for i in range(len(corr_matrix)):
    for j in range(len(corr_matrix)):
        ax.text(j, i, f'{corr_matrix.iloc[i, j]:.3f}', ha='center', va='center', fo
                color='white' if abs(corr_matrix.iloc[i, j]) > 0.5 else 'black')
plt.colorbar(im, ax=ax, label='相关系数')
plt.title('收益率相关系数热力图', fontsize=14)
plt.tight_layout()
plt.savefig('task3_correlation_heatmap.png', dpi=150, bbox_inches='tight')
plt.show()

```

	宁德时代	恒瑞医药	贵州茅台	隆基绿能	上证综指	沪深300
宁德时代	1.0000	0.3285	0.3461	0.3737	0.4351	0.5293
恒瑞医药	0.3285	1.0000	0.3666	0.3244	0.4558	0.5090
贵州茅台	0.3461	0.3666	1.0000	0.3965	0.5765	0.6554
隆基绿能	0.3737	0.3244	0.3965	1.0000	0.5985	0.6423
上证综指	0.4351	0.4558	0.5765	0.5985	1.0000	0.9551
沪深300	0.5293	0.5090	0.6554	0.6423	0.9551	1.0000



3.5 相关性分析结论

- 个股收益率与市场收益率之间存在**不同程度的正相关关系**。
- **贵州茅台**与**沪深300**相关性最高（0.655），说明其走势与大盘最为同步。
- **隆基绿能**与**沪深300**相关性为0.642，同样较高。
- **宁德时代**（0.529）和**恒瑞医药**（0.509）与**沪深300**的相关性中等，说明受个股特有因素影响较大。
- 上证综指与沪深300高度相关（0.955），两者可互为市场基准。

任务4：CAPM模型估计

CAPM资产定价模型： $R_i - R_f = \alpha + \beta(R_m - R_f) + \varepsilon$

- 以个股超额收益率为因变量，市场超额收益率为自变量
- 使用 statsmodels OLS 进行回归估计
- 输出回归系数、 R^2 、t检验、p值等
- 解释Beta经济含义，判断Alpha显著性
- 计算理论预期年化收益率

```
In [12]: # 4.1 OLS回归: 个股超额收益率 ~ 市场超额收益率 (沪深300)
market_excess_col = '000300.SH_excess'
capm_results = {}

for sc in stock_codes:
    y = returns_df[f'{sc}_excess']
    X = returns_df[market_excess_col]
    X_const = sm.add_constant(X)

    model = sm.OLS(y, X_const).fit()
    capm_results[sc] = model

    print("=" * 70)
    print(f"CAPM回归结果 — {stock_name_map[sc]} ({sc})")
    print("=" * 70)
    print(model.summary())
    print()
```

=====
CAPM回归结果 — 宁德时代 (300750.SZ)
=====

OLS Regression Results

```

=====
Dep. Variable:      300750.SZ_excess      R-squared:              0.280
Model:              OLS                   Adj. R-squared:         0.279
Method:             Least Squares         F-statistic:            281.9
Date:               Fri, 10 Apr 2026       Prob (F-statistic):     1.11e-53
Time:               22:05:11              Log-Likelihood:         1603.0
No. Observations:   726                   AIC:                    -3202.
Df Residuals:       724                   BIC:                    -3193.
Df Model:           1
Covariance Type:    nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0004	0.001	-0.420	0.674	-0.002	0.002
000300.SH_excess	1.5484	0.092	16.790	0.000	1.367	1.730

```

=====
Omnibus:            1284.439      Durbin-Watson:          2.049
Prob(Omnibus):      0.000         Jarque-Bera (JB):       1649371.502
Skew:               -11.195       Prob(JB):               0.00
Kurtosis:           235.429       Cond. No.               93.3
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

=====
CAPM回归结果 — 恒瑞医药 (600276.SH)
=====

OLS Regression Results

```

=====
Dep. Variable:      600276.SH_excess      R-squared:              0.259
Model:              OLS                   Adj. R-squared:         0.258
Method:             Least Squares         F-statistic:            253.2
Date:               Fri, 10 Apr 2026       Prob (F-statistic):     4.13e-49
Time:               22:05:11              Log-Likelihood:         1923.2
No. Observations:   726                   AIC:                    -3842.
Df Residuals:       724                   BIC:                    -3833.
Df Model:           1
Covariance Type:    nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	0.0004	0.001	0.585	0.559	-0.001	0.002
000300.SH_excess	0.9441	0.059	15.912	0.000	0.828	1.061

```

=====
Omnibus:            103.900      Durbin-Watson:          1.957
Prob(Omnibus):      0.000         Jarque-Bera (JB):       578.474
Skew:               0.492         Prob(JB):               2.43e-126
Kurtosis:           7.261         Cond. No.               93.3
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

=====

CAPM回归结果 — 贵州茅台 (600519.SH)

=====

OLS Regression Results

=====

Dep. Variable:	600519.SH_excess	R-squared:	0.430
Model:	OLS	Adj. R-squared:	0.429
Method:	Least Squares	F-statistic:	545.1
Date:	Fri, 10 Apr 2026	Prob (F-statistic):	2.56e-90
Time:	22:05:11	Log-Likelihood:	2278.5
No. Observations:	726	AIC:	-4553.
Df Residuals:	724	BIC:	-4544.
Df Model:	1		
Covariance Type:	nonrobust		

=====

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0005	0.000	-1.357	0.175	-0.001	0.000
000300.SH_excess	0.8492	0.036	23.348	0.000	0.778	0.921

=====

Omnibus:	96.048	Durbin-Watson:	1.937
Prob(Omnibus):	0.000	Jarque-Bera (JB):	627.950
Skew:	0.367	Prob(JB):	4.39e-137
Kurtosis:	7.497	Cond. No.	93.3

=====

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

=====

CAPM回归结果 — 隆基绿能 (601012.SH)

=====

OLS Regression Results

=====

Dep. Variable:	601012.SH_excess	R-squared:	0.413
Model:	OLS	Adj. R-squared:	0.412
Method:	Least Squares	F-statistic:	508.5
Date:	Fri, 10 Apr 2026	Prob (F-statistic):	1.06e-85
Time:	22:05:11	Log-Likelihood:	1865.4
No. Observations:	726	AIC:	-3727.
Df Residuals:	724	BIC:	-3718.
Df Model:	1		
Covariance Type:	nonrobust		

=====

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0015	0.001	-2.134	0.033	-0.003	-0.000
000300.SH_excess	1.4487	0.064	22.550	0.000	1.323	1.575

=====

Omnibus:	113.532	Durbin-Watson:	2.042
Prob(Omnibus):	0.000	Jarque-Bera (JB):	330.835

Skew: 0.771 Prob(JB): 1.45e-72
Kurtosis: 5.926 Cond. No. 93.3

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
In [13]: # 4.2 汇总CAPM回归结果
capm_summary = pd.DataFrame(columns=['Alpha(α)', 'Alpha_p值', 'Beta(β)', 'Beta_p值',
                                     'Beta_t统计量', 'R²', '残差标准差'])

for sc in stock_codes:
    model = capm_results[sc]
    capm_summary.loc[stock_name_map[sc]] = [
        model.params['const'],
        model.pvalues['const'],
        model.params[market_excess_col],
        model.pvalues[market_excess_col],
        model.tvalues[market_excess_col],
        model.rsquared,
        np.sqrt(model.mse_resid)
    ]

capm_summary.round(6)
```

Out[13]:

	Alpha(α)	Alpha_p 值	Beta(β)	Beta_p 值	Beta_t统计 量	R²	残差标准 差
宁德时代	-0.000416	0.674358	1.548443	0.0	16.790064	0.280251	0.026635
恒瑞医药	0.000372	0.558516	0.944127	0.0	15.912020	0.259102	0.017136
贵州茅台	-0.000529	0.175219	0.849161	0.0	23.348354	0.429537	0.010504
隆基绿能	-0.001470	0.033152	1.448684	0.0	22.549608	0.412569	0.018555

4.2 Beta系数经济含义与Alpha显著性分析

Beta系数解释:

股票	Beta	类型	含义
宁德时代	1.5484	进攻型 (β>1)	市场上涨1%时预期上涨约1.55%，系统性风险高于市场
恒瑞医药	0.9441	防御型 (β<1)	市场上涨1%时预期上涨约0.94%，系统性风险低于市场
贵州茅台	0.8492	防御型 (β<1)	市场上涨1%时预期上涨约0.85%，系统性风险低于市场
隆基绿能	1.4487	进攻型 (β>1)	市场上涨1%时预期上涨约1.45%，系统性风险高于市场

Alpha显著性判断:

股票	Alpha	p值	结论
宁德时代	-0.000416	0.6744	不显著 (p>0.05) , CAPM可解释其超额收益, 无显著异常收益
恒瑞医药	0.000372	0.5585	不显著 (p>0.05) , CAPM可解释其超额收益, 无显著异常收益
贵州茅台	-0.000529	0.1752	不显著 (p>0.05) , CAPM可解释其超额收益, 无显著异常收益
隆基绿能	-0.001470	0.0332	显著为负 (p<0.05) , 存在负的超额收益, 跑输了CAPM模型预期

拟合优度评估:

股票	R²	评估
宁德时代	0.2803	较低, 单因子模型解释力有限
恒瑞医药	0.2591	较低, 单因子模型解释力有限
贵州茅台	0.4295	一般, 可能需要更多因子来解释
隆基绿能	0.4126	一般, 可能需要更多因子来解释

```
In [14]: # 4.3 CAPM公式计算: 理论预期年化收益率
# E(Ri) = Rf + βi * [E(Rm) - Rf]
rf_annual = rf_df['rf_annual_pct'].mean() / 100 # 平均年化无风险利率
market_annual_ret = returns_df[f'000300.SH_ret'].mean() * 252 # 市场年化收益率
market_premium = market_annual_ret - rf_annual # 市场风险溢价

print(f"平均年化无风险利率 Rf = {rf_annual*100:.2f}%")
print(f"市场年化收益率 (沪深300) E(Rm) = {market_annual_ret*100:.2f}%")
print(f"市场风险溢价 [E(Rm)-Rf] = {market_premium*100:.2f}%")
print()

for sc in stock_codes:
    beta = capm_results[sc].params[market_excess_col]
    expected_return = rf_annual + beta * market_premium
    actual_return = returns_df[f'{sc}_ret'].mean() * 252
    print(f" [{stock_name_map[sc]}] ")
    print(f" Beta = {beta:.4f}")
    print(f" CAPM理论预期年化收益率 E(Ri) = {rf_annual*100:.2f}% + {beta:.4f} × {mar
    print(f" 实际年化收益率 = {actual_return*100:.2f}%")
    print(f" 差异 (Alpha效应) = {(actual_return - expected_return)*100:.2f}%")
    print()
```

平均年化无风险利率 $R_f = 1.50\%$
市场年化收益率（沪深300） $E(R_m) = 6.06\%$
市场风险溢价 $[E(R_m) - R_f] = 4.57\%$

【宁德时代】

Beta = 1.5484
CAPM理论预期年化收益率 $E(R_i) = 1.50\% + 1.5484 \times 4.57\% = 8.57\%$
实际年化收益率 = -2.05%
差异 (Alpha效应) = -10.62%

【恒瑞医药】

Beta = 0.9441
CAPM理论预期年化收益率 $E(R_i) = 1.50\% + 0.9441 \times 4.57\% = 5.81\%$
实际年化收益率 = 15.21%
差异 (Alpha效应) = 9.40%

【贵州茅台】

Beta = 0.8492
CAPM理论预期年化收益率 $E(R_i) = 1.50\% + 0.8492 \times 4.57\% = 5.37\%$
实际年化收益率 = -7.92%
差异 (Alpha效应) = -13.29%

【隆基绿能】

Beta = 1.4487
CAPM理论预期年化收益率 $E(R_i) = 1.50\% + 1.4487 \times 4.57\% = 8.11\%$
实际年化收益率 = -29.05%
差异 (Alpha效应) = -37.16%

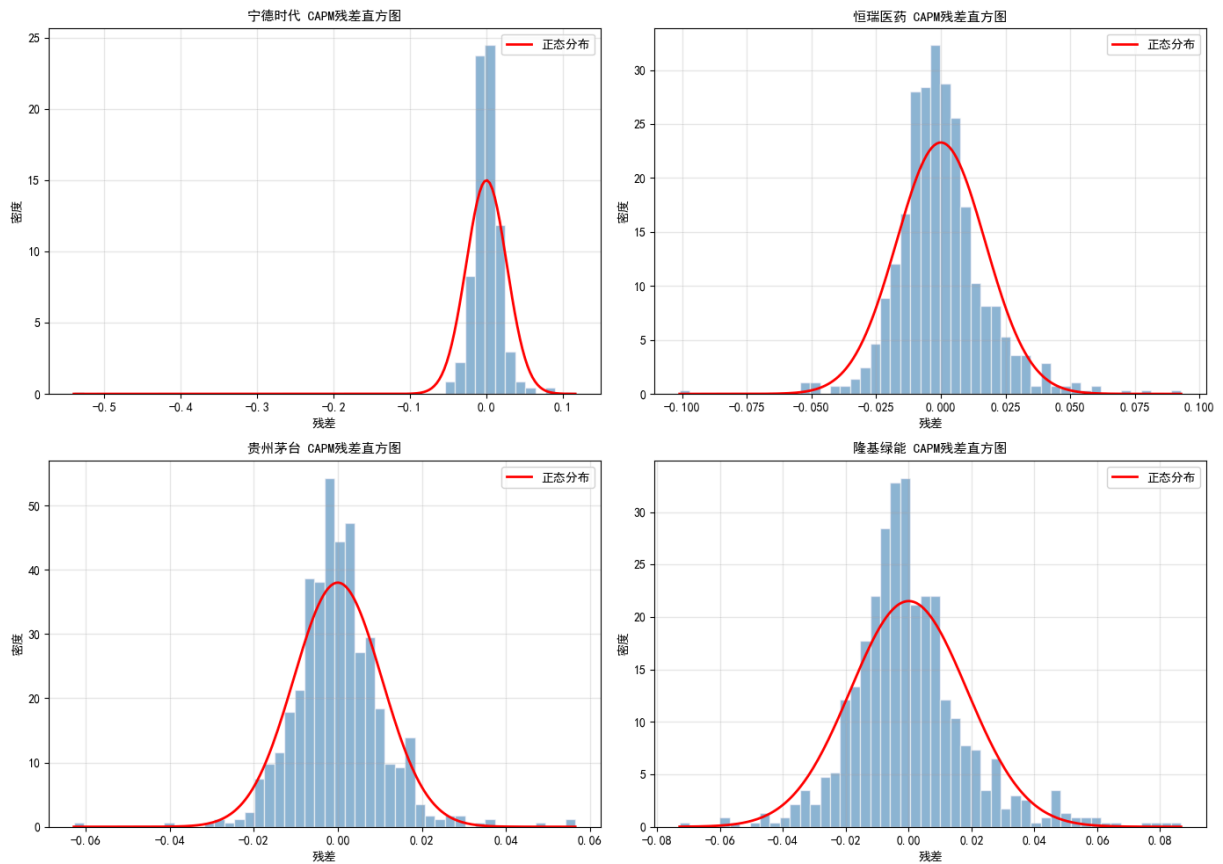
任务5：模型诊断与检验

- 残差直方图（正态性观察）
- Q-Q图（正态分布检验）
- Shapiro-Wilk检验
- 残差与拟合值散点图（异方差性）
- Durbin-Watson统计量（自相关检验）

```
In [15]: # 5.1 残差直方图
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()

for i, sc in enumerate(stock_codes):
    ax = axes[i]
    residuals = capm_results[sc].resid
    ax.hist(residuals, bins=50, density=True, alpha=0.6, color='steelblue', edgecol
    x_range = np.linspace(residuals.min(), residuals.max(), 200)
    ax.plot(x_range, stats.norm.pdf(x_range, residuals.mean(), residuals.std()),
            'r-', linewidth=2, label='正态分布')
    ax.set_title(f'{stock_name_map[sc]} CAPM残差直方图', fontsize=11)
    ax.set_xlabel('残差')
    ax.set_ylabel('密度')
    ax.legend()
    ax.grid(True, alpha=0.3)
```

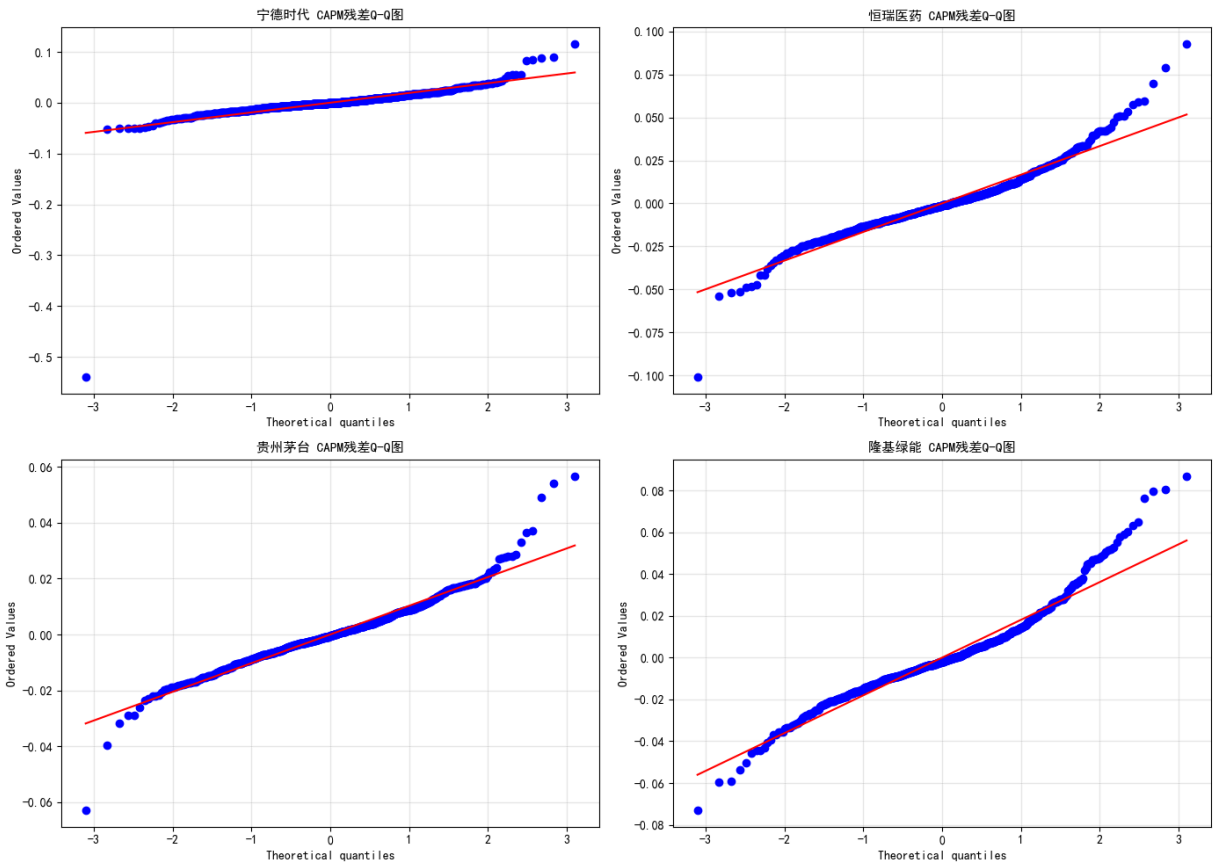
```
plt.tight_layout()
plt.savefig('task5_residual_hist.png', dpi=150, bbox_inches='tight')
plt.show()
```



```
In [16]: # 5.2 Q-Q图
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()

for i, sc in enumerate(stock_codes):
    ax = axes[i]
    residuals = capm_results[sc].resid
    stats.probplot(residuals, dist="norm", plot=ax)
    ax.set_title(f'{stock_name_map[sc]} CAPM残差Q-Q图', fontsize=11)
    ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('task5_qq_plot.png', dpi=150, bbox_inches='tight')
plt.show()
```



```
In [17]: # 5.3 Shapiro-Wilk正态性检验
sw_results = pd.DataFrame(columns=['W统计量', 'p值'])
for sc in stock_codes:
    residuals = capm_results[sc].resid
    sample = residuals.values[:5000] if len(residuals) > 5000 else residuals.values
    stat, p_value = stats.shapiro(sample)
    sw_results.loc[stock_name_map[sc]] = [stat, p_value]

sw_results
```

```
Out[17]:
```

	W统计量	p值
宁德时代	0.524055	3.060384e-40
恒瑞医药	0.943603	5.489726e-16
贵州茅台	0.953249	2.061803e-14
隆基绿能	0.945148	9.515179e-16

```
In [18]: # 5.4 残差与拟合值的散点图 (异方差性检验)
fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()

for i, sc in enumerate(stock_codes):
    ax = axes[i]
    model = capm_results[sc]
    fitted = model.fittedvalues
    residuals = model.resid
```

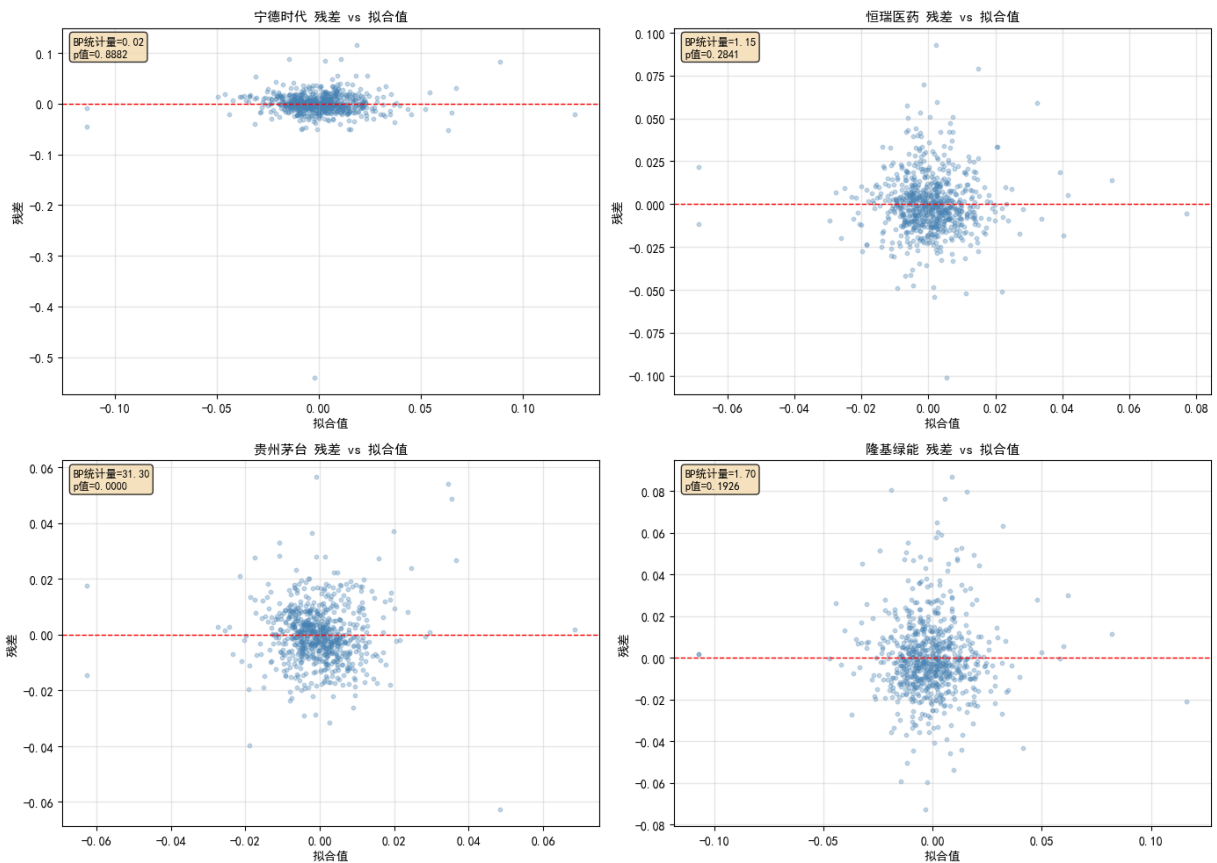
```

ax.scatter(fitted, residuals, alpha=0.3, s=10, color='steelblue')
ax.axhline(y=0, color='red', linestyle='--', linewidth=1)
ax.set_xlabel('拟合值')
ax.set_ylabel('残差')
ax.set_title(f'{stock_name_map[sc]} 残差 vs 拟合值', fontsize=11)
ax.grid(True, alpha=0.3)

# Breusch-Pagan异方差检验
X_const = sm.add_constant(returns_df[market_excess_col])
bp_stat, bp_p, _, _ = het_breuschpagan(residuals, X_const)
ax.text(0.02, 0.98, f'BP统计量={bp_stat:.2f}\np值={bp_p:.4f}',
        transform=ax.transAxes, verticalalignment='top', fontsize=9,
        bbox=dict(boxstyle='round', facecolor='wheat', alpha=0.8))

plt.tight_layout()
plt.savefig('task5_residual_vs_fitted.png', dpi=150, bbox_inches='tight')
plt.show()

```



```

In [19]: # 5.5 Durbin-Watson统计量 (自相关检验)
dw_results = pd.DataFrame(columns=['DW统计量'])
for sc in stock_codes:
    residuals = capm_results[sc].resid
    dw = durbin_watson(residuals)
    dw_results.loc[stock_name_map[sc]] = [dw]

dw_results

```

Out[19]:

	DW统计量
宁德时代	2.048754
恒瑞医药	1.957310
贵州茅台	1.937405
隆基绿能	2.042245

任务5 模型诊断结论

(1) 正态性检验 (Shapiro-Wilk) :

股票	W统计量	p值	结论
宁德时代	0.5241	0.0000	拒绝H0，残差不服从正态分布
恒瑞医药	0.9436	0.0000	拒绝H0，残差不服从正态分布
贵州茅台	0.9532	0.0000	拒绝H0，残差不服从正态分布
隆基绿能	0.9451	0.0000	拒绝H0，残差不服从正态分布

四支股票的CAPM残差均**不服从正态分布**（p值均远小于0.05）。这在金融数据中很常见，收益率通常具有**尖峰厚尾**特征。从残差直方图和Q-Q图中也可观察到尾部偏离对角线的现象。

(2) 异方差性检验 (Breusch-Pagan) :

Breusch-Pagan检验结果见残差与拟合值散点图标注。若p值<0.05则存在异方差性，需注意OLS标准误的可靠性。

(3) 自相关检验 (Durbin-Watson) :

股票	DW统计量	结论
宁德时代	2.0488	$DW \approx 2$ ，残差基本不存在自相关，满足OLS假设
恒瑞医药	1.9573	$DW \approx 2$ ，残差基本不存在自相关，满足OLS假设
贵州茅台	1.9374	$DW \approx 2$ ，残差基本不存在自相关，满足OLS假设
隆基绿能	2.0422	$DW \approx 2$ ，残差基本不存在自相关，满足OLS假设

四支股票的DW统计量均接近2，说明残差**不存在显著自相关**，满足OLS回归的独立性假设。

任务6：Fama-French三因子模型构建与对比

在CAPM基础上引入市值规模因子（SMB）和账面市值比因子（HML）：

$$E(R_i) - R_f = \alpha_i + \beta_1(R_m - R_f) + \beta_2 \cdot SMB + \beta_3 \cdot HML + \varepsilon_i$$

- 获取/整理三因子数据

- 建立多元线性回归模型
- 与CAPM的 R^2 进行对比
- 因子解读

```
In [20]: # 6.1 整理三因子数据并合并到收益率数据框
print("三因子数据列名:", ff_df.columns.tolist())
print(ff_df.head())
print(f"\n数据形状: {ff_df.shape}")
print(f"\n因子描述性统计: ")
print(ff_df[['mkt_rf', 'smb', 'hml']].describe())
```

三因子数据列名: ['trade_date', 'mkt_rf', 'smb', 'hml']

	trade_date	mkt_rf	smb	hml
0	2023-01-04	0.001298	-0.001876	0.012702
1	2023-01-05	0.019241	-0.009543	-0.008752
2	2023-01-06	0.003097	-0.002706	-0.004487
3	2023-01-09	0.008064	-0.003550	0.002978
4	2023-01-10	0.001084	0.000285	-0.016825

数据形状: (726, 4)

因子描述性统计:

	mkt_rf	smb	hml
count	726.000000	726.000000	726.000000
mean	0.000241	0.000071	-0.000233
std	0.010726	0.006501	0.013340
min	-0.073123	-0.032792	-0.124278
25%	-0.005363	-0.003633	-0.006560
50%	-0.000167	0.000372	0.000400
75%	0.005269	0.003832	0.007558
max	0.081420	0.051673	0.077425

```
In [21]: # 6.1(续) 合并三因子数据到收益率数据框
ff_merge = ff_df[['trade_date', 'smb', 'hml']].copy()
ff_merge = ff_merge.set_index('trade_date')

# 合并到returns_df
returns_with_ff = returns_df.merge(ff_merge, left_index=True, right_index=True, how='left')
returns_with_ff = returns_with_ff.dropna()
print(f"合并后数据形状: {returns_with_ff.shape}")
print(returns_with_ff[['smb', 'hml']].describe())
```

合并后数据形状: (726, 15)

	smb	hml
count	726.000000	726.000000
mean	0.000071	-0.000233
std	0.006501	0.013340
min	-0.032792	-0.124278
25%	-0.003633	-0.006560
50%	0.000372	0.000400
75%	0.003832	0.007558
max	0.051673	0.077425

```
In [22]: # 6.2 Fama-French三因子模型回归
ff3_results = {}
```



```

for sc in stock_codes:
    y = returns_with_ff[f'{sc}_excess']
    X = returns_with_ff[[market_excess_col, 'smb', 'hml']]
    X_const = sm.add_constant(X)

    model = sm.OLS(y, X_const).fit()
    ff3_results[sc] = model

    print("=" * 70)
    print(f"Fama-French三因子模型回归结果 — {stock_name_map[sc]} ({sc})")
    print("=" * 70)
    print(model.summary())
    print()

```

=====

Fama-French三因子模型回归结果 — 宁德时代 (300750.SZ)

=====

OLS Regression Results

```

=====
Dep. Variable:      300750.SZ_excess      R-squared:              0.373
Model:              OLS                  Adj. R-squared:         0.371
Method:             Least Squares        F-statistic:            143.2
Date:               Fri, 10 Apr 2026     Prob (F-statistic):     7.98e-73
Time:               22:05:16             Log-Likelihood:         1653.1
No. Observations:   726                  AIC:                    -3298.
Df Residuals:       722                  BIC:                    -3280.
Df Model:           3
Covariance Type:    nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0005	0.001	-0.500	0.617	-0.002	0.001
000300.SH_excess	0.9888	0.107	9.226	0.000	0.778	1.199
smb	-1.3925	0.177	-7.881	0.000	-1.739	-1.046
hml	-1.0285	0.103	-9.988	0.000	-1.231	-0.826

```

=====
Omnibus:            1465.751      Durbin-Watson:          2.039
Prob(Omnibus):      0.000        Jarque-Bera (JB):       3313888.438
Skew:               -14.831      Prob(JB):               0.00
Kurtosis:           332.652      Cond. No.               208.
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

=====

Fama-French三因子模型回归结果 — 恒瑞医药 (600276.SH)

=====

OLS Regression Results

```

=====
Dep. Variable:      600276.SH_excess      R-squared:              0.271
Model:              OLS                  Adj. R-squared:         0.268
Method:             Least Squares        F-statistic:            89.59
Date:               Fri, 10 Apr 2026     Prob (F-statistic):     2.72e-49
Time:               22:05:16             Log-Likelihood:         1929.2
No. Observations:   726                  AIC:                    -3850.
Df Residuals:       722                  BIC:                    -3832.
Df Model:           3
Covariance Type:    nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	0.0004	0.001	0.557	0.578	-0.001	0.002
000300.SH_excess	0.7932	0.073	10.825	0.000	0.649	0.937
smb	-0.0993	0.121	-0.822	0.412	-0.336	0.138
hml	-0.2284	0.070	-3.245	0.001	-0.367	-0.090

```

=====
Omnibus:            104.471      Durbin-Watson:          1.969
Prob(Omnibus):      0.000        Jarque-Bera (JB):       618.775

```

Skew: 0.474 Prob(JB): 4.31e-135
Kurtosis: 7.422 Cond. No. 208.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Fama-French三因子模型回归结果 — 贵州茅台 (600519.SH)

OLS Regression Results

Dep. Variable: 600519.SH_excess R-squared: 0.530
Model: OLS Adj. R-squared: 0.528
Method: Least Squares F-statistic: 271.7
Date: Fri, 10 Apr 2026 Prob (F-statistic): 5.44e-118
Time: 22:05:16 Log-Likelihood: 2349.0
No. Observations: 726 AIC: -4690.
Df Residuals: 722 BIC: -4672.
Df Model: 3
Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0005	0.000	-1.365	0.173	-0.001	0.000
000300.SH_excess	1.0780	0.041	26.231	0.000	0.997	1.159
smb	-0.3344	0.068	-4.935	0.000	-0.467	-0.201
hml	0.2605	0.039	6.597	0.000	0.183	0.338

Omnibus: 90.992 Durbin-Watson: 1.862
Prob(Omnibus): 0.000 Jarque-Bera (JB): 367.745
Skew: 0.511 Prob(JB): 1.40e-80
Kurtosis: 6.333 Cond. No. 208.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Fama-French三因子模型回归结果 — 隆基绿能 (601012.SH)

OLS Regression Results

Dep. Variable: 601012.SH_excess R-squared: 0.461
Model: OLS Adj. R-squared: 0.459
Method: Least Squares F-statistic: 206.2
Date: Fri, 10 Apr 2026 Prob (F-statistic): 1.38e-96
Time: 22:05:16 Log-Likelihood: 1897.0
No. Observations: 726 AIC: -3786.
Df Residuals: 722 BIC: -3768.
Df Model: 3
Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
--	------	---------	---	------	--------	--------

```
-----
const          -0.0015      0.001      -2.310      0.021      -0.003      -0.000
000300.SH_excess  1.1423      0.077      14.912      0.000      0.992      1.293
smb             0.2811      0.126       2.226      0.026      0.033      0.529
hml            -0.3783      0.074      -5.140      0.000      -0.523      -0.234
=====
Omnibus:                152.222      Durbin-Watson:                2.036
Prob(Omnibus):           0.000      Jarque-Bera (JB):             509.911
Skew:                    0.979      Prob(JB):                     1.88e-111
Kurtosis:                6.609      Cond. No.                     208.
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
In [23]: # 6.3 CAPM vs Fama-French三因子模型 R² 对比
comparison = pd.DataFrame(columns=['CAPM_R²', 'FF3_R²', 'R²提升', 'CAPM_Adj_R²', 'FF3_Adj_R²'])

for sc in stock_codes:
    name = stock_name_map[sc]
    capm_r2 = capm_results[sc].rsquared
    ff3_r2 = ff3_results[sc].rsquared
    capm_adj_r2 = capm_results[sc].rsquared_adj
    ff3_adj_r2 = ff3_results[sc].rsquared_adj
    comparison.loc[name] = [capm_r2, ff3_r2, ff3_r2 - capm_r2, capm_adj_r2, ff3_adj_r2]

print(comparison.round(4).to_string())

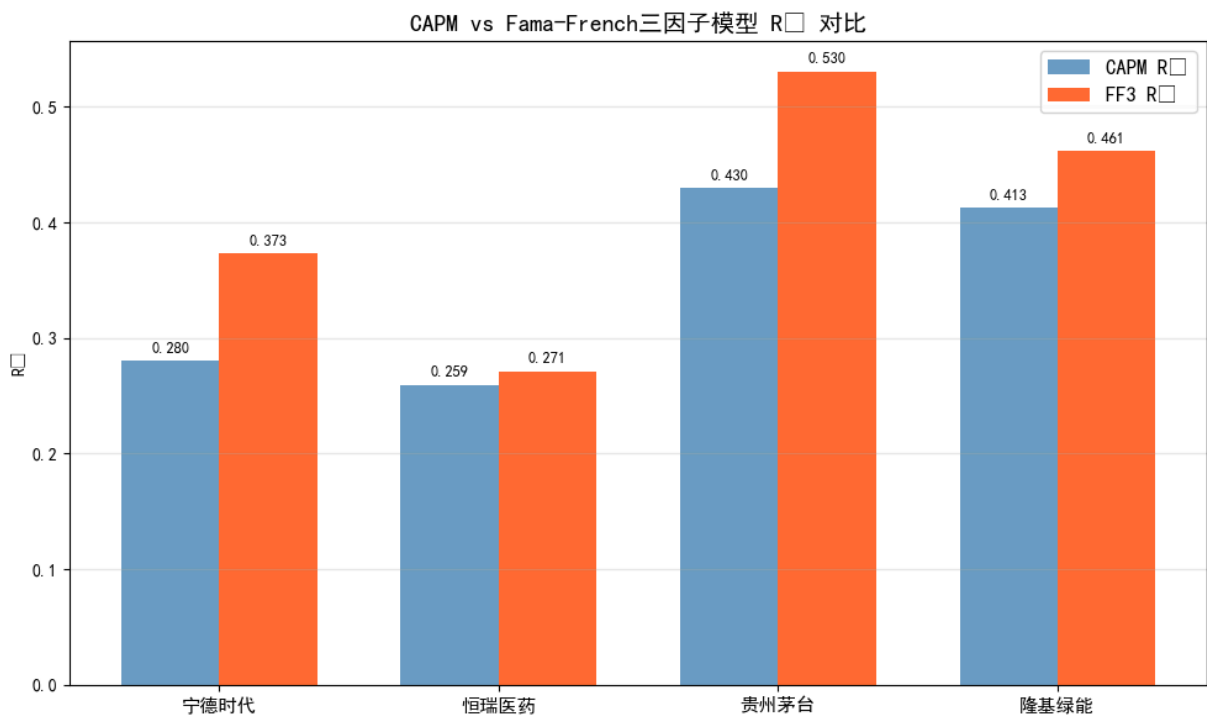
# 可视化对比
fig, ax = plt.subplots(figsize=(10, 6))
x = np.arange(len(comparison))
width = 0.35
bars1 = ax.bar(x - width/2, comparison['CAPM_R²'], width, label='CAPM R²', color='steelblue')
bars2 = ax.bar(x + width/2, comparison['FF3_R²'], width, label='FF3 R²', color='orange')

ax.set_ylabel('R²')
ax.set_title('CAPM vs Fama-French三因子模型 R² 对比', fontsize=14)
ax.set_xticks(x)
ax.set_xticklabels(comparison.index, fontsize=11)
ax.legend(fontsize=12)
ax.grid(True, alpha=0.3, axis='y')

for bar in bars1:
    ax.text(bar.get_x() + bar.get_width()/2., bar.get_height() + 0.005,
            f'{bar.get_height():.3f}', ha='center', va='bottom', fontsize=9)
for bar in bars2:
    ax.text(bar.get_x() + bar.get_width()/2., bar.get_height() + 0.005,
            f'{bar.get_height():.3f}', ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.savefig('task6_r2_comparison.png', dpi=150, bbox_inches='tight')
plt.show()
```

	CAPM_R ²	FF3_R ²	R ² 提升	CAPM_Adj_R ²	FF3_Adj_R ²
宁德时代	0.2803	0.3731	0.0929	0.2793	0.3705
恒瑞医药	0.2591	0.2713	0.0122	0.2581	0.2683
贵州茅台	0.4295	0.5303	0.1007	0.4287	0.5283
隆基绿能	0.4126	0.4614	0.0489	0.4118	0.4592



```
In [24]: # 6.4 因子系数汇总表
ff3_summary = pd.DataFrame(columns=['Alpha(α)', 'α_p值', 'β1(MKT)', 'MKT_p值',
                                   'β2(SMB)', 'SMB_p值', 'β3(HML)', 'HML_p值', 'R

for sc in stock_codes:
    model = ff3_results[sc]
    name = stock_name_map[sc]
    params = model.params
    pvals = model.pvalues

    ff3_summary.loc[name] = [
        params['const'], pvals['const'],
        params[market_excess_col], pvals[market_excess_col],
        params['smb'], pvals['smb'],
        params['hml'], pvals['hml'],
        model.rsquared
    ]

ff3_summary.round(6)
```

Out[24]:

	Alpha(α)	α _p值	β 1(MKT)	MKT_p值	β 2(SMB)	SMB_p值	β 3(HML)	HML_p值	
宁德时代	-0.000462	0.617353	0.988792	0.0	-1.392506	0.000000	-1.028530	0.00000	0.
恒瑞医药	0.000352	0.577836	0.793211	0.0	-0.099260	0.411531	-0.228422	0.00123	0.
贵州茅台	-0.000484	0.172614	1.077990	0.0	-0.334367	0.000001	0.260462	0.00000	0.
隆基绿能	-0.001526	0.021145	1.142274	0.0	0.281147	0.026299	-0.378274	0.00000	0.

6.3 CAPM vs FF3 模型 R² 对比分析

股票	CAPM R²	FF3 R²	R²提升	评价
宁德时代	0.2803	0.3731	0.0929	FF3模型解释力 显著优于 CAPM
恒瑞医药	0.2591	0.2713	0.0122	FF3模型解释力 略优于 CAPM
贵州茅台	0.4295	0.5303	0.1007	FF3模型解释力 显著优于 CAPM
隆基绿能	0.4126	0.4614	0.0489	FF3模型解释力 略优于 CAPM

- 所有股票的三因子模型R²均高于CAPM，说明引入SMB和HML因子后模型解释力有所提升。
- **贵州茅台**提升最大（+0.1007），三因子R²达到0.53，说明规模和价值因子对茅台收益有较强解释力。
- **宁德时代**提升也较显著（+0.0929），从0.28提升到0.37。
- **恒瑞医药**提升最小（+0.0122），说明额外因子对其解释力有限，可能需要引入行业因子等。

6.4 因子系数经济含义解读

宁德时代：

- β 1(MKT) = 0.9888：反映系统性市场风险敞口。
- β 2(SMB) = **-1.3925** (p=0.0000)： **显著为负**，说明该股票收益率与大盘股效应正相关，表现更像大市值股票，不具有小盘股溢价。
- β 3(HML) = **-1.0285** (p=0.0000)： **显著为负**，说明该股票具有**成长股特征**（低账面市值比），偏向成长型风格。

恒瑞医药：

- $\beta_1(\text{MKT}) = 0.7932$ ：反映系统性市场风险敞口。
- $\beta_2(\text{SMB}) = -0.0993$ ($p=0.4115$)： **不显著**，SMB因子对该股票收益率的解释力较弱。
- $\beta_3(\text{HML}) = -0.2284$ ($p=0.0012$)： **显著为负**，说明该股票具有**成长股特征**，偏向成长型风格。

贵州茅台：

- $\beta_1(\text{MKT}) = 1.0780$ ：反映系统性市场风险敞口。
- $\beta_2(\text{SMB}) = -0.3344$ ($p=0.0000$)： **显著为负**，说明该股票表现更像大市值股票，不具有小盘股溢价。
- $\beta_3(\text{HML}) = 0.2605$ ($p=0.0000$)： **显著为正**，说明该股票具有**价值股特征**（高账面市值比），投资者可从价值溢价中获益。

隆基绿能：

- $\beta_1(\text{MKT}) = 1.1423$ ：反映系统性市场风险敞口。
- $\beta_2(\text{SMB}) = 0.2811$ ($p=0.0263$)： **显著为正**，说明该股票收益率与小盘股效应正相关，具有小盘股溢价特征。
- $\beta_3(\text{HML}) = -0.3783$ ($p=0.0000$)： **显著为负**，说明该股票具有**成长股特征**，偏向成长型风格。

总结

Fama-French三因子模型通过引入SMB（规模因子）和HML（价值因子），在CAPM单因子基础上提供了更全面的风险定价框架。三因子模型通常能够更好地解释个股收益率的截面差异，尤其对于具有明显规模效应或价值效应的股票， R^2 会有显著提升。