

上市公司酒业股价分析 —— 贵州茅台 (600519)

基于本地CSV数据，对贵州茅台的历史股价进行分析，包括：

1. 读取数据并进行基本统计
2. 数据持久化（存储到SQLite数据库）
3. 数据预处理（缺失值、异常值）
4. 股价走势与成交量可视化
5. 日收益率与累计收益率分析
6. 移动均线（MA5/MA10/MA20）与RSI技术指标

1. 读取数据并进行基本统计

```
In [1]: import pandas as pd
import numpy as np
import sqlite3
import warnings
import matplotlib
import matplotlib.pyplot as plt

warnings.filterwarnings('ignore')
font = {'family': 'Microsoft YaHei', 'weight': 'bold', 'size': '12'}
matplotlib.rc('font', **font)

# 读取本地茅台CSV数据
df = pd.read_csv('4170425-maotai.csv', index_col=0)
df['date'] = pd.to_datetime(df['date'])
df = df.sort_values('date').reset_index(drop=True)

print(f'数据时间范围: {df["date"].min().strftime("%Y-%m-%d")} 至 {df["date"].max().strftime("%Y-%m-%d")}')
print(f'共 {len(df)} 条记录\n')
print('数据前5行: ')
print(df.head())
print('\n基本统计描述: ')
print(df[['open', 'close', 'high', 'low', 'volume']].describe())
```

数据时间范围：2020-01-02 至 2023-07-10
共 852 条记录

数据前5行：

	date	open	close	high	low	volume	code
0	2020-01-02	1022.186	1024.186	1039.246	1010.186	148099.0	600519
1	2020-01-03	1011.186	972.746	1011.186	971.086	130318.0	600519
2	2020-01-06	965.046	972.176	987.086	961.486	63414.0	600519
3	2020-01-07	971.686	988.716	993.186	970.586	47853.0	600519
4	2020-01-08	979.236	982.326	989.686	976.766	25008.0	600519

基本统计描述：

	open	close	high	low	volume
count	852.000000	852.000000	852.000000	852.000000	852.000000
mean	1677.952669	1679.107951	1701.035146	1657.000134	35202.976526
std	285.914969	284.409359	289.229684	280.299248	17439.177680
min	879.186000	890.186000	904.866000	854.286000	12472.000000
25%	1590.254000	1591.280750	1615.546750	1571.211000	23517.250000
50%	1720.029000	1720.659000	1739.089000	1699.756500	30835.500000
75%	1862.697750	1862.111500	1882.179000	1838.656500	40818.250000
max	2499.191000	2512.211000	2539.091000	2396.211000	148099.000000

2. 数据持久化 —— 存储到SQLite数据库

```
In [2]: conn = sqlite3.connect('liquor_stock.db')
df.to_sql('maotai_daily', conn, if_exists='replace', index=False)

# 验证存储
df_check = pd.read_sql('SELECT * FROM maotai_daily LIMIT 5', conn)
print('数据库中存储的数据 (前5行): ')
print(df_check)
conn.close()
```

数据库中存储的数据 (前5行)：

	date	open	close	high	low	volume	\
0	2020-01-02 00:00:00	1022.186	1024.186	1039.246	1010.186	148099.0	
1	2020-01-03 00:00:00	1011.186	972.746	1011.186	971.086	130318.0	
2	2020-01-06 00:00:00	965.046	972.176	987.086	961.486	63414.0	
3	2020-01-07 00:00:00	971.686	988.716	993.186	970.586	47853.0	
4	2020-01-08 00:00:00	979.236	982.326	989.686	976.766	25008.0	

	code
0	600519
1	600519
2	600519
3	600519
4	600519

3. 数据预处理（缺失值填充、异常值处理）

```
In [3]: # 从数据库读取数据
conn = sqlite3.connect('liquor_stock.db')
df = pd.read_sql('SELECT * FROM maotai_daily', conn)
conn.close()
```

```

df['date'] = pd.to_datetime(df['date'])

print('缺失值统计: ')
print(df.isnull().sum())

# 缺失值填充 (前向填充)
df = df.fillna(method='ffill')

# 异常值处理: 去除价格为0或负数的记录
df = df[(df['open'] > 0) & (df['close'] > 0) & (df['high'] > 0) & (df['low'] > 0)]
df = df.reset_index(drop=True)

print(f'\n预处理后数据量: {len(df)} 条')
print('\n预处理后统计描述: ')
print(df[['open', 'close', 'high', 'low', 'volume']].describe())

```

缺失值统计:

```

date      0
open      0
close     0
high      0
low       0
volume    0
code      0
dtype: int64

```

预处理后数据量: 852 条

预处理后统计描述:

	open	close	high	low	volume
count	852.000000	852.000000	852.000000	852.000000	852.000000
mean	1677.952669	1679.107951	1701.035146	1657.000134	35202.976526
std	285.914969	284.409359	289.229684	280.299248	17439.177680
min	879.186000	890.186000	904.866000	854.286000	12472.000000
25%	1590.254000	1591.280750	1615.546750	1571.211000	23517.250000
50%	1720.029000	1720.659000	1739.089000	1699.756500	30835.500000
75%	1862.697750	1862.111500	1882.179000	1838.656500	40818.250000
max	2499.191000	2512.211000	2539.091000	2396.211000	148099.000000

4. 股价走势与成交量可视化

```

In [4]: fig, axes = plt.subplots(2, 1, figsize=(14, 8), sharex=True)

# 收盘价走势
axes[0].plot(df['date'], df['close'], label='收盘价', color='crimson', linewidth=1.5)
axes[0].set_title('贵州茅台 (600519) 收盘价走势')
axes[0].set_ylabel('价格 (元) ')
axes[0].legend()
axes[0].grid(True, alpha=0.3)

# 成交量
axes[1].bar(df['date'], df['volume'], color='steelblue', alpha=0.7)
axes[1].set_title('成交量')
axes[1].set_ylabel('成交量 (手) ')

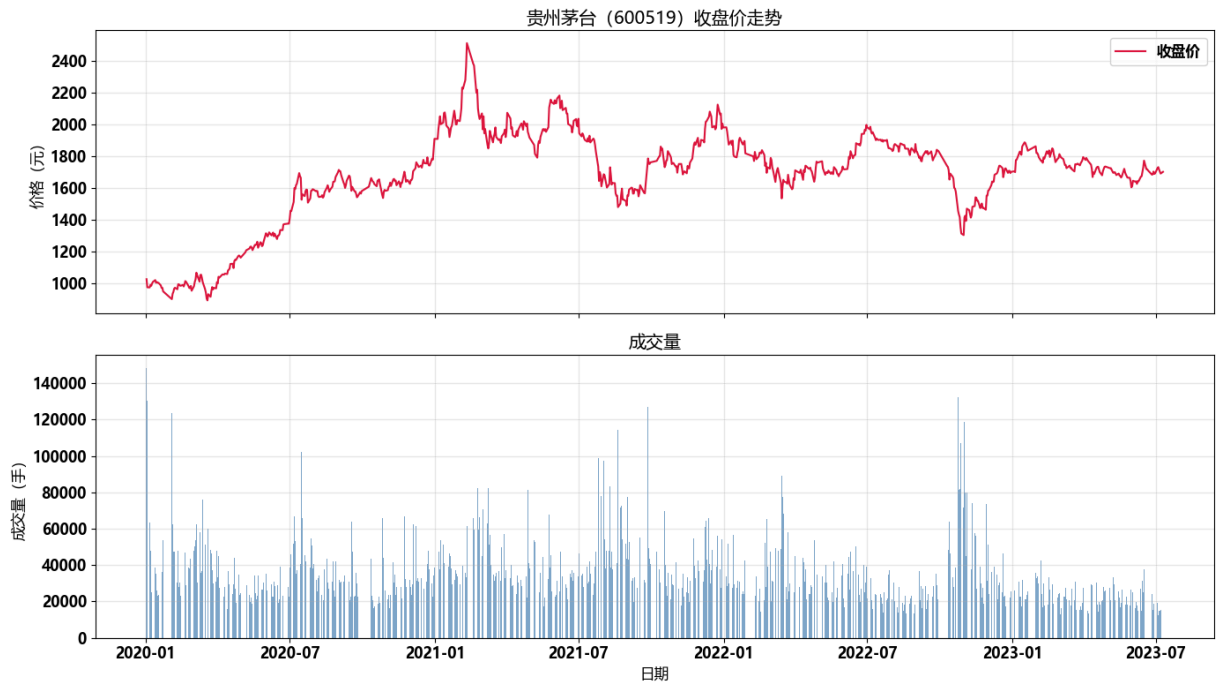
```

```

axes[1].set_xlabel('日期')
axes[1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

```



5. 日收益率与累计收益率分析

```

In [5]: # 计算日收益率
df['daily_return'] = df['close'].pct_change()

# 计算累计收益率
df['cumulative_return'] = (1 + df['daily_return']).cumprod() - 1

fig, axes = plt.subplots(2, 1, figsize=(14, 8))

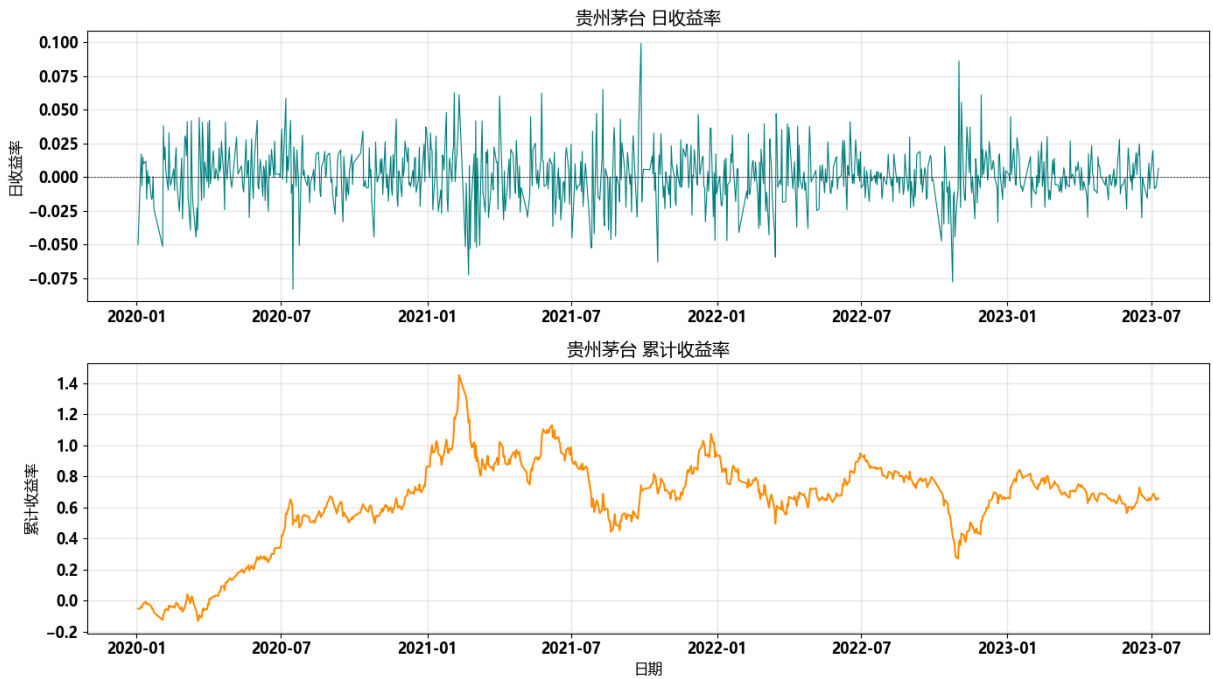
# 日收益率
axes[0].plot(df['date'], df['daily_return'], color='teal', linewidth=0.8)
axes[0].axhline(y=0, color='black', linestyle='--', linewidth=0.5)
axes[0].set_title('贵州茅台 日收益率')
axes[0].set_ylabel('日收益率')
axes[0].grid(True, alpha=0.3)

# 累计收益率
axes[1].plot(df['date'], df['cumulative_return'], color='darkorange', linewidth=1.5)
axes[1].set_title('贵州茅台 累计收益率')
axes[1].set_ylabel('累计收益率')
axes[1].set_xlabel('日期')
axes[1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

```

```
print(f'期间累计收益率: {df["cumulative_return"].iloc[-1]*100:.2f}%')
print(f'日收益率均值: {df["daily_return"].mean()*100:.4f}%')
print(f'日收益率标准差: {df["daily_return"].std()*100:.4f}%')
```



期间累计收益率: 66.07%
 日收益率均值: 0.0804%
 日收益率标准差: 2.0418%

6. 移动均线 (MA) 与RSI技术指标

```
In [6]: # 计算均线
df['MA5'] = df['close'].rolling(window=5).mean()
df['MA10'] = df['close'].rolling(window=10).mean()
df['MA20'] = df['close'].rolling(window=20).mean()

# 计算RSI (14日)
delta = df['close'].diff()
gain = delta.where(delta > 0, 0)
loss = (-delta).where(delta < 0, 0)
avg_gain = gain.rolling(window=14).mean()
avg_loss = loss.rolling(window=14).mean()
rs = avg_gain / avg_loss
df['RSI'] = 100 - (100 / (1 + rs))

# 绘图
fig, axes = plt.subplots(3, 1, figsize=(14, 12), sharex=True)

# 子图1: 股价与均线
axes[0].plot(df['date'], df['close'], label='收盘价', linewidth=1.5)
axes[0].plot(df['date'], df['MA5'], label='MA5', linewidth=1)
axes[0].plot(df['date'], df['MA10'], label='MA10', linewidth=1)
axes[0].plot(df['date'], df['MA20'], label='MA20', linewidth=1)
axes[0].set_title('贵州茅台 股价与均线')
axes[0].set_ylabel('价格 (元)')
axes[0].legend()
```

```

axes[0].grid(True, alpha=0.3)

# 子图2: 成交量
axes[1].bar(df['date'], df['volume'], color='steelblue', alpha=0.7)
axes[1].set_title('成交量')
axes[1].set_ylabel('成交量 (手)')
axes[1].grid(True, alpha=0.3)

# 子图3: RSI
axes[2].plot(df['date'], df['RSI'], label='RSI(14)', color='purple', linewidth=1.5)
axes[2].axhline(y=70, color='r', linestyle='--', alpha=0.5, label='超买线(70)')
axes[2].axhline(y=30, color='g', linestyle='--', alpha=0.5, label='超卖线(30)')
axes[2].set_title('RSI指标')
axes[2].set_ylabel('RSI')
axes[2].set_xlabel('日期')
axes[2].legend()
axes[2].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

```

