


# 代码结构

## 技术分析/报告

1. 导入 os 模块
2. 导入 pandas 模块, 并简写为 pd
3. 导入 talib 模块
4. 导入 streamlit 模块, 并简写为 st
5. 定义函数 generate\_technical\_analysis\_report(stock):
  - a) 读取名为 data/{stock.lower()}.csv 的 CSV 文件到变量 df
  - b) 删除最后一行数据
  - c) 对 df 进行反转排序
  - d) 计算技术指标: 上轨线 (Upper)、中轨线 (Middle)、下轨线 (Lower)
  - e) 计算技术指标: MACD、信号线 (Signal)、柱状图 (Hist)
  - f) 计算技术指标: ADX、+DI、-DI
  - g) 初始化变量 report, 用于存储生成的技术分析报告的字符串
  - h) 执行“成交量-价格分析”:
    - i. 如果最后 3 天的价格增长的天数大于等于 3 且最后 3 天的成交量增长的天数大于等于 3, 则向 report 添加“成交量-价格分析: 确认上涨趋势。最后价格连续 3 天上涨, 且成交量增加。”的文本
    - ii. 否则, 如果最后 3 天的价格下降的天数小于等于 -3 且最后 3 天的成交量增长的天数大于等于 3, 则向 report 添加“成交量-价格分析: 确认下降趋势。最后价格连续 3 天下降, 且成交量增加。”的文本
    - iii. 否则, 如果最后 3 天的价格增长的天数小于等于 1 且最后 3 天的成交量增长的天数大于等于 3, 则向 report 添加“成交量-价格分析: 观察反转点。最近 3 天价格大部分下降, 但成交量增加。”的文本
    - iv. 否则, 向 report 添加“成交量-价格分析: 没有明确的成交量-价格关系。”的文本
  - i) 执行“波动性分析”:
    - i. 获取最后一个数据点的上轨线 (Upper)、中轨线 (Middle)、下轨线 (Lower) 的值
    - ii. 如果最后一个数据点的价格大于上轨线 (Upper), 则向 report 添加“波动性分析: 股价触及或突破上轨线, 表示股票可能超买。”的文本
    - iii. 否则, 如果最后一个数据点的价格小于下轨线 (Lower), 则向 report 添加“波动性分析: 股价触及或突破下轨线, 表示股票可能超卖。”的文本
    - iv. 否则, 向 report 添加“波动性分析: 股价在布林带范围内, 表示正常交易范围。”的文本
  - j) 执行“趋势分析”:
    - i. 执行“MACD 分析”:
      1. 获取最后一个数据点的 MACD 线和信号线的值
      2. 如果 MACD 线大于信号线, 则向 report 添加“趋势分析 (MACD): MACD 线在信号线上方, 表示看涨信号。”的文本
      3. 否则, 如果 MACD 线小于信号线, 则向 report 添加“趋势分析 (MACD): MACD 线在信号线下方, 表示看跌信号。”的文本
      4. 否则, 向 report 添加“趋势分析 (MACD): MACD 线在信号线上下交叉, 表示可能发生趋势变化。”的文本
    - ii. 执行“ADX 分析”:
      1. 获取最后一个数据点的 ADX、+DI、-DI 的值
      2. 如果 ADX 大于 25, 则向 report 添加“趋势分析 (ADX): ADX 值大于 25, 表示市场趋势强劲。”的文本
      3. 否则, 向 report 添加“趋势分析 (ADX): ADX 值小于 25, 表示市场趋势弱或无趋势。”的文本
  - k) 返回 report
6. 定义函数 load\_chart\_html(stock):
  - a) 设置图表 HTML 文件的路径为 chart/{stock.upper()}output.html
  - b) 如果图表文件存在, 则读取文件内容到变量 chart\_html, 否则返回 None
7. 导入 streamlit.components.v1 模块, 并简写为 components
8. 定义函数 display\_chart(stock):
  - a) 设置图表文件路径为 chart/{stock.upper()}output.html
  - b) 如果图表文件存在, 则读取文件内容到变量 chart\_html
  - c) 在 streamlit 的展开块 (expander) 中显示图表, 使用组件 components.html, 并设置高度为 1600 像素
  - d) 否则, 向 st 写入“未找到股票 {stock} 的图表。”的文本
9. 定义函数 main():

- a) 设置 streamlit 页面的配置：布局为 "wide"、页面标题为 "Technical Analysis Reports"、页面图标为 "
  - b) 在页面中显示标题和侧边栏
  - c) 创建一个下拉列表，供选择股票
  - d) 如果点击"Generate Report"按钮：
    - i. 显示正在生成报告的加载状态
    - ii. 调用 generate\_technical\_analysis\_report 函数生成技术分析报告，并将其存储在变量 report 中
    - iii. 在页面中显示报告内容
    - iv. 调用 display\_chart 函数显示图表
10. 如果当前脚本是主程序：
- a) 调用 main 函数

## 投资组合理论

1. 导入所需的库和模块
2. 设置警告过滤器
3. 定义股票列表
4. 创建一个空的 DataFrame 来存储所有股票数据
5. 循环遍历每个股票：
  - a) 读取 CSV 数据
  - b) 处理和筛选数据
  - c) 将股票数据添加到 DataFrame
6. 对数据中的缺失值进行填充
7. 计算每日收益率和累积收益率
8. 绘制收益率曲线和累积收益率曲线
9. 设置投资组合的股票数量和权重
10. 计算等权重组合的收益率
11. 绘制等权重组合的累积收益率曲线
12. 计算相关矩阵和协方差矩阵
13. 创建相关矩阵的热图
14. 定义优化目标函数和约束条件
15. 通过最小化目标函数获得最优投资组合权重
16. 计算最优投资组合的收益率和累积收益率
17. 绘制最优投资组合和等权重组合的累积收益率曲线
18. 定义目标函数和约束条件来进行有效前沿优化
19. 通过优化目标函数获得有效前沿的投资组合权重
20. 绘制有效前沿和个别股票的散点图
21. 对最小方差投资组合进行风险最小化求解
22. 计算最小方差投资组合的收益率和累积收益率
23. 输出最优投资组合权重和最小方差投资组合权重

24. 计算最优投资组合和最小方差投资组合的风险和收益
25. 绘制有效前沿图，并标记最优投资组合和最小方差投资组合
26. 计算各投资组合的累积收益率
27. 绘制投资组合的累积收益率曲线
28. 计算年化收益率和波动率
29. 计算夏普比率
30. 输出最优投资组合和最小方差投资组合的年化收益率、波动率和夏普比率