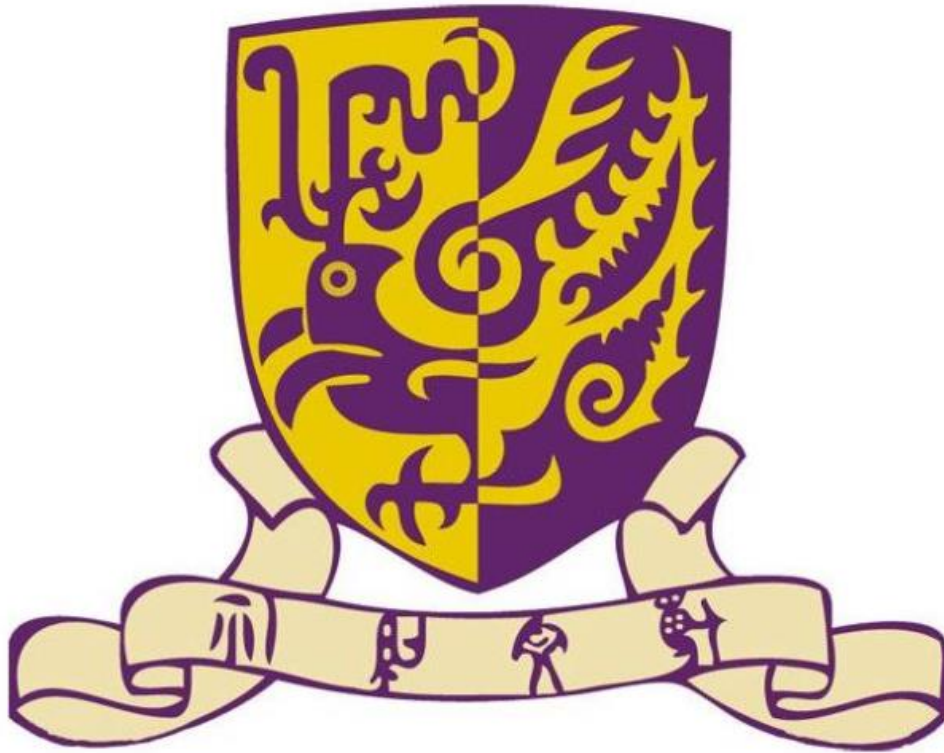


Programming for Financial Engineering

MFE5250



Group Member

王泽宪 216010039

龙静远 216010059

张蕊楠 216010036

马春晖 216010031

张博垚 216010030

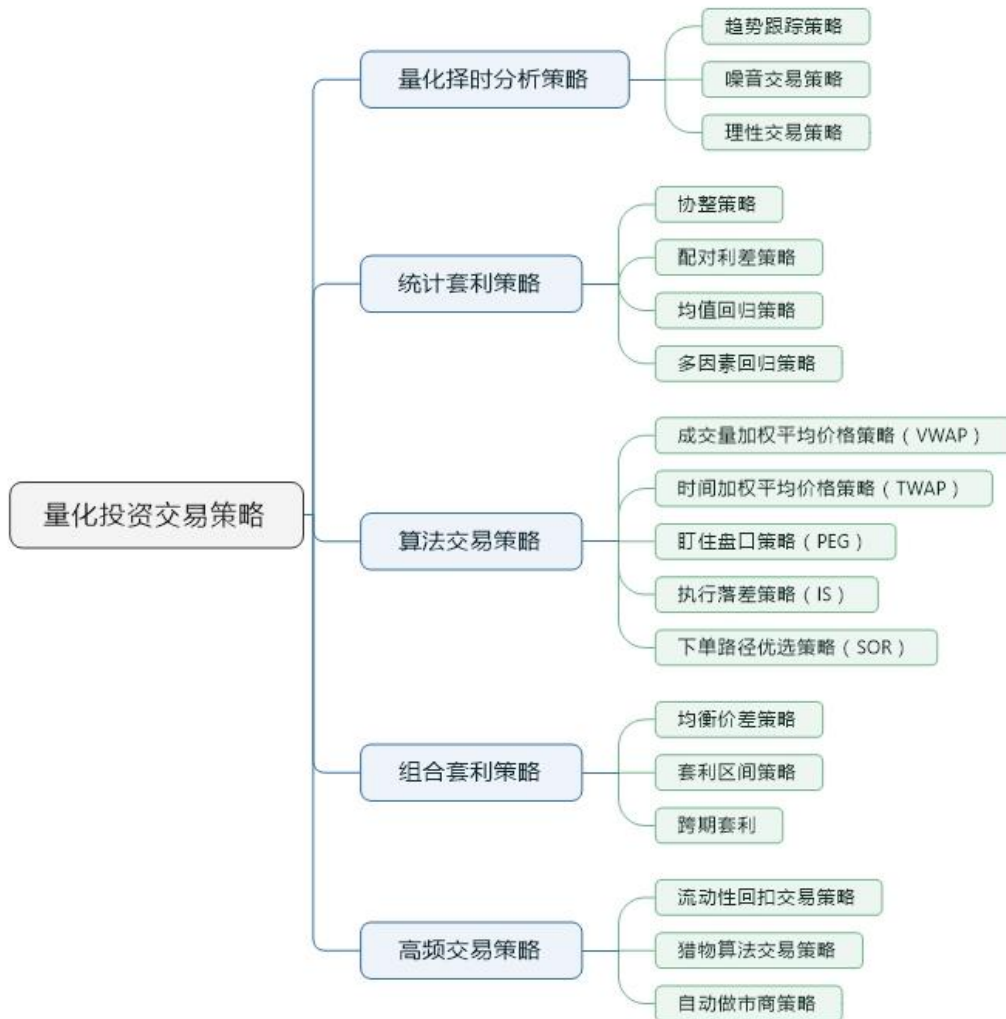
赵紫薇 216010064

目录

一、量化策略综述	3
1、趋势追踪择时策略	4
(1) MA	4
(2) MACD	5
(3) DMA	5
(4) 布林线策略	5
2、多因素回归策略	6
3、组合套利策略	7
4、高频交易策略	8
二、基于三指标的趋势策略设计	9
1、策略基本思路	9
(1) 交易标的选择	9
(2) 技术分析与基本面分析选取	9
(3) “趋势追踪” or “反转策略”	9
(4) “止损”与“止盈”的设置	10
(5) 交易费用与滑点	10
(6) 仓位控制	10
(7) 策略的所选用指标和交易信号选择	10
2、策略历史回测表现	10
(1) 收益率与资产曲线	11
(2) 总收益率、年化收益率、年化标准差、夏普比率	11
(3) 回撤、最大回撤	12
(4) 持仓情况：多空比率、多头天数、空头天数、空仓天数	12
(5) 胜率：总胜率	12
3、策略分析与改进思路	12

一、量化策略综述

量化投资交易策略主要包括如下几类：量化择时策略、统计套利策略、算法交易策略、组合套利策略、高频交易策略等。



量化择时策略是指利用数量化的方法判断大势的未来走势，若判断将上涨则买入，判断将下跌则卖出，从而获得超越大盘的收益。这其中，趋势跟踪策略是普及率最高的策略。

统计套利是一种以模型为基础的投资策略，当现实资产价格偏离理论价格时，同时构建多头和空头组合进行套利，通过计算机程序根据市场信息自动交易。

算法交易是将一笔大单进行拆分，并对拆分后的子单进行定时、定量交易的一种程序化交易方式，目的是尽可能降低因交易规模过大带来的较大冲击成本。通过事先设定好的策略，由投资者编制完成相关的计算机自动化交易程序，利用计算机实现大规模交易的拆分、报价、下单、撤单等行为。

组合套利策略主要是针对期货商品市场上的跨期、跨市及跨品种套利等交易策略。

高频交易是指从那些人们无法利用的极为短暂的市场变化中寻求获利的计算机化交易，比如，某种证券买入价和卖出价差价的微小变化，或者某只股票在不同交易所之间的微小价差。高频交易最主要的特点就是利用计算机代替交易员下单，并且在执行过程中根据市场流动性的统计分布特征，结合实时市况和成交情况，动态调整下单策略，以达到投资者预期的成交效果。

1、趋势追踪择时策略

趋势跟踪择时认为趋势存在延续性，只要找到趋势跟踪操作就可以跑赢大势。这种策略的有效与否主要取决于四个方面，一是交易频率；二是相关技术指标的历史数据跨度；三是触发交易的阈值；四是止损策略。

趋势性指标是运用度最高，也最容易获利的指标。趋势型指标通常利用两根线的交叉作为交易信号，并以此作为买卖时点的判断。

常用的趋势型指标包括：移动均线（MA）、振动升降指标（ASI）、佳庆指标（CHAIKIN）、平均差（DMA）、趋向指标（DMI）、区间振荡指标（DPO）、简易波动指标（EMA）、平滑异同移动平均线（MACD）、三重指数平滑平均线（TRIX）、终极指标（UOS）、十字滤线（VHF）、量价曲线（VPT）、威廉变异离散量（WVAD）等。其中最常用的有 MA、MACD、DMA。

(1) MA

移动均线是以道·琼斯的“平均成本概念”为理论基础，采用移动平均分析的方法，将一段时期内的股票价格平均值连成曲线，用来显示股价的历史波动情况，进而反映股价指数未来发展趋势的技术分析方法，是绝大多数分析运行趋势的基础。按照计算时间区间的不同，移动均线可分为短期、长期等类型，一般来说，计算期间在 20 天以内称为短期，20 天以上称为长期。不同计算长度的移动均线可以用来分析不同时段市场的趋势。

移动平均的计算方法有多种，最常用的是算术移动平均，又称为简单移动平均（SMA），其计算公式为：

$$SMA(N)_t = \frac{1}{N} \sum_{i=0}^{N-1} P_{t-i}$$

其中，N 为移动平均期间， MA_t 为第 t 天的移动平均数， P_{t-i} 为第 t-i 天的收盘价或股价指数。此外还有加权移动平均，包括线性加权、指数加权等方法，而在实际运用中主要以简单的算术移动平均为主。

利用移动均线进行择时交易的方法众多，其中最为著名的是由美国投资人 Jorgesb Ganvle 所创建的交易法则，通常称之为葛南维移动均线八大法则。其中四条用来研判买进时机，四条用来研判卖出时机。简单来说，移动均线在价格之下，而且又呈上升趋势时是买进时机，反之，均线在价格线之上，又呈下降趋势时则是卖出时机。利用移动均线择时的另外一种常用方法是交叉择时法则，即如果一个短期均线从下向上穿过长期均线时，形成所谓金叉，此时应该做多，而当一个长期均线从上向下穿过短期均线时，形成所谓死叉，此时应该做空或空仓。

利用金叉和死叉进行择时不仅在移动均线中运用广泛，而且也是趋势型指标的一个通用法则。

(2) MACD

MACD 指标是根据均线的构造原理,通过分析短期(常用为 12 日)指数移动平均线与长期(常用为 26 日)指数移动平均线之间的聚合与分离状况,对买进、卖出时机做出判断的技术指标,是一种典型的趋势型指标。在 MACD 的计算中,第一步是计算短期(S 日)指数移动平均线和长期(L 日)指数移动平均线 EMA1、EMA2;第二步,计算离差值 $DIF = EMA1 - EMA2$;第三步,计算 DIF 的 N 日指数移动平均线,即 DEA;最后,计算 $MACD = 2 * (DIF - DEA)$ 。

实际运用中,当 DIFF、DEA 均为正,DIFF 向上突破 DEA 时为买入信号;DIFF、DEA 均为负,DIFF 向下跌破 DEA 时为卖出信号,DEA 线与 K 线发生背离时为行情反转信号。

(3) DMA

DMA 是依据快慢两条移动平均线的差值情况分析价格趋势的一种技术分析指标。它主要通过计算两条基准周期不同的移动平均线的差值,来判断当前买入卖出的能量的大小和未来价格走势的趋势。

计算中,第一步,计算短期(S 日)移动均线 and 长期(L 日)移动均线 MA1、MA2;第二步,计算平均线差 $DMA = MA1 - MA2$;第三步,计算 DMA 的 M 日移动平均线,即 AMA。当 DMA 向上交叉其平均线 AMA 时,买进;DMA 向下交叉其平均线 AMA 时,卖出;DMA 与股价产生背离时的交叉信号,可信度较高。

技术分析的理论基础基于三项市场假设:市场行为涵盖一切信息;价格沿趋势移动;历史会重演。从这三个假设出发,利用均线型指标来追踪趋势在一定程度上过滤了偶然因素带来的影响,使得行情的趋势更加清晰,消除股价暂时的起伏。此外,当价格突破了移动均线时,移动均线对价格有助涨助跌作用。

另一方面,由于移动均线反应的是过往一段时间的行情,当原有趋势发生反转时,移动均线的反应往往比较迟缓,以致无法及时反应操作时点。

(4) 布林线策略

布林线指标是围绕股价的波动画出的一个通道范围,给出了股价波动的区间,同时通过中轨线来配合判断趋势。布林线可以用于趋势策略,也可以用于震荡策略。

布林线是由三条轨道线组成,中轨线是价格平均线,上轨线是相对于价格的压力线,下轨线是相对于价格的支撑线,价格在由上下轨线组成的带状区间内游走,随价格的变化轨道自动调整其方向和位置。一般来说,当价格上涨到上轨线时,处于超买状态,向上延伸过度;当价格跌到下轨线时,处于超卖状态,向下延伸过度。关于中轨线的设置有多种方法,包括移动平均线、加权平均线和指数平滑平均线等都被用来作为布林线的中轨线。布林线的带宽反映了波动率的变化情况。通常布林线的带宽是中轨移动平均线加减若干倍价格的标准差来计算确定。根据带宽的计算方法可知带宽会随着价格波动幅度变化,当价格向上或者向下突破时,波动幅度变大,波动带会变宽,这时阻力线和支撑线打开,称为开口;价格横盘整

理时，波动较小，波动带会收窄，阻力线和支撑线收缩，称为收口，即布林线的波动带区间是会随着股价的剧烈或者细微变化而不断调整变化的，反过来投资者通过布林线的变化就在一定程度上可察觉价格趋势可能会发生的变化。

实际上布林线的带宽在实际应用中作用巨大，在布林线所有指标中是最有效的，带宽的收敛与放大可以反映其他指标所不能反映的信号。带宽对于趋势的识别作用主要体现在当带宽突然急剧放大或者长时间收敛后，通常会发生趋势的变化。具体来说有三种情况：

(1) 当布林线带宽迅速放大，带宽值突破六个月（布林格认可的带宽收敛状态的定义是：当证券价格波动率下降至六个月以来的最低点时，可认为布林线带宽处于收缩状态）来的高点，即突破收敛状态时，意味着价格处于异动状态，波动率较高，发生价格趋势转折的可能性很大；

(2) 带宽也能预示新的价格趋势形成前的征兆，当价格波动越来越小并且持续一段时间，即价格表现为横盘窄幅整理，均线变成水平状态，上下轨与中轨线的距离会越来越小，布林线就会收窄，处于收敛状态，通常意味着趋势随时可能发生转向，上涨下跌都有可能。此后如果带宽突然放大，就会出现价格趋势的反转。因为当波动率减少到一定幅度之后的迅速放大，会导致布林线带宽迅速放大，与之前的收敛状态相反；

(3) 布林线的带宽还能预示一波强劲趋势的结束。当一段强烈的上涨或者下跌趋势形成之后，价格的波动会快速放大，带宽会以夸张的方式放大开口，此后如果下轨线转头向下或上轨线转头向上，那就表示这一波强烈趋势暂时结束。

因此，当股价向下跌穿下轨线时，买入信号出现，当价格由下向上穿越下轨线时，可以看作买入信号；价格从下向上穿越中轨线时，表明处于强势，价格可能会加速上涨，是加仓买入信号；价格在中轨线和上轨线之间运行时，走势偏强，处于多头市场，可以持仓观望；价格在较长的时间里运行在中轨线与上轨线之后，从上向下跌穿中轨线是卖出信号；价格强势上涨的股票股价短期可能会穿出布林线上轨运行，一旦上穿上轨线过多，成交量没有跟上，没办法持续放量，需要注意短期的回调压力，如果由价格回落跌破上轨线，这是卖出信号。

2、多因素回归策略

多因素回归策略是统计套利中的一种。由于国内 A 股的做空限制，统计套利目前更普遍的应用于国外对冲基金的投资活动，并创造了可观的收益，能够有效对冲系统风险。

统计套利依赖的基本原理是均值回复，即长期中市场价格将回复到它的长期均值价格。利用这个原理构建资产组合，买进被低估的资产的同时卖出被高估的资产，当未来价格偏差得到纠正，就能得到一个无风险的超额收益率。统计套利的数学定义强调零初始成本，用 $x(t)$ ： $t \geq 0$ 表示，并且累计的折现值为 $v(t)$ ，满足如下条件：

$$(1) v(0) = 0$$

$$(2) \lim_{t \rightarrow \infty} E[v(t)] > 0$$

$$(3) \lim_{t \rightarrow \infty} \frac{\text{Var}[v(t)]}{t} = 0$$

$$(4) \lim_{t \rightarrow \infty} P(v(t) < 0) = 0$$

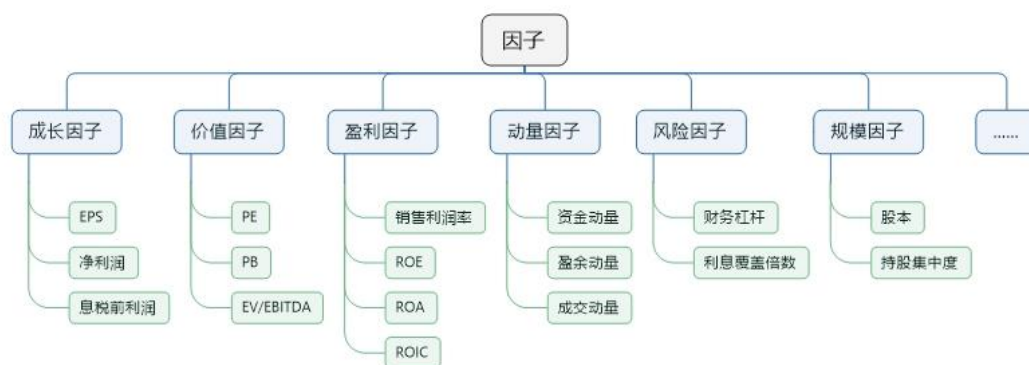
条件(1)表明零初始成本以及自融资的交易策略；条件(2)意味着利润的现值为正，即统计套利有条件地向纯套利收敛；条件(3)说明时间平均的方差趋近 0；条件(4)表示出现亏损的概率接近于 0，这一点可以通过组合的重新调整或者控制空头和多头头寸的总额来避免过大

的风险敞口。

多因素套利策略的典型代表是套利定价模型（APT），这个模型假定收益率的影响因素包括一系列的系統因素和个体的非系統風險，用這些因素進行多元回歸，考慮相關性來選擇個股組建投資組合。多因素模型通過捕捉引起收益變動的共同因素，簡化經典馬爾科夫均值方差模型所要求的證券協方差矩陣系數的個數。

$$r_i = \alpha_i + \beta_1 factor_1 + \cdots + \beta_n factor_n + \varepsilon_i$$

通常考慮的因子分類如下：



根據套利的定義，如果套利機會存在，套利組合不承擔風險，對任何因素的敏感性為零。

然而在此類策略的應用中，需要首先對各影響因素有個大致的估計，然後判斷資產的價格是被高估還是低估。這就存在許多偏差，導致不準確的估計結果，從而導致套利進入或平倉出逃的時機選擇失誤。此外，由於消除了市場風險，在牛市時無法獲得市場整體向上的收益。

3、組合套利策略

套利交易是指在兩個不同的市場中利用暫時的價格差異同時買入和賣出兩個本質相同的證券的行為，以期當未來價格回復時獲得無風險收益。具體可分為跨期套利、跨品種套利和跨市場套利。跨期套利是買賣同一市場同種商品不同到期月份的期貨合約，利用不同到期月份合約的價差變動來獲利的套利模式。跨品種套利是利用兩種不同的但相互關聯的商品之間的價格變動進行套期圖利。即買入某種商品某一月份期貨合約的同時賣出另一相互關聯商品相近交割月份期貨合約。主要有①相關商品間套利，例如銅和鋁；②原料與成品之間套利，如大豆與豆粕。跨市場套利是在某一期貨市場買入(賣出)某一月份商品期貨合約的同時在另一市場賣出(買入)同種合約以期在有利時機對沖獲利了結的交易方式。

套利策略的優點在於：

- 1、較低的初始成本。比如當進行商品跨市套利時，只需在兩個交易所分別交納保證金，通常為期貨價格的10%，並預留部分預備金，與商品價格相比是很小的一部分。另外，由於建立的是方向相反的头寸，也降低了初始投資成本；
- 2、較低的風險。與投機相比，套利交易一定程度上對沖了風險，利用市場上不合理

的价差关系进行操作，用一个市场上的收益抵消另一个市场上的亏损，回避了突发事件对盘面冲击的风险。

但是，正由于套利交易的低风险和追求价差利润的特性，使得单笔交易的获利较低，需要大资金量和高频率的操作。另外，套利交易的关键在于发现价格之间的背离，尽管价格有恢复正常的趋势，但当市场上的非理性投机行为过于严重时，价格会偏离得更远。尽管市场最终会恢复理性，但在此之前套利者可能会面临很大的浮动亏损。控制该风险的方法，包括选好入市时机、合理安排仓位等。

4、高频交易策略

高频交易是指从极为短暂的市场变化中寻求获利的计算机化交易，比如，某种证券买入价和卖出价差价的微小变化，或者某只股票在不同交易所之间的微小价差。由于利用的是微小价差，高频交易的每笔收益率很低，但是交易量巨大，日内交易次数极多。这种属于市场中性策略，不太受牛市熊市的影响，只要有波动就能赚钱。

流动性提供是高频交易最为广泛的策略。海外市场中，像注册做市商、指定做市商有责任和义务向市场提供流动性。通常，做市商可以从两方面获利：一是赚取买卖价差，二是交易回扣，即交易所或交易商为了活跃市场以及提高市场流动性在交易者达到一定交易量时降低交易费用或费用返还。

在国内市场，目前基本上没有高频交易的土壤，股票市场是 T+1，股指期货市场的持仓、交易频率都有很大的限制。商品期货市场可以做一些日内的短线交易，但是离高频交易尚且有很大的距离。

高频交易商相信极短的持仓时间使得高频交易较之传统交易模式具有更好的风险保护。而且，由于对消息的极快吸收、反应和较小的隔夜头寸，能够在突发和重要事件出现时有效地降低风险。较低的风险使得高频交易商能够寻找更多盈利的机会。

高频交易不仅带来了高利润，而且能够更好地处理风险、更有效地提升多个市场间的流动性。同时，利润最大化使得市场能够更有效率地运行，使得市场有更低的波动性、更高的流动性、更好的交易透明性以及更分散、多样的市场影响。

另一方面，高频交易涉及硬件、软件与网络通信等多个基础构件的部署，因而其最直接的风险即来自于硬件损坏、软件错误、通信中断和黑客攻击等的影响，通常将此类风险统称为操作风险或数码风险(Cyber Risk)。随着高频交易业务的不断增加，新技术的不断采用，交易系统势将愈发复杂，使得操作风险必将更频繁地出现。而且高频交易涉嫌市场公平问题，高频交易需要的设备和计算能力对中小投资者是一种不可逾越的门槛，这些利用高频交易获取收益的机构，可能造成市场的不公平。

二、基于三指标的趋势策略设计

1、策略基本思路

(1) 交易标的选择

一般而言，量化交易会选流通性好，市场容量大的标的资产。因此我们选择的是沪深300ETF。其有以下特点：1、能够较好的代表国内股市的表现。2、市场容量大，买入卖出时造成的冲击成本小。3、是被最广泛采用的指数，有较多做空及做多手段。相关ETF基金种类多样，不仅有反向指数ETF，还有自带杠杆的ETF，降低了我们融资和融券的成本。

(2) 技术与基本面分析选取

一般认为，基本面分析是指对宏观经济面、公司主营业务所处行业、公司业务同行业竞争水平和公司内部管理水平等诸多方面的分析，最终来确定标的物的内在价值。技术面分析是指以市场行为为研究对象，以判断市场趋势并跟随趋势的周期性变化来进行股票及一切金融衍生工具交易策略的方法的总和。这两种方法是实际操作中运用最广泛的分析方法。就投资策略而言，从本质上讲，分为趋势跟踪以及均值回归。均值回归，一般是套利策略的范畴，这里不作展开，我们平时大多数的投资策略都属于趋势跟踪，无论是基本面分析还是技术面分析，都是在定义自身框架内的“趋势”，只是定义的方法不同而已。从持续时间来看，基本面分析考虑宏观、行业、公司的情况，数据频率一般较长，频率以周、月见多，从而得出的“趋势”也较长，一般为数个月甚至数年；技术面分析则可长可短，由于分析的是市场数据，以价、量为主，一般用每日的开盘价、最高价、最低价、收盘价以及成交量为分析数据，从而得到信号持有期一般在一个月左右。此外，若用每周数据来分析并构建策略，信号持有期可达到半年以上与基本面策略持有期已经相仿，若用每日高频数据来分析，策略一天可以交易数十次，属于高频交易范畴。趋势的长短由数据本身的频率以及策略的参数所决定的，一般长的趋势有以下特点，信号持有期长，成功率高，最大回撤偏大；而短的趋势则信号持有期短，成功率一般，最大回撤偏小。

(3) “趋势追踪” or “反转策略”

在技术指标分析中，一般分为“趋势”型与“反转”型，但是经过众多的回顾测试，“趋势”策略要远远好于“反转”策略。主要原因是，从本质上来说“反转”策略是相信指标有预测性，也就是说能领先市场。假设市场上100%确定性的领先指标，那一定会被大多数人利用，从而变成同步指标。此外，“反转”型的策略一般称为左侧投资，可以抓住行情趋势第一波的一部分，而相对的“趋势”型策略往往就很难抓住第一波的大部分。但是由于没有100%的领先指标，假如一个“反转”信号错了，那会带来巨大的回撤。这是大多数人不能忍受的。所以在我们的策略中采用的是趋势追踪策略。

(4) “止损”与“止盈”的设置

“止损”策略一般作为辅助型策略运用在很多投资者的策略框架体系内。我们认为，需要“止损”的原因有两个，第一是主观的决策错误。第二是客观的情况变化，例如基本面环境发生了意料之外的变化。“止盈”在理解并坚持了“永远站在零点”这个理念，也可以看作“止损”。对于前两个原因，首先本系列为客观的基于历史数据分析得到的策略，不存在主观决策错误。其次，我们的策略采用买入、卖空策略，即信号改变时即为我们的“止损”点。因此，综上所述，本系列主要以日频数据为主，周频、高频数据为辅，利用各种技术指标来捕捉市场不同的“趋势”的趋势跟踪策略，并且不加入任何“止损”或“止盈”的辅助型策略。

(5) 交易费用与滑点

国内市场的沪深 300ETF 交易费用主要包括佣金和印花税。而我们选择以国外市场中的沪深 300ETF 基金作为交易对象，手续费一般为每笔 10-20 美金，当交易金额较大时，国内的交易成本远高于国外。其次，国内的指数 ETF 基金种类较为单一，多为追踪指数型基金，而国外的 ETF 基金不仅含有杠杆类指数基金，还有反向 ETF 指数基金。这可以极大的降低我们的交易成本。此外，沪深 300 的指数基金市场容量较大，我们买入和卖出对市场造成的冲击较小，所以滑点基本可以忽略。在我们这次的交易中统一设置单笔交易收取费用 20 美元。

(6) 仓位控制

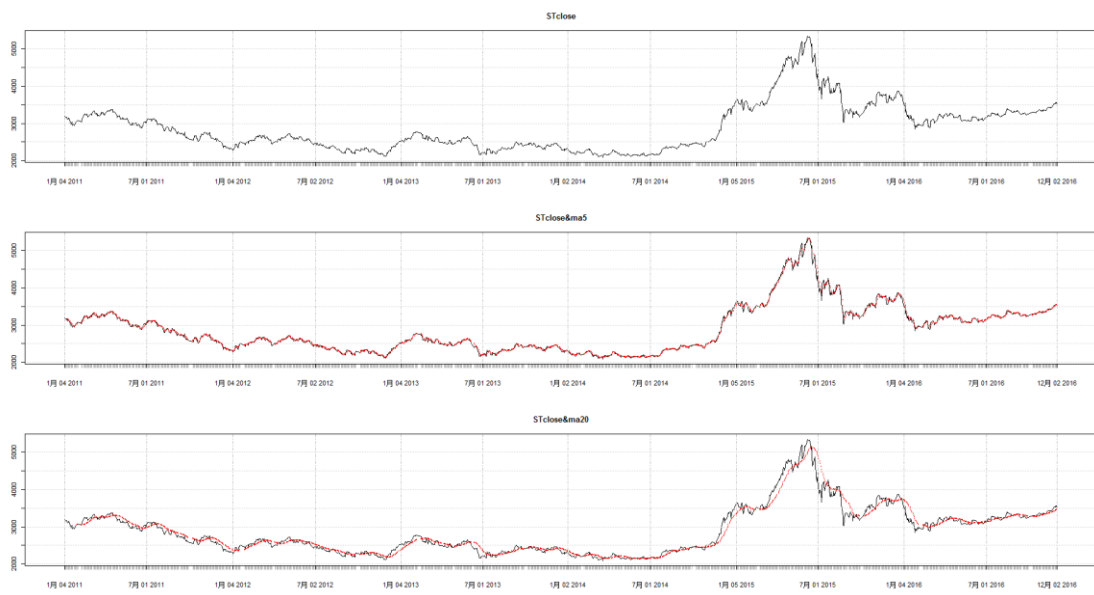
在我们的策略中共设置了四个仓位点，空仓，多头，杠杆多头，空头，这样设置的原因是由于市场中存在直接与之相关的 ETF，可以很方便的进行买入卖出并节省交易费用。四个仓位点分别用 0、1、2 和 -1 表示。

(7) 策略的所选用指标和交易信号选择

本策略尝试将简单移动平均线、双均线交叉和异同移动平均线三个交易策略综合在一起，共同捕捉买卖点。通过运用多种指标捕捉买卖点，一方面可以提高买卖点的捕捉数量，另一方面，在买卖信号方面，不同指标释放的交易信号甚至可能会相反。多种指标综合考虑有助于我们多方面判断市场行情，从而使交易决策更加谨慎，进而有可能提高交易策略的获胜率。在综合策略中，我们规定至少两个指标都释放买入信号时，才买入；至少两个指标都释放卖出信号时，才卖出。4. 将简单移动平均线（10 日）、双均线交叉（5 日和 30 日）和异同移动平均线三个交易策略综合在一起，共同捕捉买卖点，得出：有 69 个买入点，72 个卖出点。然后我们规定，如果在持有多仓时再次遇到买入点，那么就进行杠杆买入；如果在空仓时遇到卖出点，那么就进行融券买入。

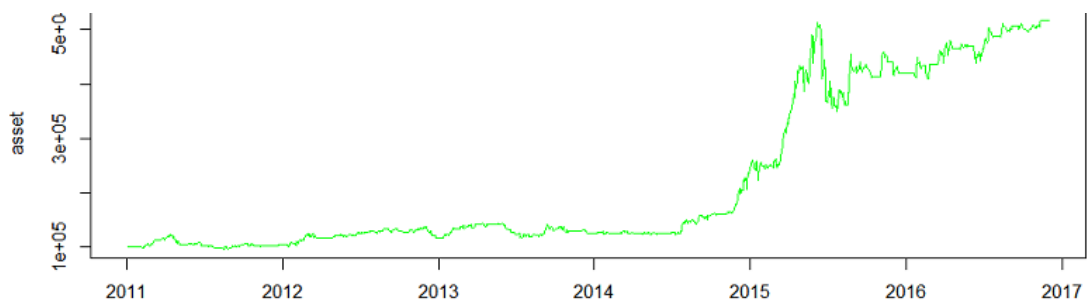
2、策略历史回测表现

回测数据为沪深 300 指数日频，回测区间（2011-1-4——2016-12-2）



图一：沪深股指走势 2011-2016 年走势及 MA5、MA20 曲线

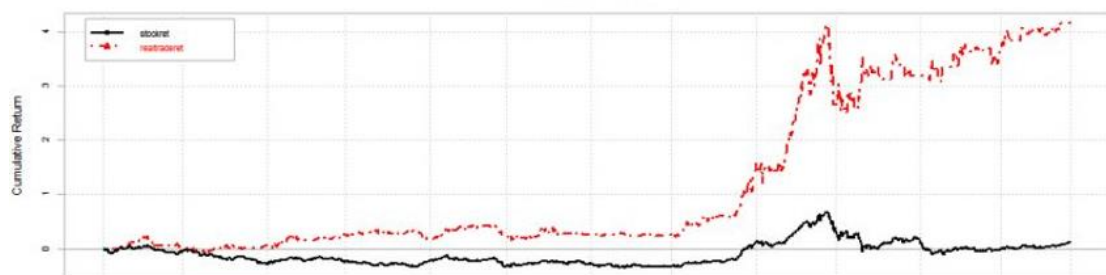
(1) 收益率与资产曲线



图二：策略历史回测资产曲线

资产在 2011 年年初至 2016 年年末六年时间里从 10 万增长至 51.8 万。

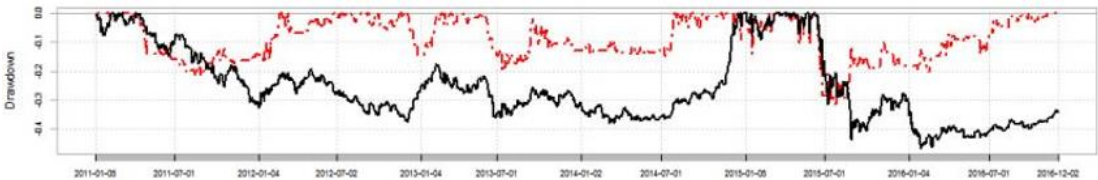
(2) 总收益率、年化收益率、年化标准差、夏普比率



图三：沪深 300 与策略累计收益率曲线

黑色收益率曲线为沪深 300 表现,红色收益率曲线为策略表现。策略存续期总收益率为 418%,平均年化收益 33.46%, 年化夏普比率为 1.22。表现明显优于沪深 300 指数。年化标准差为 8.4%, 沪深 300 指数标准差为。

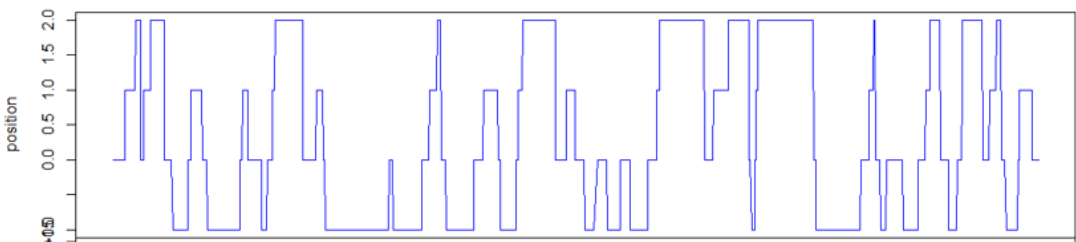
(3) 回撤、最大回撤



图四：沪深 300 与策略回撤曲线

黑色线条代表沪深 300 指数回撤幅度,红色曲线为策略回撤。策略回测在绝大多数情况下小于市场回撤,市场最大回撤为 46%,策略最大回撤为 31.6%。

(4) 持仓情况：多空比率、多头天数、空头天数、空仓天数



图五：策略持仓变化曲线图

策略的交易并不频繁,共捕获了 69 个买入点,72 个卖出点,实际进行交易笔数为 34 笔,多仓 18 次,开空仓 16 次。持有多头的天数为 572 天,持有空头的天数为 501 天,空仓天数为 353 天。平均持仓时间为 31.6 天。较为低频的交易次数减低了交易费用,但是也说明指标的灵敏度欠佳,可能会错过潜在的交易点。空仓时间较少,多头占比 40.11%,空头占比 35.13%,空仓占比为 24.75%。

(5) 胜率与每笔收益率

策略总胜率为 29.4%,多头胜率为 33.33%,空头胜率为 25%。。作为一个趋势追踪策略胜率尚可,平均每笔收益率为 4.96%低于常见的趋势追踪策略。

3、策略分析与改进思路

可以看出策略在牛市和熊市的表现是优于震荡市的,资产的主要增长时期在 2014 年底和 2015 年上半年的牛市阶段。而最大的回撤也发生在 15 年中旬熊市阶段,但是在经过短期调整后在熊市的表现依然强劲。总体而言,策略的胜率较高,能够准确的把握大势。

导致 15 年大幅回撤的原因是策略的达成条件比较苛刻,指标反应滞后,在未来进行改

进时，可以进一步提升策略的灵敏度，将移动均线的周期调整至更短。其次，策略在震荡市的表现不佳，可以对策略进行改进后加入震荡市的新策略。根据指标判断出所处的时期是震荡市还是趋势市，然后选择相应的策略进行交易，有助于进一步提升策略的收益。

附录：策略及回测代码

注：Powered by R，需加载 quantmod

```
setwd("E:/CUHK/subjects/Programming in Financial Engineering/Project/代码")
```

```
library(quantmod)
```

```
#####提取数据#####
```

```
mydata<-read.csv('hs3006.csv',header=TRUE)
```

```
mydata<-xts(mydata[,-1],order.by=as.Date(mydata$date))
```

```
head(mydata)
```

```
#####找出收盘价#####
```

```
dfclose<-mydata[,c(1,5)]
```

```
STclose<-mydata$close
```

```
##将收盘价变量命名为 Close
```

```
names(STclose)<-"Close"
```

```
#前 6 期
```

```
head(STclose)
```

```
#后 6 期
```

```
tail(STclose)
```

```
#对收盘价数据进行总结分析
```

```
summary(STclose)
```

```
#####简单移动平均价 sma#####
```

```
#####截取 2016 年收盘价数据#####
```

```
close<-STclose["2016"]
```

```
###sma10/10 日移动###
```

```
sma106<-SMA(close,10)
```

```
#尾部 6 个数据
```

```
tail(sma106)
```

```
#查看 sma106 的长度
```

```
length(sma106)
```

```
#####若不截取数据 5days#####
```

```
sma5<-SMA(STclose,5)
```

```
sma20<-SMA(STclose,20)
```

```
#####若不截取数据 10days#####
```

```
sma10<-SMA(STclose,10)
```

```
#尾部 6 个数据
```

```
tail(sma5)
```

```
#查看 sma10 的长度
length(sma5)
```

```
#####加权移动平均价 wma#####
#####截取后#####
###wma10/10 日移动###
wma10<-WMA(close,10)
names(wma10)<-"Close.WMA.10"
#尾部 6 个数据
tail(wma10)
#####截取前#####
###wma10/10 日移动###
wma10<-WMA(STclose,10)
names(wma10)<-"Close.WMA.10"
#尾部 6 个数据
tail(wma10)
```

```
#####指数移动平均价 ema#####
#####截取后#####
###ema10/10 日移动###
ema10<-EMA(close,10)
#尾部 6 个数据
tail(wma10)
#####截取前#####
ema10<-EMA(STclose,10)
tail(wma10)
```

```
#####收盘价和 sma10 的时序图#####
sma10<-SMA(STclose,10)
par(mfrow=c(2,1))
plot(STclose)
points(sma10,cex=0.1,col="red")
```

```
##### 以下暂时不跑，只为了配图
#####
par(mfrow=c(3,1))
plot(STclose,main="STclose")
```

```

plot(STclose,main="STclose&ma5")
points(sma5,cex=0.1,col="red")

plot(STclose,main="STclose&ma20")
points(sma20,cex=0.1,col="red")
#####

###接上####简单移动平均线制定 HS 的买卖点#####
STsma<-na.omit(merge(sma10,lag(sma10,1)))
head(STsma)
tail(STsma)

STstclose<-na.omit(merge(STclose,lag(STclose,1)))
STstclose<-STstclose[-(1:9),]
head(STstclose)

###作差法，求收盘价与 10 日移动平均的差值###
sigdata<-STstclose-STsma
colnames(sigdata)<-c("colse_sma10","lag_close_sma10")
head(sigdata)

###构造穿越信号函数
cross<-function(x){
  ifelse(x[1]>0&x[2]<0,1,
  ifelse(x[1]<0&x[2]>0,-1,0))
}

###捕捉价格线突破均线的日期##
SmaSignal<-apply(sigdata,1,cross)
SmaSignal<-xts(SmaSignal,order.by=index(sigdata))
head(SmaSignal)

##制定买卖交易日期
#对交易信号滞后 2 期
SmaTrade<-lag(SmaSignal,2)
SmaTrade<-na.omit(SmaTrade)
head(SmaTrade)

###提取出买入点#####
SmaBuy<-SmaTrade[SmaTrade==1]
length(SmaBuy)
head(SmaBuy)

```



```

####提取出卖出点####
SmaSell<-SmaTrade[SmaTrade==(-1)]
length(SmaSell)
head(SmaSell)

####计算日收益率####
STret<-ROC(STclose,type="discrete")
names(STret)<- "STret"
smaRet<-STret*SmaTrade
names(smaRet)<- "smaRet"
head(smaRet)

####绘制绩效表现图
library(PerformanceAnalytics)
charts.PerformanceSummary(merge(STret["2011-01-05/"],smaRet),lty=c(1,6),main="简单均线交易绩效表现")

#####3
#5 days sma#
sma5<-SMA(STclose,5)
length(sma5)
#30 days sma#
sma30<-SMA(STclose,30)
head(na.omit(sma30))

#长短期均线时序图
plot(sma5,type="l",main="长短期均线时序图")
lines(sma30,lty=6,lwd=3,col="green")
legend("bottomright",legend=c("sma5","sma30"),col=c("black","green"),lty=c(1,6))

SMA5<-embed(sma5,2)
head(SMA5)

SMA30<-embed(sma30,2)

#合并长短期 sma
smaLS<-cbind(SMA5,SMA30)

#转换成时间序列
smaLS<-xts(smaLS,order.by=index(sma5[-1]))

```

```
smaLS<-na.omit(smaLS)
colnames(smaLS)<-c("sma5","lagsma5","sma30","lagsma30")
head(smaLS)
```

```
##构造捕捉向上突破点函数#
```

```
Upcross<-function(x){
  ifelse(x[2]<x[4]&x[1]<x[3],1,0)
}
```

```
#构造捕捉向下突破点函数#
```

```
Downcross<-function(x){
  ifelse(x[2]>x[4]&x[1]>x[3],-1,0)
}
```

```
#捕捉短线向上突破长线日期
```

```
Upsig<-apply(smaLS,1,Upcross)
Upsig<-xts(Upsig,order.by=index(smaLS))
names(Upsig)<-"Upsig"
```

```
##短线向上突破长线，释放买入信号进行买入操作
```

```
UpBuy<-lag(Upsig)
```

```
#查看所有买入点
```

```
UpBuy[UpBuy==1]
```

```
##捕捉短线向下突破长线日期
```

```
Downsig<-apply(smaLS,1,Downcross)
Downsig<-xts(Downsig,order.by=index(smaLS))
names(Downsig)<-"Downsig"
```

```
#短线向上突破长线，释放卖出信号进行卖出操作
```

```
DownSell<-lag(Downsig)
```

```
#查看所有卖出点
```

```
DownSell[DownSell==-1]
```

```
###计算股票日收益率
```

```
STret<-ROC(STclose,type="discrete")
```

```
##计算买入点的预测获胜率
```

```
Long<-UpBuy*STret
names(Long)<-"Long"
```

```

winL<-Long[Long>0]
winLrate<-length(winL)/length(Long[Long!=0])
winLrate

##计算卖出点的预测获胜率
Short<-DownSell*STret
names(Short)<-"Short"
winS<-Short[Short>0]
winSrate<-length(winS)/length(Short[Short!=0])
winSrate

##计算所有买卖点的预测获胜率
UpDnTrade<-UpBuy+DownSell
UpDnTradeRet<-UpDnTrade*STclose
names(UpDnTradeRet)<-"UpDnTradeRet"
winrate<-length(UpDnTradeRet[UpDnTradeRet>0])/length(UpDnTradeRet[UpDnTradeRet!=0])
winrate

#library(PerformanceAnalytics)
#charts.PerformanceSummary(merge(UpDnTradeRet,Long,Short),lty=c(1,2,6),main="长短期均线交易绩效表现")

#####MACD#####
##求 HS 股票的 MACD
macd<-MACD(STclose,12,26,9,maType="EMA",percent=FALSE)
macd<-na.omit(macd)
head(macd)
##MACD(异同均线)捕捉 HS 股票的买卖点
head(mydata)
chartSeries(mydata,theme='white',name='HS2011-2016 蜡烛图',up.col='red',dn.col='green')

HS16<-mydata["2016/"]
chartSeries(HS16,theme='white',name='HS2016 蜡烛图',up.col='red',dn.col='green')

###DIF 和 DEA 两条线向上穿越或者向下穿越的代码思路与之前 5 和 30 一致
macdlag<-lag(macd,1)
macd<-na.omit(merge(macd,macdlag))
head(macd)

####构造 MACD 交易策略函数####
macdcross<-function(x){
  ifelse(x[1]>x[2]&x[3]<x[4]&all(x>0),1,
        ifelse(x[1]<x[2]&x[3]>x[4]&all(x<0),-1,0))
}

```

```

}

###求出 MACD 指标买卖点信号#
macdSignal<-apply(macd,1,macdcross)
#转换成时间序列类型#
macdSignal<-xts(as.numeric(macdSignal),order.by=index(macd))

##设定买卖交易日期##
macdTrade<-lag(macdSignal,1)
head(macdTrade)

macdTrade<-macdTrade[-1]

##HS 股票收益率##
STret<-ROC(STclose,type="discrete")

#MACD 指标交易收益率#
macdRet<-STret*macdTrade

#计算 MACD 指标预测获胜率#
win<-macdRet[macdRet>0]
macdwin<-length(win)/length(macdRet[macdRet!=0])
macdwin

###合并交易信号###
ComboTrade<-merge(SmaTrade,UpDnTrade,macdTrade)
ComboTrade[is.na(ComboTrade)]<-0
head(ComboTrade)
finTrade<-apply(ComboTrade,1,sum)
finTrade<-xts(finTrade,order.by=index(ComboTrade))
head(finTrade)

##至少两个指标都释放出买入信号，才买入
buy<-finTrade[finTrade>=2]
buy
length(buy)
###从上面结果可以看出，有 69 个买入点

#至少两个指标都释放出卖出信号，才卖出
sell<-finTrade[finTrade<=(-2)]
sell
length(sell)
###从上面的结果看，一共有 72 个卖出点

```

```

#
trade<-xts(rep(0,length(finTrade)),order.by=index(finTrade))
trade[finTrade>=2]<-1
trade[finTrade<=-2]<-(-1)
head(trade)

#求交易策略获胜率
ret<-Delt(STclose,k=1,type="arithmetic")
names(ret)<-"stockRet"
tradeRet<-trade*ret
names(tradeRet)<-"tradeRet"
winRate<-length(tradeRet[tradeRet>0])/length(tradeRet[tradeRet!=0])
winRate

library(PerformanceAnalytics)
charts.PerformanceSummary(cbind(ret,tradeRet),lty=c(1,4),main="MACD，简单均线，双均线综合策略",geometric=FALSE)

#####反向操作后，回撤率之类的，
该          部          分          程          序          不          跑
*****

##至少两个指标都释放出买入信号，才买入,反向操作，卖出
sell<-finTrade[finTrade>=2]
sell
length(sell)
###从上面结果可以看出，有 69 个卖出点

#至少两个指标都释放出卖出信号，才卖出，反向操作买入
buy<-finTrade[finTrade<=(-2)]
buy
length(buy)
###从上面的结果看，一共有 72 个买入点

#
trade<-xts(rep(0,length(finTrade)),order.by=index(finTrade))
trade[finTrade>=2]<-(-1)
trade[finTrade<=-2]<-1
head(trade)

#求不考虑连续做空或做多时的交易策略获胜率
ret<-Delt(STclose,k=1,type="arithmetic")
names(ret)<-"stockRet"
tradeRet<-trade*ret

```

```
names(tradeRet)<-"tradeRet"
winRate<-length(tradeRet[tradeRet>0])/length(tradeRet[tradeRet!=0])
winRate
```

```
library(PerformanceAnalytics)
charts.PerformanceSummary(cbind(ret,tradeRet),lty=c(1,4),main="MACD , SMA and Double
EMA Combined Trading Strategy(Reverse operation)",geometric=FALSE)
```

###根据交易信号构造交易状态函数，制定交易状态时，假设股票不能做空

```
Hold<-function(signal){
  n<-length(signal)
  hold<-rep(0,n)
  for (t in (2:n)){
    if(hold[t-1]==0 & signal[t]==1)
      hold[t]<-1
    if(hold[t-1]==0 & signal[t]==(-1))
      hold[t]<-0
    if(hold[t-1]==0 & signal[t]==0)
      hold[t]<-0
    if(hold[t-1]==1 & signal[t]==1)
      hold[t]<-1
    if(hold[t-1]==1 & signal[t]==0)
      hold[t]<-1
    if(hold[t-1]==1 & signal[t]==(-1))
      hold[t]<-0
  }
  return(hold)
}
```

```
hold<-Hold(trade)
```

##定义交易模拟函数

```
TradeSim<-function(price,hold){
  n<-length(price)
  positi<-xts(rep(0,n),order.by=index(price))
  names(positi)<-"position"
  positi[index(hold)]<-as.numeric(hold)
  position<-as.numeric(positi)
```

```

asset<-rep(100000,n)
numprc<-as.numeric(price)
for (t in (2:n)){
  #从 0 变成 1， 开仓
  if(position[t-1]==0 & position[t]==1){
    asset[t]<-asset[t-1]
  }
  #平仓
  if(position[t-1]==1 & position[t]==0){
    asset[t]<-asset[t-1]*(1+ret[t])*(1-0.0003)
  }
  #满仓
  if(position[t-1]==1 & position[t]==1){
    asset[t]<-asset[t-1]*(1+ret[t])
  }
  #空仓
  if(position[t-1]==0 & position[t]==0){
    asset[t]<-asset[t-1]
  }
}
# asset<-cash+price*position*1000
# cash<-xts(cash,order.by=index(price))
asset<-xts(asset,order.by=index(price))
names(asset)<-"asset"
# account<-merge(positi,cash,asset)
account<-merge(positi,asset)
return(account)
}
TradeAccount<-TradeSim(STclose,hold)
head(TradeAccount)

#绘制账户曲线图
plot.zoo(TradeAccount,type="l",col=c("blue","green","red"),main="MACD , SMA and Double
EMA Combined Trading Strategy(No borrowing stocks)")

```

###根据交易信号构造交易状态函数，制定交易状态时，假设股票不能做空，反向操作，修正后的结果（不含交易费用）

```

Hold<-function(signal){
  n<-length(signal)

```

```

hold<-rep(0,n)
for (t in (2:n)){
  if(hold[t-1]==0 & signal[t]==1)
    hold[t]<-(-1)
  if(hold[t-1]==0 & signal[t]==(-1))
    hold[t]<-1
  if(hold[t-1]==0 & signal[t]==0)
    hold[t]<-0
  if(hold[t-1]==1 & signal[t]==1)
    hold[t]<-0
  if(hold[t-1]==1 & signal[t]==0)
    hold[t]<-1
  if(hold[t-1]==1 & signal[t]==(-1))
    hold[t]<-2
  if(hold[t-1]==(-1) & signal[t]==1)
    hold[t]<-(-1)
  if(hold[t-1]==(-1) & signal[t]==0)
    hold[t]<-(-1)
  if(hold[t-1]==(-1) & signal[t]==(-1))
    hold[t]<-0
  if(hold[t-1]==2 & signal[t]==1)
    hold[t]<-0
  if(hold[t-1]==2 & signal[t]==0)
    hold[t]<-2
  if(hold[t-1]==2 & signal[t]==(-1))
    hold[t]<-2
}
return(hold)
}

```

```
hold<-Hold(trade)
```

##定义交易模拟函数

```

TradeSim<-function(price,hold){
  n<-length(price)
  positi<-xts(rep(0,n),order.by=index(price))
  names(positi)<-"position"
  positi[index(hold)]<-as.numeric(hold)
  position<-as.numeric(positi)
  asset<-rep(100000,n)
  numprc<-as.numeric(price)
  for (t in (2:n)){
    #从 0 变成 1
    if(position[t-1]==0 & position[t]==1){

```



```

        asset[t]<-asset[t-1]
    }
#平仓
    if(position[t-1]==1 & position[t]==0){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
#满仓
    if(position[t-1]==1 & position[t]==1){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
#空仓
    if(position[t-1]==0 & position[t]==0){
        asset[t]<-asset[t-1]
    }
    if(position[t-1]==0 & position[t]==(-1)){
        asset[t]<-asset[t-1]
    }
    if(position[t-1]==1 & position[t]==2){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
    if(position[t-1]==(-1) & position[t]==0){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
    if(position[t-1]==(-1) & position[t]==(-1)){
        asset[t]<-asset[t-1]
    }
    if(position[t-1]==2 & position[t]==0){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
    if(position[t-1]==2 & position[t]==2){
        asset[t]<-asset[t-1]*(1+ret[t])
    }
}

# asset<-cash+price*position*1000
# cash<-xts(cash,order.by=index(price))
asset<-xts(asset,order.by=index(price))
names(asset)<-"asset"
# account<-merge(positi,cash,asset)
account<-merge(positi,asset)
return(account)
}

```

```
TradeAccount<-TradeSim(STclose,hold)
head(TradeAccount)
```

#绘制账户曲线图

```
plot.zoo(TradeAccount,type="l",col=c("blue","green","red"),main="MACD , SMA and Double
EMA Combined Trading Strategy(reverse operation)")
```

###根据交易信号构造交易状态函数，制定交易状态时，假设股票不能做空，反向操作，修正后的结果(含交易费用)

```
Hold<-function(signal){
  n<-length(signal)
  hold<-rep(0,n)
  for (t in (2:n)){
    if(hold[t-1]==0 & signal[t]==1)
      hold[t]<-(-1)
    if(hold[t-1]==0 & signal[t]==(-1))
      hold[t]<-1
    if(hold[t-1]==0 & signal[t]==0)
      hold[t]<-0
    if(hold[t-1]==1 & signal[t]==1)
      hold[t]<-0
    if(hold[t-1]==1 & signal[t]==0)
      hold[t]<-1
    if(hold[t-1]==1 & signal[t]==(-1))
      hold[t]<-2
    if(hold[t-1]==(-1) & signal[t]==1)
      hold[t]<-(-1)
    if(hold[t-1]==(-1) & signal[t]==0)
      hold[t]<-(-1)
    if(hold[t-1]==(-1) & signal[t]==(-1))
      hold[t]<-0
    if(hold[t-1]==2 & signal[t]==1)
      hold[t]<-0
    if(hold[t-1]==2 & signal[t]==0)
```

```

        hold[t]<-2
        if(hold[t-1]==2 & signal[t]==(-1))
            hold[t]<-2
    }
    return(hold)
}

hold<-Hold(trade)
hold0<-xts(hold,order.by=index(trade))

##定义交易模拟函数
TradeSim<-function(price,hold){
    n<-length(price)
    positi<-xts(rep(0,n),order.by=index(price))
    names(positi)<-"position"
    positi[index(hold)]<-as.numeric(hold)
    position<-as.numeric(positi)
    asset<-rep(100000,n)
    numprc<-as.numeric(price)
    stock<-rep(0,n)
    cash<-rep(100000,n)
    for (t in (2:n)){
        #从 0 变成 1， 开仓
        if(position[t-1]==0 & position[t]==1){
            cash[t]<-0
            stock[t]<-cash[t-1]
            asset[t]<-cash[t]+stock[t]
        }
        #平仓
        if(position[t-1]==1 & position[t]==0){
            cash[t]<-stock[t-1]*(1+ret[t])*(1-0.0003)
            stock[t]<-0
            asset[t]<-cash[t]+stock[t]
        }
        #满仓,信号 0
        if(position[t-1]==1 & position[t]==1){
            cash[t]<-0
            stock[t]<-stock[t-1]*(1+ret[t])
            asset[t]<-cash[t]+stock[t]
        }
        #空仓， 信号 0
        if(position[t-1]==0 & position[t]==0){
            cash[t]<-cash[t-1]
            stock[t]<-stock[t-1]
        }
    }
}

```

```

        asset[t]<-cash[t]+stock[t]

    }
    #空仓做空
    if(position[t-1]==0 & position[t]==(-1)){
        cash[t]<-cash[t-1]*2*(1-0.0010)
        stock[t]<-(-cash[t-1])
        asset[t]<-cash[t]+stock[t]
    }
    #满仓借钱融资
    if(position[t-1]==1 & position[t]==2){
        cash[t]<-(-stock[t-1]*(1+ret[t]))
        stock[t]<-stock[t-1]*(1+ret[t])*2
        asset[t]<-cash[t]+stock[t]
    }
    #卖空状态时，信号 0 或-1
    if(position[t-1]==(-1) & position[t]==(-1)){
        cash[t]<-cash[t-1]*(1-0.0003)
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-cash[t]+stock[t]
    }
    #卖空状态时，信号 1
    if(position[t-1]==(-1) & position[t]==0){
        cash[t]<-(-cash[t-1]+stock[t-1]*(1+ret[t]))*(1-0.0003)
        stock[t]<-0
        asset[t]<-cash[t]+stock[t]
    }
    #借钱融资，信号 0 或 1
    if(position[t-1]==2 & position[t]==2){
        cash[t]<-cash[t-1]
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-stock[t]+cash[t]
    }
    #借钱融资，信号-1
    if(position[t-1]==2 & position[t]==0){
        cash[t]<-cash[t-1]+stock[t-1]*(1+ret[t])*(1-0.0003)
        stock[t]<-0
        asset[t]<-stock[t]+cash[t]
    }
}
# asset<-cash+price*position*1000
# cash<-xts(cash,order.by=index(price))

```

```

    asset<-xts(asset,order.by=index(price))
    names(asset)<-"asset"
#   account<-merge(positi,cash,asset)
    account<-merge(positi,asset)
    return(account)
}
TradeAccount<-TradeSim(STclose,hold)
head(TradeAccount)
TradeAccount_asset<-TradeAccount[,2]

#绘制账户曲线图
plot.zoo(TradeAccount,type="l",col=c("blue","green","red"),main="MACD , SMA and Double
EMA Combined Trading Strategy(Reverse operation,Allowing borrowing money and stocks)")

####回测,画出回撤图等###
#计算实际收益率
hold0<-xts(hold,order.by=index(trade))
l<-length(hold0)
realtraderet<-Delt(TradeAccount[-l,2],k=1,type="arithmetic")
ret<-Delt(STclose,k=1,type="arithmetic")
#simpletraderet<-hold0*ret
Return<-na.omit(merge(ret,realtraderet))
realtraderet<-na.omit(realtraderet)
ret<-na.omit(ret)
names(ret)<-"stockret"
names(realtraderet)<-"realtraderet"
#realtrade<-rep(0,length(hold0))
#realtraderet<-rep(0,length(hold0))
#Return<-na.omit(merge(ret,realtraderet))
#Return<-rep(0,length(hold0))
#for(t in (2:l)){
#   realtrade[t]<-hold[t]-hold[t-1]
#   realtraderet[t]<-ret[t]*realtrade[t]
#   Return<-na.omit(merge(ret[t],realtraderet[t]))

#}
#head(realtraderet)
library(PerformanceAnalytics)
charts.PerformanceSummary(cbind(ret,realtraderet),lty=c(1,4),main="Combined Strategy
performance")

####回测计算夏普比率, 最大回撤等
backtest<-function(ret,realtraderet){
  performance<-function(x){

```

```

    winpct=length(x[x>0])/length(x[x!=0])
    annRet=Return.annualized(x,scale=252)
    sharpe=SharpeRatio.annualized(x,scale=252)
    DDs<-findDrawdowns(x)
    maxDD<-min(DDs$return)
    perfo<-c(winpct,annRet,sharpe,maxDD)
    names(perfo)<-c("win rate","annualized return",
                  "annualized sharpe ratio","maximum drawdown")
    return(perfo)
  }
  cbind(BuyAndHold=performance(ret),RealTrade=performance(realtraderet))
}
backtest(ret["2011-01-05"],realtraderet["/2016-12-02"])

```

```

#2016 return
realtraderet2016<-realtraderet["2016"]
mean(realtraderet2016)
#2015 return
realtraderet2015<-realtraderet["2015"]
mean(realtraderet2015)
#2014 return
realtraderet2014<-realtraderet["2014"]
mean(realtraderet2014)
#2013 return
realtraderet2013<-realtraderet["2013"]
mean(realtraderet2013)
#2012 return
realtraderet2012<-realtraderet["2012"]
mean(realtraderet2012)
#2011 return
realtraderet2011<-realtraderet["2011"]
mean(realtraderet2011)

```

```

#计算空头胜率，多头胜率，交易次数
openshort<-0
openlong<-0
closeshort<-0
closelong<-0
closelong2<-0
n<-length(hold)
for (t in (2:n)){
  if(hold[t-1]==0&hold[t]==-1){
    openshort=openshort+1
  }
}

```

```

if(hold[t-1]==0&hold[t]==1){
  openlong=openlong+1
}
if(hold[t-1]==1&hold[t]==0){
  closelong=closelong+1
}
if(hold[t-1]==2&hold[t]==0){
  closelong2=closelong2+1
}
if(hold[t-1]==-1&hold[t]==0){
  closeshort=closeshort+1
}
}

```

```

openshort_2<-0
openlong_2<-0
closeshort_2<-0
closelong_2<-0
closelong2_2<-0
n<-length(hold)
for (t in (2:n)){
  if(hold[t-1]==0&hold[t]==-1&(TradeAccount_asset[t-1]<TradeAccount_asset[t])){
    openshort_2=openshort_2+1
  }
  if(hold[t-1]==0&hold[t]==1&(TradeAccount_asset[t-1]<TradeAccount_asset[t])){
    openlong_2=openlong_2+1
  }
  if(hold[t-1]==1&hold[t]==0&(TradeAccount_asset[t-1]<TradeAccount_asset[t])){
    closelong_2=closelong_2+1
  }
  if(hold[t-1]==2&hold[t]==0&(TradeAccount_asset[t-1]<TradeAccount_asset[t])){
    closelong2_2=closelong2_2+1
  }
  if(hold[t-1]==-1&hold[t]==0&(TradeAccount_asset[t-1]<TradeAccount_asset[t])){
    closeshort_2=closeshort_2+1
  }
}
}

```

###根据交易信号构造交易状态函数，制定交易状态时，假设股票只在空仓时才卖空且算交易费用

```
Hold<-function(signal){  
  n<-length(signal)  
  hold<-rep(0,n)  
  for (t in (2:n)){  
    if(hold[t-1]==0 & signal[t]==1)  
      hold[t]<-1  
    if(hold[t-1]==0 & signal[t]==(-1))  
      hold[t]<-(-1)  
    if(hold[t-1]==0 & signal[t]==0)  
      hold[t]<-0  
    if(hold[t-1]==1 & signal[t]==1)  
      hold[t]<-1  
    if(hold[t-1]==1 & signal[t]==0)  
      hold[t]<-1  
    if(hold[t-1]==1 & signal[t]==(-1))  
      hold[t]<-0  
    if(hold[t-1]==(-1) & signal[t]==0)  
      hold[t]<-(-1)  
    if(hold[t-1]==(-1) & signal[t]==(-1))  
      hold[t]<-(-1)  
    if(hold[t-1]==(-1) & signal[t]==1)  
      hold[t]<-0  
  }  
  return(hold)  
}
```

```
hold<-Hold(trade)
```

##定义交易模拟函数


```

TradeSim<-function(price,hold){
  n<-length(price)
  positi<-xts(rep(0,n),order.by=index(price))
  names(positi)<-"position"
  positi[index(hold)]<-as.numeric(hold)
  position<-as.numeric(positi)
  asset<-rep(100000,n)
  numprc<-as.numeric(price)
  stock<-rep(0,n)
  cash<-rep(100000,n)
  for (t in (2:n)){
    #从 0 变成 1， 开仓
    if(position[t-1]==0 & position[t]==1){
      cash[t]<-0
      stock[t]<-cash[t-1]
      asset[t]<-cash[t]+stock[t]
    }
    #平仓
    if(position[t-1]==1 & position[t]==0){
      cash[t]<-stock[t-1]*(1+ret[t])*(1-0.0003)
      stock[t]<-0
      asset[t]<-cash[t]+stock[t]
    }
    #满仓,信号 0
    if(position[t-1]==1 & position[t]==1){
      cash[t]<-0
      stock[t]<-stock[t-1]*(1+ret[t])
      asset[t]<-cash[t]+stock[t]
    }
    #空仓， 信号 0
    if(position[t-1]==0 & position[t]==0){
      cash[t]<-cash[t-1]
      stock[t]<-stock[t-1]
      asset[t]<-cash[t]+stock[t]
    }
    #空仓做空
    if(position[t-1]==0 & position[t]==(-1)){
      cash[t]<-cash[t-1]*2*(1-0.0010)
      stock[t]<-(-cash[t-1])
      asset[t]<-cash[t]+stock[t]
    }
    #卖空状态时， 信号 0 或-1
    if(position[t-1]==(-1) & position[t]==(-1)){

```

```

        cash[t]<-cash[t-1]
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-cash[t]+stock[t]
    }
    #卖空状态时，信号 1
    if(position[t-1]==(-1) & position[t]==0){
        cash[t]<-cash[t-1]+stock[t-1]*(1+ret[t])*(1-0.0003)
        stock[t]<-0
        asset[t]<-cash[t]+stock[t]
    }
}
# asset<-cash+price*position*1000
# cash<-xts(cash,order.by=index(price))
asset<-xts(asset,order.by=index(price))
names(asset)<-"asset"
# account<-merge(positi,cash,asset)
account<-merge(positi,asset)
return(account)
}
TradeAccount<-TradeSim(STclose,hold)
head(TradeAccount)

#绘制账户曲线图
plot.zoo(TradeAccount,type="l",col=c("blue","green","red"),main="MACD , SMA and Double
EMA Combined Trading Strategy(Only allowing borrowing stocks)")

```

###根据交易信号构造交易状态函数，制定交易状态时，假设股票在卖出时均卖空，满仓时借钱买进且算交易费用

```

Hold<-function(signal){
    n<-length(signal)
    hold<-rep(0,n)
    for (t in (2:n)){
        if(hold[t-1]==0 & signal[t]==1)
            hold[t]<-1
        if(hold[t-1]==0 & signal[t]==(-1))
            hold[t]<-(-1)
    }
}

```

```

    if(hold[t-1]==0 & signal[t]==0)
      hold[t]<-0
    if(hold[t-1]==1 & signal[t]==1)
      hold[t]<-2
    if(hold[t-1]==1 & signal[t]==0)
      hold[t]<-1
    if(hold[t-1]==1 & signal[t]==(-1))
      hold[t]<-0
    if(hold[t-1]==(-1) & signal[t]==1)
      hold[t]<-0
    if(hold[t-1]==(-1) & signal[t]==0)
      hold[t]<-(-1)
    if(hold[t-1]==(-1) & signal[t]==(-1))
      hold[t]<-(-1)
    if(hold[t-1]==2 & signal[t]==(-1))
      hold[t]<-0
    if(hold[t-1]==2 & signal[t]==0)
      hold[t]<-2
    if(hold[t-1]==2 & signal[t]==1)
      hold[t]<-2

  }
  return(hold)
}

```

```
hold<-Hold(trade)
```

##定义交易模拟函数

```

TradeSim<-function(price,hold){
  n<-length(price)
  positi<-xts(rep(0,n),order.by=index(price))
  names(positi)<-"position"
  positi[index(hold)]<-as.numeric(hold)
  position<-as.numeric(positi)
  asset<-rep(100000,n)
  numprc<-as.numeric(price)
  stock<-rep(0,n)
  cash<-rep(100000,n)
  for (t in (2:n)){
    #从 0 变成 1， 开仓
    if(position[t-1]==0 & position[t]==1){
      cash[t]<-0
      stock[t]<-cash[t-1]
    }
  }
}

```

```

        asset[t]<-cash[t]+stock[t]
    }
#平仓
    if(position[t-1]==1 & position[t]==0){
        cash[t]<-stock[t-1]*(1+ret[t])*(1-0.0003)
        stock[t]<-0
        asset[t]<-cash[t]+stock[t]
    }
#满仓,信号 0
    if(position[t-1]==1 & position[t]==1){
        cash[t]<-0
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-cash[t]+stock[t]
    }
#空仓, 信号 0
    if(position[t-1]==0 & position[t]==0){
        cash[t]<-cash[t-1]
        stock[t]<-stock[t-1]
        asset[t]<-cash[t]+stock[t]
    }
}
#空仓做空
    if(position[t-1]==0 & position[t]==(-1)){
        cash[t]<-cash[t-1]*2*(1-0.0010)
        stock[t]<-(-cash[t-1])
        asset[t]<-cash[t]+stock[t]
    }
}
#满仓借钱融资
    if(position[t-1]==1 & position[t]==2){
        cash[t]<-(-stock[t-1]*(1+ret[t]))
        stock[t]<-stock[t-1]*(1+ret[t])*2
        asset[t]<-cash[t]+stock[t]
    }
}
#卖空状态时, 信号 0 或-1
    if(position[t-1]==(-1) & position[t]==(-1)){
        cash[t]<-cash[t-1]
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-cash[t]+stock[t]
    }
}
#卖空状态时, 信号 1
    if(position[t-1]==(-1) & position[t]==0){
        cash[t]<-cash[t-1]+stock[t-1]*(1+ret[t])
        stock[t]<-0
        asset[t]<-cash[t]+stock[t]
    }
}

```

```

    }
    #借钱融资， 信号 0 或 1
    if(position[t-1]==2 & position[t]==2){
        cash[t]<-cash[t-1]
        stock[t]<-stock[t-1]*(1+ret[t])
        asset[t]<-stock[t]+cash[t]

    }
    #借钱融资， 信号-1
    if(position[t-1]==2 & position[t]==0){
        cash[t]<-cash[t-1]+stock[t-1]*(1+ret[t])*(1-0.0003)
        stock[t]<-0
        asset[t]<-stock[t]+cash[t]
    }
}
# asset<-cash+price*position*1000
# cash<-xts(cash,order.by=index(price))
asset<-xts(asset,order.by=index(price))
names(asset)<-"asset"
# account<-merge(positi,cash,asset)
account<-merge(positi,asset)
return(account)
}
TradeAccount<-TradeSim(STclose,hold)
head(TradeAccount)

#绘制账户曲线图
plot.zoo(TradeAccount,type="l",col=c("blue","green","red"),main="MACD , SMA and Double
EMA Combined Trading Strategy(Allowing borrowing money and stocks)")

```

