目录

任意给定投资组合下最优量化策略的数据挖掘 1
一、 文献综述与研究意义 3
二、 选择数据源 4
三、 数据清洗 7
3.1 投资组合有效性检验 7
3.2CAPM 回归检验 8
四、 构建分类器 12
五、 实际回测结果 14
六、 总结 15
【参考文献】 15
1.R 代码节选 16
2.Python 代码节选 23

一、 文献综述与研究意义

如何对金融收益建模一直以来是金融研究的一个长期问题。在一些经典的金融研究当中，对收益率所构建的模型主要有 Markowitz（1952）方差-均值分析，CAPM 模型（Sharpe,1964）模型随机游走模型（Fama,1970），GARCH 族模型（Bollerslev&Engle,1982）,分形模型（Mandelbrot,1991）等。这些模型各有优劣，也各有一定的适用范围。比如 GARCH 族模型主要用于风险管理，而方差-均值分析是构建最小方差投资组合的通用原理。据咨询机构（BCG，2005）调研，美国基金机构最常用的收益率模型是 CAPM，其次是一般 AR 模型。

量化策略，是指根据对收益率所构建的模型而形成的一系列交易规则。量化策略是否有必要，或者说能否给投资者带来正的 Alpha 值[1]，一般有两种观点。

第一种观点是市场有效论，这意味着任何量化策略理论上不能优于买入持有（BH）的消极投资策略。这一观点是由 Fama 在 1970 年提出，后来发展完善成为有效市场理论。这一理论的简单描述是：定义P_t为某一资产在t时刻的价格，称随机序列$\{P_t\}$服从几何布朗运动，若：

$$\log P_t = \log P_{t-1} + Y_t \quad (1.1)$$

其中Y_t是独立同分布的随机序列，近似等于该资产在t时期的收益率。因为Y_t独立同分布，所以过去的收益不能用于预测未来的收益，因此对时间序列建模是无意义的。这一理论影响深远。在美国，日本，欧洲等发达的资本市场，这一理论已经得到了充分的验证。并且由此理论外推出来的结论也深刻影响了其他收益率模型，例如：

根据 CAPM 理论可以建立实证的回归方程：$R_i = \alpha + \beta_i R_M + \varepsilon_i$，其中是$R_i$个股的超额收益率，$R_M$是指数的超额收益。在$R_M$能很好表现出市场系统性风险的情况下，α通过实证回归一般结果为 0，间接佐证了市场有效论，即在市场上不存在超额收益高于其所承担的系统性风险的资产。

第二种观点是市场弱有效论。这主要是因为R_M在一些新兴市场并不能完整体现系统性风险。这一观点的典型模型是 Fama 和 French 在 1992 年提出的多因素模型：

$$R_i = a_i + \sum_{i=1}^{n} \beta_i F_i + \varepsilon_i \quad （1.2）$$

其中F_i是第 i 种因素，而不单单是市场指数的超额收益。但是由于这个模型中，F_i因素的选取没有一个固定的标准，因此通过这个模型研究得出的结论实际上没有普遍性。而这也体现出市场是弱有效的：**通过挖掘市场上当前已有的信息来构建量化策略，投资者可能实现超额回报。**

在这一理论指导下，许多学者也构建了很多模型。使用最多的模型是线性时间模型，比如 ARIMA 模型。但是陈诗一在 2007 年论文提出：金融序列的变动性具有聚类效应，不能满足正态分布的假设，这就使得利用 MLE 方法估计的线性时间序列模型不具有很好的样本外预测性。而国外学者从对立面也证明了 ANN 与 SVM 等非参数的方法要优于线性时间序列模型（Moshiri&Cameron,2000）。

前人关于非参数机器学习的方法在股票预测上文献并不多。方匡南 2010 年提出了单独运用随机森林来预测基金超额收益的方法，但是由于该文只研究了一只特殊的基金和运用随机森林与 SVM 两种方法，得出的结论难以推广。此外，他的研究重在比较非参数机器学习的方法和时间序列方法建模的优劣，得出的结论与前人研究类似，因此本文将不在针对时间

[1] 根据 CAPM 理论，α一般假定趋于 0。因此寻找正的 Alpha 本质上就是寻找表现优于市场能提供的回报率的投资组合。

序列进行建模，转而在更好的机器学习方法之中挖掘最好的量化策略。除此以外本文还将在一定程度沿用他的变量选择。

此外，本文研究的是任意给定投资组合的情况下来寻找最优的量化策略。这一点与实务当中，直接去寻找最优的投资组合是不同的。之所以这么做，是因为选择投资组合这个行为并无数据可供挖掘。

本文接下来将分以下几个部分来叙述如何构建任意给定投资组合下最优的量化策略。第二部分是叙述如何选择有代表性投资组合与变量作为数据源。第三部分数据预处理与验证市场是否有效和构建买入持有最小方差投资组合的量化策略（BH）。第四部分是构建非参数分类器，主要有线性集成分类器（LCM），非线性集成分类器（NLCM），与综合分类器（CCM）。再根据这三个分类器构建出各自的量化策略。第五部分是比较这四种量化策略在现实中的投资收益。第六部分是总结。

二、 选择数据源

一般金融理论认为：价格收益率序列的一阶矩是对这个金融资产收益率的估计，二阶中心矩是对金融资产非系统性风险的估计。CAPM 中β系数是该金融资产的系统性风险的估计。该金融资产的收益率与系统性风险成正相关。非系统性风险可以通过分散投资来规避，因此承担这部分风险不会带来收益率的上升。

由于本文并不对选择什么样投资组合感兴趣，而是对给定某一特定的投资组合如何构建量化策略感兴趣，因此不同于之前的文献对金融资产在β系数与收益率两个层次上的聚类方法，本文选择在金融资产本身的收益率和标准差进行聚类，这样进行聚类更能体现个个金融资产之间的类别差异。此外选择标准差而不是方差是为了减小离群值对聚类带来的影响。

标的资产本文选择的是中国 A 股，原因是前人证明了在欧美日等发达资本市场中消极投资策略是有效的，因此不具有挖掘价值，而中国是新兴资本市场，市场有效性较弱，挖掘空间较大。其次是因为数据的可得性。

综合上述，本文先通过 Python 在 tushare 第三方财经库中爬取了**中国 A 股 2015 年 1 月 1 日到 2017 年 5 月 31 日的日行情数据**，并计算出了每一只股票在这段时间收益率和标准差。之所以选择这段时间，是因为中国股市在这段时期内经历了一次较为完整的涨跌周期，因此这么做可以使聚类具有代表性。**一共有 3545 条记录**，一条记录代表一只股票的收益率和标准差。下图是总体的均值标准差散点图。

可以看出总体收益率在 0 处分布有对称性，体现了这段时期涨跌的周期性。个股非系统性风险（标准差）分散。

接着本文在这 3545 条记录中随机**选取 100 只股票进行聚类**。能这么做的原因有两个：第一，由整体的散点图对称分布的趋势可知随机抽样具有代表性。第二，节约聚类的计算资源。下图是所选 100 只股票的均值标准差散点图。聚类方法选择是层次聚类法，一共聚得 6 类。下图是可视化结果。

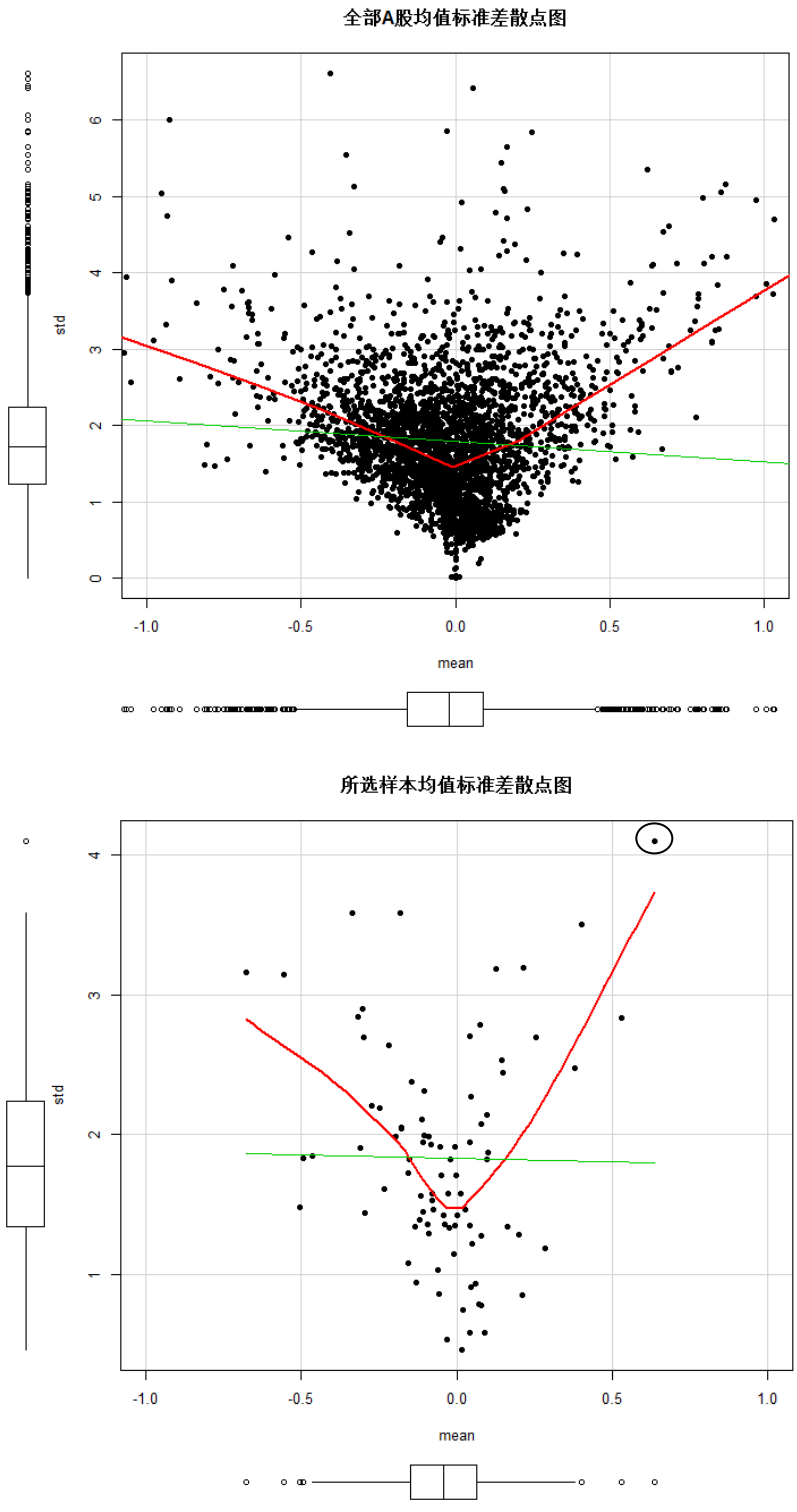


图 2.1 均值方差散点图总体 Vs 样本

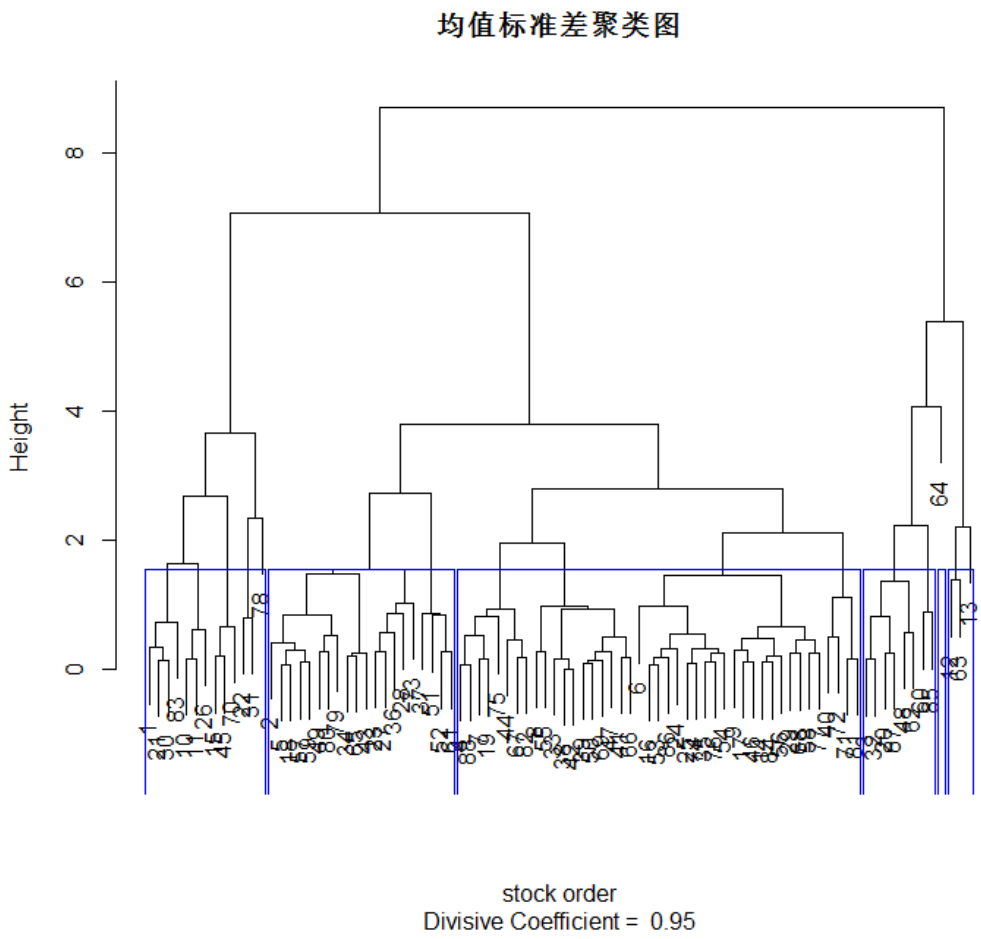


图 2.2 均值标准差聚类图

聚合的这 6 类，每一类代表具有类似风险与收益的股票的集合。编号 64 的股票只有一类，在图中表现为东北角的那一个点，其代码是 000411（英特集团）。在东北角意味着高非系统性风险，高收益。由于高的非系统性风险可以通过分散化投资组合进行规避，因此可以预见这一只股票在投资组合中占有重要地位。

剩下 5 类每一类随机抽选一只股票作为代表，于是得到了以 6 只股票作为给定投资组合的代表性投资组合。这六只股票分别是 000009（中国宝安，以下简称 ba），000062（深圳华强，以下简称 hq），000090（天健集团，以下简称 tj），000158（常山股份，以下简称 cs），000411（英特集团，以下简称 it），000491（粤高速 A，以下简称 gs）。此外沿用惯例选取 00001（上证指数）作为市场收益。

确定投资组合之后，需要确定这个投资组合的时间序列数据大小。综合数据可得性，平稳性和代表性，**本文选取了 2003 年 6 月份到 2017 年 5 月份的月价格序列数据，每只股票的记录数在 162 条到 168 条之间。数据来源是 Wind 金融终端**[2]。

因为投资组合本身怎么选取的并不是本文关注的重点，因此以上步骤并不具有金融意义，其目的仅仅是让选取的投资组合能尽可能具有普遍性。

接下来是选取能够刻画这些股票不同特征的分类自变量。此处沿用了方匡南 2010 年论文中选取的三个变量，股票收益的一阶滞后项（记为 lag_rtn），指数收益的一阶滞后项（记为 lag_index），成交量变化率的一阶滞后项（记为 lag_Q）。在他的文章中，还选取了折价率

[2] 附件将提供完整的初始数据

这个变量，以代理市值变化这个维度。但是考虑到那篇文章研究对象是基金，折价率在起初就可以通过定价公式导出，是一个较为平稳的变量，具有很强的分类意义；而本文选取的对象是股票，其折价率是一个不平稳的动态过程，分类意义可能不强。其次股票行情数据库中并不设有这个变量，若要一期一期递推导出，计算量庞大。最后，股票本身具有更直接的代理变量来代表市值变化，选取这个变量效率和效果都将更好。**因此此处选择的是股票流通市值变化率的一阶滞后项（记为 lag_M）。**最后沿用惯例，选用投资回报率的一阶滞后项（记为 lag_ROI）作为股票基本面的代理变量。

因变量是股票的当期收益率。这本身是一个连续变量，但由于本文构建最优投资组合并不对预测涨跌的幅度感兴趣，而是对涨跌的方向感兴趣。对于预测方向为正的股票将以（1-预测错判率）为权重购入，对于预测方向为负的股票将以（1-预测错判率）为权重购买指数基金代替。因此将当期收益率的方向编码为哑变量作为因变量是合理的。下表是变量的总结。

表 2.1 变量一览表

因变量	**名称**	**解释**
当期收益率涨跌方向（2 类）	上涨（1） 下跌（2）	Coding as Factor(1 and 2)
自变量	名称	解释
股票收益的一阶滞后项	Lag_rtn	行情收益的代理
股票成交率的一阶滞后项	Lag_Q	行情成交量的代理
流通市值变化率的一阶滞后项	Lag_M	市值的代理
投资回报率的一阶滞后项	Lag_ROI	公司基本面的代理
市场收益的一阶滞后项	Lag_index	市场投资组合的代理

至此，数据源和变量定义完成。但是进行构建分类器前还有两个问题：第一，需要检验这六只股票的投资组合是否满足有效性，这是因为如果满足有效性，那么理论上买入持有（BH）最小方差投资组合可能是一种最佳的策略选择。第二，在构建分类器前还需要数据清洗。这就是下一部分叙述的内容。

三、 数据清洗

3.1 投资组合有效性检验

对于投资组合有效性的检验，一般是通过检验 CAPM 的实证模型（$R_i = \alpha + \beta_i R_M + \varepsilon_i$）中的α值t检验完成的。此外也可以通过检验 CAPM 的回归前提，即对因变量R_i进行正态分布的拟合优度检验。但是后者的检验并不常用。原因是：CAPM 的推导过程其实并不是建立在R_i是正态分布的假设上的，检验因变量是否能够满足回归方程的假设实际上是统计学上的方法，不具有金融意义。其次，即使不接受因变量符合正态分布的假设，并不能因此不接受市场是有效的假设。因此用拟合优度的方法实际上放宽了拒绝域，增大了犯第一类错误的概率。但是拟合优度的方法很直观简洁，本文也将先对此进行检验。

下图是分布的可视化，下表分别是 Shapiro-Wilk 正态性检验的结果。

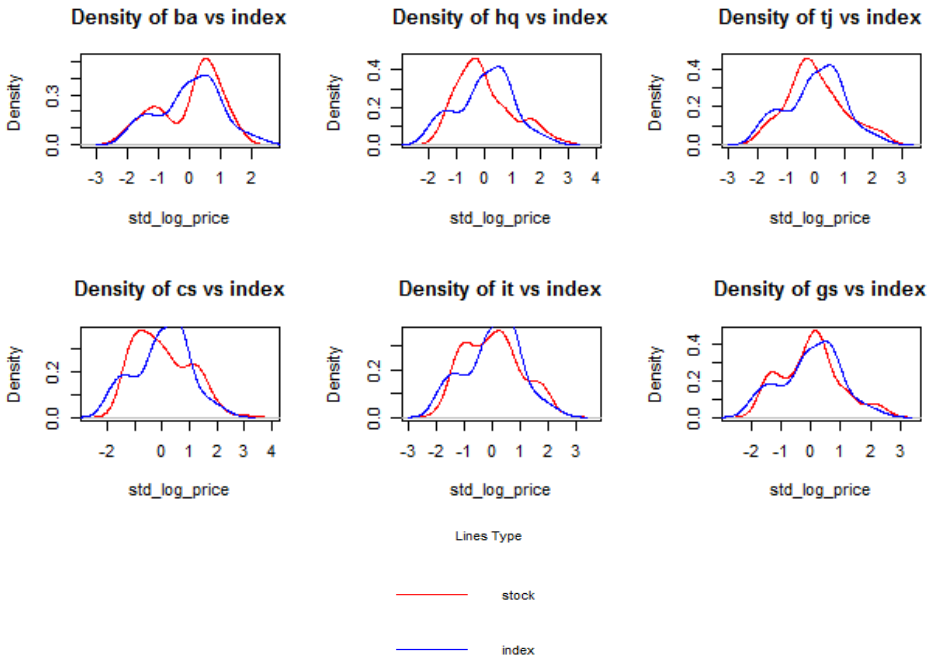


图 3.1.1 分布可视化

表 3.1.1Shapiro-Wilk 正态性检验

Stock_Name	S.W test	p-Value
ba	0.92	0.00***
hq	0.94	0.00***
tj	0.98	0.01**
cs	0.97	0.00***
it	0.97	0.00***
gs	0.97	0.00***
index	0.98	0.01**

从分布图可以看出 hq，tj，cs 的分布位置相对于指数偏左，ba，it，gs 与指数的分布类似，同时具有双峰的性质。正态性检验报告这六只股票的收益率不能接受符合正态分布的原假设。**不过正如前文所说，这个检验结果并不充分。**接下来还将进行 CAPM 的回归检验。

3.2CAPM 回归检验

在进行回归检验前，需要对缺失数据进行插补。由于数据类型是时间序列，若出现连续的数据空缺，用简单插补法显然忽略了趋势，是不成立的。若只是某一条记录出现了缺失值，那么用简单插补法将具有一定的合理性。

基于这个插补原则，本文先做出了缺失值的矩阵表示，下图是缺失值的可视化：

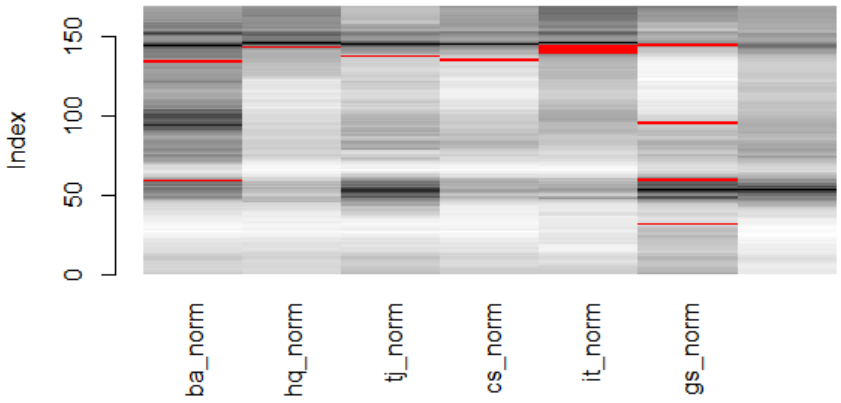


图 3.2.1 缺失数据可视化

从图中可以看出，除了 it 有连续的缺失值以外，其余变量缺失值较为离散。因此其余 5 个变量将采取简单插补法，而 it 则需要进一步研究。

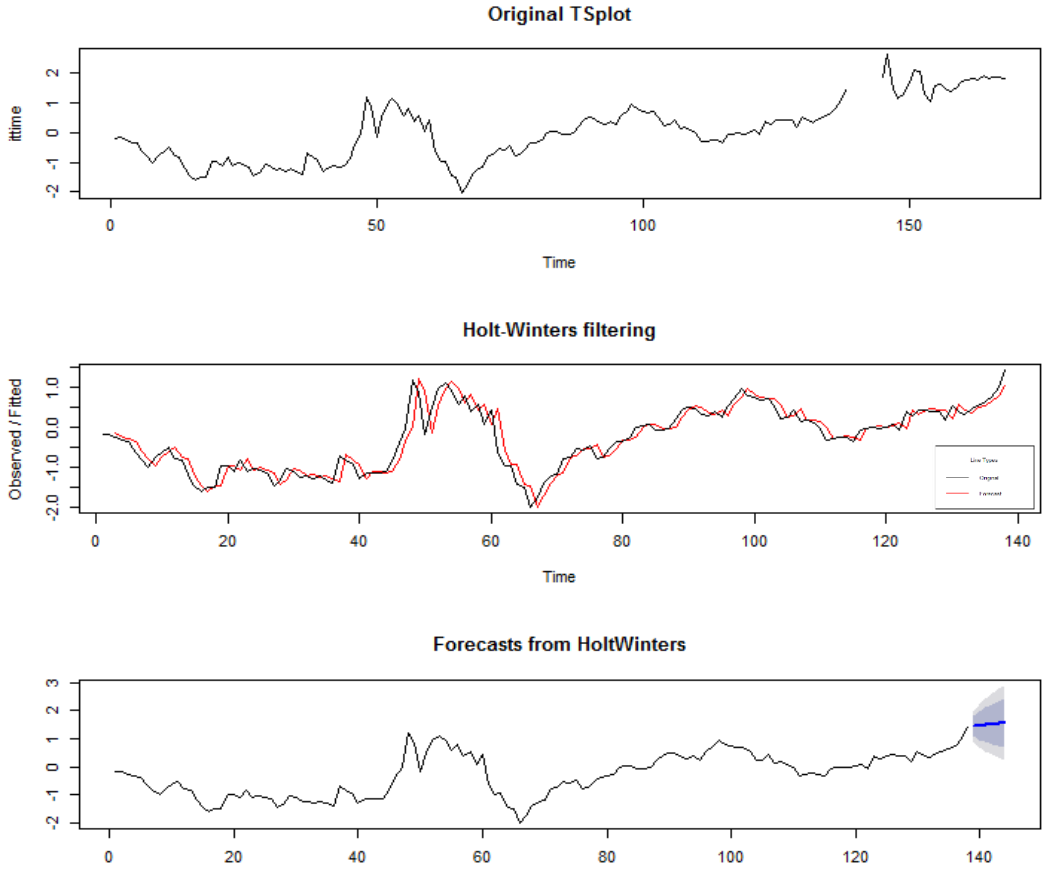


图 3.2.2 时间序列分析

首先先要给缺失值定性。通过查阅 it 的年报，本文发现之所以出现了 6 个月的缺失值

是因为 it 公司在 2015 年初发生重大资产重组而停牌。**此次重组是部分重组，性质定义为利好，重组后趋势保持上升。**

为研究 it 的时间序列特征，本文首先做出了 it 的原始数据的时间序列图，发现原始时间序列图不存在季节性的相加模型，并且在后半段存在明显的上升趋势，**因此本文决定使用霍尔特指数平滑法对缺失时间数据进行预测插补。**

用来预测的数据是前 138 个，预测缺失的后六期数据。首先进行的是样本内预测，图中黑线代表原来数据，红色是预测数据。发现样本内预测稍有延迟，但是预测效果较好（SSE 为 10.3）。

接着本文做出了 80%和 95%置信区间下的样本外预测。

进一步，本文做了预测误差在滞后 1-20 阶的非零自相关 Ljung-Box 检验，P-Value 为 0.107。意味着我们不足以证明预测误差在滞后 1-20 阶是非零自相关的。

此外，为检查整个序列的预测误差是否方差不变，服从零均值的正态分布，本文将残差与零均值的正态分布进行了比较，结果见下图：

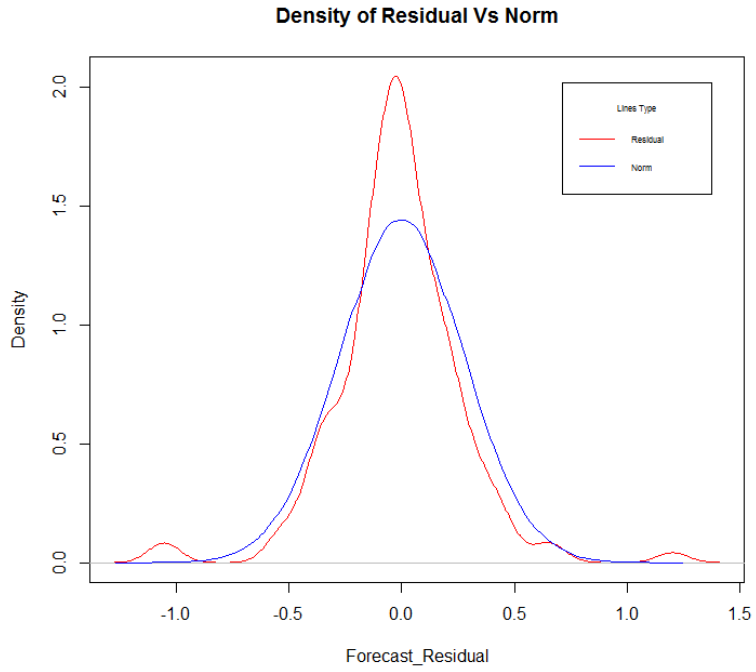


图 3.2.3 残差分析图

我们可以发现预测误差近乎是方差不变零均值的正态分布。由上检验，我们可以推断霍尔特指数平滑法在此处的运用是合理的，且不可再优化的。

根据霍尔特指数平滑法预测得到的数据插补后如下图所示。虚线是补齐的数据。

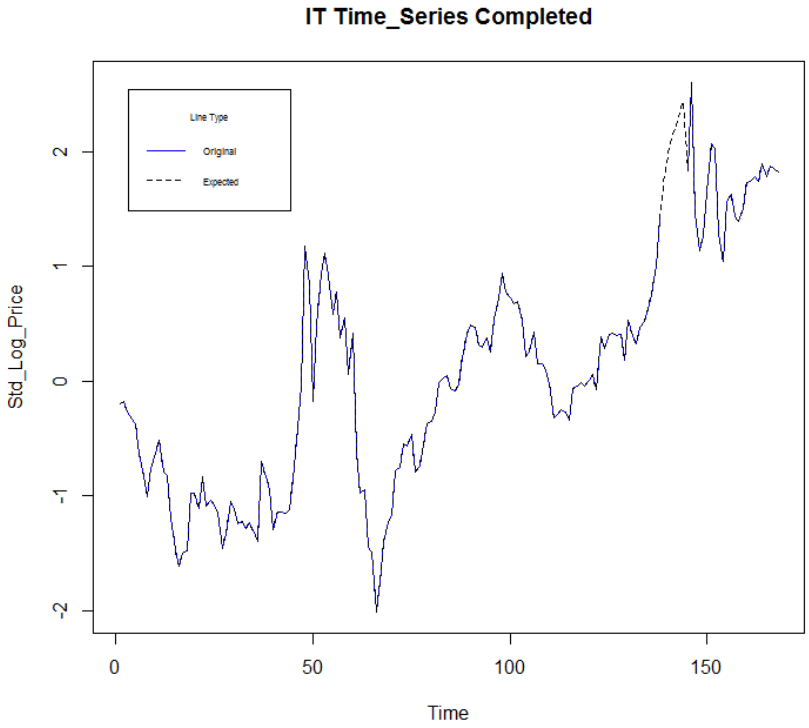


图 3.2.4it 序列插补图

对于其余 5 个变量，本文采取的是均值填补。在补齐数据后，本文开始进行 CAPM 的验证。验证结果如下表：

表 3.2.1CAPM 实证关于 α 的 T test

Alpha	t test	p-Value
ba	0.00	0.99
hq	0.00	0.99
tj	0.00	0.99
cs	0.00	0.99
it	-0.88	0.38
gs	0.00	0.99

通过检验发现，这个投资组合的所有股票都满足有效性。**这也意味这 BH 策略是首先应当考虑的量化策略。**

最小方差组合的思想是，选择一个权重w_i，求解下列线性规划问题：

$$\text{Min } w'Vw$$

$$\text{s.t.} \sum w_i = 1$$

其中 V 是收益率向量的协方差矩阵。

在允许卖空的情况下，这个问题的一般解为:

$$w_i = \frac{V_i^{-1}1}{1'V_i^{-1}1} \quad (3.2.1)$$

但是在此处，中国 A 股并不具有完善的做空机制，靠融券这种方式做空会导致整体规划非线性，建模复杂。**因此此处对做空的处理是：对权重为负的股票，将原权重按比例分配给权重为正的股票。**由此得到 BH 最小方差策略：

表 3.2.2BH 策略权重

股票	允许做空	不允许做空
ba	0.77	0.36
hq	0.56	0.27
tj	-0.21	-
cs	-0.87	-
it	-0.02	-
gs	0.77	0.37

至此数据清洗部分和 BH 策略构建结束，接下来将构建非参数的机器学习分类器。

四、 构建分类器

在构建分类器前，首先要进行数据探索。由于是分类问题，**在数据探索阶段本文重点在于分类变量的探索。**在探索变量本身的分布后，本文做出了自变量之间的相关图。通过相关图发现在 6 只股票当中 Lag_ROI 与 Lag_M 都有很强的**正相关性。**这是因为他们一个是对市场价值的直接描述，一个是对账面价值的直接描述，因此具有很强的正相关性。

而因为它们之间的这种相关性，在处理分类时这两个变量的分类效果应该类似，而事实上通过它们之间在画收益方向上的分层散点图，也能够发现这个性质。

此外，观察这两个变量的分层散点图，以及其他变量的分层散点图，本文发现大多数变量之间其实是线性可分的。这启示了本文在构建分类器上按照分类器的线性性质进行构建。具体来讲：LDA 与 QDA 和线性可分的 SVM 方法可以作为线性分类器（LMC）。Bagging，Boosting，RandomForest，线性不可分的 SVM 方法与 ANN 可作为非线性分类器（NLMC）。单个分类器的效果被证明在一般情况下不如分类器的组合（Jiawei Han，2012），因此本文决定采取单个分类器投票集成的方法构造组合分类器。**投票的权重是（1-误判率）。依此三个模型构造的量化策略是：对于预测方向为正的股票将以（1-预测错判率）为权重购入，对于预测方向为负的股票将以（1-预测错判率）为权重购买指数基金代替。**

最后为比较线性分类器与非线性分类器的差异，本文增加了将所有分类器集成的综合分类器（CCM）。由于 R 在分类器集成上没有合适的包，本文这一部分的操作是在 Python 上进行的。

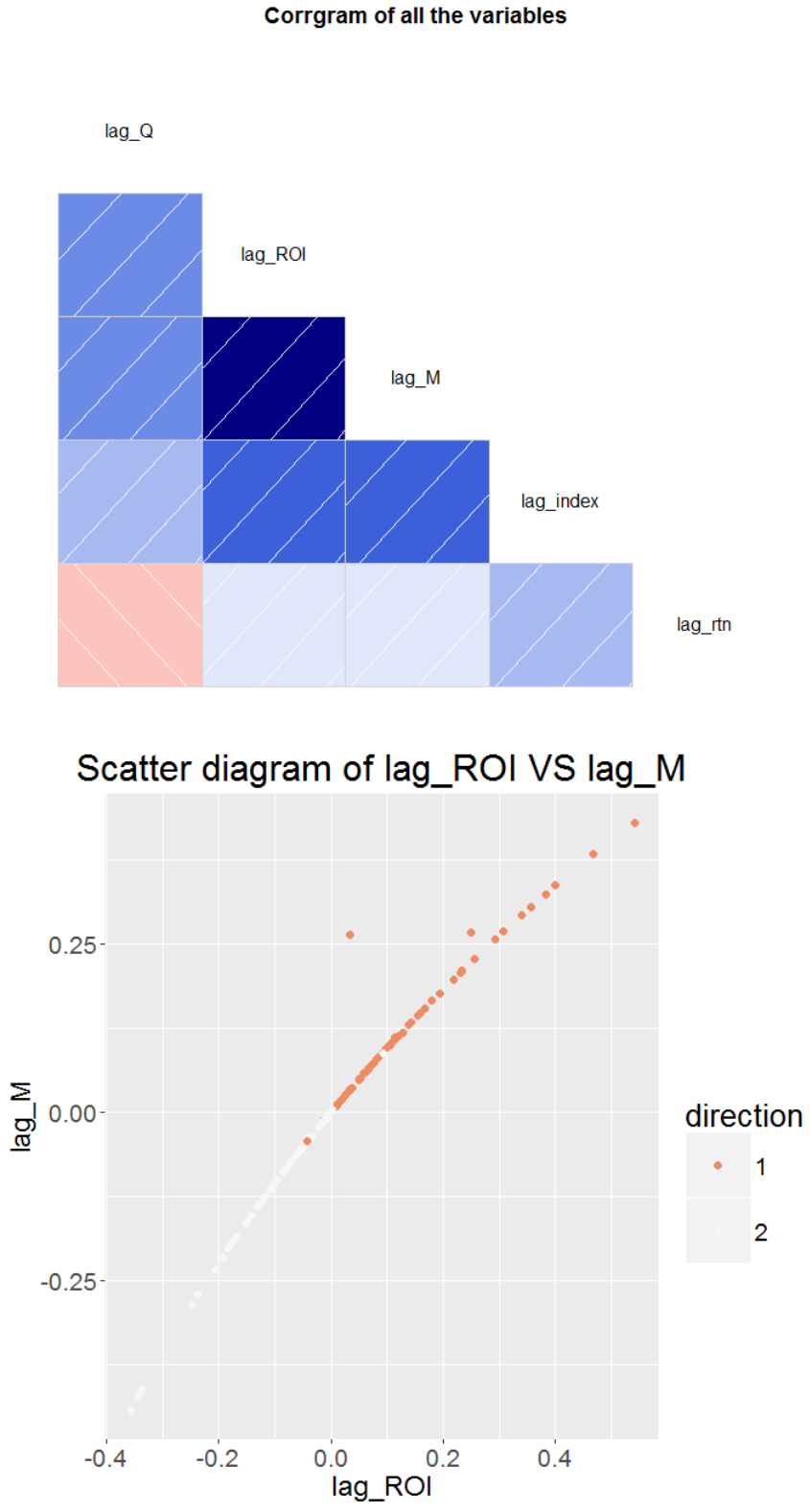
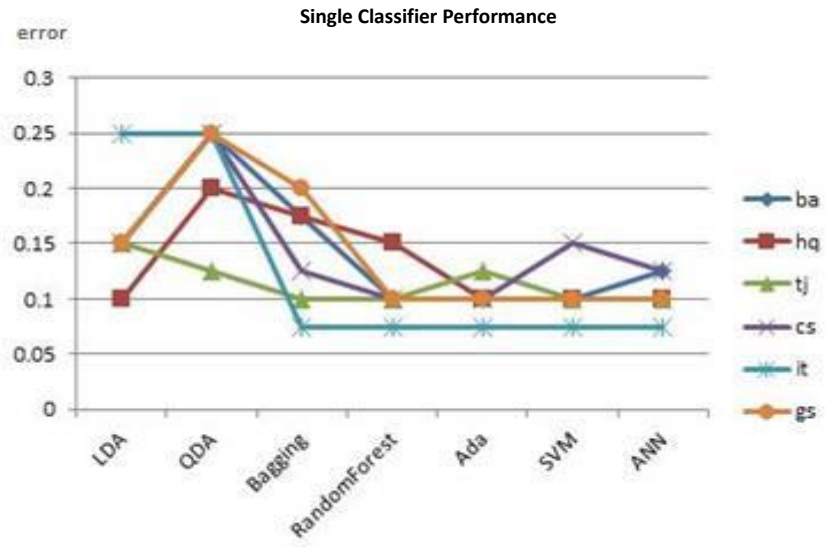


图 4.1 数据探索（以 ba 为例）

在构建单个分类器的时候，由于样本量不够大，**只选择 40 个样本作为测试集。**下图是所有单个分类器在 6 只股票上预测错误率。



4.2 单个分类器的表现

发现在单个分类器中，ANN 与 Adaboosting 表现稳定，LDA，QDA 起伏较大。在各个股票当中，hq，it 和 gs 运用非线性分类器表现要优于线性分类器。**可能的解释是具有明显趋势下线性分类方法识别度不高。**此外由于样本量不大，**不同 SVM 的核函数的选取对分类差异较小。值得一提的是 RandomForest 方法不一定像方匡南文章结论那样明显占优，但不能够充分反驳其文 RandomForest 是较为出色的分类器的结论。**

至此所有的量化策略已经构建完成。接下来的部分是回测。

五、 实际回测结果

回测需要选取合适的回测周期：既要能够体现普遍性，又不能太短。综合考虑本文选取了最近 6 个月，12 个月和 18 个月作为回测周期。下图是四种策略的表现：

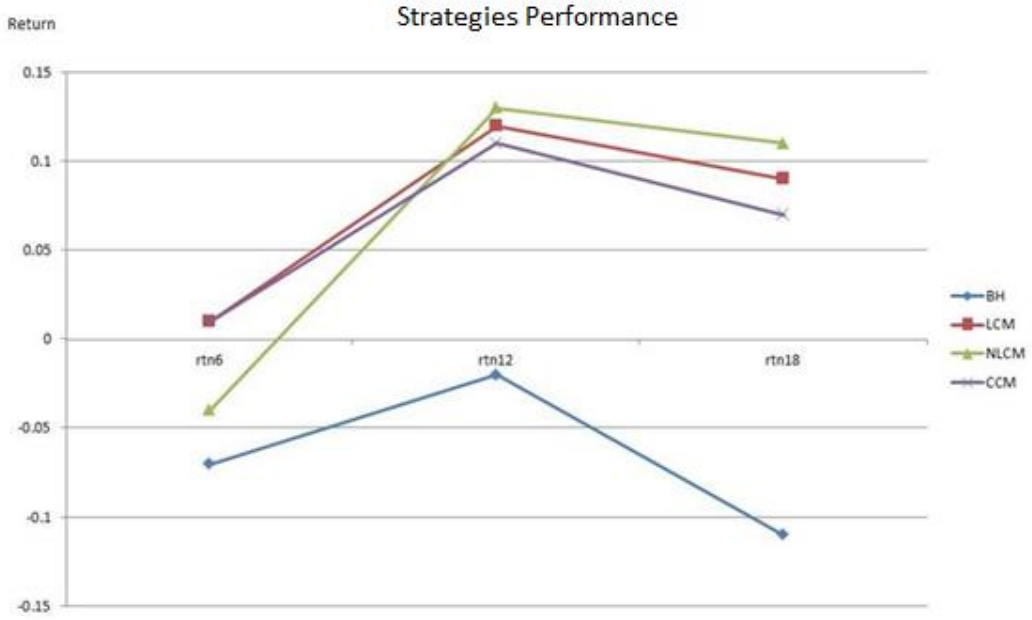


图 5.1 量化策略比较

由图中可以看出，BH 策略表现最差，18 月的累计收益率达到了-11%之多。在 12 个月的周期下，NLCM 策略带来了最大收益，约为 13%。但同时它在 6 个月表现不如 CCM 与 LCM。LCM 和 CCM 表现较为稳定，但是 CCM 严格劣于 LCM。

对这个结果进行进一步分析，发现前六个月方向预测分歧较大的在 it 与 ba 上：前六个月整体环境走低，前 4 个月投资组合的所有股票全部下跌。但是在第五个月 it 与 ba 出现回调，it 上涨了 33.4%，ba 上涨了 22.7%，指数反而下跌了 2.1%。而在第四期决策中，NLCM 对 ba 和 it 的预测符号为反，因此购入了指数，而 CCM 与 LCM 预测符号为正，并根据最小方差组合分别购入了 32%与 27%权重的股票，**因此在这一期决策中 CCM，LCM 表现超过了 NLCM 许多。**

六到十二个月的周期中，分歧较大的在 tj 上：在此期间指数涨幅为 10.9%，但是 tj 下跌了 18%。NCLM 在此期间预测 tj 方向全对，购入指数获得了较好成绩，**而 LCM 滞后了 1 期开始购买指数，CCM 滞后了 2 期。**

最后一个周期中，NLCM 与 LCM 差异不大，而指数在此阶段微涨盘整，涨 0.4%，**这样子意味着即使预测错误的机会成本也不会很大。**

BH 策略之所以表现糟糕，是因为 ba 后期暴跌，整个周期内累计下跌 58.0%。hq 同样在整个持有期内收跌，幅度为-37.4%。**这说明即使通过了有效性检验的投资组合在整体市场有效性弱的情况下如果一直被持有，那么风险也是很大的。**

综合上述，非参数机器学习的分类器在给定的投资组合上表现出色。其中 NLCM 在预测趋势上更为准确，而 LCM 表现更为稳定，CCM 表现平庸。目前尚无文献得出类似结论。

六、 总结

本文的新颖之处在于在方匡南文章的基础上探索了非参数分类方法中分类器的优劣，具体变现为通过分类器合成的方法比较了线性和非线性分类器的差别，**发现在量化策略中具有明显趋势下线性分类方法识别度不高这一结论。**其次本文较为完整地展示了数据挖掘的过程和结果，对以后类似文章提供借鉴。

但是本文在以下方面还有明显的可进一步发展的空间：**第一，本文对是否存在卖空策略的处理较为简单**，这也直接导致了 BH 策略的断崖式差距；**第二本文没有考虑手续费等交易成本，**这个的原因是因为本文认为初始投资金额是不确定的，因此考虑交易费用没有太多意义。对于这一点，本文提供一个今后研究的思路：将收益率一开始全部换成比率而不是通过带入真实金额（**本文带入的金额是 100 万人民币**）再加减算出收益率。但是这样计算量十分庞大，期待今后研究能够设计出一套效率高的程序解决这个问题。**第三，本文没有考虑 Xgboost 等前沿方法，**这主要是因为本文数据总量不大，用 RandomForest 等方法增加迭代次数完全跑得动，但是如果将时间序列再放大，变量维度再提高，更高级的分类方法肯定是需要的。**最后，本文前提是任意给定投资组合的情况下探寻最优策略，**在实际操作当中，量化策略的建立往往是先要寻找某个投资组合。规避了这一步也导致本文结论在实际当中的运用受到限制。

【参考文献】

1. 方匡南. 基金超额收益率方向预测与交易策略研究[J]. 经济经纬, 2010, (04): 61-65

2. Wolfgang. Applied Multivariate Statistical Analysis[M]. 北京大学:陈诗一, 2011. 258-264
3. 陈诗一. 非参数支持向量回归和分类理论及其在金融市场预测中的应用[M]. 北京大学:北京大学出版社, 2008. 131-160
4. Banz, B, W. Sample-dependent results using accounting and market data: some evidence[J]. Journal of Finance, 1986, (41): 135-148
5. Jaffe, J, Keim, D, Westerfield. Earning yields, market values and stock returns[J]. Journal of Finance, 1989, (47): 427-465
6. Peter, Harrington. Machine Learning in Action[M]. 北京:人民邮电出版社, 2013. 159-170
7. Clerk. Big Data Analytics with R and Hadoop[EB/OL]. http://blog.fens.me/r-hadoop-book-big-data/.
8. [1]孙娇. 多因子量化投资策略及实证检验[D].南京大学,2016.
9. [2]姜乐. 基于时间序列的股票价格分析研究与应用[D].大连理工大学,2015.
10. [3]鲍漪澜. 基于支持向量机的金融时间序列分析预测算法研究[D].大连海事大学,2013.
11. [4]全林,姜秀珍,赵俊和,汪东. 基于 SVM 分类算法的选股研究[J]. 上海交通大学学报,2009,(09):1412-1416.
12. [5]刘潭秋,谢赤. 基于 GARCH 模型与 ANN 技术组合的汇率预测[J]. 科学技术与工程,2006,(23):4690-4694.

【代码节选】

1.R 代码节选

```
rm(list = ls(all = TRUE))
gc()
library(data.table)
options(scipen = 200)

##聚类分析取样
test<-fread("all_p.csv",header = T)
head(test)
num=1:3545
num
sub=sample(x=num,size = 100)
x=test[sub,2:3]
head(x)
y=na.omit(x)
library(cluster)
dv=diana(y,diss=inherits(x,'dist'),metric='euclidean',stand = TRUE)
plot(dv,which.plots = 2,main='均值标准差聚类图',xlab='stock order')
rect.hclust(dv,k=6,border = 100)
library(car)
```

```
scatterplot(std~mean,data=test[,2:3],spread=F,lty.smooth=2,pch=19,main='全部 A 股均值标准差散点图',
           xlab='mean',
           ylab='std',
           xlim=c(-1,1))

scatterplot(std~mean,data=x,spread=F,lty.smooth=2,pch=19,main='所选样本均值标准差散点图',
           xlab='mean',
           ylab='std',
           xlim=c(-1,1))


##读取数据与修改变量类型
test1<-fread("数据库.csv",header = T)
head(test1)
test1=test1[-1,-1]
head(test1)
names(test1)[1:7]=c('ba','hq','tj','cs','it','gs','index')
head(test1)
str(test1)
test1$hq=as.numeric(test1$hq)
test1$tj=as.numeric(test1$tj)
test1$cs=as.numeric(test1$cs)
test1$it=as.numeric(test1$it)
test1$gs=as.numeric(test1$gs)
test1$ba=as.numeric(test1$ba)
test1$index=as.numeric(test1$index)
str(test1)
test1$ba_norm=scale(log(test1$ba),center = T,scale = T)
test1$hq_norm=scale(log(test1$hq),center = T,scale = T)
test1$tj_norm=scale(log(test1$tj),center = T,scale = T)
test1$cs_norm=scale(log(test1$cs),center = T,scale = T)
test1$it_norm=scale(log(test1$it),center = T,scale = T)
test1$gs_norm=scale(log(test1$gs),center = T,scale = T)
test1$index_norm=scale(log(test1$index),center = T,scale = T)
head(test1)

##画分布图
attach(test1)
d1=na.omit(ba_norm)
d2=na.omit(hq_norm)
d3=na.omit(tj_norm)
d4=na.omit(cs_norm)
```

```
d5=na.omit(it_norm)
d6=na.omit(gs_norm)
d7=na.omit(index_norm)
assist=density(d7)
opar=par(no.readonly = T)
par(mfrow=c(2,3))
plot(density(d1),main='Density of ba vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')

plot(density(d2),main='Density of hq vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')


plot(density(d3),main='Density of tj vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')


plot(density(d4),main='Density of cs vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')


plot(density(d5),main='Density of it vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')


plot(density(d6),main='Density of gs vs index',col='red',
     xlab = "std_log_price")

lines(assist,col='blue')

par(opar)
detach(test1)

##做正态性检验
```

```
head(test1)
shapiro.test(test1$ba_norm)
shapiro.test(test1$hq_norm)
shapiro.test(test1$tj_norm)
shapiro.test(test1$cs_norm)
shapiro.test(test1$it_norm)
shapiro.test(test1$gs_norm)
shapiro.test(test1$index_norm)

##缺失值可视化
head(test1)
test2=test1[,-1:-7]
head(test2)
library(mice)
c=md.pattern(test2)
library(VIM)
matrixplot(test2)

##线性插补
head(test2)
library(TTR)
library(forecast)
ittime=test2$it_norm
ittime=ts(ittime,start=c(1))
p1=plot(ittime)
##ittime 是只有 138 个值用来插补的数据
ittime1=ittime[1:138]
foreittime=HoltWinters(ittime1,gamma=F)
foreittime$SSE
plot(foreittime)##预测数据本身
f2=forecast.HoltWinters(foreittime,h=6)
p3=plot.forecast(f2)##预测区间
Box.test(f2$residuals,lag=20,type = 'Ljung-Box')

##visualize
opar=par(no.readonly = T)
par(mfrow=c(3,1))
plot(ittime,main='Original TSplot')

plot(foreittime)

legend('bottomright',inset = .05,title = 'Line Types',c('Original','Forecast'),
       lty=c(1,1),col=c('black','red'),cex=0.05)
```

```
plot.forecast(f2)
par(opar)

plot(density(na.omit(f2$residuals)),xlab='Forecast_Residual',
     main='Density of Residual Vs Norm',col='red')
sd=sd(na.omit(f2$residuals))
c=rnorm(100000,0,sd)

lines(density(c),col='blue')

legend('topright',inset=.05,title='Lines Type',c('Residual','Norm'),
       lty = c(1,1),col=c('red','blue'),cex=0.5)

##和正态比较
test2$it_norm[139:144]=f2$upper[1:6,1]
ittime3=test2$it_norm
ittime3=ts(ittime3,start=c(1))

plot.ts(ittime,col='blue',main='IT Time_Series Completed',
        ylab='Std_Log_Price')

lines(ittime3,lty=2)

legend('topleft',inset=0.05,title='Line Type',c('Original','Expected'),
       lty = c(1,2),col=c('blue','black'),cex=0.5)

##一般插补
head(test2)
library(Hmisc)
test2$hq_norm=impute(test2$hq_norm,mean)
test2$tj_norm=impute(test2$tj_norm,mean)
test2$cs_norm=impute(test2$cs_norm,mean)
test2$gs_norm=impute(test2$gs_norm,mean)
test2$ba_norm=impute(test2$ba_norm,mean)

##CAPM test

fit1=lm(index_norm~ba_norm,data=test2)
summary(fit1)

fit2=lm(index_norm~hq_norm,data=test2)
summary(fit2)
```

```
fit3=lm(index_norm~tj_norm,data=test2)
summary(fit3)

fit4=lm(index_norm~cs_norm,data=test2)
summary(fit4)

fit5=lm(index_norm~it_norm,data=test2)
summary(fit5)

fit6=lm(index_norm~gs_norm,data=test2)
summary(fit6)

##最小方差投资组合
head(test2)
v1=cov(test2[1:168,1:6])
Vlinv=solve(v1)
One=matrix(1,6,1)
Wgt=Vlinv%*%One
D=sum(Wgt*One)
Wgt=Wgt/D
print(Wgt)

##探索性数据描述 with ba as example

test3<-fread("ba.csv",header = T)
head(test3)
names(test3)[1:6]=c('lag_rtn','lag_Q','lag_M','lag_ROI','lag_index','rtn')
head(test3)
test3$rtn=ifelse(test3$rtn>0,1,2)
test3$rtn=as.factor(test3$rtn)
str(test3$rtn)

library(MASS)
set.seed(1234)
n0=sample(1:160,40)
ba.train=test3[-n0,]
ba.test=test3[n0,]

library(corrgram)##画相关图
corrgram(ba.test[,-'rtn'],order=TRUE,lower.panel=panel.shade,
        upper.panel=NULL,text.panel=panel.txt,main="Corrgram of all the
variables",cex=1.5)

library(ggplot2)
```

```
palette=scale_fill_brewer(palette = "RdBu")##设置调色板
color=scale_color_brewer(palette= "RdBu")
labs=labs(x="rtn",title='on',fill='rtn')
theme=theme(text=element_text(size=20),plot.title =
element_text(hjust=0.5),legend.key.size = unit(1.5,'cm'),
           axis.title.x = element_text(size = 18),axis.title.y=element_text(size=18))

ggplot(ba.train)+geom_point(aes(x=lag_ROI,
                               y=lag_M,color=as.factor(rtn),fill=T),size=2)+color+
  labs(title='Scatter diagram of lag_ROI VS lag_M',color='direction')+theme

##Single Classifier with ba as example

##lda qda
balda=lda(rtn~.,data=ba.train)
baqda=qda(rtn~.,data=ba.train)

prelda=predict(balda,newdata = ba.test)
s1=sum(prelda$class!=ba.test$rtn)
error1=s1/length(ba.test$rtn)
error1
s1

preqda=predict(baqda,newdata = ba.test)
s2=sum(preqda$class!=ba.test$rtn)
error2=s2/length(ba.test$rtn)
s2
error2

##bagging randomforeset adboost
library(ipred)
library(rpart)
rtn.bag=bagging(rtn~.,data=ba.train,nbagg=200,
              control=rpart.control(minsplit=2,cp=0,
                                    xval=0),coob=T)
pre.bag=predict(rtn.bag,newdata = ba.test)
error3=mean(pre.bag!=ba.test$rtn)
error3

library(randomForest)
rtn.rf=randomForest(rtn~.,data=ba.train)
plot(rtn.rf)
varImpPlot(rtn.rf)
pre.rf=predict(rtn.rf,newdata = ba.test)
```

```
error4=mean(pre.rf!=ba.test$rtn)
error4

library(ada)
rtn.ad=ada(rtn~.,data=ba.train,iter=100,loss='logistic',
          type='discrete')
pre.ada=predict(rtn.ad,newdata = ba.test,type = 'vector')
error5=mean(pre.ada!=ba.test$rtn)
error5

##SVM(Partially by R Partially by PY)
library(e1071)
svm.rtn=svm(rtn~.,data=ba.train,cross=80,
           type='C-classification',cost=1,
           kernel='poly',gamma=1,degree=2,coef0=1)
summary(svm.rtn)

##SVM(Partially by R Partially by PY)
from sklearn import cross_validation
from sklearn import preprocessing

ba_train,ba_test = cross_validation.train_test_split(features_scaler,
lables, test_size = 0.2, random_state = 0)

clf_svm = svm.SVC(C = 300,gamma = 1.03)
clf_svm.fit(ba_train)
print "预测准确率为: %0.2f" % (clf_svm.score(ba_test))

##ANN
library(nnet)
rtn.net=nnet(rtn~.,data=ba.train,size=5)
pre.net=predict(rtn.net,newdata=ba.test,type='class')
error6=mean(pre.net!=ba.test$rtn)
error7
```

2.Python 代码节选

```
"""爬取数据"""
def get_mean_std(stockcode, start, end):

    pass

data = pd.DataFrame()
with open('code_new.txt', 'rb') as fin:
    stock_code = fin.readline()
```

```
    stock_code = stock_code.strip()
    # path = 'csv' + os.sep
    # if os.path.exists(path):
    #     pass
    # else:
        # os.mkdir(path)
        # path = path + '2015' + '.csv'
    while stock_code:
        print 'Log: %s '%stock_code
        data_t = ts.get_hist_data(stock_code, retry_count=3, start='2015-01-01',
end='2015-12-31')
        if str(type(data)) == '<type \'NoneType\'>':
            pass
        else:
            data1 =
data_t.drop(['open','high','close','low','price_change','ma5','ma10','ma20','v_ma5'
,'v_ma10','v_ma20'], axis=1)
            data_out = data1.describe()
            data_out = data_out.drop(['count','min','25%','50%','75%','max'], axis=0)
            data[stock_code] = data_out['p_change']
        stock_code = fin.readline()
        stock_code = stock_code.strip()

"""线性集成分类器，LCM as example"""
def tendency_for_linear(all_predict,all_error):
    linear_pre=[all_predict[0],all_predict[1],all_predict[5]] #0 代表 LDA，1 代表 QDA，5
代表 SVM
    linear_err=[all_error[0],all_error[1],all_error[5]]
    wgt=[1-i for i in linear_err]
    std_wgt=[i/sum(wgt) for i in wgt]
    vote=[linear_pre[i]*std_wgt[i] for i in range(0,len(linear_err)-1) if
linear_pre[i]==1]
    if sum(vote)>0.5:#激活阈值取 0.5
        print('positive tendency')
    else:
        print('negative tendency')
    return sum(vote)
```