法律声明

本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

加班主任入群

第4课：仓位管理和风险控制功能实现

主　讲：刘英斐

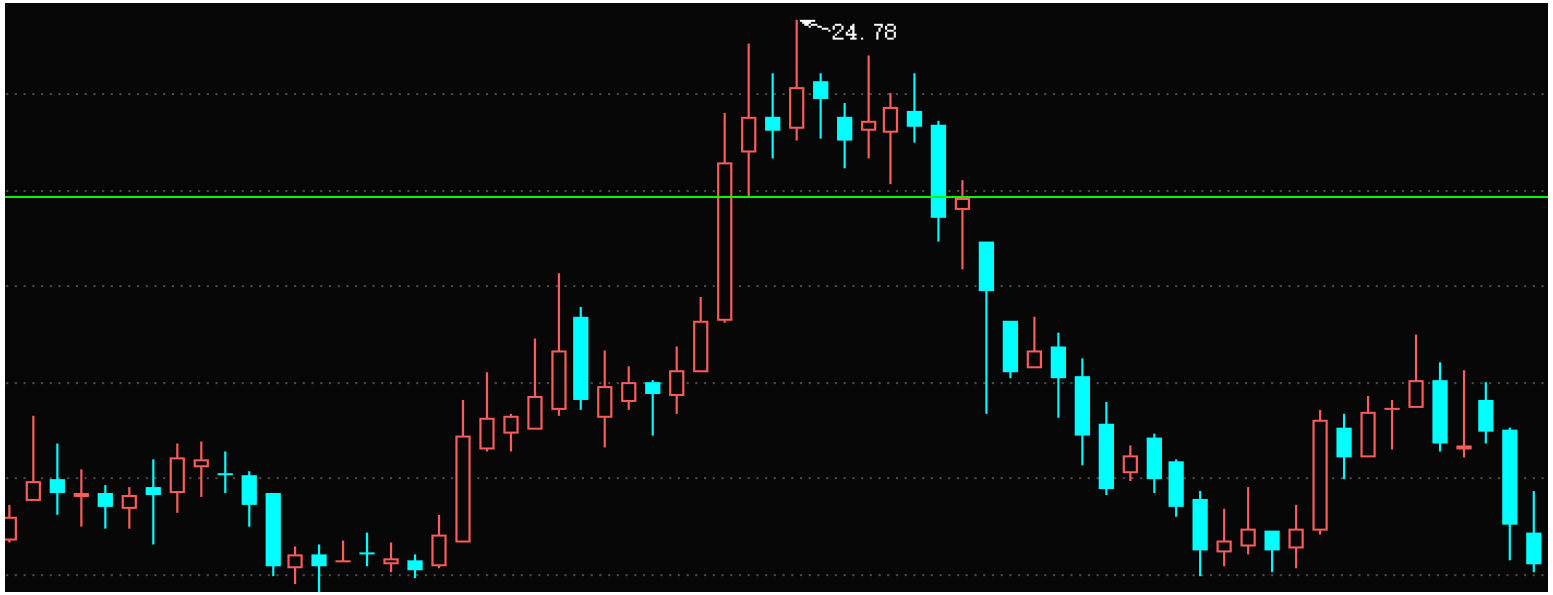
课程大纲

截断亏损 让利润奔跑

止盈和止损的功能实现

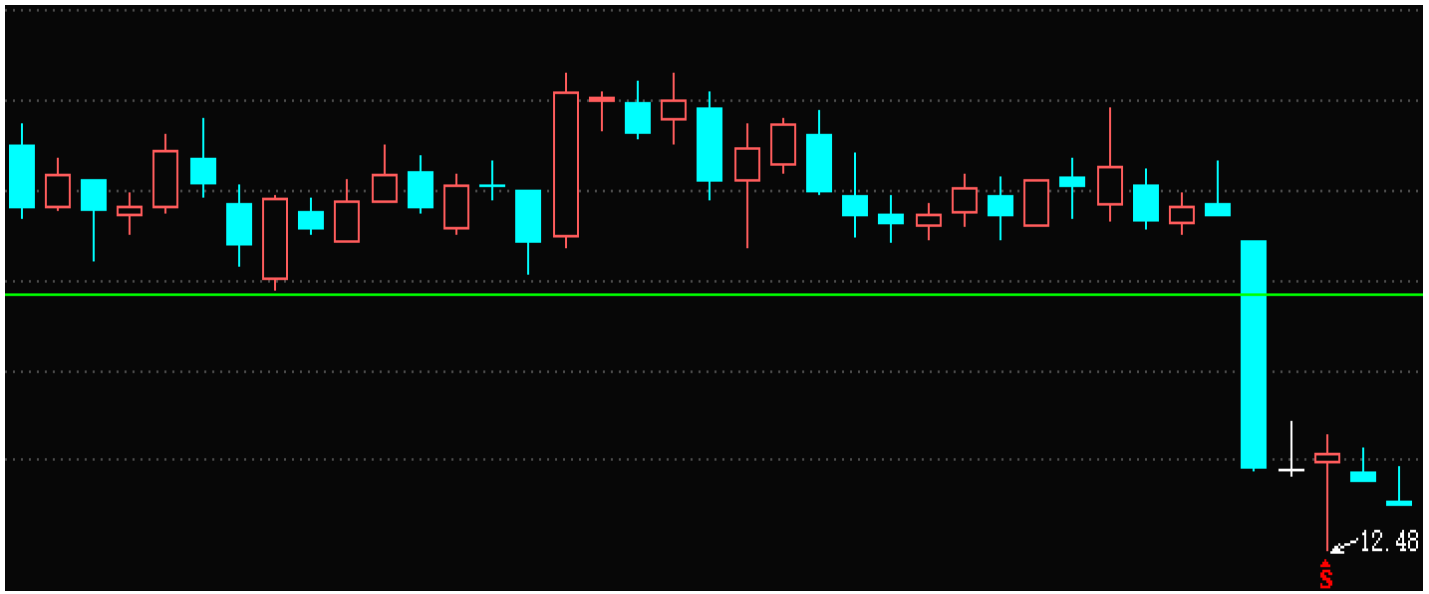
止盈的意义

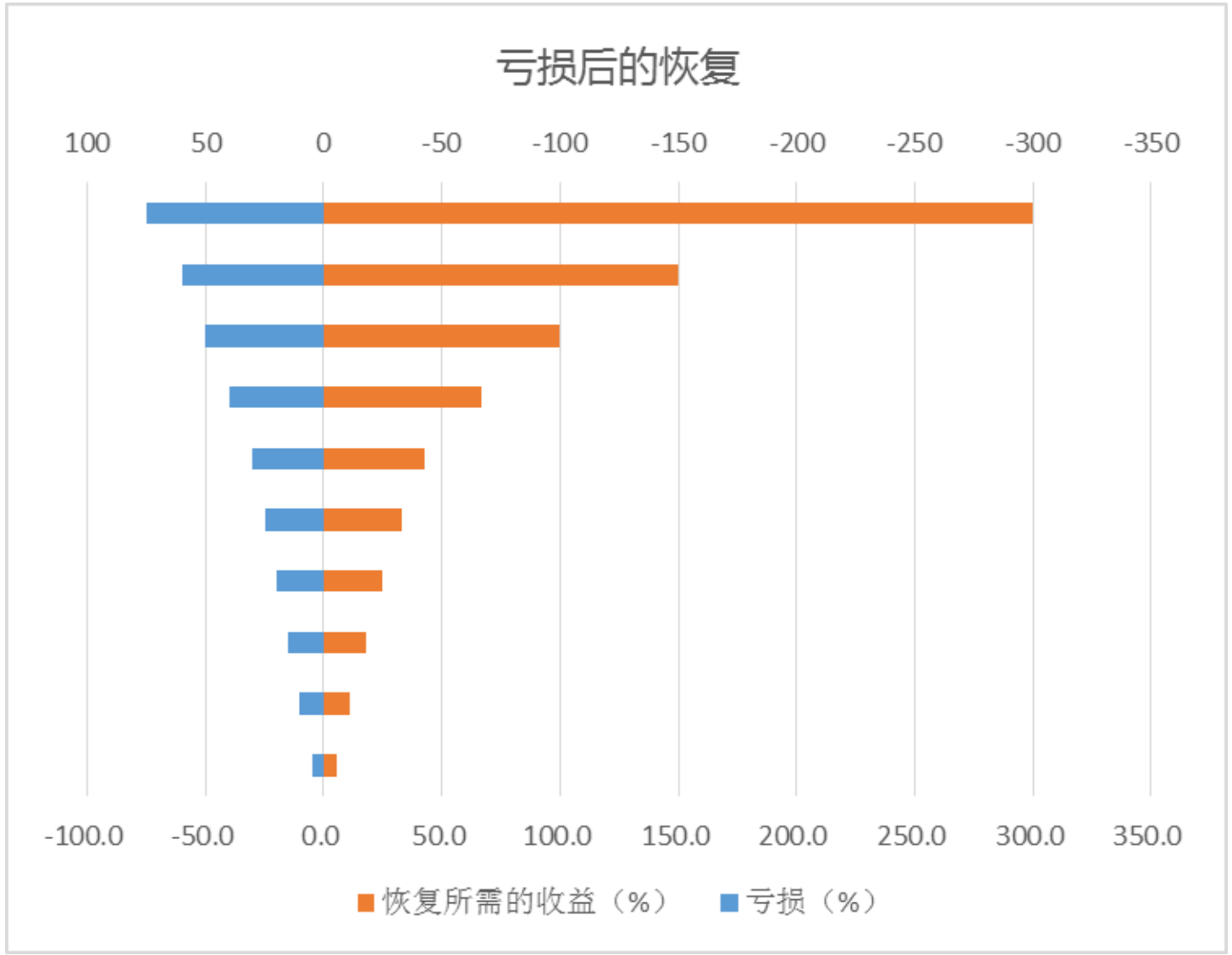
- 火鸡的故事
- 当风险逐渐变大时，落袋为安



止损的意义

- 鳄鱼法则
- 留得青山在，不怕没柴烧

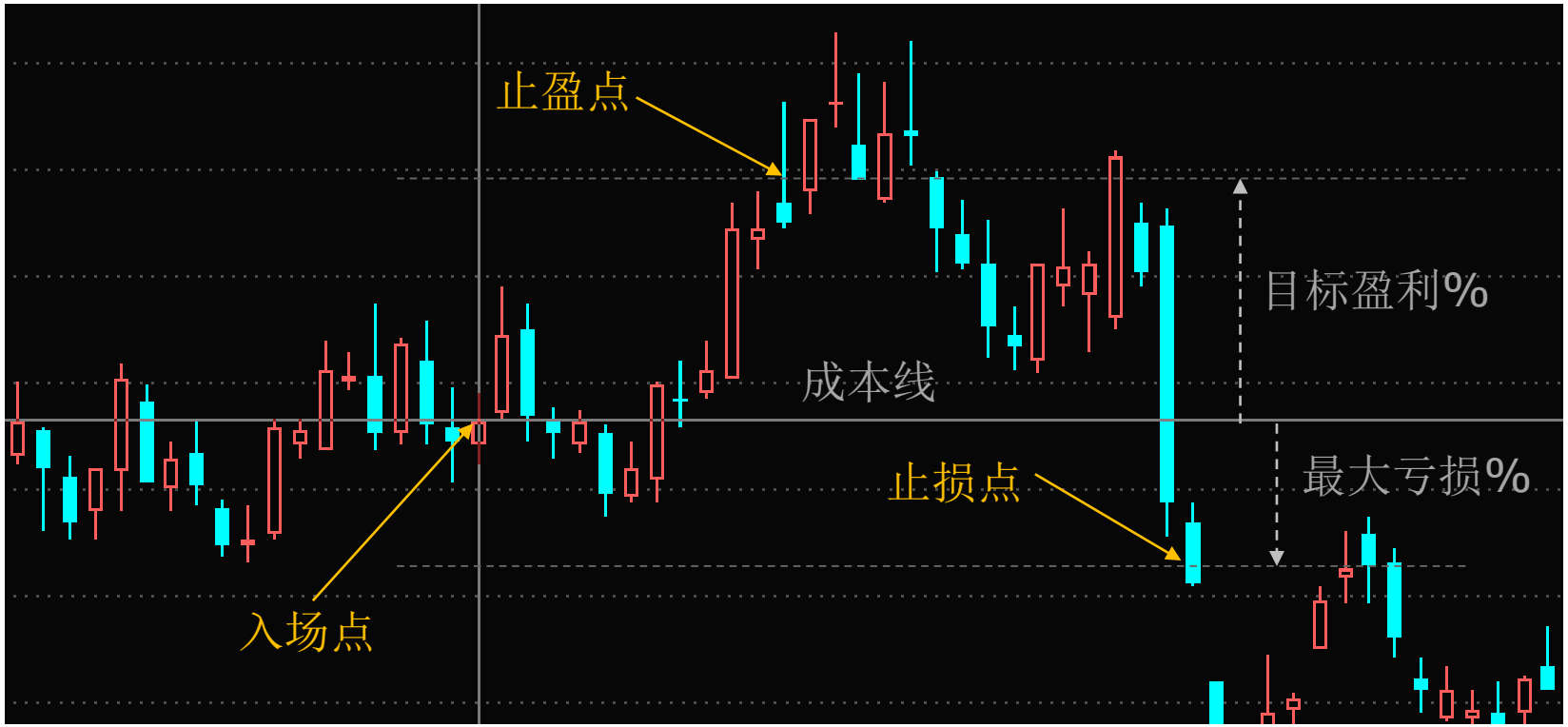



亏损后的恢复
100
50
0
-50
-100
-150
-200
-250
-300
-350
-100.0
-50.0
0.0
50.0
100.0
150.0
200.0
250.0
300.0
350.0
恢复所需的收益（%）
亏损（%）

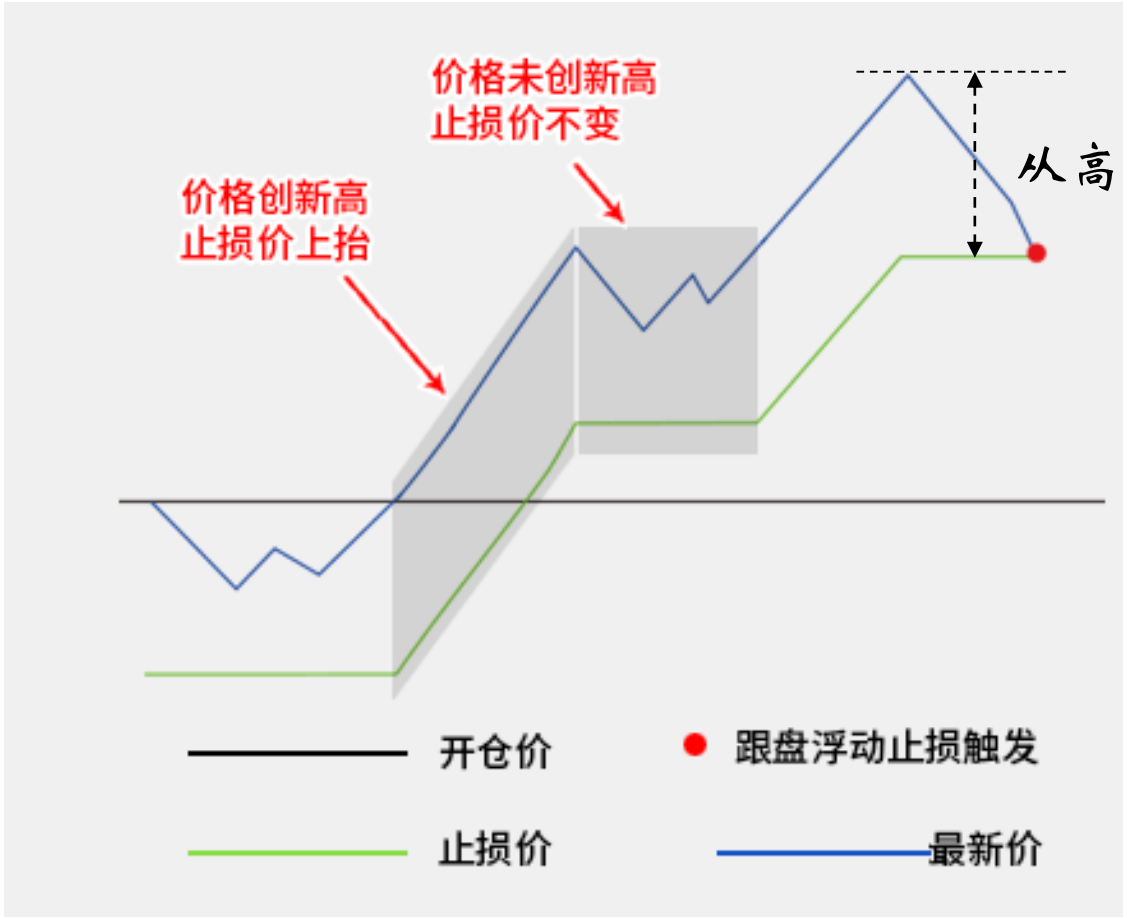
止盈止损方法

- 止盈
  - 目标盈利法
  - 技术信号止盈
- 止损
  - 最大亏损法
  - 技术信号止损
- 浮动止盈止损

目标盈利法和最大亏损法



回撤止盈（浮动/跟踪止盈）



技术信号止盈/止损

以双均线交叉信号为例

合理的头寸规模是实现盈利目标的重要保证

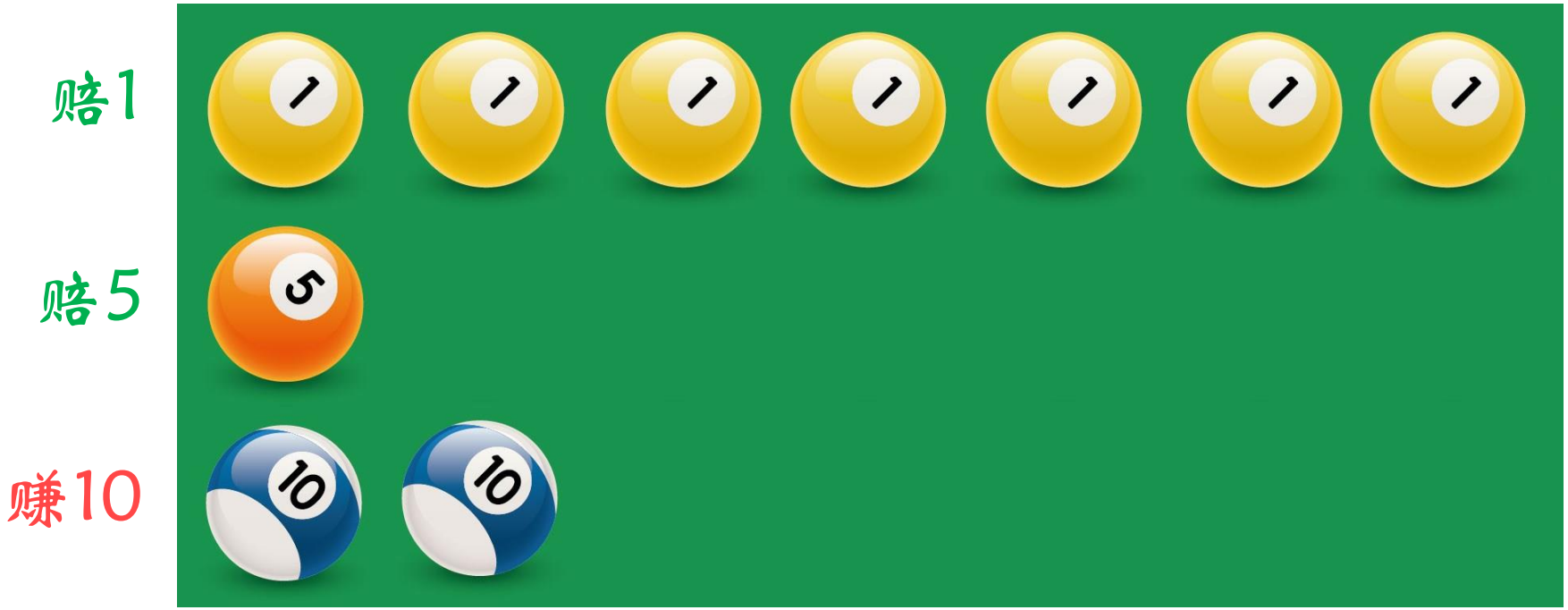
仓位管理的工程实现

交易系统中与盈利相关的要素

- 策略的胜率
- 每次交易的盈亏比
- **头寸规模的确定**
- 资金量的大小
- 交易机会的多少
- 交易成本

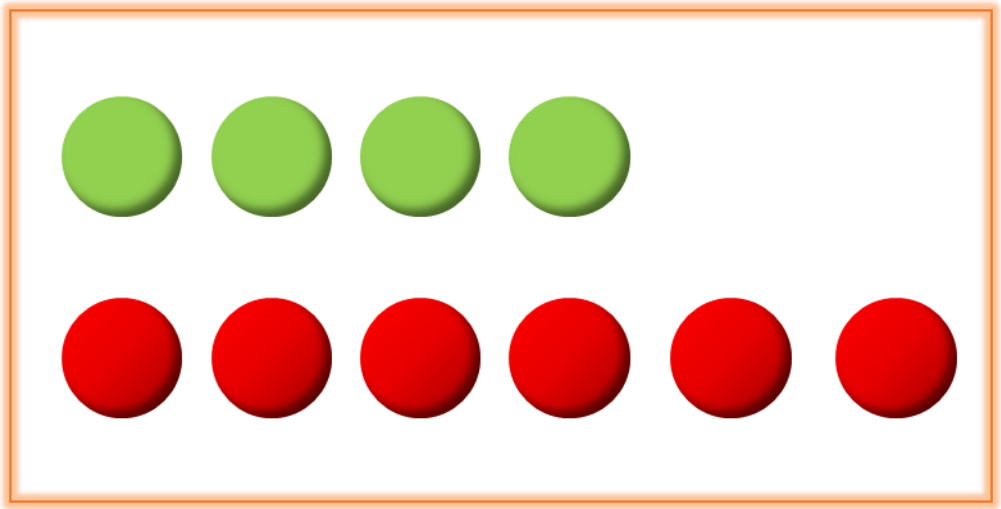
} 期望收益

取球游戏的启示（1）



胜率 = 20%　期望收益 = -0.7*1-0.1*5+0.2*10 = 0.8

取球游戏的启示（2）



- 本金1000元
- 抽中红球，赢得赌注的70%
- 抽中绿球，输掉全部赌注

试验：抽取1000次

- 每次投注10元，20次试验平均剩余1196元
- 每次投注可用资金的1%，20次试验平均剩余1223元
- 按凯利公式下注，20次试验平均剩余1879元

$$f^* = \frac{bp - q}{b} = p - q\frac{1}{b}$$

- f* = 现有资金中用于下次投注或投资的比例
- b = 赔率（即：盈亏比=期望盈利 / 可能亏损）
- p = 成功概率（胜率）
- q = 失败概率 （也就是 1-p）

b = 0.7
P = 0.6
q = 0.4
f* = 0.028

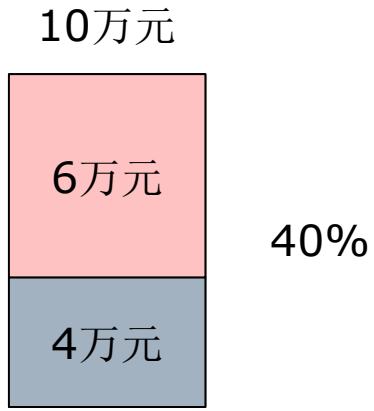
每次下注金额28元

随机入市策略的启示

- 每次凭掷硬币的结果随机进入市场，或者做多头，或者做空头；一旦得到一个退出市场的信号，就基于随机信号再次入市
- 用EMA(ATR,10)指标来确定市场的波动性，用这一波动性得数的3倍作为初始止损
- 跟踪止损，只按照对盈利有利的方式移动（多头向上，空头向下）
- 在10个期货市场上检验：只进行1笔合约的交易 vs. 使用1%风险法则来确定头寸规模

仓位管理

$$仓位(\%) = \frac{投入资金}{总资金}$$



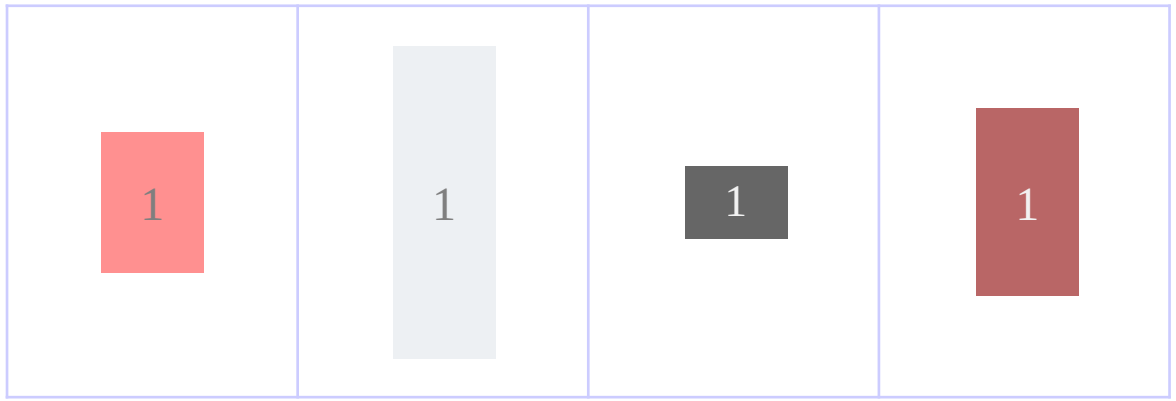
- 不同阶段：建仓、加仓/减仓、清仓
- 资金分配：头寸规模的确定和分配

确定头寸规模的三种模型

- 每一固定金额交易一个单位
- 等价值交易单位
- 百分比风险模型

模型1：每一固定金额交易一个单位

资金池：



把资金等分为相同金额的若干份，在出现买入信号的情况下，每份只允许交易一个单位的投资标的（比如1手股票，或是1份期货合约）

模型1：每一固定金额交易一个单位

总资金：一百万

平均分为10份，每份金额：10万

交易实例一：

待买入标的：工商银行　　价格：5元　一个交易单位 = 100股

买入数量 = 100股　　买入金额 = 5 * 100 = 500元

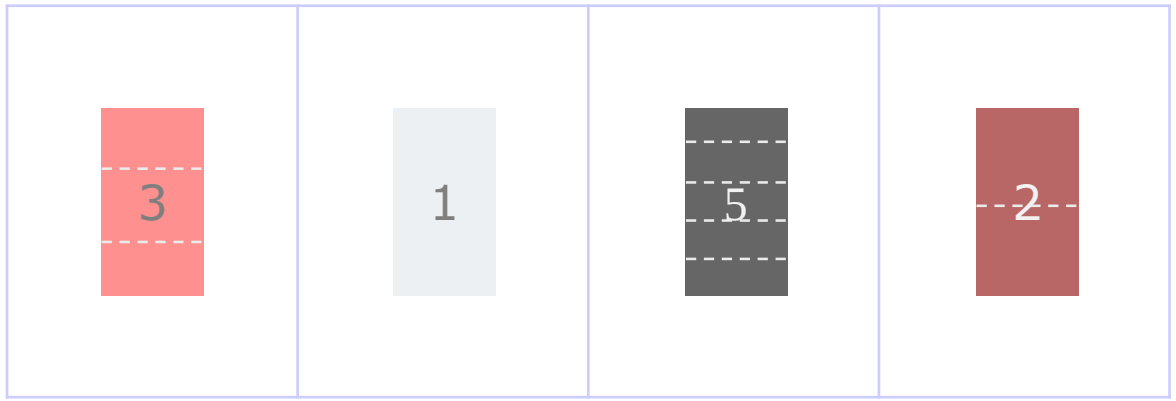
交易实例二：

待买入标的：贵州茅台　　价格：750元　一个交易单位 = 100股

买入数量 = 100股　　买入金额 = 750 * 100 = 7.5万

模型2：等价值交易单位

资金池：



把资金等分为相同金额的若干份，在出现买入信号的情况下，按该金额计算出每份允许交易的此投资标的单位个数（比如M手股票，或是N份期货合约）

模型2：等价值交易单位

总资金：一百万

平均分为10份，每份金额：10万

交易实例一：

待买入标的：工商银行　　价格：5元

买入数量 = 100000 / 5 = 20000股　　买入金额 = 10万

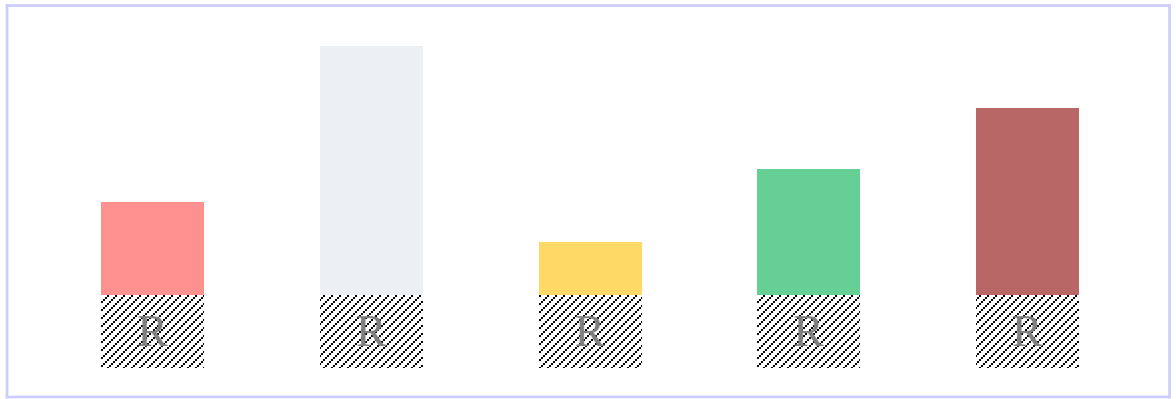
交易实例二：

待买入标的：贵州茅台　　价格：750元

买入数量 = 100000 / 750 = 100股　　买入金额 = 7.5万

模型3：百分比风险模型

资金池：



根据每次交易允许承担的最大风险占总资金的比例，以及每个投资标的可接受的最大损失（即初始止损额度R），折算出可建立头寸的单位个数

CPR公式：$P(\text{头寸规模}) = C(\text{现金})/R(\text{每股风险})$

模型3：百分比风险模型

总资金：一百万

总风险：1%

单个交易标的风险：5%

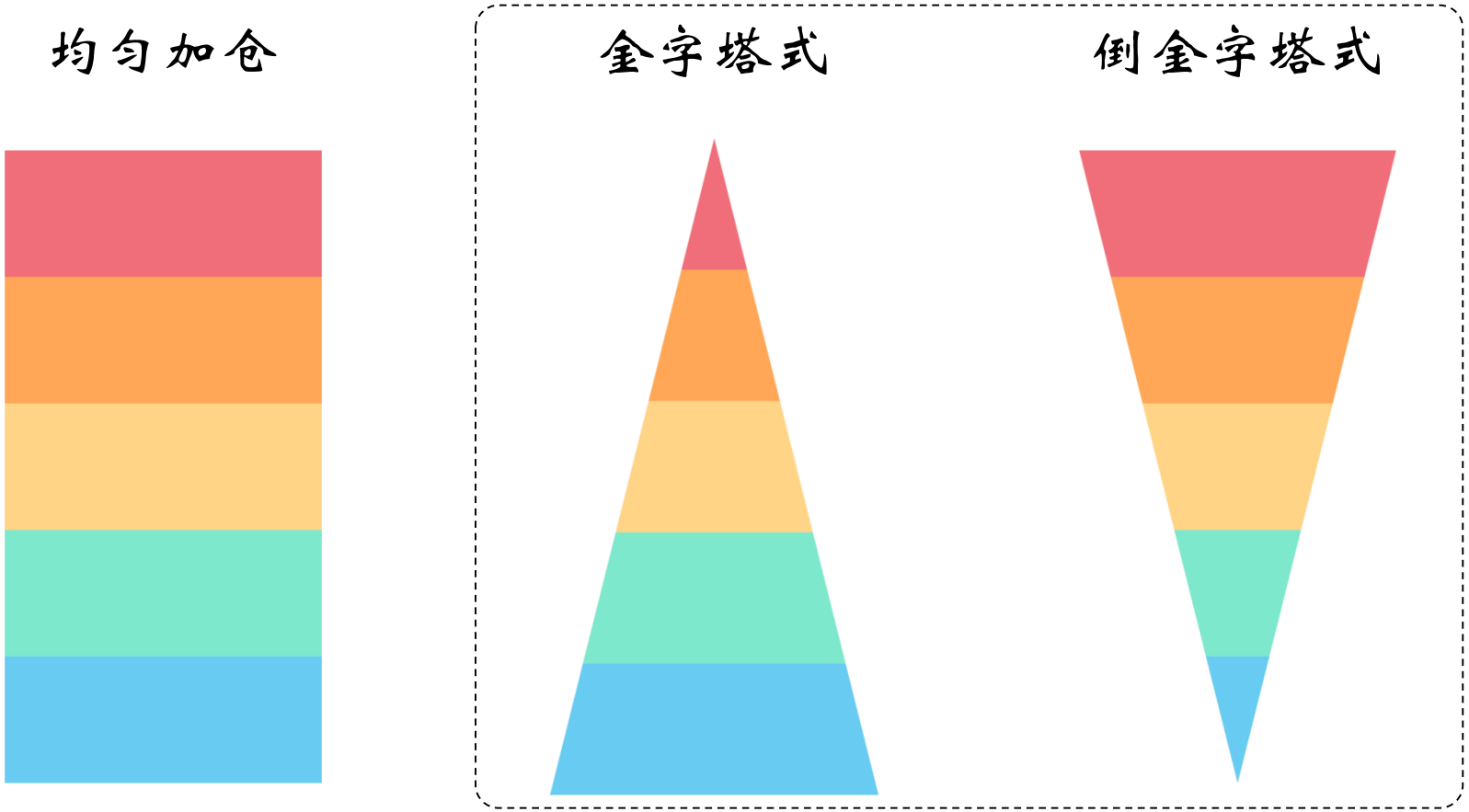
交易实例：

待买入标的：海康威视　　价格：40元

买入数量 = （1000000 * 1%）/（40 * 5%）= 5000股

买入金额 = 40 * 5000 = 20万

常见的加仓方法



仓位分配的其他考虑

- 复利/不复利方式
- 资金占比上限
- 单日分配资金上限
- 同时持仓股（数量）上限
- 单只股票仓位上限
- ……

工程化实现的考虑

- 可用资金不足时的处理
  - 限制新的购买，忽略该信号
  - 在增加新股票之前，消除业绩最差的头寸
  - 为继续购买新的，将头寸规模确定得小一些

- 入选组合的优先级
  - 根据某种指标排序
  - 根据历史表现筛选

休息一下

完整的交易系统要素 一个都不能少

编写一个完整的交易策略

交易系统的核心要素

- 市场：买卖什么？
- 头寸规模：买卖多少？
- 入市：什么时候买进？
- 止损：什么时候放弃一个亏损的头寸？
- 退出：什么时候退出一个盈利的头寸？
- 战术：怎么买卖？

1. 基准策略和测试环境

- 股票池候选：上证50成分股，不调仓
- 双均线策略
  - 日K级别，MA10 vs. MA30
  - 开盘前检测信号，金叉买、死叉卖
- 头寸规模确定
  - 均仓方案（等额资金分配）
- 测试环境
  - 回测区间：2016.1.1~2018.6.30，资金：1000万
  - 编程语言：Python，平台：www.joinquant.com

dual_ma_plus_1_baseline.py

```
# 导入函数库
from jqdata import *
# 均线
MA_WIN_1 = 10
MA_WIN_2 = 30

# 初始化函数，设定基准等等
def initialize(context):
    set_benchmark('000300.XSHG')
    set_option('use_real_price', True)
    # log.set_level('order', 'error')

    # 股票类每笔交易时的手续费是：买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税,
    # 每笔交易佣金最低扣5块钱
    set_order_cost(OrderCost(close_tax=0.001, open_commission=0.0003,
                             close_commission=0.0003, min_commission=5), type='stock')

    # 定时运行函数
    run_daily(before_market_open, time='before_open',
                          reference_security='000300.XSHG')
    run_daily(market_open, time='every_bar', reference_security='000300.XSHG')
    run_daily(after_market_close, time='after_close',
                          reference_security='000300.XSHG')
    #股票池-上证50
    g.stock_pool = get_index_stocks("000016.XSHG", date=context.current_dt)
    g.init_cash = context.portfolio.starting_cash  # 启动资金
```

```
# 开盘前运行函数
def before_market_open(context):
    look_ahead_n = max(MA_WIN_1, MA_WIN_2) + 1
    g.up_cross_signaled = set()
    g.down_cross_signaled = set()
    for code in g.stock_pool:
        df = attribute_history(code, look_ahead_n, "1d", ["close"],
                               skip_paused=True)  # 该函数返回结果不包括当天数据
        if len(df) != look_ahead_n:
            continue
        close = df["close"]
        ma_short = pd.rolling_mean(close, MA_WIN_1)  # 短时均线
        ma_long = pd.rolling_mean(close, MA_WIN_2)   # 长时均线
        # 上穿标志
        uc_flags = (ma_short.shift(1) <= ma_long.shift(1)) & (ma_short > ma_long)
        # 下穿标志
        dc_flags = (ma_short.shift(1) >= ma_long.shift(1)) & (ma_short < ma_long)

        if uc_flags.iloc[-1]:
            g.up_cross_signaled.add(code)
        if dc_flags.iloc[-1]:
            g.down_cross_signaled.add(code)

# 开盘时运行函数
def market_open(context):
    cur_dt = context.current_dt.date()  # 当前日期
```

```
    p = context.portfolio  # 资金账户
    current_data = get_current_data()

    each_cash = g.init_cash / len(g.stock_pool)  # 每只股票分配的资金

    # 卖出均线死叉信号的持仓股
    for code, pos in p.positions.items():
        if code in g.down_cross_signaled:
            order_target(code, 0)

    # 买入均线金叉信号的持仓股
    for code in g.up_cross_signaled:
        if code not in p.positions:
            if current_data[code].paused:
                continue
            open_price = current_data[code].day_open
            num_to_buy = each_cash / open_price // 100 * 100
            order(code, num_to_buy)

# 收盘后运行函数
def after_market_close(context):
    p = context.portfolio
    pos_level = p.positions_value / p.total_value
    record(pos_level=pos_level)
```

基准策略回测结果


收益概述
策略收益
5.74%
策略年化收益
2.32%
基准收益
-5.90%
Alpha
0.003
Beta
0.303
Sharpe
-0.197
胜率
0.396
盈亏比
1.384
最大回撤
12.020%
其他指标
缩放：
1个月
1年
全部
策略收益
基准收益
超额收益
普通轴
对数轴
超额收益
时间：
2016-01-04
-
2018-06-29
20%
15%
10%
5%
0%
-5%
-10%
-15%
-20%
-25%
-30%
收益
1
0.75
0.5
0.25
0
-0.25
自定义数据

2. 按盈利比例均匀加仓

- 记录每只持仓股最后一次的买入价
- 如以当日开盘价相对于前一次买入价的盈利比例超过某个阈值，则等额加一次仓
  - 前提是当日没有任何止损或其它买入信号发生

dual_ma_plus_2_inc_pos_by_pf.py

```
#加仓判断阈值
INC_POS_PF_RATE = 0.05

def initialize(context):
    # ... ...
    g.last_entry_prices = {code:None for code in g.stock_pool}

def market_open(context):
    # ... ...
    # 卖出均线死叉信号的持仓股
    # ... ...
            g.last_entry_prices[code] = None  # 更完美的实现还要判断成交状态，后面同理

    # 买入均线金叉信号的持仓股
    # ... ...
            g.last_entry_prices[code] = open_price

    # 检查有无符合加仓条件的持仓股
    for code, pos in p.positions.items():
        if current_data[code].paused:
            continue
        if pos.today_amount == 0 and pos.closeable_amount > 0:
            open_price = current_data[code].day_open
            last_entry = g.last_entry_prices[code]
            if (open_price - last_entry) / last_entry >= INC_POS_PF_RATE:
                order_value(code, each_cash)
                g.last_entry_prices[code] = open_price
```

加仓方案回测结果


收益概述
策略收益
4.78%
策略年化收益
1.94%
基准收益
-5.90%
Alpha
0.006
Beta
0.407
Sharpe
-0.175
胜率
0.317
盈亏比
1.323
最大回撤
16.224%
其他指标
缩放：
1个月
1年
全部
策略收益
基准收益
超额收益
普通轴
对数轴
超额收益
时间：
2016-01-04
-
2018-06-29
25%
20%
15%
10%
5%
0%
-5%
-10%
-15%
-20%
-25%
-30%
收益
1
0.75
0.5
0.25
0
-0.25
自定义数据

3. 按百分比风险模型分配资金

- 设定可以承担的总风险
- 每只股票允许承担一定百分比的风险，折算到具体的风险金额
- 以该总风险和个股风险的值来确定建仓个股的资金头寸
- 基于基准策略进行改进实验

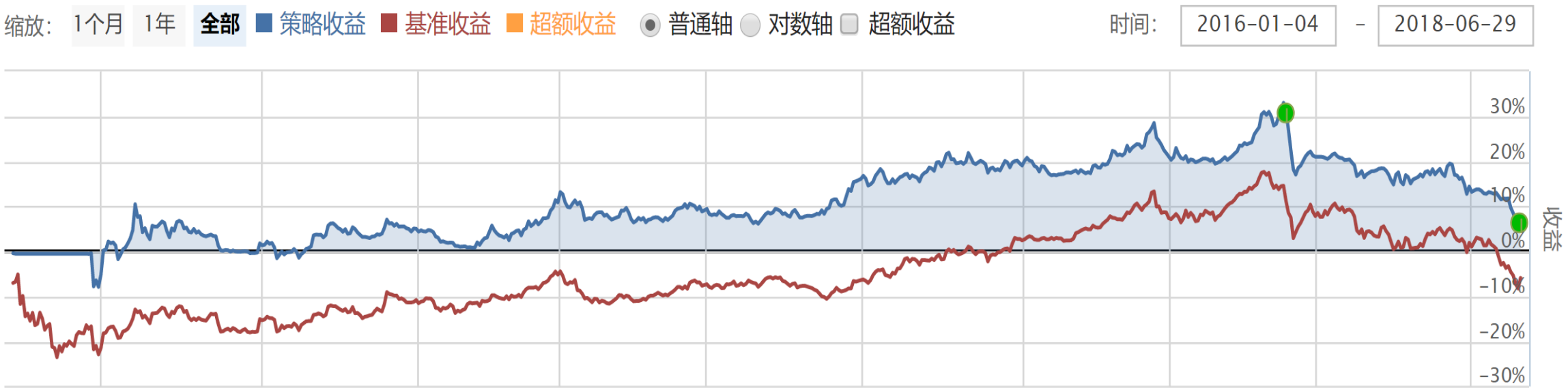
```
# 总风险因子
RISK_RATIO = 0.001
# 个股风险因子
STOCK_RISK_RATIO = 0.01

# 开盘时运行函数
def market_open(context):
    cur_dt = context.current_dt.date()  # 当前日期
    p = context.portfolio  # 资金账户
    current_data = get_current_data()
    # 卖出均线死叉信号的持仓股
    for code, pos in p.positions.items():
        if code in g.down_cross_signaled:
            order_target(code, 0)
    # 买入均线金叉信号的持仓股
    for code in g.up_cross_signaled:
        if code not in p.positions:
            if current_data[code].paused:
                continue
            num_to_buy = (g.init_cash * RISK_RATIO) / (current_data[code].last_price * STOCK_RISK_RATIO)
            order(code, num_to_buy)
```

按百分比风险模型分配资金回测结果

策略收益	策略年化收益	基准收益	Alpha	Beta	Sharpe	胜率	盈亏比	最大回撤
7.75%	3.12%	-5.90%	0.027	0.546	-0.060	0.420	1.441	19.97%

其他指标



4. 回撤止盈（浮动止损）

- 算法描述：
  - 建仓后，时刻记录股价走势的最高点
  - 计算最新价相对于前期高点下跌幅度的百分比
  - 如果下跌百分比大于某个阈值，则止盈退出

dual_ma_plus_4_floating_exit.py

```
# 回撤的幅度
MAX_DROP_RATE = 0.03

def initialize(context):
    # ... ...
    g.entry_dates = {code:None for code in g.stock_pool}

def market_open(context):
    # ... ...
    # 买入均线金叉信号的持仓股
    # ... ...
            g.entry_dates[code] = cur_dt

    # 检查有无符合回撤止盈条件的持仓股
    for code, pos in p.positions.items():
        # ... ...
        if pos.today_amount == 0 and pos.closeable_amount > 0:
            last_entry_date = g.entry_dates[code]
            prev_date = context.current_dt - timedelta(days=1)
            # 计算截止到前一天的HHV，避免未来函数，考虑到前复权，每个交易日都重新计算
            df = get_price(code, start_date=last_entry_date, end_date=prev_date,
                           frequency="1d", fields=["high"], skip_paused=True)
            hhv = df["high"].max()
            # 以当日开盘价计算，或者，也可以用前一交易日的收盘价计算
            drop_rate = (hhv - current_data[code].day_open) / hhv
            if drop_rate > MAX_DROP_RATE:
                order_target(code, 0)
```

增加回撤止盈后的回测结果

策略的可调参数

- #均线级别
- MA_WIN_1 = 10
- MA_WIN_2 = 30

- #股票池-上证50
- 000016.XSHG

- #加仓判断阈值
- INC_POS_PF_RATE = 0.05

- # 总风险因子
- RISK_RATIO = 0.001
- # 个股风险因子
- STOCK_RISK_RATIO = 0.01

- # 回撤的幅度
- MAX_DROP_RATE = 0.03

总结

- 风险控制方法
- 仓位管理方法
- 编写一个完整的交易策略
  - 完整交易系统的要素
  - 仓位管理和止盈/止损的编写示例
  - 渐进式的改进过程

课后练习

- 思考“按盈利比例均匀加仓”方案的可能改进方法，以减小它在持续下跌行情中的回撤

下节课预告

- 第5课：怎么评价和诊断交易策略

  - 几种常用评价指标的工程实现
  - 什么样的策略是好策略
  - 从哪些方面去改进策略

问答互动

在所报课的课程页面，
1、点击“全部问题”显示本课程所有学员提问的问题。
2、点击“提问”即可向该课程的老师和助教提问问题。

联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

THANKS