# 第一部分：设定策略参数

start = '2018-03-01' # 回测起始时间

end = '2018-03-31' # 回测结束时间

benchmark = 'HS300' # 策略参考标准

universe = ['601600.XSHG', '603988.XSHG', '603088.XSHG', '603998.XSHG', '600076.XSHG', '600856.XSHG', '600069.XSHG', '600682.XSHG', '600179.XSHG', '600290.XSHG'] # 证券池，支持股票和基金

capital\_base = 100000 # 起始资金

freq = 'd' # 策略类型，'d'表示日间策略使用日线回测，'m'表示日内策略使用分钟线回测

refresh\_rate = 1 # 调仓频率，表示执行handle\_data的时间间隔，若freq = 'd'时间间隔的单位为交易日，若freq = 'm'时间间隔为分钟

# 第二部分：初始化策略，回测期间只运行一次，用于设置全局变量

# account是回测期间的虚拟交易账户，存储上述全局变量参数信息，并在整个策略执行期间更新并维护可用现金、证券的头寸、每日交易指令明细、历史行情数据等

def initialize(account):

# account.i = 1

pass

# 第三部分：策略每日下单逻辑，执行完成后，会输出每天的下单指令列表

# 此函数在每个交易日开盘前被调用，模拟每个交易日开盘前，交易策略会根据历史数据或者其他信息进行交易判断，生成交易指令

def handle\_data(account):

hist = account.get\_attribute\_history('closePrice', 3)

for s in account.universe:

if hist[s][2] - hist[s][0] > 0.8 and s not in account.valid\_secpos:

order\_pct(s, 0.8)

elif hist[s][2] - hist[s][0] < 0.8 and s not in account.valid\_secpos:

order\_pct(s, 0)