"""

This is a template algorithm on Quantopian for you to adapt and fill in.

"""

import quantopian.algorithm as algo

from quantopian.pipeline import Pipeline

from quantopian.pipeline.data.builtin import USEquityPricing

from quantopian.pipeline.filters import QTradableStocksUS

def initialize(context):

"""

Called once at the start of the algorithm.

"""

# Rebalance every day, 1 hour after market open.

algo.schedule\_function(

rebalance,

algo.date\_rules.every\_day(),

algo.time\_rules.market\_open(hours=1),

)

# Record tracking variables at the end of each day.

algo.schedule\_function(

record\_vars,

algo.date\_rules.every\_day(),

algo.time\_rules.market\_close(),

)

# Create our dynamic stock selector.

algo.attach\_pipeline(make\_pipeline(), 'pipeline')

def make\_pipeline():

"""

A function to create our dynamic stock selector (pipeline). Documentation

on pipeline can be found here:

https://www.quantopian.com/help#pipeline-title

"""

# Base universe set to the QTradableStocksUS

base\_universe = QTradableStocksUS()

# Factor of yesterday's close price.

yesterday\_close = USEquityPricing.close.latest

pipe = Pipeline(

columns={

'close': yesterday\_close,

},

screen=base\_universe

)

return pipe

def before\_trading\_start(context, data):

"""

Called every day before market open.

"""

context.output = algo.pipeline\_output('pipeline')

# These are the securities that we are interested in trading each day.

context.security\_list = context.output.index

def rebalance(context, data):

"""

Execute orders according to our schedule\_function() timing.

"""

pass

def record\_vars(context, data):

"""

Plot variables at the end of each day.

"""

pass

def handle\_data(context, data):

"""

Called every minute.

"""

pass