```
# def logn():
#     tim_n = 60*60*24*10
#     rh.authentication.login(username='jamdudu@gmail.com',
#                             password='Robin@2022',
#                             expiresIn=tim_n,
#                             scope='internal',
#                             by_sms=True,
#                             store_session=True)
# logn()
# # def logout():
#     rh.authentication.logout()

# def get_stocks():
#     # add your stocks here
#     stocks = list()
#     stocks.append('FB')
#     stocks.append('AAPL')
#    # stocks.append('MSFT')
#     return(stocks)
# stocks = get_stocks()
#import config
#import trade_strat
#import grapher
#from config import *
import                          as
import          as
import
import        as

import
import                    as
import       as

#USERNAME ='267272419@qq.com'
#PASSWORD = '12345shift'
         'markliang@yahoo.com'
          'Jml168168$'

from          import          as
import


    str
    str
    str
                r'C:\Users\jizha\Desktop\seabridge_datapool1\final_strategy_data_temporaly\three_s
               '0'
def login
                    60 60 24


                                 'internal'
                                  True
                                         True
      100
```

```
def open_market
    from          import

             False

    import
                                9 30 0  # 9:30AM
                                 15 59 0  # 3:59PM

    if                          and
                 True
    else
        # print('### market is closed')
        pass

    return


# myCash =rh.profiles.load_account_profile()
# cash = myCash['margin_balances']["day_trade_buying_power"]

#import config
# def get_cash():
#     rh_cash = rh.account.build_user_profile()

#     cash = float(rh_cash['cash'])
#     equity = float(rh_cash['equity'])
#     return(cash, equity)
def get_cash

           float          'margin_balances'  "day_trade_buying_power"
    return
def get_holdings_and_bought_price
                            0 for   in range 0  len
                                0 for   in range 0  len


    for       in
        try
                            int float                       'quantity'
                                float                       'average_buy_price'
        except
                            0
                                0

    return
#holdings, bought_price = get_holdings_and_bought_price(stocks)
def sell
#   #  sell_price = round((price-0.10), 2)
#     # un-comment when ready to trade on the live market
#    # print('sell is currently commented')
#     sell_order = rh.orders.order_sell_market(symbol=stock,
#                                              quantity=holdings,
#                                              timeInForce='gfd')

    print '### Trying to SELL {} at ${}'
```

```
# def buy(stock, allowable_holdings):
# #     #buy_price = round((price+0.10), 2)
# #     # un-comment when ready to trade on the live market
# #     print('buy is currently commented')
#     # buy_order = rh.orders.order_buy_market(symbol=stock,
#                                  #        quantity=allowable_holdings,
#                                     #    timeInForce='gfd')

#       buy_order = rh.orders.order_buy_fractional_by_price(symbol=stock,
#                                          ammountInDollars=allowable_holdings,
#                                           timeInForce='gtc',
#                                           extendedHours=False)
#       print('### Trying to BUY {} at ${}'.format(stock, price))

# rs.orders.order_buy_fractional_by_price(symbol,
#                                         ammountInDollars,

#rh.orders.order_buy_fractional_by_quantity(symbol,
                                #          quantity,
                               #           timeInForce='gtc',
                               #          extendedHours=False)

def build_dataframes
    import          as
               str                          8


    return

# dg = pd.read_csv(r'C:/Users/jizha/Desktop/three_strategy_buy_point_2022_3_4.csv')
# dg.columns
# g =dg['0'].tolist()
# l =g
# stocks = l



def get_historical_prices
    def price_5mins
            from          import          as
            import
            import
                      '86dd63f6b8ae774b061232685b78eb52'

              str
              str
              str
              str          -3
              int
                       '_'     '_'
                  '_'     '_'
                              f'https://financialmodelingprep.com/api/v3/historical-chart/5min/{st

          #  df1 = df1.set_index('date')
          #  df_price1 = df1.loc[(df1.index.day==e),:]
                                                'date'
                                  'close'         'float'
```

```
                                                    1
                                            'date'

        #    df_price = df_price.iloc[date, :]
                                                'close'

            return


    def close_price
        from          import          as
        import
        import
                  '86dd63f6b8ae774b061232685b78eb52'

          str
          str
          str           -3
            str           -4
                  '_'     '_'
              '_'     '_'
                          f'https://financialmodelingprep.com/api/v3/historical-price-full/{stock}
                              'historical'
       # stock = pd.DataFrame(bs)
                    'date' 'open' 'high' 'low' 'close' 'volume'
                                          'date'
                          'close'         'float'
                                             1
        #= df_2.iloc[:10, :]
                                      'close'
                               'date'
       # df_2 = df_2.sort_index()
        return


    for       in
        print

                                                        1



    for       in
        print

                                                        1

#stock = 'AAPL'

    for       in
                                        1
                                         2
                                    0
                                                    100

                                        'price_5min' 'price_10min' 'close_p' 'pct_change'
```

```
                                      'price_5min'              'price_10min'              'close_

                                                 0
                                                             1
#        final_data1['pct_change'] =round( final_data1['price_10m']/final_data1['close_p'],2)
    return
#stocks = final_trade_stock

def final_trade_stock


    for       in
        if                      'price_5min'                        'price_10min'   and




    for    in
          if 0          'pct_change'   5

    return

#d = final_trade_stock(stocks)
#stocks =  ticker_sum
#stocks =  final_trade_stock
#if __name__ == "__main__":
   # login(days=5)
#ticker_sum.remove('COUP')
#ticker_sum.remove('CTRN')
from          import          as
import

    str
    str
    str
#f.to_csv(r'C:\Users\jizha\Desktop\seabridge_datapool1\final_strategy_data_temporaly\three_strategy
# ticker = pd.read_csv(r'C:\Users\jizha\Desktop\seabridge_datapool1\final_strategy_data_temporaly\t

# ticker_sum =ticker['0'].tolist()
# l =g

#stocks = final_trade_stock
#ticker_sum = l
print 'stocks:'

#cash = myCash/10
#cash = 5000

#print('holdings:', holdings)
def buy
#     #buy_price = round((price+0.10), 2)
#     # un-comment when ready to trade on the live market
#     print('buy is currently commented')
    # buy_order = rh.orders.order_buy_market(symbol=stock,
                                   #        quantity=allowable_holdings,
                                      #    timeInForce='gfd')
```

```
                                                        'gfd'
                                                         False
      print '### Trying to BUY {} at ${}'

#cash = myCash/10
      1000
                          0 for   in range 0  len
                          0 for   in  range 0  len


#stocks = ['AAPL','FB']
   # while open_market():


#print('holdings:', holdings)


for          in enumerate
            float
    print '{} = ${}'
        len
                          round         1
    if                     0



#df_trades, df_prices = build_dataframes(df_trades, trade_dict, df_prices, price_dict)
  #  grapher.active_graph(grapher.normalize(df_prices), df_trades)

  #  time.sleep(30)




 #  logout()
from          import          as
import

    str
    str
    str

         r'C:\Users\jizha\Desktop\auto_robinhood_ticker\robin_buy_stocks_'   '_'   '_'   '.csv'
#C:\Users\jizha\Desktop\auto_robinhood_ticker
 #  logout()
```