

技术已成为金融行业中的战略资产。而传统的金融机构现在正在转型成为科技公司，而不仅仅是专注于该领域的金融方面。

数学算法带来了创新和速度，它们可以帮助我们在市场上获得竞争优势。金融交易的速度和频率以及庞大的数据量，已引起所有大型金融机构对技术的广泛关注。算法交易或定量交易是基于数学和统计分析设计和开发交易策略的过程。这是一个极其复杂的金融领域。

本教程是使用Python进行定量交易的初学者指南，如果读者是以下行业人士，将会发现这篇文章非常有帮助：1. 旨在成为基金或银行量化分析师

2. 计划开始自己的自营量化交易的人。我们将在这篇文章中介绍以下主题：1. 股票和交易基础

2. 从Quandl API提取数据
3. 股票价格数据的探索性数据分析
4. 移动平均线
5. 用Python制定交易策略
6. 可视化策略的表现

股票交易及交易策略

买卖现有和先前发行的股票的过程称为股票交易。有一个可以买卖股票的价格，并且这个价格根据股票市场的需求和供应不断波动。这将视公司的业绩和行为而定，股价可能会上下波动，但股价的波动不仅限于公司的业绩。交易者支付钱来换取公司的所有权，希望进行一些有利可图的交易并以更高的价格出售股票。交易者遵循的另一项重要技术是卖空股票。这涉及“借入股票并立即出售它们，以期以后以较低的价格购买它们，将其退还给贷方，并赚取相关的市场保证金。因此，大多数交易者遵循特定的量化策略和模型进行交易。而这被称为交易策略。

对冲基金和投资银行的量化交易者设计并开发了这些交易策略和框架来进行测试。

它需要深厚的编程专业知识，并且需要了解构建自己的策略所需的语言。

在C++，Java，R和MATLAB等语言中，Python是最流行的编程语言之一。由于其易于使用的语法，庞大的社区和第三方支持，它已在所有领域广泛采用，尤其是在数据科学领域。为了充分利用本教程，我们需要熟悉Python和统计信息。

从Quandl API提取数据

为了提取股票价格数据，我们将使用Quandl API。但在此之前，让我们设置工作环境：

在终端中，为项目创建一个新目录（根据需要命名）：

```
mkdir <目录名称>
```

确保在计算机上安装了Python 3和virtualenv。

使用virtualenv <env_name>创建一个新的Python3

virtualenv，并使用源<env_name> / bin / activate激活它。

现在，使用pip安装jupyter-notebook，然后在终端中输入pip install jupyter-notebook。

同样，安装pandas，quandl和numpy软件包。

从终端运行您的jupyter-notebook。

现在，笔记本应该可以在本地主机上运行，如以下屏幕截图所示：

您可以通过单击右侧的“新建”下拉列表来创建您的第一个笔记本。确保您已在Quandl上创建了一个帐户。请按照此处提到的步骤创建您的API密钥。一切准备就绪后，让我们直接进入：
导入所需的软件包
将熊猫(PANDAS)作为pd导入
将quandl导入为q
熊猫将成为本教程中使用最严格的软件包，因为我们将进行大量的数据处理和绘图。
导入软件包后，我们将使用Quandl软件包向Quandl API发出请求：
设置API密钥
q.ApiConfig.api_key = "<API密钥>"
发送获取请求以查询Microsoft从2010年1月1日至2019年1月1日的当日股票价格
msft_data = q.get("EOD / MSFT", start_date = "2010-01-01", end_date = "2019-01-01")
查看数据帧的前5行
msft_data.head()

这里有微软过去9年的EOD股票定价数据，我们所要做的就是从Quandl包中调用get方法，并提供股票代码，MSFT和所需数据的时间范围。

这真的很简单，对吧？让我们继续前进，以进一步了解和探索这些数据。股票定价数据的探索性数据分析。而掌握了数据之后，我们要做的第一件事就是了解数据代表什么以及封装什么样的信息。打印DataFrame的信息，我们可以看到其中包含的所有内容：

如上面的屏幕快照所示，DataFrame包含DatetimeIndex，这意味着我们正在处理时间序列数据。

索引可以被视为帮助我们修改或引用数据的数据结构。时间序列数据是在连续的，等间隔的时间间隔拍摄的价格快照的序列。在交易中，EOD股票价格数据捕获指定时间段内有关股票的某些参数（例如股票价格）的运动，并定期记录数据点。重要术语查看其他列，让我们

尝试了解每个列所代表的含义：开盘/收盘—捕获股票的开盘价/收盘价Adj_Open / Adj_Close-调整后的开盘/收盘价是指股票在任何给定交易日的价格，该价格已被修改为包括在第二天开盘前任何时间发生的任何股息分配，股票拆分和其他公司行为。交易量—它记录在任何给定交易日交易的股票数量。

高/低-
跟踪特定交易日中股票的最高和最低价格。

这些是我们此时将重点关注的重要列。我们可以了解数据的摘要统计信息，该统计信息显示行了数，均值，最大值，标准差等。尝试在Ipython单元格中运行以下代码行：`msft_data.describe()`

`resample()`

熊猫的resample () 方法用于简化时间序列数据的频率转换的控制和灵活性。我们可以指定时间间隔以将数据重新采样到每月，每季度或每年，并对其执行所需的操作。

```
msft_data.resample('M').mean()
```

这是一种分析不同时间范围内股票表现的有趣方式。

如何计算收益财务收益仅仅是投资产生或损失的钱。投资回报可以名义上表示为投资金额随时间的变化。可以将其计算为从利润与投资之比得出的百分比。为此，我们可以使用pct_change ()。这是计算收益的方法：# 导入numpy包将numpy导入为np # 将`Adj Close`分配给`daily_close`
daily_close = msft_data['Adj_Close'] # 作为分数变化返回daily_return = daily_close.pct_change () # 将NA值替换为0
daily_return.fillna (0, inplace = True) 打印 (daily_return) 这将打印股票每天产生的收益。将数字乘以100即可得出百分比变化。pct_change () 中使用的公式为：返回值 = $\frac{\{ (t \text{ 的价格}) - (t-1 \text{ 的价格}) \}}{\{ t-1 \text{ 的价格} \}}$ 现在，要计算月度回报，您需要做的是：mdata = msft_data.resample ('M')。apply (lambda x : x [-1]) month_return = mdata.pct_change () 将数据重新采样到几个月 (一个工作日) 后，我们可以使用apply () 函数获得当月的最后交易日。apply () 接受一个函数并将其应用于熊猫系列的每一行。lambda函数是Python中的匿名函数，可以不使用名称进行定义，并且仅采用以下格式的表达式：Lambda：表达例如，lambda x : x * 2是lambda函数。在此，x是自变量，x * 2是要求值并返回的表达式。交易中的移动平均线移动平均线的概念将为基于动量的交易策略奠定基础。在金融领域，分析师通常不得不在滑动的时间范围内不断评估统计指标，这被称为移动窗口计算。让我们看看如何计算50天的滚动平均值，然后将其滑动1天。滚动 () 这是神奇的功能，可以为我们提供窍门：# 将调整后的收盘价分配给adj_prices
adj_price = msft_data ['Adj_Close'] # 计算移动平均线
mav = adj_price.rolling (window = 50) .mean () # 打印结果打印 (mav [-10 :])

我们将在50天（约2个月）的时间范围内看到滚动平均值，移动平均值有助于消除数据中的任何波动或尖峰，并为策略的业绩提供更平滑的曲线。

我们可以绘图并看到差异：# 导入matplotlib包以查看图导入matplotlib.pyplot作为pltadj_price.plot（）

现在我们可以绘制滚动的mean（）：

mav.plot（）

读者可以自己了解差异，如何调取数据中的峰值以大致了解股票的表现。

制定交易策略

这是最后也是最有趣的部分：设计和制定交易策略。这将是开发基于动量的简单移动平均交叉（SMAC）策略的分步指南。基于动量的策略基于利用市场趋势持续性的技术指标。我们购买呈现上升趋势的证券，并卖出呈现下降趋势的卖空证券。SMAC策略是众所周知的示意动量策略。这是一个长期策略。动量是股票的总回报，包括过去n个月的股息。这n个月的时间称为回溯期。回溯期主要有3种类型：短期，中期和长期。我们需要定义特定时间序列的2个不同的回溯期。当较短的回溯滚动平均值（或移动平均值）超过较长的回溯移动平均值时，会生成购买信号。当较短的回

溯移动平均线

跌落至较长的移动平均线以

下时，就会出现卖出信号。

现在，让我们看看该策略的代码如何：# step1：初始化短期和长期回溯期

```
short_lb = 50long_lb = 120
```

```
# step2：使用信号列初始化一个名为signal_df的新数据框signal_df =
```

```
pd.DataFrame ( index = msft_data.index ) signal_df ['signal'] =
```

```
0.0 # step3：在较短的回溯期内创建一个简短的简单移动平均线signal_df
```

```

['short_mav'] = msft_data ['Adj_Close'].rolling ( window =
short_lb , min_periods = 1 , center = False ) .mean ( ) # step4 : 在较长的回
溯期内创建较长的简单移动平均线signal_df ['long_mav'] = msft_data
['Adj_Close'].rolling ( window = long_lb , min_periods = 1 , center =
False ) .mean ( ) # step5 : 根据条件语句生成信号signal_df ['signal']
[short_lb : ] = np.where ( signal_df ['short_mav'] [short_lb : ] > signal_df
['long_mav']
[short_lb : ] , 1.0 , 0.0 ) # step6 : 根据头寸栏创建交易订单signal_df
['positions'] = signal_df ['signal'].diff ( ) signal_df [signal_df ['positions']
== -1.0]让我们看看这里发生了什么。我们创建了2个回溯期。短期回溯期short_lb
为50天，而长期移动平均线的较长回溯期定义为120天的long_lb。我们创建了一个
新的DataFrame，用于捕获信号。只要使用np.where短移动平均线与长移动平均
线交叉，就会生成这些信号。如果条件为假，它将为true分配1.0和0.0。DataFram
e中的头寸列会告诉我们是否存在买入信号或卖出信号，或保持原样。我们基本上
是使用diff计算与前一行的信号栏中的差异。在那里，我们使用熊猫仅用6个步骤即
可实施我们的策略。很简单，不是吗？现在，让我们尝试使用Matplotlib形象化它
。我们需要做的是初始化一个绘图，将调整后的收盘价，空头和多头移动平均线添
加到绘图中，然后使用上方signal_df中的头寸列绘制买入和卖出信号：# 使用plt
初始化图无花果= plt.figure ( ) # 为y轴添加子图和标签plt1 =
fig.add_subplot ( 111 , ylabel = '价格为$' ) msft_data ['Adj_Close'].plot ( ax
= plt1 , color = 'r' , lw = 2。 ) # 绘制短期和长期回溯移动平均线signal_df
[['short_mav' , 'long_mav']]。plot ( ax = plt1 , lw = 2。 , figsize
= ( 12,8 ) ) # 绘制卖出信号plt1.plot ( signal_df.loc [signal_df.positions ==
-1.0] .index , signal_df.short_mav [signal_df.positions ==
-1.0] , 'v' , markersize = 10 , color
='k' ) # 绘制购买信号plt1.plot ( signal_df.loc [signal_df.positions == 1.0]
.index , signal_df.short_mav [signal_df.positions == 1.0] , '^' , markersize =
10 , color = 'm' ) # 显示plotplt.show
( ) 在Jupyter笔记本中运行上面的单元格将产生类似下面的图：

```

现在，我们可以清楚地看到，每当蓝线（短期均线）上升并超过橙线（长期均线）时，都有一个粉红色的向上标记指示买入信号。

卖出信号由一个黑色的向下标记表示，其中short_mav下降到long_mav以下。可视化全地形战略的绩效Quantopian是Zipline支持的平台，具有多种用例。您可以

编写自己的算法，访问免费数据，对策略进行回测，为社区做出贡献，并在需要资金时与Quantopian合作。

我们编写了一种算法来对SMA策略进行回测，结果如下：

以下是上述指标的说明：

1.总回报(Total Return)：从回测的开始到结束，投资组合的总回报百分比。

2. 特定收益 (Specific Return)：投资组合的总收益和普通收益之差。

3. 共同收益 (Common Return)：归因于共同风险因素的收益。这些回报构成了11个行业和5种风格风险因素。风险部分中的行业暴露和风格暴露图表提供了有关这些因素的更多详细信息。

4.夏普 (Sharpe)：6个月滚动夏普比率。它是风险调整投资的一种度量。通过将投资组合的超出无风险利率的超额收益除以投资组合的标准偏差来计算。

最大跌幅：投资组合历史上所有峰谷波动的最大跌幅。

波动率：投资组合收益的标准差