Mercado	Instrumentos
Acciones/Índices	Acciones, ETF, Futuros, Opciones
Divisas	Margen/Spot, ETFs, Futuros, Opciones
materias primas	Futuros, Opciones
Renta Fija	Futuros, Opciones

A los efectos de este libro, nos concentraremos casi exclusivamente en acciones y ETF para simplificar la implementación.

8.1.3 Datos fundamentales

Aunque los comerciantes algorítmicos realizan principalmente análisis de series temporales de precios financieros, a menudo también se agregan datos fundamentales (de frecuencias variables) al análisis. Así llamado *Valor cuantitativo* (Las estrategias QV) se basan en gran medida en la acumulación y el análisis de datos fundamentales, como información macroeconómica, historiales de ganancias corporativas, índices de inflación, informes de nómina, tasas de interés y presentaciones de la SEC. Dichos datos a menudo también están en formato temporal, aunque en escalas de tiempo mucho más grandes durante meses, trimestres o años. Las estrategias QV también operan en estos plazos.

Este libro no analizará en gran medida las estrategias QV o las estrategias impulsadas por fundamentales a gran escala de tiempo, sino que se concentrará en las estrategias diarias o más frecuentes derivadas principalmente de la acción del precio.

8.1.4 Datos no estructurados

Los datos no estructurados consisten en *documentos* como artículos de noticias, entradas de blog, documentos o informes. El análisis de tales datos puede ser complicado ya que se basa en *Procesamiento natural del lenguaje* (técnicas de PNL). Uno de esos usos del análisis de datos no estructurados es tratar de determinar la *sentimiento* contexto. Esto puede ser útil para impulsar una estrategia comercial. Por ejemplo, al clasificar los textos como "alcistas", "bajistas" o "neutrales", se podría generar un conjunto de señales comerciales. El término para este proceso es *análisis de los sentimientos*.

Python proporciona una biblioteca extremadamente completa para el análisis de datos de texto conocida como Natural Language Toolkit (NLTK). De hecho, se puede descargar un libro de O'Reilly sobre NLTK de forma gratuita a través del sitio web de los autores: Procesamiento del lenguaje natural con Python[3].

Datos de texto completo

Existen numerosas fuentes de datos de texto completo que pueden ser útiles para generar una estrategia comercial. Fuentes financieras populares como Bloomberg y Financial Times, así como blogs de comentarios financieros como Seeking Alpha y ZeroHedge, brindan importantes fuentes de texto para analizar. Además, las fuentes de noticias propietarias proporcionadas por los proveedores de datos también son buenas fuentes de dichos datos.

Para obtener datos a mayor escala, se deben utilizar herramientas de “web scraping”, que están diseñadas para automatizar la descarga de sitios web en masa. Tenga cuidado aquí, ya que las herramientas automatizadas de raspado web a veces infringen los Términos de servicio de estos sitios. Asegúrese de verificar antes de comenzar a descargar este tipo de datos. Una herramienta particularmente útil para web scraping, que hace que el proceso sea eficiente y estructurado, es la biblioteca ScraPy.

Datos de redes sociales

En los últimos años ha habido un gran interés en obtener información sobre sentimientos a partir de los datos de las redes sociales, particularmente a través del servicio de microblogging de Twitter. En 2011, se lanzó un fondo de cobertura en torno al sentimiento de Twitter, conocido como Derwent Capital. De hecho, estudios académicos[4] han mostrado evidencia de que es posible generar un grado de capacidad predictiva basada en dicho análisis de sentimientos.

Si bien el análisis de sentimientos está fuera del alcance de este libro si desea realizar una investigación sobre los sentimientos, hay dos libros [15, 14] de Matt Russell sobre la obtención de datos de redes sociales a través de las API públicas proporcionadas por estos servicios web.

8.2 Frecuencia de los datos

La frecuencia de los datos es una de las consideraciones más importantes al diseñar un sistema de comercio algorítmico. Impactará cada decisión de diseño con respecto al almacenamiento de datos, la prueba retrospectiva de una estrategia y la ejecución de un algoritmo.

Es probable que las estrategias de mayor frecuencia conduzcan a un análisis estadísticamente más sólido, simplemente debido a la mayor cantidad de puntos de datos (y, por lo tanto, operaciones) que se utilizarán. Las estrategias HFT a menudo requieren una importante inversión de tiempo y capital para el desarrollo del software necesario para llevarlas a cabo.

Las estrategias de menor frecuencia son más fáciles de desarrollar e implementar, ya que requieren menos automatización. Sin embargo, a menudo generarán muchas menos transacciones que una estrategia de mayor frecuencia, lo que lleva a un análisis estadísticamente menos sólido.

8.2.1 Datos semanales y mensuales

Los datos fundamentales a menudo se informan semanalmente, mensualmente, trimestralmente o incluso anualmente. Dichos datos incluyen datos de nómina, informes de rendimiento de fondos de cobertura, presentaciones ante la SEC, índices basados en la inflación (como el índice de precios al consumidor, CPI), crecimiento económico y cuentas corporativas.

El almacenamiento de dichos datos a menudo es adecuado para bases de datos no estructuradas, como MongoDB, que puede manejar datos anidados jerárquicamente y, por lo tanto, permite un grado razonable de capacidad de consulta. La alternativa es almacenar texto de archivo sin formato en un RDBMS, que es menos apropiado, ya que la consulta de texto completo es más complicada.

8.2.2 Datos diarios

La mayoría de los operadores algorítmicos minoristas utilizan datos de series temporales financieras diarias ("fin del día"/EOD), en particular en acciones y divisas. Dichos datos están disponibles gratuitamente (ver más abajo), pero a menudo son de calidad cuestionable y están sujetos a ciertos sesgos. Los datos al final del día a menudo se almacenan en RDBMS, ya que la naturaleza del mapeo de ticker/símbolo se aplica naturalmente al modelo relacional.

Los datos EOD no implican requisitos de almacenamiento particularmente grandes. Hay 252 días de negociación en un año para las bolsas estadounidenses y, por lo tanto, durante una década habrá 2.520 barras por valor. Incluso con un universo de 10.000 símbolos, son 25.200.000 barras, que pueden manejarse fácilmente dentro de un entorno de base de datos relacional.

8.2.3 Barras Intradía

Las estrategias intradía a menudo hacen uso de barras OHLCV por hora, quince, cinco, un minuto o segundo. Los proveedores de feeds intradía, como QuantQuote y DTN IQFeed, a menudo proporcionan barras por minuto o por segundo en función de sus datos de ticks.

Los datos en tales frecuencias tendrán muchas barras "faltantes" simplemente porque no se realizaron transacciones en ese período de tiempo. Los pandas se pueden usar para aumentar estos valores, aunque con una disminución en la precisión de los datos. Además, los pandas también se pueden usar para crear datos en escalas de tiempo menos granulares si es necesario.

Durante un período de diez años, los datos minuciosos generarán casi un millón de barras por valor. De manera similar, para los datos secundarios, el número de puntos de datos durante el mismo período totalizará casi sesenta millones por valor. Por lo tanto, almacenar mil de tales valores conducirá a sesenta mil millones de barras de datos. Esta es una gran cantidad de datos que deben mantenerse en un RDBMS y, en consecuencia, se requieren enfoques más sofisticados.

El almacenamiento y la recuperación de datos secundarios de esta magnitud están algo fuera del alcance de este libro, por lo que no lo discutiré más.

8.2.4 Datos de Tick y Libro de Órdenes

Cuando una operación se completa en una bolsa u otro lugar, una *garrapata* es generado. Los feeds de ticks consisten en todas esas transacciones *por intercambio*. Los feeds de ticks minoristas se almacenan y cada dato tiene una marca de tiempo precisa al nivel de milisegundos. Los datos de ticks a menudo también incluyen el mejor precio de compra/venta actualizado. El almacenamiento de datos de ticks está mucho más allá del alcance de este libro, pero no hace falta decir que

los volúmenes de tales datos son sustanciales. Los mecanismos de almacenamiento comunes incluyen HDF5, kdb y simplemente archivos planos/CSV.

Múltiple *órdenes limitadas* en un intercambio conducen al concepto de un *libro de pedidos*. Esta es esencialmente la lista de todas las órdenes límite de oferta y demanda en ciertos volúmenes para cada participante del mercado. Conduce a la definición de la *diferencial de oferta y demanda* (o simplemente el "spread"), que es la diferencia más pequeña en los precios de compra y venta para las órdenes "top of book". La creación de una representación histórica, o un simulador de mercado, de un libro de órdenes limitadas suele ser necesario para llevar a cabo estrategias de negociación de ultra alta frecuencia (UHFT). El almacenamiento de tales datos es complejo y, como tal, estará fuera del alcance de este libro.

8.3 Fuentes de datos

Existen numerosas fuentes y proveedores de datos financieros. Varían sustancialmente en amplitud, puntualidad, calidad y precio.

En términos generales, los datos del mercado financiero proporcionados con una frecuencia diaria diferida o más larga están disponibles libremente, aunque con una calidad general dudosa y el potencial de sesgo de supervivencia. Para obtener datos intradiarios suele ser necesario adquirir un feed de datos comercial. Los proveedores de tales fuentes varían enormemente en su capacidad de servicio al cliente, calidad general de la fuente y amplitud de instrumentos.

8.3.1 Fuentes gratuitas

Los datos de barra gratuitos al final del día, que consisten en precios de instrumentos Open-High-Low-Close-Volume (OHLCV), están disponibles para una amplia gama de acciones y futuros estadounidenses e internacionales de Yahoo Finance, Google Finance y Quandl.

Yahoo Finanzas

Yahoo Finance es el recurso "ir a" cuando se forma una base de datos de acciones de EE. UU. al final del día. La amplitud de los datos es extremadamente completa y enumera miles de acciones negociadas. Además, las divisiones de acciones y los dividendos se manejan mediante un método de ajuste retroactivo, que surge como la columna "Adj Close" en la salida CSV de la API (que analizamos a continuación). Por lo tanto, los datos permiten a los comerciantes algorítmicos comenzar rápidamente y sin costo alguno.

Personalmente, tengo mucha experiencia en la limpieza de datos de Yahoo. Tengo que comentar que los datos pueden ser bastante erróneos. En primer lugar, está sujeto a un problema conocido como *relleno*. Este problema ocurre cuando los datos históricos pasados se corrigen en una fecha futura, lo que genera backtests de baja calidad que cambian a medida que se actualiza su propia base de datos. Para manejar este problema, generalmente se agrega un registro de registro al maestro de valores (en una tabla de registro adecuada) cada vez que se modifica un punto de datos históricos.

En segundo lugar, el feed de Yahoo solo agrega precios de unas pocas fuentes para formar los puntos OHLCV. Esto significa que los valores alrededor de la apertura, el máximo, el mínimo y el cierre pueden ser engañosos, ya que otras bolsas/fuentes de liquidez pueden haber ejecutado precios diferentes por encima de los valores.

En tercer lugar, he notado que al obtener datos financieros *en masa* de Yahoo, que los errores se filtran en la API. Por ejemplo, varias llamadas a la API con parámetros de fecha/ticker idénticos ocasionalmente conducen a diferentes conjuntos de resultados. Este es claramente un problema sustancial y debe ser revisado cuidadosamente.

En resumen, prepárese para llevar a cabo una limpieza de datos exhaustiva en los datos de Yahoo Finance, si elige usarlos para completar un maestro de valores grande y necesita datos muy precisos.

Quandl

Quandl es un servicio relativamente nuevo que pretende ser *"La forma más fácil de encontrar y utilizar datos numéricos en la web"*. ¡Creo que están bien encaminados para lograr ese objetivo! El servicio proporciona un importante conjunto de datos diarios de acciones estadounidenses e internacionales, tasas de interés, materias primas/futuros, divisas y otros datos económicos. Además, la base de datos se amplía continuamente y el proyecto se mantiene de forma muy activa.

Se puede acceder a todos los datos mediante una API HTTP muy moderna (CSV, JSON, XML o HTML), con complementos para una amplia variedad de lenguajes de programación, incluidos R, Python, Matlab, Excel, Stata, Maple, C#, EViews, Java , C/C++, .NET, Clojure y Julia. Sin una cuenta, se permiten 50 llamadas a la API por día, pero esto se puede aumentar a 500 si se registra una cuenta. De hecho, las llamadas se pueden actualizar a 5000 por hora si así lo desea poniéndose en contacto con el equipo.

No he tenido mucha experiencia con Quandl "a escala", por lo que no puedo comentar sobre el nivel de errores dentro del conjunto de datos, pero tengo la sensación de que es probable que cualquier error se informe y corrija constantemente. Por lo tanto, vale la pena considerarlos como una fuente de datos principal para un maestro de valores al final del día.

Más adelante en el capítulo, analizaremos cómo obtener datos de futuros de productos básicos de EE. UU. de Quandl con Python y pandas.

8.3.2 Fuentes Comerciales

Para llevar a cabo el comercio algorítmico intradiario, generalmente es necesario comprar un feed comercial. El precio puede oscilar entre $ 30 por mes y alrededor de $ 500 por mes para feeds de "nivel minorista". Los feeds de calidad institucional a menudo estarán en el rango bajo a medio de cuatro cifras por mes y, como tal, no los discutiré aquí.

EODData

He utilizado EODData en un contexto de fondos, aunque solo con datos diarios y predominantemente para divisas. A pesar de su nombre, proporcionan cierto grado de fuentes intradía. El costo es de $ 25 por mes para su paquete "platino".

El recurso es muy útil para encontrar una lista completa de los símbolos negociados en los intercambios globales, pero recuerde que esto estará sujeto al sesgo de supervivencia, ya que creo que la lista representa las entidades cotizadas actuales.

Desafortunadamente (al menos en 2010) descubrí que el feed de división de acciones era algo inexacto (al menos en comparación con la información de Morningstar). Esto condujo a algunos problemas de picos sustanciales (ver a continuación) en los datos, lo que aumentó la fricción en el proceso de limpieza de datos.

DTN IQFeed

DTN IQFeed es una de las fuentes de datos más populares para el comerciante algorítmico minorista de alto nivel. Afirman tener más de 80.000 clientes. Proporcionan datos en tiempo real tick-by-tick sin filtrar del intercambio, así como una gran cantidad de datos históricos.

El precio comienza en $ 50 por mes, pero en realidad estará en el rango de $ 150- $ 200 por mes una vez que se seleccionen los servicios particulares y se tengan en cuenta las tarifas de cambio. Utilizo DTN IQFeed para todas mis estrategias de futuros y acciones intradía. En términos de datos históricos, IQFeed proporciona acciones, futuros y opciones:

- 180 días naturales de tick (cada operación)
- Más de 7 años de barras históricas de 1 minuto
- Más de 15 años de barras históricas diarias

La principal desventaja es que el software DTN IQFeed (el mini servidor, no las herramientas de gráficos) solo se ejecutará en Windows. Esto puede no ser un problema si todo su comercio algorítmico se lleva a cabo en este sistema operativo, pero personalmente desarrollo todas mis estrategias en Ubuntu Linux. Sin embargo, aunque no lo he probado activamente, he oído que es posible ejecutar DTN IQFeed bajo el emulador WINE.

A continuación, discutiremos cómo obtener datos de IQFeed usando Python en Windows.

Cuantificación

QuantQuote proporciona datos históricos a nivel de minutos, segundos y ticks a precios razonables para las acciones de EE. UU. que se remontan a 1998. Además, proporcionan feeds de ticks en tiempo real a nivel institucional, aunque esto es de menor interés para los operadores algorítmicos minoristas. Uno de los principales beneficios de QuantQuote es que sus datos se proporcionan sin sesgo de supervivencia, debido a su software de coincidencia de símbolos TickMap y la inclusión de todas las acciones dentro de un índice determinado a lo largo del tiempo.

Como ejemplo, para comprar la historia completa del S&P500 desde 1998 en barras de minutos, incluidas las acciones que no cotizan en bolsa, el costo en el momento de escribir este artículo era de $895. Las escalas de precios con el aumento de la frecuencia de los datos.

QuantQuote es actualmente el principal proveedor de datos de mercado para el servicio de backtesting basado en la web de QuantConnect. QuantQuote hace todo lo posible para garantizar la minimización de errores, por lo que si está buscando un feed de acciones de EE. UU. Solo en alta resolución, entonces debería considerar usar su servicio.

8.4 Obtención de datos

En esta sección, analizaremos cómo usar Quandl, pandas y DTN IQFeed para descargar datos del mercado financiero en una variedad de mercados y períodos de tiempo.

8.4.1 Yahoo Finanzas y Pandas

La biblioteca de pandas simplifica enormemente la descarga de datos EOD de Yahoo Finance. Pandas se envía con un componente DataReader que se vincula con Yahoo Finance (entre otras fuentes). Especificar un símbolo con una fecha de inicio y finalización es suficiente para descargar una serie EOD en un DataFrame de pandas, lo que permite realizar operaciones vectorizadas rápidas:

```
de__futuro__importarimprimir_funcion

importarfecha y hora
importarpandas.io.data como web

si__nombre__ == "__principal__":
    espía = web.DataReader(
        "ESPÍA", "yahoo",
        fechahora.fechahora(2007,1,1),
        fechahora.fechahora(2015,6,15)
    )
    impresión(espía.cola())
```

La salida se da a continuación:

```
                Abierto       Alto        Bajo       Cerca      Volumen \
Fecha
2015-06-09 208.449997 209.100006 207.690002 208.449997 2015-06-10 209.369995 98148200
211.410004 209.300003 210.960007 129936200 2015-06-11 211.479996 212.089996
211.199997 211.649994 72672100 2015-06-12 210.639999 211.479996 209.679993
209.929993 127811900 2015-06-15 208.639999 209.449997 207.789993 209.100006
121425800

                 cerrar
Fecha
2015-06-09  208.449997
2015-06-10  210.960007
2015-06-11  211.649994
2015-06-12  209.929993
2015-06-15  209.100006
```

*Tenga en cuenta que en pandas 0.17.0,**pandas.io.datos**será reemplazado por uno separado**pandas-lector de datos** paquete. Sin embargo, por el momento (es decir, pandas versiones 0.16.x) la sintaxis para importar los datos el lector es**importar**pandas.io.data como web*.

En la siguiente sección, utilizaremos Quandl para crear una solución de descarga permanente más completa.

8.4.2 Quandl y pandas

Hasta hace poco, era bastante difícil y costoso obtener datos de futuros consistentes en los intercambios de manera actualizada con frecuencia. Sin embargo, el lanzamiento del servicio Quandl ha cambiado la situación drásticamente, con datos financieros que en algunos casos se remontan a la década de 1950. En esta sección, utilizaremos Quandl para descargar un conjunto de contratos de futuros al final del día en varias fechas de entrega.

Registrarse en Quandl

Lo primero que debe hacer es registrarse en Quandl. Esto aumentará la asignación diaria de llamadas a su API. El registro otorga 500 llamadas por día, en lugar de las 50 predeterminadas. Visite el sitio en www.quandl.com:

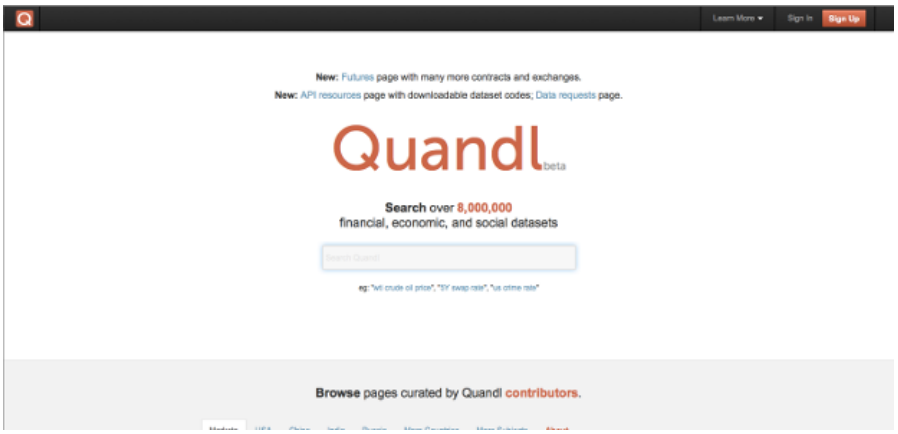


Figura 8.1: La página de inicio de Quandl

Haga clic en el botón de registro en la parte superior derecha:

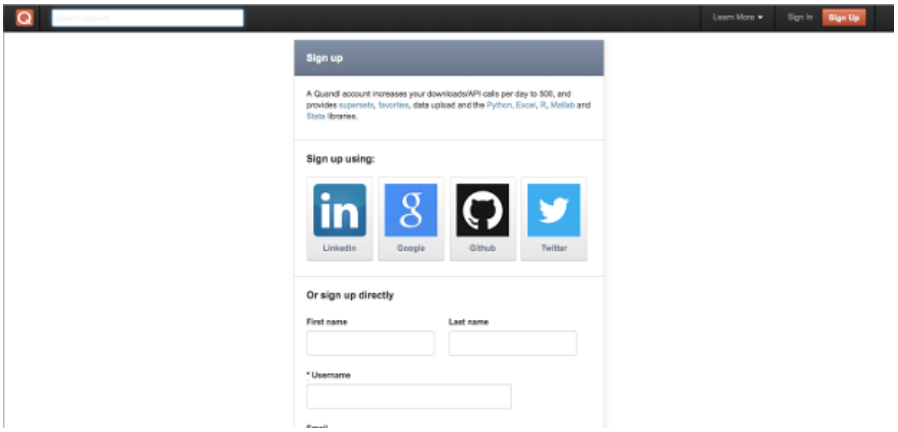


Figura 8.2: La página de registro de Quandl

Una vez que haya iniciado sesión, volverá a la página de inicio:

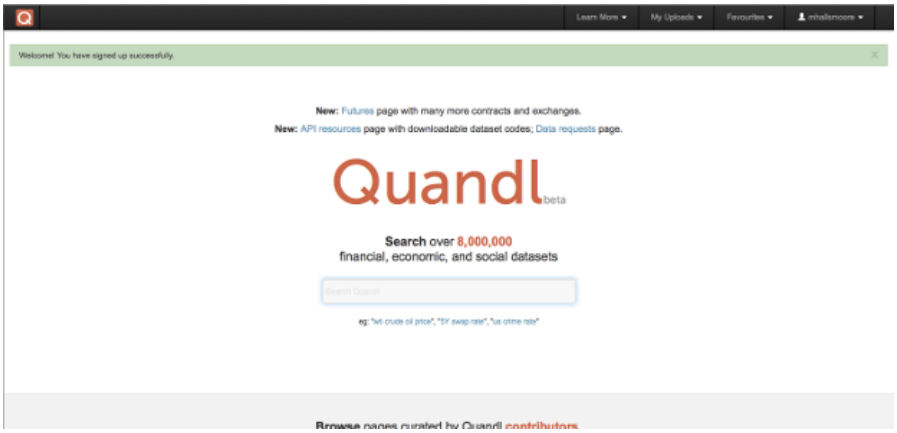


Figura 8.3: La página de inicio autorizada de Quandl

Datos de futuros de Quandl

Ahora haga clic en el enlace "Nuevo: página de futuros..." para ir a la página de inicio de futuros:

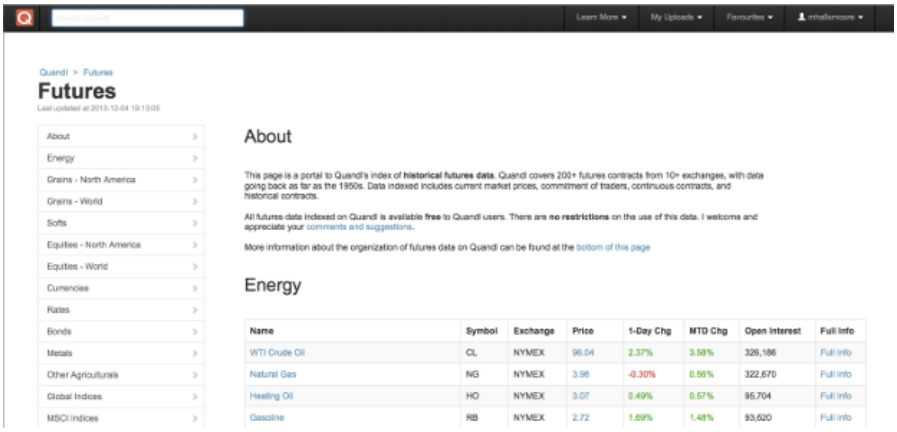


Figura 8.4: La página de inicio de los contratos de futuros de Quandl

Para este tutorial consideraremos el contrato de futuros E-Mini S&P500 altamente líquido, que tiene el símbolo de futuros ES. Para descargar otros contratos, el resto de este tutorial se puede realizar con símbolos adicionales que reemplazan la referencia a ES.

Haga clic en el enlace E-Mini S&P500 (o el símbolo de futuros elegido) y accederá a la siguiente pantalla:

Desplazarse hacia abajo en la pantalla muestra la lista de contratos históricos que se remontan a 1997:

Haga clic en uno de los contratos individuales. Como ejemplo, he elegido ESZ2014, que se refiere al contrato para la 'entrega' de diciembre de 2014. Esto mostrará un gráfico de los datos:

Al hacer clic en el botón "Descargar", los datos se pueden obtener en múltiples formatos: HTML, CSV, JSON o XML. Además, podemos descargar los datos directamente en un DataFrame de pandas utilizando los enlaces de Python. Si bien este último es útil para la "creación de prototipos" rápidos y la exploración de los datos, en esta sección estamos considerando el desarrollo de un almacén de datos a más largo plazo. Haga clic en el botón de descarga, seleccione "CSV" y luego copie y pegue la llamada API:

La llamada a la API tendrá la siguiente forma:

```
http://www.quandl.com/api/v1/datasets/OFDP/FUTURE_ESZ2014.csv?
&auth_token=MI_AUTH_TOKEN&trim_start=2013-09-18
&trim_end=2013-12-04&sort_order=desc
```

El token de autorización se ha redactado y reemplazado por MY_AUTH_TOKEN. Será necesario copiar la cadena alfanumérica entre "auth_token=" y "&trim_start" para

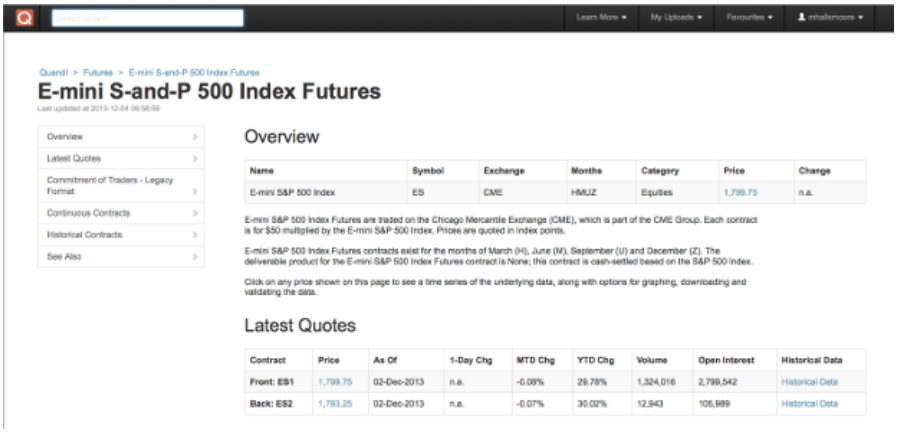


Figura 8.5: Página de contrato de E-Mini S&P500

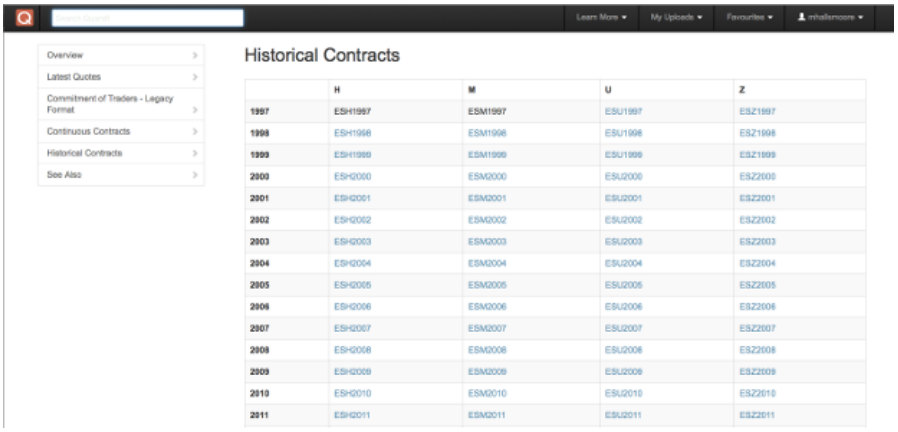


Figura 8.6: Contratos históricos de E-Mini S&P500

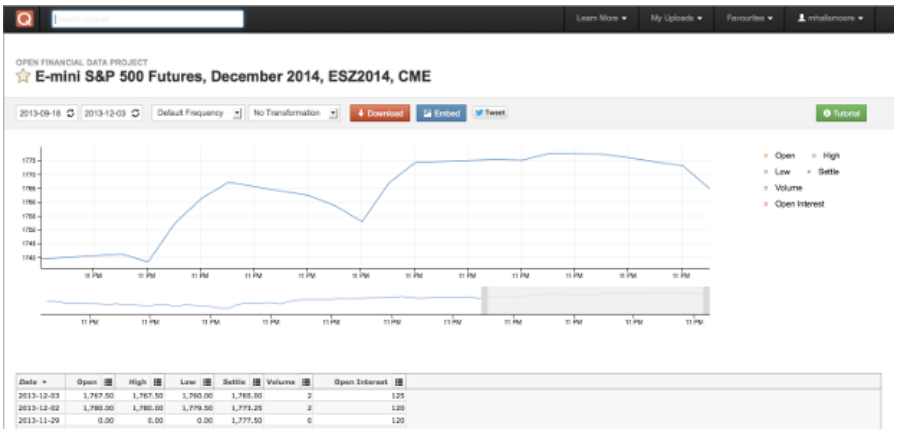


Figura 8.7: Gráfico de ESZ2014 (entrega en diciembre de 2014)

uso posterior en el script de Python a continuación. No lo comparta con nadie, ya que es su token de autorización único para las descargas de Quandl y se usa para determinar su tasa de descarga del día.

Esta llamada a la API formará la base de un script automatizado que escribiremos a continuación para descargar un subconjunto del contrato de futuros histórico completo.

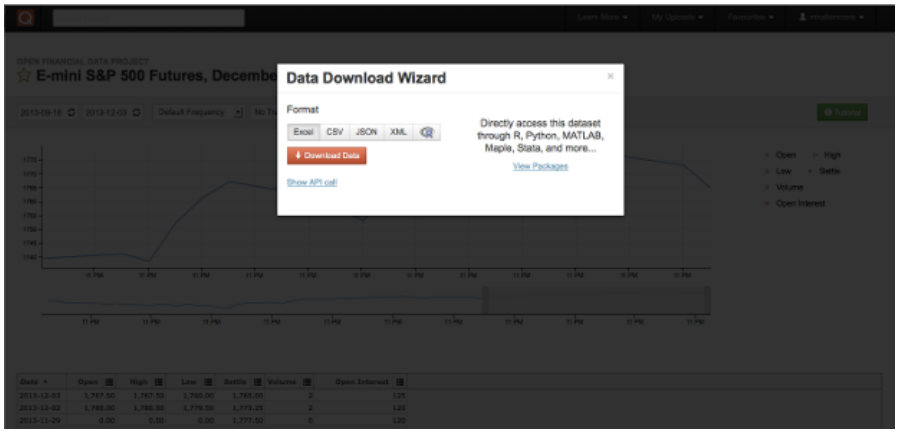


Figura 8.8: Descargar modal para archivo CSV ESZ2014

Descargar Quandl Futures en Python

Debido a que estamos interesados en utilizar los datos de futuros a largo plazo como parte de una estrategia de base de datos maestra de valores más amplia, queremos almacenar los datos de futuros en el disco. Por lo tanto, necesitamos crear un directorio para almacenar nuestros archivos CSV de contrato E-Mini. En Mac/Linux (dentro de la terminal/consola) esto se logra con el siguiente comando:

```
cd /RUTA/HACIA/TU/quandl_data.py
mkdir -p quandl/futures/ES
```

Nota: Reemplazar /RUTA/HACIA/TUarriba con el directorio donde está suquandl_data.py file está ubicado.

Esto crea un subdirectorio de llamadoquandl,que contiene dos subdirectorios más para futuros y para los contratos ES en particular. Esto nos ayudará a organizar nuestras descargas de manera continua.

Para poder realizar la descarga mediante Python necesitaremos importar algunas librerías. En particular necesitaremospeticionespara la descarga ypandasymatplotlibpara el trazado y la manipulación de datos:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

# quandl_data.py

de__futuro__importarimprimir_funcion

importarmatplotlib.pyplot como plt
importarpandas como pd importar
peticiones
```

La primera función dentro del código generará la lista de símbolos de futuros que deseamos descargar. Agregué parámetros de palabras clave para los años de inicio y finalización, configurándolos en valores razonables de 2010 y 2014. Por supuesto, puede optar por utilizar otros períodos de tiempo:

```
definitivamenteconstruir_futuros_simbolos(
        símbolo, start_year=2010, end_year=2014
    ) :
    """
    Construye una lista de códigos de contratos de futuros
    para un símbolo y marco de tiempo en particular.
    """
    futuros = []
```

```
    # marzo, junio, septiembre y
    # códigos de entrega de
    diciembre meses = 'HMUZ'
    poryenrango (año_inicial, año_final+1):
        pormetroenmeses:
            futuros.append("%s%s%s" % (símbolo, m, y)) devolver
            futuros
```

Ahora necesitamos recorrer cada símbolo, obtener el archivo CSV de Quandl para ese contrato en particular y luego escribirlo en el disco para que podamos acceder a él más tarde:

```
definitivamentedescargar_contrato_desde_quandl(contrato, dl_dir):
    """
    Descargue un contrato de futuros individual de Quandl y luego guárdelo en
    el disco en el directorio 'dl_dir'. Se requiere un auth_token, que se obtiene de
    Quandl al registrarse. """

    # Construya la llamada API desde el contrato y auth_token api_call =
    "http://www.quandl.com/api/v1/datasets/" api_call += "OFDP/
    FUTURE_%s.csv" % contrato
    # Si desea agregar un token de autenticación para más descargas, simplemente
    # comenta la siguiente línea y reemplaza MY_AUTH_TOKEN con
    # su token de autenticación en la línea de
    abajo params = "?sort_order=asc"
    # params = "?auth_token=MY_AUTH_TOKEN&sort_order=asc"
    full_url = "%s%s" % (api_call, parámetros)

    # Descarga los datos de Quandl datos =
    solicitudes.get(url_completa).texto

    # Almacenar los datos en el disco
    fc = open('%s/%s.csv' % (dl_dir, contrato), 'w') fc.write(datos)

    fc.cerrar()
```

Ahora unimos las dos funciones anteriores para descargar todos los contratos deseados:

```
definitivamentedescargar_contratos_históricos(
        símbolo, dl_dir, start_year=2010, end_year=2014
    ) :
    """
    Descarga todos los contratos de futuros para un símbolo específico
    entre start_year y end_year.
    """
    contratos = construct_futures_symbols(
        símbolo, año_inicio, año_fin
    )
    porCencontratos:
        impresión("Contrato de descarga: %s" % c)
        download_contract_from_quandl(c, dl_dir)
```

Finalmente, podemos agregar uno de los precios de futuros a un marco de datos de pandas usando la función principal. Luego podemos usar matplotlib para trazar el precio de liquidación:

```
si__nombre__ == "__principal__":
    símbolo = 'ES'

    # Asegúrate de haber creado esto
    # directorio relativo de antemano
```

```
dl_dir = 'quandl/futuros/ES'

# Crear los años de inicio y fin año_inicio
= 2010
fin_año = 2014

# Descargar los contratos en el directorio
descargar_contratos_históricos(
        símbolo, dl_dir, start_year, end_year
)

# Abra un solo contrato a través de read_csv
# y trazar el precio de liquidación es
= pd.io.parsers.read_csv(
        "%s/ESH2010.csv" % dl_dir, index_col="Fecha"
)
es["Conciliar"].plot()
plt.mostrar()
```

El resultado de la gráfica se muestra en la Figura 8.4.2.

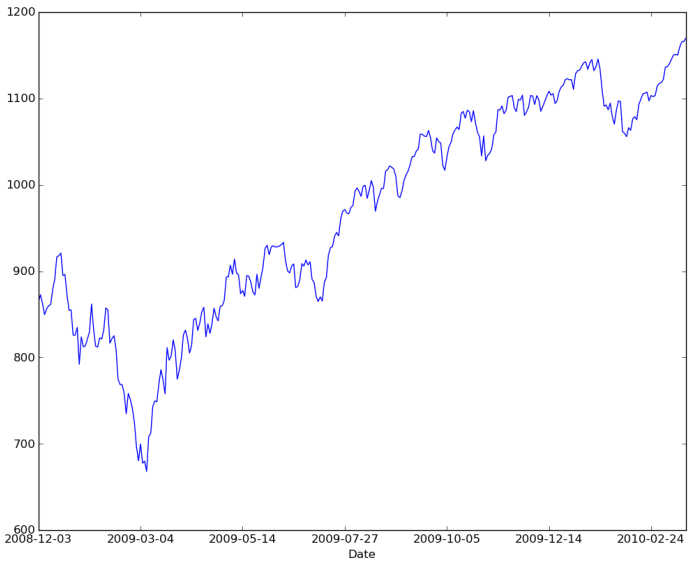


Figura 8.9: Precio de liquidación ESH2010

El código anterior se puede modificar para recopilar cualquier combinación de contratos de futuros de Quandl según sea necesario. Recuerde que, a menos que se realice una solicitud de API superior, el código se limitará a realizar 50 solicitudes de API por día.

8.4.3 DTN IQFeed

Para aquellos de ustedes que poseen una suscripción a DTN IQFeed, el servicio proporciona un mecanismo cliente-servidor para obtener datos intradía. Para que esto funcione es necesario descargar el servidor IQLink y ejecutarlo en Windows. Desafortunadamente, es complicado ejecutar este servidor en Mac o Linux a menos que utilice el emulador WINE. Sin embargo, una vez que el servidor se está ejecutando, se puede conectar a través de un socket, momento en el que se pueden consultar los datos.

En esta sección, obtendremos datos de barra minuciosamente para un par de ETF de EE. UU. desde el 1 de enero de 2007 en adelante utilizando una interfaz de socket de Python. Dado que hay aproximadamente 252 días hábiles dentro de

cada año para los mercados estadounidenses, y cada día de negociación tiene 6,5 horas de negociación, esto equivaldrá a al menos 650 000 barras de datos, cada una con siete puntos de datos: Marca de tiempo, Apertura, Mínimo, Máximo, Cierre, Volumen e Interés abierto.

Elegí los ETF de SPY e IWM para descargarlos en CSV. Haga lo siguiente para iniciar el programa IQLink en Windows antes de ejecutar este script:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

# iqfeed.py

importarsistema
importarenchufe

definitivamenteread_historical_data_socket(calcetín, recv_buffer=4096):
    """
    Lea la información del socket, en forma de búfer, recibiendo solo
    4096 bytes a la vez.

    Parámetros:
    calcetín - El objeto del zócalo
    recv_buffer - Cantidad en bytes a recibir por lectura """

    búfer = ""
    datos = ""
    tiempoVerdadero:
        datos = calcetín.recv(recv_buffer) búfer
        += datos

        # Comprobar si llega la cadena del mensaje final si"!
        ENDMSG!"enbuffer:
            descanso

    # Eliminar la cadena del mensaje final
    búfer = búfer[:-12]
    devolverbuffer

si__nombre__ == "__principal__":
    # Defina el host del servidor, el puerto y los símbolos para
    descargar servidor = "127.0.0.1"#servidor local
    puerto = 9100#Puerto de socket de datos históricos
    syms = ["SPY", "IWM"]

    # Descarga cada símbolo al disco por
    simensims:
        impresión"Símbolo de descarga: %s..." % sym

        # Construya el mensaje que necesita IQFeed para recuperar datos mensaje =
        "HIT,%s,60,20070101 075000,,,093000,160000,1\n" % sim

        # Abra un socket de transmisión al servidor IQFeed localmente calcetín
        = socket.socket(socket.AF_INET, socket.SOCK_STREAM) sock.connect((host,
        puerto))

        # Enviar la solicitud de datos históricos
        # mensaje y almacenamiento en búfer de los
        datos calcetín.sendall(mensaje)
```

```
data = read_historical_data_socket(sock) sock.close


# Eliminar todas las líneas finales y finales de línea
# delimitador de coma de cada registro datos
= "".join(datos.dividir("\r")) datos =
datos.reemplazar(",\n","\n")[:-1]

# Escribir el flujo de datos en el disco f =
abrir("%s.csv" % sym, "w") f.write(datos)

f.cerrar()
```

Con opciones de suscripción adicionales en la cuenta DTN IQFeed, es posible descargar contratos de futuros individuales (y contratos continuos retroajustados), opciones e índices. DTN IQFeed también proporciona transmisión de ticks en tiempo real, pero esta forma de datos queda fuera del alcance de este libro.

8.5 Limpieza de datos financieros

Posterior a la entrega de datos financieros de los proveedores, es necesario realizar *limpieza de datos*. Desafortunadamente, este puede ser un proceso laborioso, pero muy necesario. Hay varios problemas que requieren resolución: datos incorrectos, consideración de agregación de datos y reposición. Los contratos de acciones y futuros poseen sus propios desafíos únicos que deben abordarse antes de la investigación de la estrategia, incluido el ajuste hacia atrás/adelante, la unión continua de contratos y el manejo de acciones corporativas.

8.5.1 Calidad de los datos

La reputación de un proveedor de datos a menudo se basa en la calidad (percibida) de sus datos. En términos simples, los datos incorrectos o faltantes conducen a señales comerciales erróneas y, por lo tanto, a pérdidas potenciales. A pesar de este hecho, muchos proveedores todavía se ven afectados por una calidad de datos deficiente o inconsistente. Por lo tanto, siempre es necesario llevar a cabo un proceso de limpieza.

Los principales culpables de la mala calidad de los datos son los datos contradictorios/incorrectos, la agregación opaca de múltiples fuentes de datos y la corrección de errores ("relleno").

Datos contradictorios e incorrectos

Los datos incorrectos pueden ocurrir en cualquier parte de la transmisión. Los errores en el software de un intercambio pueden dar lugar a precios erróneos al igualar operaciones. Esto se filtra al vendedor y, posteriormente, al comerciante. Los proveedores acreditados intentarán marcar los "ticks malos" aguas arriba y, a menudo, dejarán la "corrección" de estos puntos al comerciante.

8.5.2 Contratos de Futuros Continuos

En esta sección vamos a discutir las características de los contratos de futuros que presentan un desafío de datos desde el punto de vista del backtesting. En particular, la noción de "contrato continuo". Describiremos las principales dificultades de los futuros y proporcionaremos una implementación en Python con pandas que pueden aliviar parcialmente los problemas.

Breve resumen de los contratos de futuros

Los futuros son una forma de *contrato* elaborado entre dos partes para la compra o venta de una cantidad de un activo subyacente en una fecha específica en el futuro. Esta fecha es conocida como la *entrega* o *vencimiento* . Cuando se alcance esta fecha, el comprador deberá entregar el subyacente físico (o su equivalente en efectivo) al vendedor por el precio pactado en la fecha de formación del contrato.

En la práctica, los futuros se negocian en bolsas (a diferencia de*En el mostrador*-OTC trading) para cantidades y calidades estandarizadas del subyacente. los precios son*marcado para el mercado*todos los días. Los futuros son increíblemente líquidos y se utilizan mucho con fines especulativos. Si bien los futuros se utilizaron a menudo para cubrir los precios de los productos agrícolas o industriales, se puede formar un contrato de futuros sobre cualquier subyacente tangible o intangible, como índices bursátiles, tasas de interés o valores de divisas.

Puede encontrar una lista detallada de todos los códigos de símbolos utilizados para los contratos de futuros en varios intercambios en el sitio de datos de CSI: Hoja informativa de futuros.

La principal diferencia entre un contrato de futuros y la propiedad de acciones es el hecho de que un contrato de futuros tiene una ventana limitada de disponibilidad en virtud de la fecha de vencimiento. En cualquier momento habrá una variedad de contratos de futuros sobre el mismo subyacente, todos con fechas de vencimiento variables. El contrato con la fecha de vencimiento más cercana se conoce como el*cerca de contrato*. El problema al que nos enfrentamos como comerciantes cuantitativos es que en cualquier momento tenemos la opción de múltiples contratos con los que operar. Por lo tanto, estamos tratando con un conjunto superpuesto de series de tiempo en lugar de un flujo continuo como en el caso de las acciones o las divisas.

El objetivo de esta sección es delinear varios enfoques para construir un flujo continuo de contratos a partir de este conjunto de series múltiples y resaltar las ventajas y desventajas asociadas con cada técnica.

Formación de un contrato de futuros continuo

La principal dificultad de tratar de generar un contrato continuo a partir de los contratos subyacentes con entregas variables es que los contratos a menudo no se negocian a los mismos precios. Por lo tanto, surgen situaciones en las que no proporcionan un empalme uniforme de uno a otro. Esto se debe a los efectos de contango y de retroceso. Hay varios enfoques para abordar este problema, que ahora discutiremos.

Desafortunadamente, no existe un único método "estándar" para unir contratos de futuros en la industria financiera. En última instancia, el método elegido dependerá en gran medida de la estrategia que emplee los contratos y el método de ejecución. A pesar de que no existe un método único, existen algunos enfoques comunes:

losAjuste atrás/adelante ("Panamá")El método alivia la "brecha" entre múltiples contratos cambiando cada contrato de manera que las entregas individuales se unan sin problemas a los contratos adyacentes. Por lo tanto, la apertura/cierre de los contratos anteriores al vencimiento coincide.

El problema clave con el método de Panamá incluye la introducción de un sesgo de tendencia, que introducirá una gran desviación en los precios. Esto puede conducir a datos negativos para contratos suficientemente históricos. Además hay una pérdida de la*pariente*diferencias de precios debido a un cambio absoluto en los valores. Esto significa que las devoluciones son complicadas de calcular (o simplemente incorrectas).

losAjuste de proporcionalidadEl enfoque es similar a la metodología de ajuste de manejo de divisiones de acciones en acciones. En lugar de tomar un cambio absoluto en los contratos sucesivos, la relación entre el precio de liquidación (cierre) más antiguo y el precio de apertura más nuevo se utiliza para ajustar proporcionalmente los precios de los contratos históricos. Esto permite un flujo continuo sin interrupción del cálculo de los rendimientos porcentuales.

El problema principal con el ajuste proporcional es que cualquier estrategia comercial que dependa de un nivel de precio absoluto también deberá ajustarse de manera similar para ejecutar la señal correcta. Este es un proceso problemático y propenso a errores. Por lo tanto, este tipo de flujo continuo a menudo solo es útil para el análisis estadístico resumido, a diferencia de la investigación directa de backtesting.

losRollover/Serie perpetuaEl método crea un contrato continuo de contratos sucesivos al tomar una proporción ponderada linealmente de cada contrato durante una cantidad de días para garantizar una transición más fluida entre cada uno.

Por ejemplo, considere cinco días de suavizado. El precio el día 1,$PAGS_1$, es igual al 80% del precio del contrato lejano (F_1) y 20% del precio cercano al contrato ($norte_1$). Del mismo modo, el día 2 el precio es $PAGS_2=0.6\times F_2+0.4\times norte_2$. Para el día 5 tenemos$PAGS_5=0.0\times F_5+1.0\times norte_5=norte_5$y el contrato entonces se convierte simplemente en una continuación del precio cercano. Por lo tanto, después de cinco días, el contrato pasa sin problemas de lo lejano a lo cercano.

El problema con el método de rollover es que requiere negociar los cinco días, lo que puede aumentar los costos de transacción. Hay otros enfoques menos comunes para el problema, pero vamos a

evitarlos aquí.

El resto de la sección se concentrará en implementar el método de series perpetuas, ya que es el más apropiado para realizar pruebas retrospectivas. Es una forma útil de llevar a cabo*investigación de canal de estrategia*.

Vamos a unir el contrato de futuros "cerca" y "lejos" del petróleo crudo WTI (símbolo CL) para generar una serie de precios continua. Al momento de escribir (enero de 2014), el contrato cercano es CLF2014 (enero) y el contrato lejano es CLG2014 (febrero).

Para realizar la descarga de datos de futuros he hecho uso del complemento Quandl. Asegúrese de configurar el entorno virtual Python correcto en su sistema e instale el paquete Quandl escribiendo lo siguiente en la terminal:

```
pip instalar Quandl
```

Ahora que el paquete Quandl está instalado, necesitamos hacer uso de NumPy y pandas para llevar a cabo la construcción del rollover. Cree un nuevo archivo e ingrese las siguientes declaraciones de importación:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

#cont_futures.py

de__futuro__importarimprimir_funcion

importarfecha y hora

importarnumpy como np
importarpandas como pd
importarQuandl
```

El trabajo principal se lleva a cabo en elfuturos_rollover_pesosfunción. Requiere una fecha de inicio (la primera fecha del contrato cercano), un diccionario de fechas de liquidación del contrato (fechas_de_caducidad), los símbolos de los contratos y el número de días para renovar el contrato (el valor predeterminado es cinco). Los comentarios a continuación explican el código:

```
definitivamentefutures_rollover_weights(fecha_de_inicio, fechas_de_caducidad,
    contratos, rollover_days=5):
    """Esto construye un Pandas DataFrame que contiene ponderaciones (entre 0,0 y
    1,0) de posiciones de contrato que se deben mantener para llevar a cabo un
    rollover de rollover_days antes del vencimiento del primer contrato. La matriz se
    puede 'multiplicar' por otra DataFrame que contiene los precios de liquidación de
    cada contrato para producir un contrato de futuros de serie de tiempo
    continuo."""


    # Construir una secuencia de fechas comenzando
    # desde la primera fecha de inicio del contrato hasta el final
    # fecha del contrato final
    fechas = pd.rango_de_fechas(fecha_de_inicio, fechas_de_caducidad[-1], frecuencia='B')

    # Cree el DataFrame de 'pesos de rollo' que almacenará los multiplicadores para
    # cada contrato (entre 0.0 y 1.0)
    roll_weights = pd.DataFrame(np.zeros((len(fechas), len(contratos))),
                                índice=fechas, columnas=contratos)
    fecha_anterior = roll_weights.index[0]

    # Recorra cada contrato y cree las ponderaciones específicas para
    # cada contrato dependiendo de la fecha de liquidación y rollover_days poryo, (artículo,
    ex_date)enenumerar (expiry_dates.iteritems()):
        sii < len(fechas_de_caducidad) - 1:
```

```
                roll_weights.ix[prev_date:ex_date - pd.offsets.BDay(), item] = 1 roll_rng =
                pd.date_range(end=ex_date - pd.offsets.BDay(),
                                                periodos=rollover_days + 1, freq='B')

                # Cree una secuencia de pesos de rollo (es decir, [0.0,0.2,...,0.8,1.0]
                # y utilícelos para ajustar las ponderaciones de cada futuro
                decay_weights = np.linspace(0, 1, rollover_days + 1)
                roll_weights.ix[roll_rng, item] = 1 - decay_weights roll_weights.ix[roll_rng,

                    fechas_de_caducidad.index[i+1]] = pesos_decaídos
            más:
                roll_weights.ix[prev_date:, artículo] = 1 prev_date
            = ex_date
        devolverrollo_pesos
```

Ahora que se ha producido la matriz de ponderación, es posible aplicarla a la serie temporal individual. La función principal descarga los contratos cercanos y lejanos, crea un DataFrame único para ambos, construye la matriz de ponderación de rollover y finalmente produce una serie continua de ambos precios, ponderados adecuadamente:

```
si__nombre__ == "__principal__":
    # Descargue los contratos de futuros actuales Front and Back (cerca y lejos)
    # para WTI Crude, negociado en NYMEX, de Quandl.com. Necesitaras
    # ajustar los contratos para reflejar sus contratos cercanos/lejos actuales
    # ¡dependiendo del punto en el que leas esto! wti_near =
    Quandl.get("OFDP/FUTURE_CLF2014") wti_far = Quandl.get("OFDP/
    FUTURE_CLG2014") wti = pd.DataFrame({'CLF2014':
    wti_near['Liquidar'],
                                    'CLG2014': wti_far['Settle']}, index=wti_far.index)

    # Crear el diccionario de fechas de vencimiento para cada contrato
    fechas_de_caducidad = pd.Series(
        {'CLF2014': fechahora.fechahora(2013, 12, 19), 'CLG2014':
          fechahora.fechahora(2014, 2, 21)}).pedido()

    # Obtener la matriz de ponderación de rollover/DataFrame
    ponderaciones = futures_rollover_weights(wti_near.index[0],
                                                    fechas_de_caducidad, wti.columns)

    # Construir el futuro continuo de los contratos WTI CL wti_cts = (wti *
    pesos).sum(1).dropna()

    # Salida de la serie fusionada de precios de liquidación de
    contratos impresión(wti_cts.tail(60))
```

La salida es la siguiente:

```
2013-10-14          102.230
2013-10-15          101.240
2013-10-16          102.330
2013-10-17          100.620
2013-10-18          100.990
2013-10-21           99.760
2013-10-22           98.470
2013-10-23           97.000
2013-10-24           97.240
2013-10-25           97.950
. .
```

```
. .
2013-12-24    99.220
2013-12-26    99.550
2013-12-27   100.320
2013-12-30    99.290
2013-12-31    98.420
2014-01-02    95.440
2014-01-03    93.960
2014-01-06    93.430
2014-01-07    93.670
2014-01-08    92.330
Longitud: 60, tipo d: float64
```

Se puede ver que la serie ahora es continua a través de los dos contratos. Esto se puede extender para manejar múltiples entregas a lo largo de una variedad de años, dependiendo de sus necesidades de backtesting.

Parte IV

Modelado

Capítulo 9

Aprendizaje Estadístico

El objetivo de la sección Modelado de este libro es proporcionar un marco cuantitativo sólido para identificar relaciones en los datos del mercado financiero que se pueden explotar para generar estrategias comerciales rentables. El enfoque que se utilizará es el de *Aprendizaje Estadístico*. Este capítulo describe la filosofía del aprendizaje estadístico y las técnicas asociadas que se pueden utilizar para crear modelos cuantitativos para el comercio financiero.

9.1 ¿Qué es el aprendizaje estadístico?

Antes de discutir los aspectos teóricos del aprendizaje estadístico, es apropiado considerar un ejemplo de una situación de finanzas cuantitativas donde tales técnicas son aplicables. Considere un fondo cuantitativo que desee hacer predicciones a largo plazo del índice bursátil S&P500. El fondo ha logrado recaudar una cantidad sustancial de *datos fundamentales* asociados a las empresas que constituyen el índice. Los datos fundamentales incluyen *Tasa de ganancias sobre precio* o *valor en libros*, por ejemplo. ¿Cómo debería utilizar el fondo estos datos para hacer predicciones del índice con el fin de crear una herramienta de negociación? El aprendizaje estadístico proporciona uno de esos enfoques para este problema.

En un sentido más cuantitativo, estamos intentando modelar el comportamiento de un *Salir* o *respuesta* basado en un conjunto de *predictores* o *caracteristicas* suponiendo una relación entre los dos. En el ejemplo anterior, el valor del índice bursátil es la respuesta y los datos fundamentales asociados con las empresas constituyentes son los predictores.

Esto se puede formalizar considerando una respuesta Y con $pags$ diferentes caracteristicas $X_1, X_2, ..., X_{pags}$. si utilizamos *notación vectorial* entonces podemos definir $X = (X_1, X_2, ..., X_{pags})$, que es un vector de longitud $pags$. Entonces el modelo de nuestra relación viene dado por:

$$Y = F(X) + \varepsilon \tag{9.1}$$

Dónde F es una función desconocida de los predictores y ε representa un *error* o *término de ruido*. En tono rimbombante, ε no depende de los predictores y tiene una media de cero. Este término se incluye para representar información que no se considera dentro F. Por lo tanto, podemos volver al ejemplo del índice bursátil para decir que Y representa el valor del S&P500 mientras que el X_i Los componentes representan los valores de los factores fundamentales individuales.

El objetivo del aprendizaje estadístico es *estimar* la forma de F en base a los datos observados y para evaluar qué tan precisas son esas estimaciones.

9.1.1 Predicción e Inferencia

Hay dos tareas generales que son de interés en el aprendizaje estadístico: *predicción* y *inferencia*.

La predicción se ocupa de predecir una respuesta. Y basado en *recién observado* vaticinador, X. Si se ha determinado la relación del modelo, entonces es sencillo predecir la respuesta utilizando una estimación de F para producir una estimación de la respuesta:

$$\hat{Y} = \hat{F}(X) \tag{9.2}$$

La forma funcional de $\hat{F}$ a menudo no es importante en un escenario de predicción, suponiendo que las respuestas estimadas están cerca de las respuestas verdaderas y, por lo tanto, es precisa en sus predicciones. Diferentes estimaciones de $\hat{F}$ producirá varias precisiones de las estimaciones de Y. El error asociado con tener una mala estimación $\hat{F}$ de F se llama el *error reducible*. Tenga en cuenta que siempre hay un grado de *error irreducible* porque nuestra especificación original del problema incluía la ε término de error. Este término de error encapsula los factores no medidos que pueden afectar la respuesta, Y. El enfoque adoptado es tratar de minimizar el error reducible con el entendimiento de que siempre habrá un límite superior de precisión basado en el error irreducible.

La inferencia se ocupa de la situación en la que es necesario comprender la relación entre X y Y y por lo tanto su forma exacta debe ser determinada. Uno puede desear identificar predictores importantes o determinar la relación entre los predictores individuales y la respuesta. También se podría preguntar si la relación es *lineal* o *no lineal*. Lo primero significa que es probable que el modelo sea más interpretable, pero a expensas de una previsibilidad potencialmente peor. Este último proporciona modelos que generalmente son más predictivos pero a veces son menos interpretables. Por lo tanto, un intercambio entre *previsibilidad* y *interpretabilidad* a menudo existe.

En este libro estamos menos interesados en los modelos de inferencia ya que la forma real de F no es tan importante como su capacidad para hacer predicciones precisas. Por lo tanto, una gran parte de la sección Modelado del libro se basará en el modelado predictivo. La siguiente sección trata de cómo hacemos para construir una estimación $\hat{F}$ por F.

9.1.2 Modelos paramétricos y no paramétricos

En una situación de aprendizaje estadístico, a menudo es posible construir un conjunto de tuplas de predictores y respuestas de la forma $\{(X_1, Y_1),(X_2, Y_2), ...,(X_{norte}, Y_{norte})\}$, dónde X_j se refiere al j-ésimo vector predictor y no al j-ésimo componente de un vector predictor particular (que se denota por X_j). Un conjunto de datos de esta forma se conoce como *datos de entrenamiento* ya que estará acostumbrado *tren* un método particular de aprendizaje estadístico sobre cómo generar F. Para estimar realmente F tenemos que encontrar un $\hat{F}$ que proporciona una aproximación razonable a un determinado Y bajo un predictor particular X. Hay dos amplias categorías de modelos estadísticos que nos permiten lograr esto. son conocidos como *paramétrico* y *no paramétrico* modelos

Modelos paramétricos

La característica definitoria de los métodos paramétricos es que requieren la *especificación* o *suposición* de la forma de F. Esta es una decisión de modelado. La primera opción es si considerar un modelo lineal o no lineal. Consideremos el caso más simple de un modelo lineal. Tal modelo reduce el problema de la estimación de alguna función desconocida de dimensión *pags* al de estimar un vector de coeficientes $\beta = (\beta_0, \beta_1, ..., \beta_{pags})$ de longitud *pags*+1.

Por qué *pags*+1 y no *pags*? Dado que los modelos lineales pueden ser *afín*, es decir, no pueden pasar por el origen al crear una "línea de mejor ajuste", se requiere un coeficiente para especificar la "intersección". En una configuración de modelo lineal unidimensional (regresión), esto a menudo se representa como α. Para nuestro modelo lineal multidimensional, donde hay *pags* predictores, necesitamos un valor adicional β_0 para representar nuestro intercepto y por lo tanto hay *pags*+1 componentes en nuestro β estimado de β.

Ahora que hemos especificado una forma funcional (lineal) de F Necesitamos que *tren* eso. "Entrenamiento" en este caso significa encontrar una estimación para β tal que:

$$Y \approx \hat{\beta}^T X = \beta_0 + \beta_1 X_1 + ... + \beta_{pags} X_{pags} \tag{9.3}$$

En la configuración lineal podemos usar un algoritmo como *mínimos cuadrados ordinarios* (OLS), pero también hay otros métodos disponibles. Es mucho más sencillo estimar β que encajar en un (potencialmente no lineal) F.

Sin embargo, al elegir un enfoque lineal paramétrico, nuestra estimación $\hat{F}$ es poco probable que esté replicando la verdadera forma de F. Esto puede conducir a estimaciones pobres porque el modelo no es *Floridaflexible* suficiente.

Un remedio potencial es considerar agregar más parámetros, eligiendo formas alternativas para F. Desafortunadamente, si el modelo se vuelve demasiado flexible, puede conducir a una situación muy peligrosa conocida como *sobreajuste*, de la que hablaremos extensamente en capítulos posteriores. En esencia, el modelo sigue demasiado de cerca el ruido y no la señal.

Modelos no paramétricos

El enfoque alternativo es considerar una forma no paramétrica de F. El beneficio es que potencialmente puede adaptarse a una gama más amplia de posibles formas de F y por lo tanto es más flexible. Desafortunadamente, los modelos no paramétricos sufren la necesidad de tener una gran cantidad de puntos de datos de observación, a menudo mucho más que en una configuración paramétrica. Además, los métodos no paramétricos también son propensos al sobreajuste si no se tratan con cuidado, como se describe anteriormente.

Los modelos no paramétricos pueden parecer una opción natural para los modelos comerciales cuantitativos, ya que aparentemente hay una gran cantidad de datos (históricos) sobre los cuales aplicar los modelos. Sin embargo, los métodos no siempre son óptimos. Si bien la mayor flexibilidad es atractiva para modelar las no linealidades en los datos del mercado de valores, es muy fácil sobreajustar los datos debido a la mala relación señal/ruido que se encuentra en las series temporales financieras.

Por lo tanto, se prefiere un "término medio" de considerar modelos con cierto grado de flexibilidad. Discutiremos tales problemas en el capítulo sobre Optimización más adelante en el libro.

9.1.3 Aprendizaje supervisado y no supervisado

A menudo se hace una distinción en el aprendizaje automático estadístico entre *supervisado* y *sin supervisión* métodos. En este libro nos interesaremos casi exclusivamente en las técnicas supervisadas, pero las técnicas no supervisadas ciertamente son aplicables a los mercados financieros.

Un modelo supervisado requiere que para cada vector predictor X_i hay una respuesta asociada Y_i. La "supervisión" del procedimiento se produce cuando el modelo de F es *entrenado* o *fit* a este dato en particular. Por ejemplo, cuando se ajusta un modelo de regresión lineal, se usa el algoritmo OLS para entrenarlo y, en última instancia, se produce una estimación. β al vector de coeficientes de regresión, β.

En un modelo no supervisado no hay una respuesta correspondiente Y_i para cualquier predictor en particular X_i. Por lo tanto, no hay nada que "supervise" el entrenamiento del modelo. Este es claramente un entorno mucho más desafiante para que un algoritmo produzca resultados, ya que no existe una forma de "función de aptitud" con la que evaluar la precisión. A pesar de este contratiempo, las técnicas no supervisadas son extremadamente poderosas. Son especialmente útiles en el ámbito de *agrupamiento*.

Un modelo de conglomerados parametrizado, cuando se le proporciona un parámetro que especifica el número de conglomerados a identificar, a menudo puede discernir relaciones imprevistas dentro de los datos que de otro modo no se habrían determinado fácilmente. Tales modelos generalmente caen dentro del dominio de *Análisis de negocio* y *optimización de marketing de consumo* pero tienen usos dentro de las finanzas, particularmente en lo que respecta a la evaluación de la agrupación dentro de la volatilidad, por ejemplo.

Este libro se concentrará predominantemente en los métodos de aprendizaje supervisado, ya que existe una gran cantidad de datos históricos sobre los cuales entrenar tales modelos.

9.2 Técnicas

El aprendizaje automático estadístico es un vasto campo interdisciplinario, con muchas áreas de investigación dispares. El resto de este capítulo considerará las técnicas más relevantes para las finanzas cuantitativas y el comercio algorítmico en particular.

9.2.1 Regresión

La regresión se refiere a un amplio grupo de técnicas de aprendizaje automático supervisado que brindan capacidades predictivas e inferenciales. Una parte importante de las finanzas cuantitativas utiliza técnicas de regresión y, por lo tanto, es esencial estar familiarizado con el proceso. La regresión trata de modelar la relación entre una variable dependiente (respuesta) y un conjunto de variables independientes (predictores). En particular, el objetivo de la regresión es determinar el cambio en una respuesta, cuando

una de las variables independientes cambia, bajo el supuesto de que las variables independientes restantes se mantienen fijas.

La técnica de regresión más conocida es*Regresión lineal*, que supone una relación lineal entre los predictores y la respuesta. Dicho modelo conduce a estimaciones de parámetros (generalmente denotadas por el vectorβ) para la respuesta lineal a cada predictor. Estos parámetros se estiman a través de un procedimiento conocido como*mínimos cuadrados ordinarios*(MCO). La regresión lineal se puede utilizar tanto para la predicción como para la inferencia.

En el primer caso, se puede agregar un nuevo valor del predictor (sin una respuesta correspondiente) para *predecir*un nuevo valor de respuesta. Por ejemplo, considere un modelo de regresión lineal utilizado para predecir el valor del S&P500 al día siguiente, a partir de los datos de precios de los últimos cinco días. El modelo se puede ajustar usando OLS a través de datos históricos. Luego, cuando llegan nuevos datos de mercado para el S&P500, se pueden ingresar en el modelo (como predictor) para generar una respuesta pronosticada para el precio diario de mañana. Esto puede formar la base de una estrategia comercial simplista.

En el último caso (inferencia) se puede evaluar la fuerza de la relación entre la respuesta y cada predictor para determinar el subconjunto de predictores que tienen un efecto sobre la respuesta. Esto es más útil cuando el objetivo es entender*por qué*la respuesta varía, como en un estudio de marketing o ensayo clínico. La inferencia suele ser menos útil para quienes realizan transacciones algorítmicas, ya que la calidad de la predicción es fundamentalmente más importante que la relación subyacente. Dicho esto, no se debe confiar únicamente en el enfoque de "caja negra" debido a la prevalencia del sobreajuste al ruido en los datos.

Otras técnicas incluyen*Regresión logística*, que está diseñado para predecir un*categórico*respuesta (como "ARRIBA", "ABAJO", "FLAT") en lugar de una*continuo*respuesta (como un precio de mercado de valores). Esto técnicamente lo convierte en un*herramienta de clasificación*(ver más abajo), pero generalmente se agrupa bajo el estandarte de regresión. Un procedimiento estadístico general conocido como*Estimación de máxima verosimilitud*(MLE) se utiliza para estimar los valores de los parámetros de una regresión logística.

9.2.2 Clasificación

La clasificación abarca técnicas de aprendizaje automático supervisado que tienen como objetivo*clasificar*un *observación*(similar a un predictor) en un conjunto de categorías predefinidas, basadas en características asociadas con la observación. Estas categorías pueden estar desordenadas, por ejemplo, "rojo", "amarillo", "azul" u ordenadas, por ejemplo, "bajo", "medio", "alto". En el último caso, tales grupos categóricos se conocen como *ordinales*. Algoritmos de clasificación -*clasificadores*-son ampliamente utilizados en finanzas cuantitativas, especialmente en el ámbito de la predicción de la dirección del mercado. En este libro estudiaremos los clasificadores extensamente.

Los clasificadores se pueden utilizar en el comercio algorítmico para predecir si una serie de tiempo en particular tendrá rendimientos positivos o negativos en períodos de tiempo posteriores (desconocidos). Esto es similar a una configuración de regresión, excepto que no se predice el valor real de la serie temporal, sino su dirección. Una vez más, podemos utilizar predictores continuos, como los precios de mercado anteriores, como observaciones. Consideraremos clasificadores lineales y no lineales, incluida la regresión logística, el análisis discriminante lineal/ cuadrático, las máquinas de vectores de soporte (SVM) y las redes neuronales artificiales (ANN).*Tenga en cuenta que algunos de los métodos anteriores también se pueden usar en una configuración de regresión.*

9.2.3 Modelos de series temporales

Un componente clave en el comercio algorítmico es el tratamiento y la predicción de*fiserie de tiempo financiero*. Nuestro objetivo generalmente es predecir valores futuros de series temporales en función de valores anteriores o factores externos. Por lo tanto, el modelado de series de tiempo puede verse como un subconjunto mixto de regresión y clasificación. Los modelos de series de tiempo difieren de los modelos no temporales porque los modelos hacen un uso deliberado del *ordenamiento temporal*de la serie Por lo tanto, los predictores a menudo se basan en valores pasados o actuales, mientras que las respuestas suelen ser valores futuros que deben predecirse.

Hay una gran cantidad de literatura sobre diferentes modelos de series de tiempo. Hay dos familias amplias de modelos de series temporales que nos interesan en el comercio algorítmico. El primer conjunto son los lineales.*media móvil integrada autorregresiva*(ARIMA) familia de modelos, que se utilizan para modelar las variaciones en el valor absoluto de una serie de tiempo. La otra familia de series temporales son las*autorregresivo*

heteroscedasticidad condicional(ARCH), que se utilizan para modelar la varianza (es decir, la volatilidad) de series temporales a lo largo del tiempo. Los modelos ARCH utilizan valores anteriores (volatilidades) de la serie temporal para predecir valores futuros (volatilidades). Esto es en contraste con*volatilidad estocástica*modelos, que utilizan más de una serie temporal estocástica (es decir, múltiples ecuaciones diferenciales estocásticas) para modelar la volatilidad.

Todas las series temporales de precios históricos brutos son*discreto*en que contienen valores finitos. En el campo de las finanzas cuantitativas es común estudiar*continuo*modelos de series temporales. En particular, el famoso *Movimiento Browniano Geométrico*, la*Volatilidad estocástica de Heston*modelo y el *Ornstein-Uhlenbeck*Todos los modelos representan series temporales continuas con diferentes formas de comportamiento estocástico. Utilizaremos estos modelos de series de tiempo en capítulos posteriores para intentar caracterizar el comportamiento de las series de tiempo financieras con el fin de explotar sus propiedades para crear estrategias comerciales viables.

Capítulo 10

Análisis de series temporales

En este capítulo vamos a considerar las pruebas estadísticas que nos ayudarán a identificar series de precios que tengan un comportamiento de tendencia o de reversión a la media. Si podemos identificar tales series estadísticamente, entonces podemos capitalizar este comportamiento formando estrategias comerciales de impulso o de reversión a la media.

En capítulos posteriores, utilizaremos estas pruebas estadísticas para ayudarnos a identificar series temporales candidatas y luego crear estrategias algorítmicas en torno a ellas.

10.1 Prueba de reversión a la media

Uno de los conceptos comerciales cuantitativos clave esreversión media.Este proceso se refiere a una serie de tiempo que muestra una tendencia a volver a un valor medio histórico. Dicha serie de tiempo puede explotarse para generar estrategias comerciales a medida que ingresamos al mercado cuando una serie de precios está lejos de la media bajo la expectativa de que la serie volverá a un valor medio, por lo que salimos del mercado para obtener una ganancia. Las estrategias de reversión a la media forman un gran componente del*arbitraje estadístico* fondos de cobertura cuantitativos. En capítulos posteriores, crearemos estrategias intradiarias e interdiarias que explotan el comportamiento de reversión a la media.

La idea básica cuando se trata de determinar si una serie de tiempo es de reversión a la media es usar una prueba estadística para ver si difiere del comportamiento de una serie de tiempo.*Caminata aleatoria*. Una caminata aleatoria es una serie de tiempo en la que el siguiente movimiento direccional es completamente independiente de cualquier movimiento anterior; en esencia, la serie de tiempo no tiene "memoria" de dónde ha estado. Sin embargo, una serie de tiempo con reversión a la media es diferente. El cambio en el valor de la serie temporal en el siguiente período de tiempo es proporcional al valor actual. En concreto, es proporcional a la diferencia entre el precio medio histórico y el precio actual.

Matemáticamente, una serie de tiempo (continua) de este tipo se conoce comoOrnstein-Uhlenbeck proceso. Si podemos mostrar, estadísticamente, que una serie de precios se comporta como una serie de Ornstein-Uhlenbeck, entonces podemos comenzar el proceso de formar una estrategia comercial a su alrededor. Por lo tanto, el objetivo de este capítulo es delinear las pruebas estadísticas necesarias para identificar la reversión a la media y luego usar las bibliotecas de Python (en particular*modelos estadisticos*) con el fin de implementar estas pruebas. En particular, estudiaremos el concepto deestacionariedady cómo probarlo.

Como se indicó anteriormente, un*continuo*Las series de tiempo que revierten a la media se pueden representar mediante una ecuación diferencial estocástica de Ornstein-Uhlenbeck:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \tag{10.1}$$

Dóndeθes la tasa de reversión a la media,μes el valor medio del proceso,σes la varianza del proceso y$W_t$es un proceso de Wiener o movimiento browniano.

Esta ecuación establece esencialmente que el cambio de la serie de precios en el siguiente período de tiempo continuo es proporcional a la diferencia entre el precio medio y el precio actual, con la adición del ruido gaussiano.

Podemos usar esta ecuación para motivar la definición de la Prueba Dickey-Fuller Aumentada, que ahora describiremos.

10.1.1 Prueba Dickey-Fuller aumentada (ADF)

La prueba ADF utiliza el hecho de que si una serie de precios posee una reversión a la media, el siguiente nivel de precios será proporcional al nivel de precios actual. Matemáticamente, el ADF se basa en la idea de probar la presencia de unraíz unitariaen unautorregresivomuestra de serie de tiempo.

Podemos considerar un modelo para una serie de tiempo, conocido como*modelo de orden de rezago linealpags*. Este modelo dice que el cambio en el valor de la serie temporal es proporcional a una constante, el tiempo mismo y el anterior.*pags*valores de la serie temporal, junto con un término de error:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{pag1} \Delta y_{t-p+1} + \varepsilon_t \tag{10.2}$$

Dóndeαes una constante,βrepresenta el coeficiente de una tendencia temporal y$\Delta y_t = y(t) - y(t-1)$.El papel de la prueba de hipótesis ADF es determinar, estadísticamente, si$\gamma = 0$,lo que indicaría (con$\alpha = \beta = 0$)que el proceso es un paseo aleatorio y, por lo tanto, no revierte a la media. Por lo tanto, estamos probando para el*hipótesis nula*que$\gamma = 0$.

Si la hipótesis de que$\gamma = 0$puede rechazarse, entonces el siguiente movimiento de la serie de precios es proporcional al precio actual y, por lo tanto, es poco probable que sea un paseo aleatorio. Esto es lo que entendemos por una "prueba estadística".

Entonces, ¿cómo se lleva a cabo la prueba ADF?

- Calcula el*Estadística de prueba*,DF_τ,que se utiliza en la decisión de rechazar la hipótesis nula
- Utilizar el*distribución*del estadístico de prueba (calculado por Dickey y Fuller), junto con los valores críticos, para decidir si se rechaza la hipótesis nula

Empecemos por calcular el estadístico de prueba (DF_τ). Esto viene dado por el*muestra*proporcionalmente constante$\hat{\gamma}$dividido por el*Error estándar*de la constante de proporcionalidad muestral:

$$DF_\tau = \frac{\hat{\gamma}}{SE(\hat{\gamma})} \tag{10.3}$$

Ahora que tenemos el estadístico de prueba, podemos usar la distribución del estadístico de prueba calculada por Dickey y Fuller para determinar el rechazo de la hipótesis nula para cualquier valor crítico porcentual elegido. El estadístico de prueba es un número negativo y, por lo tanto, para que sea significativo más allá de los valores críticos, el número debe ser menor (es decir, más negativo) que estos valores.

Un problema práctico clave para los comerciantes es que cualquier desviación constante a largo plazo en un precio es de una magnitud mucho menor que cualquier fluctuación a corto plazo y, por lo tanto, a menudo se supone que la desviación es cero ($\beta = 0$)para el modelo de retraso lineal descrito anteriormente.

Dado que estamos considerando un modelo de orden rezagado*pags*, tenemos que establecer realmente*pags*a un valor particular. Por lo general, es suficiente, para la investigación comercial, establecer$pags = 1$que nos permita rechazar la hipótesis nula. Sin embargo, tenga en cuenta que técnicamente esto introduce un parámetro en un modelo comercial basado en el ADF.

Para calcular la prueba Dickey-Fuller aumentada podemos hacer uso de las bibliotecas pandas y statsmodels. El primero nos proporciona un método sencillo para obtener datos de Volumen abierto-alto-bajo-cerrado (OHLCV) de Yahoo Finance, mientras que el segundo envuelve la prueba ADF en una función fácil de llamar. Esto evita que tengamos que calcular la estadística de prueba manualmente, lo que nos ahorra tiempo.

Llevaremos a cabo la prueba ADF en una serie de precios de muestra de acciones de Amazon, desde el 1 de enero de 2000 hasta el 1 de enero de 2015.

Aquí está el código de Python para llevar a cabo la prueba:

```
de__futuro__importarimprimir_funcion

# Importar la biblioteca de series temporales
importarstatsmodels.tsa.stattools como ts

# Importar fecha y hora y Pandas DataReader
```

```
defecha y horaimportarfecha y hora
importarpandas.io.data como web

# Descargue los datos de Amazon OHLCV del 1/1/2000 al 1/1/2015
amzn = web.DataReader("AMZN", "yahoo", fechahora(2000,1,1), fechahora(2015,1,1)                    )

# Muestra los resultados de la prueba Dickey-Fuller aumentada para Amazon
# con un valor de orden de retraso de 1
ts.adfuller(amzn['Adj Cerrar'], 1)
```

Aquí está el resultado de la prueba Dickey-Fuller aumentada para Amazon durante el período. El primer valor es la estadística de prueba calculada, mientras que el segundo valor es el*valor p*. El cuarto es el número de puntos de datos en la muestra. El quinto valor, el diccionario, contiene los valores críticos de la estadística de prueba en los valores de 1, 5 y 10 por ciento respectivamente.

```
(0.049177575166452235,
 0.96241494632563063,
 1,
 3771,
 {'1%': -3.4320852842548395, '10%':
   -2.5671781529820348, '5%':
   -2.8623067530084247},
  19576.116041473877)
```

Dado que el valor calculado del estadístico de prueba es mayor que cualquiera de los valores críticos en los niveles de 1, 5 o 10 por ciento, no podemos rechazar la hipótesis nula deγ=0y, por lo tanto, es poco probable que hayamos encontrado una serie temporal de reversión a la media. Esto está en consonancia con nuestra enseñanza, ya que la mayoría de las acciones se comportan de forma similar al Movimiento Browniano Geométrico (GBM), es decir, un paseo aleatorio.

Esto concluye cómo utilizamos la prueba ADF. Sin embargo, existen métodos alternativos para detectar la reversión a la media, particularmente a través del concepto deestacionariedad,que ahora discutiremos.

10.2 Pruebas de estacionariedad

Una serie de tiempo (o proceso estocástico) se define comofuertemente estacionariosi es*distribución de probabilidad conjunta*es invariante bajo traslaciones en tiempo o espacio. En particular, y de importancia clave para los comerciantes, la media y la varianza del proceso no cambian con el tiempo o el espacio y cada uno de ellos no sigue una tendencia.

Una característica crítica de las series de precios estacionarias es que los precios dentro de la serie se difunden desde su valor inicial a un ritmo más lento que el de un GBM. Al medir la tasa de este comportamiento difusivo, podemos identificar la naturaleza de la serie temporal y, por lo tanto, detectar si se está revirtiendo a la media.

Ahora esbozaremos un cálculo, a saber, el Exponente de Hurst, que nos ayuda a caracterizar la estacionariedad de una serie de tiempo.

10.2.1 Exponente de Hurst

El objetivo del Exponente de Hurst es proporcionarnos un valor escalar que nos ayudará a identificar (dentro de los límites de la estimación estadística) si una serie está revirtiendo a la media, caminando aleatoriamente o con tendencia.

La idea detrás del cálculo del exponente de Hurst es que podemos usar la varianza de una serie de precios logarítmicos para evaluar la tasa de comportamiento difusivo. Por un lapso de tiempo arbitrarioτ, la varianza deτes dado por:

$$\text{Var}(\tau) = \langle |\text{Iniciar sesión}(t+\tau) - \text{Iniciar sesión}(t)|^2 \rangle \tag{10.4}$$

donde los paréntesis$\langle y \rangle$referirse al promedio de todos los valores de$t$.

La idea es comparar la tasa de difusión con la de un GBM. En el caso de un GBM, en tiempos largos (es decir, cuandoτes grande) la varianza deτes proporcional aτ:

$$\langle |\text{Iniciar sesión}(t+\tau) - \text{Iniciar sesión}(t)|^2 \rangle \sim \tau \quad (10.5)$$

Si encontramos un comportamiento que difiere de esta relación, entonces hemos identificado una serie de tendencia o de reversión a la media. La idea clave es que si cualquier movimiento secuencial de precios posee una correlación distinta de cero (conocida como autocorrelación), entonces la relación anterior no es válida. En su lugar, se puede modificar para incluir un valor de exponente "2*H*", que nos da el valor del exponente de Hurst *H*:

$$\langle |\text{Iniciar sesión}(t+\tau) - \text{Iniciar sesión}(t)|^2 \rangle \sim \tau^{2H} \quad (10.6)$$

Así se puede ver que si*H*=0.5tenemos un GBM, ya que simplemente se convierte en la relación anterior. Sin embargo, si*H6*= 0.5entonces tenemos un comportamiento de tendencia o de reversión a la media. En particular:

- *H* <0.5 -La serie de tiempo es revirtiendo a la media
- *H*=0.5 -La serie de tiempo es un Movimiento Browniano Geométrico
- *H* >0.5 -La serie temporal es tendencia

Además de la caracterización de la serie temporal, el Exponente de Hurst también describe el grado en que una serie se comporta de la manera categorizada. Por ejemplo, un valor de*H*cerca de 0 es una serie de reversión altamente media, mientras que para*H*cerca de 1, la serie tiene una fuerte tendencia.

Para calcular el exponente de Hurst para la serie de precios de Amazon, como se utilizó anteriormente en la explicación del ADF, podemos usar el siguiente código de Python:

```
de__futuro__importarimprimir_funcion

deentumecidoimportarcumsum, iniciar sesión, polyfit, sqrt, std, restar de
numpy.aleatorioimportarrancio

definitivamenteduele(ts):
    """Devuelve el exponente de Hurst del vector de serie temporal ts"""
    # Crear el rango de valores de retraso
    retrasos = rango (2, 100)

    # Calcular la matriz de las varianzas de las diferencias rezagadas tau = [raíz
    cuadrada(estándar(restar(ts[retraso:], ts[:-retraso])))porretrasoenretrasos]

    # Use un ajuste lineal para estimar el exponente de Hurst poly
    = polyfit(registro(retrasos), registro(tau), 1)

    # Devuelve el exponente de Hurst de la salida de polyfit devolver
    poli[0]*2.0

# Cree una serie de movimiento browniano gométrico, reversión a la media y tendencia
gbm = log(suma acumulada(randn(100000))+1000) mr = log(randn(100000)+1000)

tr = log(cumsum(randn(100000)+1)+1000)

# Salida del exponente de Hurst para cada una de las series anteriores
# y el precio de Amazon (el precio de cierre ajustado) para
# la prueba ADF dada arriba en el artículo impresión
("Hurst (GBM):                %s" % daño(gbm))
impresión("Hurst (MR):        %s" % herido(señor))
impresión("Hurst(TR):         %s" % daño(tr))
```

```
# ¡Suponiendo que haya ejecutado el código anterior para obtener
'amzn'! impresión("Hurst(AMZN): %s" % hurst(amzn['Adj Close']))
```

El resultado del código de Hurst Exponent Python se muestra a continuación:

```
Hurst (GBM):    0.502051910931
Hurst (MR):     0.000166110248967
Hurst(TR):      0.957701001252
Hurst (AMZN):   0.454337476553
```

De esta salida podemos ver que el GBM posee un exponente de Hurst, H, eso es casi exactamente 0.5. La serie de reversión a la media tiene H casi igual a cero, mientras que la serie de tendencias tiene H cerca de 1.

Curiosamente, Amazon tiene H también cerca de 0,5, lo que indica que es similar a un GBM, ¡al menos para el período de muestra que estamos utilizando!

10.3 Cointegración

En realidad, es muy difícil encontrar un activo negociable que posea un comportamiento de reversión a la media. En términos generales, las acciones se comportan como GBM y, por lo tanto, hacen que las estrategias comerciales de reversión a la media sean relativamente inútiles. Sin embargo, nada nos impide crear un *portafolio* de la serie de precios que es estacionaria. Por lo tanto, podemos aplicar estrategias comerciales de reversión a la media a la cartera.

La forma más simple de estrategias comerciales de reversión a la media es el clásico "comercio de pares", que generalmente involucra un par de acciones largo-corto neutral en dólares. La teoría dice que es probable que dos empresas del mismo sector estén expuestas a factores de mercado similares, que afectan a sus negocios. Ocasionalmente, los precios relativos de sus acciones divergirán debido a ciertos eventos, pero volverán a la media de largo plazo.

Consideremos dos acciones del sector energético, Approach Resources Inc, dada por el ticker AREX y Whiting Petroleum Corp, dada por el ticker WLL. Ambos están expuestos a condiciones de mercado similares y, por lo tanto, es probable que tengan una relación de pares estacionarios. Ahora vamos a crear algunos gráficos usando pandas y las bibliotecas de Matplotlib para demostrar la naturaleza de cointegración de AREX y WLL. La primera gráfica (Figura 10.1) muestra sus respectivos historiales de precios para el período del 1 de enero de 2012 al 1 de enero de 2013.

Si creamos un diagrama de dispersión de sus precios, vemos que la relación es básicamente lineal (ver Figura 10.2) para este período.

El comercio de pares funciona esencialmente mediante el uso de un modelo lineal para una relación entre los dos precios de las acciones:

$$y(t) = \beta x(t) + \varepsilon(t) \tag{10.7}$$

Dónde $y(t)$ es el precio de las acciones de AREX y $x(t)$ es el precio de las acciones de WLL, tanto en el día t. Si graficamos los residuos $\varepsilon(t) = y(t) - \beta x(t)$ (por un valor particular de β que determinaremos a continuación) creamos una nueva serie temporal que, a primera vista, parece relativamente estacionaria. Esto se da en la Figura 10.3.

Describiremos el código para cada una de estas parcelas a continuación.

10.3.1 Prueba Dickey-Fuller aumentada cointegrada

Para confirmar estadísticamente si esta serie revierte a la media, podríamos usar una de las pruebas que describimos anteriormente, a saber, la prueba de Dickey-Fuller aumentada o el exponente de Hurst. Sin embargo, ninguna de estas pruebas realmente nos ayudará a determinar β, el ratio de cobertura necesario para formar la combinación lineal, sólo nos dirán si, para un determinado β, la combinación lineal es estacionaria.

Aquí es donde entra en juego la prueba Dickey-Fuller aumentada cointegrada (CADF). Determina la relación de cobertura óptima realizando una regresión lineal contra las dos series de tiempo y luego prueba la estacionariedad bajo la combinación lineal.

Figura 10.1: Gráficas de series de tiempo de AREX y WLL

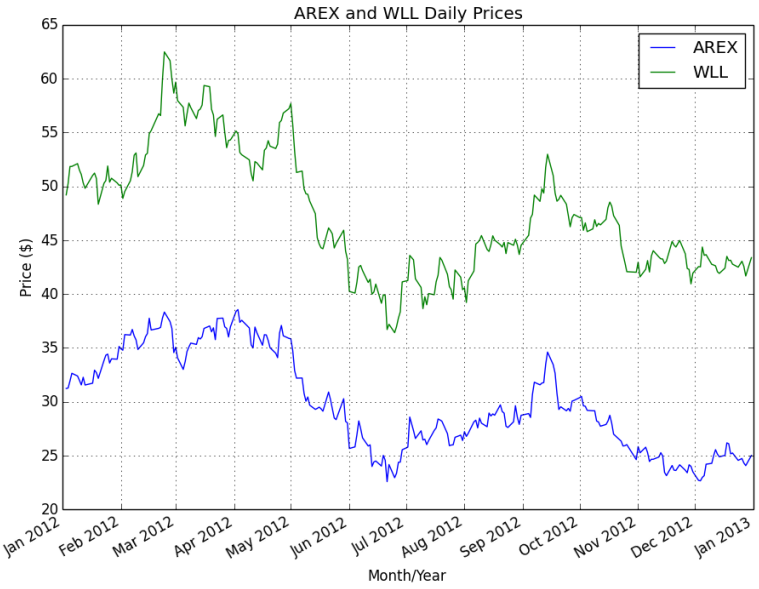
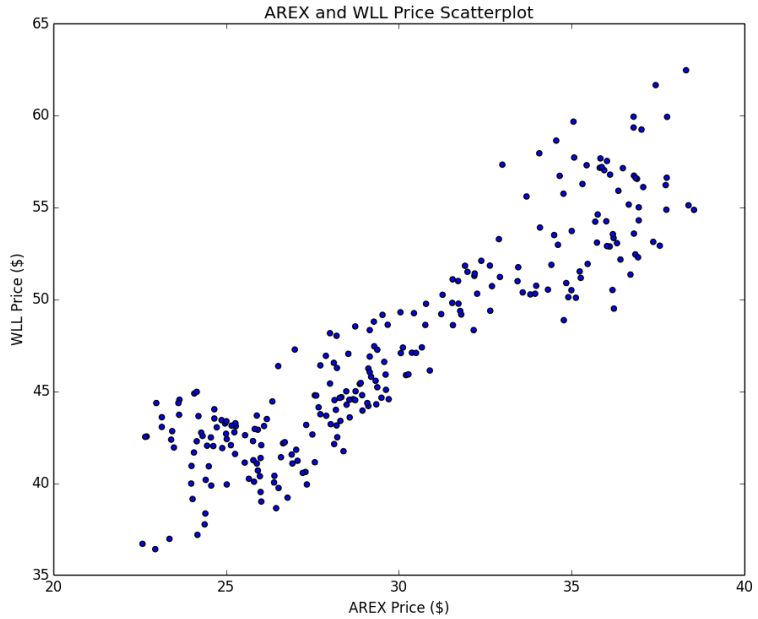


Figura 10.2: Diagrama de dispersión de los precios AREX y WLL

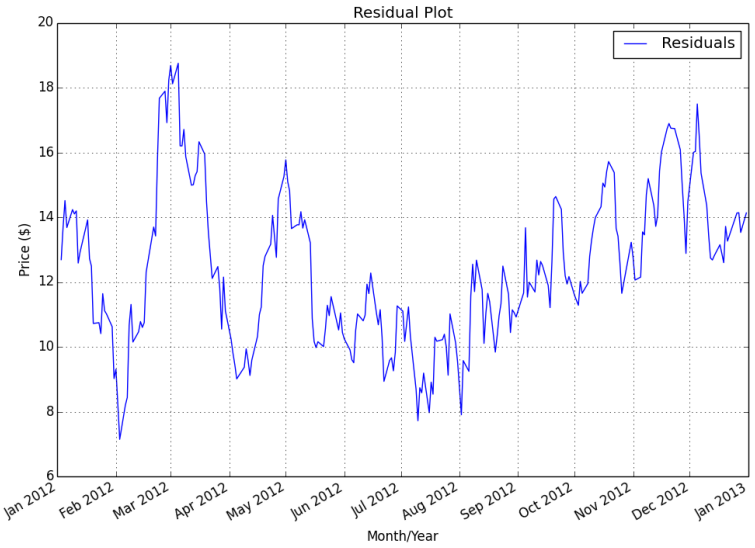


Implementación de Python

Ahora usaremos bibliotecas de Python para probar una relación de cointegración entre AREX y WLL para el período del 1 de enero de 2012 al 1 de enero de 2013. Usaremos Yahoo Finance para la fuente de datos y Statsmodels para realizar la prueba ADF, como se indicó anteriormente.

La primera tarea es crear un nuevo archivo, cadf.py, e importar las bibliotecas necesarias. El código utiliza NumPy, Matplotlib, Pandas y Statsmodels. Para etiquetar correctamente los ejes

Figura 10.3: Gráfico residual de combinación lineal AREX y WLL



y descargamos datos de Yahoo Finance a través de pandas, importamos el módulo matplotlib.dates y el módulo pandas.io.data. También hacemos uso de la función Ordinary Least Squares (OLS) de pandas:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

#cadf.py

importarfecha y hora
importarnumpy como np
importarmatplotlib.pyplot como plt
importarmatplotlib.dates como mdates
importarpandas como pd importar
pandas.io.data como web importarpprint

importarstatsmodels.tsa.stattools como ts

depandas.stats.apiimportarviejos
```

La primera función, plot_price_series, toma un DataFrame de pandas como entrada, con dos columnas dadas por las cadenas de marcadores de posición "ts1" y "ts2". Estos serán nuestros pares de acciones. La función simplemente traza las dos series de precios en el mismo gráfico. Esto nos permite inspeccionar visualmente si es probable alguna cointegración.

Usamos el módulo de fechas de Matplotlib para obtener los meses de los objetos de fecha y hora. Luego creamos una figura y un conjunto de ejes sobre los cuales aplicar el etiquetado/trazado. Finalmente trazamos la figura:

```
#cadf.py

definitivamenteplot_price_series(df, ts1, ts2):
    meses = mdates.MonthLocator()#cada mes fig, ax =
    plt.subplots() ax.plot(df.index, df[ts1], label=ts1)
```

```
    ax.plot(df.index, df[ts2], label=ts2) ax.xaxis.set_major_locator(meses)
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y')) ax.set_xlim(datetime.datetime
    (2012, 1, 1), datetime.datetime(2013, 1, 1)) ax.grid(Verdadero)



    fig.autofmt_xdate()

    plt.xlabel('Mes/Año')
    plt.ylabel('Precio ($)')
    plt.title('%s y %s precios diarios' % (ts1, ts2)) plt.legend()

    plt.mostrar()
```

La segunda función, plot_scatter_series, traza un diagrama de dispersión de los dos precios. Esto nos permite inspeccionar visualmente si existe una relación lineal entre las dos series y, por lo tanto, si es un buen candidato para el procedimiento OLS y la posterior prueba ADF:

```
#cadf.py

definitivamenteplot_scatter_series(df, ts1, ts2):
    plt.xlabel('%s Precio ($)' % ts1) plt.ylabel('%s
    Precio ($)' % ts2)
    plt.title('%s y %s Diagrama de dispersión de precios' % (ts1, ts2))
    plt.scatter(df[ts1], df[ts2])
    plt.mostrar()
```

La tercera función, plot_residuals, está diseñada para trazar los valores residuales del modelo lineal ajustado de las dos series de precios. Esta función requiere que pandas DataFrame tenga una columna "res", que represente los precios residuales:

```
#cadf.py

definitivamenteplot_residuals(df):
    meses = mdates.MonthLocator() fig, ax =    # cada mes
    plt.subplots()
    ax.plot(df.index, df["res"], label="Residuals")
    ax.xaxis.set_major_locator(meses)
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y')) ax.set_xlim(datetime.datetime(2012,
    1, 1), datetime.datetime(2013, 1, 1)) ax.grid(True)

    fig.autofmt_xdate()

    plt.xlabel('Mes/Año')
    plt.ylabel('Precio ($)')
    plt.title('Gráfica de residuos')
    plt.legend()

    plt.plot(df["res"])
    plt.mostrar()
```

Finalmente, el procedimiento se envuelve en una función __main__. La primera tarea es descargar los datos de OHLCV para AREX y WLL de Yahoo Finance. Luego creamos un DataFrame separado, df, utilizando el mismo índice que el marco AREX para almacenar ambos valores de precios de cierre ajustados. Luego trazamos la serie de precios y el diagrama de dispersión.

Una vez completadas las gráficas, los residuales se calculan llamando a la función pandas ols en las series WLL y AREX. Esto nos permite calcular la β relación de cobertura. Luego, la relación de cobertura se usa para crear una columna "res" a través de la formación de la combinación lineal de WLL y AREX.

Finalmente se grafican los residuales y se realiza la prueba ADF sobre los residuales calculados. Luego imprimimos los resultados de la prueba ADF:

```
#cadf.py

si__nombre__ == "__principal__":
    inicio = fechahora.fechahora(2012, 1, 1) fin =
    fechahora.fechahora(2013, 1, 1)

    arex = web.DataReader("AREX", "yahoo", inicio, final) wll =
    web.DataReader("WLL", "yahoo", inicio, final)

    df = pd.DataFrame(index=arex.index)
    df["AREX"] = arex["Adj Cerrar"] df["WLL"] =
    wll["Adj Cerrar"]

    # Trazar las dos series de tiempo
    plot_price_series(df, "AREX", "WLL")

    # Mostrar un gráfico de dispersión de las dos series de
    tiempo plot_scatter_series(df, "AREX", "WLL")

    # Calcular la relación de cobertura óptima
    "beta" res = ols(y=df['WLL'], x=df["AREX"]) beta_hr =
    res.beta.x

    # Calcular los residuos de la combinación lineal df["res"] =
    df["WLL"] - beta_hr*df["AREX"]

    # Graficar los residuos
    plot_residuals(df)

    # Calcule y emita la prueba CADF en los residuos cadf =
    ts.adfuller(df["res"]) pprint.pprint(cadf)
```

El resultado del código (junto con los diagramas de Matplotlib) es el siguiente:

```
(-2.9607012342275936,
 0.038730981052330332,
 0,
 249,
 {'1%': -3.4568881317725864, '10%':
  -2.5729936189738876, '5%':
  -2.8732185133016057},
  601.96849256295991)
```

Puede verse que el estadístico de prueba calculado de -2,96 es menor que el valor crítico del 5 % de -2,87, lo que significa que podemos rechazar la hipótesis nula de que no existe una relación de cointegración al nivel del 5 %. Por tanto, podemos concluir, con un grado razonable de certeza, que AREX y WLL poseen una relación de cointegración, al menos para el período de tiempo considerado.

Usaremos este par en capítulos posteriores para crear una estrategia comercial real utilizando un sistema de backtesting implementado basado en eventos.

10.4 ¿Por qué pruebas estadísticas?

Fundamentalmente, en lo que respecta al comercio algorítmico, las pruebas estadísticas descritas anteriormente son tan útiles como las ganancias que generan cuando se aplican a las estrategias comerciales. Por lo tanto, seguramente tiene sentido simplemente evaluar el desempeño a nivel de estrategia, en oposición al nivel de precio/serie de tiempo. ¿Por qué tomarse la molestia de calcular todas las métricas anteriores cuando simplemente podemos usar análisis de nivel comercial, medidas de riesgo/recompensa y evaluaciones de reducción?

En primer lugar, cualquier estrategia comercial implementada basada en una medida estadística de serie temporal tendrá una muestra mucho más grande con la que trabajar. Esto es simplemente porque al calcular estas pruebas estadísticas, estamos haciendo uso de cada*bar*de información, en lugar de cada*comercio*. Habrá muchas menos operaciones de ida y vuelta que barras y, por lo tanto, la importancia estadística de cualquier métrica a nivel de operación será mucho menor.

En segundo lugar, cualquier estrategia que implementemos dependerá de ciertos parámetros, como los períodos retrospectivos para las medidas móviles o las medidas de puntuación z para entrar/salir de una operación en una configuración de reversión a la media. Por lo tanto, las métricas de nivel de estrategia solo son apropiadas*para estos parámetros*, mientras que las pruebas estadísticas son válidas para la muestra de serie temporal subyacente.

En la práctica queremos calcular ambos conjuntos de estadísticas. Python, a través de las bibliotecas statsmodels y pandas, hace que esto sea extremadamente sencillo. ¡El esfuerzo adicional es en realidad bastante mínimo!

Capítulo 11

Pronóstico

En este capítulo crearemos un proceso estadísticamente robusto para pronosticar series temporales financieras. Estos pronósticos formarán la base para futuras estrategias comerciales automatizadas. Ampliaremos el tema del aprendizaje estadístico discutido en los capítulos anteriores y usaremos un grupo de*algoritmos de clasificación*para ayudarnos a predecir la dirección del mercado de las series temporales financieras.

Dentro de este capítulo haremos uso descikit-aprender,una biblioteca de aprendizaje automático estadístico para Python. Scikit-learn contiene implementaciones "listas para usar" de muchas técnicas de aprendizaje automático. Esto no solo nos ahorra una gran cantidad de tiempo en la implementación de nuestros algoritmos comerciales, sino que minimiza el riesgo de errores introducidos por nuestro propio código. También permite verificación adicional contra bibliotecas de aprendizaje automático escritas en otros paquetes como R o C++. Esto nos da mucha confianza si necesitamos crear nuestra propia implementación personalizada, por razones de velocidad de ejecución, digamos.

Comenzaremos discutiendo formas de medir el desempeño del pronosticador para el caso particular de las técnicas de aprendizaje automático utilizadas. Luego, consideraremos los factores predictivos que se pueden usar en las técnicas de pronóstico y cómo elegir buenos factores. Luego consideraremos varios algoritmos clasificadores supervisados. Finalmente, intentaremos pronosticar la dirección diaria del S&P500, que luego formará la base de una estrategia comercial algorítmica.

11.1 Medición de la precisión del pronóstico

Antes de discutir las opciones de predictor y algoritmos de clasificación específicos, debemos discutir sus características de rendimiento y cómo evaluarlas. La clase particular de métodos que nos interesa implica*clasificación supervisada binaria*. Es decir, intentaremos predecir si el rendimiento porcentual para un día futuro en particular es positivo o negativo (es decir, si nuestro activo financiero ha subido o bajado de precio).

En un pronosticador de producción, utilizando una técnica de tipo regresión, estaríamos muy preocupados por la*magnitud*de esta predicción y las desviaciones de la predicción del valor real.

Para evaluar el desempeño de estos clasificadores podemos hacer uso de las siguientes dos medidas, a saber, laTasa de aciertosyMatriz de confusión.

11.1.1 Tasa de aciertos

La pregunta más simple que podríamos hacerle a nuestro clasificador supervisado es*"¿Cuántas veces predijimos la dirección correcta, como porcentaje de todas las predicciones?"*. Esto motiva la definición de la *tasa de aciertos de entrenamiento*viene dada por la siguiente fórmula[9]:

$$\frac{1}{norte}\sum_{j=1}^{norte} yo(y_j=\hat{y}_j) \tag{11.1}$$

Dónde$\hat{y}_j$es la predicción (hacia arriba o hacia abajo) para el$j$período de tiempo (por ejemplo, un día) usando un clasificador particular.$yo(y_j=\hat{y}_j)$es el*función indicadora*y es igual a 1 si$y_j=\hat{y}_j$y 0 si$y_j6=\hat{y}_j$.

Por lo tanto, la tasa de aciertos proporciona un valor porcentual en cuanto al número de veces que un clasificador predijo correctamente la dirección hacia arriba o hacia abajo.

Scikit-Learn proporciona un método para calcular la tasa de aciertos como parte del proceso de clasificación/formación.

11.1.2 Matriz de confusión

los*matriz de confusión*(o*mesa de contingencia*) es el siguiente paso lógico después de calcular la tasa de aciertos. Está motivado por la pregunta "¿Cuántas veces predijimos correctamente hacia arriba y cuántas veces predijimos correctamente hacia abajo? ¿Difierieron sustancialmente?".

Por ejemplo, podría resultar que un algoritmo en particular sea consistentemente más preciso en la predicción de "días inactivos". Esto motiva una estrategia que enfatiza la venta corta de un activo financiero para aumentar la rentabilidad.

Una matriz de confusión caracteriza esta idea al determinar la*tasa de falsos positivos*(conocido estadísticamente como un error Tipo I) y*tasa de falsos negativos*(conocido estadísticamente como un error Tipo II) para un clasificador supervisado. Para el caso de clasificación binaria (hacia arriba o hacia abajo) tendremos una matriz de 2x2:

$$\begin{pmatrix} tu_T & tu_F \\ D_F & D_T \end{pmatrix}$$

Dóndetu_Trepresenta períodos correctamente clasificados,tu_Frepresenta periodos ascendentes clasificados incorrectamente (es decir, clasificados como descendentes),$D_F$representa periodos bajos incorrectamente clasificados (es decir, clasificados como altos) y$D_T$representa periodos inactivos correctamente clasificados.

Además de la tasa de aciertos, Scikit-Learn proporciona un método para calcular la matriz de confusión como parte del proceso de clasificación/capacitación.

11.2 Elección de factores

Uno de los aspectos más cruciales de la previsión de precios de activos es la elección de los factores utilizados como predictores. Hay una cantidad asombrosa de factores potenciales para elegir y esto puede parecer abrumador para una persona que no esté familiarizada con los pronósticos financieros. Sin embargo, incluso las técnicas simples de aprendizaje automático producirán resultados relativamente buenos cuando se utilizan con factores bien elegidos. Tenga en cuenta que lo contrario no suele ser el caso. "Lanzar un algoritmo a un problema" generalmente conducirá a una pobre precisión de pronóstico.

La elección de factores se lleva a cabo tratando de determinar los impulsores fundamentales del movimiento de activos. En el caso del S&P500, está claro que los 500 constituyentes, de manera ponderada, serán impulsores fundamentales del precio, ¡por definición! Claramente, sabríamos el precio exacto de la serie S&P500 si supiéramos el valor instantáneo de sus componentes, pero ¿hay algún poder predictivo en el uso del historial previo de rendimientos de cada componente para predecir la serie en sí?

Alternativamente, ¿podríamos considerar los tipos de cambio con países que realizan mucho comercio con los EE. UU. como impulsores del precio? Incluso podríamos considerar factores económicos y corporativos más fundamentales, como las tasas de interés, la inflación, las ganancias trimestrales.

La precisión del pronosticador se debe en gran parte a la habilidad del modelador para determinar los factores correctos antes de realizar el ajuste del modelo.

11.2.1 Factores de precio rezagado y volumen

El primer tipo de factor que a menudo se considera al pronosticar una serie de tiempo son los valores históricos anteriores de la propia serie de tiempo. Así un conjunto de$pags$factores podrían obtenerse fácilmente creando$pags$retrasos del precio de cierre de la serie temporal. Considere una serie de tiempo diaria. Para cada día actual en particular$k$, los factores serían los valores diarios históricos en períodos de tiempo$k$-1,k-2, ... ,$k - pag$.

Además de la propia serie de precios, también podemos incorporar el volumen negociado como indicador, ya que se proporciona cuando se utilizan datos de OHLCV (como se obtiene de Yahoo Finance, Google Finance o Quandl, por ejemplo). Así podemos crear un$pags$+1-vector de características dimensionales para cada día de la serie temporal, que incorpora el$pags$lapsos de tiempo y la serie de volumen. Esto lleva naturalmente

a un conjunto de pares (X_k, y_k)representando a la*pags*+1-vector de característica dimensional$X_k$en el díaky el precio de cierre actual actual en el díak,y_k. Esto es todo lo que necesitamos para comenzar un ejercicio de clasificación supervisada.

A continuación, consideraremos una serie de tiempo tan retrasada para el S&P500 y aplicaremos múltiples técnicas de aprendizaje automático para ver si podemos pronosticar su dirección.

11.2.2 Factores externos

Si bien las series de tiempo rezagadas y la información de volumen son un buen punto de partida para el análisis de series de tiempo, estamos lejos de limitarnos a tales datos. Hay una gran cantidad de series de tiempo macroeconómicas y series de precios de activos sobre las cuales considerar los pronósticos. Por ejemplo, es posible que deseemos proporcionar un pronóstico a largo plazo de los precios de los productos básicos en función de los patrones climáticos, o determinar los movimientos de dirección de los precios de divisas a través de los movimientos de las tasas de interés internacionales.

Si tal relación entre series puede determinarse y demostrarse que es estadísticamente significativa, entonces estamos en el punto de poder considerar un modelo comercial sólido. No nos detendremos demasiado en tales relaciones aquí, ya que nuestro objetivo es introducir la idea de técnicas de modelado y aprendizaje automático. Es bastante fácil formular hipótesis sobre las relaciones económicas y obtener los datos de series temporales de un repositorio como Quandl o directamente de los sitios web de estadísticas gubernamentales.

11.3 Modelos de clasificación

El campo del aprendizaje automático es amplio y hay muchos modelos para elegir, particularmente en el ámbito de la clasificación supervisada. Los nuevos modelos se están introduciendo mensualmente a través de la literatura académica. No sería práctico proporcionar una lista exhaustiva de clasificadores supervisados en este capítulo, sino que consideraremos algunas de las técnicas más populares del campo.

11.3.1 Regresión logística

La primera técnica que consideraremos esRegresión logística (LR). En nuestro caso vamos a utilizar la regresión logística para medir la relación entre un*variable dependiente categórica binaria* (es decir, períodos de "alza" o "baja") y múltiples*Variables continuas*, como los rendimientos porcentuales rezagados de un activo financiero.

El modelo de regresión logística proporciona la*probabilidad*que un período de tiempo subsiguiente en particular se clasificará como "arriba" o "abajo". Así, el modelo introduce una*parámetro*, a saber, el umbral de probabilidad para clasificar si un período de tiempo subsiguiente es "alto" o "abajo". A continuación, consideraremos que este umbral es del 50 % (es decir, 0,5), pero ciertamente se puede modificar para producir predicciones alternativas.

La regresión logística se basa en la fórmula logística para modelar la probabilidad de obtener un día "alto" ($Y=tu$) basado en los factores continuos.

En este caso, considere la situación en la que estamos interesados en predecir el período de tiempo subsiguiente a partir de los dos rendimientos retrasados anteriores, que denotaremos por (L_1,L_2). La siguiente fórmula da la probabilidad de tener un día al alza, dado que hemos observado los rendimientos en los períodos de tiempo anteriores,L_1yL_2:

$$pags(Y=U/L_1,L_2) = \frac{mi^{\beta_0+\beta_1 L_1+\beta_2 L_2}}{1+mi^{\beta_0+\beta_1 L_1+\beta_2 L_2}} \tag{11.2}$$

La función logística se usa en lugar de una función lineal (es decir, en regresión lineal) porque proporciona una probabilidad entre [0,1]para todos los valores deL_1yL_2. En una configuración de regresión lineal, es posible obtener probabilidades negativas para estas variables continuas, por lo que necesitamos otra función.

Para ajustar el modelo (es decir, estimar la$\beta_i$coeficientes) elmétodo de máxima verosimilitudse usa Afortunadamente para nosotros la implementación del ajuste y predicción de la regresión logística

El modelo ya está manejado por la biblioteca Scikit-Learn. La técnica se describirá a continuación cuando intentemos pronosticar la dirección del S&P500.

11.3.2 Análisis Discriminante

El análisis discriminante es una técnica estadística alternativa a la regresión logística. Si bien la regresión logística es menos restrictiva en sus suposiciones que el análisis discriminante, puede brindar un mayor rendimiento predictivo si se cumplen las suposiciones más restrictivas.

Ahora consideraremos un método lineal y un método no lineal de análisis discriminante.

Análisis Discriminante Lineal

En la regresión logística, modelamos la probabilidad de ver un período de tiempo "alcista", dados los dos retornos retrasados anteriores ($PAGS(Y=U|L_1,L_2)$) como una distribución condicional de la respuesta$Y$dados los predictoresL_i, utilizando una función logística.

En el Análisis Discriminante Lineal (LDA) la distribución de la$L_i$las variables se modelan por separado, dado$Y$,y$PAGS(Y=U|L_1,L_2)$se obtiene mediante el teorema de Bayes.

Esencialmente, LDA resulta de asumir que los predictores se extraen de una distribución gaussiana multivariada. Después de calcular las estimaciones de los parámetros de esta distribución, los parámetros se pueden insertar en el teorema de Bayes para hacer predicciones sobre a qué clase pertenece una observación.

Una suposición matemática importante de LDA es que todos*clases*(por ejemplo, "arriba" y "abajo") comparten el mismo*Matriz de covarianza*.

No me detendré en las fórmulas para estimar la distribución o*probabilidades posteriores*que se necesitan para hacer predicciones, ya que, una vez más, scikit-learn se encarga de esto por nosotros.

Análisis Discriminante Cuadrático

El Análisis Discriminante Cuadrático (QDA) está estrechamente relacionado con LDA. La diferencia significativa es que cada clase ahora puede poseer su propia matriz de covarianza.

QDA generalmente funciona mejor cuando los límites de decisión no son lineales. LDA generalmente funciona mejor cuando hay menos observaciones de entrenamiento (es decir, cuando se necesita reducir la varianza). QDA, por otro lado, funciona bien cuando el conjunto de entrenamiento es grande (es decir, la varianza es menos preocupante). El uso de uno u otro se reduce en última instancia a la compensación de sesgo-varianza.

Al igual que con LR y LDA, Scikit-Learn se encarga de la implementación de QDA, por lo que solo debemos proporcionarle datos de prueba/entrenamiento para la estimación y predicción de parámetros.

11.3.3 Máquinas de vectores de soporte

Para motivar las máquinas de vectores de soporte (SVM), debemos considerar la idea de un clasificador que separa diferentes clases a través de un límite de separación lineal. Si existiera una separación tan directa, podríamos crear un clasificador supervisado basado únicamente en decidir si las nuevas características se encuentran por encima o por debajo de este plano de clasificación lineal. En realidad, tales separaciones rara vez existen en situaciones comerciales cuantitativas y, como tal, debemos considerar*clasificadores de margen suave*o Clasificadores de vectores de soporte (SVC).

Los SVC funcionan al intentar ubicar un límite de separación lineal en el espacio de características que clasifique correctamente la mayoría, pero no todas, las observaciones de entrenamiento mediante la creación de un límite de separación óptimo entre las dos clases. A veces, tal límite es bastante efectivo si la separación de clases es principalmente lineal. Sin embargo, otras veces tales separaciones no son posibles y es necesario utilizar otras técnicas.

La motivación detrás de la extensión de un SVC es permitir límites de decisión no lineales. Este es el dominio de laMáquina de vectores de soporte (SVM).La principal ventaja de las SVM es que permiten una ampliación no lineal del espacio de características para incluir una no linealidad significativa, al tiempo que conservan una eficiencia computacional significativa, utilizando un proceso conocido como "truco del núcleo".

Las SVM permiten límites de decisión no lineales a través de muchas opciones diferentes de "núcleo". En particular, en lugar de usar un límite de separación completamente lineal como en el SVC, podemos usar cuadrático

polinomios, polinomios de orden superior o incluso núcleos radiales para describir límites no lineales. Esto nos da un grado significativo de flexibilidad, a costa del siempre presente sesgo en nuestras estimaciones.

Usaremos el SVM a continuación para probar y dividir el espacio de características (es decir, los factores de precio y volumen rezagados) a través de un límite no lineal que nos permita hacer predicciones razonables sobre si el día siguiente será un movimiento hacia arriba o hacia abajo.

11.3.4 Árboles de decisión y bosques aleatorios

Los árboles de decisión son una técnica de clasificación supervisada que utiliza una estructura de árbol para dividir el espacio de características en subconjuntos recursivos a través de una "decisión" en cada nodo del árbol.

Por ejemplo, uno podría preguntar si el precio de ayer estuvo por encima o por debajo de cierto umbral, lo que inmediatamente divide el espacio de características en dos subconjuntos. Para cada uno de los dos subconjuntos, se podría preguntar si el volumen estaba por encima o por debajo de un umbral, creando así cuatro subconjuntos separados. Este proceso continúa hasta que ya no se puede obtener más poder predictivo mediante la partición.

Un árbol de decisión proporciona un mecanismo de clasificación naturalmente interpretable en comparación con los enfoques opacos más de "caja negra" de SVM o analizadores discriminantes y, por lo tanto, es una técnica de clasificación supervisada popular.

A medida que ha aumentado el poder computacional, ha surgido un nuevo método para atacar el problema de la clasificación, el de*aprendizaje conjunto*. La idea básica es simple. Cree una gran cantidad de clasificadores a partir del mismo modelo base y entrénelos a todos con diferentes parámetros. A continuación, combine los resultados de la predicción en un promedio para obtener, con suerte, una precisión de predicción mayor que la provocada por cualquiera de los constituyentes individuales.

Uno de los métodos de conjunto más difundidos es el de unbosque aleatorio,que toma múltiples aprendices del árbol de decisiones (generalmente decenas de miles o más) y combina las predicciones. Tales conjuntos a menudo pueden funcionar extremadamente bien. Scikit-Learn viene convenientemente con una clase RandomForestClassifier (RFC) en suconjuntomódulo.

Los dos parámetros principales de interés para el RFC son`n_estimadores`,que describe cuántos árboles de decisión crear, y`n_trabajos`,que describe cuántos núcleos de procesamiento distribuir los cálculos. Discutiremos estas configuraciones en la sección de implementación a continuación.

11.3.5 Análisis de componentes principales

Todas las técnicas anteriores descritas anteriormente pertenecen a la*clasificación supervisada*dominio. Un enfoque alternativo para realizar la clasificación es no supervisar el procedimiento de entrenamiento y, en cambio, permitir que un algoritmo determine las "características" por sí mismo. Tales métodos se conocen como *aprendizaje sin supervisión*tecnicas

Los casos de uso comunes para las técnicas no supervisadas incluyen la reducción del número de dimensiones de un problema a solo aquellas que se consideran importantes, el descubrimiento de temas entre grandes cantidades de documentos de texto o el descubrimiento de características que pueden proporcionar poder predictivo en el análisis de series temporales.

De interés para nosotros en esta sección es el concepto de*reducción de dimensionalidad*, cuyo objetivo es identificar los componentes más importantes de un conjunto de factores que proporcionan la mayor previsibilidad. En particular, vamos a utilizar una técnica no supervisada conocida comoAnálisis de componentes principales (PCA)para reducir el tamaño del espacio de características antes de su uso en nuestros clasificadores supervisados.

La idea básica de un PCA es transformar un conjunto de variables posiblemente correlacionadas (como con la autocorrelación de series de tiempo) en un conjunto de variables linealmente no correlacionadas conocido como el *componentes principales*. Dichos componentes principales se ordenan de acuerdo con la cantidad de varianza que describen, de manera ortogonal. Por lo tanto, si tenemos un espacio de características de muy alta dimensión (más de 10 características), entonces podríamos reducir el espacio de características a través de PCA a quizás 2 o 3 componentes principales que proporcionen casi toda la variabilidad en los datos, lo que conduciría a un más robusto. clasificador supervisado cuando se usa en este conjunto de datos reducido.

11.3.6 ¿Qué pronosticador?

En situaciones financieras cuantitativas donde hay una gran cantidad de datos de entrenamiento, se debe considerar el uso de un modelo como una Máquina de vectores de soporte (SVM). Sin embargo, las SVM sufren de falta de interpretabilidad. Este no es el caso de los conjuntos Decision Trees y Random Forest.

Estos últimos se utilizan a menudo para preservar la interpretabilidad, algo que los clasificadores de "caja negra" como SVM no proporcionan.

En última instancia, cuando los datos son tan extensos (por ejemplo, datos de ticks), importará muy poco qué clasificador se utilice en última instancia. En esta etapa surgen otros factores como la eficiencia computacional y la escalabilidad del algoritmo. La regla empírica general es que una duplicación de los datos de entrenamiento proporcionará un aumento lineal en el rendimiento, pero a medida que el tamaño de los datos se vuelve sustancial, esta mejora se reduce a un aumento sublineal en el rendimiento.

La teoría estadística y matemática subyacente para los clasificadores supervisados es bastante complicada, pero la intuición básica de cada clasificador es sencilla de entender. Además, tenga en cuenta que cada uno de los siguientes clasificadores tendrá un conjunto diferente de suposiciones sobre cuándo funcionarán mejor, por lo que si encuentra que un clasificador funciona mal, puede deberse a que el conjunto de datos que se utiliza viola una de las suposiciones utilizadas para generar la teoría.

Clasificador bayesiano ingenuo

Si bien no hemos considerado un clasificador Naive Bayes en nuestros ejemplos anteriores, quería incluir una discusión al respecto para completarlo. Naive Bayes (específicamente Multinomial Naive Bayes - MNB) es bueno para usar cuando existe un conjunto de datos limitado. Esto se debe a que es un clasificador de alto sesgo. El principal supuesto del clasificador MNB es el de la independencia condicional. Esencialmente, esto significa que no puede discernir interacciones entre características individuales, a menos que se agreguen específicamente como características adicionales.

Por ejemplo, considere una situación de clasificación de documentos, que aparece en la configuración financiera al intentar realizar un análisis de sentimiento. El MNB podría aprender que palabras individuales como "gato" y "perro" podrían referirse respectivamente a documentos relacionados con gatos y perros, pero la frase "gatos y perros" (jerga británica para llover fuertemente) no sería considerada meteorológica por el clasificador! El remedio a esto sería tratar a los "perros y gatos" como una característica adicional, específicamente, y luego asociar eso a una categoría meteorológica.

Regresión logística

La regresión logística proporciona algunas ventajas sobre un modelo Naive Bayes en el sentido de que hay menos preocupación por la correlación entre características y, por la naturaleza del modelo, existe una interpretación probabilística de los resultados. Esto se adapta mejor a un entorno en el que es necesario utilizar umbrales. Por ejemplo, podríamos desear colocar un umbral del 80 % (digamos) en un resultado "alto" o "bajo" para que se seleccione correctamente, en lugar de elegir la categoría de mayor probabilidad. En este último caso, la predicción de "arriba" podría ser del 51 % y la predicción de "abajo" podría ser del 49 %. Establecer la categoría en "arriba" no es una predicción muy fuerte en este caso.

Árbol de decisión y bosques aleatorios

Los árboles de decisión (DT) dividen un espacio en una jerarquía de opciones booleanas que conducen a una categorización o agrupación basada en las decisiones respectivas. Esto los hace altamente interpretables (¡asumiendo un número "razonable" de decisiones/nodos en el árbol!). DT tiene muchos beneficios, incluida la capacidad de manejar interacciones entre funciones, además de ser *no paramétrico*.

También son útiles en casos en los que no es sencillo (o imposible) separar linealmente los datos en clases (que es una condición necesaria para las máquinas de vectores de soporte). La desventaja de usar árboles de decisión individuales es que tienden a sobreajustarse (alta varianza). Este problema se resuelve usando un bosque aleatorio. Los bosques aleatorios son en realidad algunos de los "mejores" clasificadores cuando se usan en competencias de aprendizaje automático, por lo que siempre deben tenerse en cuenta.

Máquinas de vectores soporte

Las máquinas de vectores de soporte (SVM), aunque poseen un procedimiento de ajuste complicado, en realidad son relativamente sencillas de entender. Las SVM lineales esencialmente intentan dividir un espacio utilizando límites de separación lineal, en múltiples grupos distintos. Para ciertos tipos de datos, esto puede funcionar extremadamente bien y conduce a buenas predicciones. Sin embargo, una gran cantidad de datos no se pueden separar linealmente, por lo que las SVM lineales pueden tener un rendimiento deficiente aquí.

La solución es modificar el núcleo utilizado por SVM, lo que tiene el efecto de permitir límites de decisión no lineales. Por lo tanto, son modelos bastante flexibles. Sin embargo, se debe elegir el límite correcto de SVM para obtener los mejores resultados. SVM son especialmente buenos en problemas de clasificación de texto con alta dimensionalidad. Están en desventaja por su complejidad computacional, dificultad de ajuste y el hecho de que el modelo ajustado es difícil de interpretar.

11.4 Pronóstico del movimiento del índice bursátil

El S&P500 es un índice ponderado de las 500 empresas más grandes que cotizan en bolsa por capitalización de mercado en el mercado de valores de EE. UU. A menudo se utiliza como referencia de acciones. Existen muchos productos derivados para permitir la especulación o la cobertura del índice. En particular, el contrato de futuros del índice S&P500 E-Mini es un medio extremadamente líquido para negociar el índice.

En esta sección vamos a utilizar un conjunto de clasificadores para predecir la dirección del precio de cierre en el día.*k*basado únicamente en la información de precios conocida en el día*k* -1.Un movimiento direccional hacia arriba significa que el precio de cierre en*k*es más alto que el precio en*k*−1,mientras que un movimiento a la baja implica un precio de cierre en*k*más bajo que en*k* -1.

Si podemos determinar la dirección del movimiento de una manera que exceda significativamente una tasa de aciertos del 50%, con un error bajo y una buena significancia estadística, entonces estamos en camino de formar una estrategia comercial sistemática básica basada en nuestros pronósticos.

11.4.1 Implementaciones de Python

Para la implementación de estos predictores haremos uso de NumPy, Pandas y Scikit-Learn, los cuales fueron instalados en los capítulos anteriores.

El primer paso es importar los módulos y bibliotecas relevantes. Vamos a importar los clasificadores LogisticRegression, LDA, QDA, LinearSVC (una máquina de vectores de soporte lineal), SVC (una máquina de vectores de soporte no lineal) y RandomForest para este pronóstico:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

# pronóstico.py

de__futuro__importarimprimir_funcion

importarfecha y hora
importarnumpy como np
importarpandas como pd
importaraprender

depandas.io.datosimportarLector de datos
desklearn.ensembleimportarRandomForestClassifier de
sklearn.modelo_linealimportarRegresión logística desklearn.lda
importarLDA
desklearn.metricsimportarmatriz de confusión de
sklearn.qdaimportarQDA
desklearn.svmimportarSVC lineal, SVC
```

Ahora que se importaron las bibliotecas, necesitamos crear un marco de datos de pandas que contenga los rendimientos porcentuales retrasados de un número anterior de días (el valor predeterminado es cinco).crear_lag_serie tomará un símbolo bursátil (como lo reconoce Yahoo Finance) y creará un marco de datos retrasado durante el período especificado. El código está bien comentado, por lo que debería ser sencillo ver lo que está sucediendo:

```
definitivamentecreate_lagged_series(símbolo, start_date, end_date, lags=5):
    """
    Esto crea un Pandas DataFrame que almacena el
```

```
    rendimientos porcentuales del valor de cierre ajustado de una acción
    obtenida de Yahoo Finance, junto con una cantidad de rendimientos
    retrasados   de los días de negociación anteriores (retrasos
    predeterminados de 5 días). También se incluyen el volumen de
    negociación, así como la dirección del día anterior. """


    # Obtener información bursátil de Yahoo Finance ts =
    lector de datos (
        símbolo, "yahoo",
        start_date-datetime.timedelta(days=365), end_date

    )

    # Crear el nuevo marco de datos retrasado
    tslag = pd.DataFrame(index=ts.index)
    tslag["Hoy"] = ts["Cerrar Adj"] tslag["Volumen"] =
    ts["Volumen"]

    # Cree la serie de retraso desplazada de los valores de cierre del período comercial anterior
    porienrango (0, retrasos):
        tslag["Lag%s" % str(i+1)] = ts["Adj Cerrar"].shift(i+1)

    # Crear el DataFrame de devoluciones tsret =
    pd.DataFrame(index=tslag.index) tsret["Volumen"] =
    tslag["Volumen"] tsret["Hoy"] = tslag["Hoy"].pct_change()*100.0


    # Si alguno de los valores de porcentaje devuelve igual a cero, configúrelos en
    # un número pequeño (detiene problemas con el modelo QDA en Scikit-Learn)
    poryo,xenenumerar(tsret["Hoy"]):
        si(abs(x) < 0,0001):
            tsret["Hoy"][i] = 0.0001

    # Crear las columnas de rendimientos porcentuales rezagados
    porienrango (0, retrasos):
        tsret["Lag%s" % str(i+1)] = \ tslag["Lag%s" %
        str(i+1)].pct_change()*100.0

    # Cree la columna "Dirección" (+1 o -1) que indica un día arriba/abajo tsret["Dirección"]
    = np.sign(tsret["Hoy"]) tsret = tsret[tsret.index >= fecha_inicio]


    devolvertsret
```

Vinculamos el procedimiento de clasificación junto con un __principal__función. En este caso, vamos a intentar pronosticar la dirección del mercado de valores de EE. UU. en 2005, utilizando datos de rendimiento de 2001 a 2004.

En primer lugar, creamos una serie retrasada del S&P500 utilizando cinco retrasos. La serie también incluye el volumen de negociación. Sin embargo, vamos a restringir el conjunto de predictores para usar solo los dos primeros retrasos. Por lo tanto, implícitamente le estamos diciendo al clasificador que los retrasos adicionales tienen menos valor predictivo.*Por otro lado, este efecto se estudia más concretamente bajo el concepto estadístico de autocorrelación, aunque esto está más allá del alcance del libro.*

Después de crear la matriz predictora*X*y el vector de respuesta*y*, podemos particionar las matrices en una *capacitación*y un*prueba*establecer. El primer subconjunto se usa para entrenar al clasificador, mientras que el último se usa para probar el rendimiento. Vamos a dividir el conjunto de entrenamiento y prueba el 1 de enero de 2005, dejando un año de negociación completo de datos (aproximadamente 250 días) para

el conjunto de prueba.

Una vez que creamos la división de entrenamiento/prueba, necesitamos crear una matriz de modelos de clasificación, cada uno de los cuales está en una tupla con un nombre abreviado adjunto. Si bien no hemos establecido ningún parámetro para los modelos de regresión logística, analizadores discriminantes lineales/cuadráticos o clasificador de vectores de soporte lineal, hemos utilizado un conjunto de parámetros predeterminados para la máquina de vectores de soporte radial (RSVM) y el bosque aleatorio (RF).

Finalmente iteramos sobre los modelos. Entrenamos (ajustamos) cada modelo en los datos de entrenamiento y luego hacemos predicciones en el conjunto de prueba. Finalmente, generamos la tasa de aciertos y la matriz de confusión para cada modelo:

```
si__nombre__ == "__principal__":
    # Crear una serie rezagada del índice bursátil estadounidense S&P500 snpret
    = create_lagged_series(
        " ĜSPC", datetime.datetime(2001,1,10),
        datetime.datetime(2005,12,31), lags=5
    )

    # Use los dos días anteriores de devoluciones como predictor
    # valores, con dirección como respuesta X =
    snpret[["Lag1","Lag2"]] y = snpret["Dirección"]


    # Los datos de prueba se dividen en dos partes: antes y después del 1 de enero de 2005.
    prueba_inicio = fechahora.fechahora(2005,1,1)

    # Crear conjuntos de entrenamiento y
    prueba X_train = X[X.index < start_test]
    X_test = X[X.index >= start_test] y_train =
    y[y.index < start_test] y_test = y[y.index >=
    start_test]

    # Crear los modelos (parametrizados) impresión
    ("Tasas de aciertos/Matrices de confusión:\n")
    modelos= [("LR", Regresión Logística()),
                ("LDA", LDA()),
                ("QDA", QDA()),
                ("LSVC", LinearSVC()),
                ("RSVM", SVC(
                    C=1000000.0, cache_size=200, class_weight=Ninguno, coef0=0.0,
                    grado=3, gamma=0.0001, kernel='rbf', max_iter=-1,
                    probabilidad=Falso, random_state=Ninguno, reducción=Verdadero,
                    tol=0.001 , detallado=Falso)
                ) ,
                ("RF", ClasificadorBosqueAleatorio(
                    n_estimators=1000, criterio='gini', max_ depth=Ninguno,
                    min_samples_split=2, min_samples_leaf=1,
                    max_features='auto', bootstrap=True, oob_score=False,
                    n_jobs=1, random_state=Ninguno, detallado=0)

                ) ]

    # Iterar a través de los modelos por
    metroen modelos:

        # Entrena a cada uno de los modelos en el set de
        entrenamiento m[1].fit(tren_X, tren_y)
```

```
        # Hacer una serie de predicciones en el conjunto de
        prueba pred = m[1].predecir(X_prueba)

        # Muestra la tasa de aciertos y la matriz de confusión para cada modelo
        impresión("%s:\n%0.3f" % (m[0], m[1].score(X_test, y_test))) impresión("%s\n" %
        confusion_matrix(pred, y_test))
```

11.4.2 Resultados

El resultado de todos los modelos de clasificación es el siguiente. Es probable que vea valores diferentes en la salida de RF (bosque aleatorio), ya que su construcción es inherentemente estocástica:

```
Tasas de acierto/matrices de confusión:

LR:
0.560
[[ 35  35]
 [ 76 106]]

LDA:
0.560
[[ 35  35]
 [ 76 106]]

QDA:
0.599
[[ 30  20]
 [ 81 121]]

LSVC:
0.560
[[ 35  35]
 [ 76 106]]

RSVM:
0.563
[[ 9  8]
 [102 133]]

FR:
0.504
[[48 62]
 [63 79]]
```

Tenga en cuenta que todas las tasas de éxito se encuentran entre el 50% y el 60%. Por lo tanto, podemos ver que las variables rezagadas no son muy indicativas de la dirección futura. Sin embargo, si observamos el analizador discriminante cuadrático, podemos ver que su rendimiento predictivo general en el conjunto de prueba está justo por debajo del 60 %.

La matriz de confusión para este modelo (y los demás en general) también establece que la verdadera tasa positiva para los días "bajos" es mucho más alta que los días "altos". Por lo tanto, si vamos a crear una estrategia comercial basada en esta información, podríamos considerar restringir las operaciones a posiciones cortas del S&P500 como un medio potencial para aumentar la rentabilidad.

En capítulos posteriores, utilizaremos estos modelos como base de una estrategia comercial incorporándolos directamente en el marco de backtesting basado en eventos y utilizando un instrumento directo, como un fondo cotizado en bolsa (ETF), para darnos acceso a la negociación. el S&P500.

Parte V

Gestión de Riesgos y Desempeño

Capítulo 12

Medición del desempeño

La medición del rendimiento es un componente absolutamente crucial del comercio algorítmico. Sin una evaluación del desempeño, junto con un sólido mantenimiento de registros, es difícil, si no imposible, determinar si los retornos de nuestra estrategia se deben a la suerte oa alguna ventaja real sobre el mercado.

Para tener éxito en el comercio algorítmico, es necesario conocer todos los factores que pueden afectar la rentabilidad de los intercambios y, en última instancia, las estrategias. Deberíamos estar constantemente tratando de encontrar mejoras en todos los aspectos de la pila de operaciones algorítmicas. En particular, siempre debemos intentar minimizar nuestros costos de transacción (tarifas, comisiones y deslizamientos), mejorar nuestro software y hardware, mejorar la limpieza de nuestras fuentes de datos y buscar continuamente nuevas estrategias para agregar a una cartera. La medición del desempeño en todas estas áreas proporciona un criterio sobre el cual medir las alternativas.

En última instancia, el comercio algorítmico se trata de generar ganancias. Por lo tanto, es imperativo que midamos el desempeño, en múltiples niveles de granularidad, de cómo y por qué nuestro sistema produce estas ganancias. Esto motiva la evaluación del desempeño a nivel de operaciones, estrategias y carteras. En particular estamos buscando:

- Si las reglas sistemáticas codificadas por la estrategia realmente producen un rendimiento consistente y si la estrategia posee un rendimiento positivo en las pruebas retrospectivas.
- Si una estrategia mantiene este desempeño positivo en una implementación en vivo o si necesita ser retirada.
- La capacidad de comparar múltiples estrategias/carteras de modo que podamos reducir el *costo de oportunidad* asociados con la asignación de una cantidad limitada de capital comercial.

Los elementos particulares del análisis cuantitativo del desempeño que nos interesarán son los siguientes:

- Devoluciones -El aspecto más visible de una estrategia comercial se refiere al porcentaje de ganancia desde el inicio, ya sea en un backtest o en un entorno comercial en vivo. Las dos principales medidas de rendimiento aquí son el rendimiento total y la tasa de crecimiento anual compuesto (CAGR).
- Disposiciones -A *reducción* es un período de rendimiento negativo, tal como se define a partir de un *alta marca de agua*, definido en sí mismo como el pico más alto anterior en una estrategia o cartera *curva de equidad*. Definiremos esto más concretamente a continuación, pero por ahora puede pensar en ello como una pendiente descendente (¡algo dolorosa!) en su gráfico de rendimiento.
- Riesgo -El riesgo comprende muchas áreas, y dedicaremos mucho tiempo a repasarlas en el siguiente capítulo, pero generalmente se refiere tanto al riesgo de pérdida de capital, como con retiros, como a la volatilidad de los rendimientos. Este último suele calcularse como una desviación estándar anualizada de los rendimientos.
- Relación Riesgo/Recompensa -Los inversores institucionales están interesados principalmente en *rendimientos ajustados al riesgo*. Dado que una mayor volatilidad a menudo puede conducir a mayores rendimientos a expensas de una mayor

detracciones, siempre les preocupa cuánto riesgo se asume por unidad de rendimiento. En consecuencia, se ha inventado una gama de medidas de desempeño para cuantificar este aspecto del desempeño de la estrategia, a saber, el índice de Sharpe, el índice de Sortino y el índice de CALMAR, entre otros. los*fuera de muestra Sharpe*es a menudo la primera métrica que analizan los inversores institucionales cuando analizan el rendimiento de la estrategia.

- Análisis comercial -Todas las medidas de rendimiento anteriores son aplicables a*estrategias* y*portafolios* . También es instructivo observar el desempeño de operaciones individuales y existen muchas medidas para caracterizar su desempeño. En particular, cuantificaremos el número de operaciones ganadoras/perdedoras, la ganancia media por operación y la proporción de ganancias/pérdidas, entre otros.

Las operaciones son el aspecto más granular de una estrategia algorítmica y, por lo tanto, comenzaremos discutiendo el análisis comercial.

12.1 Análisis comercial

El primer paso para analizar cualquier estrategia es considerar el rendimiento de las operaciones reales. Tales métricas pueden variar dramáticamente entre estrategias. Un ejemplo clásico sería la diferencia en las métricas de rendimiento de una estrategia de seguimiento de tendencias en comparación con una estrategia de reversión a la media.

Las estrategias de seguimiento de tendencias generalmente consisten en muchas operaciones perdedoras, cada una con una pequeña pérdida probable. La menor cantidad de operaciones rentables se produce cuando se ha establecido una tendencia y el rendimiento de estas operaciones positivas puede superar significativamente las pérdidas de la mayor cantidad de operaciones perdedoras. Las estrategias de reversión de la media de comercio de pares muestran el carácter opuesto. Por lo general, consisten en muchas pequeñas operaciones rentables. Sin embargo, si una serie no significa una reversión de la manera esperada, la naturaleza larga/corta de la estrategia puede conducir a pérdidas sustanciales. Esto podría acabar con la gran cantidad de pequeñas ganancias.

Es esencial ser consciente de la naturaleza del perfil comercial de la estrategia y de su propio perfil psicológico, ya que los dos deberán estar alineados. De lo contrario, descubrirá que es posible que no pueda perseverar durante un período de reducción difícil.

Repasamos ahora las estadísticas que nos interesan como el nivel comercial.

12.1.1 Resumen de estadísticas

Al considerar nuestras operaciones, estamos interesados en el siguiente conjunto de estadísticas. Aquí, "período" se refiere al período de tiempo cubierto por la barra comercial que contiene datos OHLCV. Para las estrategias a largo plazo, a menudo se utilizan barras diarias. Para estrategias de mayor frecuencia, podemos estar interesados en barras por hora o por minuto.

- Ganancia/pérdida total (PnL) -El PnL total establece directamente si una operación en particular ha sido rentable o no.
- Período Promedio PnL -El promedio El período PnL establece si una barra, en promedio, genera una ganancia o una pérdida.
- Beneficio máximo del período -La mayor ganancia del período de barra obtenida por esta operación hasta el momento.
- Pérdida Máxima del Período -La pérdida más grande del período de barra hecha por este comercio hasta ahora. ¡Tenga en cuenta que esto no dice nada sobre la pérdida futura del período máximo! Una pérdida futura podría ser mucho mayor que esto.
- Beneficio medio del período -El promedio durante la vida comercial de todos los períodos rentables.
- Período promedio de pérdida -El promedio durante la vida comercial de todos los períodos no rentables.
- Períodos Ganadores -El conteo de todos los períodos ganadores.
- períodos perdedores -La cuenta de todos los períodos perdedores.
- Períodos de porcentaje de ganancia/pérdida -El porcentaje de todos los periodos ganadores frente a los periodos perdedores. Diferirá notablemente para las estrategias de tipo de seguimiento de tendencia y reversión de la media.

Afortunadamente, es sencillo generar esta información a partir de la salida de nuestra cartera y, por lo tanto, se elimina por completo la necesidad de llevar un registro manual. Sin embargo, esto conlleva el peligro de que nunca nos detengamos a analizar los datos.

Es imperativo que las operaciones se evalúen al menos una o dos veces al mes. Hacerlo es un sistema útil de detección de alerta temprana que puede ayudar a identificar cuándo el desempeño de la estrategia comienza a degradarse. A menudo es mucho mejor que simplemente considerar solo el PnL acumulativo.

12.2 Análisis de la estrategia y la cartera

El análisis del nivel de operaciones es extremadamente útil en estrategias a más largo plazo, particularmente con estrategias que emplean operaciones complejas, como las que involucran derivados. Para las estrategias de mayor frecuencia, estaremos menos interesados en cualquier operación individual y, en cambio, querremos considerar las medidas de rendimiento de la estrategia. Obviamente, para las estrategias a más largo plazo, estamos igualmente interesados en el rendimiento general de la estrategia. Estamos interesados principalmente en las siguientes tres áreas clave:

- Análisis de devoluciones -Los retornos de una estrategia encapsulan el concepto de rentabilidad. En entornos institucionales, generalmente se cotizan netos de tarifas y, por lo tanto, brindan una imagen real de cuánto dinero se ganó con el dinero invertido. Las devoluciones pueden ser difíciles de calcular, especialmente con entradas/salidas de efectivo.
- Análisis de Riesgo/Recompensa -Generalmente, la primera consideración que los inversores externos tendrán en una estrategia es que está fuera de muestra. *Relación de Sharpe* (que describimos a continuación). Esta es una métrica estándar de la industria que intenta caracterizar cuánto retorno se logró por unidad de riesgo.
- Análisis de reducción -En un entorno institucional, este es probablemente el más importante de los tres aspectos. El perfil y el alcance de las disposiciones de una estrategia, cartera o fondo forman un componente clave en la gestión de riesgos. Definiremos las reducciones a continuación.

A pesar de que he enfatizado su desempeño institucional, como comerciante minorista estas siguen siendo métricas muy importantes y con una gestión de riesgos adecuada (ver el próximo capítulo) formarán la base de un procedimiento de evaluación de estrategia continua.

12.2.1 Análisis de devoluciones

Las cifras más citadas cuando se analiza el desempeño de la estrategia, tanto en entornos institucionales como minoristas, a menudo son *regreso trotal*, *rendimientos anuales* y *rendimientos mensuales*. Es extremadamente común ver un boletín informativo sobre el desempeño de los fondos de cobertura con una "cuadrícula" de rendimiento mensual. Además, todos querrán saber cuál es el "retorno" de la estrategia.

El rendimiento total es relativamente sencillo de calcular, al menos en un entorno minorista sin inversores externos ni entradas/salidas de efectivo. En términos porcentuales se calcula simplemente como:

$$r_t = (PAGS_F - PAGS_I)/PAGS_I \times 100 \tag{12.1}$$

Dónde r_t es el rendimiento total, $PAGS_F$ es el valor final en dólares de la cartera y $PAGS_I$ es el valor inicial de la cartera. Nos interesa sobre todo *red* rendimiento total, es decir, el valor de la cartera/fondo después de deducir todos los costos comerciales/de negociación.

Tenga en cuenta que esta fórmula solo se aplica a carteras largas sin apalancamiento. Si deseamos agregar ventas en corto o apalancamiento, debemos modificar la forma en que calculamos los rendimientos porque técnicamente estamos operando en una cartera prestada más grande que la utilizada aquí. Esto se conoce como *cartera de margen*.

Por ejemplo, considere el caso en el que una estrategia comercial ha ido en largo 1000 USD de un activo y luego ha ido en corto 1000 USD de otro activo. Esto es un *dólar neutral* cartera y el total *nocional negociado* es de 2.000 dólares. Si se generaron 200 USD a partir de esta estrategia, el rendimiento bruto de este nocional es del 10 %. Se vuelve más complejo cuando se tienen en cuenta los costos de los préstamos y las tasas de interés para financiar el margen. La consideración de estos costos conduce al rendimiento total neto, que es el valor que a menudo se cita como "rendimiento total".

Curva de Equidad

La curva de renta variable es a menudo una de las visualizaciones más destacadas en un informe de rendimiento de un fondo de cobertura, ¡suponiendo que el fondo esté funcionando bien! Es un gráfico del valor de la cartera del fondo a lo largo del tiempo. En esencia, se utiliza para mostrar cómo ha crecido la cuenta desde el inicio del fondo. Del mismo modo, en un entorno minorista se utiliza para mostrar el crecimiento del capital de la cuenta a lo largo del tiempo. Consulte la Fig. 12.2.1 para ver un gráfico de curva de equidad típico:

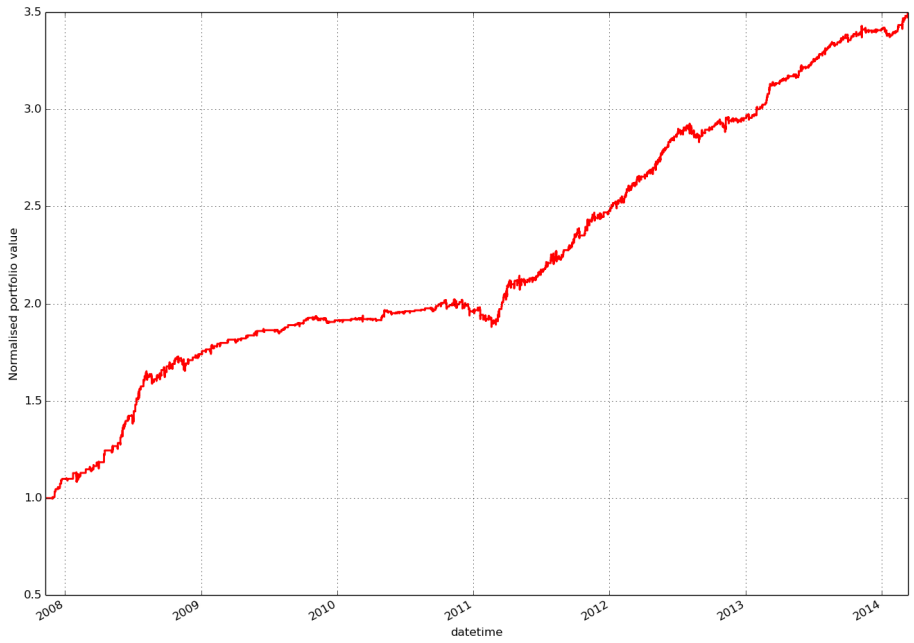


Figura 12.1: Curva típica de renta variable de la estrategia intradía

¿Cuál es el beneficio de tal parcela? En esencia, da un "sabor" en cuanto a la volatilidad pasada de la estrategia, así como una indicación visual de si la estrategia ha sufrido períodos prolongados de estancamiento o incluso reducción. Esencialmente proporciona respuestas sobre cómo se llegó a la cifra de rendimiento total calculada al final del período de negociación de la estrategia.

En una curva de acciones, buscamos determinar cómo los eventos históricos inusuales han dado forma a la estrategia. Por ejemplo, una pregunta común es si hubo un exceso de volatilidad en la estrategia alrededor de 2008. Otra pregunta podría estar relacionada con la consistencia de los rendimientos.

Hay que tener mucho cuidado con la interpretación de las curvas de renta variable, ya que cuando se comercializan, generalmente se muestran como "líneas con pendiente ascendente". Se puede obtener información interesante a través del truncamiento de tales curvas, lo que puede enfatizar períodos de intensa volatilidad o reducción prolongada que, de otro modo, no parecerían tan severos al considerar el período de tiempo completo. Por lo tanto, una curva de capital debe considerarse en contexto con otro análisis, en particular, el análisis de riesgo/recompensa y el análisis de reducción.

12.2.2 Análisis de riesgo/recompensa

Como mencionamos anteriormente, el concepto de análisis riesgo-recompensa es extremadamente importante en un entorno institucional. Esteno essignifica que como inversor minorista podemos ignorar el concepto. Debe prestar mucha atención a las métricas de riesgo/recompensa para su estrategia, ya que tendrán un impacto significativo en sus reducciones, apalancamiento y tasa de crecimiento compuesta general.

Estos conceptos se ampliarán en el próximo capítulo sobre Gestión de riesgos y dinero. Por ahora, analizaremos los índices comunes y, en particular, el índice de Sharpe, que es omnipresente como medida comparativa en las finanzas cuantitativas. Dado que se tiene en tan alta estima en el comercio cuantitativo institucionalizado, entraremos en una cantidad razonable de detalles.

Relación de Sharpe

Considere la situación en la que se nos presentan dos estrategias que poseen rendimientos idénticos. ¿Cómo sabemos cuál contiene más riesgo? Además, ¿qué queremos decir con "más riesgo"? En finanzas, a menudo nos preocupa la volatilidad de los rendimientos y los períodos de reducción. Por lo tanto, si una de estas estrategias tiene una volatilidad de rendimientos significativamente mayor, probablemente la encontraríamos menos atractiva, a pesar de que sus rendimientos históricos podrían ser similares, si no idénticos. Estos problemas de comparación de estrategias y evaluación de riesgos motivan el uso de laRelación de Sharpe.

William Forsyth Sharpe es un economista ganador del premio Nobel, que ayudó a crear el Modelo de fijación de precios de activos de capital (CAPM) y desarrolló el Índice de Sharpe en 1966 (posteriormente actualizado en 1994). La relación de Sharpe$S$se define por la siguiente relación:

$$S = \frac{\text{MI}(R_a - R_b)}{\sqrt{\text{Var}(R_a - R_b)}} \tag{12.2}$$

Dónde$R_a$es el rendimiento del período del activo o estrategia y$R_b$es el período de retorno de un adecuado *punto de referencia*, como una tasa de interés libre de riesgo.

La razón compara el promedio medio de los*rendimientos excesivos*del activo o estrategia con la desviación estándar de esos retornos en exceso. Por lo tanto, una menor volatilidad de los rendimientos conducirá a un mayor índice de Sharpe, suponiendo rendimientos medios idénticos.

El "Ratio de Sharpe" citado a menudo por aquellos que llevan a cabo estrategias comerciales es elsharpe anualizado,cuyo cálculo depende del período de negociación del que se miden los rendimientos. Suponiendo que hay$norte$períodos de negociación en un año, el Sharpe anualizado se calcula de la siguiente manera:

$$S_A = \sqrt{norte}\,\frac{\text{MI}(R_a - R_b)}{\sqrt{\text{Var}(R_a - R_b)}}$$

Tenga en cuenta que el propio índice de Sharpe DEBE calcularse en función del Sharpe de ese tipo de período de tiempo en particular. Para una estrategia basada en el período de negociación de días,$norte$=252 (ya que hay 252 días hábiles en un año, no 365), yR_a, R_bdeben ser los rendimientos diarios. Del mismo modo durante horas $norte$=252$\times$6.5 = 1638,no$norte$=252$\times$24 = 6048,ya que solo hay 6,5 horas en un día de negociación (¡al menos para la mayoría de los mercados de acciones de EE. UU.!).

La fórmula para la relación de Sharpe anterior alude al uso de un*punto de referencia*. Un punto de referencia se utiliza como una "vara de medir" o un "obstáculo" que una estrategia en particular debe superar para que valga la pena considerarla. Por ejemplo, una estrategia simple solo larga que utilice acciones de gran capitalización de EE. UU. debería esperar superar el índice S&P500 en promedio, o igualarlo por menos volatilidad; de lo contrario, ¿qué se gana al no simplemente invertir en el índice a una gestión mucho más baja? honorarios de rendimiento?

La elección del punto de referencia a veces puede ser poco clara. Por ejemplo, ¿debería utilizarse un Fondo Cotizado en Bolsa (ETF) sectorial como punto de referencia de desempeño para acciones individuales, o el propio S&P500? ¿Por qué no el Russell 3000? Del mismo modo, ¿una estrategia de fondos de cobertura debería compararse con un índice de mercado o un índice de otros fondos de cobertura?

También existe la complicación de la "tasa libre de riesgo". ¿Deberían utilizarse bonos del gobierno nacional? ¿Una canasta de bonos internacionales? ¿Facturas a corto o largo plazo? ¿Una mezcla? Claramente, hay muchas maneras de elegir un punto de referencia. El índice de Sharpe generalmente utiliza la tasa libre de riesgo y, a menudo, para las estrategias de acciones de EE. UU., esto se basa en letras del Tesoro del gobierno a 10 años.

En un caso particular, para las estrategias neutrales al mercado, existe una complicación particular con respecto a si utilizar la tasa libre de riesgo o cero como punto de referencia. El índice de mercado en sí mismo no debe utilizarse ya que la estrategia es, por diseño, neutral para el mercado. La elección correcta para una cartera neutral al mercado es no restar la tasa libre de riesgo porque es*autofinanciamiento*. Como usted gana un interés de crédito,R_F,de mantener un margen, el cálculo real de los rendimientos

es: $(R_a + R_f) - R_f = R_a$. Por lo tanto, no existe una sustracción real de la tasa libre de riesgo para las estrategias neutrales en dólares.

A pesar de la prevalencia de la relación de Sharpe dentro de las finanzas cuantitativas, adolece de algunas limitaciones. La relación de Sharpe es *mirando hacia atrás*. Solo da cuenta *histórico* distribución de rentabilidades y volatilidad, no las que se producen en el *futuro*. Cuando se hacen juicios basados en la relación de Sharpe, existe la suposición implícita de que el pasado será similar al futuro. Evidentemente, éste no es siempre el caso, en particular bajo *cambios en el régimen de mercado*.

El cálculo de la relación de Sharpe supone que los rendimientos que se utilizan se distribuyen normalmente (es decir, *gaussiano*). Desafortunadamente, los mercados a menudo sufren de una curtosis superior a la de una distribución normal. Esencialmente, la distribución de rendimientos tiene "colas más gruesas" y, por lo tanto, es más probable que ocurran eventos extremos de lo que una distribución gaussiana nos haría creer. Por lo tanto, la relación de Sharpe es pobre para caracterizar *riesgo de cola*.

Esto se puede ver claramente en estrategias que son altamente propensas a tales riesgos. Por ejemplo, la venta de opciones de compra, también conocidas como "centavos bajo una apisonadora". La venta de opciones de compra genera un flujo constante de primas de opciones a lo largo del tiempo, lo que genera una baja volatilidad de los rendimientos, con un fuerte exceso de rendimiento por encima de un índice de referencia. En este caso, la estrategia tendría un alto índice de Sharpe basado en datos históricos. Sin embargo, no tiene en cuenta que tales opciones pueden ser *llamó*, lo que lleva a reducciones significativas o incluso a la desaparición de la curva de capital. Por lo tanto, como con cualquier medida del rendimiento de la estrategia comercial algorítmica, la relación de Sharpe no se puede utilizar de forma aislada.

Aunque este punto puede parecer obvio para algunos, los costos de transacción DEBEN incluirse en el cálculo del índice de Sharpe para que sea realista. Hay innumerables ejemplos de estrategias comerciales que tienen Sharpes altos y, por lo tanto, una probabilidad de gran rentabilidad, solo para reducirse a estrategias de baja rentabilidad de Sharpe bajo una vez que se han tenido en cuenta los costos realistas. Esto significa hacer uso de la *retornos netos* cuando se calcula en exceso del punto de referencia. Por lo tanto, los costos de transacción deben tenerse en cuenta *río arriba* del cálculo de la relación de Sharpe.

Una pregunta obvia que ha permanecido sin respuesta hasta el momento es "¿Cuál es un buen índice de Sharpe para una estrategia?". En realidad, esta es una pregunta bastante difícil de responder porque cada inversor tiene un perfil de riesgo diferente. La regla general es que las estrategias cuantitativas con Sharpe Ratio anualizado $S < 1$ a menudo no debe ser considerado. Sin embargo, hay excepciones a esto, particularmente en el espacio de futuros que siguen tendencias.

Los fondos cuantitativos tienden a ignorar cualquier estrategia que posea una relación de Sharpe $S < 2$. Un destacado fondo de cobertura cuantitativo con el que estoy familiarizado ni siquiera consideraría estrategias que tuvieran índices de Sharpe $S < 3$ mientras estaba en la investigación. Como comerciante minorista algorítmico, si puede lograr una relación de Sharpe fuera de muestra (¡es decir, comercio en vivo!) $S > 2$ entonces lo estás haciendo muy bien.

La relación de Sharpe a menudo aumentará con la frecuencia de negociación. Algunas estrategias de alta frecuencia tendrán índices de Sharpe altos de un solo dígito (ya veces bajos de dos dígitos), ya que pueden ser rentables casi todos los días y ciertamente todos los meses. Estas estrategias rara vez sufren de un riesgo catastrófico (en el sentido de una gran pérdida) y, por lo tanto, minimizan la volatilidad de sus rendimientos, lo que conduce a índices de Sharpe tan altos. Sin embargo, tenga en cuenta que las estrategias de alta frecuencia como estas pueden simplemente dejar de funcionar muy repentinamente, que es otro aspecto del riesgo que no se refleja completamente en el índice de Sharpe.

Consideremos ahora algunos ejemplos reales de Sharpe. Comenzaremos de manera simple, considerando una compra y retención en posición larga de una acción individual y luego consideraremos una estrategia de mercado neutral. Ambos ejemplos se han llevado a cabo con Pandas.

La primera tarea es obtener los datos y colocarlos en un objeto Pandas DataFrame. En el capítulo anterior sobre la implementación del maestro de valores con Python y MySQL, creamos un sistema para lograrlo. Alternativamente, podemos utilizar este código más simple para obtener datos de Google Finance directamente y colocarlos directamente en un DataFrame. En la parte inferior de este script, he creado una función para calcular el índice de Sharpe anualizado en función de un flujo de retornos de período de tiempo:

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

# sharpe.py

de__futuro__importarimprimir_funcion
```

```
importarfecha y hora
importarnumpy como np importar
pandas como pd importar
pandas.io.data como web


definitivamenteanualizado_nítido(devoluciones, N=252):
    """
    Calcule la relación de Sharpe anualizada de un flujo de rendimientos en función de
    una cantidad de períodos de negociación, N. El valor predeterminado de N es 252,
    que luego asume un flujo de rendimientos diarios.

    La función asume que los rendimientos son el exceso de aquellos en
    comparación con un punto de referencia.
    """
    devolvernp.sqrt(N) * devuelve.media() / devuelve.std()
```

Ahora que tenemos la capacidad de obtener datos de Google Finance y calcular directamente el índice de Sharpe anualizado, podemos probar una estrategia de comprar y mantener para dos acciones. Usaremos Google (GOOG) del 1 de enero de 2000 al 1 de enero de 2013.

Podemos crear una función de ayuda adicional que nos permita ver rápidamente comprar y mantener Sharpe en múltiples acciones durante el mismo período (codificado):

```
definitivamenteequidad_sharp (ticker):
    """
    Calcula el índice de Sharpe anualizado en función de los rendimientos
    diarios de un símbolo bursátil que aparece en Google Finance.

    Las fechas han sido codificadas aquí por razones de brevedad.
    """
    inicio = fechahora.fechahora(2000,1,1) fin =
    fechahora.fechahora(2013,1,1)

    # Obtenga los datos históricos diarios de acciones para el período de tiempo deseado
    # y agregar a un marco de datos de pandas
    pdf = web.DataReader(ticker, 'google', inicio, fin)

    # Use el método de cambio porcentual para calcular fácilmente los rendimientos
    diarios pdf['daily_ret'] = pdf['Cerrar'].pct_change()

    # Suponga una tasa libre de riesgo anual promedio durante el período del 5%
    pdf['excess_daily_ret'] = pdf['daily_ret'] - 0.05/252

    # Devuelva el índice de Sharpe anualizado basado en los rendimientos diarios en exceso
    devolveranualizado_nítido(pdf['excess_daily_ret'])
```

Para Google, la relación de Sharpe para comprar y mantener es0.703:

```
> > > equidad_sharp('GOOG')
0.70265563285799615
```

Ahora podemos probar el mismo cálculo para una estrategia de mercado neutral. El objetivo de esta estrategia es aislar completamente el desempeño de una acción en particular del mercado en general. La forma más sencilla de lograr esto es quedarse corto con una cantidad igual (en dólares) de un Fondo Cotizado en Bolsa (ETF) que está diseñado para rastrear dicho mercado. La opción más obvia para el mercado de acciones de gran capitalización de EE. UU. es el índice S&P500, que es rastreado por el SPDR ETF, con el ticker de SPY.

Para calcular el ratio de Sharpe anualizado de dicha estrategia obtendremos los precios históricos de SPY y calcularemos los rendimientos porcentuales de forma similar a las acciones anteriores, con

con la excepción de que no utilizaremos el índice de referencia libre de riesgo. calcularemos el*retornos diarios netos*lo que requiere restar la diferencia entre los rendimientos largos y cortos y luego dividir por 2, ya que ahora tenemos el doble de capital comercial. Aquí está el código de Python/pandas para llevar a cabo esto:

```
definitivamentemarket_neutral_sharpe (ticker, punto de referencia):
    """
    Calcula el índice de Sharpe anualizado de una estrategia larga/corta neutral
    del mercado que combina la posición larga del 'ticker' con la posición corta
    correspondiente del 'punto de referencia'.
    """
    inicio = fechahora.fechahora(2000, 1, 1) fin =
    fechahora.fechahora(2013, 1, 1)

    # Obtenga datos históricos para un símbolo/ticker y un ticker de referencia
    # Las fechas han sido codificadas, ¡pero puede modificarlas como mejor le parezca!tick =
    web.DataReader(ticker, 'google', inicio, final) banco = web.DataReader(punto de referencia,
    'google', inicio, final)

    # Calcule los rendimientos porcentuales en cada una de las series de tiempo
    tick['daily_ret'] = tick['Cerrar'].pct_change() banco['daily_ret'] =
    banco['Cerrar'].pct_change()

    # Crear un nuevo DataFrame para almacenar la información de la estrategia
    # Los retornos netos son (largo - corto)/2, ya que hay el doble
    # el capital comercial para esta estrategia strat =
    pd.DataFrame(index=tick.index)
    estrategia['net_ret'] = (tick['daily_ret'] - banco['daily_ret'])/2.0

    # Devuelva el índice de Sharpe anualizado para esta estrategia
    devolveranualizado_sharpe(strat['net_ret'])
```

Para Google, la relación de Sharpe para la estrategia neutral de mercado largo/corto es0.832:

```
> > > market_neutral_sharpe('GOOG', 'SPY')
0.83197496084314604
```

Ahora consideraremos brevemente otras relaciones de riesgo/beneficio.

Ratio Sortino

El índice de Sortino está motivado por el hecho de que el índice de Sharpe captura la volatilidad tanto al alza como a la baja en su denominador. Sin embargo, los inversores (y los administradores de fondos de cobertura) generalmente no se preocupan demasiado cuando tenemos una volatilidad alcista significativa. Lo que realmente interesa desde la perspectiva de la gestión de riesgos es la volatilidad a la baja y los períodos de reducción.

Por tanto, el ratio de Sortino se define como el exceso de rendimiento medio dividido por la desviación a la baja media:

$$\text{sortino} = \sqrt{\frac{\text{MI}(R_a - R_b)}{\text{Var}(R_a - R_b)_{\text{d}}}} \tag{12.3}$$

El Sortino a veces se cita en un entorno institucional, pero ciertamente no es tan frecuente como el índice de Sharpe.

Relación CALMAR

También se podría argumentar que los inversores/comerciantes están preocupados únicamente por el alcance máximo de la reducción, en lugar de la reducción promedio. Esto motiva a CALMAR (CALifornia

Informes de cuentas administradas), también conocido como índice de reducción, que proporciona una relación entre el exceso de rendimiento medio y la reducción máxima:

$$\text{CALMAR} = \frac{\text{MI}(R_a - R_b)}{\text{máx. reducción}} \tag{12.4}$$

Una vez más, el CALMAR no se usa tanto como el índice de Sharpe.

12.2.3 Análisis de reducción

En mi opinión, el concepto de reducción es el aspecto más importante de la medición del rendimiento de un sistema de comercio algorítmico. En pocas palabras, si el capital de su cuenta desaparece, ¡ninguna de las otras métricas de rendimiento importa! El análisis de reducción se refiere a la medición de las caídas en el patrimonio de la cuenta de*marcas de agua altas*. Una marca de agua alta se define como el último pico de capital de la cuenta alcanzado en la curva de capital.

En un entorno institucional, el concepto de reducción es especialmente importante ya que la mayoría de los fondos de cobertura se remuneran solo cuando el patrimonio de la cuenta crea continuamente nuevas marcas de agua. Es decir, a un administrador de fondos no se le paga una comisión de rendimiento mientras el fondo permanece "bajo el agua", es decir, el patrimonio de la cuenta se encuentra en un período de*reducción*.

A la mayoría de los inversores les preocuparía una reducción del 10 % en un fondo y probablemente recuperarían su inversión una vez que la reducción supere el 30 %. En un entorno minorista, la situación es muy diferente. Es probable que las personas puedan sufrir reducciones más profundas con la esperanza de obtener mayores rendimientos.

Reducción máxima y duración

Las dos métricas clave de reducción son la*reducción máxima*y el*duración de la reducción*. El primero describe la mayor caída porcentual desde un pico anterior hasta el mínimo actual o anterior en el patrimonio de la cuenta. A menudo se cotiza en un entorno institucional cuando se intenta comercializar un fondo. Los comerciantes minoristas también deberían prestar mucha atención a esta cifra. El segundo describe la duración real de la reducción. Esta cifra generalmente se expresa en días, pero las estrategias de mayor frecuencia pueden usar un período de tiempo más granular.

En backtests estas medidas proporcionan*alguno*idea de cómo una estrategia*puede que*realizar en el futuro. La curva de equidad general de la cuenta puede parecer bastante atractiva después de una prueba retrospectiva calculada. Sin embargo, una curva de capital ascendente puede enmascarar fácilmente lo difícil que podría haber sido experimentar los períodos anteriores de reducción.

Cuando una estrategia comienza a caer por debajo del 10 % del capital de la cuenta, o incluso por debajo del 20 %, se requiere una gran fuerza de voluntad para continuar con la estrategia, a pesar de que la estrategia puede haber pasado históricamente, al menos en las pruebas retrospectivas, por períodos similares. Este es un problema constante con el comercio algorítmico y el comercio sistemático en general. Naturalmente, motiva la necesidad de establecer límites de retiro previos y reglas específicas, como un "stop loss" en toda la cuenta que se llevará a cabo en caso de que un retiro supere estos niveles.

Curva de reducción

Si bien es importante conocer la reducción máxima y la duración de la reducción, es significativamente más instructivo ver un gráfico de serie temporal de la reducción de la estrategia durante la duración de la negociación.

La figura 12.2.3 muestra claramente que esta estrategia en particular sufrió un período de reducción relativamente sostenido que comenzó en el tercer trimestre de 2010 y finalizó en el segundo trimestre de 2011, alcanzando una reducción máxima del 14,8 %. Si bien la estrategia en sí siguió siendo significativamente rentable a largo plazo, este período en particular habría sido muy difícil de soportar. Además, este es el máximo*histórico*reducción que ha ocurrido*hasta la fecha*. La estrategia puede estar sujeta a una reducción aún mayor en el futuro. Por lo tanto, es necesario considerar las curvas de reducción, al igual que con otras medidas de rendimiento de aspecto histórico, en el contexto en el que se han generado, es decir, a través de datos históricos y no futuros.

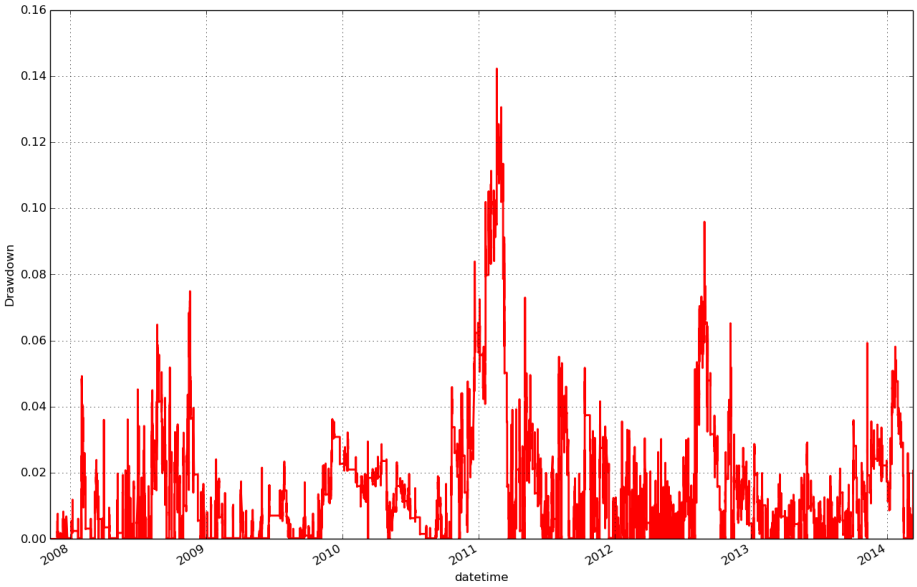


Figura 12.2: Curva típica de reducción de la estrategia intradía

En el siguiente capítulo, consideraremos el concepto de gestión cuantitativa del riesgo y describiremos técnicas que pueden ayudarnos a minimizar las reducciones y maximizar los rendimientos, manteniendo al mismo tiempo un grado razonable de riesgo.

Capítulo 13

Gestión de riesgos y dinero

Este capítulo se ocupa de la gestión del riesgo aplicada a las estrategias comerciales cuantitativas. Por lo general, esto viene en dos formas, en primer lugar, identificando y mitigando los factores internos y externos que pueden afectar el rendimiento o el funcionamiento de una estrategia comercial algorítmica y, en segundo lugar, cómo administrar de manera óptima la cartera de estrategias para maximizar la tasa de crecimiento y minimizar las reducciones de cuenta.

En la primera sección, consideraremos diferentes fuentes de riesgo (tanto intrínsecas como extrínsecas) que podrían afectar el desempeño a largo plazo de un negocio de comercio algorítmico, ya sea minorista o institucional.

En la segunda sección, analizaremos las técnicas de administración del dinero que pueden proteger nuestra cartera de la ruina y, al mismo tiempo, intentar maximizar la tasa de crecimiento a largo plazo de las acciones.

En la sección final, consideramos técnicas de gestión de riesgos a nivel institucional que se pueden aplicar fácilmente en un entorno minorista para ayudar a proteger el capital comercial.

13.1 Fuentes de riesgo

Existen numerosas fuentes de riesgo que pueden tener un impacto en el correcto funcionamiento de una estrategia comercial algorítmica. "Riesgo" generalmente se define en este contexto para significar la posibilidad de pérdidas en la cuenta. Sin embargo, voy a definirlo en un contexto mucho más amplio para referirme a cualquier factor que proporcione un grado de incertidumbre y pueda afectar el desempeño de nuestras estrategias o cartera.

Las amplias áreas de riesgo que consideraremos incluyenRiesgo de estrategia, riesgo de cartera, riesgo de mercado, riesgo de contraparteyRiesgo operacional.

13.1.1 Riesgo de estrategia

Riesgo de estrategia, o*riesgo de modelo*, engloba la clase de riesgos que surgen del diseño e implementación de una estrategia comercial basada en un modelo estadístico. Incluye todos los problemas anteriores que hemos discutido en el capítulo Backtesting exitoso, como el ajuste de curvas, el sesgo de supervivencia y el sesgo de anticipación. También incluye otros temas relacionados directamente con el análisis estadístico del modelo de estrategia.

Cualquier modelo estadístico se basa en suposiciones. Estas suposiciones a veces no se consideran en profundidad o se ignoran por completo. Esto significa que el modelo estadístico basado en estos supuestos puede ser inapropiado y, por lo tanto, conducir a una pobre capacidad predictiva o inferencial. Un ejemplo general ocurre en el marco de la regresión lineal. La regresión lineal supone que los datos de respuesta son*homocedástico*(es decir, las respuestas tienen una varianza constante en sus errores). Si este no es el caso, la regresión lineal proporciona menos precisión en las estimaciones de los parámetros.

Muchas estrategias cuantitativas utilizan*estadísticas descriptivas*de datos históricos de precios. En particular, a menudo usarán*momentos*de los datos como la media, la varianza, el sesgo y la curtosis de los rendimientos de la estrategia. Dichos modelos (incluido el Criterio de Kelly que se describe a continuación) generalmente se basan en que estos momentos son constantes en el tiempo. Debajo de*cambio de régimen de mercado*estos momentos pueden ser

alterado drásticamente y por lo tanto conduce a la degradación del modelo. Generalmente se utilizan modelos con "parámetros móviles" para mitigar este problema.

13.1.2 Riesgo de cartera

Una cartera contiene una o más estrategias. Por lo tanto, está indirectamente sujeto al riesgo de estrategia como se describe anteriormente. Además, existen riesgos específicos que ocurren a nivel de cartera. Por lo general, estos solo se consideran en un entorno institucional o en un entorno minorista de alto nivel donde el seguimiento de la cartera se lleva a cabo en un conjunto de estrategias comerciales.

Cuando la cartera en regresión vuelve a un conjunto de *factores*, como sectores industriales, clases de activos o grupos de entidades financieras, es posible determinar si la cartera está fuertemente "cargada" en un factor en particular. Por ejemplo, una cartera de acciones puede ser extremadamente pesada en acciones tecnológicas y, por lo tanto, está extremadamente expuesta a cualquier problema que afecte al sector tecnológico en su conjunto. Por lo tanto, a menudo es necesario, a nivel de cartera, anular estrategias particulares para tener en cuenta el factor de riesgo sobrecargado. Esta es a menudo una preocupación más importante en un entorno institucional donde hay más capital para asignar y la preservación del capital tiene prioridad sobre la tasa de crecimiento a largo plazo del capital. Sin embargo, ciertamente debe considerarse incluso como un inversor comercial algorítmico minorista.

Otro problema que es en gran medida un problema institucional (a menos que se negocien más activos ilíquidos) son los límites en el volumen diario de negociación. Para los comerciantes minoristas, que ejecutan estrategias en los mercados de futuros de materias primas o de gran capitalización, no existe una preocupación real con respecto al impacto en el mercado. Sin embargo, en instrumentos menos líquidos hay que tener cuidado de no negociar un porcentaje significativo del volumen diario negociado, debido al posible impacto en el mercado y, por lo tanto, a la invalidación de un modelo de negociación previamente probado (que a menudo no tiene en cuenta el impacto en el mercado). Para evitar esto es necesario calcular el *volumen diario promedio* (utilizando una media durante un período de bucle invertido, por ejemplo) y manténgase dentro de los límites porcentuales pequeños de esta cifra.

Ejecutar una cartera de estrategias trae a colación la cuestión de *correlación de estrategia*. Las correlaciones se pueden estimar a través de técnicas estadísticas como el coeficiente de correlación del momento del producto de Pearson. Sin embargo, la correlación en sí misma no es una entidad estática y también está sujeta a cambios rápidos, especialmente bajo restricciones de liquidez en todo el mercado, a menudo conocidas como *ficontagio financiero*. En general, las estrategias deben diseñarse para evitar la correlación entre sí en virtud de diferentes clases de activos o horizontes temporales. Las correlaciones móviles se pueden estimar durante un largo período de tiempo y deberían ser una parte estándar de su backtest, si considera un enfoque de cartera.

13.1.3 Riesgo de contraparte

El riesgo de contraparte generalmente se considera una forma de *riesgo crediticio*. Es el riesgo de que una contraparte no pague una obligación sobre un activo financiero del que es responsable. Hay todo un subconjunto de finanzas cuantitativas relacionado con la fijación de precios de los instrumentos de cobertura de riesgo de contraparte, pero esto no es de nuestro interés principal como comerciantes algorítmicos minoristas. Nos preocupa más el riesgo de impago de *proveedores* como un intercambio o corretaje.

Si bien esto puede parecer académico, puedo asegurarles personalmente que estos problemas son bastante reales. En un entorno institucional, experimenté de primera mano la quiebra de una agencia de corretaje en condiciones que significaban que no se devolvía todo el capital comercial. Por lo tanto, ahora factorizo tales riesgos en una cartera. El medio sugerido para mitigar este problema es utilizar múltiples corredores, aunque cuando se negocia con margen, esto puede hacer que la logística comercial sea algo complicada.

El riesgo de contraparte es generalmente más preocupante en un entorno institucional, ¡así que no me detendré demasiado en eso aquí!

13.1.4 Riesgo Operacional

El riesgo operativo abarca las fuentes de riesgo dentro de un fondo o una infraestructura operativa comercial, incluido el riesgo comercial/empresarial, el riesgo de TI y los cambios normativos o legales externos. Estos temas a menudo no se discuten en profundidad, lo que creo que es algo miope, ya que tienen el potencial de detener por completo una operación comercial de forma permanente.

El riesgo de infraestructura a menudo se asocia con los sistemas de tecnología de la información y otras infraestructuras comerciales relacionadas. Esto también incluye el riesgo de los empleados (como fraude, partida repentina). A medida que crece la escala de una infraestructura, también lo hace la probabilidad del "punto único de falla" (SPOF). Este es un componente crítico en la infraestructura comercial que, en caso de mal funcionamiento, puede provocar una interrupción catastrófica de toda la operación. En el sentido de TI, esto suele ser consecuencia de una arquitectura mal pensada. En un sentido no relacionado con TI, esto puede ser consecuencia de un organigrama mal diseñado.

Estos temas siguen siendo completamente relevantes para el comerciante minorista. A menudo, una infraestructura de TI/comercio puede terminar siendo "irregular" y "pirateada". Además, el mantenimiento deficiente de registros y otras fallas administrativas pueden generar enormes cargas fiscales potenciales. Afortunadamente, la arquitectura de "nube" brinda la capacidad de redundancia en los sistemas y la automatización de los procesos puede conducir a hábitos administrativos sólidos. Este tipo de comportamiento, que es la consideración de riesgos de fuentes distintas al mercado y la estrategia, a menudo puede marcar la diferencia entre un operador algorítmico exitoso a largo plazo y el individuo que se da por vencido debido a una falla catastrófica en la operación.

Un tema que afecta al mundo de los fondos de cobertura es el de la presentación de informes y el cumplimiento. La legislación posterior a 2008 ha supuesto una pesada carga para las empresas de gestión de activos, lo que puede tener un gran impacto en su flujo de caja y gastos operativos. Para un individuo que esté pensando en incorporar una empresa de este tipo, con el fin de expandir una estrategia o funcionar con fondos externos, es prudente mantenerse al tanto de la legislación y el entorno regulatorio, ya que es algo así como un "objetivo móvil".

13.2 Gestión del dinero

Esta sección trata uno de los conceptos más fundamentales en el comercio, tanto discrecional como algorítmico, a saber, la gestión del dinero. Un inversionista/comerciante ingenuo podría creer que el único objetivo de inversión importante es simplemente ganar tanto dinero como sea posible. Sin embargo, la realidad del comercio a largo plazo es más compleja. Dado que los participantes del mercado tienen diferentes preferencias y restricciones de riesgo, los inversores pueden tener muchos objetivos.

Muchos comerciantes minoristas consideran que la*solamente*objetivo ser un aumento continuo de la equidad de la cuenta, con poca o ninguna consideración dada al "riesgo" de una estrategia que logre este crecimiento. Los inversores minoristas más sofisticados miden los retiros de la cuenta y, según sus preferencias de riesgo, pueden hacer frente a una caída sustancial en el capital de la cuenta (digamos 50%). La razón por la que pueden hacer frente a una reducción de esta magnitud es que se dan cuenta, cuantitativamente, de que este comportamiento puede ser óptimo para la tasa de crecimiento a largo plazo de la cartera, mediante el uso de*aprovechar*.

Es probable que un inversor institucional considere el riesgo bajo una luz diferente. A menudo, los inversionistas institucionales han ordenado retiros máximos (por ejemplo, 20 %), con una consideración significativa dada a la asignación del sector y los límites de volumen diario promedio. Serían restricciones adicionales al "problema de optimización" de la asignación de capital a las estrategias. Estos factores podrían incluso ser más importantes que maximizar la tasa de crecimiento a largo plazo de la cartera.

Por lo tanto, estamos en una situación en la que podemos lograr un equilibrio entre maximizar la tasa de crecimiento a largo plazo a través del apalancamiento y minimizar nuestro "riesgo" tratando de limitar la duración y el alcance de la reducción. La principal herramienta que nos ayudará a lograr esto se llama el Criterio de Kelly.

13.2.1 Criterio de Kelly

Dentro de esta sección, el criterio de Kelly será nuestra herramienta para controlar el apalancamiento y la asignación hacia un conjunto de estrategias comerciales algorítmicas que conforman una cartera de múltiples estrategias.

vamos a definiraprovecharcomo la relación entre el tamaño de una cartera y el patrimonio real de la cuenta dentro de esa cartera. Para aclarar esto podemos usar la analogía de comprar una casa con una hipoteca. Su pago inicial (¡o "depósito" para aquellos de nosotros en el Reino Unido!) constituye el capital de su cuenta, mientras que el pago inicial más el valor de la hipoteca constituye el equivalente al tamaño de una cartera. Por lo tanto, un pago inicial de 50 000 USD en una casa de 200 000 USD (con una hipoteca de 150 000 USD) constituye un apalancamiento de (150000 + 50000)/50000 = 4.Por lo tanto, en este caso, tendría un apalancamiento de 4x en la casa. Una cartera de cuenta de margen se comporta de manera similar. Hay un componente de "efectivo" y luego se pueden pedir prestadas más acciones en el margen, para proporcionar el apalancamiento.

Antes de establecer específicamente el Criterio de Kelly, quiero esbozar las suposiciones que intervienen en su derivación, que tienen diversos grados de precisión:

- Se supondrá que cada estrategia comercial algorítmica posee un flujo de retornos que es *Normalmente distribuido* (es decir *gaussiano*). Además, cada estrategia tiene su propia *fifijo* media y desviación estándar de los rendimientos. La fórmula asume que estos valores medios y estándar *no cambies*, es decir, que son los mismos en el pasado que en el futuro. Claramente, este no es el caso con la mayoría de las estrategias, así que tenga en cuenta esta suposición.

- Los rendimientos que se consideran aquí son *rendimientos excesivos*, lo que significa que son netos de todos los costos de financiamiento, como los intereses pagados sobre el margen y los costos de transacción. Si la estrategia se lleva a cabo en un entorno institucional, esto también significa que los rendimientos son netos de las comisiones de gestión y rendimiento.

- Todos los beneficios comerciales se reinvierten y no se realizan retiros de capital. Claramente, esto no es tan aplicable en un entorno institucional donde se cobran las tarifas de gestión mencionadas anteriormente y los inversores a menudo realizan retiros.

- Todas las estrategias son estadísticamente independientes (no hay correlación entre las estrategias) y, por lo tanto, la matriz de covarianza entre los rendimientos de las estrategias es diagonal.

¡Ahora llegamos al Criterio de Kelly real! Imaginemos que tenemos un conjunto de *norte* estrategias comerciales algorítmicas y deseamos determinar cómo aplicar el apalancamiento óptimo por estrategia para maximizar la tasa de crecimiento (pero minimizar las reducciones) y cómo asignar capital entre cada estrategia. Si denotamos la asignación entre cada estrategia *i* como vector *F* de longitud *norte*, S t $F = (f_1, \ldots, f_{norte})$, luego el Criterio de Kelly para la asignación óptima a cada estrategia F_i es dado por:

$$F_i = \mu_i / \sigma^2_i \tag{13.1}$$

Dónde μ_i son los excesos de rendimiento medios y σ_i son la desviación estándar del exceso de rendimiento de una estrategia *i*. Esta fórmula describe esencialmente el apalancamiento óptimo que debe aplicarse a cada estrategia.

Mientras que el criterio de Kelly *F* nos da el apalancamiento y la asignación de estrategia óptimos, todavía necesitamos calcular nuestra tasa de crecimiento compuesta a largo plazo esperada de la cartera, que denotamos por *gramo*. La fórmula para esto está dada por:

$$gramo = r + S^2/2 \tag{13.2}$$

Dónde *r* es la tasa de interés libre de riesgo, que es la tasa a la que puede pedir prestado al corredor, y *S* es el índice de Sharpe anualizado de la estrategia. Este último se calcula a través del exceso de rendimiento medio anualizado dividido por las desviaciones estándar anualizadas del exceso de rendimiento. Consulte el capítulo anterior sobre Medición del rendimiento para obtener detalles sobre la relación de Sharpe.

Un ejemplo realista

Consideremos un ejemplo en el caso de una sola estrategia (i=1). Supongamos que compramos una acción mítica XYZ que tiene un rendimiento anual medio de *metro* =10.7% y una desviación estándar anual de σ=12.4%. Además, suponga que podemos pedir prestado a una tasa de interés libre de riesgo de r=3.0%. Esto implica que los excesos de rendimiento medios son $\mu = m - r$ =10.7 −3.0 = 7.7%. Esto nos da una relación de Sharpe de S=0.077/0.124 = 0.62.

Con esto podemos calcular el apalancamiento óptimo de Kelly a través de $F = \mu/\sigma^2$ =0.077/0.124^2 =5.01. Por lo tanto, el apalancamiento de Kelly dice que para una cartera de 100 000 USD debemos pedir prestados 401 000 USD adicionales para tener un valor de cartera total de 501 000 USD. *En la práctica, es poco probable que nuestro corretaje nos permita operar con un margen tan sustancial, por lo que sería necesario ajustar el criterio de Kelly.*

Entonces podemos usar la relación de Sharpe *S* y la tasa de interés *r* calcular *gramo*, la tasa de crecimiento compuesta esperada a largo plazo. $gramo = r + S^2/2$ = 0.03 + 0.62^2/2 = 0.22, es decir 22%. Por lo tanto, deberíamos *suponer* un rendimiento del 22% anual de esta estrategia.

Criterio de Kelly en la práctica

Es importante tener en cuenta que el Criterio de Kelly requiere un reequilibrio continuo de la asignación de capital para seguir siendo válido. Evidentemente, esto no es posible en el entorno discreto del comercio real, por lo que se debe realizar una aproximación. La "regla general" estándar aquí es actualizar la asignación de Kelly una vez al día. Además, el Criterio de Kelly en sí mismo debe recalcularse periódicamente, utilizando una media móvil y una desviación estándar con una ventana retrospectiva. Una vez más, para una estrategia que opere aproximadamente una vez al día, esta retrospectiva debe establecerse en el orden de 3 a 6 meses de rendimiento diario.

Aquí hay un ejemplo de reequilibrio de una cartera según el Criterio de Kelly, que puede conducir a un comportamiento contrario a la intuición. Supongamos que tenemos la estrategia descrita anteriormente. Hemos utilizado el criterio de Kelly para pedir prestado efectivo para dimensionar nuestra cartera a 501 000 USD. Supongamos que obtenemos un rendimiento saludable del 5 % al día siguiente, lo que aumenta el tamaño de nuestra cuenta a 526 050 USD. El Criterio de Kelly nos dice que debemos pedir prestado*más*mantener el mismo factor de apalancamiento de 5.01. En particular, el patrimonio de nuestra cuenta es de 126 050 USD en una cartera de 526 050, lo que significa que el factor de apalancamiento actual es de 4,17. Para aumentarlo a 5,01, necesitamos pedir prestados 105 460 USD adicionales para aumentar el tamaño de nuestra cuenta a 631 510,5 USD (esto es5.01×126050).

Ahora considere que al día siguiente perdemos un 10% en nuestra cartera (¡ay!). Esto significa que el tamaño total de la cartera es ahora de 568 359,45 USD (631510.5×0.9).Nuestra cuenta total*equidad* ahora es 62,898.95 USD ($126050 - 631510.45 \times 0.1$).Esto significa que nuestro factor de apalancamiento actual es 568359. $45 / 62898.95 = 9.03$.Por lo tanto, necesitamos reducir nuestra cuenta en*venta*253.235,71 USD de acciones para reducir el valor total de nuestra cartera a 315.123,73 USD, de modo que tengamos un apalancamiento de 5,01 nuevamente ($315123.73 / 62898.95 = 5.01$).

Por lo tanto, tenemos*comprado*en una ganancia y*vendido*en una pérdida. Este proceso de vender a pérdida puede ser extremadamente difícil emocionalmente, pero matemáticamente es lo "correcto" que se puede hacer, ¡suponiendo que se hayan cumplido las suposiciones de Kelly! Es el enfoque a seguir para maximizar la tasa de crecimiento compuesto a largo plazo.

Es posible que haya notado que los valores absolutos del dinero reasignado entre días fueron bastante severos. Esta es una consecuencia tanto de la naturaleza artificial del ejemplo como de la amplia influencia empleada. La pérdida del 10% en un día no es particularmente común en el comercio algorítmico de alta frecuencia, pero sirve para mostrar cuán extenso puede ser el apalancamiento en términos absolutos.

Dado que la estimación de las medias y las desviaciones estándar siempre están sujetas a incertidumbre, en la práctica, muchos comerciantes tienden a utilizar un régimen de apalancamiento más conservador, como el Criterio de Kelly dividido por dos, conocido cariñosamente como "medio-Kelly". El Criterio de Kelly realmente debe considerarse como un límite superior de apalancamiento para usar, en lugar de una especificación directa. Si no se tiene en cuenta este consejo, el uso del valor de Kelly directo puede conducir a la ruina (es decir, el patrimonio de la cuenta puede llegar a cero) debido a la naturaleza no gaussiana de los rendimientos de la estrategia.

¿Deberías usar el criterio de Kelly?

Cada comerciante algorítmico es diferente y lo mismo ocurre con las preferencias de riesgo. Al elegir emplear una estrategia de apalancamiento (de la cual el Criterio de Kelly es un ejemplo), debe considerar los mandatos de riesgo bajo los cuales necesita trabajar. En un entorno minorista, puede establecer sus propios límites máximos de extracción y, por lo tanto, puede aumentar su apalancamiento. En un entorno institucional, deberá considerar el riesgo desde una perspectiva muy diferente y el factor de apalancamiento será un componente de un marco mucho más amplio, generalmente bajo muchas otras restricciones.

13.3 Gestión de riesgos

13.3.1 Valor en riesgo

Estimar el riesgo de pérdida de una estrategia comercial algorítmica, o una cartera de estrategias, es de suma importancia para el crecimiento del capital a largo plazo. Se han desarrollado muchas técnicas para la gestión de riesgos para su uso en entornos institucionales. Una técnica en particular, conocida comoValor en riesgoo VaR,será el tema de esta sección.

Aplicaremos el concepto de VaR a una única estrategia o a un conjunto de estrategias para ayudarnos a cuantificar el riesgo en nuestra cartera de negociación. La definición de VaR es la siguiente:

VaR proporciona una *estimar*, bajo un grado de confianza dado, del tamaño de una pérdida de una cartera durante un período de tiempo dado.

En este caso, "cartera" puede referirse a una sola estrategia, un grupo de estrategias, un libro de comerciante, una mesa de utilería, un fondo de cobertura o un banco de inversión completo. El "grado de confianza dado" será un valor de, digamos, 95% o 99%. El "período de tiempo dado" se elegirá para reflejar uno que conduzca a un mínimo *impacto en el mercado* si se liquidara una cartera.

Por ejemplo, un VaR igual a 500 000 USD con un nivel de confianza del 95 % para un período de tiempo de un día simplemente indicaría que existe una probabilidad del 95 % de perder no más de 500 000 USD en el día siguiente. Matemáticamente esto se expresa como:

$$PAGS(L \leq -5.0 \times 10_5) = 0.05 \tag{13.3}$$

O, más generalmente, por pérdida L exceder un valor VAR con un nivel de confianza C tenemos:

$$PAGS(L \leq -VAR) = 1 - C \tag{13.4}$$

El cálculo "estándar" del VaR hace las siguientes suposiciones:

- Condiciones estándar del mercado -Se supone que el VaR no considera eventos extremos o "riesgo de cola", sino que se supone que proporciona la expectativa de una pérdida en la operación normal "día a día".
- Volatilidades y Correlaciones -El VaR requiere las volatilidades de los activos considerados, así como sus respectivas correlaciones. Estas dos cantidades son difíciles de estimar y están sujetas a cambios continuos.
- Normalidad de Devoluciones -VaR, en su forma estándar, asume que los rendimientos del activo o cartera son *Normalmente distribuido*. Esto conduce a un cálculo analítico más sencillo, pero es bastante poco realista para la mayoría de los activos.

13.4 Ventajas y desventajas

El VaR es omnipresente en la industria financiera, por lo que debe estar familiarizado con las ventajas y desventajas de la técnica. Algunas de las ventajas del VaR son las siguientes:

- El VaR es muy sencillo de calcular para activos individuales, estrategias algorítmicas, carteras cuantitativas, fondos de cobertura o incluso mesas de apoyo bancarias.
- El periodo de tiempo asociado al VaR puede modificarse para múltiples estrategias de negociación que tienen diferentes horizontes de tiempo.
- Se pueden asociar diferentes valores de VaR con diferentes formas de riesgo, por ejemplo, desglosados por clase de activo o tipo de instrumento. Esto facilita la interpretación de dónde puede agruparse la mayor parte del riesgo de la cartera, por ejemplo.
- Las estrategias individuales pueden estar limitadas al igual que las carteras enteras en función de su VaR individual.
- El VaR es sencillo de interpretar por (potencialmente) inversores externos y gestores de fondos no técnicos.

Sin embargo, el VaR no está exento de desventajas:

- VaR no discute la magnitud de la pérdida esperada más allá del valor de VaR, es decir, nos dirá que es probable que veamos una pérdida *excesivo* un valor, pero no cuánto lo excede.

- No tiene en cuenta los eventos extremos, sino solo las condiciones típicas del mercado.
- Dado que utiliza datos históricos (es retrospectivo), no tendrá en cuenta los futuros cambios en el régimen del mercado que pueden modificar las volatilidades y las correlaciones de los activos.

El VaR no debe utilizarse de forma aislada. Siempre debe utilizarse con un conjunto de técnicas de gestión de riesgos, como la diversificación, la asignación óptima de la cartera y el uso prudente del apalancamiento.

Métodos de cálculo

Hasta el momento no hemos discutido el cálculo real del VaR, ya sea en el caso general o en un ejemplo de negociación concreto. Hay tres técnicas que serán de nuestro interés. El primero es el método de varianza-covarianza (usando supuestos de normalidad), el segundo es un método de Monte Carlo (basado en una distribución subyacente, potencialmente no normal) y el tercero se conoce como arranque histórico, que hace uso de información histórica de retornos. para los activos bajo consideración.

En esta sección nos concentraremos en el método de varianza-covarianza.

Método de varianza-covarianza

Considere una cartera de $PAGS$ dólares, con un nivel de confianza C. Estamos considerando rendimientos diarios, con la desviación estándar histórica del activo (o estrategia) σ y media μ. Entonces el *diariamente* El VaR, bajo el método de varianza-covarianza para un solo activo (o estrategia) se calcula como:

$$PAG \text{ -}(PAGS(\alpha(1-C) + 1)) \tag{13.5}$$

Dónde α es la inversa de la función de distribución acumulativa de una distribución normal con media μ y desviación estándar σ.

Podemos usar las bibliotecas SciPy y pandas para calcular estos valores. si establecemos $PAGS = 10_6$ y $C = 0.99$, podemos usar el método SciPy ppf para generar los valores para la función de distribución acumulativa inversa a una distribución normal con μ y σ obtenido de algunos datos financieros reales, en este caso los rendimientos diarios históricos de CitiGroup (podríamos sustituir fácilmente los rendimientos de una estrategia algorítmica aquí):

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

#var.py

de__futuro__importarimprimir_funcion

importarfecha y hora

importarnumpy como np importar
pandas.io.data como web de
scipy.statsimportarnorma


definitivamentevar_cov_var(P, c, mu, sigma):
    """
    Cálculo de Varianza-Covarianza del Valor en Riesgo diario utilizando el
    nivel de confianza c, con media de retornos mu y desviación estándar
    de retornos sigma, sobre una cartera de valor P.

    """
    alfa = norma.ppf(1-c, mu, sigma) devolverP
    - P*(alfa + 1)
```

```
si__nombre__ == "__principal__":
    inicio = fechahora.fechahora(2010, 1, 1) fin =
    fechahora.fechahora(2014, 1, 1)

    citi = web.DataReader("C", 'yahoo', inicio, final) citi["rets"] =
    citi["Adj Close"].pct_change()

    P = 1e6         # 1,000,000 dolares
    c = 0,99        # Intervalo de confianza del 99%
    mu = np.mean(citi["rets"]) sigma =
    np.std(citi["rets"])

    var = var_cov_var(P, c, mu, sigma) impresión
    ("Valor en riesgo: $%0.2f" % var)
```

El valor calculado del VaR viene dado por:

```
Valor en Riesgo: $56503.12
```

VaR es una técnica extremadamente útil y omnipresente en todas las áreas de la gestión financiera, pero no está exenta de fallas. David Einhorn, el renombrado administrador de fondos de cobertura, describió el VaR como "una bolsa de aire que funciona todo el tiempo, excepto cuando tienes un accidente automovilístico". De hecho, siempre debe usar el VaR como un aumento de su superposición de gestión de riesgos, ¡no como un indicador único!

Parte VI

Comercio automatizado

capitulo 14

Implementación del motor de negociación basado en eventos

Este capítulo proporciona una implementación para un sistema de backtest basado en eventos totalmente autónomo escrito en Python. En particular, este capítulo se ha escrito para ampliar los detalles que generalmente se omiten en otros textos y documentos de comercio algorítmico. El siguiente código le permitirá simular estrategias de alta frecuencia (minuto a segundo) en los dominios de pronóstico, impulso y reversión a la media en los mercados de acciones, divisas y futuros.

Sin embargo, con los detalles extensos viene la complejidad. El sistema de backtesting proporcionado aquí requiere muchos componentes, cada uno de los cuales son entidades integrales en sí mismos. Por lo tanto, el primer paso es describir qué es el software basado en eventos y luego describir los componentes del backtester y cómo encaja todo el sistema.

14.1 Software controlado por eventos

Antes de profundizar en el desarrollo de un backtester de este tipo, debemos comprender el concepto de sistemas controlados por eventos. Los videojuegos proporcionan un caso de uso natural para el software basado en eventos y proporcionan un ejemplo sencillo para explorar. Un videojuego tiene múltiples componentes que interactúan entre sí en un entorno en tiempo real a altas velocidades de fotogramas. Esto se maneja ejecutando todo el conjunto de cálculos dentro de un bucle "infinito" conocido como bucle de eventos o bucle de juego.

En cada tic del bucle del juego, se llama a una función para recibir el último evento, que habrá sido generado por alguna acción anterior correspondiente dentro del juego. Dependiendo de la naturaleza del evento, que podría incluir presionar una tecla o hacer clic con el mouse, se realiza alguna acción posterior que finalizará el bucle o generará algunos eventos adicionales. El proceso luego continuará.

Aquí hay un ejemplo de pseudocódigo:

```
tiempoVerdadero:#Ejecutar el ciclo para siempre
    nuevo_evento = obtener_nuevo_evento()    # Obtener el último evento

    # Según el tipo de evento, realiza una acción si
    new_event.type == "CLICK_IZQUIERDO_MOUSE":
        menú abierto()
    elifnuevo_evento.tipo == "ESCAPE_TECLA_PRESIONAR":
        salir del juego()
    elifnew_event.type == "UP_KEY_PRESS":
        mover_jugador_norte()
    # . . . y muchos eventos mas

    redraw_screen()        # Actualice la pantalla para proporcionar animación
    garrapata(50)    # Espere 50 milisegundos
```

El código busca continuamente nuevos eventos y luego realiza acciones basadas en estos eventos. En particular, permite la ilusión de un manejo de respuestas en tiempo real porque el código se repite continuamente y se verifican los eventos. Como quedará claro, esto es precisamente lo que necesitamos para llevar a cabo una simulación comercial de alta frecuencia.

14.1.1 ¿Por qué un backtester basado en eventos?

Los sistemas basados en eventos ofrecen muchas ventajas sobre un enfoque vectorizado:

- Reutilización de código -Un backtester basado en eventos, por diseño, se puede usar tanto para backtesting histórico como para operaciones en vivo con un cambio mínimo de componentes. Esto no es cierto para los backtesters vectorizados, donde todos los datos deben estar disponibles a la vez para llevar a cabo el análisis estadístico.
- Sesgo anticipado -Con un backtester basado en eventos, no hay sesgo de anticipación, ya que la recepción de datos de mercado se trata como un "evento" sobre el que se debe actuar. Por lo tanto, es posible "alimentar por goteo" un backtester basado en eventos con datos de mercado, replicando cómo se comportaría un sistema de gestión de pedidos y cartera.
- realismo -Los backtesters basados en eventos permiten una personalización significativa sobre cómo se ejecutan los pedidos y se incurre en los costos de transacción. Es sencillo manejar órdenes básicas de mercado y límite, así como mercado al abrir (MOO) y mercado al cerrar (MOC), ya que se puede construir un controlador de intercambio personalizado.

Aunque los sistemas controlados por eventos vienen con muchos beneficios, tienen dos desventajas principales sobre los sistemas vectorizados más simples. En primer lugar, son significativamente más complejos de implementar y probar. Hay más "partes móviles" que conducen a una mayor posibilidad de introducir errores. Para mitigar esta metodología adecuada de prueba de software, como el desarrollo basado en pruebas, se puede emplear.

En segundo lugar, son más lentos de ejecutar en comparación con un sistema vectorizado. Las operaciones vectorizadas óptimas no se pueden utilizar al realizar cálculos matemáticos.

14.2 Objetos componentes

Para aplicar un enfoque basado en eventos a un sistema de backtesting, es necesario definir nuestros componentes (u objetos) que manejarán tareas específicas:

- Evento -El Evento es la unidad de clase fundamental del sistema dirigido por eventos. Contiene un tipo (como "MERCADO", "SEÑAL", "PEDIDO" o "LLENAR") que determina cómo se manejará dentro del bucle de eventos.
- Cola de eventos -La cola de eventos es un objeto de cola de Python en memoria que almacena todos los objetos de subclase de eventos que genera el resto del software.
- manejador de datos -DataHandler es una clase base abstracta (ABC) que presenta una interfaz para manejar datos de mercado históricos o en vivo. Esto proporciona una flexibilidad significativa ya que los módulos de Estrategia y Portafolio pueden reutilizarse entre ambos enfoques. El DataHandler genera un nuevo MarketEvent en cada latido del sistema (ver más abajo).
- Estrategia -La estrategia también es un ABC que presenta una interfaz para tomar datos de mercado y generar SignalEvents correspondientes, que finalmente son utilizados por el objeto Portafolio. Un SignalEvent contiene un símbolo de teletipo, una dirección (LARGA o CORTA) y una marca de tiempo.
- Portafolio -Esta es una jerarquía de clases que maneja la gestión de pedidos asociada con posiciones actuales y posteriores para una estrategia. También lleva a cabo la gestión de riesgos en toda la cartera, incluida la exposición sectorial y el tamaño de la posición. En una implementación más sofisticada, esto podría delegarse a una clase RiskManagement. El Portafolio toma SignalEvents de la Cola y genera OrderEvents que se agregan a la Cola.

- Manejador de ejecución -El ExecutionHandler simula una conexión a un corretaje. El trabajo del controlador es tomar OrderEvents de la Cola y ejecutarlos, ya sea a través de un enfoque simulado o una conexión real a un corredor de hígado. Una vez que se ejecutan las órdenes, el manejador crea FillEvents, que describen lo que realmente se operó, incluidas las tarifas, la comisión y el deslizamiento (si se modelaron).
- Prueba retrospectiva -Todos estos componentes están envueltos en un bucle de eventos que maneja correctamente todos los tipos de eventos, enrutándolos al componente apropiado.

A pesar de la cantidad de componentes, este es un modelo bastante básico de un motor comercial. Existe un margen significativo para la expansión, particularmente en lo que respecta a cómo se utiliza la Cartera. Además, los diferentes modelos de costos de transacción también podrían abstraerse en su propia jerarquía de clases.

14.2.1 Eventos

El primer componente que se discutirá es la jerarquía de clases de eventos. En esta infraestructura hay cuatro tipos de eventos que permiten la comunicación entre los componentes anteriores a través de una cola de eventos. Son MarketEvent, SignalEvent, OrderEvent y FillEvent.

Evento

La clase padre en la jerarquía se llama Evento. Es una clase base y no proporciona ninguna funcionalidad o interfaz específica. Dado que en muchas implementaciones los objetos Event probablemente desarrollarán una mayor complejidad, por lo tanto, se está "preparando para el futuro" mediante la creación de una jerarquía de clases.

```
#!/usr/bin/python
# - * - codificación: utf-8 -*-

# evento.py

de__futuro__importarimprimir_funcion


claseEvento (objeto):
    """
    El evento es una clase base que proporciona una interfaz para todos los
    eventos posteriores (heredados), que desencadenarán más eventos en la
    infraestructura comercial.
    """
    pasar
```

MercadoEvento

Los MarketEvents se activan cuando el ciclo while externo del sistema de backtesting comienza un nuevo "latido". Ocurre cuando el objeto DataHandler recibe una nueva actualización de datos de mercado para cualquier símbolo que se esté rastreando actualmente. Se utiliza para activar el objeto Estrategia generando nuevas señales comerciales. El objeto de evento simplemente contiene una identificación de que es un evento de mercado, sin otra estructura.

```
# evento.py

claseEvento de mercado (evento):
    """
    Maneja el evento de recibir una nueva actualización del mercado con las barras
    correspondientes.
    """


    definitivamente__en si mismo):
```