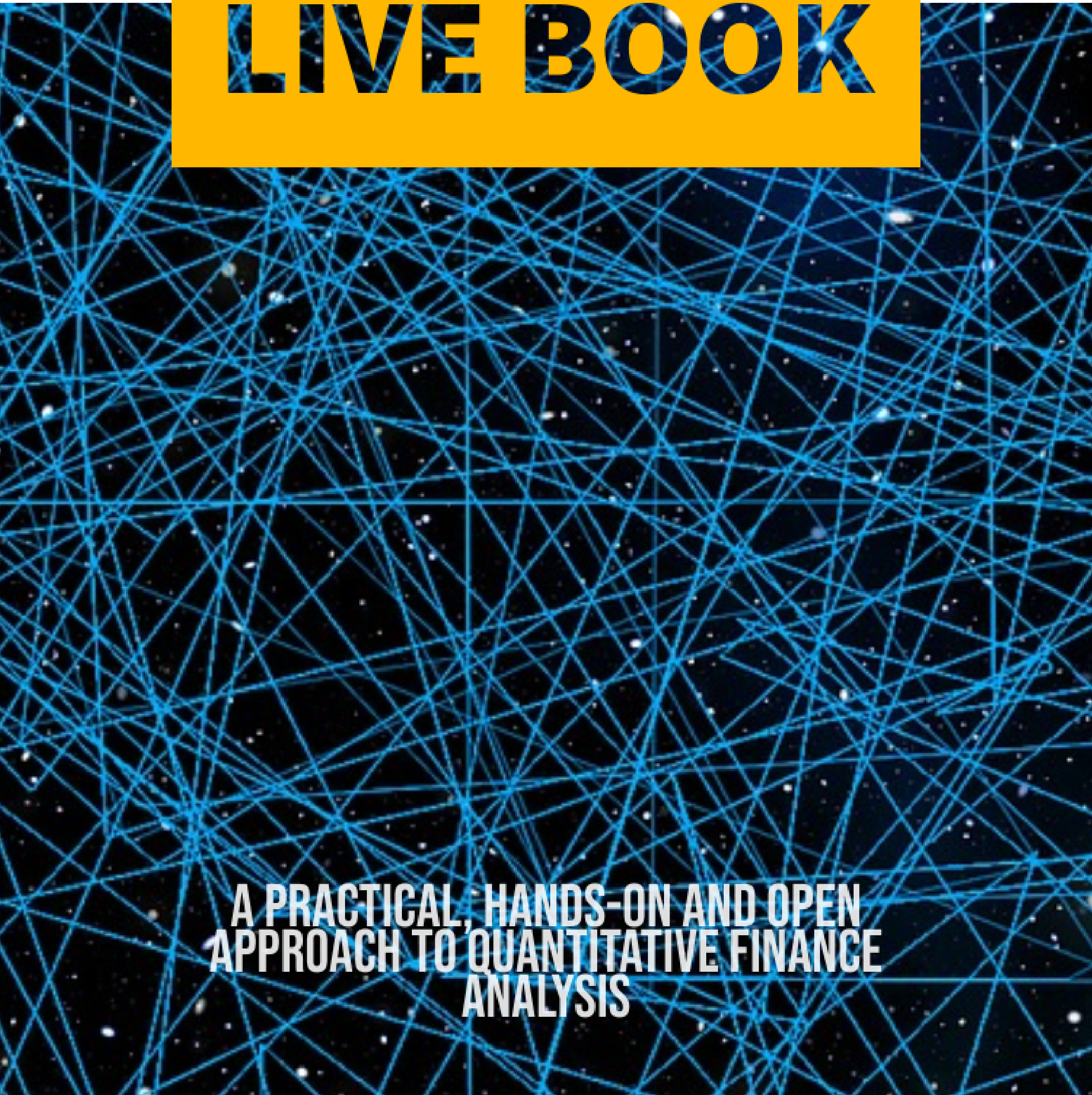
Thársis T. P. Souza

OPEN QUANT LIVE BOOK



A PRACTICAL, HANDS-ON AND OPEN APPROACH TO QUANTITATIVE FINANCE ANALYSIS

The Open Quant Live Book

Thársis T. P. Souza

2019-01-08

Contents

Preface

Description

The book aims to be an Open Source introductory reference of the most important aspects of financial data analysis, algo trading, portfolio selection, econophysics and machine learning in finance with an emphasis in reproducibility and openness not to be found in most other typical Wall Street-like references.

The Book is Open[1] and we are looking for co-authors. Feel free to reach out[2] or simply create a pull request with your contribution on our Github project[3].

Working Contents

1. The Basics

- I/O
- Stylized Facts
- Correlation & Causation

2. Algo Trading

- Investment Process

[1] https://github.com/souzatharsis/open-quant-live-book
[2] http://www.souzatharsis.com/
[3] https://github.com/souzatharsis/open-quant-live-book

- Backtesting
- Factor Investing
- Limit Order

3. Portfolio Optimization

- Modern Portfolio Theory
- Measuring Risk
- Linear Programming

4. Machine Learning

- Intro
- Agent-Based Models
- Binary Classifiers
- AutoML
- Hierarchical Risk Parity

5. Econophysics

- Entropy, Efficiency and Coupling
- Transfer Entropy, Information Transfer and Causality
- Financial Networks

6. Alternative Data

Book's information

First published at: openquant.netlify.com[4].

[4] https://openquant.netlify.com/

[5] https://creativecommons.org/licenses/by-nc-sa/4.0/

Part I

The Basics

Chapter 1

I/O

In this Chapter, we will introduce basic functions to read text and excel files as well as large files. We will also show how to obtain free financial and economic data from sources such as Quandl, IEX and Alpha Vantage.

1.1 Reading and Writing

1.1.1 Text Files

The most basic and commonly used option to import data from text files in R is the use of the function `read.table` from the **r-base**. We can use this function to read text files with extensions such as `.txt` and `.csv`.

```
dat.table <- read.table(file = "<name of your file>.txt")
dat.csv <- read.csv(file = "<name of your file>.csv")
```

The package **readr** provides functions for reading text data into R that are much faster that the functions from the **r-base**. The `read_table` function from the package **readr** provides a near-replacement for the `read.table` function.

```
library(readr)
dat.table <- readr::read_table2(file = "<name of your file>.txt")
dat.csv <- readr::read_csv(file = "<name of your file>.csv")
```

Another option to save data is to write it in `rds` format. Data stored in `rds` format has the advantage to keep the original data struture and type of the object saved. Also, `.rds` files are compressed and consume less space than files saved in `.csv` format. A data.frame object can be saved in `rds` format and then loaded back as follows:

```
write_rds(dat.frame, path = "<name of your file>.rds")
dat.frame <- read_rds(path = "<name of your file>.rds")
```

1.1.2 Excel Files

The package `readxl` has an ease to use interface to functions that load excel documents in R. The functions `read_xls` and `read_xlsx` can be used to read excel files as follows:

```
library(readxl)
readxl::read_xls(path = "<name of your file>.xls")
readxl::read_xlsx(path = "<name of your file>.xlsx")
```

The function `read_excel()` automatically detects the extension of the input file as follows:

```
readxl::read_excel("<name and extension of your file>", sheet = "<sh
```

In the `read_excel` function, the `sheet` argument can receive either the target sheet name or index number, where sheet indexing starts at 1.

The `readxl` has been oberving increased use compared to other comparable packages such as **gdata** and the **xlsx** due to its relative ease of use and performance. Also, the `readxl` do not have depency with external code libraries while the packages **gdata** and **xlsx** depend on `ActiveState PERL` and the `Java JDK`, respectively.

1.1.3 Large Files

Fast data manipulation in a short and flexible syntax.

1.2 Data Sources

In this section, we will show how to obtain financial and economic data from public sources.

1.2.1 Alpha Vantage

Alpha Vantage offers free access to pricing data including:

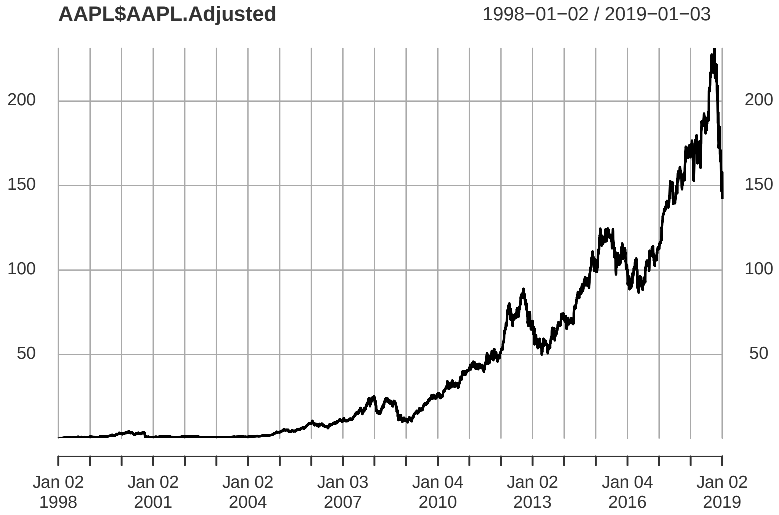
- Stock Time Series Data;
- Physical and Digital/Crypto Currencies (e.g., Bitcoin);
- Technical Indicators and
- Sector Performances.

The data are available in JSON and CSV formats via REST APIs. The **quantmod** and the **alphavantager** R packages offer a lightweight R interface to the Alpha Vantage API. Daily stock prices can be obtained with the `quantmod::getSymbols` function as follows:

```r
getSymbols(Symbols = "AAPL", src = "av", output.size = "full",
  adjusted = TRUE, api.key = "your API key")
```

The output data is stored in an object with the same name as the corresponding symbol, in this example `AAPL`. The output data looks like the following

AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AA
13.6	16.2	13.5	16.2	6411700	
16.5	16.6	15.2	15.9	5820300	
15.9	20.0	14.8	18.9	16182800	
18.8	19.0	17.3	17.5	9300200	
17.4	18.6	16.9	18.2	6910900	
18.1	19.4	17.5	18.2	7915600	



We called the `quantmod::getSymbols` function with the following arguments:

- `Symbols='AAPL'` defines a character vector specifying the names of each symbol to be loaded, here specified by the symbol of the company Apple Inc.;
- `src="av"` specifies the sourcing method, here defined with the value corresponding to Alpha Vantage;
- `output.size="full"`specified length of the time series returned. The strings `compact` and `full` are accepted with the following specifications: `compact` returns only the latest 100 data points;

  `full` returns the full-length time series of up to 20 years of historical data;
- `adjusted=TRUE` defines a boolean variable to include a column of closing prices adjusted for dividends and splits;
- `api.key` specifies your Alpha Vantage API key.

1.2.2 IEX

The IEX Group operates the Investors Exchange (IEX), a stock exchange for U.S. equities that is built for investors and companies. IEX offers U.S. reference and market data including end-of-day and intraday pricing data. IEX offers an API with “a set of services designed for developers and engineers. It can be used to build high-quality apps and services”. Data sourced from the IEX API is freely available for commercial subject to conditions[1] and the use of their API is subject to additional terms of use[2].

IEX lists the following github project as an unofficial API for R: https://github.com/imanuelcostigan/iex. We will provide examples on how to obtain intraday pricing data using this package. First, we will use the **devtools** to install the package directly from its github repository as follows:

```r
library(devtools)
install_github("imanuelcostigan/iex")
```

The **iex** package provides 4 set of functions as follows:

- `last`: Provides IEX near real time last sale price, size and time. Last is ideal for developers that need a lightweight stock quote. IEX API real time API documentation[3].
- `market`: Provides exchange trade volume data in near real time. IEX market API documentation[4].

[1] https://iextrading.com/api-exhibit-a/

[2] https://iextrading.com/api-terms/

[3] https://iextrading.com/developer/docs/#last

[4] https://iextrading.com/developer/#market-market

- `stats`: A set of functions that return trading statistics. IEX stats API documentation[5].
- `tops`: Provides IEX's aggregated bid and offer position in near real time for all securities on IEX's displayed limit order book. IEX API TOPS documentation[6].

For instance, the `last` function has the following arguments:

- `symbols`: A vector of tickers (case insensitive). Special characters will be escaped. A list of eligible symbols is published daily[7] by the IEX. When set to `NULL` (default) returns values for all symbols.
- `fields`: A vector of fields names to return (case sensitive). When set to `NULL` (default) returns values for all fields.
- `version`: The API version number, which is used to define the API URL.

We can obtain intraday stock price data with the `last` function as follows:

```
dat <- iex::last(symbols = c("AAPL"), fields = c("symbol",
  "price", "size"))
```

The function returns an S3 object of class `iex_api` which has three accessible fields: `path` , `response` and `content`.

- The `path` contains the corresponding IEX API path:

```
dat$path
```

```
## [1] "tops/last"
```

- The `response` contains the unparsed IEX API response:

```
dat$response
```

```
## Response [https://api.iextrading.com/1.0/tops/last?symbols=AAPL&f
##   Date: 2019-01-07 03:12
##   Status: 200
```

[5] https://iextrading.com/developer/#stats
[6] https://iextrading.com/developer/#tops-tops
[7] https://iextrading.com/trading/eligible-symbols/

```
##   Content-Type: application/json; charset=utf-8
##   Size: 45 B
```

- The `content` contains the parsed content from the API's response:

```
dat$content
```

```
## [[1]]
## [[1]]$symbol
## [1] "AAPL"
##
## [[1]]$price
## [1] 142
##
## [[1]]$size
## [1] 100
```

According to the developer, this package causes R to pause 0.2 seconds after executing an API call to avoid the user being throttled by the IEX API (which enforces a 5 request per second limit). Documentation about the other set of functions can be obtained at https://github.com/imanuelcostigan/iex/tree/master/man.

1.2.3 Quandl

1.3 Conclusion

- We showed how to load and import data from both local files and external sources.
- We provided examples on how to read tabular data and how to handle large files.
- We showed how to obtain financial and economic data from freely available sources.

1.3.1 Key Packages Used

- Importing Flat Files
  - **readr** and **data.table**
- Obtaining Financial Data
  - **quantmod**, **imanuelcostigan/iex** and **quandl**

1.3.2 Further Reading

To further learn how to use R to load, transform, visualize and model data see (Wickham and Grolemund, 2017). Additional relevant R packages include:

- dplyr: Fast data frames manipulation and database query.
- reshape2: Flexibly rearrange, reshape and aggregate data.
- readr: A fast and friendly way to read tabular data into R.
- tidyr: Easily tidy data with spread and gather functions.
- rlist: A toolbox for non-tabular data manipulation with lists.
- jsonlite: A robust and quick way to parse JSON files in R.
- ff: Data structures designed to store large datasets.
- lubridate: A set of functions to work with dates and times.

Chapter 2

Stylized Facts

2.1 Introduction

2.2 Distribution of Returns

2.2.1 Fat Tails

2.2.2 Skewness

2.3 Volatility

2.3.1 Time-invariance

2.3.2 Volatility Clustering

2.3.3 Correlation with Trading Volume

2.4 Correlation

2.4.1 Time-invariance

2.4.2 Auto-correlation

Chapter 3

Correlation & Causation

3.1 Introduction

3.2 A First Definition of Causality

We quantify causality by using the notion of the causal relation introduced by Granger (Wiener, 1956; Granger, 1969) where a signal X is said to Granger-cause Y if the future realizations of Y can be better explained using the past information from X and Y rather than Y alone.

The most common definitions of Granger-causality rely on the prediction of a future value of the variable Y by using the past values of X and Y itself. In that form, X is said to *G-cause* Y if the use of X improves the prediction of Y.

Let X_t be a random variable associated at time t while X^t represents the collection of random variables up to time t. We consider X_t, Y_t and Z_t to be three stochastic processes. Let $\hat{Y}_{t+1}$ be a predictor for the value of the variable Y at time $t + 1$.

We compare the expected value of a loss function $g(e)$ with the error

$e = \hat{Y}_{t+1} - Y_{t+1}$ of two models:

1. The expected value of the prediction error given only Y^t

$$\mathcal{R}(Y^{t+1} \,|\, Y^t, Z^t) = \mathbb{E}[g(Y_{t+1} - f_1(X^t, Z^t))] \tag{3.1}$$

2. The expected value of the prediction error given Y^t and X^t

$$\mathcal{R}(Y^{t+1} \,|\, X^t, Y^t, Z^t) = \mathbb{E}[g(Y_{t+1} - f_2(X^t, Y^t, Z^t))]. \tag{3.2}$$

In both models, the functions $f_1(.)$ and $f_2(.)$ are chosen to minimize the expected value of the loss function. In most cases, these functions are retrieved with linear and, possibly, with nonlinear regressions. Typical forms for $g(.)$ are the $l1$- or $l2$-norms.

We can now provide our first definition of statistical causality under the Granger causal notion as follows:
Definition 3.1. X does not Granger-cause Y relative to side information Z if and only if $\mathcal{R}(Y_{t+1} \mid X^t, Y^t, Z^t) = \mathcal{R}(Y_{t+1} \mid Y^t, Z^t)$.

A more general definition than Def. 3.1 that does not depend on assuming prediction functions can be formulated by considering conditional probabilities. A probabilistic definition of G-causality assumes that Y_{t+1} and X^t are independent given the past information (X^t, Y^t) if and only if $p(Y_{t+1} \,|\, X^t, Y^t, Z^t) = p(Y_{t+1} \,|\, Y^t, Z^t)$, where $p(. \,|\, .)$ represents the conditional probability distribution. In other words, omitting past information from X does not change the probability distribution of Y. This leads to our second definition of statistical causality as follows:
Definition 3.2. X does not Granger-cause Y relative to side information Z if and only if $Y_{t+1} \perp\!\!\!\perp X^t \mid Y^t, Z^t$.

Def. 3.2 does not assume any functional form in the coupling between X and Y. Nevertheless, it requires a method to assess their conditional dependency.

In the next Section, we define a parametric linear specification of G-causality based on Def. 3.1. Later in the book, in the Section 6.2, when we cover Econophysics techniques, we will present a nonlinear specification for G-causality based on Def. 3.2.

3.3 Quantifying Granger-Causality

We will take the following procedure to quantify Granger-causality according to Def. 3.1:

1. Specify two predictive models:

- The first considers Y^t to predict Y^{t+1} (Model $\mathcal{M}$);
- The second considers Y^t and X^t to predict Y^{t+1} (Model $\mathcal{M}^*$);

2. Test for model misspecification;
3. Test the hypothesis that the expected value of the prediction error of the Models $\mathcal{M}$ and $\mathcal{M}^*$ are statistically the same;
4. Apply correction for multiple hypothesis testing.

If the null hypothesis from 3. is rejected then there is evidence that X Granger-causes Y under Def. 3.1.

3.3.1 Model Specification

Standard Granger-causality tests assume a linear relationship among the causes and effects and are implemented by fitting autoregressive models (Wiener, 1956; Granger, 1969).

Consider the linear vector-autoregressive (VAR) equations:

$$Y(t) = \alpha + \sum_{\Delta t=1}^{k} \beta_{\Delta t} Y(t - \Delta t) + \epsilon_t, \tag{3.3}$$

$$Y(t) = \hat{\alpha} + \sum_{\Delta t=1}^{k} \hat{\beta}_{\Delta t} Y(t - \Delta t) + \sum_{\Delta t=1}^{k} \hat{\gamma}_{\Delta t} X(t - \Delta t) + \hat{\epsilon}_t, \tag{3.4}$$

where k is the number of lags considered.

From Def 3.1, X does not G-cause Y if and only if the prediction errors of X in the restricted Eq. (3.3) and unrestricted regression models Eq. (3.4) are equal (i.e., they are statistically indistinguishable).

3.3.2 Test for Misspecification

A statistically significant causality can be reported only if the linear models from Eqs. (3.3) and (3.4) are not misspecified. For that purpose, we utilize the BDS test (Brock et al., 1996) for the model misspecification.

The BDS test (Brock et al., 1996) is used to detect nonlinear dependence in time series. When applied to the residuals of a linear model, the BDS tests the null hypothesis that these residuals are independent and identically distributed. The BDS test is a powerful test to detect linear misspecification and nonlinearity (Brock et al., 1996; Barnett et al., 1997).

Let $\epsilon_t = (\epsilon_{t=1}, \dots, \epsilon_{t=n})$ be the residuals of the linear fitted model and define its m-embedding as $\epsilon_t^m = (\epsilon_t, \epsilon_{t-1}, \dots, \epsilon_{t-m+1})$. The m-embedding correlation integral is given by

$$C_{m,n}(\Delta\epsilon) = \frac{2}{k(k-1)} \sum_{s=1}^{t} \sum_{t=s}^{n} \chi(\|\epsilon_s^m - \epsilon_t^m\|, \Delta\epsilon),$$

and

$$C_m(\Delta\epsilon) = \lim_{n\to\infty} C_{m,n}(\Delta\epsilon),$$

where χ is an indicator function where $\chi(\|\epsilon_s^m - \epsilon_t^m\|, \Delta\epsilon) = 1$ if $\|\epsilon_s^m - \epsilon_t^m\| < \Delta\epsilon$ and zero, otherwise.

The null hypothesis of the BDS test assumes that ϵ_t is iid. In this case,

$$C_m(\Delta\epsilon) = C_1(\Delta\epsilon)^m.$$

The BDS statistic is a measure of the extent that this relation holds in the data. This statistic is given by the following:

$$V_m(\Delta\epsilon) = \sqrt{n}\frac{C_m(\Delta\epsilon) - C_1(\Delta\epsilon)^m}{\sigma_m(\Delta\epsilon)},$$

where $\sigma_m(\Delta\epsilon)$ can be estimated as described in (Brock et al., 1996).

The null hypothesis of the BDS test indicates that the model tested is not misspecified and it is rejected at the 5% significance level if $\|V_m(\Delta\epsilon)\| > 1.96$. The parameter $\Delta\epsilon$ is commonly set as a factor of the variance (σ_ϵ) of ϵ.

3.3.3 Analysis of Variance

A one-way ANOVA test is utilized to test if the residuals from Eqs. (3.3) and (3.4) differ from each other significantly.

3.3.4 Multiple Hypotheses Testing Correction

When more than one lag k is tested, a Bonferroni correction is applied to control for multiple hypotheses testing.

3.4 Conclusion

Part II

Algo Trading

Chapter 4

Limit Order

Part III

Portfolio Optimization

Part IV

Machine Learning

Part V

Econophysics

Chapter 5

Entropy

Let X be a random variable and $P_X(x)$ be its probability density function (pdf). The entropy $H(X)$ is a measure of the uncertainty of X and is defined in the discrete case as follows:

$$H(X) = -\sum_{x \in X} P_X(x) \log P_X(x). \tag{5.1}$$

If the log is taken to base two, then the unit of H is the *bit* (binary digit). We employ the natural logarithm which implies the unit in *nat* (natural unit of information).

Given a coupled system (X, Y), where $P_Y(y)$ is the pdf of the random variable Y and $P_{X,Y}$ is the joint pdf between X and Y, the joint entropy between X and Y is given by the following:

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} P_{X,Y}(x, y) \log P_{X,Y}(x, y). \tag{5.2}$$

The conditional entropy is defined by the following:

$$H\left(Y|X\right) = H(X, Y) - H(X). \tag{5.3}$$

We can interpret $H(Y|X)$ as the uncertainty of Y given a realization of X.

5.1 Efficiency and Bubbles in the Crypto and Equity Markets

5.2 Quantifying Non-linear Correlation Between Equity and Commodity Markets

Chapter 6

Transfer Entropy

6.1 Introduction

6.2 Nonlinear G-Causality

To compute the nonlinear G-Causality, we use the concept of Transfer Entropy. Since its introduction (Schreiber, 2000), Transfer Entropy has been recognized as an important tool in the analysis of causal relationships in nonlinear systems (Hlavackovaschindler et al., 2007). It detects directional and dynamical information (Montalto et al., 2014) while not assuming any particular functional form to describe interactions among systems.

The Transfer Entropy can be defined as the difference between the conditional entropies:

$$TE\left(X \rightarrow Y \middle| Z\right) = H\left(Y^F \middle| Y^P, Z^P\right) - H\left(Y^F \middle| X^P, Y^P, Z^P\right), \quad (6.1)$$

which can be rewritten as a sum of Shannon entropies:

$$TE\left(X \rightarrow Y\right) = H\left(Y^P, X^P\right) - H\left(Y^F, Y^P, X^P\right) + H\left(Y^F, Y^P\right) - H\left(Y^P\right), \tag{6.2}$$

where Y^F is a forward time-shifted version of Y at lag Δt relatively to the past time-series X^P, Y^P and Z^P. Within this framework we say that X does not G-cause Y relative to side information Z if and only if $H\left(Y^F|Y^P, Z^P\right) = H\left(Y^F|X^P, Y^P, Z^P\right)$, i.e., when $TE\left(X \rightarrow Y, Z^P\right) = 0$.

Empirically, we reject this null hypothesis of causality if the Transfer Entropy from X to Y is significantly higher than the shuffled version of the original data.

For this we estimate 400 replicates of $TE(X_{Shuffled} \rightarrow Y)$, where $X_{Shuffled}$ is a random permutation of X relatively to Y. We compute the randomized Transfer Entropy at each permutation for each time-shift (Δt) from 1 to 10 days. We then calculated the frequency at which the observed Transfer Entropy was equal or more extreme than the randomized Transfer Entropy. The statistical significance was assessed using p-value < 0.05 after Bonferroni correction.

6.3 The Link Between Linear Granger-causality and Transfer Entropy

It has been shown (Barnett et al., 2009) that linear G-causality and Transfer Entropy are equivalent if all processes are jointly Gaussian. In particular, by assuming the standard measure ($l2$-norm loss function) of linear G-causality for the bivariate case as follows (see Section 3.3 for more details on linear-Granger causality):

$$GC_{X \rightarrow Y} = \log\left(\frac{var(\epsilon_t)}{var(\hat{\epsilon}_t)}\right), \tag{6.3}$$

the following can be proved (Barnett et al., 2009):

$$TE_{X\rightarrow Y} = GC_{X\rightarrow Y}/2. \tag{6.4}$$

This result provides a direct mapping between the Transfer Entropy and the linear G-causality implemented in the standard VAR framework. Hence, it is possible to estimate the TE both in its general form and with its equivalent form for linear G-causality.

6.4 Net Information Flow

Transfer-entropy is an asymmetric measure, i.e., $T_{X\rightarrow Y} \neq T_{Y\rightarrow X}$, and it thus allows the quantification of the directional coupling between systems. The Net Information Flow is defined as

$$\widehat{TE}_{X\rightarrow Y} = TE_{X\rightarrow Y} - TE_{Y\rightarrow X}\ . \tag{6.5}$$

One can interpret this quantity as a measure of the dominant direction of the information flow. In other words, a positive result indicates a dominant information flow from X to Y compared to the other direction or, similarly, it indicates which system provides more predictive information about the other system (Michalowicz et al., 2013).

In the next sections we will provide empirical examples that show that Transfer Entropy can capture information flow in both linear and nonlinear systems.

6.5 Empirical Experiment: Information Flow on Simulated Systems

In this section, we construct simulated systems and test the nonlinear and linear formulations of the net information flow. We show that only the nonlinear formulation of net information flow is able to capture the nonlinear relationships in the simulated systems.

For the nonlinear case, we compute Transfer Entropy as defined in Eq. (6.1). Conversely, to estimate the linear version of the Net Information Flow, we computed the Transfer Entropy using Eq. (6.4), i.e., by estimating linear G-causality Eq. (6.3) under a linear-VAR framework.

6.6 Empirical Experiment: Information Flow on Global Markets

Chapter 7

Financial Networks

7.1 Introduction

Financial markets can be regarded as a complex network in which nodes represent different financial assets and edges represent one or many types of relationships among those assets. Filtered correlation-based networks have successfully been used in the literature to study financial markets structure particularly from observational data derived from empirical financial time series (Bardoscia et al., 2017; Tumminello et al., 2011; Mantegna, 1999; Aste et al., 2010; Tumminello et al., 2010, Tumminello et al. (2005)). The underlying principle is to use correlations from empirical financial time series to construct a sparse network representing the most relevant connections. Analyses on filtered correlation-based networks for information extraction (Song et al., 2008; Aste et al., 2010) have widely been used to explain market interconnectedness from high-dimensional data. Applications include asset allocation (Li et al., 2018; Pozzi et al., 2013), market stability assessments (Morales et al., 2012), hierarchical structure analyses (Mantegna, 1999; Aste et al., 2010; Tumminello et al., 2010; Musmeci et al., 2014; Song et al., 2012) and the identification of lead-lag relationships (Curme et al., 2015).

In this Chapter we will describe how to

- Construct and filter financial networks;
- Build price-based dynamic industry taxonomy;
- Implement a trading strategy based on financial network structure.

7.2 Network Construction

We selected $N = 100$ of the most capitalized companies that were part of the S&P500 index from 09/05/2012 to 08/25/2017. The list of these companies' ticker symbols is reported in the Appendix **??**. For each stock i the financial variable was defined as the daily stock's log-return $R_i(\tau)$ at time τ.

Stock returns R_i and social media opinion scores O_i each amounted to a time series of length equals to 1251 trading days. These series were divided time-wise into $M = 225$ windows $t = 1, 2, ..., M$ of width $T = 126$ trading days. A window step length parameter of $\delta T = 5$ trading days defined the displacement of the window, i.e., the number of trading days between two consecutive windows. The choice of window width T and window step δT is arbitrary, and it is a trade-off between having analysis that is either too dynamic or too smooth. The smaller the window width and the larger the window steps, the more dynamic the data are.

To characterize the synchronous time evolution of assets, we used equal time Kendall's rank coefficients between assets i and j, defined as

$$\rho_{i,j}(t) = \sum_{t' < \tau} sgn(V_i(t') - V_i(\tau)) sgn(V_j(t') - V_j(\tau)), \tag{7.1}$$

where t' and τ are time indexes within the window t and $V_i \in \{R_i, O_i\}$.

Kendall's rank coefficients takes into account possible nonlinear (monotonic) relationships. It fulfill the condition $-1 \leq \rho_{i,j} \leq 1$ and form the $N \times N$ correlation matrix $C(t)$ that served as the basis for the networks constructed in this work. To construct the asset-based financial and social networks, we defined a distance between a pair of stocks. This distance was associated with the edge connecting the stocks, and it reflected the level at which they were correlated. We used a simple non-linear transformation $d_{i,j}(t) = \sqrt{2(1 - \rho_{i,j}(t))}$ to obtain distances with the property $2 \geq d_{i,j} \geq 0$, forming a $N \times N$ symmetric distance matrix $D(t)$.

7.2.1 Network Filtering: Asset Graphs

We extract the $N(N - 1)/2$ distinct distance elements from the upper triangular part of the distance matrix $D(t)$, which were then sorted in an ascending order to form an ordered sequence $d_1(t), d_2(t), \dots, d_{N(N-1)/2}(t)$. Since we require the graph to be representative of the market, it is natural to build the network by including only the strongest connections. This is a network filtering procedure that has been successfully applied in the construction of *asset graphs* for the analyses of market structure Onnela et al. (2003, 2004). The number of edges to include is arbitrary, and we included those from the bottom quartile, which represented the 25% shortest edges in the graph (largest correlations), thus giving $E(t) = \{d_1(t), d_2(t), \dots, d_{\lfloor N/4 \rfloor}(t)\}$.

We denoted $E^F(t)$ as the set of edges constructed from the distance matrix derived from stock returns $R(t)$. The financial network considered is $G^F = (V, E^F)$, where V is the vertex set of stocks.

7.2.2 Network Filtering: MST

7.2.3 Network Filtering: PMFG

7.3 Applications

7.3.1 Industry Taxonomy

7.3.2 Portfolio Construction

Appendix A

Statistical Methods

This Appendix provides details to some of statistical methods used in the book.

A.1 Kernel Density Estimation

In the entropy computation (see Section 5) the empirical probability distribution must be estimated. Histogram-based methods and kernel density estimations are the two main methods for that. Histogram-based is the simplest and most used nonparametric density estimator. Nonetheless, it yields density estimates that have discontinuities and vary significantly depending on the bin size choice.

Also known as the Parzen-Rosenblatt window method, the kernel density estimation (KDE) approach approximates the density function at point x using neighboring observations. However, instead of building up the estimate according to the bin edges as in histograms, the KDE method uses each point of estimation x as the center of a bin of width $2h$ and weight it according to a kernel function. Thereby, the kernel estimate of the probability density function $f(x)$ is defined as

$$\hat{f} = \frac{1}{nh} \sum_{x' \in X} K\left(\frac{x - x'}{h}\right). \tag{A.1}$$

A usual choice for the kernel K, which we use here, is the (Gaussian) radial basis function:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}x^2}. \tag{A.2}$$

The problem of selecting the bandwidth h in Eq. (A.1) is crucial in the density estimation. A large h will oversmooth the estimated density and mask the structure of the data. On the other hand, a small bandwidth will reduce the bias of the density estimate at the expense of a larger variance in the estimates. If we assume that the true distribution is Gaussian and we use a Gaussian kernel, the optimal value of h that minimizes the mean integrated squared error (MISE) is

$$h^* = 1.06\sigma N^{-1/5},$$

where N is the total number of points and σ can be estimated as the sample standard deviation. This bandwidth estimation is often called the Gaussian approximation or Silverman's rule of thumb for kernel density estimation (Silverman and Green, 1986). This is the most commonly-used method and it is here employed. Other common methods are given by (Sheather and Jones, 1991) and (Scott, 1992).

Bibliography

Aste, T., Shaw, W., and Di Matteo, T. (2010). Correlation structure and dynamics in volatile markets. *New Journal of Physics*, 12(8):085009.

Bardoscia, M., Battiston, S., Caccioli, F., and Caldarelli, G. (2017). Pathways towards instability in financial networks. *Nature Communications*, 8:14416.

Barnett, L., Barrett, A. B., and Seth, A. K. (2009). Granger causality and transfer entropy are equivalent for gaussian variables. *Phys. Rev. Lett.*, 103:238701.

Barnett, W. A., Gallant, A. R., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T., and Jensen, M. J. (1997). A single-blind controlled competition among tests for nonlinearity and chaos. *Journal of Econometrics*, 82:157–192.

Brock, W. A., Scheinkman, J. A., Dechert, W. D., and LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*, 15(3):197–235.

Curme, C., Stanley, H. E., and Vodenska, I. (2015). Coupled network approach to predictability of financial market returns and news sentiments. *International Journal of Theoretical and Applied Finance*, 18(07):1550043.

Granger, C. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38.

Hlavackovaschindler, K., Palus, M., Vejmelka, M., and Bhattacharya, J. (2007). Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46.

Li, Y., Jiang, X.-F., Tian, Y., Li, S.-P., and Zheng, B. (2018). Portfolio optimization based on network topology. *Physica A: Statistical Mechanics and its Applications.*

Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B - Condensed Matter and Complex Systems*, 11(1):193–197.

Michalowicz, J. V., Nichols, J. M., and Bucholtz, F. (2013). *Handbook of Differential Entropy.* Chapman & Hall/CRC.

Montalto, A., Faes, L., and Marinazzo, D. (2014). Mute: A matlab toolbox to compare established and novel estimators of the multivariate transfer entropy. *PLoS ONE*, 9(10):e109462.

Morales, R., Di Matteo, T., Gramatica, R., and Aste, T. (2012). Dynamical generalized hurst exponent as a tool to monitor unstable periods in financial time series. *Physica A: Statistical Mechanics and its Applications*, 391(11):3180–3189.

Musmeci, N., Aste, T., and di Matteo, T. (2014). Clustering and hierarchy of financial markets data: advantages of the dbht. *CoRR.*

Onnela, J. P., Chakraborti, A., Kaski, K., Kertész, J., and Kanto, A. (2003). Asset trees and asset graphs in financial markets. *Physica Scripta*, 2003(T106):48.

Onnela, J. P., Kaski, K., and Kertész, J. (2004). Clustering and information in correlation based financial networks. *The European Physical Journal B*, 38(2):353–362.

Pozzi, F., Di Matteo, T., and Aste, T. (2013). Spread of risk across financial markets: better to invest in the peripheries. *Scientific reports*, 3.

Schreiber, T. (2000). Measuring information transfer. *Phys. Rev. Lett.*, 85:461–464.

Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization.* A Wiley-interscience publication. Wiley.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):pp. 683–690.

Silverman, B. W. and Green, P. J. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Song, W.-M., Aste, T., and Di Matteo, T. (2008). Analysis on filtered correlation graph for information extraction. *Statistical Mechanics of Molecular Biophysics*, page 88.

Song, W.-M., Di Matteo, T., and Aste, T. (2012). Hierarchical information clustering by means of topologically embedded graphs. *PLoS One*, 7(3):e31929.

Tumminello, M., Aste, T., Di Matteo, T., and Mantegna, R. N. (2005). A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10421–10426.

Tumminello, M., Lillo, F., and Mantegna, R. N. (2010). Correlation, hierarchies, and networks in financial markets. *Journal of Economic Behavior & Organization*, 75(1):40 – 58. Transdisciplinary Perspectives on Economic Complexity.

Tumminello, M., Miccichè, S., Lillo, F., Piilo, J., and Mantegna, R. N. (2011). Statistically validated networks in bipartite complex systems. *PLoS ONE*, 6(3):1–11.

Wickham, H. and Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.* O'Reilly Media, Inc., 1st edition.

Wiener, N. (1956). The theory of prediction. In Beckenbach, E. F., editor, *Modern mathematics for engineers*, chapter 8. McGraw-Hill, New York.