UNIVERSITÄT
BERN

Empirical Finance

Empirical Asset Pricing & Time Series Models

Sascha Jakob
Institute for Financial Management
Spring 2022

Roadmap

Course topics:

1. Statistics for financial time series
2. Efficient market hypothesis and anomalies
3. CAPM & factor models

Backup topics:

a. Time series models
b. Introduction to financial machine learning

Housekeeping

- Only chapter 1-3 are part of the official curriculum, i.e., up to slide 115.
- There is a lot of information in this set of slides. **By no means do I expect that you understand everything in detail**. Rather, the goal is to make you aware of the methods and tools that exists and to give you an idea of how they work and also to provide you with slides where you can look them up in case you need them at a later point in time.
- Many of the methodological tools we discuss are readily available in `STATA`, `Python`, `MATLAB`, and `R`.
- Your grade will be a weighted average:

Assignment 1	ECF	40%
Assignment 2	EAP	40%
Presentation	Paper	20%

- Assistant: Google, Stack Exchange, Stack Overflow, Stata forum,...
- All errors are mine!

Course Goal and Structure

The goal of this course is to get you acquainted with the empirical methods used in asset pricing. Therefore, sessions are structured as follows:

- $\rightsquigarrow$ 2 Lectures of theory
- $\rightsquigarrow$ 1 Lecture of coding empirical exercises

Assignment

A few words regarding the assignment:

- I will upload the assignment the day after the second session.
- The goal of the assignment is to get you acquainted with the fundamental methodologies used in empirical asset pricing.
- You can stick to `STATA` but you may also work with `Python`, `MATLAB` or `R`. Feel free to choose whatever suits you best!

Helpful Readings

- Bali, T. G., R.F. Engle, and S. Murray (2016), Empirical Asset Pricing - The Cross Section of Stock Returns, Wiley
  - Course topics: Efficient market hypothesis and anomalies
- Campbell, John Y., A. W. Lo, and A. C. MacKinlay (1997), The Econometrics of Financial Markets, Princeton
  - Course topics: Statistics for financial time series, Efficient market hypothesis and anomalies, CAPM & factor models
- Hamilton, J. (1994), Time Series Analysis, Princeton
  - Course topics: Time series models
- López de Prado, M. (2018), Advances in Financial Machine Learning, Wiley
  - Course topics: Introduction to financial machine learning
- Wasserman, Larry (2004), All of Statistics: A Concise Course in Statistical Inference, Springer
  - Course topics: Statistics for financial time series

Presentation Papers

- Baker, M. and Jeffrey Wurgler, 2006, Investor sentiment and the cross-section of stock returns, *The Journal of Finance* 61, 1645 - 1680
- Fama, E.F. and Kenneth R. French, 2015, A five-factor asset pricing model, *The Journal of Financial Economics* 116, 1 - 22
- Frazzini, A. and Lasse Heje Pedersen, 2014, Betting against beta, *The Journal of Financial Economics* 111, 1 - 25
- Hou, K. and Tobias J. Moskowitz, 2005, Market Frictions, Price Delay, and the Cross-Section of Expected Returns, *The Review of Financial Studies* 18, 981 - 1020
- Peyer, U. and Theo Vermaelen, 2009, The Nature and Persistence of Buyback Anomalies, *The Review of Financial Studies* 22, 1693 - 1745
- You may also propose an empirical asset pricing paper of your choice that fits the curriculum.

Statistics for financial time series

Statistics for financial time series

The objectives of this chapter are the following:

- Brief description of the variables we are going to use – Asset Returns
- Main properties of the distribution of asset returns – Moments
- How to test assumptions regarding the distribution of asset returns
- How to test assumptions regarding the time dependency of asset returns

Prices or Returns?

In order to analyse a time series it is essential that the series behaves the same anywhere along its time domain. Put differently, the expectations we build about the time series should be the same at any given point in time. This does not imply that the time series does not change over time. Rather, it implies that the manner in which the time series changes does not change over time. This is the concept of stationarity.

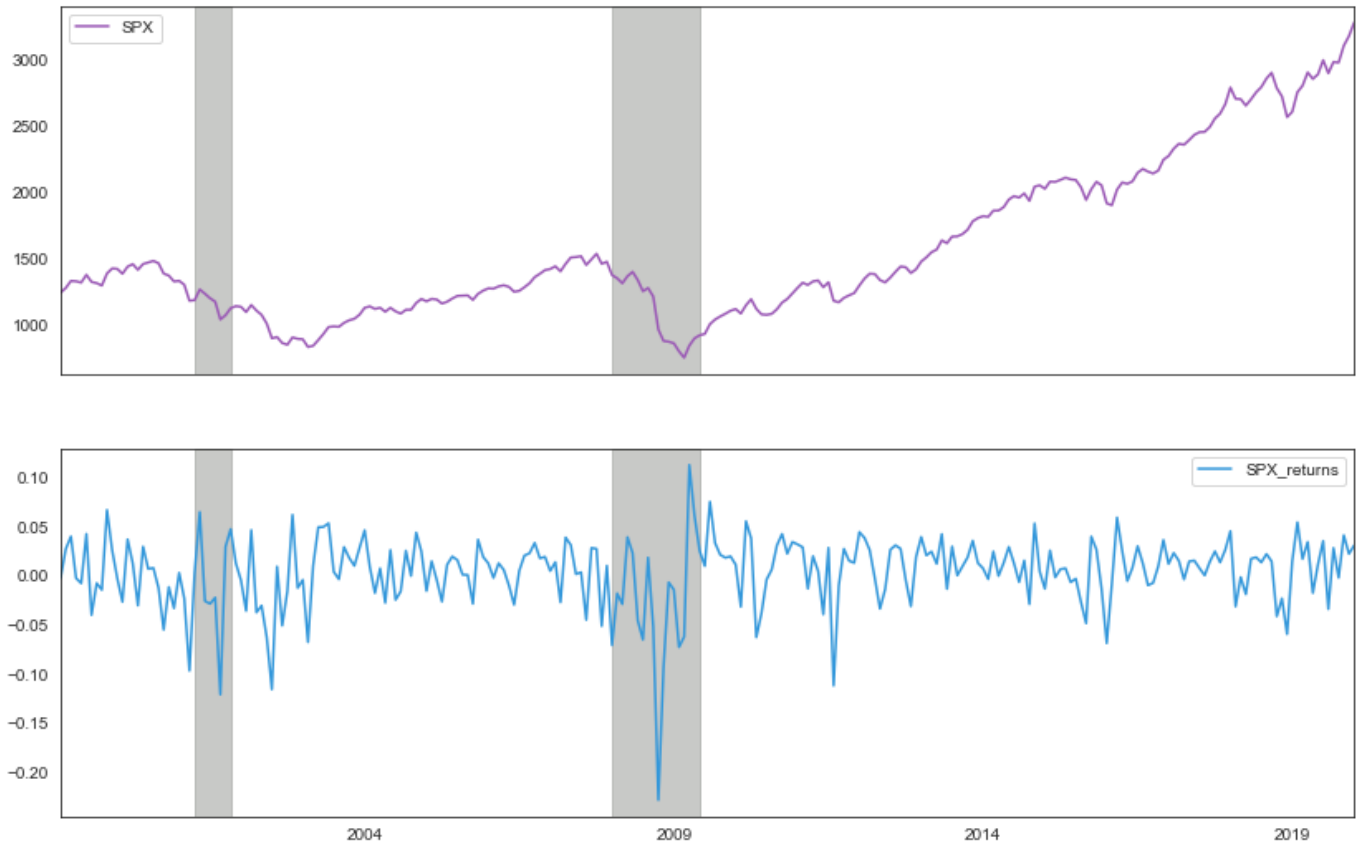
Stationarity

Stationarity

Formally, let $\{X_t\}$ be a stochastic process with cumulative distribution function $F_X(x_t, x_{t+1}, \ldots, x_{t+n})$. Then $\{X_t\}$ is said to be strictly stationary if

$$F_X(x_t, x_{t+1}, \ldots, x_{t+n}) = F_X(x_{t+i}, x_{t+i+1}, \ldots, x_{t+i+n}), \forall i, t \tag{1}$$

Prices or Returns?



Augmented Dickey-Fuller test

Stationarity implies that the mean be constant. This can only hold if the process reverts to its mean whenever it deviates. Assume we have some higher-order autocorrelated process, then we can test whether the previous level of the process predicts the subsequent change. Specifically, we test whether $\gamma < 0$ in the following specification:

$$\Delta x_t = \alpha + \beta t + \gamma x_{t-1} + \delta_1 \Delta x_{t-1} + \ldots + \delta_{p-1} \Delta x_{t-p+1} + \varepsilon_t \tag{2}$$

where α is a constant, β the coefficient on a time trend t, and p is the order of lag.

Augmented Dickey-Fuller test

The test statistic for the ADF test is:

$$ADF = \frac{\hat{\gamma}}{se(\hat{\gamma})} \tag{3}$$

and follows a Dickey-Fuller distribution. If $\gamma = 0$ the process is non-stationary (H_0). The optimal lag order p is set using some information criterion such as the Bayesian or Akaike information criterion.

Command

In `STATA` the command for the augmented Dickey-Fuller test is dfuller, in `MATLAB` it is adftest, and in `Python` it is statsmodels.tsa.stattools.adfuller

Prices or Returns?

It turns out that, in almost all cases, only returns are stationary and thus allow for proper statistical inference. Generally, there are two main reasons as to why investigating the properties of returns is more appropriate:

- Properties of returns are easier to handle than the properties of prices.
- Investors are mostly interested in returns for their investment decisions rather than the price level.

Types of Returns

We distinguish between two types of returns:

- Simple (discrete) returns, $R_t = \frac{P_t}{P_{t-1}} - 1$
- Log (continuous) returns, $r_t = \ln(P_t) - \ln(P_{t-1}) = \ln(\frac{P_t}{P_{t-1}})$

Using simple returns or log returns is an empirical issue, which depends in general on the problem at hand.

Simple vs. Log Returns

Temporal Aggregation

Holding the asset for n periods from t to $t+n$ yields the n-period simple return:

$$R_{t+n,t} = \left[\prod_{i=1}^{N}(1+R_{t+i})\right] - 1 \tag{4}$$

If one uses log returns, the n-period log return is simply the sum of continuously compounded one-period log returns, so that:

$$r_{t+n,t} = \ln\left[\prod_{i=1}^{N}(1+R_{t+i})\right] = \sum_{i=1}^{N}\ln(1+R_{t+i}) = \sum_{i=1}^{N} r_{t+i} \tag{5}$$

If we aggregate over time, log returns are often less cumbersome than simple returns since the product of a normally distributed variable need not be normally distributed while the sum remains or converges to a normal distribution (Central Limit Theorem).

Simple vs. Log Returns

Contemporaneous Aggregation

Log returns have one major drawback when computing a portfolio. The simple return of an N-asset portfolio is the weighted average of the simple returns of the N assets. Let p be the portfolio w_i the portfolio weight of asset $i \in N$. Then the simple portfolio return is:

$$R_{p,t} = \sum_{i=1}^{N} w_i R_{i,t} \tag{6}$$

The log portfolio return is:

$$r_{p,t} = \ln\left(\sum_{i=1}^{N} w_i e^{r_{i,t}}\right) \neq \sum_{i=1}^{N} w_i r_{i,t} \tag{7}$$

Therefore, weighted log returns lead to flawed portfolio returns.

Simple vs. Log Returns

There exist other reasons as to why log returns might be preferable:

- Non-negativity of prices: If we assume that prices are distributed log normally, then $\ln(1+R_i)$ tends to be normally distributed.
- Approximate simple-log equality: when returns are very small (common for trades with short holding durations), the following log returns are very close in value to simple returns.
- Mathematical ease: from calculus, we are reminded (ignoring the constant of integration) that $e^x = \int e^x dx = \frac{\partial}{\partial x} e^x = e^x$
- Differencing: log returns are a first difference which is useful in some time series models and machine learning applications.
- From probability theory we know that the sum of normally distributed variables follows a normal distribution itself while the product need not to.

Other Return Definitions

Dividends

For assets with periodic dividend payments, asset returns have to be redefined:

- Simple returns: $R_t = \frac{P_t + D_t}{P_{t-1}}$
- Log returns: $r_t = \ln(P_t + D_t) - \ln(P_{t-1})$

where D_t is the dividend payment of an asset between dates $t-1$ and t. Most reference indices include dividend payments (exception: German DAX index). Some financial institutions (Morgan Stanley, MSCI indices) produce reference indices without (price index) or with reinvested dividend payments (total return index).

Excess Returns

Excess return is simply the difference between the asset's return and the return on the risk-free asset, in practice the short-term Treasury bill return, $Z_{i,t} = R_{i,t} - R_{f,t}$.

Characteristics of Returns

Early work in finance has made some very strong assumptions for asset returns:

- Normality of Returns
- Time independency (i.i.d. process)

For the remainder of this chapter we will have a more detailed look on these two assumptions.

Moments of a Random Variable

Assume r.v. X (continuous) has the following cumulative distribution function (cdf):

$$F_X(x) = \mathbb{P}[X \leq x] = \int_{-\infty}^{x} f_X(u)du \tag{8}$$

where $f_X(x)$ is the probability density function (pdf, or probability mass function if discrete).

Moments of a Random Variable

The mean or expected value of X is the first moment:

$$\mu = \int_{\mathbb{R}} x f_X(x) dx = \mathbb{E}[X] \equiv \frac{1}{T} \sum_{t=1}^{T} r_t \tag{9}$$

and the variance of X is the second moment:

$$\sigma^2 = \mathbb{V}[X] = \int_{\mathbb{R}} (x-\mu)^2 f_X(x) dx = \mathbb{E}[(X-\mu)^2] \equiv \frac{1}{T} \sum_{t=1}^{T} (r_t - \mu)^2 \tag{10}$$

Higher Moments

The 3^{rd} moment, skewness, measures the asymmetry of the distribution:

$$m_3 = \mathbb{S}[X] = \int_{\mathbb{R}} (x-\mu)^3 f_X(x) dx = \mathbb{E}[(X-\mu)^3] \tag{11}$$

and the 4^{th} moment, kurtosis, measures the presence of fat tails:

$$m_4 = \mathbb{K}[X] = \int_{\mathbb{R}} (x-\mu)^4 f_X(x) dx = \mathbb{E}[(X-\mu)^4] \tag{12}$$

Skewness and Kurtosis

Skewness and Kurtosis

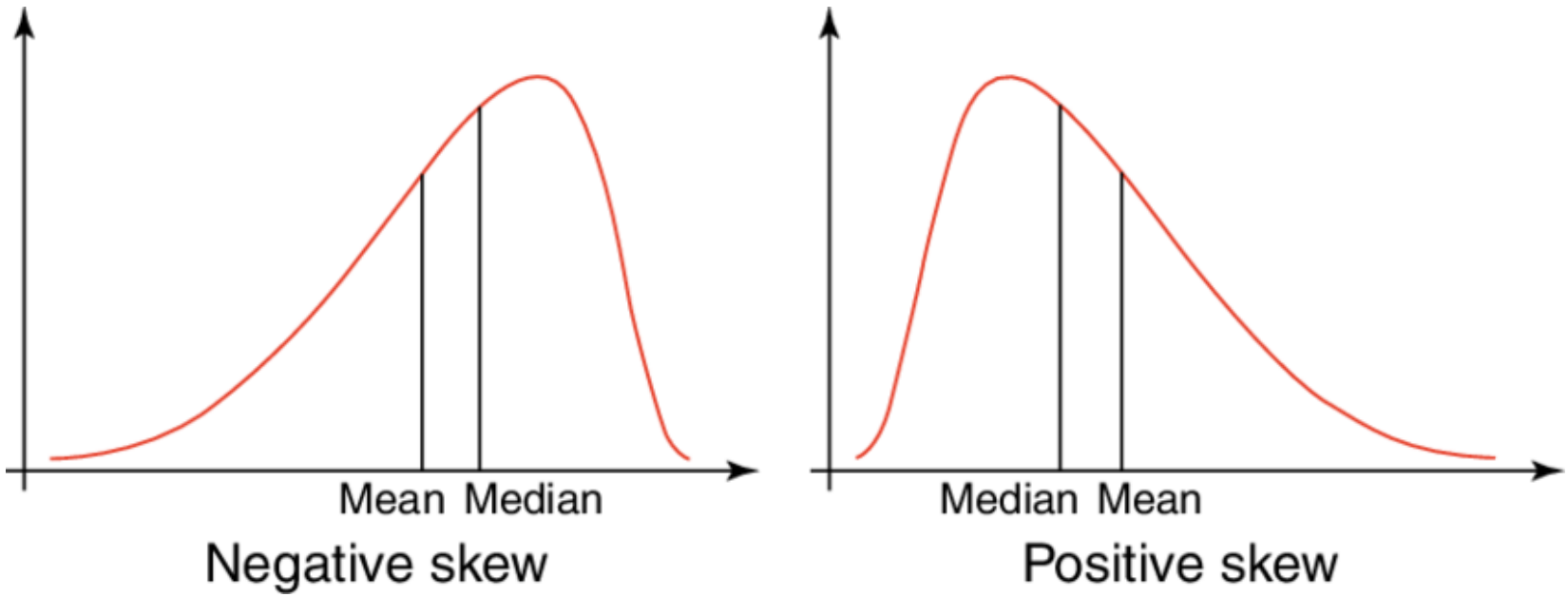
Standardized skewness and kurtosis are defined as:

$$\mathbb{S}[X] = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{13}$$

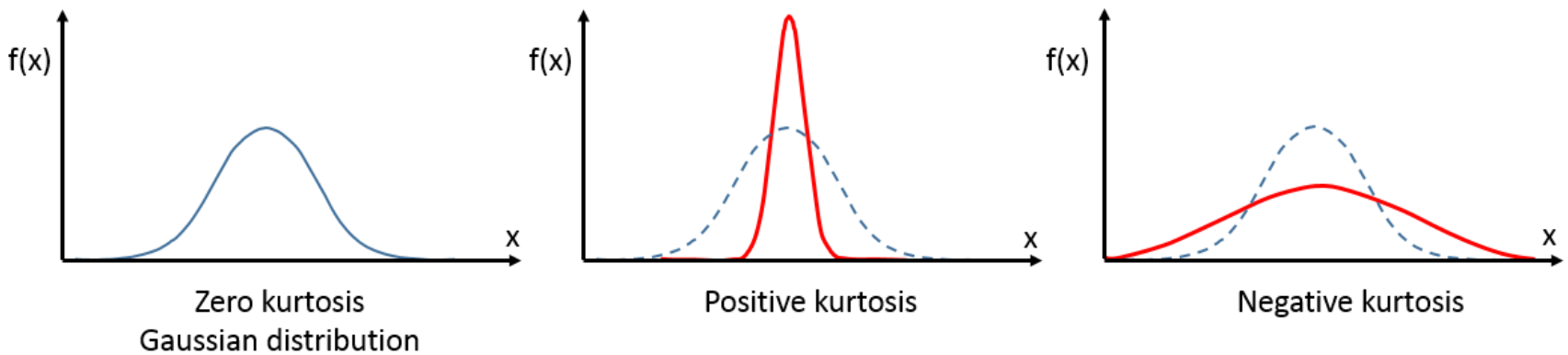
$$\mathbb{K}[X] = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \tag{14}$$

- When $S[X]$ is negative, large realizations of X are more often negative than positive, suggesting that crashes are more likely to occur than booms.
- A large kurtosis $\mathbb{K}[X]$ implies that large realizations (either positive or negative) are more likely to occur.
- For the normal distribution, skewness is equal to zero, while standardized kurtosis is equal to 3. We define the excess kurtosis as $\mathbb{K}[X] - 3$, such that for a normal distribution both are zero.

Skewness



Kurtosis


f(x)
x
Zero kurtosis
Gaussian distribution
f(x)
x
Positive kurtosis
f(x)
x
Negative kurtosis

Skewness and Kurtosis

The sample counterparts of the standardized skewness and kurtosis of returns are given by:

$$\mathbb{S}[r] = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{r_t - \mu}{\sigma}\right)^3 \tag{15}$$

$$\mathbb{K}[r] = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{r_t - \mu}{\sigma}\right)^4 \tag{16}$$

For a normal distribution, the skewness and the excess kurtosis are zero. This implication is easily testable!

Command

In `STATA` skewness and kurtosis are reported in the detailed version of the summarize command sum, d and stored in r(skewness) and r(kurtosis), respectively, in `MATLAB` skewness and kurtosis are computed with the command skewness and kurtosis, respectively, and in `Python` the respective commands are scipy.stats.skew and scipy.stats.kurtosis

Test for Normality

Several tests for the null hypothesis of normality have been developed, focusing on different implications of the normality assumption. We will have a look at the following two:

- the moments of the distribution (Jarque-Bera test)
- the properties of the empirical distribution function (Kolmogorov-Smirnov test)

Jarque-Bera Test

The Jarque-Bera test is based on the fact that skewness and excess kurtosis are jointly equal to zero under normality. The null hypothesis of the test is: Skewness = 0 and Kurtosis = 3. Since the normal distribution is defined by the first two moments, there is no constraint on the mean and variance under the null.

The JB test statistic is defined as:

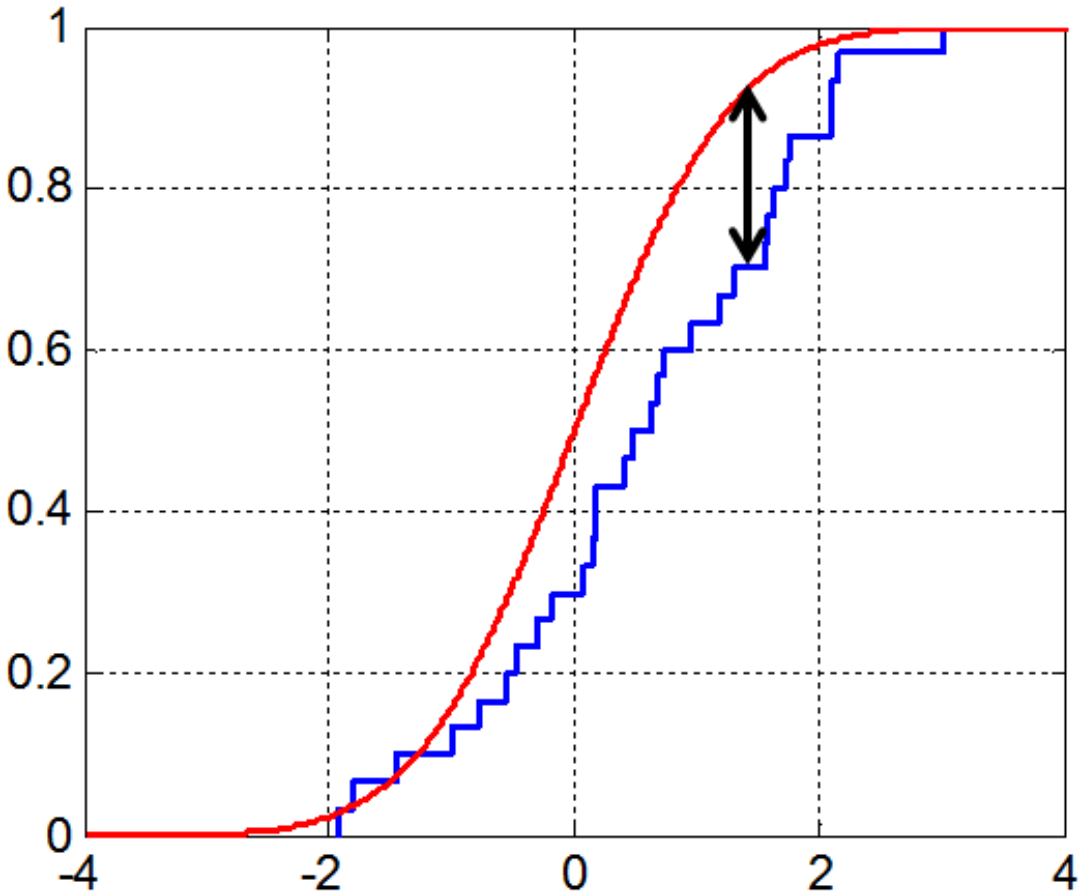
$$JB = T\left[\frac{\mathbb{S}^2}{6} + \frac{(\mathbb{K}-3)^2}{24}\right] \tag{17}$$

Under the null hypothesis, it is asymptotically distributed as a $\chi^2(2)$.
If $JB \geq \chi^2_\alpha(2)$, then the null hypothesis has to be rejected at level α. For instance, If the JB statistic is larger than 6, we should reject the normality hypothesis with only 5% of chances of being wrong.

Command

In `STATA` the relevant command for the Jarque-Bera test is varnorm,in `MATLAB` it is jbtest, and in `Python` it is scipy.stats.jarque_bera

Kolmogorov-Smirnov Test



Kolmogorov-Smirnov Test

The idea of the Kolmogorov-Smirnov test is that of comparing the empirical distribution with the assumed theoretical cdf $F^*(x;\theta)$, e.g., a normal distribution. It belongs to the empirical distribution function goodness-of-fit tests.

Assume that the data consist of a time series of returns $(r_1, r_2, \ldots, r_T)$ associated with some unknown cdf, denoted by $F_r(.)$. As the true distribution $F_r(.)$ is unknown, it is approximated by the empirical cdf, denoted $G_T(.)$, defined as:

$$G_T(X) = \frac{1}{T}\sum_{t=1}^{T} 1_{\{r_t \leq x\}} \tag{18}$$

$G_T(.)$ is a step function that has steps of height $[1/T]$ at each observation. The empirical cdf $G_T(x)$ is compared with the assumed distribution function $F^*(x;\theta)$ in order to evaluate whether there is a good fit.

Kolmogorov-Smirnov Test

One of the simplest measures of the difference between two cdfs is the largest distance between the two functions $G_T(x)$ and $F^*(x;\theta)$. This is the KS test statistic suggested by Kolmogorov (1933), defined as:

$$KS = \sup_{t=1,\ldots,T} \mid G_T(x) - F^*(x;\theta) \mid \tag{19}$$

In practice, this test is very easy to implement:

- 1. Sort the sample data in ascending order and denote the new sample $\{\tilde{r}_t\}_{t=1}^T$, with $\tilde{r}_1 \neq \ldots \neq \tilde{r}_T$. Then, by construction, we have $G_T(\tilde{r}_t) = \frac{t}{T}$
- 2. Evaluate the assumed theoretical cdf $F^*(\tilde{r}_t;\theta)$ for all values $\{\tilde{r}_t\}_{t=1}^T$, e.g., for a normal distribution.
- 3. Compute the KS test statistic: $KS = \sup_{t=1,\ldots,T} \mid F^*(\tilde{r}_t) - \frac{t}{T} \mid$

The KS statistic follows a Kolmogorov distribution.

Command

In `STATA` the relevant command for the Kolmogorov-Smirnov test is ksmirnov, in `MATLAB` it is kstest, and in `Python` it is scipy.stats.kstest

Serial correlation

We now consider if the moments of a given time series are time dependent. Suppose the time series $(r_1, r_2, \ldots, r_T)$ is weakly (or covariance) stationary, such that:

$$\begin{aligned} \mathbb{E}[r_t] &= \mu \quad \forall t \\ \mathbb{V}[r_t] &= \sigma^2 \quad \forall t \\ \mathbb{COV}[r_t, r_{t-k}] &= \gamma_k \quad \forall t, k \neq 0 \end{aligned}$$

The auto-covariance γ_k measures the dependency of r_t with respect to its own past observation r_{t-k}. For a weakly stationary process, the auto-covariance γ_k depends on the horizon k, but not on the date t.

Serial correlation

Since the auto-covariance is an unbounded measure of dependency, we generally prefer the autocorrelation function (ACF):

$$\rho_k = \frac{\mathbb{COV}[r_t, r_{t-k}]}{\sqrt{\mathbb{V}[r_t]\mathbb{V}[r_{t-k}]}} = \frac{\mathbb{COV}[r_t, r_{t-k}]}{\mathbb{V}[r_t]} = \frac{\gamma_k}{\gamma_0} \tag{20}$$

Simply put, this is the correlation between an observation and its lagged observation with horizon k.

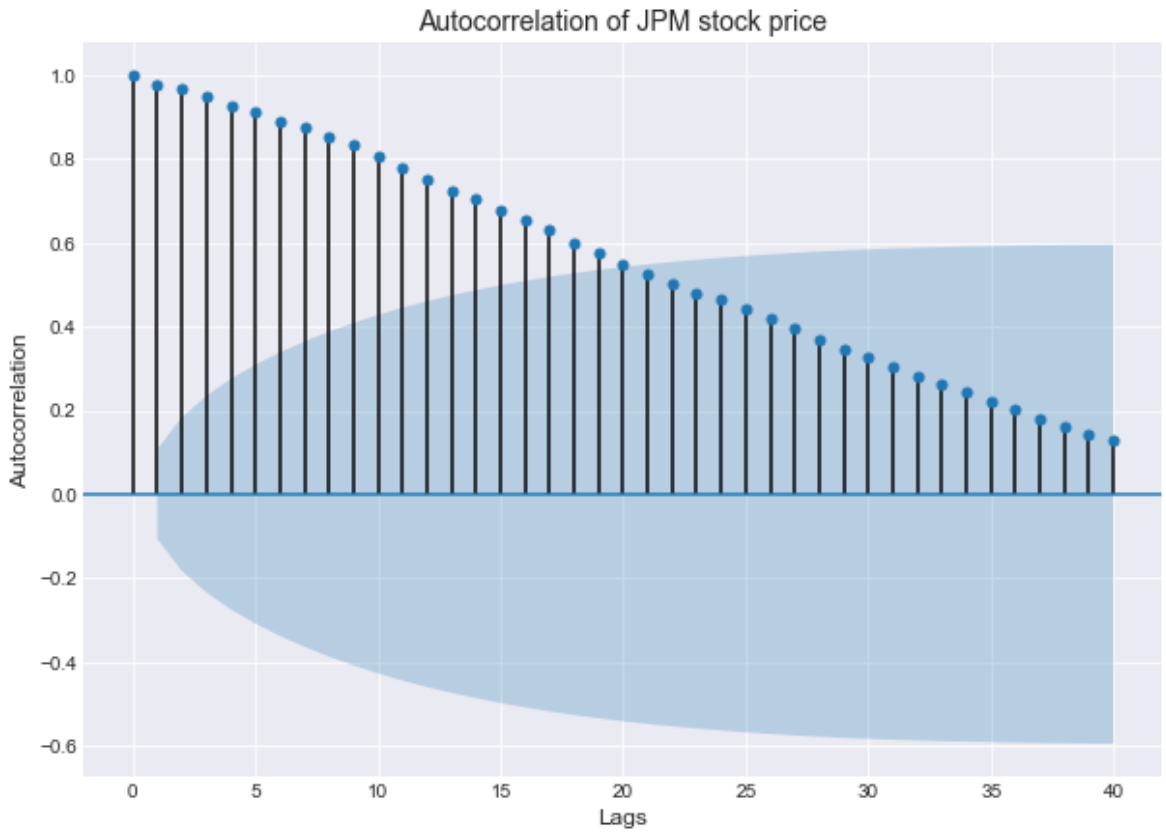
We now want to test the null hypothesis that the n first serial correlations are equal to 0, $H_0 : \rho_1 = ... = \rho_n = 0$ versus the alternative $H_a : \rho_k \neq 0$, for some $k = 1, \ldots, n$.

Command

In `STATA` the relevant command to plot the ACF is ac, in `MATLAB` it is autocorr, and in `Python` it is statsmodels.graphics.tsaplots.plot_acf

Autocorrelation Function

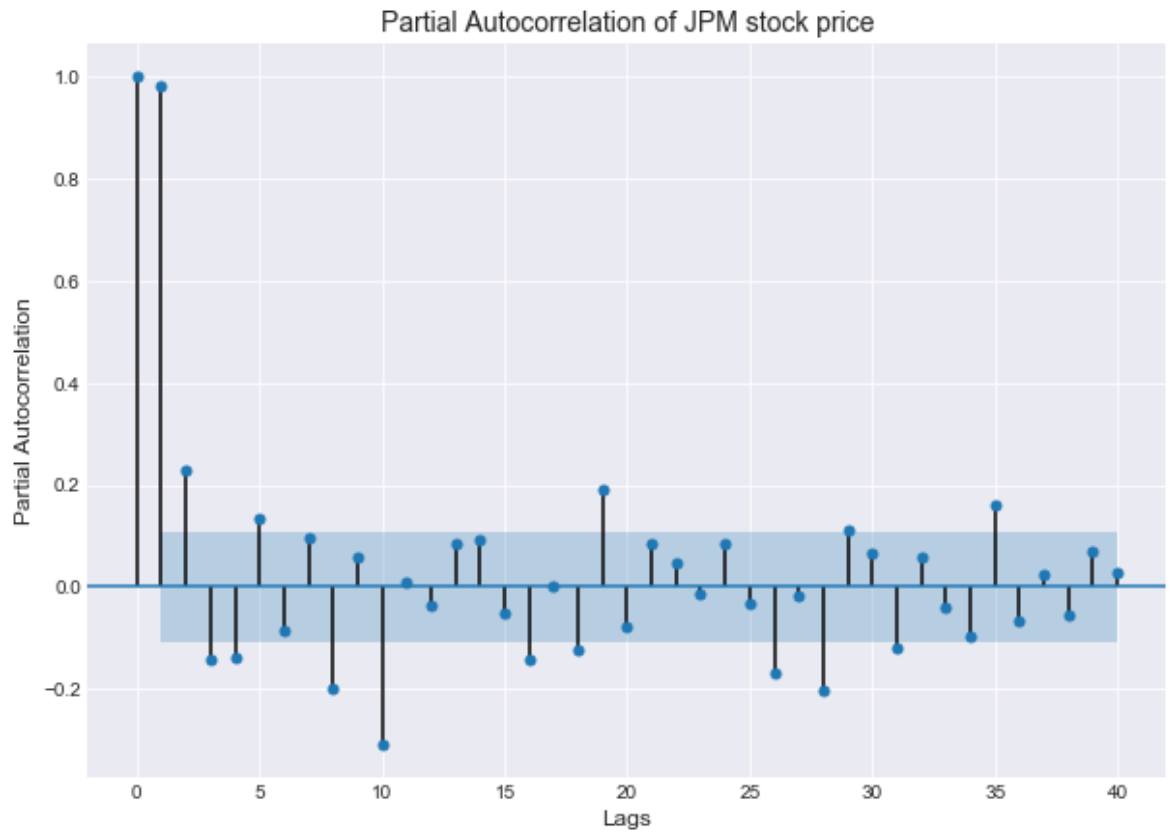
We often start by plotting the ACF, i.e., ρ_k for k lags:



The ACF measures both direct and indirect autocorrelation, i.e., through shorter lags. If we are interested in direct autocorrelation only, the partial autocorrelation function is a better fit.

Partial Autocorrelation Function

The partial autocorrelation function (PACF) measures only the direct correlation between two observations and not the indirect autocorrelation through shorter lags. It is a regression of a variable on its own lags:



Ljung-Box Test

A statistic to test the significance of autocorrelation coefficients with god finite sample properties is the Lung-Box Q-statistic:

$$Q_n = T(T+2)\sum_{j=1}^{n}\frac{1}{T-j}\rho_j^2 \quad \backsim \chi^2(n) \tag{21}$$

If $Q_n \geq \chi^2_\alpha(n)$, then the null hypothesis of no autocorrelation has to be rejected at level α. A common practice consists in testing the nullity of the n first serial correlations for different values of n.

Command

In `STATA` the relevant command for the Box-Pierce/Ljung-Box test is wntestq, in `MATLAB` it is lbqtest, and in `Python` it is statsmodels.stats.diagnostic.acorr_ljungbox

Newey-West Adjusted Standard Errors

If time-series data is autocorrelated (and/or heteroskedastic), standard errors and thus p-values and t-statistics may be inaccurate. To account for these issues in a time-series analysis, one should use Newey–West adjusted standard errors (Newey and West (1987)), similarly to using robust standard errors to account for heteroskedasticity in panel data. Consider the following linear least squares model with k regressors:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon \tag{22}$$

with covariance matrix of $\boldsymbol{\beta}$:

$$\mathbb{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\hat{\boldsymbol{\Omega}}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \tag{23}$$

where $\boldsymbol{\Omega} = \mathbb{E}[\varepsilon'\varepsilon]$

Newey-West Adjusted Standard Errors

For k lags, Newey-West adjusted errors compute $(\mathbf{X}'\hat{\mathbf{\Omega}}\mathbf{X})$ as:

$$(\mathbf{X}'\hat{\mathbf{\Omega}}\mathbf{X}) = \underbrace{\frac{T}{T-k}\sum_{t}^{T}\hat{\varepsilon}_t^2\mathbf{x}_t'\mathbf{x}_t}_{heteroskedasticity} + \underbrace{\frac{T}{T-k}\sum_{l=1}^{L}\left(1-\frac{l}{L+1}\right)\sum_{t=l+1}^{T}\hat{\varepsilon}_t\hat{\varepsilon}_{t-l}(\mathbf{x}_t'\mathbf{x}_{t-l}+\mathbf{x}_{t-l}'\mathbf{x}_t)}_{autocorrelation} \tag{24}$$

where T is the number of periods (observations), k the number of regressors, $\hat{\varepsilon}_t = y_t - \mathbf{x}_t'\hat{\boldsymbol{\beta}}$, and $\mathbf{x}_t$ the vector of observations at time t in $\mathbf{X}$.

There is no closed-form solution for the number of lags, L. It is approximately:

$$4(T/100)^{\alpha}$$

where α is either 2/9 or 4/25. As a rule of thumb, this leads to a lag of 4 to 6 when using with monthly data when data starts in the 60's.

Command

In `STATA` the relevant command to compute Newey-West standard errors is newey and in `Python` it is statsmodels.regression.linear_model.OLSResults.get_robustcov_results after fitting the model.

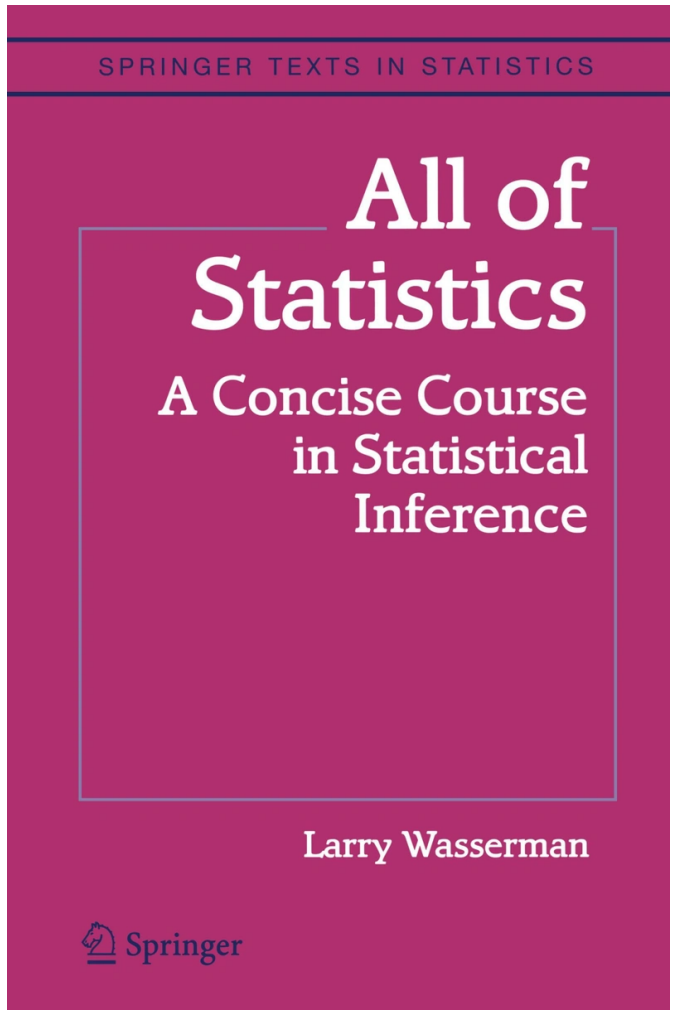
Reality Check

For actual financial returns, the assumptions of normality and time independency have been rejected in several ways:

- Normality
  - Asymmetry
  - Fat tails
- Time independency
  - Serial correlations
  - Volatility clustering
  - Asymmetric volatility
  - Time-varying correlation between different assets

Reading Tip

Wasserman, Larry (2004), All of Statistics: A Concise Course in Statistical Inference, Springer



Efficient Market Hypothesis and Anomalies

Efficient Market Hypothesis

The objectives of this chapter are the following:

- EMH-definitions
- Reviewing the three forms of market efficiency
- Market anomalies
- Portfolio analysis

Definitions

The efficient market hypothesis dates back at least to Samuelson (1965):

- "In an informationally efficient market, price changes must be unforecastable if they are properly anticipated."

Definition of EMH by Fama (1970)

- "A market in which prices always «fully reflect» available information is called «efficient»."

Definitions

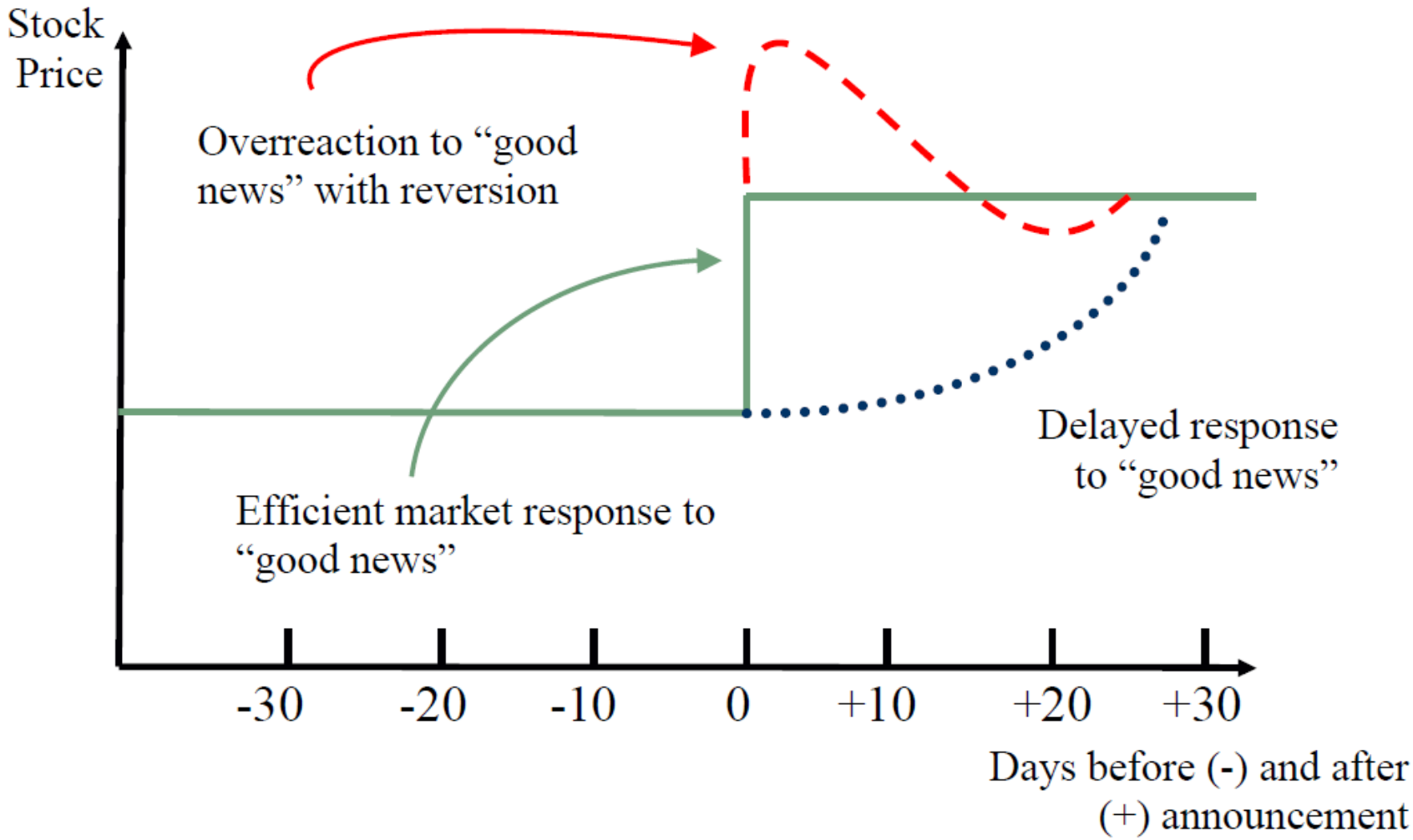
More explicit definition by Malkiel (1992):

1. "A capital market is said to be efficient if it fully and correctly reflects all relevant information in determining security prices"
   - Fama's definition

2. "Formally, the market is said to be efficient with respect to some information set if security prices would be unaffected by revealing that information to all participants"
   - Test the EMH by revealing information to market participants and measuring the reaction of security prices. If prices do not move when information is revealed, then the market is efficient with respect to that information.

3. "Moreover, efficiency with respect to an information set implies that it is impossible to make economic profits by trading on the basis of that information set"
   - Test the EMH by measuring the profits that can be made by trading on information.

Market Reaction to new Information



Empirical Implications

The EMH states that abnormal returns are unpredictable. There are several important implications in this statement:

- Anything beyond normal returns is unpredictable, i.e., returns can be predictable, but abnormal returns cannot.
- Abnormal returns can be unpredictable, but some functions of abnormal returns can be predictable.
- Predictability is conditional on a given set of information. Depending on the information set, returns may be predictable or may not.

Empirical Implications

Some other important concepts of the EMH are:

- Market efficiency does not imply that the return should be zero. Rather, it means that there is no profit beyond the normally required return.
- Abnormal returns map to the difference between the return on a security and its normal, or expected, return.
- Forecasts of the abnormal return are constructed using the chosen information set. If the abnormal return is forecastable, then the EMH is rejected.
- Tests of the EMH are in fact joint tests, since they test (1) the EMH and (2) the model for normal returns. Therefore, if the joint hypothesis is rejected, it may be due to the rejection of the equilibrium model or to the rejection of the EMH.

Forms of Efficiency

There are three forms of efficiency:

- Weak-form efficiency: The information set includes only the history of prices or returns. Using past prices or returns (technical analysis) does not yield a positive return. Weak-form efficiency is related to the hypothesis of random walk in prices or non-predictability of returns.
- Semistrong-form efficiency: The information set includes all information known to all market participants (publicly available information). Using public information (fundamental analysis) does not yield a positive abnormal return.
- Strong-form efficiency: The information set includes all information known to any market participants (private information). Even private information does not yield a positive abnormal return.

Tests of Efficiency

Tests for weak-form efficiency

- Test whether returns are martingales, i.e., $\mathbb{E}[r_{t+1} \mid \mathscr{F}_t] = r_t$, such as the Portmanteau test or variance-ratio test.

Tests for semistrong-form efficiency

- Event-studies: You can find more details in the appendix ▸ appendix
- Portfolio analysis: Tests of whether actual "trading rules" can earn abnormal profits after accounting for transaction costs and various risk sources of the assets (anomalies).
- Test of information efficiency: Tests of whether abnormal returns, $\varepsilon_{t+1} = r_{t+1} - \mathbb{E}_t[r_{t+1}]$, are independent of the variables $\mathbf{\Gamma}_t$ in the information set. Such a test can be based on the regression:

$$r_{t+1} = \mathbb{E}_t[r_{t+1}] + \boldsymbol{\beta}' \mathbf{\Gamma}_t + \varepsilon_{t+1} \tag{25}$$

where $\mathbb{E}_t[r_{t+1}]$ is the equilibrium return. If $\mathbf{\Gamma}_t$ has any additional explanatory power, then $r_{t+1} - \mathbb{E}_t[r_{t+1}]$ is predictable.

Anomalies

Some anomalies have been detected in capital markets, i.e., market behaviour that is inconsistent with the EMH:

- Calendar anomalies
  - Week-end effect
  - January effects
- Other anomalies
  - Size effect
  - Value effect
  - Momentum effect
  - Share buybacks
  - Betting ahainst beta (BAB)
  - ... and any other of the 300+(!) data mined anomalies in the market. See Harvey et al. (2016) for more details

Week-end effect: Tendency for returns to be negative on Mondays whereas they are positive on the other days of the week (French (1980)).

January effect: Tendency for returns to be higher in January as compared to other months as documented by Rozeff and Kinney (1976). This effect seems to be explained by taxes (window-dressing).

Anomalies - Size Effect

Small firms tend to have a higher return on average than large firms. Estimates based on daily returns for the CRSP index (United States) between 1963 and 2008 (annualized returns).

Portfolio sorted by size	Excess return $(\mu_p - r_f)$	Total risk (σ_p)	Sharpe ratio $((\mu_p - r_f)/\sigma_p)$
1 (Smallest firms)	5.564	12.616	**0.441**
2	5.534	15.379	0.360
3	6.401	15.512	0.413
4	6.118	15.444	0.396
5	6.496	15.398	0.422
6	5.754	14.715	0.391
7	5.894	14.873	0.396
8	5.548	15.223	0.364
9	5.014	15.044	0.333
10 (Largest firms)	4.014	16.017	**0.251**

Anomalies - Value Effect

Value stocks (firms with high book-value to market-value ratio (B/M)) tend to outperform growth stocks. Estimates based on daily returns for the CRSP index (United States) between 1963 and 2008 (annualized returns).

Portfolio sorted by B/M ratio	Excess return $(\mu_p - r_f)$	Total risk (σ_p)	Sharpe ratio $((\mu_p - r_f)/\sigma_p)$
1 (Lowest B/M ratio firms)	3.120	17.719	**0.176**
2	4.450	16.129	0.276
3	4.933	15.383	0.321
4	4.932	15.361	0.321
5	4.714	15.148	0.311
6	5.665	14.417	0.393
7	6.580	14.306	0.460
8	7.168	14.805	0.484
9	8.217	15.060	0.546
10 (Highest B/M ratio firms)	8.687	16.248	**0.535**

Anomalies - Momentum Effect

Winner stocks (firms with a high return over the recent period) tend to outperform looser stocks. Estimates based on daily returns for the CRSP index (United States) between 1963 and 2008 (annualized returns). The recent period is measured from month -12 to -2.

Portfolio sorted by prior return	Excess return $(\mu_p - r_f)$	Total risk (σ_p)	Sharpe ratio $((\mu_p - r_f)/\sigma_p)$
1 (Lowest prior return firms)	-5.868	23.110	**-0.254**
2	0.605	19.046	0.032
3	3.721	16.616	0.224
4	3.805	15.878	0.240
5	2.881	15.229	0.189
6	4.407	14.865	0.296
7	4.408	14.873	0.296
8	7.474	15.232	0.491
9	6.771	16.241	0.417
10 (Highest prior return firms)	12.266	20.357	**0.603**

Testing Anomalies/Factors

One of the most commonly used statistical methodologies in empirical asset pricing in order to examine factors is portfolio analysis. Its objective is to examine the cross-sectional relation between two or more variables in order to understand the cross-sectional relations of a set of variables, for instance., the effect of size and value on returns. Here, we cover two approaches of portfolio analysis:

- Univariate portfolio analysis
- Bivariate portfolio analysis

What both approaches have in common is that they examine portfolios rather than returns!

Univariate Portfolio Analysis

We start with the most basic type of portfolio analysis: univariate analysis. It is univariate because we only have one sort variable, for instance, size (the implicit question here is whether the market cap of a stock affects its return). The univariate portfolio analysis has four steps:

1. Calculate break points according to which assign individual stocks into portfolios
2. Forming portfolios according to the break points
3. Calculate the (average) portfolio returns
4. Examine the variation in portfolio returns

Break Points

Break points are used to assign individual stocks to their corresponding portfolio according to the sort variable (variable of interest). For instance, if we are interested in the effect of size on returns, we divide all stocks into (equally-sized) quantiles according to their market cap percentile within each period. In effect, the lowest quantile will contain the smallest stocks while the largest quantile contains the largest stocks in terms of market cap.

Note: The number of break points to use is based on trade-offs between the number of stocks in each portfolio and the dispersion among the portfolios of the sort variables. Convention says 10 portfolios if the sample is sufficient.

Command

In `STATA` the relevant command for assigning observations to quantiles is xtile, in `MATLAB` quantiles can be computed using quantile, and in `Python` quantiles are obtained with pandas.DataFrame.quantile

Forming Portfolios

The second and third steps consists of forming portfolios and compute the return time series for each period. To that end, one can simply compute the average return over the individual stocks within the quantile in a given period, where the average return is either:

- equally-weighted
- value-weighted

Note: Your results should hold across weighting schemes. Equal-weighting overweighs small firms and generally amplifies effects induced by small firms.

Examining the Variation

The final step consists of examining the variation in portfolio returns in order to determine whether there is a cross-sectional relation between the sort variable and portfolio returns. We can compute various statistics for each portfolio and examine whether they systematically differ across the portfolios sorted on the variable of interest. Common statistics to examine are:

- Expected return
- Risk
- t-tests on portfolio returns ($\neq 0$)
- Alpha (performance regressions)

Bivariate Portfolio Analysis

Bivariate portfolio analysis is very similar to univariate portfolio analysis. However, in the bivariate case we have two sort variables rather than one, for instance, size and value. The steps are essentially the same:

1. Calculate break points according to which individual stocks are assigned to portfolios
2. Forming portfolios according to the break point intersections
3. Calculate the (average) portfolio returns
4. Examine the variation in portfolio returns

What differs is that we have two types of sorts which affects step 1:

- independent sort
- dependent sort

Independent Sort

In an independent sort, the break points are computed independently from one another. Put differently, one computes the break points for the first sort variable over the whole sample then the break points for the second sort variable over the whole sample, independently from the first sort variable (the order does not matter). An independent sort is appropriate if we want to understand the joint effect of two variables, here size and value, on returns as a whole, i.e., whether the return for small value firms differs from the return of large growth firms.

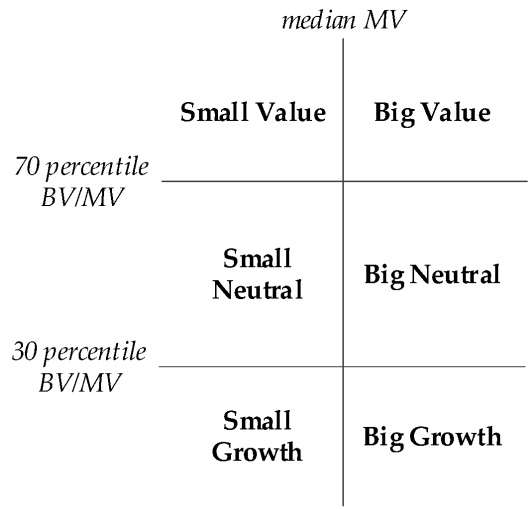
Dependent Sort

In an dependent sort, the break points for the second sort variable are formed within each group of the first sorting variable. For this reason, the dependent sort is used when the objective is to understand the relation between the second sort variable and returns conditional on the first sorting variable. For instance, if the objective is to understand the isolated or marginal effect of value on returns while controlling for the effect of size, a dependent sort is appropriate. In such a case, one starts by computing the break points for size and then computes the breakpoints for value within each size quantile.

Note: Intuitively, it's as one were to conduct a univariate analysis for the second sort variable (value) but is then afraid that the results are partially driven by the first sort variable (size). In order to partial out the contribution of the first sort variable (size) to the results, one can then conduct a bivariate dependent sort analysis to retrieve the actual contribution/effect of the second sort variable (value).

Independent Sort Portfolio

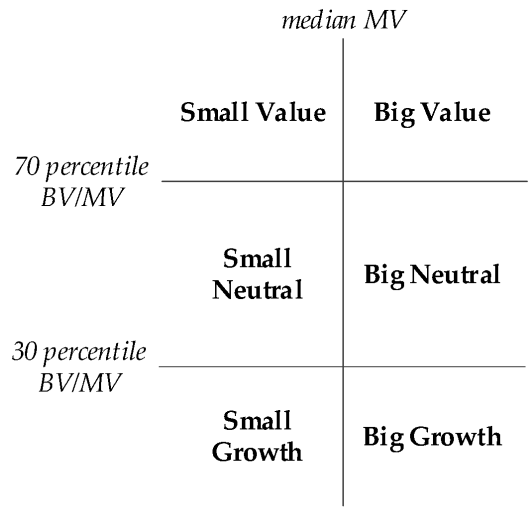
If one has P groups for the first sort variable and Q groups for the second variable, one ends up with $P \times Q$ portfolios. This is the result of forming portfolios for all possible combinations of intersection of groups. The Fama-French size-value portfolios look as follows:



In this case, the independent sort yields 6 portfolios.

Dependent Sort Portfolio

The portfolio formation under a dependent sort is similar to that of an independent sort. However, the number of portfolios one ends up with amounts to the number of groups one has for the second sort variable. If one has P groups for the first sort variable and Q groups for the second variable (estimated within the first sort variable), one ends up with Q portfolios. Again, the Fama-French size-value portfolios look as follows:



However, since one is interested in the effect of the second sort variable (value) controlling for the first sort variable (size), the resulting portfolios are averages of the portfolios across the first sort variable. For instance the value portfolio is an average of the small value portfolio and big value portfolio.

The difference portfolio

For any type of portfolio analysis, we also compute the return difference between the highest portfolio and the lowest portfolio in terms of our sorting variable. This portfolio represents the difference for entities with high values of the sort variable compared to those with low values. This is called the difference portfolio and is the primary time series used to detect a cross-sectional relation between the sort variable and returns, which is the main objective of portfolio analysis. We can then:

- use a t-test to detect whether its returns are significantly different from zero.
- use a regression to detect outperformance (see next slides).

An outcome that is statistically different from zero is evidence that a cross-sectional relation exists between the sort variable and expected returns.

Testing Anomalies/Factors

In addition to simple statistical tests, we can analyse the performance of portfolios or returns sorted on some variable using regressions and document possible cross-sectional differences. We will look at the following two:

- Calendar time: Performance regressions (Jensen's alpha)
- Event time: Cross-sectional regressions (Fama-MacBeth regressions)

Performance Regressions

This approach is straight forward. One builds portfolios based on some exposure to a factor (variable). At the beginning of each calendar period, stocks are ranked in ascending order on the basis of their estimated exposure to the factor at the end of the previous period. The ranked stocks are eventually assigned to portfolios based on breakpoints. Finally, one can run for each portfolio a performance regression against some benchmark model, such as the CAPM or Fama-French 3-Factor model:

$$r_{p,t} = \alpha_p + \beta_{1,p} MRP_t + \beta_{2,p} SMB_t + \beta_{3,p} HML_t + \varepsilon_{p,t} \tag{26}$$

If the intercept α_p, called Jensen's alpha, is significant, this additional factor partially explains the cross-section of stock returns. Furthermore, one could also examine the difference in alphas by constructing a long short portfolio.

Note: Using Newey-West adjusted standard errors is recommended.

Cross-Sectional Regressions

For event-time distance, we can run cross-sectional regressions for each period individually at the stock level and then analyse the average intercept. Specifically, for each group of stocks we run the following cross-sectional regression in each period t:

$$r_{i,\tau} = \alpha_\tau + \beta_{1,\tau} MRP_\tau + \beta_{2,\tau} SMB_\tau + \beta_{3,\tau} HML_\tau + \varepsilon_{i,\tau}, \forall \tau, \tag{27}$$

where τ is the relative time distance to the firm-specific event of firm i (IPO, earnings announcements, inlcusion to index, etc.). We can then examine the average intercept:

$$\alpha_p = \frac{1}{T} \sum_{t=1}^{T} \alpha_t \tag{28}$$

This procedure yields the average alpha of an equally-weighted portfolio. We then test the null hypothesis $H_0 : \alpha_p = 0$:

$$t(\alpha_p) = \frac{\alpha_p}{\hat{se}(\alpha_p)} \overset{a}{\sim} t_{(T-1)} \tag{29}$$

where:

$$\hat{se}(\alpha_p) = \frac{\hat{\sigma}(\alpha_p)}{\sqrt{T}} \tag{30}$$

Note: This maps exactly to second stage Fama-MacBeth regressions. ▸ Fama-MacBeth regressions
In effect, testing whether α_p is different from zero is subject to the same testing procedure as testing the CAPM with the Fama-MacBeth approach. Using Newey-West adjusted standard errors is recommended.

Factor loadings

Regressions can also be helpful in analysing the portfolio's factor loadings on the various risk premia. The following table reports long-run cumulative average abnormal returns subsequent to a repurchase announcement for under- and overpriced firms using the following cross-sectional regression each month after the event:

$$(r_{i,t} - r_{f,t}) = \alpha_j + b_j(r_{m,t} - r_{f,t}) + c_j SMB_t + d_j HML_t + e_j UMD_t + f_j RMW_t + g_j CMA_t + \varepsilon_{j,t} \quad (31)$$

	12M	24M	36M	12M	24M	36M
MRP	0.7968***	0.7994***	0.8159***	1.0464***	1.0387***	1.0332***
	(19.46)	(22.06)	(24.40)	(27.06)	(38.98)	(43.37)
SMB	0.7363***	0.7590***	0.8054***	0.2633***	0.2804***	0.3322***
	(11.09)	(15.79)	(19.97)	(4.33)	(6.65)	(8.94)
HML	0.3178**	0.3514***	0.5081***	0.1271	0.1013	0.1354**
	(2.77)	(4.66)	(7.31)	(1.48)	(1.54)	(2.50)
Momentum	-0.2999***	-0.2496***	-0.1909***	-0.2176***	-0.1784***	-0.1644***
	(-5.75)	(-5.69)	(-4.79)	(-8.14)	(-6.41)	(-5.42)
Profitability	0.1143	0.1695**	0.1403*	0.4841***	0.4237***	0.4447***
	(1.41)	(2.22)	(1.91)	(7.06)	(8.88)	(8.72)
Investment	0.3188**	0.2526***	0.0693	0.1863	0.2840***	0.2940***
	(2.50)	(2.97)	(0.85)	(1.43)	(2.81)	(3.71)
Alpha (intercept)	0.0081**	0.0062***	0.0054***	-0.0009	-0.0003	-0.0004
	(2.89)	(3.19)	(3.58)	(-0.57)	(-0.24)	(-0.49)

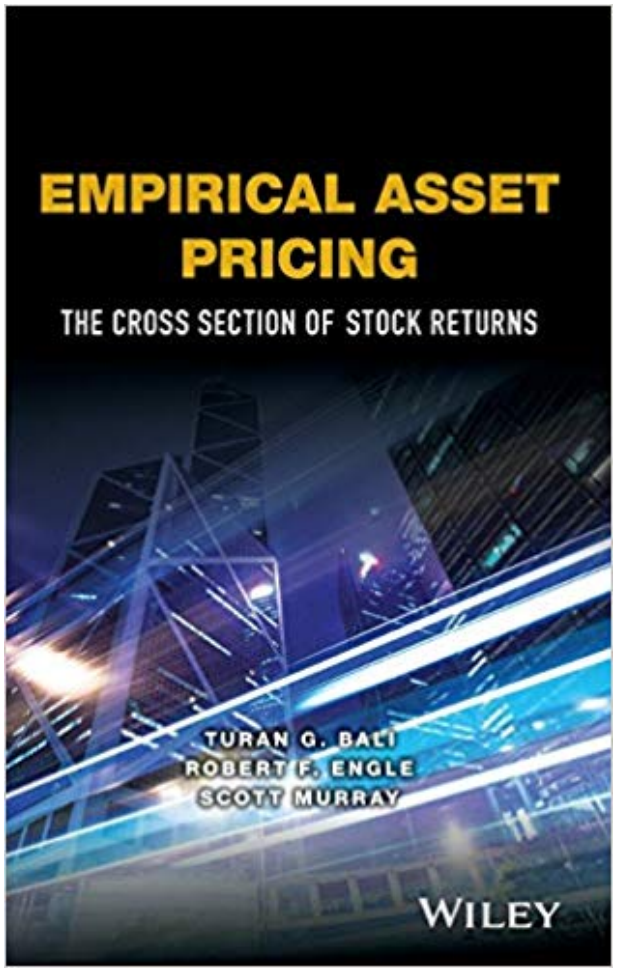
Examples

Example 1: Bali, Cakici, and Whitelaw (2011) examine the relation between "lottery stocks" and stock performance using portfolio sorts and Fama-MacBeth regressions. The results indicate a negative and significant relation between the maximum daily return over the past one month and expected stock returns.

Example 2: Frazzini and Pedersen (2014) build portfolios based on stock betas and find that the portfolio with the lowest beta stocks consistently outperforms the portfolio with highest beta stocks (This paper is up for presentation).

Reading Tip

Bali, T. G., R.F. Engle, and S. Murray (2016), Empirical Asset Pricing - The Cross Section of Stock Returns, Wiley



Practice Session - Beat the clock

If any one successfully completes the tasks before class is over, all students will receive 5 bonus points in the second assignment. Another 5 points are at stake during next weeks session. You can work together, alone (diversification!), or split into groups (optimization!). The best solution is relevant.

Using the daily SMI data on ILIAS, complete the following tasks:

1) Compute simple daily returns on the SMI.

2) Discard zero return (holidays) and missing return observations.

3) Report the annualized mean and annualized standard deviation of SMI returns assuming a trading year consists of 252 trading days.

4) Perform an augmented dickey fuller test using five lags for both the index and the return series and decide whether the series are stationary.

5) Plot the ACF and PACF with 20 lags for both the index and the return series.

As time passes, I will display code snippets that should help dealing with certain aspects of the tasks!

CAPM & Multi-Factor Models

CAPM & Factor Models

The objectives of this chapter are the following:

- Reviewing the Capital Asset Pricing Model
- Testing the CAPM using different approaches
- Investigate some econometric issues
- Multi-factor models
- Principal Component Analysis (PCA)

Review of the CAPM

The CAPM is a natural way to model asset returns:

- The expected return of a given asset depends on the expected return of the market portfolio, with a regression parameter (the beta) constant over time.

The CAPM is based on Markowitz (1952)'s analysis of the investor's portfolio selection problem:

- Investors will optimally hold a mean-variance efficient portfolio, i.e. a portfolio with the highest expected return for a given level of variance.

Sharpe (1964) and Lintner (1965) develop equilibrium implications:

- If investors have homogeneous expectations and optimally hold mean-variance efficient portfolios, then all investors should hold the same portfolio (market portfolio) and this portfolio of all invested wealth will itself be a mean-variance efficient portfolio.

The usual CAPM equation is a direct implication of the mean-variance efficiency of the market portfolio.

Review of the CAPM

The Capital Asset Pricing Model

The standard (Sharpe-Lintner) version of the CAPM is written as:

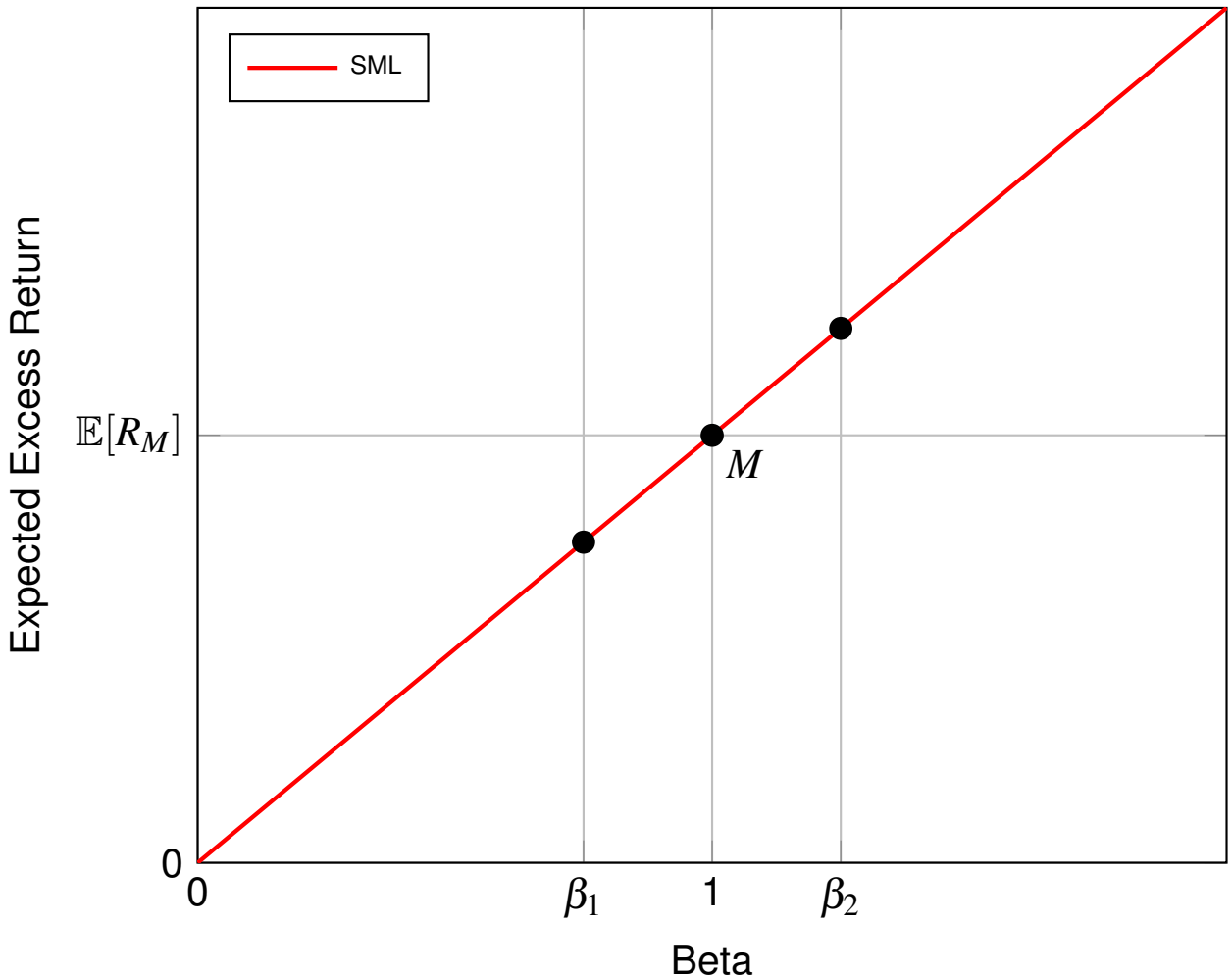
$$\mathbb{E}[r_{i,t}] = r_{f,t} + \beta_i(\mathbb{E}[r_{m,t} - r_{f,t}]) \tag{32}$$

or in terms of excess returns:

$$\begin{aligned} \mathbb{E}[r_{i,t}] - r_{f,t} &= \beta_i(\mathbb{E}[r_{m,t} - r_{f,t}]) \\ \mathbb{E}[z_{i,t}] &= \beta_i(\mathbb{E}[z_{m,t}]) \end{aligned} \tag{33}$$

The risk premium on asset i, $\mathbb{E}[z_{i,t}]$, only depends on its β_i , with a proportionality factor equal to the risk premium on the portfolio market (MRP), $\mathbb{E}[z_{m,t}]$. The proportionality factor is the same for all assets!

CAPM: Security Market Line



Tests of the CAPM

An important question is whether the CAPM is successful in pricing assets. Empirical tests of the Sharpe-Lintner CAPM have focused on three implications:

- The intercept is zero.
- Beta completely captures the cross-sectional variation of expected excess returns.
- The market risk premium $\mathbb{E}[z_{m,t}]$ is positive.

Univariate Time Series Tests

A first strategy to test the CAPM is based on the univariate time-series regression of Black et al. (1972):

$$z_{it} = \alpha_i + \beta_i z_{m,t} + \varepsilon_{i,t}, \qquad \forall t \tag{34}$$

for each asset separately, where:

- z are excess returns, i.e., $r - r_f$
- the time-series behaviour of the error term, $\varepsilon_{i,t}$, is assumed to be i.i.d.

We first focus on the first implication of the model, i.e. that the intercept is zero, or to be more precise, that it is not significantly different from zero.

Univariate Time Series Tests

A first test of the CAPM may be based on the null hypothesis $H_0 : \alpha_i = 0$ for all assets using a standard t-statistic:

$$t[\hat{\alpha}_i] = \frac{\hat{\alpha}_i}{\sqrt{\hat{\mathbb{V}}[\hat{\alpha}_i]}} \tag{35}$$

with:

$$\mathbb{V}[\hat{\alpha}_i] = \frac{1}{T}\left(1 + \frac{\mathbb{E}[z_m]^2}{\sigma_{z_m}^2}\right)\hat{\sigma}_{\varepsilon_i}^2 = \frac{1}{T}\left(1 + SR_m^2\right)\hat{\sigma}_{\varepsilon_i}^2 \tag{36}$$

Under the null hypothesis, $t[\hat{\alpha}_i]$ is distributed as a $t(T-2)$. An equivalent test is the chi-square test based on:

$$\frac{\alpha_i^2}{\mathbb{V}[\hat{\alpha}_i]} = \frac{\alpha_i^2}{(1 + SR_m^2)\sigma_{\varepsilon_i}^2/T} \tag{37}$$

which is distributed as a $\chi^2(1)$.

Multivariate Time Series Tests

The CAPM is actually more restrictive: all the α_i should be jointly equal to zero. The null hypothesis is therefore $H_0 : \alpha_1 = \ldots = \alpha_N = 0$, while the alternative hypothesis is $H_a : \exists i$ so that $\alpha_i \neq 0$. Testing H_0 involves the joint distribution of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)'$.

The test of $H_0 : \boldsymbol{\alpha} = 0$ is the multivariate form of equation (37):

$$\hat{\boldsymbol{\alpha}}' (\mathbb{V}[\hat{\boldsymbol{\alpha}}])^{-1} \hat{\boldsymbol{\alpha}} \sim \chi^2(N), \tag{38}$$

where N is the number of assets and $\mathbb{V}[\hat{\boldsymbol{\alpha}}]$ is the $N \times N$ covariance matrix of $\hat{\boldsymbol{\alpha}}$. This maps to a basic Wald test.

Command

In `STATA` the relevant command for *basic* Wald tests is test, in `MATLAB` it is waldtest, and in `Python` it is statsmodels.regression.linear_model.RegressionResults.wald_test

Multivariate Time Series Tests

It turns out that, since the regressor $z_{m,t}$ is the same in all regressions, the covariance between $\hat{\alpha}_i$ and $\hat{\alpha}_j$ is:

$$\mathbb{COV}[\hat{\alpha}_i, \hat{\alpha}_j] = \frac{1}{T}\left(1 + \frac{\mathbb{E}[z_m]^2}{\sigma_{z_m}^2}\right)\mathbb{COV}[\varepsilon_{i,t}, \varepsilon_{j,t}] \tag{39}$$

such that the covariance matrix of $\hat{\boldsymbol{\alpha}}$ maps to:

$$\mathbb{V}[\hat{\boldsymbol{\alpha}}] = \frac{1}{T}\left(1 + \frac{\mathbb{E}[z_m]^2}{\sigma_{z_m}^2}\right)\boldsymbol{\Sigma} \tag{40}$$

where $\boldsymbol{\Sigma} = \mathbb{E}[\varepsilon_t \varepsilon']$, i.e., the covariance matrix of the residuals from the regression in equation (34).

Multivariate Time Series Tests

In general, $\mathbb{V}[\hat{\alpha}]$ is not known and hence, this expression cannot be used directly. We have to preliminary estimate the covariance matrix $\hat{\mathbf{\Sigma}} = \frac{1}{T}\sum_{t=1}^{T}\hat{\varepsilon}_t'\hat{\varepsilon}_t$ and then compute:

$$\hat{\mathbb{V}}[\hat{\boldsymbol{\alpha}}] = \frac{1}{T}\left(1 + \frac{\mathbb{E}[z_m]^2}{\sigma_{z_m}^2}\right)\hat{\mathbf{\Sigma}} \tag{41}$$

Because $\hat{\mathbf{\Sigma}}$ has been preliminarily estimated, the χ^2 distribution only holds asymptotically under the null hypothesis. We then have the following Wald test statistic:

$$J_0 = T\left(1 + \frac{\mathbb{E}[z_m]^2}{\sigma_{z_m}^2}\right)^{-1}\hat{\boldsymbol{\alpha}}'\hat{\mathbf{\Sigma}}^{-1}\hat{\boldsymbol{\alpha}} = T(1+SR_m^2)^{-1}\hat{\boldsymbol{\alpha}}'\hat{\mathbf{\Sigma}}^{-1}\hat{\boldsymbol{\alpha}} \overset{a}{\sim} \chi^2(N) \tag{42}$$

The Wald test is based only on the estimators from an unconstrained model, i.e. the intercepts are not restricted to zero. We may also consider a Likelihood Ratio test, which is based on the estimators of the constrained model, i.e. the intercepts are forced to be zero. The LR test would then test for a significant difference in the covariance matrix of the residuals of the restricted and unrestricted models.

Note: A multivariate version of the Wald test for multiple factors will be discussed later:

▸ Multivariate Wald Test

The GRS Test

The normal version of the Wald test does not recognize sampling variation in $\hat{\boldsymbol{\Sigma}}$. The Gibbons, Ross, and Shanken (1989) or GRS test statistic can account for that and takes the form:

$$J_{GRS} = \frac{T-N-1}{N}(1+SR_m^2)^{-1}\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\alpha}} \overset{a}{\sim} F(N, T-N-1) \tag{43}$$

as opposed to:

$$J_0 = T(1+SR_m^2)^{-1}\hat{\boldsymbol{\alpha}}'\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\alpha}} \overset{a}{\sim} \chi^2(N) \tag{44}$$

The GRS test is generally preferable.

The GRS Test

Period	J_{GRS}	*p-value*
10-year samples		
1/65-12/74	2.400	*(0.013)*
1/75-12/84	2.248	*(0.020)*
1/85-12/94	1.900	*(0.053)*
30-year sample		
1/65-12/94	57.690	*(0.001)*

Source: Campbell, Lo and McKinlay (1997, chapter 5) using US data from 1965 to 1994.

Cross-Sectional Tests

A second approach for testing the Sharpe-Lintner version of the CAPM is based on a two-stage procedure

The first-pass time series regression consists of estimating the β_i for each asset using the time-series dimension. Under the assumption that β_i is constant over time, the first-pass regression is, for each asset separately:

$$z_{i,t} = \alpha_i + \beta_i z_{m,t} + \varepsilon_{i,t} \tag{45}$$

and yields $\hat{\beta}_i$ as the estimate for β_i. We also expect $\alpha_i = 0$ for each asset i.

Cross-Sectional Tests

The second-pass cross-sectional regression consists of regressing excess returns on betas. For each asset i, we compute the sample average excess return $\bar{z}_i = \bar{r}_i - \bar{r}_f$. Then, we estimate the following cross-sectional regression:

$$\bar{z}_i = \psi_0 + \psi_1 \hat{\beta}_i + v_i, \forall i \tag{46}$$

where $\hat{\beta}_i$ is the estimate obtained from the first-pass regression. If the CAPM is valid, then $\psi_0 = 0$, i.e. the intercept is zero, and $\psi_1 = \bar{z}_i = \bar{r}_i - \bar{r}_f$, the sample average market portfolio excess return. For instance, the test of the null hypothesis $H_0 : \psi_0 = 0$ can be based on the t-statistic:

$$t(\hat{\psi}_0) = \frac{\hat{\psi}_0}{\hat{se}_{\hat{\psi}_0}} \tag{47}$$

This expression does not take into account that the regressor $\hat{\beta}_i$ has been preliminarily estimated. This introduces a bias in the estimation of the variance of $\hat{\psi}_0$ and $\hat{\psi}_1$.

Cross-Sectional Tests

Cross-sectional results in Black et al. (1972):

Summary of Cross-sectional Regression Coefficients and Their t Values

	Time Period				
	Total Period	*Subperiods*			
	1/31-12/65	1/31-9/39	10/39-6/48	7/48-3/57	4/57-12/65
$\hat{\gamma}_0$	0.00359	-0.00801	0.00439	0.00777	0.01020
$\hat{\gamma}_1$	0.0108	0.0304	0.0107	0.0033	-0.0012
$\gamma_1 = \overline{R}_M$	0.0142	0.0220	0.0149	0.0112	0.0088
$t(\hat{\gamma}_0)$	6.52	-4.45	3.20	7.40	18.89
$t(\gamma_1 - \hat{\gamma}_1)$	6.53	-4.91	3.23	7.98	19.61

Error-in-Variables in Beta

Another econometric problem is that, in the first-pass time-series regression, the estimate $\hat{\beta}_i$ may be unbiased but it is measured with error. Hence, in the second-pass regression, there is a classic error in variables problem, which means that the OLS estimate of ψ_1 is downward biased and the estimate of ψ_0 is upward biased.

- The betas used in the cross-section regression are $\hat{\beta}_i = \beta_i + u_i$
- The relation estimated is of the form $\bar{z}_i = \psi_0 + \psi_1 \hat{\beta}_i + v_i$
- While the true relation is $\bar{z}_i = \psi_0^* + \psi_1^* \beta_i + v_i^*$
- The bias amounts to:

$$\psi_1 = \frac{\psi_1^*}{1 + \frac{\mathbb{VAR}[u_i]}{\mathbb{VAR}[\beta_i]}} \leq \psi_1^*$$

Fama and MacBeth (1973) have proposed to group stocks into portfolios to minimize the error in measurement in betas. In this regard, the best portfolios are those formed on the basis of past betas.

Fama-MacBeth Procedure

The basic idea of Fama and MacBeth (1973) is, for each cross section, to project the returns on the betas and then aggregate the estimates in the time-series dimension. The estimation is performed in three steps:

1. The first step consists of estimating betas with time-series regressions for each stock using rolling regressions for S subsamples, e.g., yearly:

- $r_{i,t} = \alpha_i + \beta_i r_{m,t} + \varepsilon_{i,t}$ within each of the S subsamples for every stock i, where r_i and r_m are excess returns.
- N portfolios are formed on the basis of ranked $\hat{\beta}_i$ of individual stocks in every subsample S, e.g. deciles.
- For each portfolio p you get S betas in total (here one beta for every sample year, S, for each portfolio p). The portfolio beta is a weighted beta of its constituents.

2. Then, we run a cross-sectional regression for each subset S (on portfolio level):

- $\bar{r}_{p,s} = \psi_{0,s} + \psi_{1,s}\hat{\beta}_{p,s} + u_{p,s}, \forall S$
- where $\hat{\beta}_{p,s} = K^{-1}\sum_{i=1}^{K}\hat{\beta}_{i,s}$ denotes the equally weighted portfolio beta of portfolio p (of K stocks) and $\bar{r}_{p,s}$ is the average portfolio excess return in susbample (year) S of portfolio p.

Fama-MacBeth Procedure

3. Then we estimate $\hat{\psi}_0$ and $\hat{\psi}_1$ as the average of the S cross-sectional regressions in step 2:

- $\hat{\psi}_0 = S^{-1} \sum_{s=1}^{S} \hat{\psi}_{0,S}$ and $\hat{\psi}_1 = S^{-1} \sum_{s=1}^{S} \hat{\psi}_{1,s}$

It turns out that the standard error of the cross-sectional average is simply the standard deviation of the cross-sectional estimates divided by the square root of S:

- $\hat{se}(\hat{\psi}_0) = \frac{\hat{\sigma}(\hat{\psi}_0)}{\sqrt{S}}$ and $se(\hat{\psi}_1) = \frac{\hat{\sigma}(\hat{\psi}_1)}{\sqrt{S}}$

The test of the null hypothesis $H_0 : \psi_i = x$ is based on the t-statistic:

$$t(\hat{\psi}_i) = \frac{\hat{\psi}_i - x}{\hat{se}(\hat{\psi}_i)} \overset{a}{\sim} t_{(S-1)} \tag{48}$$

One problem often found in time series data is that the error terms might be correlated over time. To that end, one is better off using Newey–West standard errors (Newey and West (1987)) for time series coefficients estimated by OLS regression, similar to robust standard errors to account for heteroskedasticity in panel data.

Fama-MacBeth Procedure

An even stronger implication of the CAPM is that only β_p should explain $r_{p,t}$. To that end, Fama and MacBeth (1973) also consider the following second-pass cross-sectional regression:

$$r_{p,t} = \psi_{0,t} + \psi_{1,t}\hat{\beta}_{p,t} + \psi_{2,t}\hat{\beta}^2_{p,t} + \psi_{3,t}\hat{s}_{p,t}(\hat{\varepsilon}_i) + u_{p,t}, \tag{49}$$

where $\hat{s}_{p,t}(\hat{\varepsilon}_i)$ denotes the portfolio's idiosyncratic risk in period t. This gives raise to the following testable implications:

- If $\psi_2 \neq 0$, then there are non-linearities in the CAPM relation.
- If $\psi_3 \neq 0$, then non-systematic risk affects the expected portfolio return.

Note: Myriad papers use Fama-MacBeth regressions. What they actually mean is that they adopt the second stage. i.e., period-by-period cross-sectional regressions and then average coefficients. This is also often used to evaluate the performance of portfolios: ▶ Potfolio analysis

Command

In `STATA` the second stage of the Fama-MacBeth procedure is facilitated via ssc install xtfmb, including an option to implement Newey-West corrected standard errors, and in `Python` the second stage procedure is estimated using linearmodels.panel.model.FamaMacBeth

Empirical Issues of the CAPM

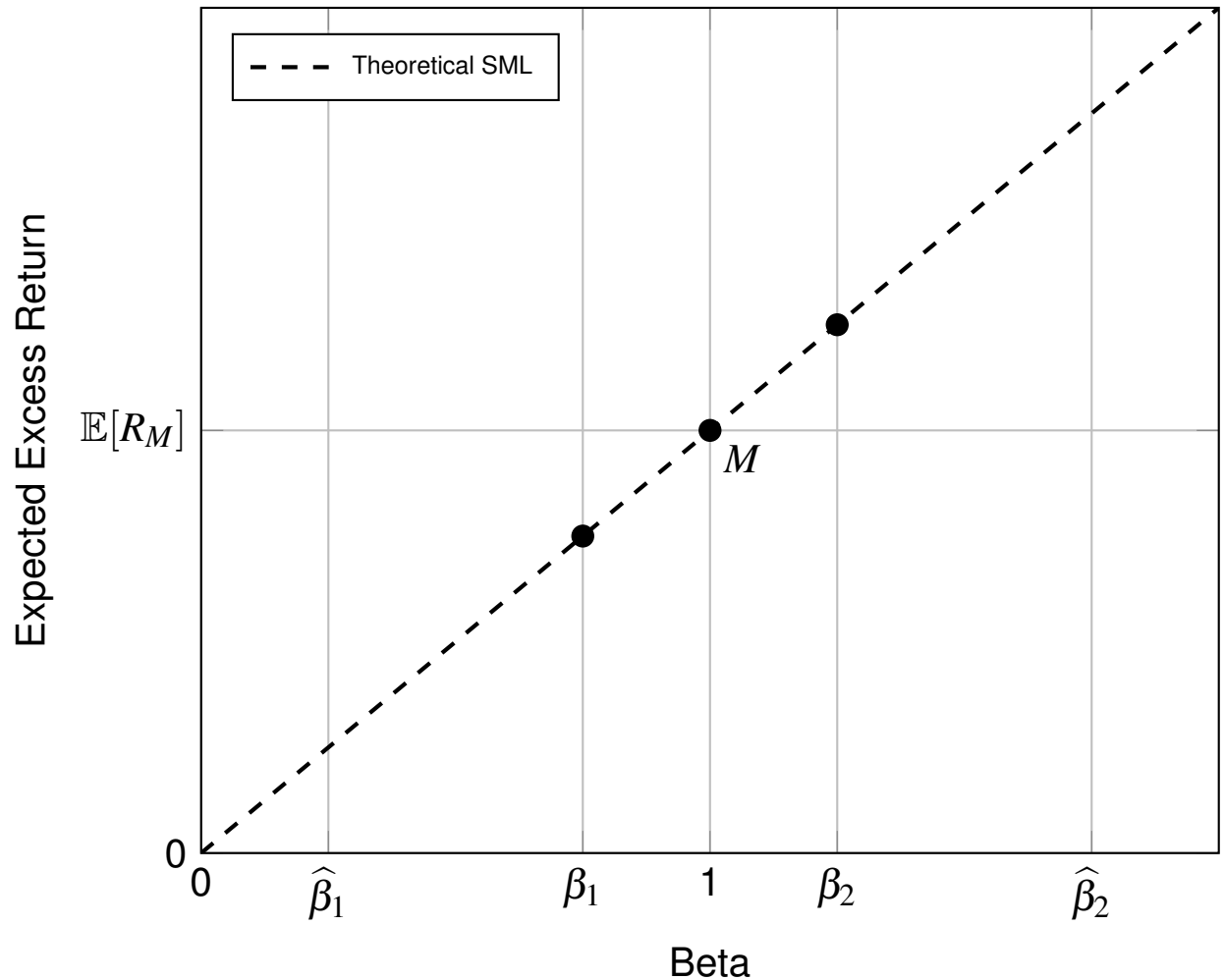
The CAPM is an empirical failure (Black et al. (1972) and the literature that follows):

- The econometrician does not observe the market portfolio (Roll (1977)), the so-called "Roll Critique"
- Suppose we could measure β_n on each individual asset n without error (this is actually a major practical issue). Then we could regress betas cross-sectionally onto unconditional expected excess returns:

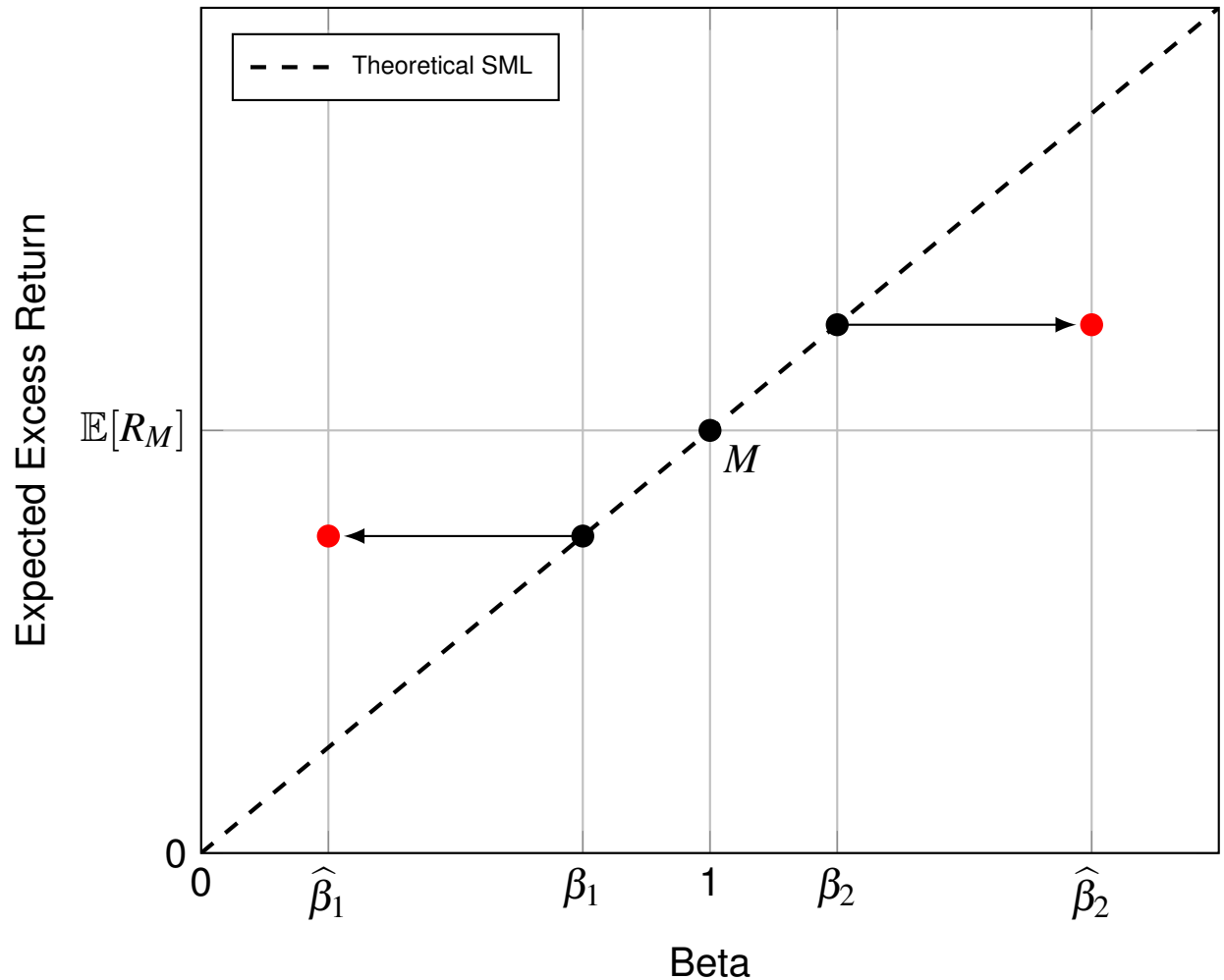
$$\bar{z}_i = \psi_0 + \psi_1 \hat{\beta}_i + v_i, \forall i, \quad n = 1, ..., N \tag{50}$$

- The CAPM directly implies that the intercept of this regression is zero, $\psi_0 \equiv 0$, and the slope is the market risk premium, $\psi_1 = \mathbb{E}[r_{m,t} - r_{f,t}]$.
- When running this regression empirically, we typically find that ψ_0 is statistically different from zero and the CAPM is way too flat (its slope is much less than $\approx 6\%$).
- Beta simply cannot explain variations in expected returns.

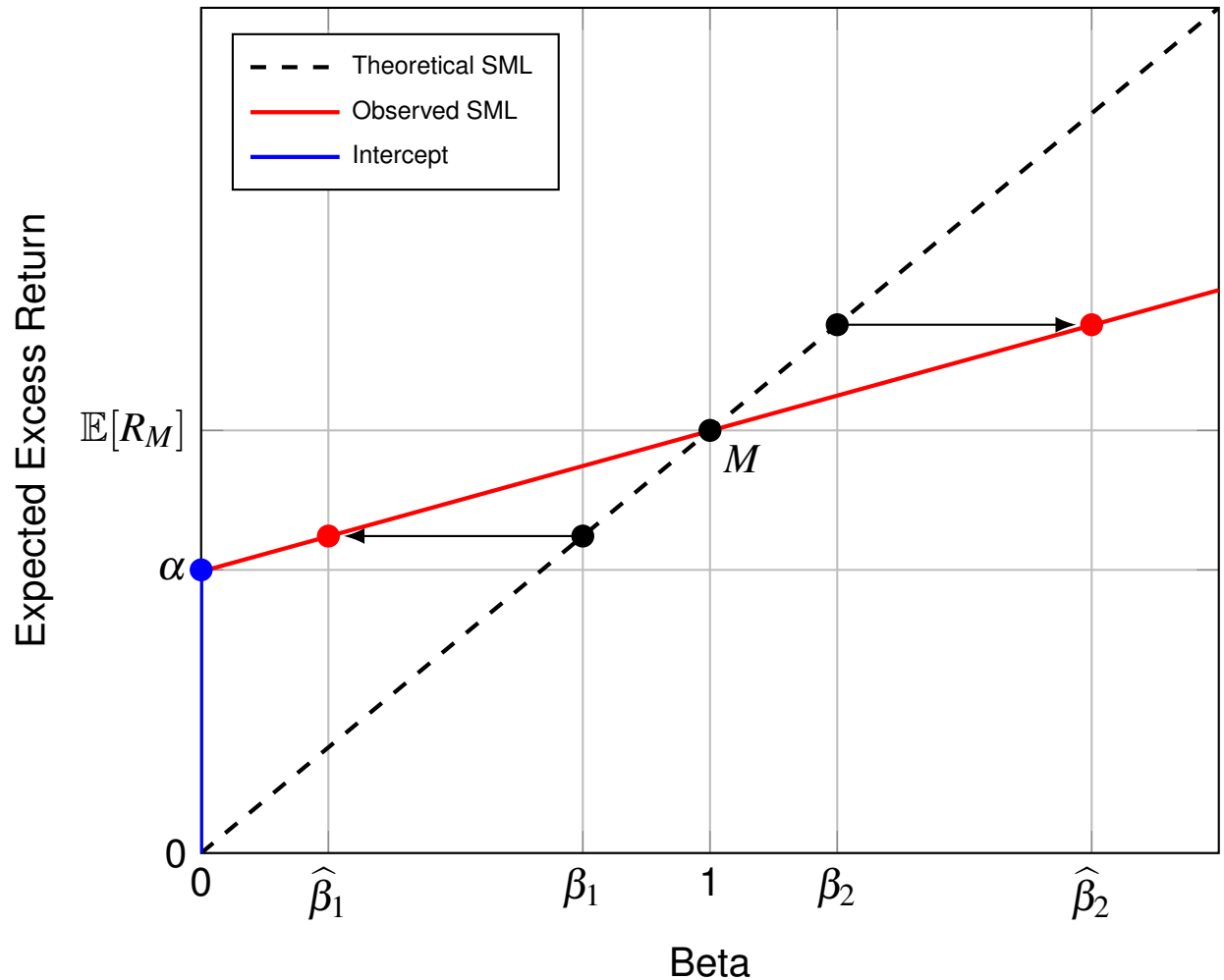
Empirical Issues of the CAPM



Empirical Issues of the CAPM



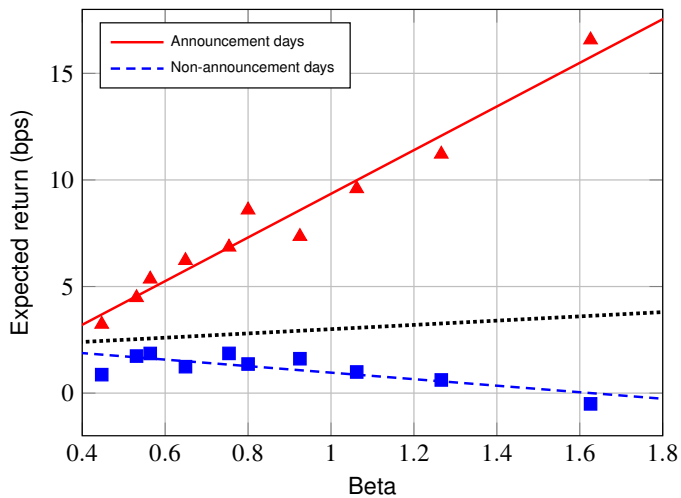
Empirical Issues of the CAPM



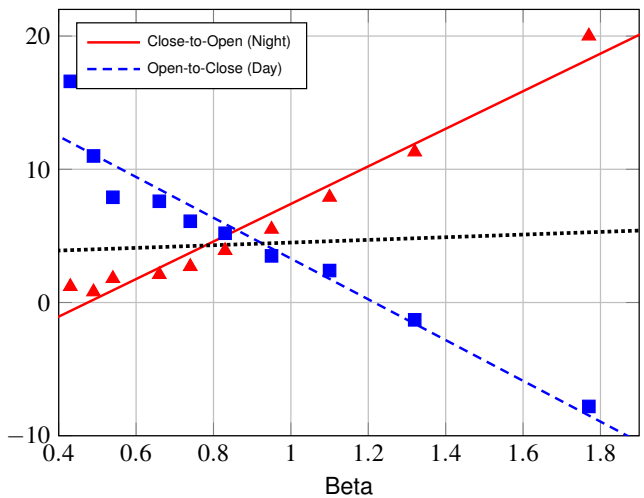
Empirical Issues of the CAPM

- Yet the CAPM remains to this day the model most widely used by practitioners to make investment decisions and compute the cost of capital.
- Furthermore, the CAPM does sometimes work:

Savor and Wilson (2014)



Hendershott et al. (2018)



Multi-Factor Models

To summarize, the CAPM (or single-factor model) is often rejected by the data, because of its overly simplistic implications, first and foremost:

- The market beta is enough to capture all the dispersion in expected returns across assets.

Multi-factor models try to model the various types of systematic risk on top of the market risk and how these risks are priced.

Multi-Factor Models

The basic idea of multi-factor models is that the expected return of the investor for holding a given stock is a linear combination of the various sources of risk of the stock times the prices of the respective risk:

$$\mathbb{E}[r_{i,t}] = \mu_i = \lambda_0 + \lambda_1 \beta_{1,i} + \ldots + \lambda_K \beta_{K,i} = \lambda_0 + \sum_{k=1}^{K} \lambda_k \beta_{k,i} \tag{51}$$

or in matrix notation:

$$\mathbb{E}[\boldsymbol{r}_t] = \boldsymbol{\mu} = \mathbf{1}\lambda_0 + \boldsymbol{B}\boldsymbol{\lambda} \tag{52}$$

where λ_0 is the risk-free rate, λ_k is the risk premium on risk factor k, $\mathbf{1}$ is a vector of ones, and $\boldsymbol{\lambda}$ is the vector of the K risk premia. We generally distinguish between three different types of factors:

- Factors are observed portfolios of traded assets
- Statistical factors
- Factors are observed macroeconomic variables but not traded assets

Factors are observed portfolios

N excess returns are linearly related to K factors such that we have:

$$\boldsymbol{Z}_t = \boldsymbol{\alpha} + \boldsymbol{B}\boldsymbol{F}_{K,t} + \boldsymbol{\varepsilon}_t \tag{53}$$

where $\boldsymbol{B}$ is a $(N \times K)$ sensitivity matrix and with:

- $\mathbb{E}[\boldsymbol{\varepsilon}_t] = 0$
- $\mathbb{E}[\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t'] = \boldsymbol{\Sigma}$
- $\mathbb{E}[\boldsymbol{F}_{K,t}] = \boldsymbol{\mu}_K$
- $\mathbb{E}[(\boldsymbol{F}_{K,t} - \boldsymbol{\mu}_K)'(\boldsymbol{F}_{K,t} - \boldsymbol{\mu}_K)] = \boldsymbol{\Omega}_K$
- $\mathbb{COV}[\boldsymbol{F}_{K,t}, \boldsymbol{\varepsilon}_t] = 0$

We assume that the K factors capture the entire cross covariance between asset returns so that the variance of $\boldsymbol{\varepsilon}_t$ is diagonal. Then the covariance matrix of returns is the sum of two components:

$$\boldsymbol{\Omega} = \boldsymbol{B}\boldsymbol{\Omega}_K\boldsymbol{B}' + \boldsymbol{\Sigma} \tag{54}$$

Factors are observed portfolios

Similar to testing the CAPM we can test for constrained models whether the intercepts are jointly zero, i.e., $H_0 = \alpha_1 = \ldots = \alpha_N$, where:

$$\boldsymbol{\alpha} = \boldsymbol{\mu} - \boldsymbol{B}\boldsymbol{\mu}_K \tag{55}$$

The asymptotic test of $H_0 : \boldsymbol{\alpha} = 0$ is based on the multivariate Wald test statistic:

$$J_0 = T(1 + \hat{\boldsymbol{\mu}}_K' \hat{\boldsymbol{\Omega}}_K^{-1} \hat{\boldsymbol{\mu}}_K)^{-1} \hat{\boldsymbol{\alpha}}' \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\alpha}} \overset{a}{\sim} \chi^2(N) \tag{56}$$

with:

$$\hat{\boldsymbol{\Omega}}_K = T^{-1} \sum_{t=1}^{T} (\boldsymbol{F}_{K,t} - \boldsymbol{\mu}_K)'(\boldsymbol{F}_{K,t} - \boldsymbol{\mu}_K) \tag{57}$$

The multivariate GRS test statistic takes the form:

$$J_{GRS} = \frac{T - N - K}{N} (1 + \hat{\boldsymbol{\mu}}_K' \hat{\boldsymbol{\Omega}}_K^{-1} \hat{\boldsymbol{\mu}}_K)^{-1} \hat{\boldsymbol{\alpha}}' \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\alpha}} \overset{a}{\sim} F(N, T - N - K) \tag{58}$$

The univariate Wald test can be re-examined here: ▸ Wald test

Factors are observed portfolios

Examples: Fama and French (1992) propose the seminal three-factor model:

- Expected return of the market portfolio – market risk premium (MRP)
- Difference between the return on a portfolio of firms with a low market value of equity (small cap) and the return on a portfolio of firms with a high market value of equity (large cap) – small minus big (SMB) factor
- Difference between the return on a portfolio of firms with a high book-to-market value (value stocks) and the return on a portfolio of firms with a low book-to market value (growth stocks) – high minus low (HML) factor

Another important factor is the momentum factor by Carhart (1997) : difference between the return on a portfolio of firms with a low prior return (between months $t-13$ and $t-2$) (losers) and the return on a portfolio of firms with a high prior return (between months $t-13$ and $t-2$) (winners).

Statistical Factors

The idea of the statistical approaches is to reduce the dimension of the cross-sectional dimension. We have a large number of stock returns that we try to summarize with a small number of factors.

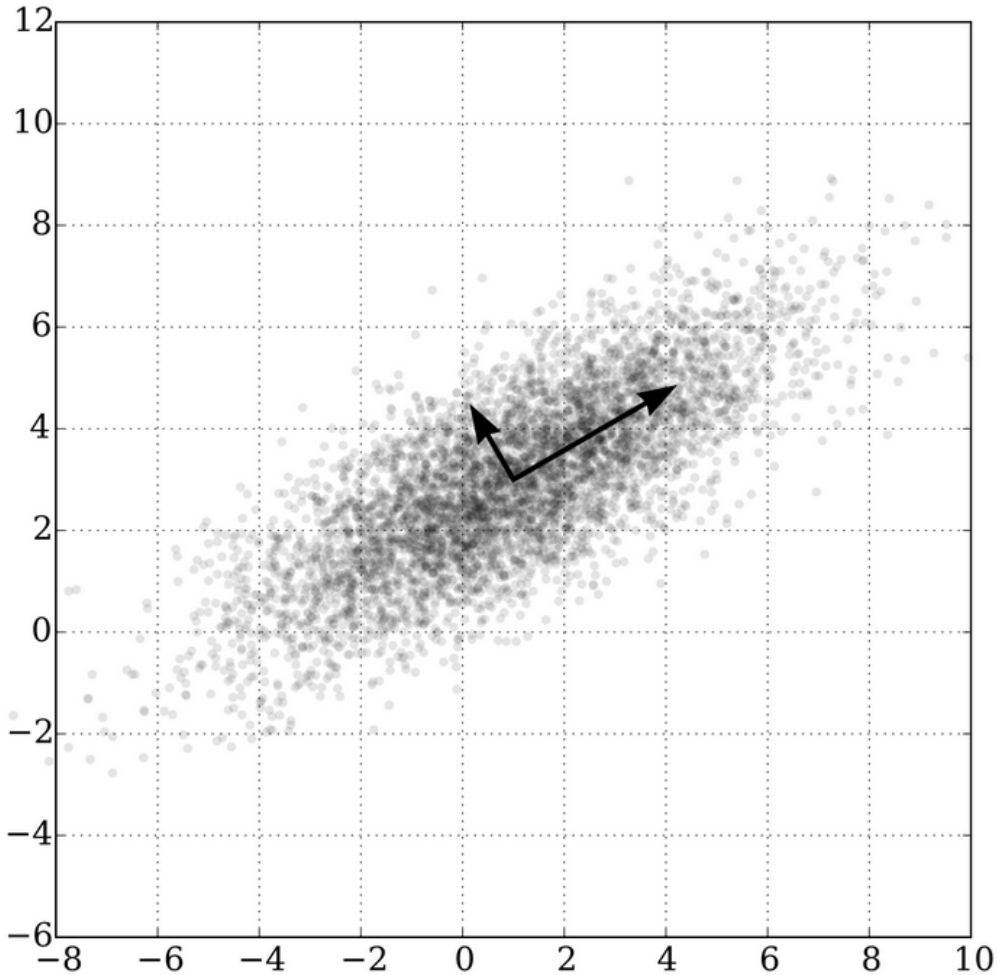
- In this approach, the factors are not known, so that we need to generate the factors. There are different ways to estimate the factors among correlated financial series.
- Factor models are also extremely useful in the context of large portfolios. When we have a lot of assets in the portfolio, a complete model for all the assets simultaneously is not possible. In this case, we summarize the information contained in all the assets through a factor model (with a small number of factors) and we model the factors instead of the assets.

Principal Component Analysis

One of the most widely used techniques is the principal component analysis (PCA). The main goal of PCA is to reduce the dimensionality of correlated data by identifying a small number of uncorrelated linear combinations that explain most of the variability in the original data.

- PCA is a data-reduction technique. It determines what the drivers of a data sample are However, it does not explain why they are the drivers of the data or what they are in the first place.
- PCA is based on the spectral decomposition of the covariance (or correlation) matrix.
- PCA is often used to summarize the information in the term structure of interest rates.

Principal Component Analysis



Principal Component Analysis

Spectral Decomposition Theorem

Any symmetric matrix $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ can be written as

$$\mathbf{\Sigma} = \mathbf{\Gamma \Lambda \Gamma}^{-1} = \mathbf{\Gamma \Lambda \Gamma}' \tag{59}$$

where

- $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues of $\mathbf{\Sigma}$. The eigenvalues are ordered so that $\lambda_1 \geq \ldots \geq \lambda_N$
- $\mathbf{\Gamma}$ is an orthogonal matrix satisfying $\mathbf{\Gamma \Gamma}' = \mathbf{\Gamma}' \mathbf{\Gamma} = \boldsymbol{I}_N$, whose columns are the standardized eigenvectors of $\mathbf{\Sigma}$ or, put differently, factor loadings.

Properties when $\mathbf{\Sigma}$ is the covariance matrix of $\boldsymbol{r}_t$:

- The first principal component explains the greatest amount of the total variation of $\boldsymbol{r}_t$, the second component explains the greatest amount of the remaining variation, and so on.
- The principal components are orthogonal to one another.

Principal Component Analysis

The dimensions of $\mathbf{\Lambda}$ are:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix}_{N \times N}$$

and the dimensions of $\mathbf{\Gamma}$ are:

$$\mathbf{\Gamma} = [\mathbf{\Gamma}_1 \quad \mathbf{\Gamma}_2 \quad \dots \quad \mathbf{\Gamma}_N] = \begin{bmatrix} \gamma_{1,1} & \gamma_{2,1} & \cdots & \gamma_{N,1} \\ \gamma_{1,2} & \gamma_{2,2} & \cdots & \gamma_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{1,N} & \gamma_{2,N} & \cdots & \gamma_{N,N} \end{bmatrix}_{N \times N}$$

Principal Component Analysis

Once we have obtained the eigenvalues $\mathbf{\Lambda}$ and the eigenvectors $\mathbf{\Gamma}$, we can deduce the principal components:

$$\boldsymbol{P}_t = \mathbf{\Gamma}'(\boldsymbol{r}_t - \boldsymbol{\mu}), \text{ with } \boldsymbol{\mu} = \mathbb{E}[\boldsymbol{r}_t] \tag{60}$$

In particular, the first principal component is defined as:

$$\boldsymbol{P}_{1,t} = \mathbf{\Gamma}_1'(\boldsymbol{r}_t - \boldsymbol{\mu}) = \gamma_{1,1}(r_{1,t} - \mu_1) + \ldots + \gamma_{1,N}(r_{N,t} - \mu_N) \tag{61}$$

It satisfies $\max\{\mathbb{VAR}[\mathbf{\Gamma}_n'(\boldsymbol{r}_t - \boldsymbol{\mu})]\}$, for $n = \{1, \ldots, N\}$

The second principal component is the standardized linear combination of $\boldsymbol{r}_t$ with maximal variance among all linear combinations that are orthogonal to the first principal component, and so on. We also have:

- $\mathbb{E}[\boldsymbol{P}_t] = 0$
- $\mathbb{COV}[\boldsymbol{P}_t] = \mathbf{\Gamma}'\mathbf{\Sigma}\mathbf{\Gamma} = \mathbf{\Gamma}'\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'\mathbf{\Gamma} = \mathbf{\Lambda}$, i.e., the principal components are uncorrelated

Principal Component Analysis

To measure the ability of the first few principal components to explain the variability in $\boldsymbol{r}_t$, we notice that:

$$\sum_{i=1}^{N} \mathbb{V}[\boldsymbol{P}_{i,t}] = \sum_{i=1}^{N} \lambda_i = trace(\boldsymbol{\Sigma}) = \sum_{i=1}^{N} \mathbb{V}[\boldsymbol{r}_{i,t}] \tag{62}$$

Then, for $K \leq N$, the ratio $\sum_{i=1}^{K} \lambda_i / \sum_{i=1}^{N} \lambda_i$ represents the amount of the variability explained by the first K principal components.

Remark: If the data $\boldsymbol{r}_t$ have very different sample variances (with different scales), applying the PCA to the covariance matrix may raise some issues, because the variables with the highest variance will dominate the first principal component. In these situations, it is recommended to standardize the data and use the correlation matrix instead.

Command

In `STATA` the relevant command for the PCA is pca or pcamat, in `MATLAB` it is pca, and in `Python` it is sklearn.decomposition.PCA

Principal Component Analysis

Finally, the principal component transform:

$$\boldsymbol{P}_t = \boldsymbol{\Gamma}'(\boldsymbol{r}_t - \boldsymbol{\mu}) \tag{63}$$

can be inverted to the following equation:

$$\boldsymbol{r}_t = \boldsymbol{\mu} + \boldsymbol{\Gamma}\boldsymbol{P}_t \tag{64}$$

If we think the first K factors are enough to explain most of the variability in $\boldsymbol{r}_t$, we obtain the following factor representation:

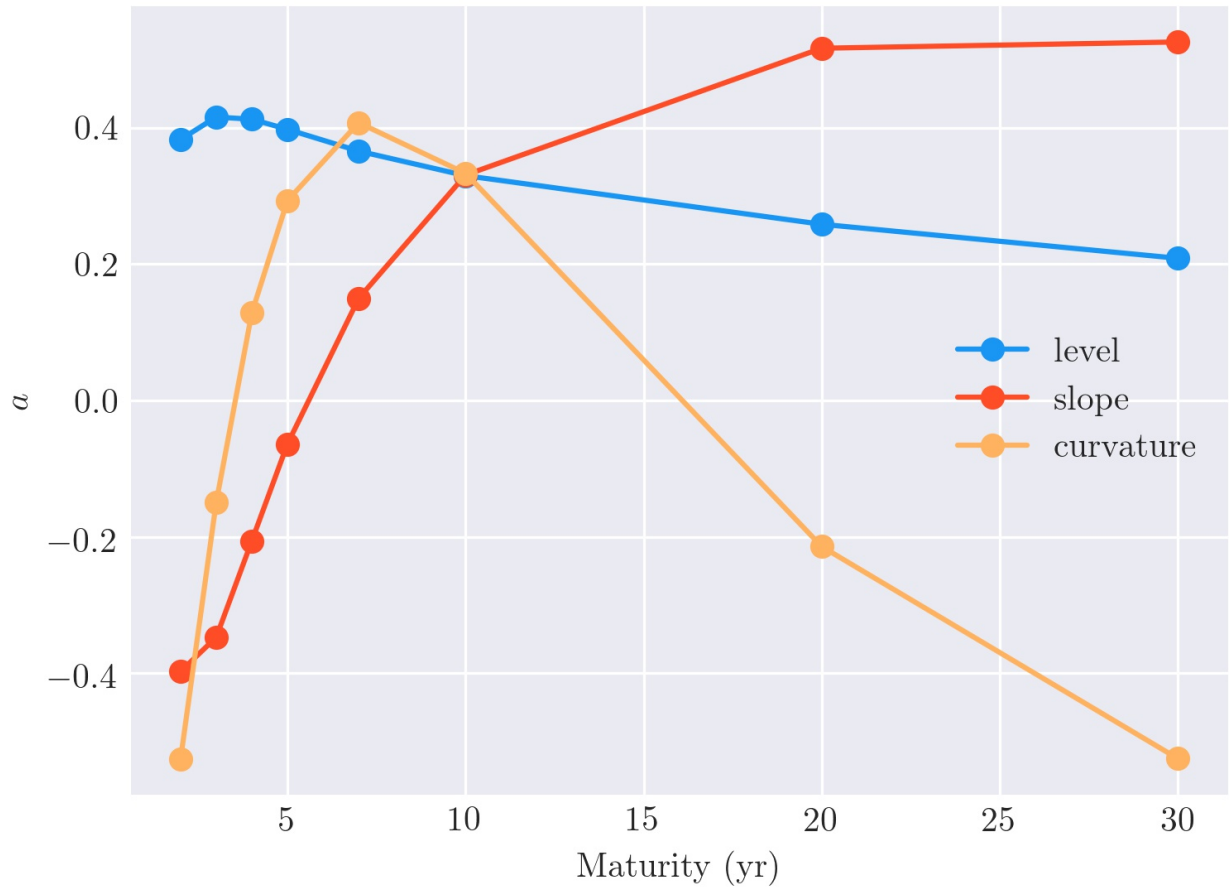
$$\boldsymbol{r}_t = \boldsymbol{\mu} + \boldsymbol{\gamma}_1 \boldsymbol{P}_{1,t} + \ldots + \boldsymbol{\gamma}_K \boldsymbol{P}_{K,t} + \boldsymbol{\varepsilon}_t \tag{65}$$

where the $\boldsymbol{\varepsilon}_t$ capture the effect of the remaining $N - K$ principal components and are orthogonal to the first K components.

Principal Component Analysis

Example 1: PCA of yield curve change

A major problem in estimating the term structure (or change thereof) is the high dimensionality of the yield curve. Using a PCA, the first three principal components map to level, slope, and curvature of yield curves (Swiss data):



Principal Component Analysis

Example 1: PCA of yield curve change

First three yield curve loadings:

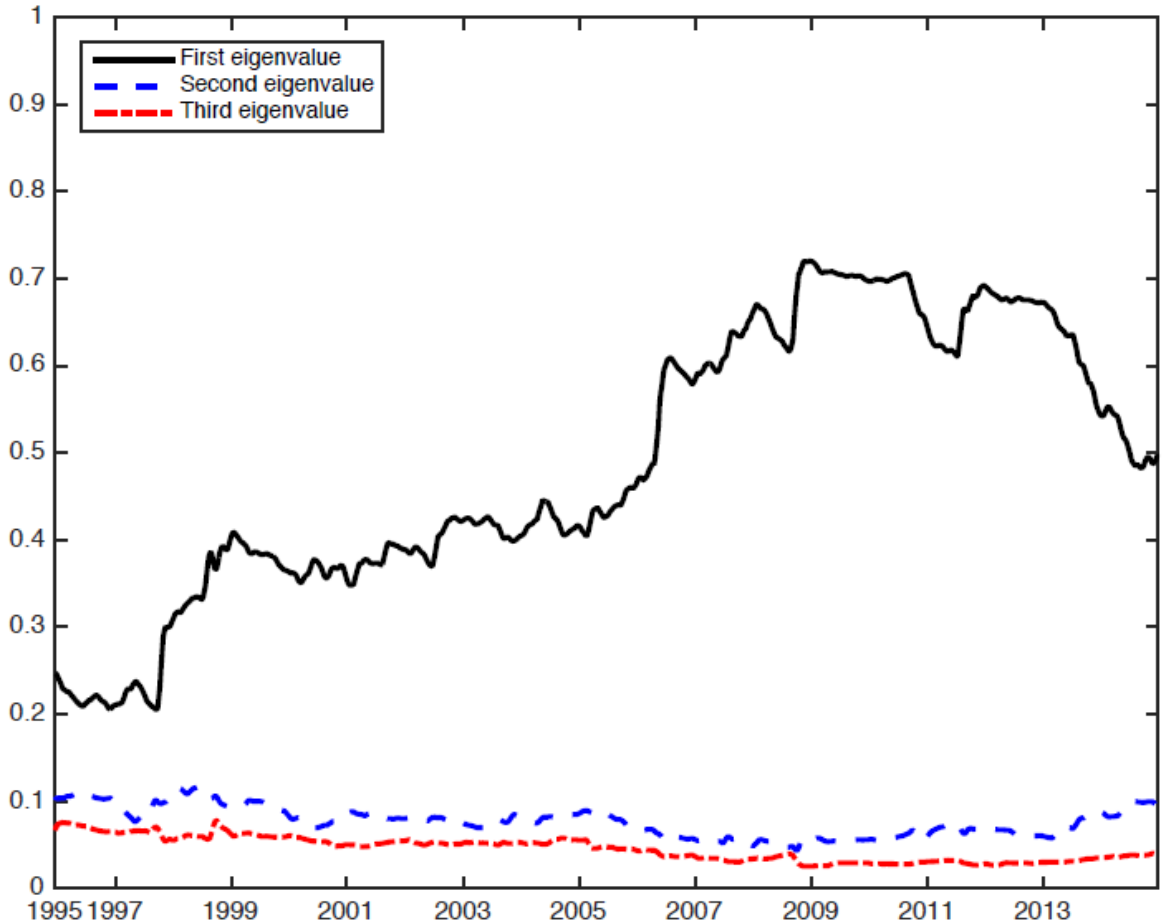
- First loading is roughly flat: parallel shifts of the yield curve (level)
- Second loading is upward sloping: tilting of the yield curve (slope)
- Third loading is hump-shaped: flexing of the yield curve (curvature)

Principal Component Analysis

Example 2: Jondeau, Jurczenko, and Rockinger (2016) conduct a PCA for 44 international stock markets between July 1994 and December 2014 (1068 weekly observations). The data are log-returns in US dollar.

Component	Raw	Cum (in %)
1	20.550	46.700
2	2.780	53.020
3	1.600	56.650
4	1.190	59.370
5	1.130	61.930
6	0.990	64.190
7	0.880	66.200
8	0.850	68.140
9	0.820	69.990
10	0.780	71.780

Principal Component Analysis

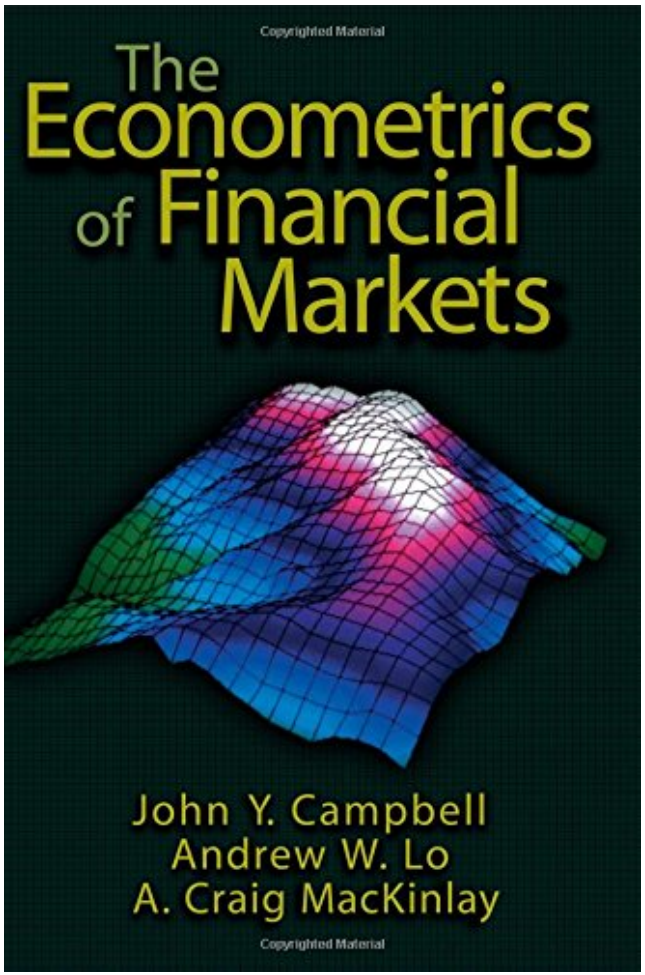


Principal Component Analysis

Example 3: Baker and Wurgler (2006) study how investor sentiment affects the cross-section of stock returns. To that end, they conduct a PCA to construct a market sentiment index to capture investors' attitude regarding the stock market. They choose a set of variables of which each is supposed to proxy for market sentiment, such as the number of IPOs, first-day returns, etc. In order to filter out and aggregate common factors underlying different sentiment variables they conduct a PCA and construct a sentiment index using the first principal component.

Reading Tip

Campbell, John Y., A. W. Lo, and A. C. MacKinlay (1997), The Econometrics of Financial Markets, Princeton



Practice Session - Beat the clock

If any one successfully completes the tasks before class is over, all students will receive 5 bonus points in the second assignment. You can work together, alone (diversification!), or split into groups (optimization!). The best solution is relevant.

Using the Swiss government bonds data on ILIAS, complete the following tasks:

1) Run a PCA on the time series of Swiss government bonds. For the entire exercise, keep in mind that Stata automatically standardizes variables in the PCA, i.e $z = \frac{r-\mu}{\sigma}$. How much variation do the first three principal components explain?

2) Plot the the factor loadings for the first three principal components (Hint: ssc install eofplot). Are the loadings (more or less) equivalent to the ones on slide 109?

3) Generate the time series for the first principal component (Hint: predict)

4) Compute the fitted time series for the 10 Year government bond using the first principal component and plot it together with the actual time series (Hint: Use equation (64) but keep in mind that factor loadings pertain to standardized variables, z.)

As time passes, I will display code snippets that should help you deal with certain aspects of the tasks!

Additional Topics

Time Series Models

Time Series Models

The objectives of this chapter are the following:

- General idea of predictive time series models
- Understanding the autoregressive model (AR)
- Understanding the moving average model (MA)
- Understanding the autoregressive moving average model (ARMA/ARIMA)
- Maximum Likelihood Estimation

General Idea of Time Series Models

So far, we have mostly dealt with the question of how to test whether asset returns fulfil the assumptions imposed by literature, how to test whether fundamental asset pricing models hold, and how to test whether markets are efficient. We will now shift the scope of the course towards the predictability of financial time series using time series models.

The general idea of time series models is that time series are governed by some underlying dynamics. To that end, time series models aim at capturing and representing these dynamics in order to make predictions about its future path. Most models require that the time series be:

- weakly (covariance) stationary
- ergodic

Autoregressive Model

A straight forward model is the autoregressive model (AR). It describes the dynamics of a time series as a function of its past realizations. For instance, if we expect that the current realization is a function of the prior realization only, i.e., one lag, we have the following first-order AR(1) process:

$$X_t = c + \phi X_{t-1} + \varepsilon_t \tag{66}$$

where c is a constant (intercept) and ε_t is white noise, i.e., an error term with zero mean and finite variance, often referred to as shock or innovation. More generally, we can write the pth order AR(p) process as follows:

$$X_t = \mu + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t \tag{67}$$

What we essentially have, is a regression of a variable on p lags of itself. Instead of OLS, one can also use maximum likelihood estimation.

Moving Average Model

Another essential representation of time series dynamics is the moving average model (MA). It represents the time series as a linear combination of contemporaneous and past realizations of shocks ε, i.e., error terms. For instance, if we expect that the current realization is a function of the current and prior realization of the shock only, i.e., one lag, we have the following first-order MA(1) process:

$$X_t = c + \varepsilon_t + \theta \varepsilon_{t-1} \tag{68}$$

The name moving average comes from the fact that X_t is a weighted average of contemporaneous and past shocks. More generally we can write the qth order MA(1) process as follows:

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{69}$$

Note that fitting the MA estimates is more complicated than it is in AR models, because the lagged error terms are not observable. This means that iterative non-linear fitting procedures need to be used in place of linear least squares.

AR(p) Model vs. MA(q) Model

Some important differences between the AR and MA models are:

- The primary difference between an AR and MA model is based on the correlation between time series observations at different points in time. The covariance between X_t and X_{t-1} is zero for MA models. However, the correlation of X_t and X_{t-1} gradually declines with p becoming larger in AR models.
- The AR(p) model uses the past forecasts to predict future values.
- The moving average model specifies that the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable) term. Rather than using the past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

Autoregressive Moving Average Model

The autoregressive moving average model (ARMA) describes weakly stationary stochastic time series in terms of two polynomials:

- An autoregressive polynomial of order p to capture autocorrelation.
- An moving average polynomial of order q to account for the persistence in shocks.

In essence, it is a model with p autoregressive terms and q moving average terms such that it maps to nothing different than a combination of an AR(p) model and a MA(q) model. In general, a ARMA(p,q) model has the following form:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \phi X_{t-i} + \sum_{i=1}^{q} \theta \varepsilon_{t-i} \tag{70}$$

An extension of the ARMA(p,q) model is the ARIMA(p,d,q) model where d specifies the order of integration I of a time series.

Command

In `STATA` the relevant command for the AR(I)MA family is arima, in `MATLAB` it is also arima, and in `Python` it is statsmodels.tsa.arima_model.ARIMA

Estimating Time Series Models

Technically, one can resort to any fitting methodology that minimizes some error term. However, convention suggests one of the following two:

- Maximum likelihood
- Least squares family, if feasible

The procedure to estimate the model parameters looks as follows:

1. Choose some (arbitrary) initial model parameters and predict the model outcome for each period in your data sample.
2. Compare the predicted values (fit) with the actual outcome in all periods t and assess the most effective parameter changes to improve your fit.
3. Change the model parameters and repeat step 2 until the fit can no longer be improved, i.e., the model converged to the optimum.

Maximum Likelihood Estimation

The MLE principle provides a means of estimating a parameter or set of parameters, $\boldsymbol{\Theta}$, by maximizing a likelihood function, so that under the assumed statistical model the observed data is most *likely*.

Maximum Likelihood Estimation

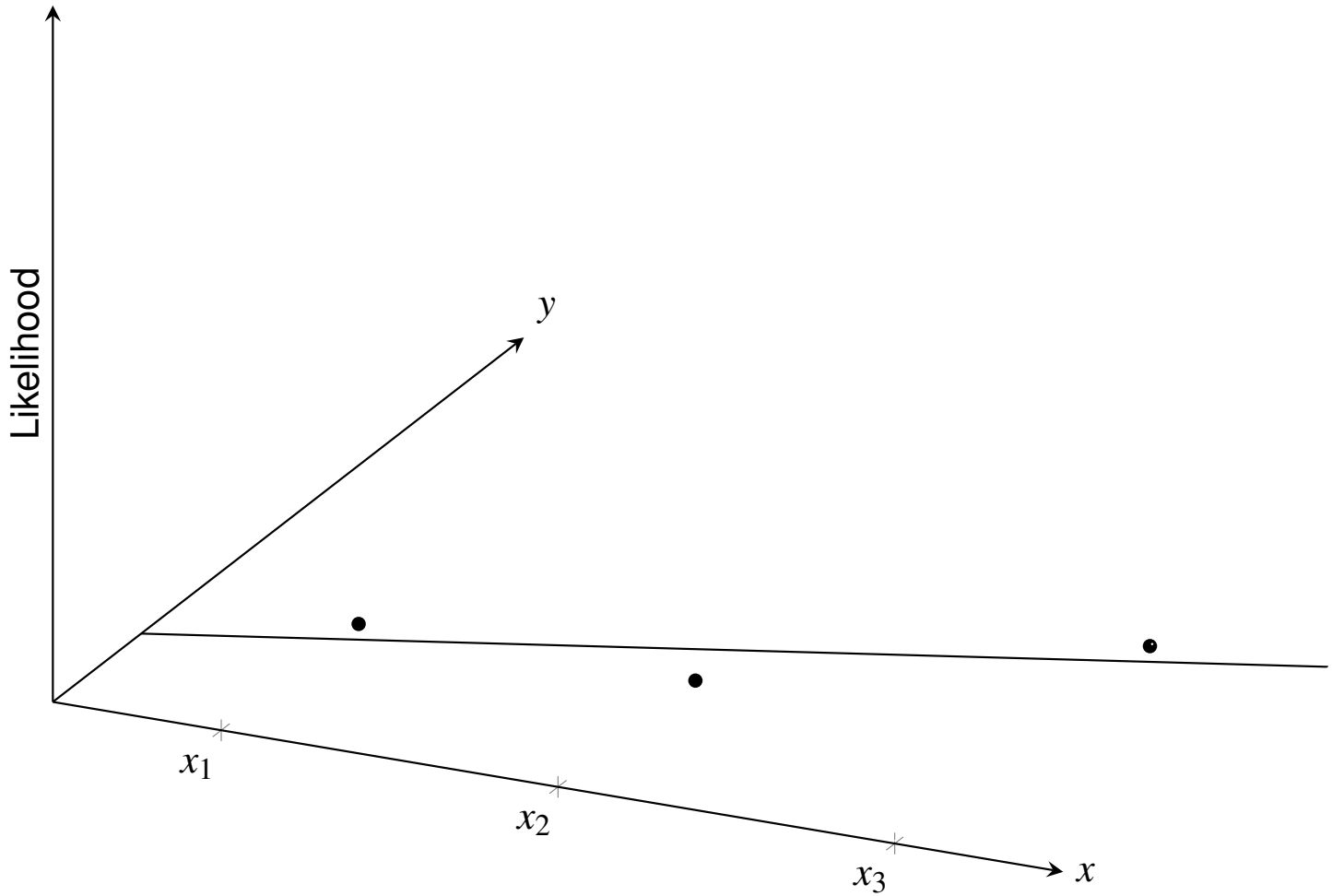
Given some PDF of error terms and observations $X_t \in \{X_1, \ldots, X_T\}$

$$L_T(\boldsymbol{\Theta}) = \prod_{t=1}^{T} f(X_t; \boldsymbol{\Theta}) \tag{71}$$

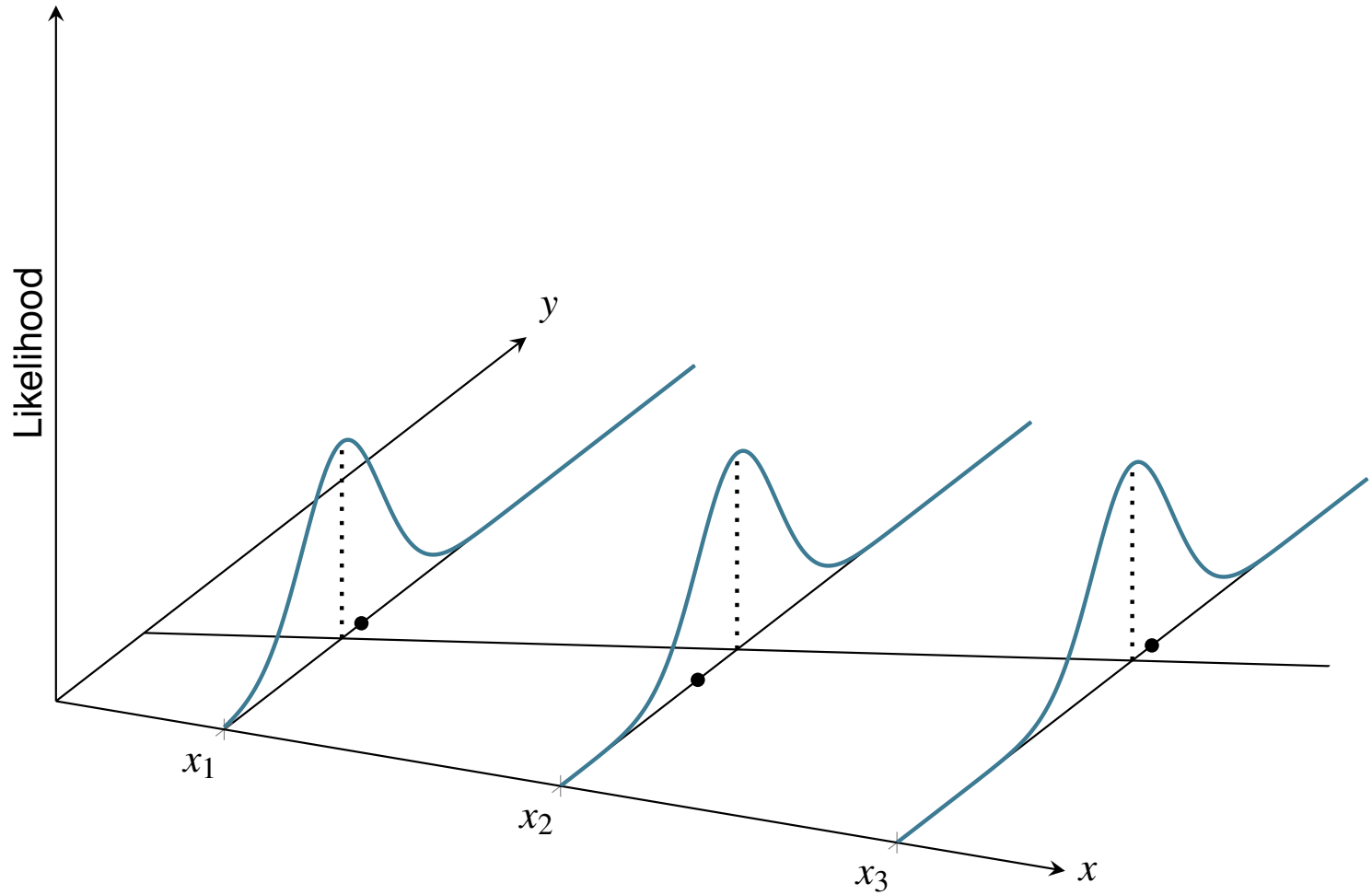
is known as the likelihood function. The likelihood equation for the maximum is

$$\nabla \ln(L_T(\boldsymbol{\Theta})) = \frac{\partial \ln(L_T(\boldsymbol{\Theta}))}{\partial \boldsymbol{\Theta}} = 0 \tag{72}$$

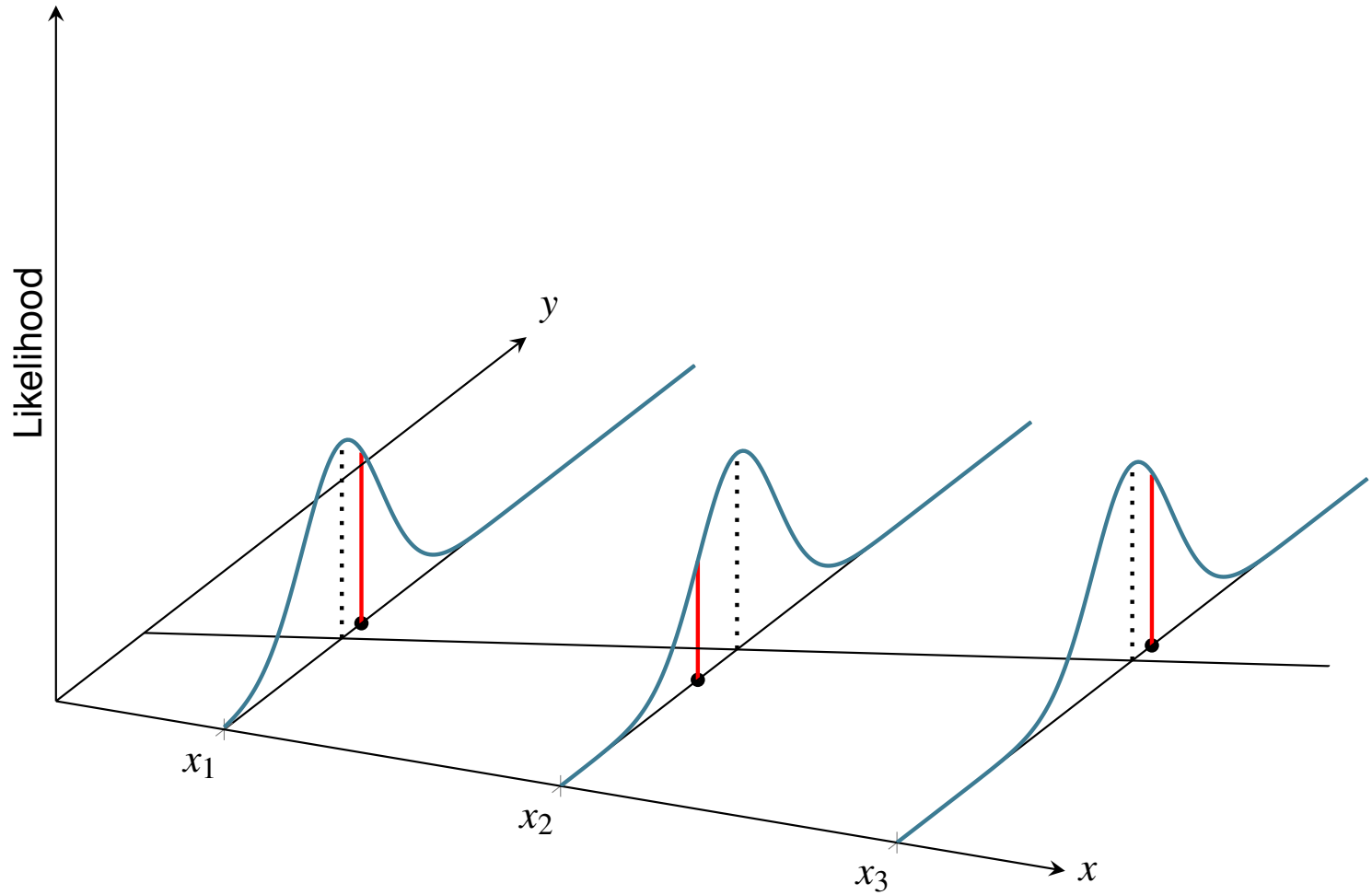
Maximum Likelihood Estimation



Maximum Likelihood Estimation



Maximum Likelihood Estimation



Maximum Likelihood Estimation

The ML estimator is obtained by maximizing the likelihood. Since logarithmic functions are injective on the domain $(0,\infty]$, this is equivalent to maximizing the log-likelihood function:

$$\underset{\boldsymbol{\Theta}}{\arg\max} \quad \ln(L_T(\boldsymbol{\Theta})) = \ell_T(\boldsymbol{\Theta}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{\Theta}) \tag{73}$$

where:

$$\ell_t(\boldsymbol{\Theta}) = -\frac{1}{2}\left(\ln(2\pi) + \ln(\sigma_t^2) + \frac{\varepsilon_t^2}{\sigma_t^2}\right) \tag{74}$$

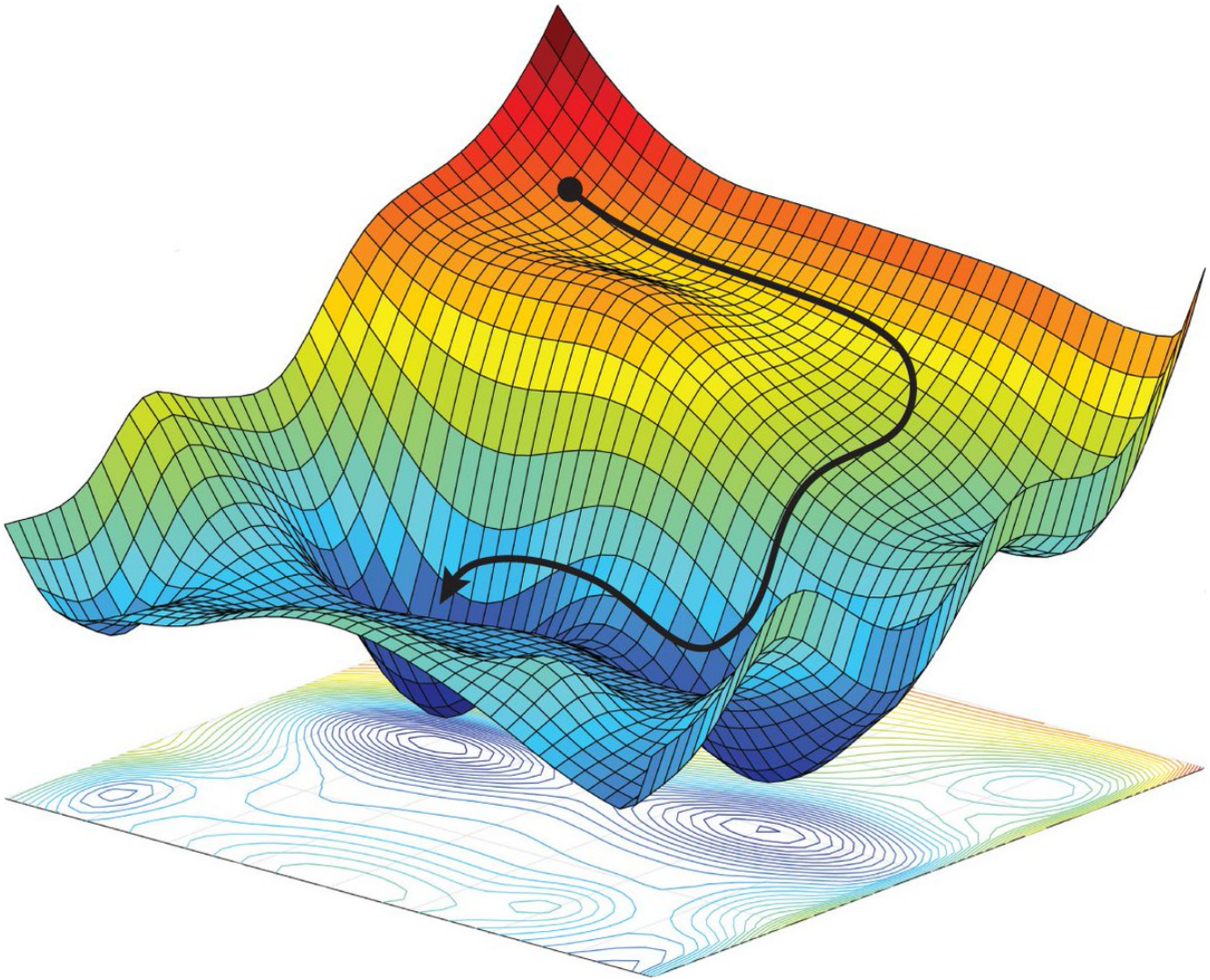
is the log-likelihood of the observation in period t, assuming a normal distribution for the error terms. The first-order derivative vector gradient maps to

$$\nabla \ell_T(\boldsymbol{\Theta}) = \frac{\partial \ell_T(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \sum_{t=1}^{T} \frac{\partial \ell_t(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \sum_{t=1}^{T} \nabla \ell_t(\boldsymbol{\Theta}) \tag{75}$$

Command

In `STATA` the relevant command for maximum likelihood estimation is ml, in `MATLAB` it is mle, and in `Python` MLE is facilitated via the GenericLikelihoodModel class in statsmodels.base.model

MLE - Gradient Descent



Choosing p and q

Generally, the optimal lag order p and q can be determined using some information criterion based on the models' likelihood functions:

- Akaike information criterion (AIC): Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models, i.e., different p and q, using penalties for additional parameters.
- Bayesian information criterion (BIC): does essentially the same but has a different penalty function.

It turns out that for most financial time series a lag of one is often sufficient and including higher order terms has no or not much marginal benefit.

Command

In `STATA` the relevant command for computing the AIC and BIC is estat ic (only works after commands that report the log likelihood), in `MATLAB` both are obtainable in aicbic using log likelihoods or in aic for models in general, and in python AIC and BIC are obtained with statsmodels.tools.eval_measures.aic and statsmodels.tools.eval_measures.bic, respectively.

Volatility Models

Volatility Models

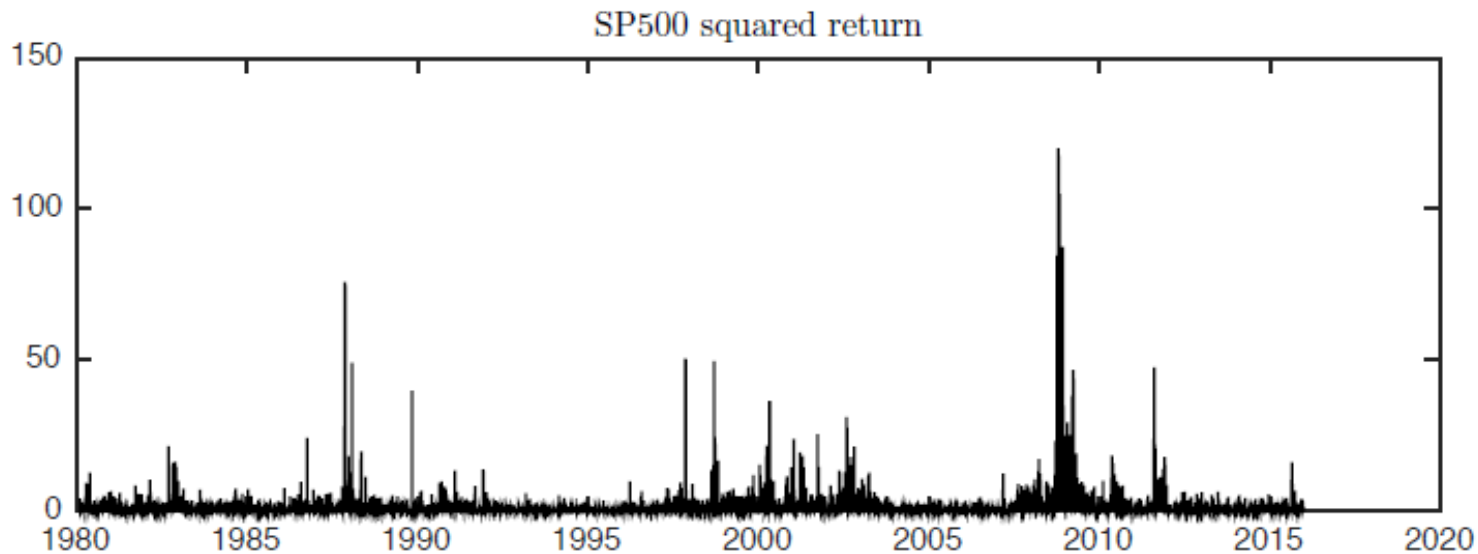
The objectives of this chapter are the following:

- Understanding the role of volatility
- Structure of a volatility model
- (G)ARCH models
- Estimation of (G)ARCH models
- Testing for (G)ARCH effects
- Forecasting

The Role of Volatility

Old view of financial returns: returns are approximately uncorrelated (efficient market hypothesis) and have constant variance (homoskedasticity). However:

- constant volatility hypothesis is clearly rejected by the data as volatility tends to cluster over time (Mandelbrot (1963)).
- Volatility clustering suggests that the conditional return distributions are time-varying.



The Role of Volatility

Volatility is of paramount importance in finance because it is a proxy for risk:

- for the pricing of derivatives such as options
- for asset allocation (trade-off between return and risk)
- for risk management (evaluation of the risk of a portfolio)

Volatility is not directly observable from returns and it is not constant through time, i.e., we have **heteroskedasticity**. Therefore, we are interested in measuring the conditional volatility at time t. Common approaches are:

- squared returns
- absolute returns
- historical volatility
- volatility models

Historical Volatility

A very simple measure of conditional historical volatility is the moving average:

$$\sigma_t^2 = \frac{1}{N}\sum_{i=1}^{N}(r_{t-i} - \mathbb{E}[r_t])^2 \tag{76}$$

There are, however, a few drawbacks:

- this definition gives the same weight to old and new information.
- it generates ghost features such that after an extreme event, there is a huge increase in the historical volatility that stays at that level for as long as the averaging period .

Historical volatility can be useful to measure expected long-term volatility, but is not successful in measuring expected short-term volatility. To that end, we need better models to assess the variability in short-term volatility.

Structure of a Volatility Model

Let r_t be the log-return of an asset at time t and assume that it is characterized by the following model:

$$r_t = \mu_t + \varepsilon_t$$
$$\varepsilon_t = \sigma_t z_t, \qquad \text{with } z_t \sim iid(0,1)$$

where:

- $\mu_t = \mathbb{E}[r_t \mid \mathscr{F}_{t-1}]$ is the conditional mean of r_t
- $\sigma_t^2 = \mathbb{E}\left[(r_t - \mu_t)^2 \mid \mathscr{F}_{t-1}\right] = \mathbb{E}\left[\varepsilon_t^2 \mid \mathscr{F}_{t-1}\right]$ is the conditional variance of r_t

A volatility model is a model that describes the evolution of σ_t^2. There are essentially two types of models for describing the dynamics of volatility:

- Models that describe volatility as an exact function of a given set of variables. This category includes (G)ARCH models: $\sigma_t^2 = \mathbb{E}\left[\varepsilon_t^2 \mid \mathscr{F}_{t-1}\right]$
- Models that describe volatility as stochastic function such as stochastic volatility models: $\sigma_t^2 = \mathbb{E}\left[\varepsilon_t^2 \mid \mathscr{F}_{t-1}\right] + u_t$

ARCH Model

The ARCH (Autoregressive Conditional Heteroskedasticity) model was introduced by Engle (1982),

Basic idea: unexpected returns ε_t are serially uncorrelated but dependent. The dependency in ε_t is described by a quadratic function of its lagged values:

$$\varepsilon_t = \sigma_t z_t \tag{77}$$

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_p \varepsilon_{t-p}^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 \qquad \text{ARCH(p)} \tag{78}$$

We also have:

$$\varepsilon_t^2 = \sigma_t^2 z_t^2 = \sigma_t^2 + \sigma_t^2 (z_t^2 - 1) = \sigma_t^2 + v_t = \mathbb{E}_t[\varepsilon_t^2] + v_t \tag{79}$$

where v_t is the shock in the ε_t^2 process, with $\mathbb{E}_t[v_t] = 0$. Therefore, we have:

$$\varepsilon_t^2 = \sigma_t^2 + v_t = \omega + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_p \varepsilon_{t-p}^2 + v_t \tag{80}$$

and ε_t^2 is therefore an AR(p) process.

ARCH Model

Example: Estimation of an ARCH(10) for the S&P500 daily returns between January 1980 and December 2015.

	Estimate	Std. dev	t-stat
a_0	0.261	0.022	11.656
a_1	0.036	0.013	2.913
a_2	0.102	0.018	5.572
a_3	0.069	0.014	5.056
a_4	0.086	0.019	4.527
a_5	0.096	0.016	6.109
a_6	0.069	0.014	4.783
a_7	0.083	0.017	4.842
a_8	0.074	0.020	3.648
a_9	0.081	0.017	4.678
a_{10}	0.071	0.015	4.796

It turns out that the shocks, ε_t, are very persistent and using an AR(p) process fails to capture the dynamics in the volatility efficiently.

GARCH Model

Due to the large persistence in volatility, the ARCH model often requires a large p to fit the data. In such cases, it is more parsimonious to use the GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model proposed by Bollerslev (1986).

Generalized Autoregressive Conditional Heteroskedasticity

The GARCH(p,q) model is defined as:

$$\varepsilon_t = \sigma_t z_t \tag{81}$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{82}$$

Since $\sigma_t^2 = \mathbb{E}\left[\varepsilon_t^2 \mid \mathscr{F}_{t-1}\right]$, the GARCH(1,1) model writes:

$$\varepsilon_t^2 = \sigma_t^2 + v_t = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + v_t \tag{83}$$

and ε_t^2 is therefore an ARMA(p,q) process meaning that variance is a combination of past volatilities and past shocks.

GARCH Model

Example: Estimation of a GARCH(1,1) for the S&P500 daily returns between January 1980 and December 2015.

	Estimate	Std. dev	t-stat
ω	0.011	0.003	3.956
α_1	0.063	0.008	8.263
β_1	0.927	0.009	104.242

Although the model specification contains first order lags only, its log-likelihood is better than that of the ARCH(10) model (-12484 vs. -12545 for the ARCH(10))

Estimation of a GARCH(p,q)

Manual estimation procedure for the volatility (variance) process of unexpected returns ε, $\sigma^2_{\varepsilon,t}$:

1. Estimate the mean equation $r_t = \mu_t + \varepsilon_t$. Deduce $\hat{\varepsilon}_t = r_t - \hat{\mu}_t$, $\hat{\varepsilon}^2_t$, and the unconditional volatility $\hat{\sigma}^2_\varepsilon = \frac{1}{T}\sum_{t=1}^{T} \hat{\varepsilon}^2_t$
2. Select initial arbitrary values for $\boldsymbol{\Theta}^{(0)} = [\omega^{(0)}, \alpha_1^{(0)}, \ldots, \alpha_p^{(0)}, \beta_1^{(0)}, \ldots, \beta_q^{(0)}]'$ and set the first q conditional volatilities to $\hat{\sigma}^2_{\varepsilon,1}, \ldots, \hat{\sigma}^2_{\varepsilon,q} = \hat{\sigma}^2_\varepsilon$
3. Compute the conditional volatility $\hat{\sigma}^2_{\varepsilon,t} = \omega^{(0)} + \sum_{i=1}^{p} \alpha_i^{(0)} \hat{\varepsilon}^2_{t-i} + \sum_{j=1}^{q} \beta_j^{(0)} \hat{\sigma}^2_{\varepsilon,t-j}, \forall t > \max[p,q]$
4. Compute the log-likelihood in equation 74
5. Change the parameter vector to, say $\boldsymbol{\Theta}^{(1)}$ such that the log-likelihood increases
6. Iterate step 3-5 until the log-likelihood converges to a fixed value

Command

In `STATA` the relevant command to fit models from the (G)ARCH family is arch, in `MATLAB` it is available via the MFE Toolbox using the tarch command, and in `Python` it is arch.univariate.GARCH

Testing for (G)ARCH Effects

Assume we want to test the null hypothesis that the variance is homoskedastic against the alternative that the variance is heteroskedastic and is characterized by a (G)ARCH(1,1) process. It turns out that the test is the same for ARCH and GARCH effects. Engle (1982) proposes a Lagrange-multiplier (LM) test for ARCH effects:

- $H_0 : \varepsilon_t \mid \mathscr{F}_{t-1} \sim N(0, \sigma^2)$
- $H_a : \varepsilon_t \mid \mathscr{F}_{t-1} \sim ARCH(p)$

The LM test statistic is equivalent to the $T \times R^2$ test statistic, where T is the sample size and R^2 is computed from the regression:

$$\hat{\varepsilon}_t^2 = \gamma_0 + \gamma_1 \hat{\varepsilon}_{t-1}^2 + \ldots + \gamma_p \hat{\varepsilon}_{t-p}^2 + u_t, \forall t \tag{84}$$

Under the null of no ARCH effects, the $T \times R^2$ statistic is asymptotically distributed as a $\chi^2(p)$. Alternatively, we could also use the Ljung-Box statistic for $\hat{\varepsilon}_t^2$ with p lags, which is also asymptotically distributed as a $\chi^2(p)$.

Forecasting Conditional Volatility

Forecasts of the GARCH model are obtained recursively. Let t be the starting date for forecasting. The 1-step ahead forecast for σ^2_{t+1} is:

$$\sigma^2_t(1) = \hat{\omega} + \hat{\alpha}_1 \hat{\varepsilon}^2_t + \hat{\beta}_1 \hat{\sigma}^2_t \tag{85}$$

Since $\varepsilon^2_t = \sigma^2_t z^2_t$, the GARCH(1,1) can be rewritten:

$$\sigma^2_t = \omega + \alpha_1 \varepsilon^2_{t-1} + \beta_1 \sigma^2_{t-1} = \omega + (\alpha_1 + \beta_1)\sigma^2_{t-1} + \alpha_1 \sigma^2_t (z^2_t - 1) \tag{86}$$

such that at time $t+2$ we have:

$$\sigma^2_{t+2} = \omega + (\alpha_1 + \beta_1)\sigma^2_{t+1} + \overbrace{\alpha_1 \sigma^2_{t+1}(z^2_{t+1} - 1)}^{\mathbb{E}_t[(z^2_{t+1}-1)|\mathscr{F}_t]=0} \tag{87}$$

The 2-step ahead forecast for σ^2_{t+2} is then:

$$\sigma^2_t(2) = \hat{\omega} + (\hat{\alpha}_1 + \hat{\beta}_1)\sigma^2_t(1) \tag{88}$$

and so on: $\sigma^2_t(K) = \hat{\omega} + (\hat{\alpha}_1 + \hat{\beta}_1)\sigma^2_t(K-1)$

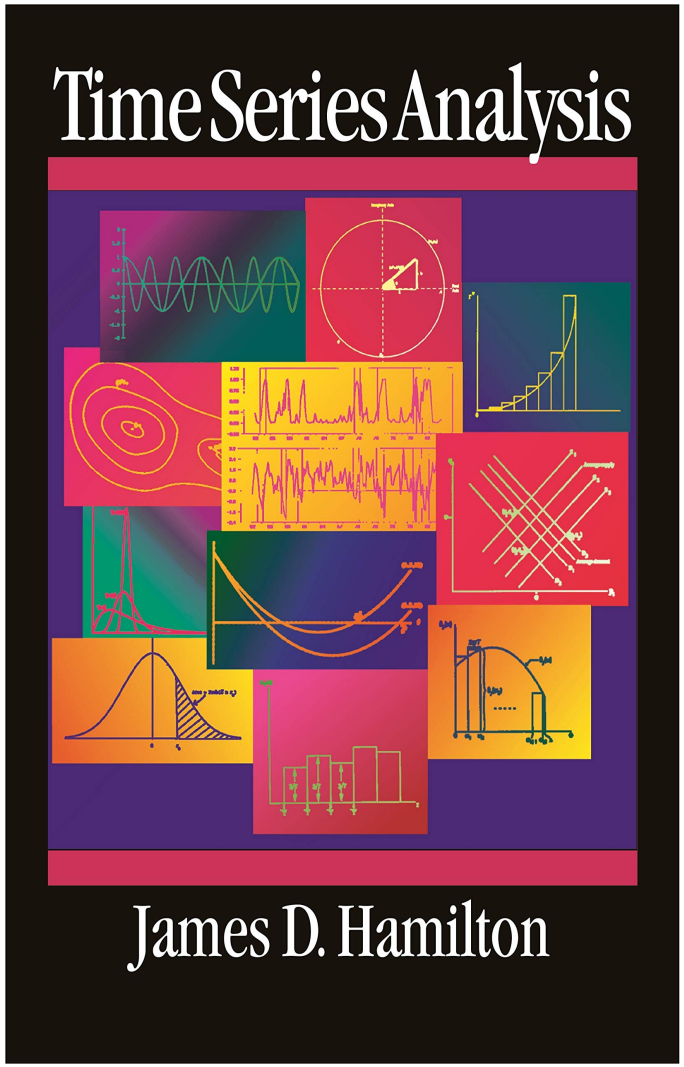
Multivariate Time Series Extensions

(G)ARCH models are univariate models and are fitted to a single time series. In effect, (G)ARCH models are limited to forecasting variances of univariate time series. In order to forecast covariance matrices and account for effects across time series it is necessary that the (G)ARCH models be augmented to a multivariate time series setting. To that end, various multivariate generalizations of the univariate (G)ARCH models have been established, among others:

- Constant conditional correlation GARCH (CCC-GARCH, Bollerslev (1990))
- Dynamic conditional correlation GARCH (DCC-GARCH, Engle (2002))

Reading Tip

Hamilton, J. (1994), Time Series Analysis, Princeton



Introduction to Financial Machine Learning

Financial ML

The objectives of this chapter are the following:

- Understanding what ML is
- How to use it
- Applications of ML in Finance
- Neural networks
- Reinforcement learning
- Data preprocessing
- Cross-validation and overfitting

ML Definition

"Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task." (Source: Wikipedia)

ML Definition

There are at least four distinct features in this definition:

- Task
- Sample Data (input)
- Algorithms and statistical models to recognize patterns (learning process)
- Predictions and decisions (output)

Input

The input denominates the data from which you want to learn something and can literally be of any type:

- Textual data
- Numerical data
- Cross-sectional data
- Time series
- Images
- Motion pictures
- etc.

However, it is important that the data be pre-processed such as to fit the input requirements of your ML technique/algorithm.

Output

The ouput denominates the "knowledge" you obtained from your data. This insight most likely belongs to one of the three following types:

- Prediction of a single numerical output value
- Classification into a finite set of at least two distinct categories
  - binary (yes/no)
  - discrete set of categories (cat/dog/horse/...)
  - decision to achieve optimal terminal state(move up/down/sideward)
- Insight into your Data
  - Importance of individual variables
  - Dependencies in your data
  - Outlier detection

ML Algorithms

ML techniques that seem reasonable for financial applications include:

- Recurrent neural networks (predictions)
- Reinforcement learning (decisions)
- Support Vector Machines
- Restricted Boltzmann machines
- XGboost
- ...

Applications in finance

The goal of ML is to maximize the precision of the out-of-sample prediction or classification. As such, ML is a poor choice when one intends to gain true economic insight such as causal effects. However, whenever the goal is to improve predictability, ML may excel as it is capable of accounting for complex non-linear structures and dependencies in the data. For instance, good applications for ML in finance might be:

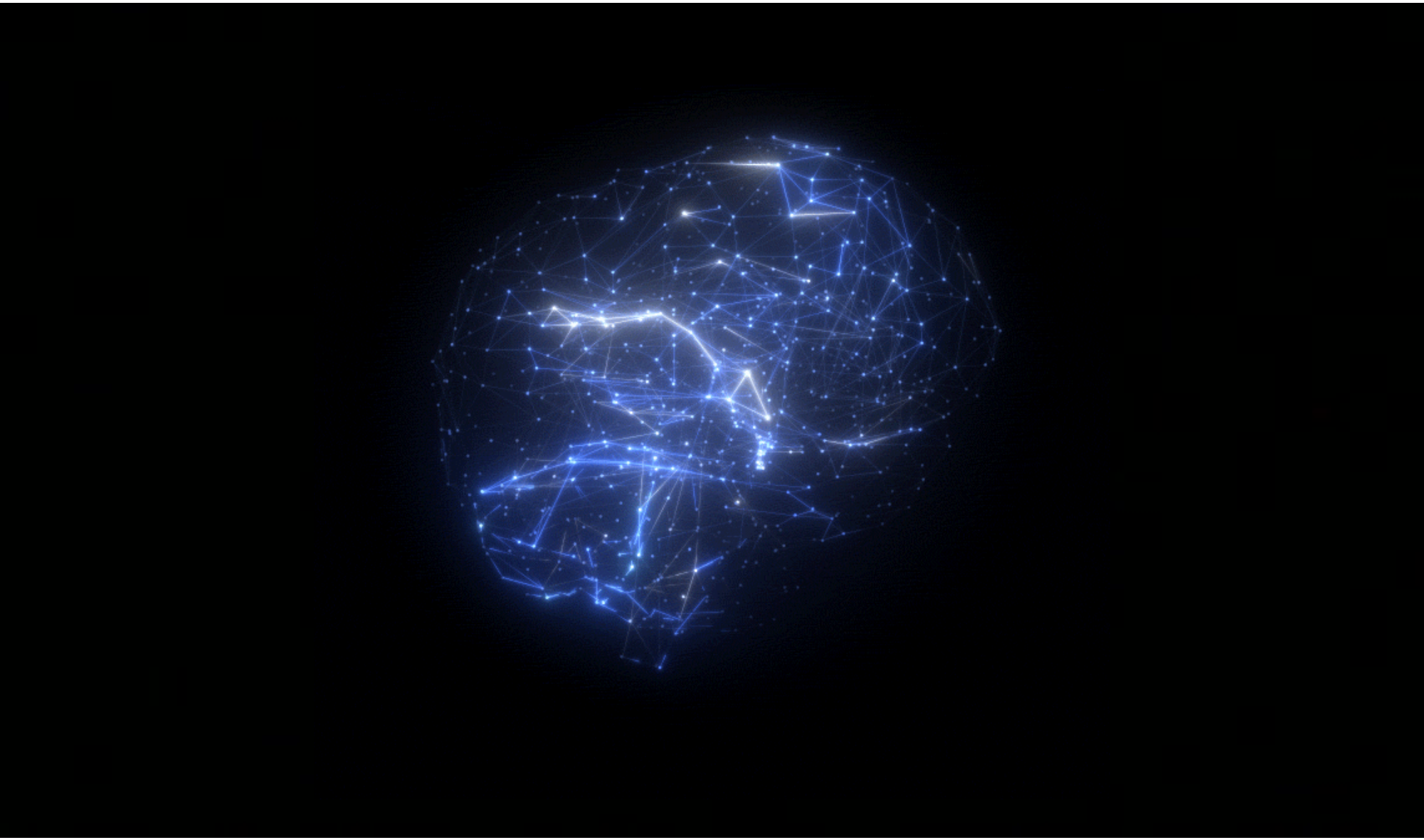
- Prediction of financial time series
- Market-timing and trading rules
- Dynamic optimization
  - improve asset management decisions
  - improve corporate decisions
- Detect anomalies in the data, including outlier detection
- Risk management

Implementation

Python offers readily available libraries that let you implement state of the art ML techniques:

- Tensoflow 2.0 together with Keras (pip install Tensorflow, pip install Keras)
- Scikit-learn (pip install sklearn)
- Prophet (pip install fbprophet)
- Gym (pip install gym)
- PyTorch (pip install pytorch)

Neural Networks



Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing model that is inspired by the way the human brain processes information. An ANN is a network of connected nodes called artificial neurons where each neuron is able to transmit a signal to other connected neurons. The signal which is transmitted is a real number, and the signal output of each neuron is computed by some non-linear function of the sum of its weighted inputs. Signals travel from the first layer, the input layer, to the last layer, the output layer, passing through hidden layers.

Components of a ANN

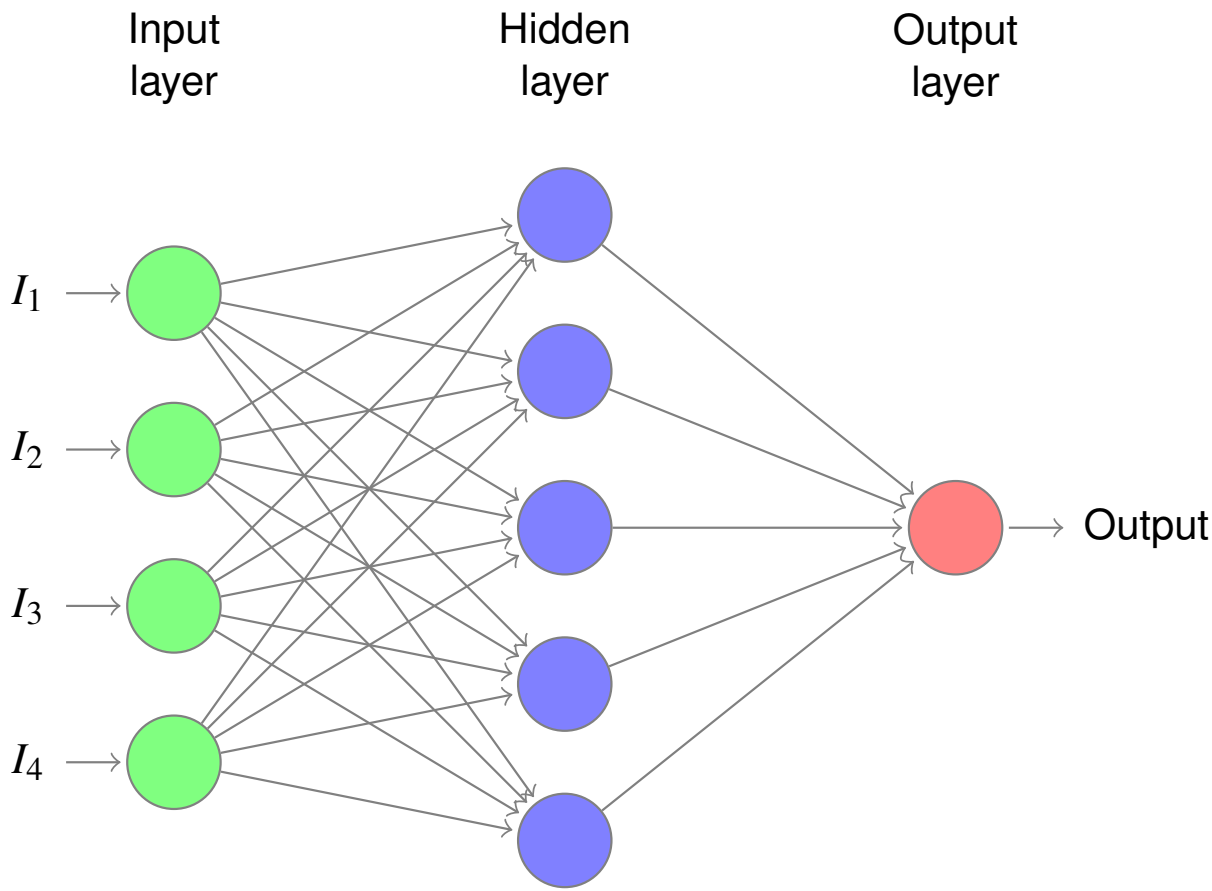
A neural network is essentially a directed and weighted graph. Its major components are:

- **Neurons**: Nodes which receive input, x, combine the input, and produce the output, $\hat{y}$.
- **Connections and weights**: The connections that link neurons along which the signal is transmitted. Each connection is assigned a weight that represents its relative importance. A neuron can have multiple input and output connections.

with additional components specifically pertaining to hidden layers:

- **Propagation function**: The propagation function, Σ, computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum and passes it on to an activation function. A bias term can be added to the result of the propagation.
- **Activation function**: A non-linear function, f, that a neuron uses to produce an output from its propagated input. Usually, a neuron is only activated if the propagated input value exceeds a specified threshold value.

Network Architecture



Network Architecture

The number of layers and the number of neurons in each layer are model hyperparameters that one must specify. The architecture for the input and output layers are determined by your input data and output variable, respectively:

- **Input layer**: The number of neurons in the input layer must match the number of input variables.
- **Output layer**: The number of neurons in the final output layer depends on the type of task your network is supposed to perform. In a classification problem the number of neurons must match the number of different classes. When predicting a numerical value the output layer consists of a single neuron. If the output is a binary decision, the final output layer consists of two neurons.

Network Depth

In general, there is no analytical way to determine the number of hidden layers or the number of nodes to use per hidden layer in an artificial neural network. Generally:

- first-order hidden layers capture linear effects (shallow neural network).
- second-order and deeper hidden layers capture non-linear effects and other higher-order dependencies (deep neural network).
- the deeper the network the more the level of abstraction increases from the space of the input variables to the output variables.
- the number of neurons decreases from one hidden layer to another.

At the end of the day one has to configure the hyperparameters for the specific task via systematic experimentation such as grid-search.

Financial Applications

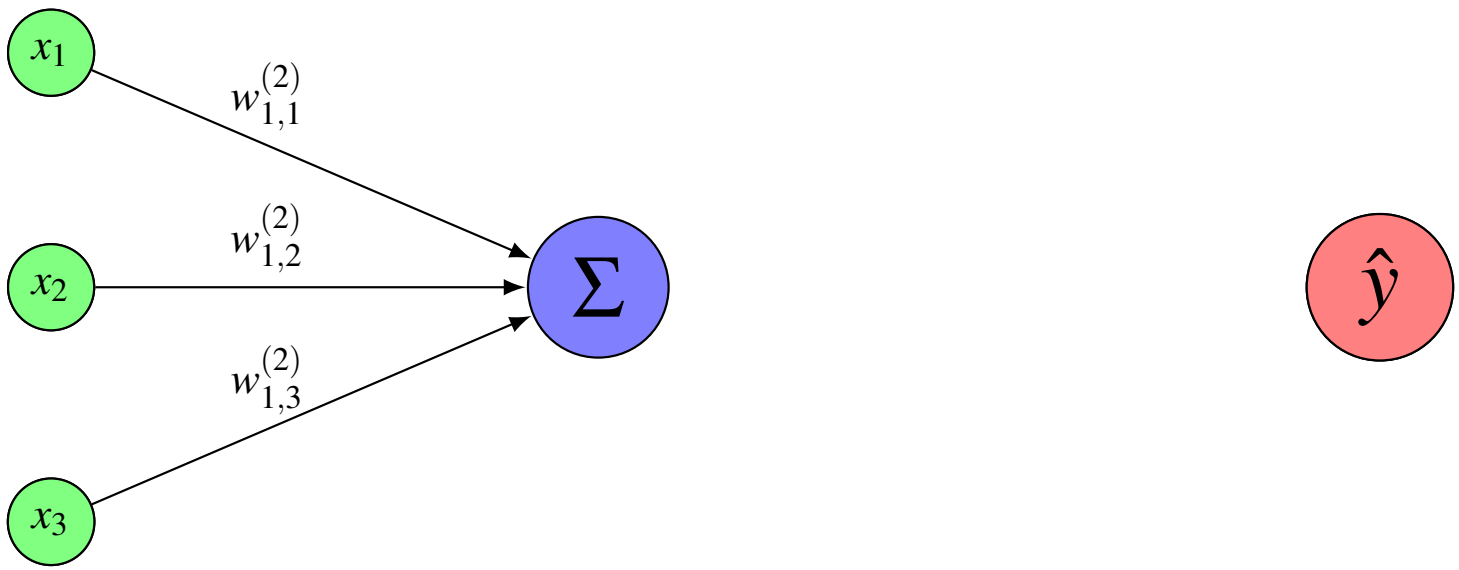
Neural networks are a natural candidate for predicting or classifying financial time series:

- predicting the sign of next period's return
- predicting the magnitude of next period's return
- classifying next period's return (strongly negative, negative, neutral, positive, strongly positive)
- predicting next period's volatility
- ...

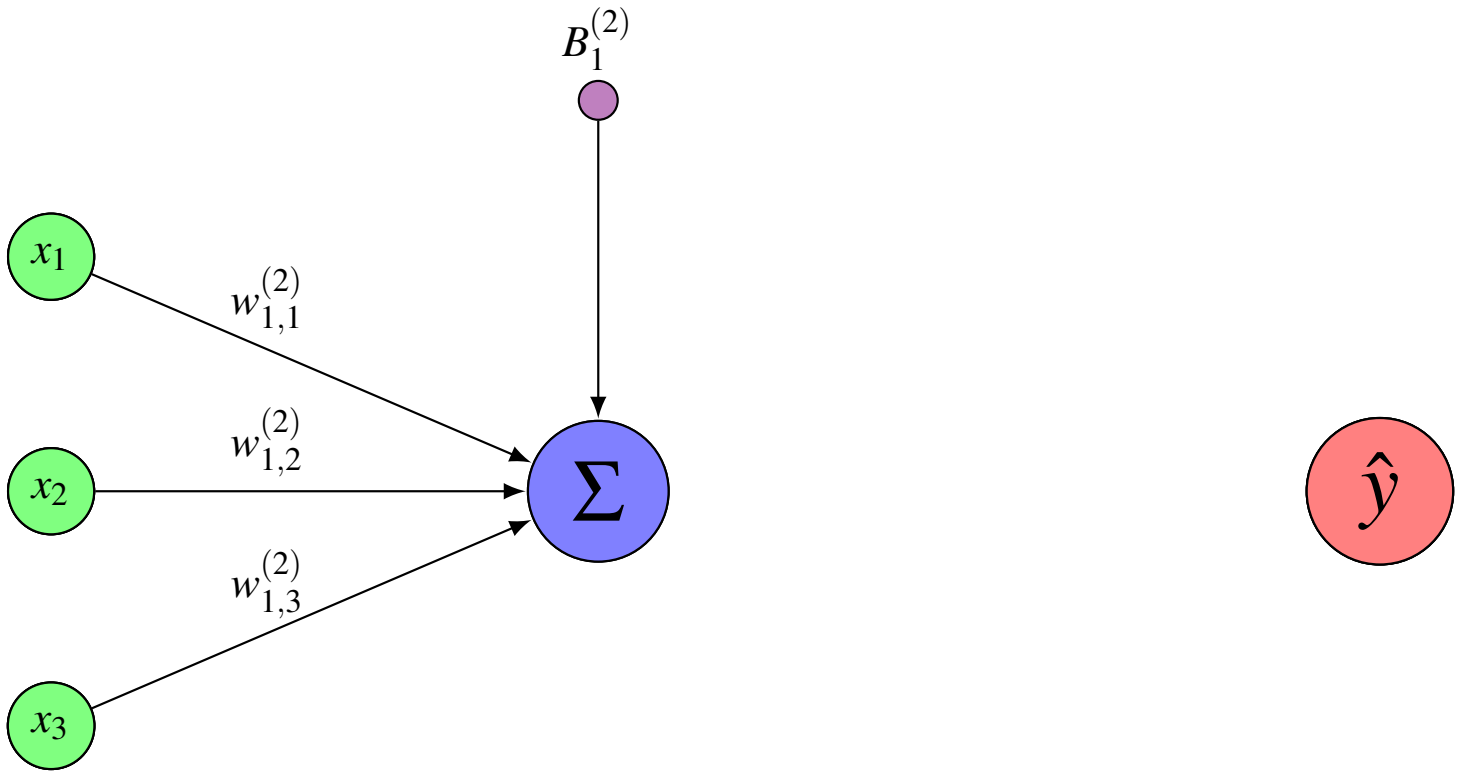
Generally, it is recommended to work with stationary time series and not absolute price levels. With absolute price levels, neural networks will learn very quickly that taking the current price level as a forecast for next period's price level leads to high accuracy due to a low absolute error and the forecasted result is simply a lagged time series.

Neuron Activation

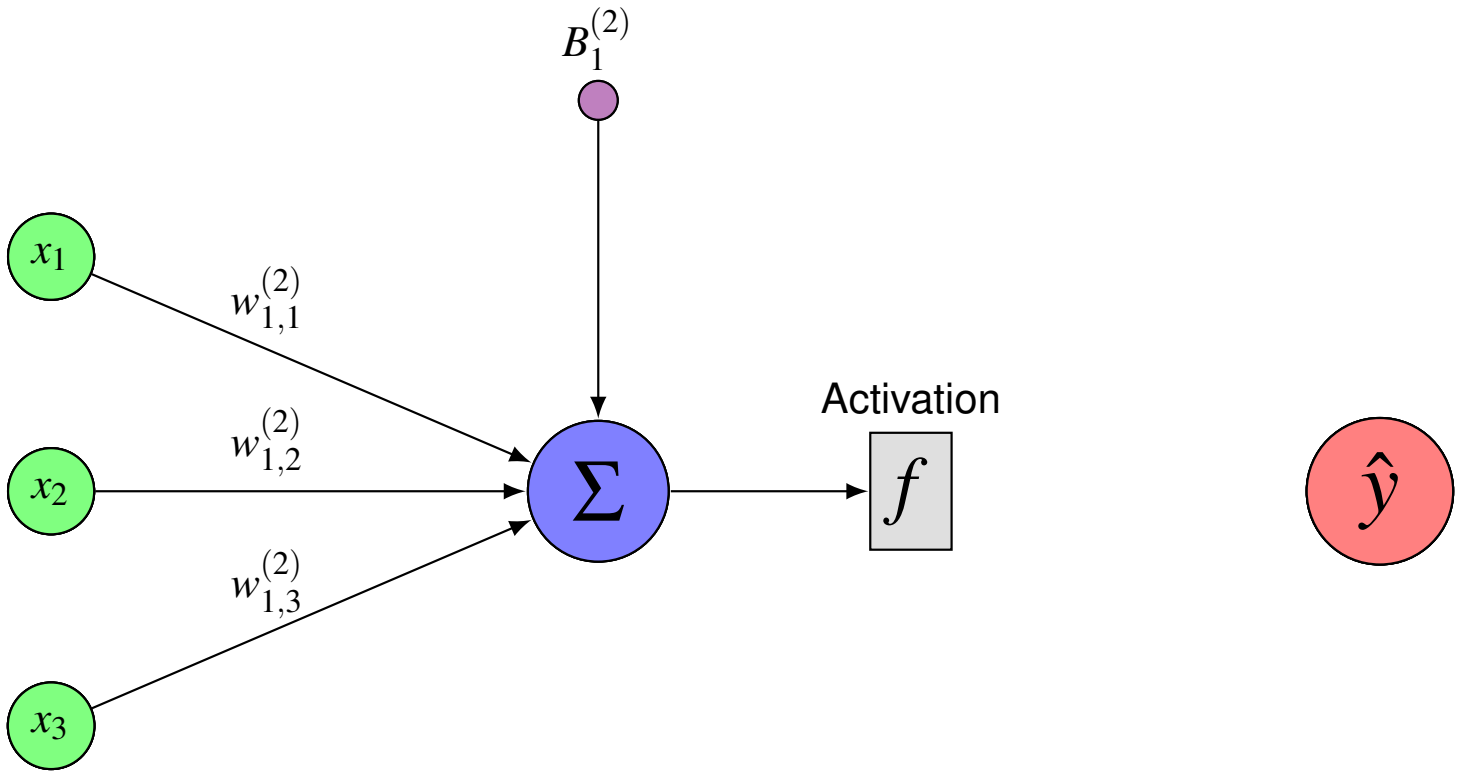
Neuron Activation



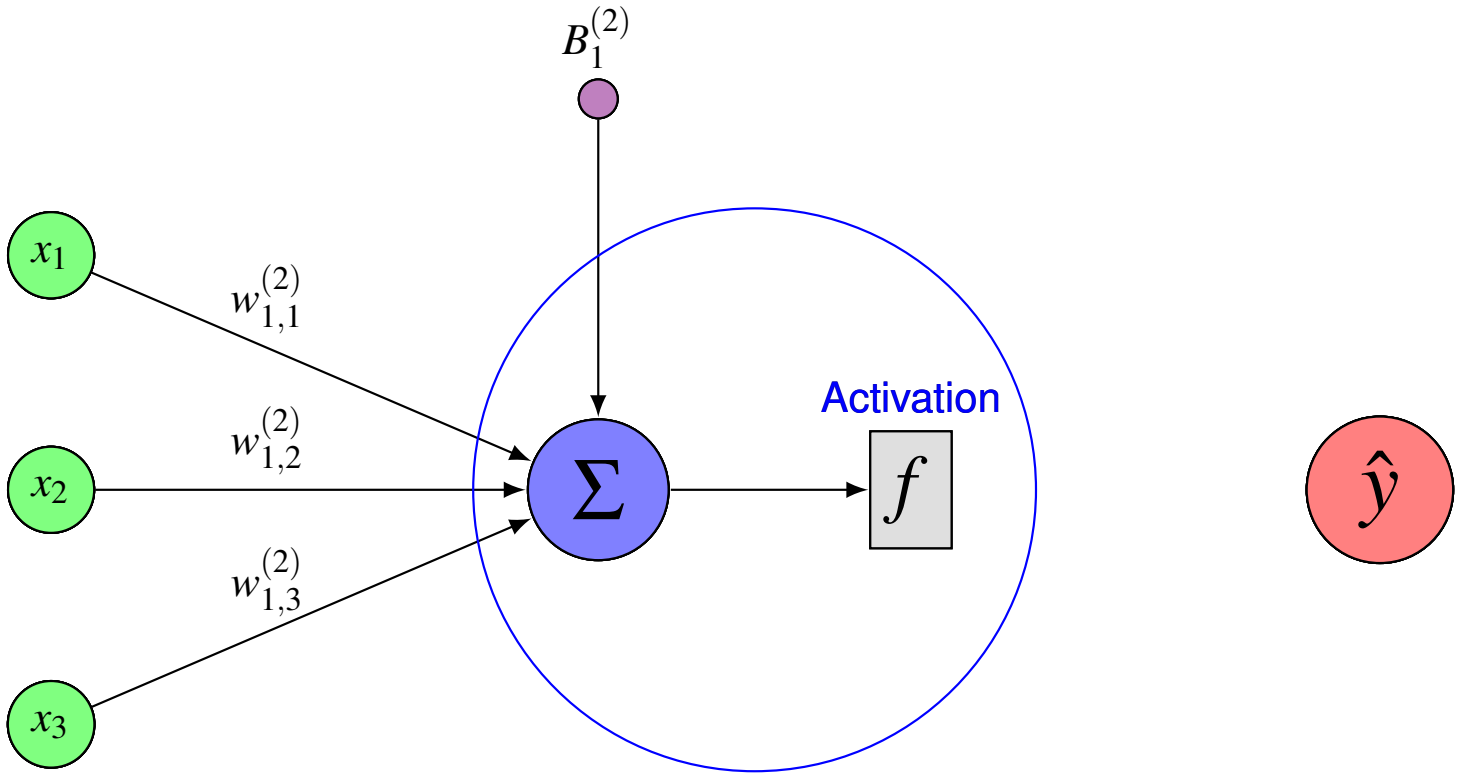
Neuron Activation



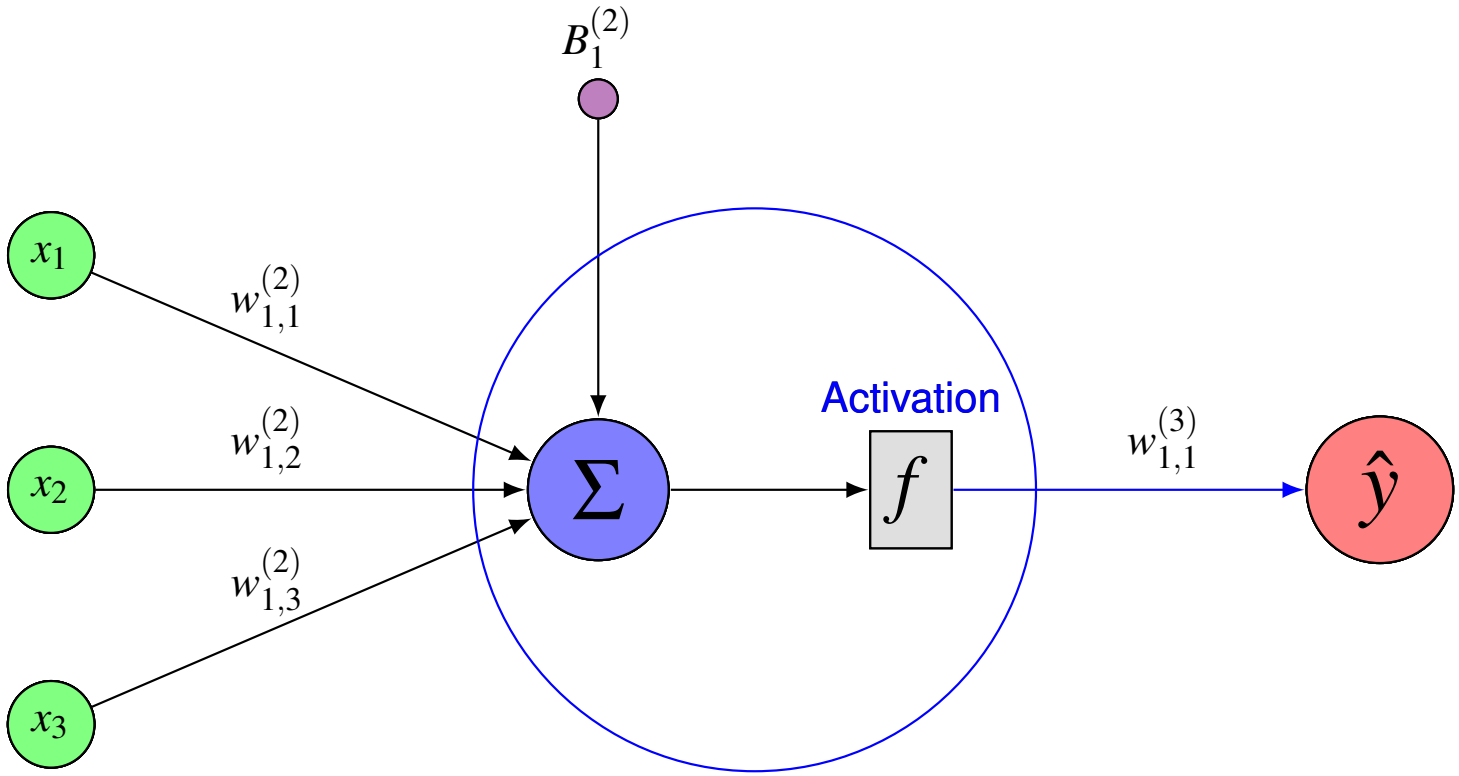
Neuron Activation



Neuron Activation



Neuron Activation



Activation Function

Some prevalent activation functions are:

- **Identity activation function**: $f(z) = z$
- **Rectified linear unit (ReLU)**: $f(z) = \max\{z, 0\}$; the activation function will return whatever propagated input value, z, is given to it, so long as it is above 0
- **Sigmoid**: $f(z) = \frac{1}{1+e^{-z}}$; non-linear function which has a maximum that approaches 1 and a minimum that approaches 0. Natural candidate for cases in which the classification is false (0) or true (1)
- **Softmax**: $f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{-z}}$; A function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities for the K inputs.

The activation function of your final output should be able to reflect the possible values the final output can take! For instance, standardized returns can be both positive and negative. Hence, activation functions that are truncated at zero are a bad fit and an identity activation function is preferable. In hidden layers, this is not a problem

Training Neural Networks

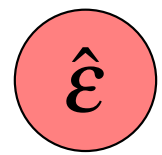
The training of an ANN is the process in which the network learns through iteration to better handle the objective at hand by considering sample observations. Learning involves adjusting the weights (and in effect, activation) of the network to improve the accuracy of the result. This is done by minimizing the observed errors, $\hat{\varepsilon} = \hat{y} - y$, or in some cases, a cost function (reinforcement learning). Learning is complete when the error, ε cannot be reduced any further. Important concepts pertaining to the learning process are:

- **Backpropagation**: A method to adjust the connection weights in order to reduce the error, ε, in the next training iteration.
- **Learning rate**: Defines the size of the corrective steps that the model takes on weights in order to reduce the error, ε. A high learning rate shortens the training time, but lowers accuracy, while a lower learning rate takes longer, but potentially increased accuracy.

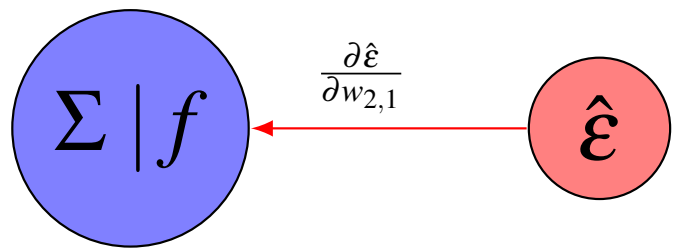
Backpropagation

During backpropagation the error is "fired" back through the network in order to asses which nodes and weights contributed the most to the error and, in effect, where a change of weights reduces the error most effectively.

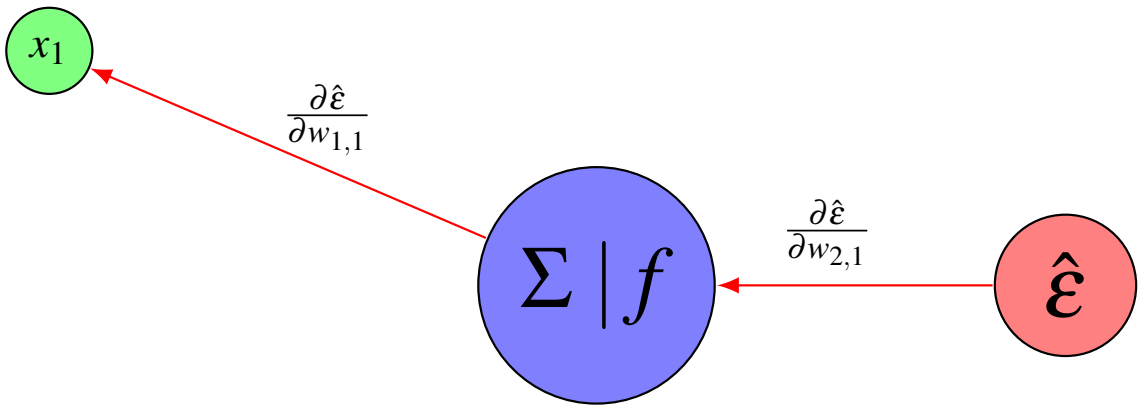
Backpropagation



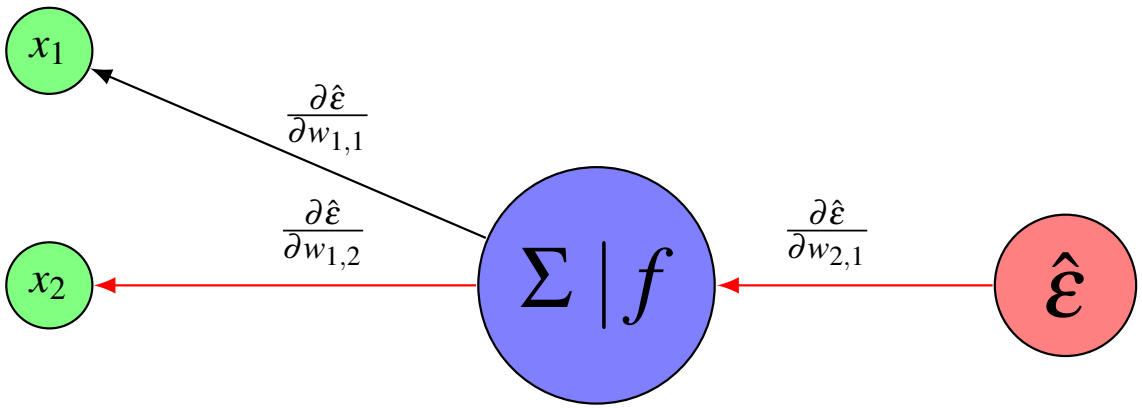
Backpropagation



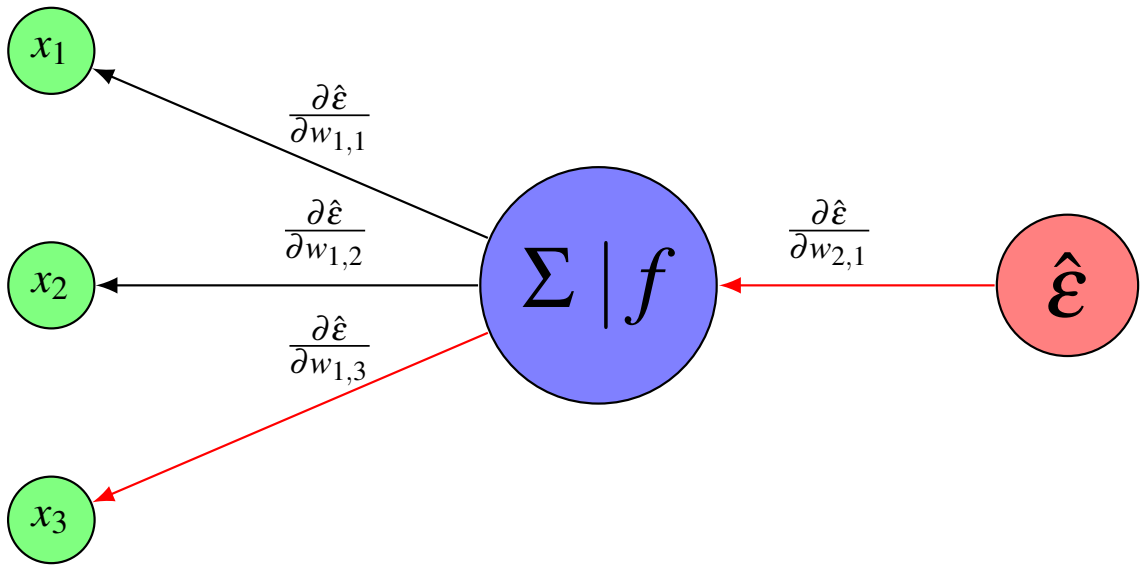
Backpropagation



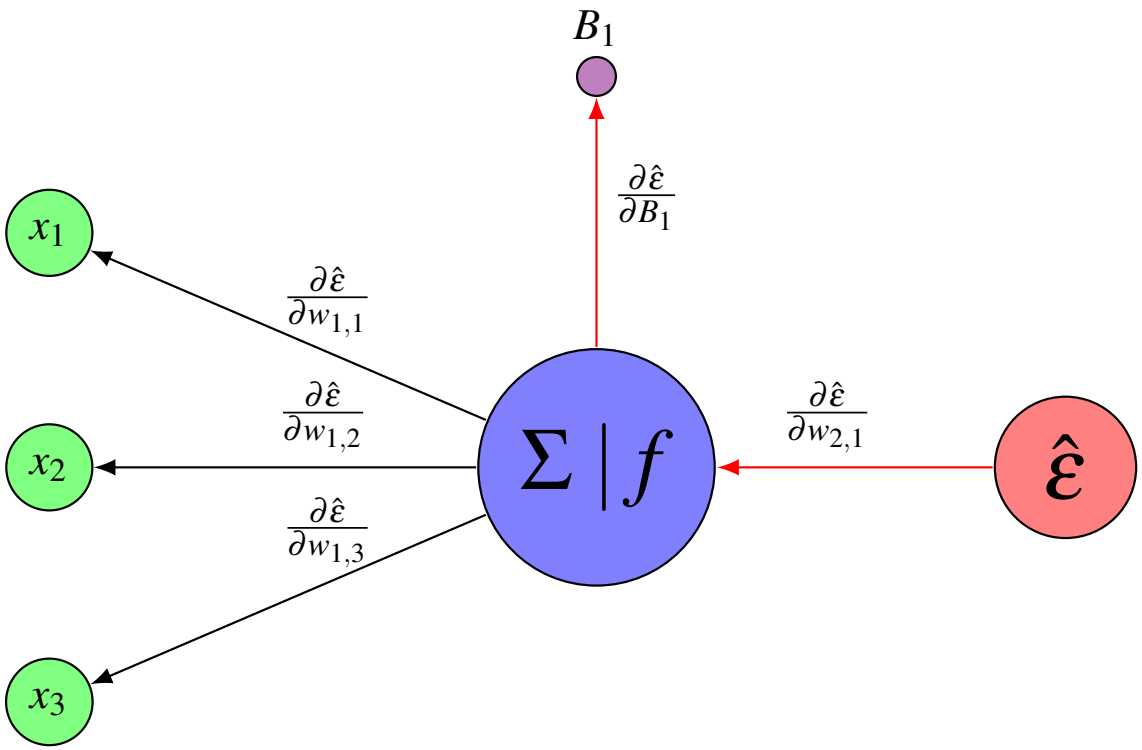
Backpropagation



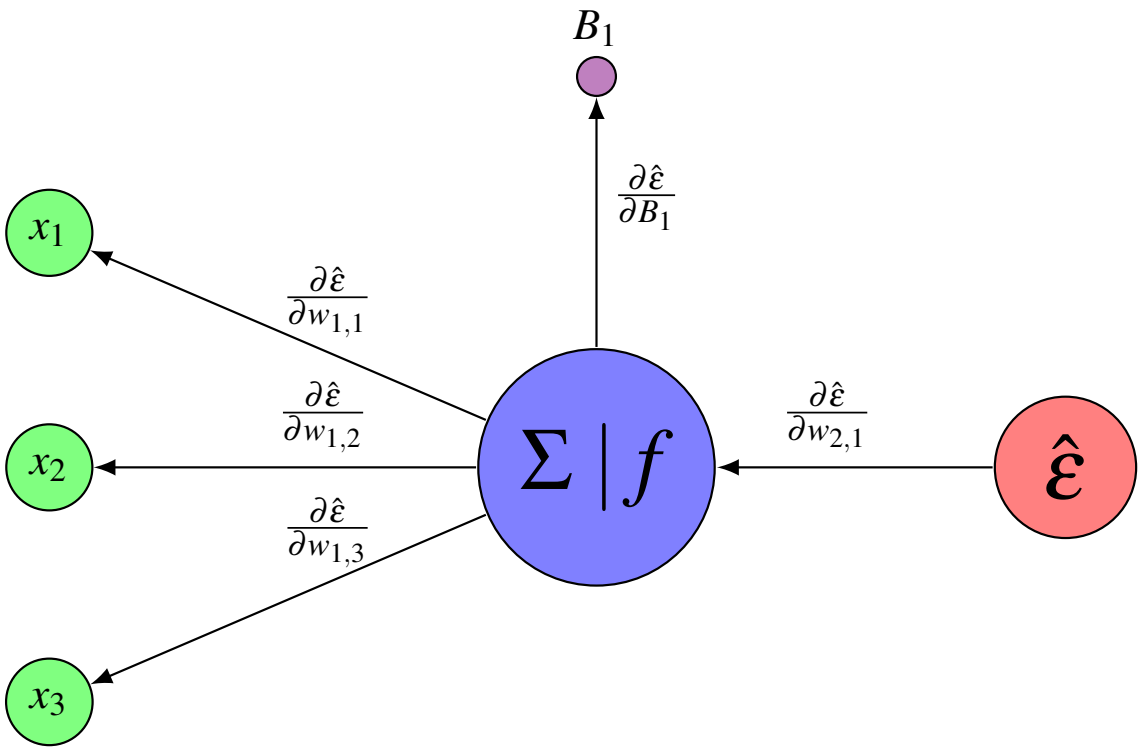
Backpropagation



Backpropagation



Backpropagation

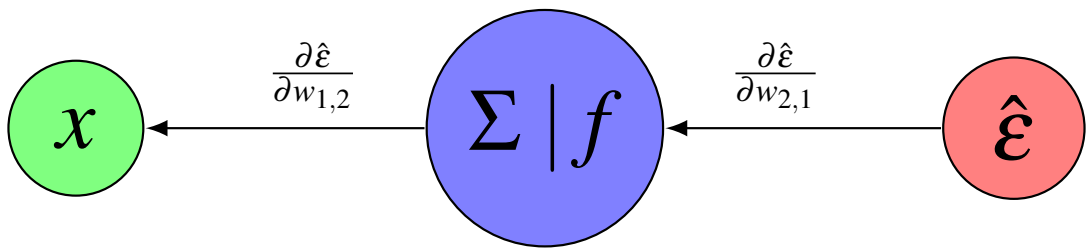


Vanishing Gradient Problem

The degree to which a parameter, for instance weight, affects the final output is dictated by the number of layers there are between the parameter and the output. With every additional layer the effect decreases. This is called the vanishing gradient problem. The reason therefore is, that a parameter change affects all other parameters towards the output and the sensitivity (gradient) is, simplified, the result of a chain rule:

$$\frac{\partial \hat{\varepsilon}}{\partial w_{1,2}} = \frac{\partial \hat{\varepsilon}}{\partial w_{2,1}} \cdots \frac{\partial w_{2,1}}{\partial w_{1,2}} \tag{89}$$

(90)



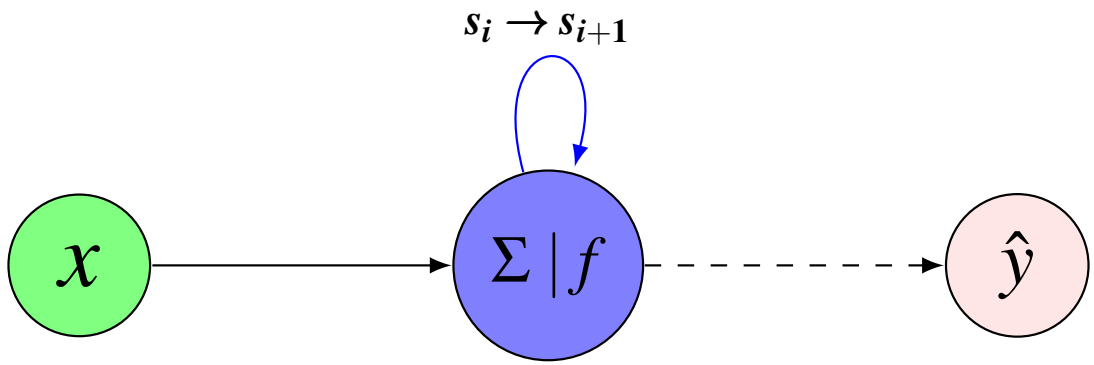
Recurrent Neural Network

A specific family of ANNs are Recurrent Neural Networks. RNNs are widely used for data with sequential structure. For instance, financial time series data have an intrinsic ordering based along the time dimension. RNNs are endowed with a sequential architecture. This property makes them a good fit for time structured data because the sequential architecture:

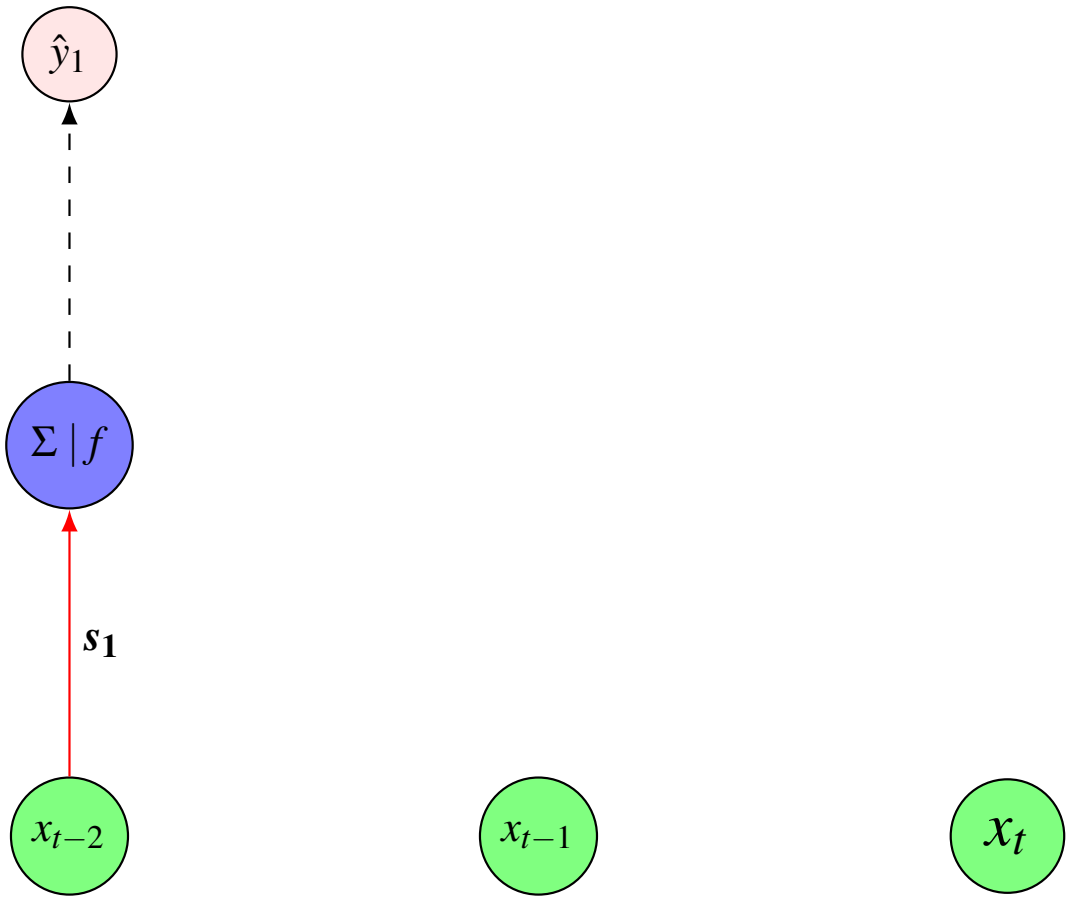
- facilitates sequential signal flow through the neural network.
- enables the network to have artificial "memory".

Recurrent Neural Network

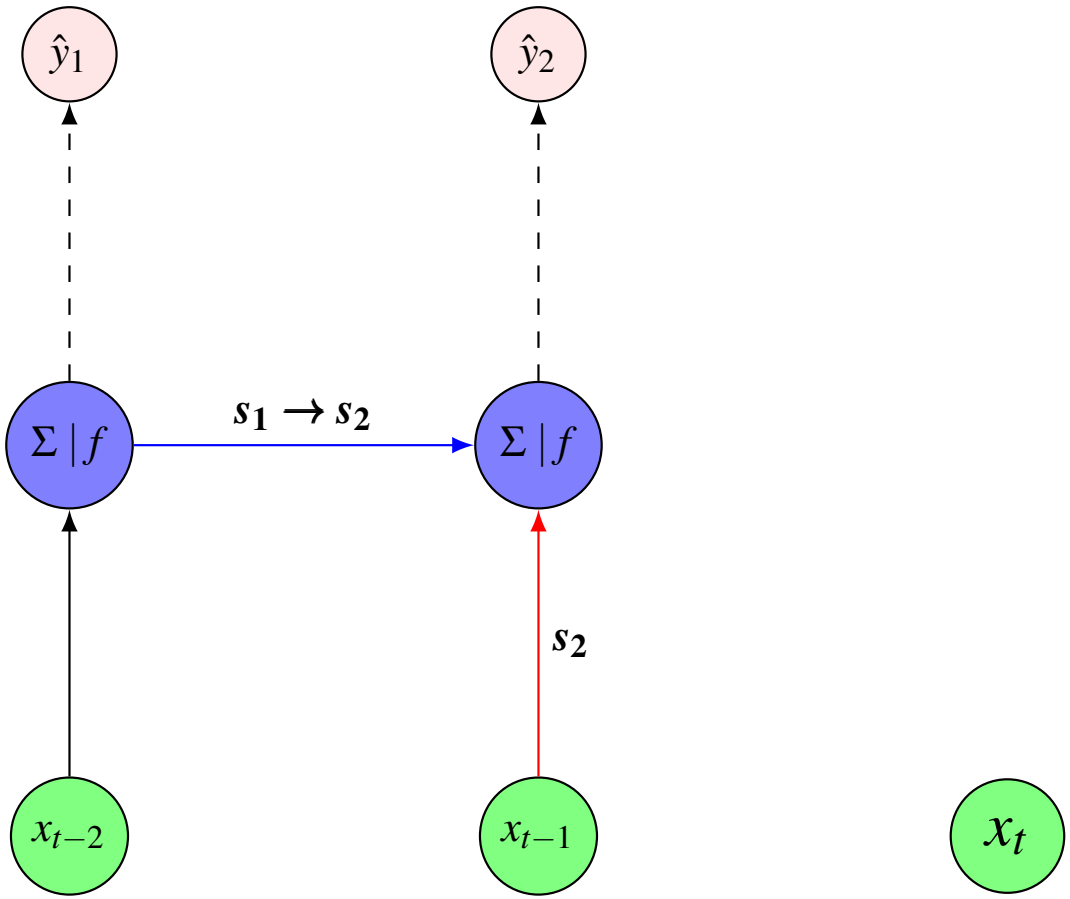
RNNs can build up sequential memory due to loops in their architecture. Neurons in hidden layers are not only connected with other neurons but are also connected with themselves, allowing them to loop and store the signal sequentially:



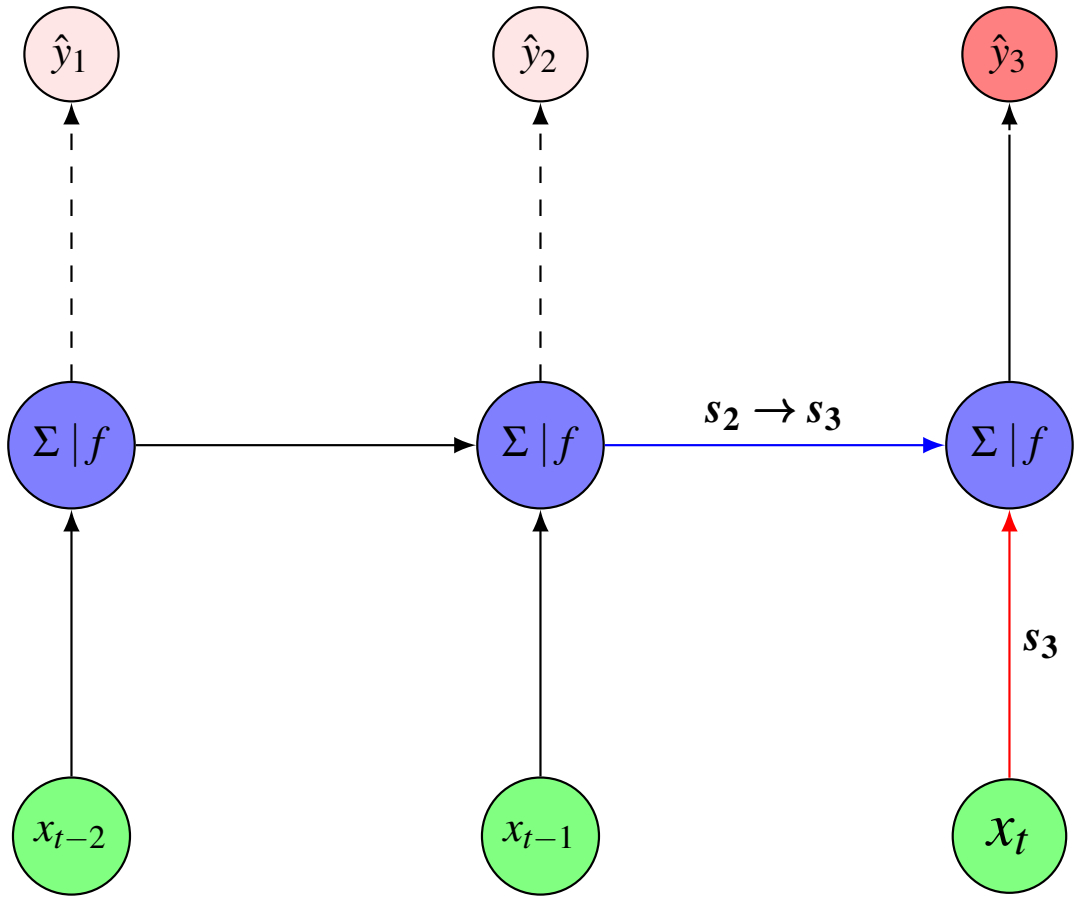
Recurrent Neural Network



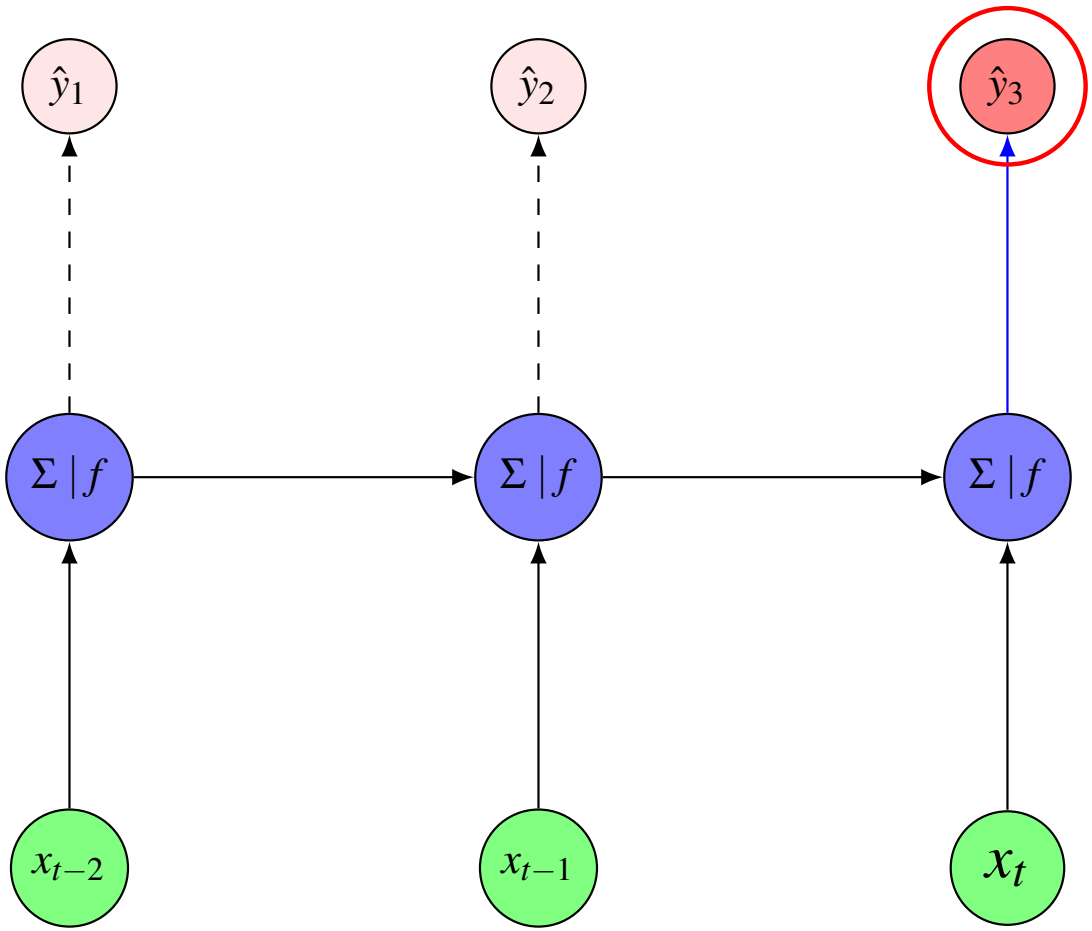
Recurrent Neural Network



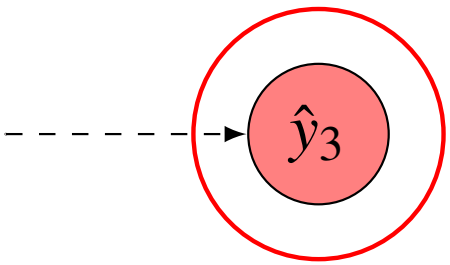
Recurrent Neural Network



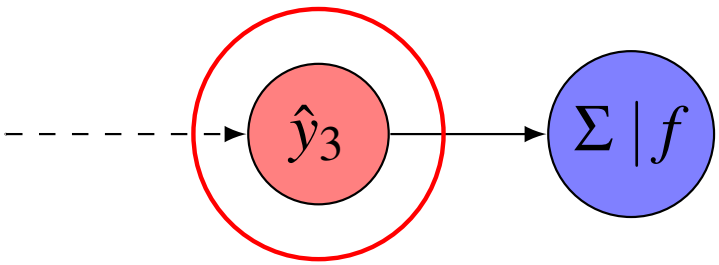
Recurrent Neural Network



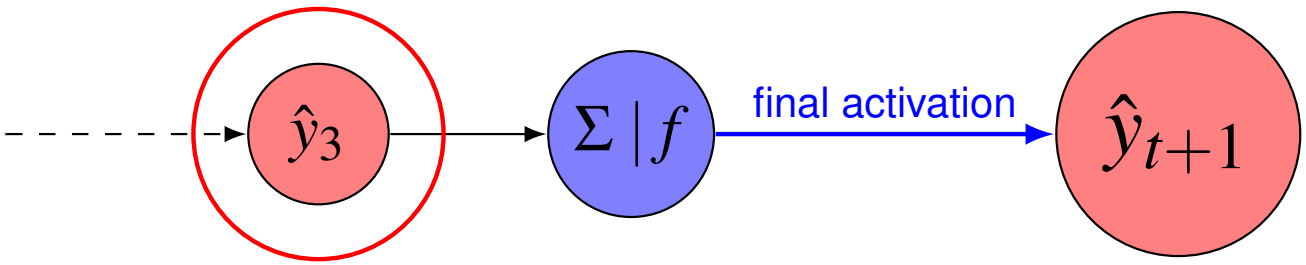
Recurrent Neural Network



Recurrent Neural Network



Recurrent Neural Network

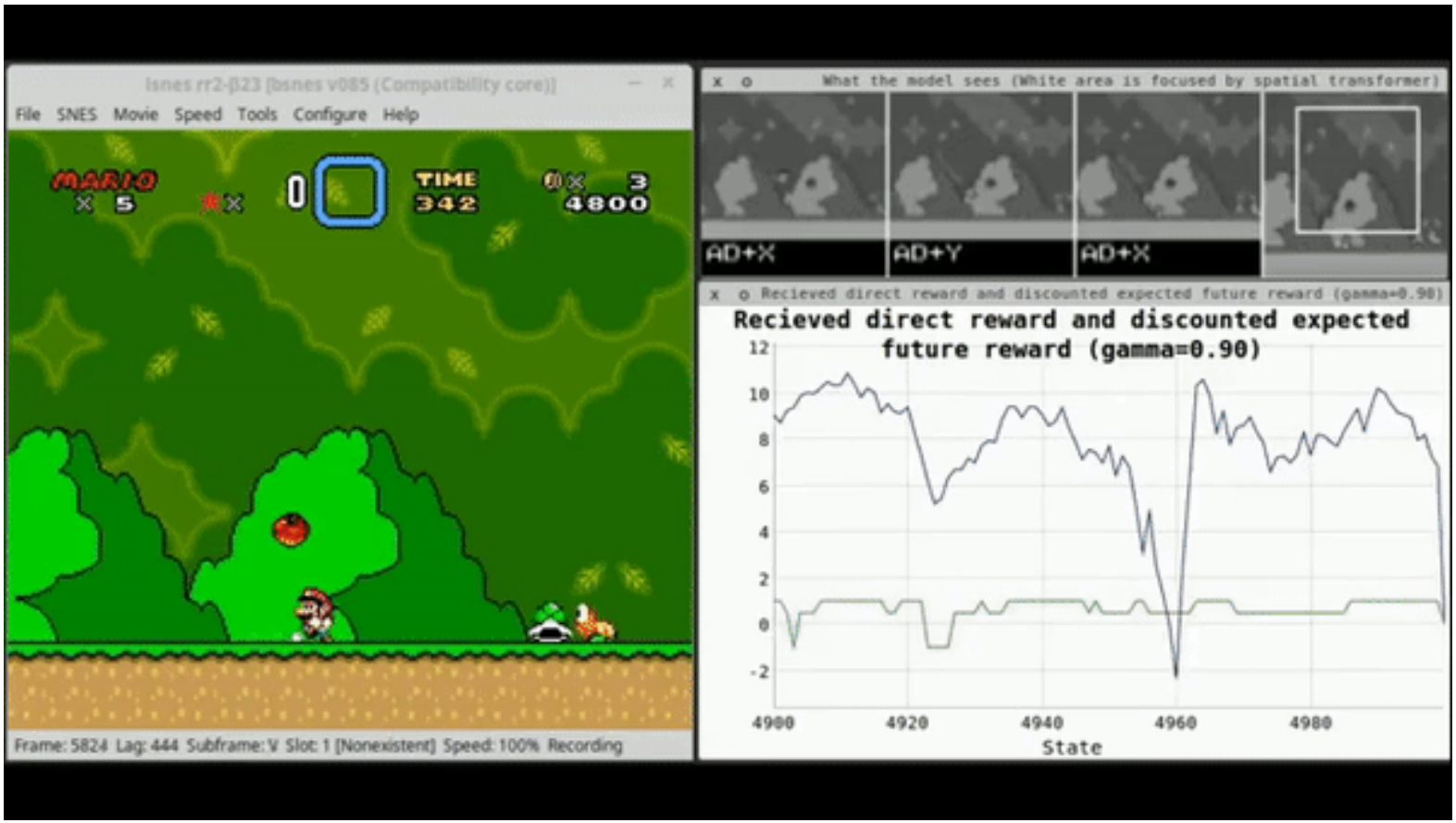


RNN Vanishing Gradient Problem

In RNNs information is looped sequentially and mixed up with new input information. The amount of initial information quickly diminishes such that after only a few sequences not much of the initial information is left and has little impact on the final output (vanishing gradient problem). In effect, standard RNNs have short memory and can only take limited advantage of time series' intrinsic structures. There are augmented RNNs that overcome this issue, most prominently:

- LSTM (Long Short-Term Memory)
- GRU (Gated Recurrent Unit)

Reinforcement Learning



Reinforcement Learning

Reinforcement Learning (RL) is a framework to model the decision making process:

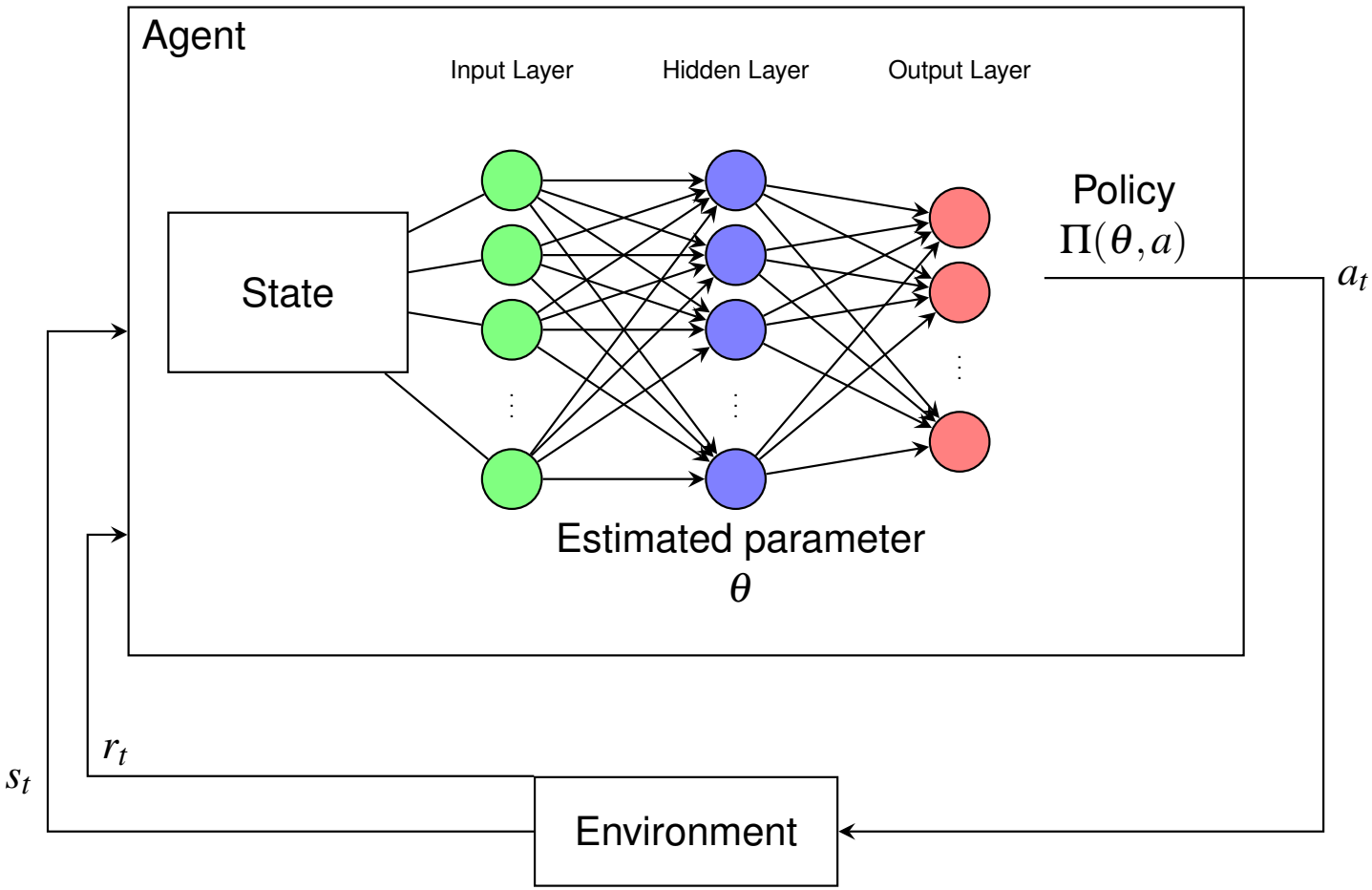
- of an agent,
- in a environment with a sequence of possible states, s_t,
- in order to maxmize a sequence of rewards, r.

Reinforcement learning does not necessarily need labelled input and output variables. Rather, a RL problem is typically stated in the form of a Markov decision process (MDP) and uses dynamic programming techniques:

- to find the optimal action, a, to be taken,
- given some state, s_t,
- in order to maximize some notion of cumulative reward (final value), V.

The rule that assigns a given state to the best possible action is called policy, $\Pi(\theta, a)$, and is learned using some ML technique such as neural networks.

Deep Reinforcement Learning



Optimal Managerial Decision

Example 1: Optimal managerial decision

Assume that the firm value for shareholders is the amount of all future cash flows which are distributed to them. The distributions are defined using following accounting identity (a simplified model of Nikolov and Whited (2014)):

$$d(k,k',\varepsilon) \equiv (1-\tau)\varepsilon k^{\theta} + \delta k\tau - I - \Psi(I,k) + f(1-\phi), \tag{91}$$

where the first two terms represent after-tax operating profits, the second and third term account for cash outflows due to an investment and its cost, and the last term captures any outside financing that can be used to distribute value to shareholders. Furthermore, assume that the manager acts in the shareholder's best interest.

Optimal Managerial Decision

The manager chooses the capital stock, k, through investment, i, each period to maximize the present value of the firm, that is, all future distribution to shareholders. The corresponding Bellman equation for the problem is then:

$$V(k,\varepsilon) = \max_{k'} \left\{ d(k,k',\varepsilon) + \frac{1}{1+r} \int V(k,'\varepsilon')dq(\varepsilon',\varepsilon) \right\} \tag{92}$$

The policy assigns the optimal investment behaviour to a given state of the environment (characterized by innovations/shocks, ε) in order to maximise the capital stock in the long term.

Intertemporal Asset Allocation

Example 2: Intertemporal asset allocation

Assume that investors seek to maximize expected "lifetime" utility over consumption C_t and bequest W_T:

$$\mathbb{E}\left[\sum_{t=0}^{T-1} u(C_t, t) + F(W_T)\right] \tag{93}$$

Where:

- F is increasing and concave
- u is continuous in c and t and increasing and concave in c. Further assume that either u or F is strictly concave

Furthermore, the investors' information set is generated by a vector $\mathbf{X}_t$ of K state variables.

Intertemporal Asset Allocation

Suppose that at date t an investor has wealth W_t. She can choose to consume C_t out of it, which leaves her with $W_t - C_t$ to invest. She can invest fractions $\mathbf{w}_t$ of this amount in the N risky assets and the remaining fraction $1 - \mathbf{w}_t'\mathbf{1}$ in the riskless account at rate $r_{f,t}$. Hence, her wealth next period will be:

$$W_{t+1} = (W_t - C_t)\underbrace{(\mathbf{w}_t'(\mathbf{r}_{t+1} - r_{f,t+1}\mathbf{1}) + r_{f,t+1})}_{\equiv r_{p,t+1}} \tag{94}$$

For convenience, denote the set of budget-feasible strategies by $\mathscr{S}(\mathbf{X}_t)$. Putting everything together the investor's problem is to find the policy, i.e., choose a sequence of vectors of weights $\{\mathbf{w}_t\}_{t=0}^{T-1}$ and a sequence of consumptions $\{C_t\}_{t=0}^{T-1}$, to maximize her lifetime expected utility:

$$\max_{\{\mathbf{w}_t\}_{t=0}^{T-1},\{C_t\}_{t=0}^{T-1}} \mathbb{E}\left[\sum_{t=0}^{T-1} u(C_t,t) + F(W_T)\right] \tag{95}$$

$$\text{subject to} \quad (\mathbf{w}_t, C_t) \in \mathscr{S}(\mathbf{X}_t), \quad \mathbf{X}_{t+1} = L(\mathbf{X}_t) \quad \forall t \in \{0,\ldots,T-1\}$$

Finding the Optimal Policy

In both of the above settings one could use reinforcement learning to obtain the optimal policy:

- the manager who chooses how much to invest each period in order to maximize firm value, depending on the state of the environment (investment opportunities).
- the investor choosing the optimal trading/investment strategy in order to maximize her wealth, for instance, by rotating between value, size, and momentum strategies, or by switching between active and passive investing, depending on the state of the market environment.

Preprocessing - Data Samples

The performance of a neural network can only be evaluated when cast on data it has never seen before. Therefore, it is required that the data be split into two sets:

- **Taining set**: A subset, A, of the original data from which the neural network learns (in-sample).
- **Validation set**: A subset, B, of the original data to asses whether the neural network produces a good fit for data it has not seen before (out-of-sample).
- The two subsets have no common observations, i.e., $\{A\} \cap \{B\} = \emptyset$

For time series data one must additionally specify a:

- **Data window**: A chronological sequence of input variables from which one future observation is predicted.

Most observations of the original data should be assigned to the training set while only a minority of observations should be assigned to the validation set, for instance, 90%/10%. A good neural network produces a good fit, i.e., a small error, for both sets. Partitioning your data into a training and validation set is the simplest form of cross-validation.

Preprocessing - Input Variables

Most of the times, your dataset will contain input variables highly varying in magnitudes, units and range. But since, many machine learning algorithms use a distance based error measure, this is a problem. Hence, it is of towering importance that the input data be standardized in some way:

- **Standardization**: Standardize the values to have a zero mean and unit variance
- **Normalization**: Rescale the values to a domain of $[0, 1]$
- The data is naturally homogeneous

Command

In `Python` standardizing variables is facilitated using sklearn.preprocessing.StandardScaler, while normalizing variables is facilitated using sklearn.preprocessing.normalize

Preprocessing - Labelling

In some cases one might be interested into classifying time series observations into classes and assign labels. Following Lopez De Prado (2018), the following labelling schemes are especially suited for returns:

- **Binary encoding**: Assigning a value of 1 to positive returns, and 0 otherwise.
- **Quantile encoding**: Assigning return observations to a specified number of quantiles.
- **Sigma encoding**: Rather than prespecifying the number of possible quantiles, sigma encoding lets the return stream decide the set of values and uses the standard deviation to discretize the return sequence. The number of values in the set is defined by $\left\lceil \frac{max\{r\}-min\{r\}}{\sigma} \right\rceil$ and each return is assigned to a value according to $\left\lfloor \frac{r_t-min\{r\}}{\sigma} \right\rfloor$. In effect, we assign 0 to $r_t \in [min\{r\}, min\{r\}+\sigma)$, 1 to $r_t \in [min\{r\}+\sigma, min\{r\}+2\sigma)$, and so on.

Preprocessing - Data Shuffling

It is important that the training data is shuffled, i.e., it comes in a non-specific, random order. Elsewise, the neural network is subject to learning an order that does not arise from the data itself (ascending order, alphabetical, etc.) This even pertains to time series with intrinsic ordering. However, this does not mean that the time series observations itself should be shuffled. Rather, it means that one should feed the data windows in a random pattern:

1.

25.03	26.03	27.03	28.03
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

$\longrightarrow$ **1.**

25.03	26.03	27.03	28.03
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

2.

26.03	27.03	28.03	29.03
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

$\longrightarrow$ **2.**

11.08	12.08	13.08	14.08
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

3.

27.03	28.03	29.03	30.03
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

$\longrightarrow$ **3.**

05.05	06.05	07.05	08.05
x_{t-2}	x_{t-1}	x_t	$\hat{y}_{t+1}$

$\vdots$

Preprocessing - Class Weights

Another issue often encountered in machine learning are imbalanced datasets. This means that a certain value of a variable is represented much more often in the data sample than other values of the same variable. Depending on the question at hand, this can become a substantial issue. One remedy are class weights.

- Class weights specify that a certain realization of a variable should have more weight when training neural networks than other variables.
- This is achieved by giving them more weighted impact on the fitting error such that they cannot be neglected just because they are rare.

For instance, if x_1 is represented twice as much as x_2, then x_2 should have a class weight of $2/3$ and x_1 a class weight of $1/3$.

Command

In `Python` adding class weights is an option in the model fitter command in Tensorflow and Keras. It is implemented using model.fit(...,class_weight={class_weight_dictionary},...).

Preprocessing - Stationarity vs. Memory

What makes time series non-stationary is memory, the presence of a long history of previous levels that shift the series mean over time.

- Taking returns undoubtedly renders a time series stationary.
- This happens at the cost of erasing all its memory.
- Being able to exploit memory in time series is a powerful ability found in some ML algorithms .

Hence, one should desire to give up only so much memory such that the time series becomes stationary while keeping as much memory as possible at the same time. To that end, Lopez De Prado (2018) suggests fractional differencing, which allows the order of integration to take on any real positive number and not only integers, and idea that dates back at least to Hosking (1981).

Preprocessing - Stationarity vs. Memory

Formally, using a lag operator L applied to a time series $\{x_t\}$ we have that $x_{t-k} = L^k x_t$ for any integer satisfying $k \geq 0$, where k denotes the order of lag. For example, $(1-L)^2 = 1 - 2L + L^2$, so that $(1-L)^2 x_t = x_t - 2x_{t-1} + x_{t-2}$. From the fact that:

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k, \tag{96}$$

it follows that for any real number d we have:

$$(1+b)^d = \sum_{k=0}^{\infty} \binom{d}{k} b^k, \tag{97}$$

the binomial series.

Preprocessing - Stationarity vs. Memory

In order to achieve a fractional model we map (97) onto our lag operator and allow for d to be any real number and use the following binomial series expansion:

$$
\begin{aligned}
(1-L)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-L)^k &= \sum_{k=0}^{\infty} (-L)^k \prod_{i=0}^{k-1} \frac{d-i}{k-i} \\
&= \sum_{k=0}^{\infty} (-L)^k \frac{\prod_{i=0}^{k-1}(d-i)}{k!} \\
&= 1 - dL + \frac{d(d-1)}{2!}L^2 - \frac{d(d-1)(d-2)}{3!}L^3 + \ldots
\end{aligned} \tag{98}
$$

Preprocessing - Stationarity vs. Memory

If we have a time series $\{x_t\}$ we can transform it into a fractionally differentiated time series of order d by defining it as a dot product of realizations of $\{x_t\}$ and weights w_k obtained from the binomial series expansion in equation (98) such that:

$$\{\tilde{x}_t^d\} = \left\{ \sum_{k=0}^{\infty} w_k x_{t-k}, \sum_{k=0}^{\infty} w_k x_{t-1-k}, \ldots, \sum_{k=0}^{\infty} w_k x_{t-l-k} \right\} \tag{99}$$

where $\{\tilde{x}_t^d\}$ denotes the fractionally differentiated time series of $\{x_t\}$ with order d and where the weight vector correspond to:

$$\boldsymbol{w} = \left\{ 1, -d, \frac{d(d-1)}{2!}, -\frac{d(d-1)(d-2)}{3!}, \ldots, (-1)^k \prod_{i=0}^{k-1} \frac{d-i}{k!} \right\} \tag{100}$$

Preprocessing - Stationarity vs. Memory

When d is a positive integer, the memory beyond that point is purged since $\prod_{i=0}^{k-1} \frac{d-i}{k!} = 0, \forall k > d$. If the time series $\{x_t\}$ are log prices, when setting d to one it becomes obvious from equation (100) that the resulting weights are $w = \{1, -1, 0, 0, \ldots\}$ and in effect, the time series maps to a series of log returns, $(1-L)x_t = x_t - Lx_t = x_t - x_{t-1}$, where all memory beyond a lag of one vanishes. In order to find the optimal order of differencing, one can gradually increase the order until the augmented Dickey-Fuller test rejects the null hypothesis of non-stationarity.

- Lopez De Prado (2018) finds that many securities are stationary when d is between 0.3 and 0.6 which seems to imply that taking standard returns casts away valuable memory of the time series.

The augmented Dickey-Fuller test can be revisited here: ▸ Augmented Dickey-Fuller test

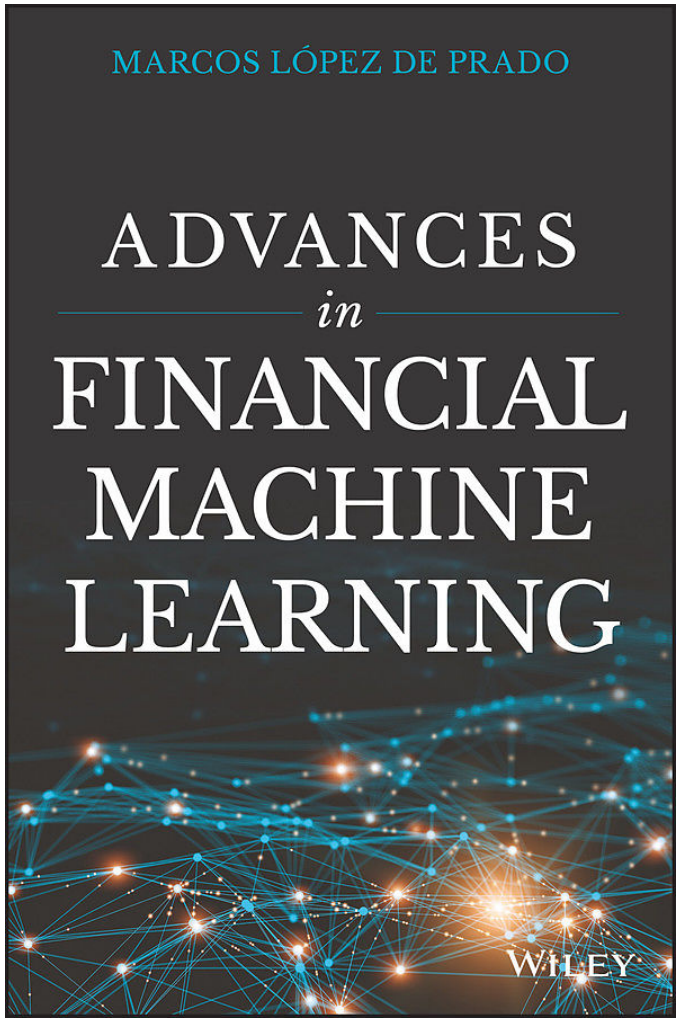
Overfitting and Cross-validation

A model is said to be overfit when it performs well in-sample but performs poorly out-of-sample. The discrepancy between in-sample and out-of-sample performance is known as generalization error. Specifically, an error estimation, $\hat{\varepsilon}$, is made for the model when training. However, this only gives us an idea about how well our model does on data used to train it. The problem with evaluating this training error is that it does not give an indication of how well the model will generalize to an independent and unseen data set. Evaluating whether the model performs well on unseen data is known as cross-validation. Popular cross-validation approaches are:

- **Holdout method**: Splitting the data into a training and validation set.
- **K-Fold cross-validation**: The data is divided into K equally sized subsets. Now, the holdout method is repeated K times, such that each time, one of the K subsets is used as the validation set and the other $K-1$ subsets form the training set.
- **Repeated random sub-sampling**: Similar to K-fold cross-validation, but this time in each training iteration a random fraction of observations in the dataset are randomly set aside for training.

Reading Tip

López de Prado, M. (2018), Advances in Financial Machine Learning, Wiley



Bibliography

References I

Baker, Malcolm, and Jeffrey Wurgler, 2006, Investor sentiment and the cross-section of stock returns, *Journal of Finance* 61, 1645–1680.

Black, Fischer, Michael Jensen, and Myron Scholes, 1972, *The Capital Asset Pricing Model: Some empirical tests*.

Bollerslev, Tim, 1986, Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics* 31, 307–327.

Bollerslev, Tim, 1990, Modelling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model, *The Review of Economics and Statistics* 72, 498–505.

Campbell, John Y., Andrew W. Lo, and A. Craig MacKinley, 1997, *The econometrics of financial markets*.

Carhart, Mark M., 1997, On persistence in mutual fund performance, *The Journal of Finance* 52, 57–82.

Engle, Robert, 2002, Dynamic conditional correlation – A simple class of multivariate Generalized Autoregressive Conditional Heteroskedasticity models, *Journal of Business and Economic Statistics* 20, 339–350.

Engle, Robert F, 1982, Autoregressive conditional hetersokedasticity with westimates of the variance of United Kingdom inflation, *Econometrica* 50, 987–1007.

Fama, Eugene F., 1970, Efficient capital markets: A review of theory and empirical work, *The Journal of Finance* 25, 383.

References II

Fama, Eugene F, and Kenneth R French, 1992, The cross-cection of expected stock returns, *Journal of Finance* 47, 427–465.

Fama, Eugene F., and James D. MacBeth, 1973, Risk, return, and equilibrium: Empirical tests, *Journal of political economy* 81, 607–636.

Frazzini, Andrea, and Lasse Heje Pedersen, 2014, Betting against beta, *Journal of Financial Economics* 111, 1–25.

French, Kenneth R., 1980, Stock returns and the weekend effect, *Journal of Financial Economics* 8, 55–69.

Harvey, Campbell R., Yan Liu, and Heqing Zhu, 2016, ... and the cross-section of expected returns, *Review of Financial Studies* 29, 5–68.

Hendershott, Terrence, Dmitry Livdan, and Dominik Rösch, 2018, Asset Pricing: A Tale of Night and Day, *SSRN Electronic Journal* .

Hosking, J. R.M., 1981, Fractional differencing, *Biometrika* 68, 165–176.

Lintner, John, 1965, Security prices, risk, and maximal gains from diversification, *The Journal of Finance* 20, 587–615.

Lopez De Prado, Marcos, 2018, *Advances in financial machine learning*, first edition (Wiley & Sons, Inc., Hoboken, New Jersey).

Markowitz, Harry, 1952, Portfolio selection, *The Journal of Finance* 7, 77.

References III

Newey, Whitney K, and Kenneth D West, 1987, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, Technical Report 3.

Nikolov, Boris, and Toni M. Whited, 2014, Agency conflicts and cash: Estimates from a dynamic model, *Journal of Finance* 69, 1883–1921.

Roll, Richard, 1977, A critique of the asset pricing theory's tests. Part I: On past and potential testability of the theory, *Journal of Financial Economics* 4, 129–176.

Rozeff, Michael S., and William R. Kinney, 1976, Capital market seasonality: The case of stock returns, *Journal of Financial Economics* 3, 379–402.

Savor, Pavel, and Mungo Wilson, 2014, Asset pricing: A tale of two days, *Journal of Financial Economics* 113, 171–201.

Sharpe, W. F., 1964, Capital asset prices: A theory of market equilibrium under conditions of risk, *The Journal of Finance* 19, 425–442.

Appendix

Event-Studies

An event study measures the effect of some company-specific information (usually the announcement of a corporate decision) on the prices of stocks of the company.

Event studies are tests of market efficiency. If abnormal returns after the announcement are significantly different from zero, markets are not semi strong efficient. Hence, through event studies, we can check for profitable trading strategies.

If the market is efficient, event studies allow estimating the economic impact of corporate decisions. We care about the "average" response to announcements, not the response to some specific company's announcement.

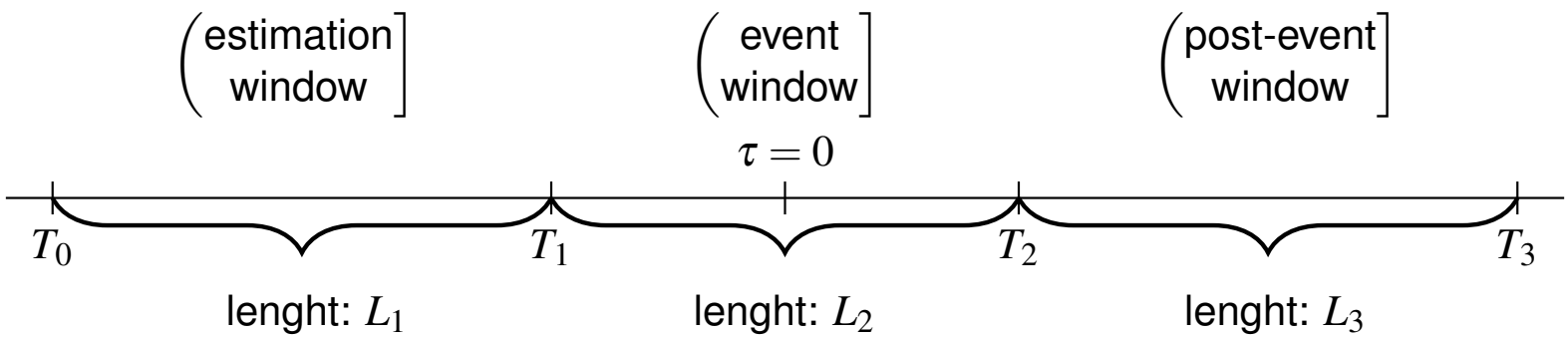
Corporate decisions have been the focus of lots of event studies (mainly in the US):

- earnings announcement
- IPOs
- share repurchases
- M&A
- etc.

Event-Studies

General procedure

1. Find a sample of firms that have announced the same news (event) in the past
2. Define time $\tau = 0$ as the event date, i.e. the announcement in a widely disseminated publication
3. Define the windows useful for the analysis around the event date



4. Over the estimation window, estimate the parameters of the normal return model. $\mathbb{E}[r_{i,t}]$ is the normal return that the stock would experienced if there were no event

Event-Studies

5. Over the event window, estimate the series of abnormal returns $\forall i, i \in N$:

$$AR_{i,t} = r_{i,t} - \mathbb{E}[r_{i,t}], \qquad \text{for } t = T_1 + 1, \ldots, T_2 \tag{101}$$

6. Compute, on each day in the event window, the average abnormal return across all N firms in the sample

$$AAR_t = \frac{1}{N} \sum_{i=1}^{N} AR_{i,t}, \qquad \text{for } t = T_1 + 1, \ldots, T_2 \tag{102}$$

7. Compute the Cumulative Average Abnormal Return (CAAR)

$$CAAR_t = \sum_{t=T_1+1}^{T_2} AAR_t, \qquad \text{for } t = T_1 + 1, \ldots, T_2 \tag{103}$$

8. Test for statistical significance and interpret the results

Event-Studies

Example: Impact of earnings announcements on stocks constituting the Dow Jones index, Campbell et al. (1997)

Announcement of quarterly results of 30 firms over 1988-93 (600 announcements). 4 types of information have been brought together for the study:

- date of announcement
- security prices
- announced earnings
- forecasted earnings

To examine the impact of the announcement on prices, three groups are constructed:

- **Good news:** $Result > 1.025\,\mathbb{E}[Resullt]$ (189)
- **No news:** $0.975\,\mathbb{E}[Resullt] < Result < 1.025\,\mathbb{E}[Resullt]$ (173)
- **Bad news:** $0.975\,\mathbb{E}[Resullt] < Result$ (238)

Event-Studies

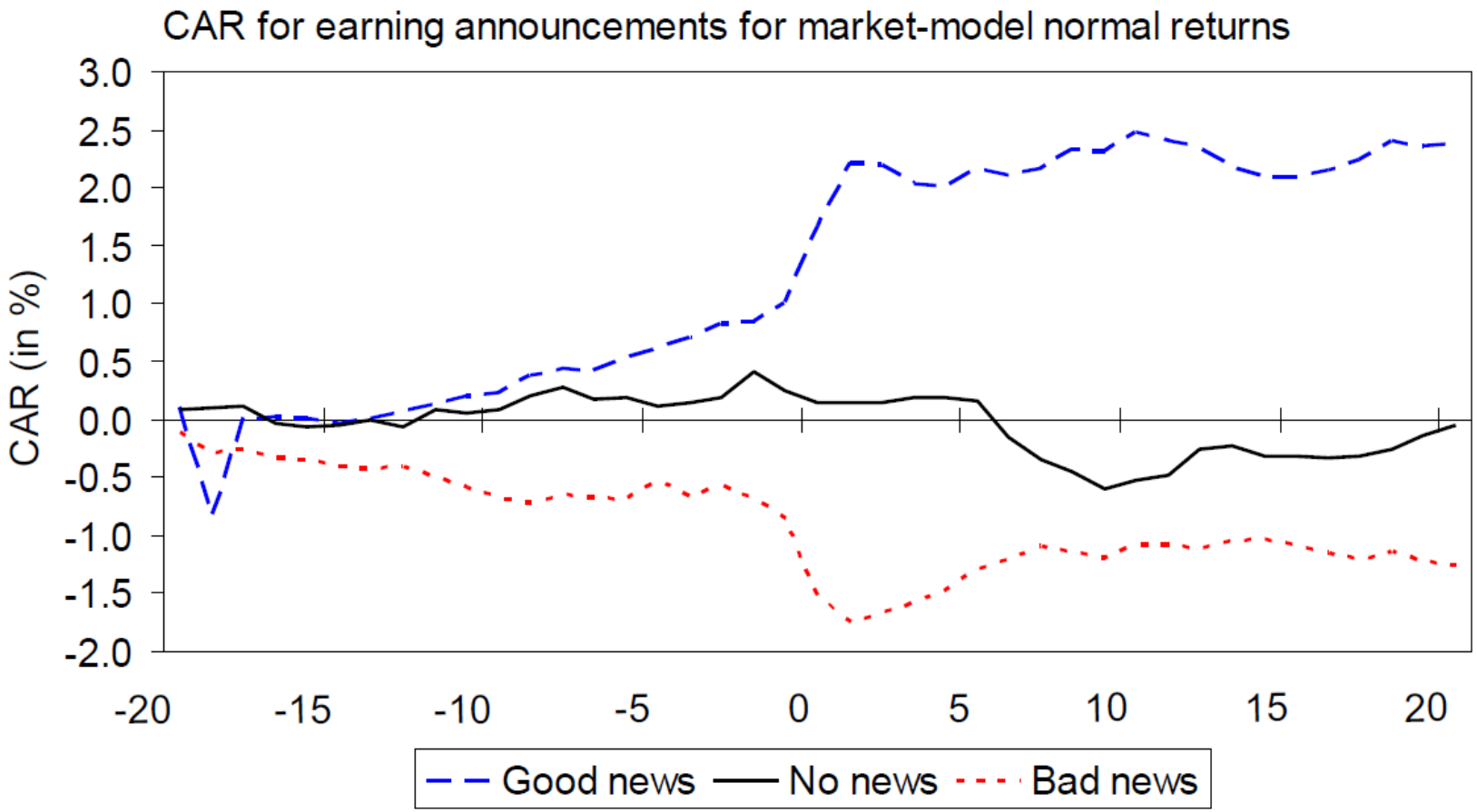
Other specifications:

- Frequency of observations: daily
- Event window: (-20,+20)
- Estimation period: 250 trading days
- Model for normal returns: market and constant mean return model

Main results: Earnings announcements convey information for the valuation of firms

- The market gradually learns about the forthcoming announcement
- The sample average abnormal return is 0.965% (with t-stat = 9.28) for good news firms and –0.679% (with t-stat = 6.93) for bad-news firms. The null hypothesis that the event has no impact is strongly rejected. The day after the event, we also have a slightly significant abnormal return, because some earnings are announced after the market close
- Beyond the day after the event, there is no abnormal return, implying semi strong efficiency

Event-Studies



Event-Studies

To measure abnormal returns, we need a model of "normal" returns. The parameters are estimated over the estimation window $(T_0 + 1, T1)$. Hence, two decisions have to be made:

- The choice of the market equilibrium model
- The relevant estimation window to estimate the parameters of the equilibrium model, i.e., L_1

Here, we look at two different normal return models:

- The constant mean return model
- The market model

Event-Studies

Constant mean return model

Assume that $\mathbb{E}[r_{i,t}] = c_i$, the average return on stock i over the estimation window. Hence, the abnormal return is:

$$AR_{i,t} = r_{i,t} - c_i \tag{104}$$

and assume that the expected rate of return is constant over time. One drawback is that this type of model does not incorporate contemporaneous market information.

Event-Studies

Market model

Assume that returns are generated by the following model:

$$r_{i,t} = \alpha_i + \beta_i r_{m,t} + \varepsilon_{i,t}, \qquad \text{with } \mathbb{E}[\varepsilon_{i,t}] = 0 \text{ and } \mathbb{V}[\varepsilon_{i,t}] = \sigma_i^2 \tag{105}$$

where $r_{m,t}$ denotes the market return. We estimate $\hat{\alpha}_i$ and $\hat{\beta}_i$ over the estimation window $(T_0 + 1, T1)$ with lenght L_1. Then, we estimate abnormal returns over the event window, i.e., for each $t \in \{T_1 + 1, \ldots, T_2\}$

$$AR_{i,t} = r_{i,t} - \hat{\alpha}_i - \hat{\beta}_i r_{m,t} \tag{106}$$

This approach does incorporate contemporaneous market information. Furthermore, it is assumed that the parameters of the return generating process are not affected by the event.

Note: This is not the CAPM since we do not have any restriction on the intercept. One could use the CAPM as the normal return model theoretically, but the additional constraint of a zero-intercept (or risk-free rate) leads to a higher error term $\varepsilon_{i,t}$ and thus a higher variance.

Event-Studies

Model Precision

The variance of the constant mean return model is

$$\tilde{\sigma}_i^2 = \mathbb{V}[r_{i,t} - c_i] = \mathbb{V}[r_{i,t}] \tag{107}$$

The variance of the market model is

$$\sigma_i^2 = \mathbb{V}[r_{i,t} - \hat{\alpha}_i - \hat{\beta}_i r_{m,t}] = \mathbb{V}[r_{i,t}] - \hat{\beta}_i{}^2 \mathbb{V}[r_{m,t}] \tag{108}$$

From the fact that

$$R^2 = \frac{\mathbb{V}[\hat{r}_{i,t}]}{\mathbb{V}[r_{i,t}]} = \frac{\hat{\beta}_i{}^2 \mathbb{V}[r_{m,t}}{\mathbb{V}[r_{i,t}]} \tag{109}$$

it follows that

$$\sigma_i^2 = \mathbb{V}[r_{i,t}] - R^2 \mathbb{V}[r_{i,t}] \tag{110}$$

$$(1 - R^2)\tilde{\sigma}_i^2 \leq \tilde{\sigma}_i^2 \tag{111}$$

Conclusion: The market model is expected to lead to more precise results.

Event-Studies

Now we want to test if the AAR and the CAAR are significantly different from 0.

a) For one event (AR)

– Compute the AR over both the estimation window and the event window

$$AR_{i,t} = r_{i,t} - \mathbb{E}[r_{i,t}] = \varepsilon_{i,t} \tag{112}$$

with

$$\hat{\mathbb{V}}[AR_{i,t}] = \hat{\mathbb{V}}[\varepsilon_{i,t}], \qquad \text{for } t \in \{T_0+1, \ldots, T_1\}, \tag{113}$$

i.e., the variance of abnormal returns over the estimation window.

– Then, the test of the null hypothesis, $H_0 : \mathbb{E}[AR_{i,t}] = 0$, is based on the assumptions that 1) the variance is constant over time, 2) there is no measurement error in the market model parameters, and 3) that returns are $i.i.d.N(\mu, \sigma^2)$.

Event-Studies

a) For one event (AR)

– Under these assumptions the t-statistic for a single AR is

$$t(AR_{i,t}) = \frac{AR_{i,t}}{\sqrt{\hat{\mathbb{V}}[AR_{i,t}]}} \sim t_{(L_1)} \tag{114}$$

When L_1 is large enough (estimation window), the test statistic follows a normal distribution.

– Now, the test of the null, $H_0 : \mathbb{E}[CAR_i] = 0$, is based on the following test statistic

$$t(CAR_i) = \frac{\sum_{t=T_1+1}^{T_2} AR_{i,t}}{\sqrt{\hat{\mathbb{V}}[CAR_i]}} \sim t_{(L_2-1)} \qquad \text{with } \hat{\mathbb{V}}[CAR_i] = \sum_{t=T_1+1}^{T_2} \mathbb{V}[AR_{i,t}] \tag{115}$$

– Tests on one stock are not very interesting, however, because we want to have an idea over the whole sample of companies that have experienced the event.

Event-Studies

b) For multiple events (AAR)

– The average abnormal return across all N events is defined as

$$AAR_t = \frac{1}{N}\sum_{i=1}^{N} AR_{i,t} \qquad \text{with } \hat{\mathbb{V}}[AAR_t] = \frac{1}{N^2}\sum_{i=1}^{N} \hat{\mathbb{V}}[AR_{i,t}] \tag{116}$$

– If the AR are independent, the test statistics to test $H_0 : \mathbb{E}[AAR_t] = 0$ is

$$t(AAR_t) = \frac{AAR_t}{\sqrt{\hat{\mathbb{V}}[AAR_t]}} \sim t_{(N-1)} \tag{117}$$

Event-Studies

c) For multiple events across time (CAAR)

- We may also test the null hypothesis that $H_0 : \mathbb{E}[CAAR] = 0$
- The cumulative average abnormal return across N events and L_2 periods is defined as

$$CAAR = \sum_{t=T_1+1}^{T_2} AAR_t \qquad \text{with } \hat{\mathbb{V}}[CAAR] = \sum_{t=T_1+1}^{T_2} \mathbb{V}[AAR_t] \tag{118}$$

- To test $H_0 : \mathbb{E}[CAAR] = 0$ the test statistic amounts to

$$\frac{CAAR}{\sqrt{\hat{\mathbb{V}}[CAAR]}} \sim t_{(L_2-1)} \tag{119}$$

Return to the main slides ◂ main slides