

The Empirical Application of Finance with R and ChatGPT

Carlo A. Favero¹

October 8, 2023

¹Bocconi University, IGER and CEPR

Contents

Preface	1
1 Empirical Models in Finance	3
1.1 Introduction	3
1.2 The distribution of future returns in finance.	4
1.2.1 Standard Portfolio Theory	4
1.2.2 Risk Parity Portfolios	5
1.3 Predicting returns: The Econometric Modelling Process	6
1.3.1 The Challenges of Financial Econometrics	7
1.4 Empirical Modelling of Asset Prices	8
1.4.1 The view from the 1960s: Efficient Markets and CER	8
1.5 Empirical Challenges to the traditional model	11
1.5.1 The time-series evidence on expected returns	12
1.5.2 Anomalies	13
1.5.3 The Cross-section Evidence on Expected Returns	14
1.5.4 The behaviour of returns at high-frequency	15
1.6 The Implications of the new evidence	16
1.6.1 Asset Pricing with Predictable Returns	16
1.7 Quantitative Risk Management and returns at high-frequency	18
1.8 Predictive Models in Finance: a General Representation.	18
2 Asset Prices and Returns	21
2.1 Introduction	21
2.2 Returns	21
2.2.1 Simple and log Returns	21
2.2.2 Statistical models for asset prices and returns.	22
2.2.3 Multi-period returns and annualized returns	23
2.2.4 Working with Returns	24
2.3 Stock and Bond Returns	26
2.3.1 Stock Returns and the dynamic dividend growth model	26

2.3.2	Bond Returns: Yields-to-Maturity and Holding Period Returns	30
2.4	Going to the data with R	33
2.4.1	Getting Started	34
2.4.2	Data Objects in R	36
2.4.3	Data Handling in R	37
2.4.4	Data Exploration and Graphics	39
2.4.5	Interacting with Chat GPT	50
2.5	Appendix: The Data	59
3	The Modelling Process at Work: the CER model	61
3.1	Introduction	61
3.2	Model Specification: the Constant Expected Return Model	61
3.2.1	Stocks for the long run	63
3.3	Model Estimation	64
3.3.1	Parameters Estimation in a linear model	65
3.4	Model Simulation: Monte-Carlo and Bootstrap Methods	67
3.5	The CER model at work with R	68
3.5.1	Asset Allocation with the CER	68
3.5.2	Model Simulation with the CER: backtesting and VaR	74
4	Factor Models for Asset Prices and Returns	81
4.1	Introduction: Factor Models and Reduction in Dimensionality	81
4.2	Factor Models:Time-Series Representation	82
4.3	Factor Models: Cross-Sectional representation	83
4.4	Factor-based Portfolios and Factor Exposures	83
4.5	A single factor model:The CAPM	84
4.5.1	Asset Allocation with the CER and the CAPM in R	85
4.6	Validating Factor Models	92
4.6.1	Which Factors ?	93
4.7	Factor Models with Predictability	93
4.7.1	An illustration with R	97
5	Models for Risk Measurement	105
5.1	Risk Measurement	105
5.1.1	Value at Risk (VaR)	105
5.2	VaR without predictability	106
5.2.1	VaR with the CER	106
5.2.2	VaR with the CAPM:	106
5.3	The Evidence from high-frequency data	107
5.4	A general model for high-frequency data	109

5.4.1	GARCH Models for Heteroscedasticity	109
5.4.2	GARCH Properties	109
5.4.3	GARCH Forecasting	110
5.4.4	Testing for GARCH	110
5.5	Estimation of GARCH Models	111
5.5.1	Quasi MLE Estimation	112
5.6	From GARCH to VaR	112
5.6.1	GARCH with factors	113
5.7	Measuring risk: an illustration with R	113
5.8	Backtesting VaR	119
5.8.1	Unconditional Coverage Testing	119
5.8.2	Independence Testing	120
5.8.3	Conditional Coverage Testing	122
5.8.4	Backtesting VaR in R	122
5.9	Beyond GARCH: non-linear and multivariate models	124

Preface

This set of lecture notes is meant to support a course illustrating the practical application of the Theory of Finance to real-life asset allocation problems. Empirical modelling in finance is introduced by illustrating its working and its historical evolution. The process of translating financial theory into action on data is driven by building predictive models for asset prices and returns. Basic models will be explored, and programming will emerge as an essential prerequisite for data manipulation. We will acquaint ourselves with the statistical software R and exhibit the application of theoretical concepts to financial data, illustrated by sample programs, exercises, and corresponding solutions. The main focus is on the hands-on implementation of this approach using actual data, utilizing specific models to exemplify its practicality. Through the notes we shall also show how Chat GPT could be used to help in doing Empirical Finance with R.

Lectures based on these notes should actively involve the demonstration and discussion of R codes. Attendees are expected to engage in real-time coding on their laptops during lectures.

Chapter 1

Empirical Models in Finance

1.1 Introduction

Predicting the distribution of returns of financial assets is a task of primary importance for identifying desirable investments, performing optimal asset allocation within a portfolio, as well as measuring and managing portfolio risk. Optimal asset management depends on the statistical properties of returns at different frequencies. Portfolio allocation, i.e., the choice of optimal weights to be attributed to the different (financial) assets in a portfolio, is typically based on a long-horizon perspective, while the measurement of risk of a given portfolio takes typically a rather short-horizon perspective. This means that a long-run investor decides her optimal portfolio allocation on the basis of the (joint) distribution of the returns of the relevant (i.e., from some pertinent asset menu from which to choose) financial assets at low frequency. However, the monitoring of the daily risk of a portfolio normally depends on the statistical properties of the distribution of returns at high frequencies.

As the distribution of future returns is not observable, the implementation of the theory of finance requires the estimation of the distribution of future expected returns. This distribution is derived by using the available data to build a model and then by simulating the model to build artificial observations from which a model-based distribution of future returns is derived.

This project, in its characteristically applied nature, is designed to illustrate the statistical techniques to perform the analysis of time series of financial assets and returns at different frequencies and their utilization to build models for asset management and performance evaluation, portfolio allocation, and financial risk management.

The relevant concepts will be introduced and their application will be discussed by using a set of programs written using R, a free software environment for statistical

computing and graphics, specifically designed for each chapter. Draft codes for the solutions of the exercises, which are designed to allow the reader to understand how the different econometric techniques could be put to work, are made available on the book webpage. The main emphasis will be given to the application of econometric techniques, readers interested in the statistical properties of the estimation and the simulation of econometric techniques applied here should refer to appropriate textbooks. All empirical applications will be based on publicly available databases of US data. Note that there are three relevant dimensions of the data on financial returns: time series, cross-section and the horizon at which returns are defined. In general, we shall define $r_{t,t+k}^i$ as the returns realized by holding between time t and time $t+k$, the asset i . So the t index captures the time-series dimension, the i index the cross-sectional dimension, and the k index the horizon dimension.

1.2 The distribution of future returns in finance.

To illustrate the relevance of the distribution of future returns in finance we consider the problem of the optimal choice at time t of the weights to be given to n risky assets in building an optimal portfolio between time t and time $t+k$. We shall consider two alternative approaches to choosing weights: Standard Portfolio Theory and Risk Parity Portfolios. In these applications estimates of the first two moments of the distribution of future returns are necessary for the practical implementation of optimal portfolios.

1.2.1 Standard Portfolio Theory

Let's denote with \mathbf{r} the random vector of linear total returns from time t to time $t+k$ from a given menu of N risky assets for interval $[t, t+k]$, $\mathbf{r} \sim \mathcal{D}(\mu, \Sigma)$

Given a degree of risk aversion λ , a standard *mean-variance* description of this allocation problem is the following:

$$\max_{\mathbf{w}} (1 - \mathbf{w}'\mathbf{e}) r^f + \mathbf{w}'\mu - \frac{1}{2}\lambda(\mathbf{w}'\Sigma\mathbf{w})$$

where $E[\mathbf{r}] = (1 - \mathbf{w}'\mathbf{e}) r^f + \mathbf{w}'\mu = r^f + \mathbf{w}'(\mu - r^f\mathbf{e})$ and $Var[\mathbf{w}'\mathbf{r}] = \mathbf{w}'\Sigma\mathbf{w}$. First-order conditions (FOCs) are necessary and sufficient and define the following system of N linear equations in N unknowns, the portfolio weights $\mathbf{w} \in \mathcal{R}^N$:

$$(\mu - r^f\mathbf{e}) - \lambda\Sigma\mathbf{w} = \mathbf{0}.$$

Solving the FOCs yields:

$$\hat{\mathbf{w}} = \frac{1}{\lambda}\Sigma^{-1}(\mu - r^f\mathbf{e}),$$

which makes clear that optimal weights depend on preferences and the first two moments of the distribution of future returns.

Consider now the special case in which $\hat{\mathbf{w}}'\mathbf{e} = 1$, that is no investment in the risk-free asset is allowed. The optimal portfolio in this case is the famous *tangency portfolio* which depends exclusively on the first two moments of the distribution of future returns:

$$\mathbf{e}'\hat{\mathbf{w}} = \frac{1}{\lambda}\mathbf{e}'\Sigma^{-1}(\mu - r^f\mathbf{e}) = 1 \implies \lambda = \mathbf{e}'\Sigma^{-1}(\mu - r^f\mathbf{e})$$

$$\hat{\mathbf{w}}^T = \frac{\Sigma^{-1}(\mu - r^f\mathbf{e})}{\mathbf{e}'\Sigma^{-1}(\mu - r^f\mathbf{e})}$$

Similarly, when the target is to find the minimum variance portfolio, we have: In case the target is to find the minimum variance portfolio:

$$\min_{\mathbf{w}}(\mathbf{w}'\Sigma\mathbf{w})$$

subject to

$$\mathbf{w}'\mathbf{e} = 1$$

the solution will be:

$$\mathbf{w} = \frac{\Sigma^{-1}\mathbf{e}}{\mathbf{e}'\Sigma^{-1}\mathbf{e}}$$

In this case, only the second moment of the distribution of returns matters.

1.2.2 Risk Parity Portfolios

An alternative approach to building a portfolio is to choose weights in such a way that the contribution of each asset to the volatility of the portfolio is the same (risk parity). To determine optimal weights in this scenario, decompose the total variance of a portfolio in the sum of the contributions of each asset to the total portfolio variance:

$$\begin{aligned} Var[\mathbf{w}'\mathbf{r}] &= \sum_{i=1}^N w_i Cov(r_i, \mathbf{w}'\mathbf{r}) \\ \mathbf{w}'\Sigma\mathbf{w} &= \sum_{i=1}^N w_i(\Sigma\mathbf{w})_i \end{aligned}$$

the risk contribution of each asset to total risk can then be written as follows:

$$RRC_i = \frac{w_i(\Sigma\mathbf{w})_i}{\mathbf{w}'\Sigma\mathbf{w}}$$

Risk Parity Portfolios are constructed by choosing weights so that:

$$RRC_i = \frac{1}{N}$$

Figure 1 illustrates the difference between an equally weighted portfolio and a risk parity portfolio:

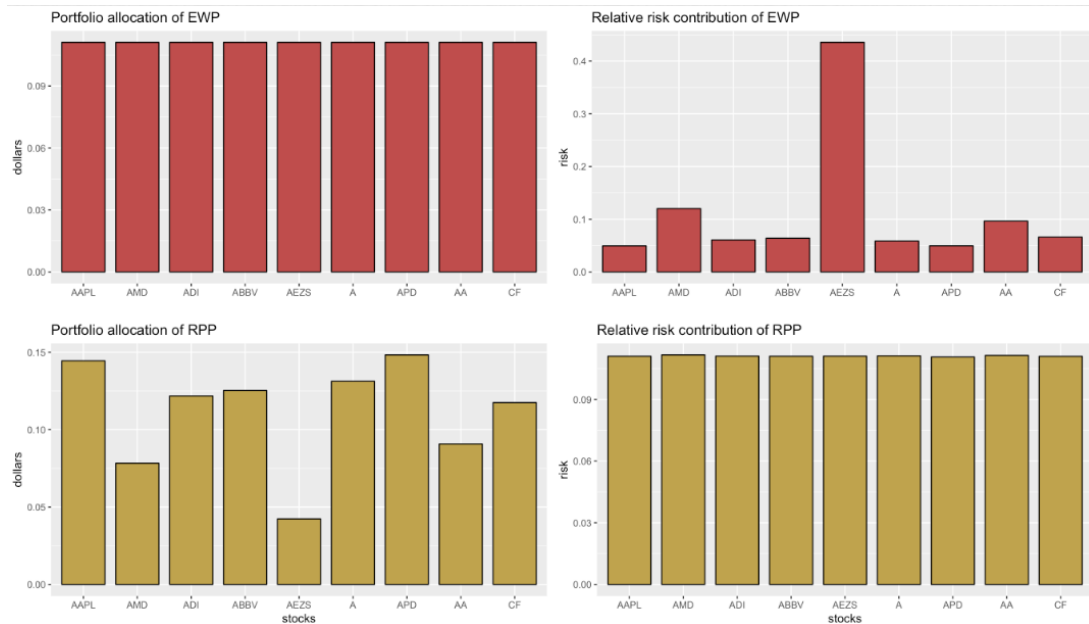


FIGURE 1.1. Portfolio allocation in Equal Weights Portfolios (EWP) and Risk Parity Portfolios (RPP)

Weights in the risk parity portfolio are fully determined by the variance-covariance matrix of the joint distribution of future returns.

1.3 Predicting returns: The Econometric Modelling Process

Econometrics uses the "past available data" to predict the future distribution of returns. In practice, the information contained in past data is used to build a model that describes the behaviour of returns; a model relates different returns and predictors by using some functional form and some unknown parameters that norm the interaction among relevant variables. The data are used to estimate the unknown parameters, using the general principle of minimizing the distance between the value predicted by the model for the variables of interest and those observed. After the unknown

1.3. PREDICTING RETURNS: THE ECONOMETRIC MODELLING PROCESS⁷

parameters have been estimated, model can be simulated to generate predictions for some moments or the entire distribution of returns. Ex-post comparison of model predictions and realized observation helps model validation. After validation, model simulation can be used for forecasting the distribution of returns for asset allocation and risk measurement. To sum up the Econometric Modelling Process involves several steps:

- Data collection and transformation
- Graphical and descriptive data analysis
- Model Specification
- Model Estimation
- Model Validation
- Model Simulation
- Use of the output of simulation for asset allocation and risk measurement

1.3.1 The Challenges of Financial Econometrics

In general, financial data are not generated by experiments, what is available to the econometrician are observational data, which are given. To investigate the effect of a medicine an investigator can take a set of patients and attribute them randomly to a "treatment" group and a "control" group. The medicine is then administered to the members of the treatment group while a "placebo" is given to the control group members. The effect of the medicine can then be measured by the difference in the average health of the members of the two groups after the administration of the treatment.

If a researcher is interested in using monetary policy to predict stock market returns, the only available data are those on monetary policy indicators and the stock market returns which are given and not generated by a controlled experiments.

Special issues arise in routinely in financial data that are different in special days (say, for example, the days of the FOMC meetings), that are affected by seasonality, trends and cycles. Moreover, rare-events affect financial returns and rare events are, by definition, not regularly observed. As [Taleb \(2012\)](#) forcefully stresses in his book *Antifragile*, absence of evidence in a given sample of data cannot be taken as evidence of absence.

Econometricians face questions of different natures: sometimes the interest lies in non-causal predictive modeling which can be handled by analyzing conditional expectations, while this is not sufficient to understand causation to which end correlation

and conditional expectations are little informative. One issue is to evaluate if the monetary policy stance helps to predict stock market returns, which is very different from establishing a causation from monetary policy to the stock market, as the evidence of correlation between monetary policy and the stock market might very well reflect the response of monetary policy to stock market fluctuations taken as an indicator of (present and future) economic activity. In the specification of models for financial data, it is crucial that the econometrician uses the same information that is available to agents operating in the market, i.e. that models are not affected by the so-called "look-ahead bias". To this end, the sample of available data is usually split into two subsamples: a training sample and a test sample. the training sample is used to get the model ready for use i.e. to estimate the unknown parameters, while the test sample is used for model simulation and evaluation.

1.4 Empirical Modelling of Asset Prices

There has been a remarkable evolution in the understanding and empirical modelling of asset prices and financial returns from the 1960s onwards. The view from the sixties was based on the Constant Expected Returns (CER) model and the CAPM, when a simple econometric model serves the purpose of modelling returns at all horizons and a one-factor model determines the cross-section of asset returns. Several empirical failures of this view have led to the development of Time-Varying Expected Returns (TVER) model where predictability becomes a factor and heterogeneity in predictability is introduced according to the horizon of returns.

1.4.1 The view from the 1960s: Efficient Markets and CER

The history of empirical finance starts with the "efficient market hypothesis" [Fama \(1970\)](#). This view, that dominated the field in the 1960s and 1970s, can be summarized as follows (see also the discussion in [Cochrane \(1999\)](#)) :

- expected returns are constant and normally independently distributed;
- the CAPM is a good measure of risk and thus a good explanation of why some stocks earn higher average returns than others;
- excess returns are close to being unpredictable: any predictability is a statistical artefact or cannot be exploited after transaction costs are taken into account;
- the volatility of returns is constant.

Fama (1970) clearly stated:

“... For data on common stocks, tests of ‘fair game’ (and random walk) properties seem to go well when conditional expected returns are estimated as the average return for the sample of data at hand. Apparently, the variation in common stock returns about their expected values is so large relative to any changes in expected values that the latter can be safely ignored...”

Time-Series Implications

In practice, the traditional view can be recast in terms of the simplest possible specification for the predictive models for returns, i.e., the constant expected returns model:

$$r_{t,t+1}^i = \mu^i + \sigma^i \epsilon_{it} \quad \epsilon_{it} \sim NID(0, 1)$$

$$Cov(\epsilon_{it}, \epsilon_{js}) = \begin{cases} \sigma_{ij} & t = s \\ 0 & t \neq s \end{cases} .$$

Note that the absence of predictability of excess returns is not a consequence of market efficiency per se but it instead results from a joint hypothesis: market efficiency plus some assumptions on the process generating returns (i.e., the Constant Expected Returns model).

Returns at different horizons

In this world, the horizon n does not matter for the prediction of returns because once μ_i and σ_i are estimated, expected returns at all horizons and the variance of returns at all horizon are derived deterministically.

$$E(r_{t,t+n}^i) = E\left(\sum_{k=1}^n r_{t+k,t+k-1}^i\right) = \sum_{k=1}^n E(r_{t+k,t+k-1}^i) = n\mu$$

$$Var(r_{t,t+n}^i) = Var\left(\sum_{i=1}^n r_{t+k,t+k-1}^i\right) = \sum_{i=1}^n Var(r_{t+k,t+k-1}^i) = n\sigma^2$$

As a consequence of these properties of the data, weights in an optimal multi-horizon portfolio coincide with weights in a single period horizon portfolio:

$$\begin{aligned} \hat{\mathbf{w}}^T &= \frac{\Sigma^{-1}(\boldsymbol{\mu} - r^f \mathbf{e})}{\mathbf{e}'\Sigma^{-1}(\boldsymbol{\mu} - r^f \mathbf{e})} \\ &= \frac{\Sigma^{-1}(n\mathbf{1}) (\boldsymbol{\mu} - r^f \mathbf{e})}{\mathbf{e}'\Sigma^{-1}(n\mathbf{1}) (\boldsymbol{\mu} - r^f \mathbf{e})} \end{aligned}$$

The Cross-Section of Returns

The CER view allows for cross-sectional heterogeneity of returns, but such cross-sectional heterogeneity is related to a single factor, the market factor, and the CAPM determines all the cross-sectional variation in μ^i . The statistical model that determines all returns r_t^i and the market return r_t^m , can be described as follows:

$$\begin{aligned} \begin{pmatrix} r_t^i - r_t^{rf} \end{pmatrix} &= \mu_i + \beta_i u_{m,t} + u_{i,t} \\ \begin{pmatrix} r_t^m - r_t^{rf} \end{pmatrix} &= \mu_m + u_{m,t} \\ \begin{pmatrix} u_{i,t} \\ u_{m,t} \end{pmatrix} &\sim n.i.d. \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{ii} & \sigma_{im} \\ \sigma_{im} & \sigma_{mm} \end{pmatrix} \right] \end{aligned}$$

where r_t^{rf} is the return on the risk-free asset. We shall see that $\sigma_{im} = 0$ is a crucial assumption for the valid estimation of the CAPM betas, and that assumption that risk-adjusted excess returns are zero (usually known as zero alpha assumption) requires that $\mu_i = \beta_i \mu_m$.

The Volatility of Returns

The volatility of returns is constant in the CER model which therefore is not capable of explaining time-varying volatility in the markets and the presence of alternating periods of high and low volatility.

Implications for Asset Allocation

When the data are generated by CER optimal asset allocation can be achieved by utility maximization that uses as inputs the historical moments of the distribution of returns, optimal portfolio weights are constant through the investment horizon. The optimal portfolio is always a combination between the market portfolio and the risk-free asset. The risk associated to any given asset or portfolio of assets is constant over time. Think of measuring the risk of a portfolio with its Value-at-Risk (VaR). The VaR is the percentage loss obtained with a probability at most of α percent:

$$\Pr(R^p < -VaR_\alpha) = \alpha.$$

where R^p are the returns on the portfolio. If the distribution of returns is normal, then α -percent VaR_α is obtained as follows (assume $\alpha \in (0, 1)$):

$$\begin{aligned} \Pr(R^p < -VaR_\alpha) &= \alpha \iff \Pr\left(\frac{R^p - \mu_p}{\sigma_p} < -\frac{VaR_\alpha + \mu_p}{\sigma_p}\right) = \alpha \\ &\iff \Phi\left(-\frac{VaR_\alpha + \mu_p}{\sigma_p}\right) = \alpha, \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative density of a standard normal. At this point, defining $\Phi^{-1}(\cdot)$ as the inverse CDF function of a standard normal, we have that

$$-\frac{VaR_\alpha + \mu_p}{\sigma_p} = \Phi^{-1}(\alpha) \iff VaR_\alpha = -\mu_p - \sigma_p \Phi^{-1}(\alpha).$$

and, given that μ_p and σ_p are constant over time, VaR_α is also constant over-time. Consider the case of a researcher interested in the one per cent value at risk. Because $\Phi^{-1}(0.01) = -2.33$ under the normal distribution, we can easily obtain VaR if we have available estimates of the first and second moments of the distribution of *portfolio returns*:

$$\widehat{VaR}_{0.01} = -\hat{\mu}_p - 2.33\hat{\sigma}_p$$

1.5 Empirical Challenges to the traditional model

Over the course of time the traditional view has been empirically challenged on many grounds. In particular, it has been observed that

- The tenet that expected returns are constant is not compatible with the observed volatility of stock prices. Stock prices in fact are "too volatile" to be determined only by expected dividends [Shiller \(1981\)](#);
- there is evidence of returns predictability that increases with the horizon at which returns are defined.
- There are anomalies that make returns predictable on the occasion of special events.
- The CAPM is rejected when looking at the cross-section of returns and multi-factor models are needed to explain the cross-sectional variability of returns
- high-frequency returns are non-normal and heteroscedastic, therefore risk is not constant over time and there is predictability of risk at high-frequency

1.5.1 The time-series evidence on expected returns

Practitioners implementing portfolio allocation based on the CER model experienced rather soon a number of problems that stressed the limitations of this model but it was the work of Robert Shiller and co-authors that led the profession to go beyond the CER model. The basic empirical evidence against the CER model was the excessive volatility of asset prices and returns which is clearly illustrated in [Shiller \(1981\)](#).

We shall illustrate the excess volatility evidence by considering a simple model of stock market returns: the Dynamic Dividend Growth (DDG) model. As we shall discuss in detail in one of the next chapters, total returns to a stock i can be satisfactorily approximated as follows:

$$r_{t+1}^s = \kappa + \rho(p_{t+1} - d_{t+1}) + \Delta d_{t+1} - (p_t - d_t)$$

where P_t is the stock price at time t and D_t is the dividend paid at time t , $p_t = \ln(P_t)$, $d_t = \ln(D_t)$, κ is a constant and $\rho = \frac{P/D}{1+P/D}$, P/D is the average price to dividend ratio. In practice ρ can be interpreted as a discount parameter ($0 < \rho < 1$). By forward recursive substitution one obtains:

$$(p_t - d_t) = \frac{\kappa}{1 - \rho} + \sum_{j=1}^m \rho^{j-1} (\Delta d_{t+j}) - \sum_{j=1}^m \rho^{j-1} (r_{t+j}^s) + \rho^m (p_{t+m+1} - d_{t+m+1})$$

which shows that the $(p_t - d_t)$ measures the value of a very long-term investment strategy (buy and hold). This value, in the absence of bubbles, is equal to the stream of future dividend growth discounted at the appropriate rate, which reflects the risk-free rate plus risk premium required to hold risky assets.

By introducing uncertainty, we have:

$$(p_t - d_t) = \frac{\kappa}{1 - \rho} + \sum_{j=1}^m \rho^{j-1} E_t (\Delta d_{t+j}) - \sum_{j=1}^m E_t \rho^{j-1} (r_{t+j}^s) + \rho^m E_t (p_{t+m+1} - d_{t+m+1})$$

Two considerations are relevant here. First, note that under the CER and no bubbles the price-dividend ratio should reflect only expected dividend growth. The empirical evidence is strongly against this prediction (see the [Campbell and Shiller \(1987\)](#)). Stock prices are too volatile to be determined only by expected dividends. [Campbell-Shiller\(1987\)](#) illustrate the point by comparing the observed price-dividend ratio and a counterfactual price-dividend ratio which is obtained by assuming constant future expected returns and by using a Vector Autoregressive Model to predict future dividend-growth: The volatility in the price-dividend ratio is much higher than that predicted by the CER model.

Second, once the hypothesis of CER is rejected, the DDG model has interesting implications for the predictability of returns at different horizons. If we decompose future variables into their expected component and the unexpected one (an error term) we can write the relationship between the dividend yield and the returns one period ahead and over the long-horizon as follows:

$$\begin{aligned}
 r_{t+1}^s &= \kappa + \rho E_t(p_{t+1} - d_{t+1}) + E_t \Delta d_{t+1} - (p_t - d_t) + \rho u_{t+1}^{pd} + u_{t+1}^{\Delta d} \\
 \sum_{j=1}^m \rho^{j-1} r_{t+j}^s &= \frac{\kappa}{1-\rho} + \sum_{j=1}^m \rho^{j-1} E_t(\Delta d_{t+j}) - (p_t - d_t) + \rho^m E_t(p_{t+m} - d_{t+m}) + \\
 &\quad \rho^m u_{t+m}^{pd} + \sum_{j=1}^m \rho^{j-1} u_{t+j}^{\Delta d}
 \end{aligned}$$

These two expressions illustrate that when the price dividends ratio is a noisy process, such noise dominates the variance of one-period returns and the statistical relation between the price-dividend ratio and one period returns is weak. However, as the horizon over which returns are defined gets longer, noise tends to be dampened and the predictability of returns given the price-dividend ratio increases.

The DDG model predicts a tighter relation between aggregate stock market returns and the price-dividend ratio as the horizon at which returns are defined increases. The first evidence of the increasing explanatory power of the dividend yield as the investment horizon increases is reported in Table (1). Here we report the slopes, the adjusted R^2 , as well as the adjusted t-stats as in [Valkanov \(2003\)](#), of the following predictive regression

$$r_{t:t+k} = \alpha_k + \beta_k \log(D_t/P_t) + \sigma \varepsilon_{t+k} \quad \varepsilon_{t+k} \sim N(0, 1)$$

where $r_{t:t+k}$ the aggregate US stock market returns from t to $t+k$, D_t the aggregate dividend, P_t the index, ε_{t+k} an idiosyncratic error component and σ its corresponding risk.

The sensitivity of the aggregate cumulative returns on the log dividend-yield β_k increases with the investment horizon. The same is true for the adjusted R^2 , meaning, the longer the forecasting term, the higher the predictive power of the value-weighted dividend-yield.

1.5.2 Anomalies

The evidence of predictability of returns is strengthened by the presence of episodes of "anomalies". An interesting illustration of this type of evidence is the one reported

TABLE 1.1. The Predictive Power of the Dividend-Yield

This table reports the OLS estimates of the aggregate US stock market returns on the value-weighted dividend-price ratio. The sample is monthly and goes from 1946:01 to 2012:12. The first column reports the forecasting horizon. The second column reports the slope coefficients while the third the adjusted t-stats, i.e. t/\sqrt{T} as in [Valkanov \(2003\)](#). The last column reports the adjusted R^2 .

Horizon k	$\hat{\beta}$	t/\sqrt{T}	R^2
1	0.726	0.092	0.007
4	3.369	0.187	0.032
8	7.105	0.269	0.066
16	15.96	0.412	0.144
24	23.59	0.523	0.214
60	54.69	0.976	0.487

in [Lucca and Moench \(2015\)](#), who document large average excess returns on U.S. equities in anticipation of monetary policy decisions made at scheduled meetings of the Federal Open Market Committee (FOMC) in the past few decades. Following up on this evidence [Cieslak et al. \(2019\)](#) document that since 1994 the US equity premium has followed an alternating weekly pattern measured in FOMC cycle time, i.e. in time since the last Federal Open Market Committee meeting.

1.5.3 The Cross-section Evidence on Expected Returns

The CAPM has important empirical implications for the cross sections of assets.

$$E(r^i - r^f) = \beta_i E(r^M - r^f)$$

then heterogeneity in excess returns to different assets should be totally explained by the different exposure to a single common risk factor, the market excess returns.

Given a sample of observations on r_t^i, r_t^f, r_t^M , the β_i can be estimated first by OLS regression over the time series of returns, then the following second-pass equations can be estimated over the cross-section of returns:

$$\bar{r}_i = \gamma_0 + \gamma_1 \beta_i + u_i$$

Where \bar{r}_i are the average returns in the period over which the β_i have been computed.

If the CAPM is valid, then γ_0 and γ_1 should satisfy:

$$\gamma_0 = \bar{r}^f, \gamma_1 = \bar{r}^M$$

where \bar{r}^M is the mean market excess return.

When the model is estimated with appropriate methods, the restrictions are strongly rejected (Fama and French (1993), Fama and MacBeth (1973)). This evidence has paved the way to the estimation of multi-factor models of returns. Fama and French (1993) introduced a three-factor model based on the integration of the CAPM with a “small-minus-big” market value (SMB) and “high-minus-low” book-to-market ratio (HML). These factors are equivalent to zero-cost arbitrage portfolio that takes a long position in high book-to-market (small-size) stocks and finances this with a short position in low book-to-market (large-size) stocks. Jegadeesh and Titman (2011) discovered the importance of a further additional factor in explaining excess returns: momentum(MOM). An investment strategy that buys stocks that have performed well and sells stocks that have performed poorly over the past 3-to 12-month period generates significant excess returns over the following year. More recently Fama and French (2015) have extended the standard factors model based on the Market, SMB, HML and MOM, to include two more factors: RMW and CMA. RMW (Robust Minus Weak) is the return on a portfolio long on robust operating profitability stocks and short on weak operating profitability stocks, while CMA (Conservative Minus Aggressive) is the average return on a position long on conservative investment portfolios and short on aggressive investment. It is interesting to note that augmenting the CAPM with SMB and HML, does not challenge per se the CER model, which still holds as valid if the constant expected return model can be applied to the two additional factors. However, momentum provides direct evidence against the CER model as it indicates that the conditional expectations of future returns is not constant.

1.5.4 The behaviour of returns at high-frequency

At small horizon (i.e. when k is small: infra-daily, daily, weekly or at most monthly returns) the following modelling framework is consistent with the data:

$$\begin{aligned} R_{t,t+k} &= \sigma_{k,t} u_{t+k} \\ \sigma_{k,t}^2 &= f(\mathcal{I}_t) \quad u_{t+k} \sim IID \mathcal{D}(0, 1). \end{aligned}$$

The following features of this model at high frequency are noteworthy:

1. The distribution of returns is centered around a mean of zero, and the zero mean model dominates any alternative model based on predictors.

2. The variance is time-varying and predictable, given the information set, \mathcal{I}_t , available at time t .
3. The distribution of returns at high frequency is not normal, i.e., $\mathcal{D}(0, 1)$ may often differ from $\mathcal{N}(0, 1)$

1.6 The Implications of the new evidence

1.6.1 Asset Pricing with Predictable Returns

The evidence that the CER model does not provide the best representation of the data opens a very interesting question on the determinants of time-varying expected returns. An immediate motivation for predictability can be found in market malfunctions or expectations mechanisms that do not efficiently process the available information. However, time-varying expected returns can also be understood in the context of a basic model that stems from the assumption of the absence of "arbitrage opportunities" (i.e. by the impossibility of making profits without taking risk). Consider a situation in which in each period k state of nature can occur and each state has a probability $\pi(k)$, in the absence of arbitrage opportunities the price of an asset i at time t can be written as follows:

$$P_{i,t} = \sum_{s=1}^k \pi_{t+1}(s) m_{t+1}(s) X_{i,t+1}(s)$$

where $m_{t+1}(s)$ is the discounting weight attributed to future pay-offs, which (as the probability π) is independent from the asset i , $X_{i,t+1}(s)$ are the payoffs of the assets (we have seen that in case of stocks we have $X_{i,t+1} = P_{t+1} + D_{t+1}$), and therefore returns on assets are defined as $1 + R_{s,t+1} = \frac{X_{i,t+1}}{P_{i,t}}$. For the safe asset, whose payoffs do not depend on the state of nature, we have:

$$P_{s,t} = X_{i,t+1} \sum_{s=1}^k \pi_{t+1}(s) m_{t+1}(s)$$

$$1 + R_{s,t+1} = \frac{1}{\sum_{j=1}^m \pi_{t+1}(s) m_{t+1}(s)}$$

In general, we can write:

$$P_{i,t} = E_t(m_{t+1}X_{i,t+1})$$

$$1 + R_{s,t+1} = \frac{1}{E_t(m_{t+1})}$$

consider now a risky asset :

$$E_t(m_{t+1}(1 + R_{i,t+1})) = 1$$

$$Cov(m_{t+1}R_{i,t+1}) = 1 - E_t(m_{t+1})E_t(1 + R_{i,t+1})$$

$$E_t(1 + R_{i,t+1}) = -\frac{Cov(m_{t+1}R_{i,t+1})}{E_t(m_{t+1})} + (1 + R_{s,t+1})$$

Turning now to excess returns we can write:

$$E_t(R_{i,t+1} - R_{s,t+1}) = -(1 + R_{s,t+1})cov(m_{t+1}R_{i,t+1})$$

Assets whose returns are low when the stochastic discount factor is high (i.e. when agents value payoffs more) require a higher risk premium, i.e. a higher excess return on the risk-free rate. Turning to predictability at different horizon, if you consider the case in which t is defined by taking two points in time very close to each other the safe interest rate will be approximately zero and m will not vary too much across states. The constant expected return model (with expected returns equal to zero) is compatible with the no-arbitrage approach at high frequency. However, consider now the case of low frequency, when t is defined by taking two very distant points in time; in this case, safe interest rate will be different from zero and m will vary sizeably across different states. The constant expected return model is not a good approximation at long horizons. Predictability is not necessarily a symptom of market malfunction but rather the consequence of fair compensation for risk-taking, then it should reflect attitudes toward risk and variation in market risk over time. Different theories on the relationship between risk and asset prices should then be assessed on the basis of their ability to explain the predictability that emerges from the data.

Also, different theories or return predictability can be interpreted as different theories of the determination of m and/or different mechanism of formation of expectations. On the one hand we have theories of m based on rational investor behaviour and rational expectations, on the other hand, we have alternative approaches based on psychological models of investor behaviour. Our main interest is on how the predictability of returns can be used for optimal portfolio allocation purposes, rather than in discriminating between the possible sources of predictability.

1.7 Quantitative Risk Management and returns at high-frequency

Once the portfolio weights ($\hat{\mathbf{w}}$) are chosen, possibly exploiting the predictability of the distribution of the relevant future returns, the distribution of portfolio returns can be described as follows:

$$\begin{aligned} R^p &\sim \mathcal{D}(\mu_p, \sigma_p^2) \\ \mu_p &= \boldsymbol{\mu}'\hat{\mathbf{w}} \quad \sigma_p^2 = \hat{\mathbf{w}}'\boldsymbol{\Sigma}\hat{\mathbf{w}} \end{aligned}$$

Having solved the portfolio problem and having committed to a given allocation described by $\hat{\mathbf{w}}$, there is a different role that econometrics can play at high frequencies: measuring volatility and providing information on portfolio risk. As our simple specification of the previous section shows, noise is not predictable but its volatility is. The role of econometrics in applied risk management is best seen through a different statistical model of high-frequency returns. When k is small (i.e., when one is considering infra-daily, daily, weekly or at most monthly returns) the following framework is normally referred to:

$$\begin{aligned} R_{t,t+k} &= \sigma_{k,t}u_{t+k} \\ \sigma_{k,t}^2 &= f(\mathcal{I}_t) \quad u_{t+k} \sim IID \mathcal{D}(0, 1). \end{aligned}$$

The following features of the model at high frequency are noteworthy:

1. The distribution of returns is centered around a mean of zero, and the zero mean model dominates any alternative model based on predictors.
2. The variance is time-varying and predictable, given the information set, \mathcal{I}_t , available at time t .
3. The distribution of returns at high frequency is not normal, i.e., $\mathcal{D}(0, 1)$ may often differ from $\mathcal{N}(0, 1)$

Given these features of the data, econometrics can still be used at high frequency to assess the risk of a given portfolio. In particular, we shall investigate the role of econometrics in deriving the time-varying Value-at-Risk (VaR) of a given portfolio.

1.8 Predictive Models in Finance: a General Representation.

Predictive models are statistical models of future behaviour in which relations between the variables to be predicted and the predictors are specified as functional relation

determined by parameters to be estimated. Predictive models can be univariate, when there is only one variable of interest, or multivariate when we have a vector of variables of interest.

All predictive models we shall analyze are special cases of the following general representation:

$$\mathbf{r}_{t,t+k} = f(X_t^\mu, \Theta_t^\mu) + \mathbf{H}_{t+k}\epsilon_{t+k} \quad (1.1)$$

$$\Sigma_{t+k} = \mathbf{H}_{t+k}\mathbf{H}_{t+k}'$$

$$\Sigma_{t+k} = g(X_t^\sigma, \Theta_t^\sigma) + \sum_{j=1}^q \mathbf{B}_j \Sigma_{t+k-j} \mathbf{B}_j', \quad (1.2)$$

$$\epsilon_{t+k} \sim \mathcal{D}(\mathbf{0}, \mathbf{I})$$

where $\mathbf{r}_{t,t+k}$ is the vector of returns between time t and time $t+k$ in which we are interested, X_t^μ is the vector of predictors for the mean of our returns that we observe at time t , f specifies the functional relation (which is potentially time-varying) between the mean returns and the predictors that depend also on a set of parameters Θ_t^μ , the matrix \mathbf{H}_{t+k} determines the potentially time varying variance-covariance of the vector of returns. The process for the variance is predictable as there is a functional relation determining the relationship between \mathbf{H}_{t+k} and a vector of predictors X_t^σ that is driven by a vector of unknown parameters Θ_t^σ .

Our initial discussion of this chapter illustrates that the appropriate specification of the general predictive model depends on the horizon at which returns are defined. Consider, for example, the problem of univariate modelling of stock market returns. When k is small and high-frequency returns On the one hand, in the simple asset allocation model, the econometric framework considered for returns is as follows:¹

$$\begin{aligned} r_{t,t+k} &= 0 + \sigma_{t+k}u_{t+k} & u_{t+k} &\sim IID \mathcal{D}(0, 1), \\ \sigma_{t+k}^2 &= \omega + \alpha\sigma_{t+k-1}^2 + \beta u_{t+k-1}^2, & |\alpha + \beta| &< 1 \end{aligned}$$

This is a model that features no predictability in the mean of r returns (the expected future return at any horizon is constant at zero), but there is predictability in the variance of returns that it is mean reverting towards a long-term value of $\omega/(1 - \alpha - \beta)$. No assumption of normality is made for the innovation in the process generating returns. Consider now the case of large k , i.e. long-horizon returns (note that in the continuously compounded case, $r_{t,t+k} \equiv \sum_{j=1}^k r_{t,t+j}$), in this case the relevant predictive model can be written as follows:

$$r_{t,t+k} = \alpha + \beta' \mathbf{X}_t + \sigma u_{t+k} \quad u_{t+k} \sim IID \mathcal{N}(0, 1),$$

¹The sum of IIDness of returns and of normality has also been denoted as $u_{t+k} \sim n.i.d.(0, 1)$. Note that IID $\mathcal{N}(0, 1)$ and n.i.d.(0, 1) have identical meaning.

where \mathbf{X}_t is a set of predictors observed at time t . In this case we have that returns feature predictability in mean, constant variance and the innovations are normally distributed. As the horizon k increases, predictability increases and therefore the uncertainty related to the unexpected components of returns decreases (i.e., the annualized variance of returns is a downward-sloping function of the horizon). Moreover—as we have already discussed—the dependence of $\sigma_{t,k}$ on time (i.e., its time-varying nature) declines and long-horizon returns can be described as a (conditional) normal homoskedastic processes. In the short-run noise dominates and modelling returns on the basis of fundamentals is very difficult. However, as the horizon increases fundamentals become more important to explain returns and the risk associated with portfolio allocation based on econometric models is reduced. The statistical model becomes more and more precise as k gets large.

Chapter 2

Asset Prices and Returns

2.1 Introduction

In this chapter we shall investigate the main objects of our analysis by illustrating first how returns can be defined and their relationships with prices and by then illustrating how returns and prices can be empirically analyzed by using R.

2.2 Returns

Consider an asset that does not pay any intermediate cash income (a zero-coupon bond, such as a Treasury Bill, or a share in a company that pays no dividends). Let P_t be the price of the security at time t .

2.2.1 Simple and log Returns

The linear or simple return between times t and $t - 1$ is defined as¹:

$$R_t = P_t/P_{t-1} - 1 \tag{2.1}$$

The log, or continuously compounded, return is defined as:

$$r_t = \ln(P_t/P_{t-1}) = \ln(1 + R_t)$$

Note that, while P_t means “price at time t ”, r_t is a shorthand for “return between time $t - 1$ and t ” so that the notation is not really complete, and its interpretation

¹Note that (2.1) defines period returns, there is usually an accrual convention applied to returns according to which they are transformed on a yearly basis.

depends on the context. When needed for clarity, we shall specify returns as indexed by the start and the end point of the interval in which they are computed as, for instance, in $r_{t-1,t}$.

The two definitions of return yield different numbers when the ratio between consecutive prices is far from 1.

Consider the Taylor formula for $\ln(x)$ for x in the neighbourhood of 1:

$$\ln(x) = \ln(1) + (x - 1)/1 - (x - 1)^2/2 + \dots$$

if we truncate the series at the first order term we have:

$$\ln(x) \cong 0 + x - 1$$

so that if x is the ratio between consecutive prices, then for x close to one the two definitions give similar values. Note however that $\ln(x) \leq x - 1$. In fact, $x - 1$ is equal to and tangent to $\ln(x)$ in $x = 1$ and above it anywhere else (in fact, the second derivative of $\ln(x)$ is negative). This implies that if one definition of return is used in place of the other, the approximation errors shall be all of the same sign. This fact has important consequences when multi-period returns are computed as the difference between the two definitions will become larger and larger.

2.2.2 Statistical models for asset prices and returns.

A standard model for asset prices is the log random walk model with Gaussian residuals

$$\begin{aligned} \ln P_t &= \alpha_0 + \ln P_{t-1} + u_t \\ u_t &\sim N.I.D.[0, \sigma^2] \end{aligned} \tag{2.2}$$

in this case, log returns are normally distributed, this implies that single period gross returns are i.i.d lognormal variables, as $r_{t+1} \equiv \log(1 + R_{t+1})$. Note that, under the lognormal model

$$\begin{aligned} r_{t,t+1} &\sim n.i.d.(\mu, \sigma^2) \\ E(R_{t,t+1}) &= \exp\left(\mu + \frac{1}{2}\sigma^2\right) - 1 \\ Var(R_{t,t+1}) &= \exp(2\mu + \sigma^2) (e^{\sigma^2} - 1) \end{aligned}$$

In the case we have a vector of log returns that are normally distributed:

$$\begin{aligned} \mathbf{r}_{t,t+1} &\sim i.i.d.(\mu, \Sigma) \\ E(R_{t,t+1}^i) &= \exp\left(\mu_i + \frac{1}{2}\sigma_{ii}\right) - 1 \\ Cov(R_{t,t+1}^i, R_{t,t+1}^j) &= \exp\left(\mu_i + \mu_j + \frac{1}{2}(\sigma_{ii} + \sigma_{jj})\right)(e^{\sigma_{ij}} - 1) \end{aligned}$$

2.2.3 Multi-period returns and annualized returns

What are multiperiod returns? Multiperiod returns are returns to an investment which is made with a horizon larger than one. Let us consider the case of the returns to an investment made in time t until time $t + n$. In this case, we define the simple multi-period return as:

$$\begin{aligned} R_{t,t+n} &= P_{t+n}/P_t - 1 \\ &= \frac{P_{t+n}}{P_{t+n-1}} \frac{P_{t+n-1}}{P_{t+n-2}} \dots \frac{P_{t+1}}{P_t} - 1 \\ &= \prod_{i=1}^n (1 + R_{t+i,t+i-1}) - 1 \end{aligned} \tag{2.3}$$

in the case of log returns we have instead:

$$\begin{aligned} r_{t,t+n} &= \ln(P_{t+n}/P_t) \\ &= \ln\left(\frac{P_{t+n}}{P_{t+n-1}} \frac{P_{t+n-1}}{P_{t+n-2}} \dots \frac{P_{t+1}}{P_t}\right) \\ &= \sum_{i=1}^n r_{t+i,t+i-1} \end{aligned} \tag{2.4}$$

Consider the case in which the length of our period is one year, given any multiperiod returns one can define its annualized value i.e. as the constant annual rate of return equivalent to the multiperiod returns of an investment in asset i over the period $t, \dots, t+n$.

In the case of simple returns, we have

$$\begin{aligned}
(1 + R_{t,t+n}^A)^n &= 1 + R_{t,t+n} \\
&= \prod_{i=1}^n (1 + R_{t+i,t+i-1}) \\
R_{t,t+n}^A &= \left(\prod_{i=1}^n (1 + R_{t+i,t+i-1}) \right)^{\frac{1}{n}} - 1
\end{aligned}$$

the annualized simple rate of return is the geometric mean of the annual returns over the period $t, t+n$.

Consider now continuously compounded returns:

$$\begin{aligned}
nr_{t,t+n}^A &= r_{t,t+n} \\
&= \sum_{i=1}^n r_{t+i,t+i-1} \\
r_{t,t+n}^A &= \frac{1}{n} \sum_{i=1}^n r_{t+i,t+i-1}
\end{aligned}$$

The annualized log return is the arithmetic mean of annual log returns.

2.2.4 Working with Returns

Consider the value of a buy-and-hold portfolio invested in shares of k different companies, that pay no dividend, at time t be:

$$V_t = \sum_{i=1}^k n_i P_{it}$$

The simple one-period return of the portfolio shall be a linear function of the returns of each stock.

$$\begin{aligned}
R_t &= \frac{V_t}{V_{t-1}} - 1 = \sum_{i=1..k} \frac{n_i P_{it}}{\sum_{j=1..k} n_j P_{jt-1}} - 1 \\
&= \sum_{i=1..k} \frac{n_i P_{it-1}}{\sum_{j=1..k} n_j P_{jt-1}} \frac{P_{it}}{P_{it-1}} - 1 =
\end{aligned}$$

$$= \sum_{i=1..k} w_{it}(R_{it} + 1) - 1 = \left(\sum_{i=1..k} w_{it}R_{it} + \sum_{i=1..k} w_{it}1 \right) - 1 = \sum_{i=1}^k w_{it}R_{it}$$

Where $w_{it} = \frac{n_i P_{it-1}}{\sum_i n_i P_{it-1}}$ are non negative "weights" summing to 1 which represent the percentage of the portfolio invested in the i -th stock at time $t - 1$.

This simple result is very useful. Suppose, for instance, that you know at time $t - 1$ the expected values for the returns between time $t - 1$ and t . Since the expected value is a linear operator (the expected value of a sum is the sum of the expected values, moreover additive and multiplicative constants can be taken out of the expected value) and the weights w_{it} are known, hence non-stochastic, at time $t - 1$ we can easily compute the return for the portfolio as:

$$E(R_t) = \sum_{i=1..k} w_{it}E(R_{it})$$

Moreover, if we know all the covariances between r_{it} and r_{jt} (if $i = j$ we simply have a variance) we can find the variance of the portfolio return as:

$$V(R_t) = \sum_{i=1..k} \sum_{j=1..k} w_i w_j Cov(R_{it}; R_{jt})$$

This cross-sectional additivity property does not apply to log returns. In fact, we have:

$$\begin{aligned} r_t &= \ln\left(\frac{V_t}{V_{t-1}}\right) \\ &= \ln\left(\frac{\sum_{i=1}^k n_i P_{it-1} \frac{P_{it}}{P_{it-1}}}{\sum_{i=1}^k n_i P_{it-1}}\right) = \ln\left(\sum_{i=1}^k w_{it} \exp(r_{it})\right) \end{aligned}$$

The log return of the portfolio is not a linear function of the log (and also the linear) returns of the components. In this case assumptions on the expected values and covariances of the components cannot be translated into assumptions on the expected value and the variance of the portfolio by simple use of basic "expected value of the sum" and "variance of the sum" formulas.

On the other hand, log returns are additive when we consider the time series of returns

$$r_{t,t+n} = \sum_{i=1}^n r_{t+i,t+i-1}$$

It is then easy, for instance, given the expected values and the covariances of the sub-period returns, to compute the expected value and the variance of the full-period return. Interestingly, additivity does not apply to simple returns.

$$R_{t,t+n} = \prod_{i=1}^n R_{t+i,t+i-1} - 1$$

In general, the expected value of a product is difficult to evaluate and does not depend only on the expected values of the terms.

To sum up: the two definitions of returns yield different values when the ratio between consecutive prices is not in the neighbourhood of the unit value. The linear definition works very well for portfolios over single periods, in the sense that expected values and variances of portfolios can be derived by expected values variances and covariances of the components, as the portfolio linear return over a time period is a linear combination of the returns of the portfolio components. For analogous reasons, the log definition works very well for single securities over time. However, care must be exercised when long-horizon returns are computed by cumulating continuously compounded returns.

2.3 Stock and Bond Returns

The computation of returns for stock and bonds must take into account the existence of intermediate cash income. In this section we show how this is performed and how linearization can help the empirical analysis of the stock and bond markets.

2.3.1 Stock Returns and the dynamic dividend growth model

Consider the one-period total holding returns in the stock market, that are defined as follows:²

$$H_{t+1}^s \equiv \frac{P_{t+1} + D_{t+1}}{P_t} - 1 = \frac{P_{t+1} - P_t + D_{t+1}}{P_t} = \frac{\Delta P_{t+1}}{P_t} + \frac{D_{t+1}}{P_t}, \quad (2.5)$$

where P_t is the stock price at time t , D_t is the (cash) dividend paid at time t , and the superscript s denotes “stock”. The last equality decomposes a discrete holding

²The use of ‘ \equiv ’ emphasizes that (2.5) provides a definition. Moreover, ΔX_{t+1} denotes the first difference of a generic variable, or $\Delta X_{t+1} \equiv X_{t+1} - X_t$.

period return as the sum of the percentage capital gain and of (a definition of) the *dividend yield*, D_{t+1}/P_t . Given that one-period returns are usually small, it is sometimes convenient to approximate them with logarithmic, continuously compounded returns, defined as:

$$r_{t+1}^s \equiv \log(1 + H_{t+1}^s) = \log\left(\frac{P_{t+1} + D_{t+1}}{P_t}\right) = \log(P_{t+1} + D_{t+1}) - \log(P_t). \quad (2.6)$$

Interestingly, while linear returns are additive in the percentage capital gain and the dividend yield components, log returns are not as

$$\log\left(\frac{P_{t+1} + D_{t+1}}{P_t}\right) \neq \log\left(\frac{P_{t+1}}{P_t}\right) + \log\left(\frac{D_{t+1}}{P_t}\right)$$

However, it is still possible to express log returns as a linear function of the log of the price dividend and the (log) dividend growth. Dividing both sides of (2.5) by $(1 + H_{t+1}^s)$ and multiplying both sides by P_t/D_t we have:

$$\frac{P_t}{D_t} = \frac{1}{(1 + H_{t+1}^s)} \frac{D_{t+1}}{D_t} \left(1 + \frac{P_{t+1}}{D_{t+1}}\right).$$

Taking logs (denoted by lowercase letters, i.e., $x_t \equiv \log X_t$ for a generic variable X_t), we have:³

$$p_t - d_t = -r_{t+1}^s + \Delta d_{t+1} + \ln(1 + e^{p_{t+1} - d_{t+1}}) \quad (2.7)$$

as $\log(D_{t+1}/D_t) = \log D_{t+1} - \log D_t = \Delta \log D_{t+1} = \Delta d_{t+1}$. Taking a first-order Taylor expansion of the last term about the point $\bar{P}/\bar{D} = e^{\bar{p}-\bar{d}}$ (where the bar denotes a sample average), the logarithm term on the right-hand side can be approximated

³ $-r_{t+1}^s$ follows from

$$\begin{aligned} \log \frac{1}{(1 + H_{t+1}^s)} &= \log 1 - \log(1 + H_{t+1}^s) \\ &= -\log(1 + H_{t+1}^s) = -r_{t+1}^s \end{aligned}$$

based on our earlier definitions and the fact that $\log 1 = 0$ for natural logs. Moreover, notice that

$$\frac{P_{t+1}}{D_{t+1}} = e^{\log(P_{t+1}/D_{t+1})} = e^{\log P_{t+1} - \log D_{t+1}} = e^{p_{t+1} - d_{t+1}}$$

as:

$$\begin{aligned}
\ln(1 + e^{p_{t+1} - d_{t+1}}) &\simeq \ln(1 + e^{\bar{p} - \bar{d}}) + \frac{e^{\bar{p} - \bar{d}}}{1 + e^{\bar{p} - \bar{d}}} [(p_{t+1} - d_{t+1}) - (\bar{p} - \bar{d})] \\
&= -\ln(1 - \rho) - \rho \ln\left(\frac{1}{1 - \rho} - 1\right) + \rho(p_{t+1} - d_{t+1}) \\
&= \kappa + \rho(p_{t+1} - d_{t+1})
\end{aligned}$$

where

$$\rho \equiv \frac{e^{\bar{p} - \bar{d}}}{1 + e^{\bar{p} - \bar{d}}} = \frac{\bar{P}/\bar{D}}{1 + (\bar{P}/\bar{D})} < 1 \quad \kappa \equiv -\ln(1 - \rho) - \rho \ln\left(\frac{1}{1 - \rho} - 1\right).$$

Although $\rho \in (0, 1)$ is just a factor that depends on the average price-dividend ratio, in what follows will be used in a way that resembles a discount factor. At this point, substituting the expression for the approximated term in (2.7), we obtain that the log price-dividend ratio is defined as:⁴

$$p_t - d_t \simeq \kappa - r_{t+1}^s + \Delta d_{t+1} + \rho(p_{t+1} - d_{t+1}).$$

Re-arranging this expression shows that total stock market returns can be written as:

$$r_{t+1}^s = \kappa + \rho(p_{t+1} - d_{t+1}) + \Delta d_{t+1} - (p_t - d_t),$$

or a constant κ , plus the log dividend growth rate (Δd_{t+1}), plus the (discounted, at rate ρ) change in the log price-dividend ratio, $\rho(p_{t+1} - d_{t+1}) - (p_t - d_t) = \Delta(p_{t+1} - d_{t+1}) - (1 - \rho)(p_{t+1} - d_{t+1})$. Moreover, by *forward* recursive substitution one obtains:

$$\begin{aligned}
(p_t - d_t) &= \kappa - r_{t+1}^s + \Delta d_{t+1} + \rho(p_{t+1} - d_{t+1}) \\
&= \kappa - r_{t+1}^s + \Delta d_{t+1} + \rho(\kappa - r_{t+2}^s + \Delta d_{t+2} + \rho(p_{t+2} - d_{t+2})) \\
&= (\kappa + \rho\kappa) - (r_{t+1}^s + \rho r_{t+2}^s) + (\Delta d_{t+1} + \rho\Delta d_{t+2}) + \rho^2(p_{t+2} - d_{t+2}) \\
&= (\kappa + \rho\kappa) - (r_{t+1}^s + \rho r_{t+2}^s) + (\Delta d_{t+1} + \rho\Delta d_{t+2}) + \\
&\quad + \rho^2(\kappa - r_{t+3}^s + \Delta d_{t+3} + \rho(p_{t+3} - d_{t+3})) \\
&= \kappa(1 + \rho + \rho^2) - (r_{t+1}^s + \rho r_{t+2}^s + \rho^2 r_{t+3}^s) + (\Delta d_{t+1} + \rho\Delta d_{t+2} + \rho^2\Delta d_{t+3}) + \rho^3(p_{t+3} - d_{t+3}) \\
&= \dots = \kappa \sum_{j=1}^m \rho^{j-1} + \sum_{j=1}^m \rho^{j-1} (\Delta d_{t+j} - r_{t+j}^s) + \rho^m (p_{t+m} - d_{t+m}).
\end{aligned}$$

⁴The approximation notation ‘ \simeq ’ appears to emphasize that this expression is derived from an application of a Taylor expansion.

Under the assumption that there can be no rational bubbles, i.e., that⁵

$$\lim_{m \rightarrow \infty} \rho^m (p_{t+m} - d_{t+m}) = 0,$$

from

$$\lim_{m \rightarrow \infty} \sum_{j=1}^m \rho^{j-1} = \frac{1}{1 - \rho}$$

if $\rho \in (0, 1)$, we get

$$(p_t - d_t) = \frac{\kappa}{1 - \rho} + \sum_{j=1}^m \rho^{j-1} (\Delta d_{t+j} - r_{t+j}^s).$$

This result shows that the log price-dividend ratio, $(p_t - d_t)$, measures the value of a very long-term investment strategy (buy and hold) which—apart from a constant $\kappa/(1 - \rho)$ —is equal to the stream of future dividend growth discounted at the appropriate rate, which reflects the risk-free rate plus the risk premium required to hold risky assets, $r_{t+j}^s \equiv r^f + (r_{t+j}^s - r^f)$.⁶ Therefore, for long investment horizons, econometric methods may hope to infer from the data two different types of “information”: information concerning the forecasts of future (continuously compounded) dividend growth rates, i.e., $\Delta d_{t+1}, \Delta d_{t+2}, \dots, \Delta d_{t+m}$ as $m \rightarrow \infty$, which are measures of the cash flows paid out by the risky assets (e.g., how well a company will do); information concerning future discount rates, and in particular future risk premia, i.e., $(r_{t+1}^s - r^f), (r_{t+2}^s - r^f), \dots, (r_{t+m}^s - r^f)$ as $m \rightarrow \infty$. Note that, under the null hypothesis of constancy of returns, the volatility of the price dividend ratio should be completely explained by that of the dividend process. The empirical evidence is strongly against this prediction (see the Shiller(1981) and Campbell-Shiller(1987)).

If we decompose future variables into their expected component and the unexpected one (an error term) we can write the relationship between the dividend yield and the returns one period ahead and over the long horizon as follows:

⁵This assumption means that as the horizon grows without bounds, the log price-dividend ratio (hence, the underlying price-dividend ratio) may grow without bounds, but this needs to happen at a speed that is inferior to $1/\rho > 1$, so that when $p_{t+m} - d_{t+m}$ is discounted at the rate ρ^m , the limit of the quantity $\rho^m (p_{t+m} - d_{t+m})$ is zero.

⁶Here we have assumed that the risk-free interest rate is approximately constant. We shall see that, at least as a first approximation, this is an assumption that holds in practice.

$$\begin{aligned}
r_{t+1}^s &= \kappa + \rho E_t(p_{t+1} - d_{t+1}) + E_t \Delta d_{t+1} - (p_t - d_t) + \rho u_{t+1}^{pd} + u_{t+1}^{\Delta d} \\
\sum_{j=1}^m \rho^{j-1} r_{t+j}^s &= \frac{\kappa}{1-\rho} + \sum_{j=1}^m \rho^{j-1} E_t(\Delta d_{t+j}) - (p_t - d_t) + \rho^m E_t(p_{t+m} - d_{t+m}) + \\
&\quad \rho^m u_{t+m}^{pd} + \sum_{j=1}^m \rho^{j-1} u_{t+j}^{\Delta d}
\end{aligned}$$

These two expressions illustrate that when the price dividends ratio is a noisy process, such noise dominates the variance of one-period returns and the statistical relation between the price dividend ratio and one-period returns is weak. However, as the horizon over which returns are defined gets longer, noise tends to be dampened and the predictability of returns given the price dividend ratio increases.

2.3.2 Bond Returns: Yields-to-Maturity and Holding Period Returns

We turn now to bonds. We distinguish between two types of bonds: those paying a coupon each given period and those that do not pay a coupon but just reimburse the entire capital upon maturity (zero-coupon bonds).

Zero-Coupon Bonds

Define the relationship between price and yield to maturity of a zero-coupon bond as follows:

$$P_{t,T} = \frac{1}{(1 + Y_{t,T})^{T-t}}, \quad (2.8)$$

where $P_{t,T}$ is the price at time t of a bond maturing at time T , and $Y_{t,T}$ is yield to maturity. Taking logs of the left and the right-hand sides of the expression for $P_{t,T}$, and defining the continuously compounded *yield*, $y_{t,T}$, as $\log(1 + Y_{t,T})$, we have the following relationship:

$$p_{t,T} = -(T - t) y_{t,T}, \quad (2.9)$$

which clearly illustrates that the elasticity of the yield to maturity to the price of a zero-coupon bond is the maturity of the security. Therefore, the duration of the bond equals maturity as no coupons are paid. The one-period uncertain holding-period

return on a bond maturing at time T , $r_{t,t+1}^T$, is then defined as follows:

$$r_{t,t+1}^T \equiv p_{t+1,T} - p_{t,T} = -(T-t-1)y_{t+1,T} + (T-t)y_{t,T} \quad (2.10)$$

$$\begin{aligned} &= y_{t,T} - (T-t-1)(y_{t+1,T} - y_{t,T}), \\ &= (T-t)y_{t,T} - (T-t-1)y_{t+1,T}, \end{aligned} \quad (2.11)$$

which means that yields and returns differ by a scaled measure of the change between the yield at time $t+1$, $y_{t+1,T}$, and the yield at time t , $y_{t,T}$.

Coupon Bonds

The relationship between price and yield to maturity of a constant coupon (C) bond is given by:

$$P_{t,T}^c = \frac{C}{(1+Y_{t,T}^c)} + \frac{C}{(1+Y_{t,T}^c)^2} + \dots + \frac{1+C}{(1+Y_{t,T}^c)^{T-t}}.$$

When the bond is selling at par, the yield to maturity is equal to the coupon rate. To measure the length of time that a bondholder has invested money for we need to introduce the concept of duration:

$$\begin{aligned} D_{t,T}^c &= \frac{\frac{C}{(1+Y_{t,T}^c)} + 2\frac{C}{(1+Y_{t,T}^c)^2} + \dots + (T-t)\frac{1+C}{(1+Y_{t,T}^c)^{T-t}}}{P_{t,T}^c} \\ &= \frac{C \sum_{i=1}^{T-t} \frac{i}{(1+Y_{t,T}^c)^i} + \frac{(T-t)}{(1+Y_{t,T}^c)^{T-t}}}{P_{t,T}^c}. \end{aligned}$$

Note that when a bond is floating at par we have

$$\begin{aligned} D_{t,T}^c &= Y_{t,T}^c \sum_{i=1}^{T-t} \frac{i}{(1+Y_{t,T}^c)^i} + \frac{(T-t)}{(1+Y_{t,T}^c)^{T-t}} \\ &= Y_{t,T}^c \frac{\left((T-t) \frac{1}{1+Y_{t,T}^c} - (T-t) - 1 \right) \frac{1}{(1+Y_{t,T}^c)^{T-t+1}} + \frac{1}{1+Y_{t,T}^c}}{\left(1 - \frac{1}{1+Y_{t,T}^c} \right)^2} + \frac{(T-t)}{(1+Y_{t,T}^c)^{T-t}} \\ &= \frac{1 - (1+Y_{t,T}^c)^{-(T-t)}}{1 - (1+Y_{t,T}^c)^{-1}}, \end{aligned}$$

because when $|x| < 1$,

$$\sum_{k=0}^n kx^k = \frac{(nx - n - 1)x^{n+1} + x}{(1-x)^2}.$$

Duration can be used to find approximate linear relationships between log-coupon yields and holding period returns. Extending the formula of zero-coupon bonds (where duration is equal to maturity) to coupon bonds, we have

$$r_{t+1}^c = D_{t,T}^c y_{t,T}^c - (D_{t,T}^c - 1) y_{t+1,T}^c,$$

Shiller (1979) proposes a *linearization* which takes duration as constant and considers the following approximation in the neighbourhood $y_{t,T} = y_{t+1,T} = \bar{y} = C$:

$$\begin{aligned} H_{t,T} &\simeq D_T y_{t,T} - (D_T - 1) y_{t+1,T} \\ D_T &= \frac{1 - \left(1 + \bar{Y}_{t,T}^c\right)^{-(T-t)}}{1 - \left(1 + \bar{Y}_{t,T}^c\right)^{-1}} \\ D_T &= \frac{1 - \gamma^{T-t-1}}{1 - \gamma} = \frac{1}{1 - \gamma_T} \\ \gamma_T &= \left\{ 1 + \bar{Y}_{t,T}^c \left[1 - 1/(1 + \bar{Y}_{t,T}^c)^{T-t-1} \right]^{-1} \right\}^{-1} \\ \lim_{T \rightarrow \infty} \gamma_T &= \gamma = 1/(1 + \bar{y}) \end{aligned}$$

Solving this expression forward, we generate the equivalent of the DDG model in the bond market:

$$y_{t,T} = \sum_{j=0}^{T-t-1} \gamma^j (1 - \gamma) H_{t+j,T} + \gamma^{T-t} y_{T-1,T}$$

In this case, by equating one-period risk-adjusted returns, we have

$$E \left[\frac{y_{t,T} - \gamma y_{t+1,T}}{1 - \gamma} \mid I_t \right] = r_t + \phi_{t,T} \quad (2.12)$$

From the above expression, by recursive substitution, under the terminal condition

that at maturity the price equals the principal, we obtain:

$$y_{t,T} = y_{t,T}^* + E[\Phi_T | I_t] = \frac{1 - \gamma}{1 - \gamma^{T-t}} \sum_{j=0}^{T-t-1} \gamma^j E[r_{t+j} | I_t] + E[\Phi_T | I_t] \quad (2.13)$$

where the constant $\Phi_{t,T}$ is the term premium over the whole life of the bond:

$$\Phi_{t,T} = \frac{1 - \gamma}{1 - \gamma^{T-t}} \sum_{j=0}^{T-t-1} \gamma^j \phi_{t+j,T}$$

For long-bonds, when $T - t$ is very large, we have :

$$y_{t,T} = y_{t,T}^* + E[\Phi_T | I_t] = (1 - \gamma) \sum_{j=0}^{T-t-1} \gamma^j E[r_{t+j} | I_t] + E[\Phi_T | I_t]$$

Subtracting the risk-free rate from both sides of this equation we have

$$\begin{aligned} S_{t,T} &= y_{t,T} - r_t = \sum_{j=1}^{T-1} \gamma^j E[\Delta r_{t+j} | I_t] + E[\Phi_T | I_t] \\ &= S_{t,T}^* + E[\Phi_T | I_t] \end{aligned}$$

2.4 Going to the data with R

A statistical package is needed to take all the modelling steps in practice. Going from loading the data to transforming them, graphing them, analyzing them, modelling them and using them to estimate parameters and generate model-based simulations and, eventually, producing a report. We will use R and run R through the user interface RStudio. When using R and Rstudio Chat GPT can be used as an online assistant. If you ask Chat GPT "what is R Studio?" you will get an answer similar to the following one.

RStudio is an integrated development environment (IDE) for the R programming language. It provides a user-friendly and feature-rich environment for working with R, making it easier for data scientists, statisticians, and researchers to write, run, and manage R code. Here are some key features and components of RStudio:

- Script Editor: RStudio includes a script editor with features like syntax highlighting, code completion, and code formatting to enhance your coding experi-

ence.

- **Console:** It has an interactive R console that allows you to run R code and see the results in real-time. You can also save your command history for future reference.
- **Data Viewer:** RStudio provides a data viewer that lets you examine and manipulate data frames and other data structures visually.
- **Plots and Visualization:** You can create and customize plots and charts using R's graphical capabilities. RStudio displays plots within the IDE, making it easier to visualize your data.
- **Package Management:** It simplifies the management of R packages, allowing you to install, update, and load packages with ease.
- **Integrated Help:** RStudio integrates with R's documentation and help system, making it simple to access information about functions, packages, and syntax.
- **Version Control:** It offers version control integration with Git and SVN, allowing you to manage your R projects and code revisions efficiently.
- **Project Management:** RStudio supports project organization, making it easy to keep your code, data, and files organized within a project folder.
- **R Markdown:** RStudio supports R Markdown, a format for creating dynamic documents that combine R code, text, and visualizations. This is useful for creating reports, presentations, and publications.
- **Shiny:** RStudio includes Shiny, a web application framework for building interactive web applications with R. This is particularly useful for creating data dashboards and interactive data applications.

2.4.1 Getting Started

To get started with R and Rstudio you should first install them, then learn what R packages R, how to install them and how to call them. A very useful short piece by [Torfs and Brauer \(2014\)](#) provides all the relevant information to download, install and get started with R and RStudio. You should also be aware of how to get help from the web. The best way to do so is to ask Chat GPT.

Ask Chat GPT

Here is the Chat GPT answer to the question "How can you help me in using R with RStudio?"

I can assist you in using R with RStudio in various ways, including:

- **Coding and Syntax Assistance:** I can provide explanations, examples, and code snippets to help you understand and write R code in RStudio. If you encounter errors or have questions about specific functions or packages, feel free to ask for guidance.
- **Data Manipulation and Analysis:** I can assist you in performing data manipulation, cleaning, and analysis tasks in RStudio. Whether you need help with data frames, statistical analysis, or visualization, I can provide guidance and suggestions.
- **Package Recommendations:** If you're looking for R packages to solve specific tasks or problems, I can suggest relevant packages and explain how to install and use them in your RStudio environment.
- **R Markdown and Reports:** I can assist you in creating dynamic reports using R Markdown in RStudio. I can provide guidance on formatting, adding code chunks, and generating documents with embedded R code and output.
- **Plotting and Visualization:** I can help you create various types of plots and visualizations in RStudio using packages like ggplot2, base R graphics, and others. I can also explain how to customize and export your plots.
- **Debugging and Troubleshooting:** If you encounter errors or bugs in your R code within RStudio, I can help you understand and debug the issues. I can provide strategies for identifying and fixing common programming problems.
- **Version Control with Git:** I can guide you through the process of using Git for version control within RStudio. This includes initializing repositories, committing changes, and collaborating with others using Git.
- **Project Organization:** I can provide advice on organizing your R projects within RStudio, including creating project directories, managing files, and maintaining a tidy project structure.
- **Customization and Configuration:** I can help you customize your RStudio environment by explaining how to change themes, set preferences, and install extensions or add-ons that enhance your workflow.

- **Tips and Best Practices:** I can offer tips and best practices for efficient coding, data analysis, and project management in RStudio to improve your productivity.

To get assistance with any specific R-related task or question you have in RStudio, simply provide me with the details of your problem or topic of interest, and I'll do my best to provide guidance and support.

2.4.2 Data Objects in R

When working with data in R the first step is to know how to identify them and organize them. R makes available various common **datatypes**: double, integer, complex, logical, character, factor, date and time, and missing data. Data are then organized into datasets that are called **data structures**. The basic data structures we shall use are: vector, matrix, array, dataframes, tibble, time-series, and lists. Chapter 2 of [Singh and Allen \(2017\)](#) provides a good introduction to data objects in R.

Ask Chat GPT

Here is the Chat GPT answer to the question "What is the difference between a data-frame a tibble and a time-series object in R?"⁷ In R, dataframes, tibbles, and time-series objects are all data structures used for different purposes, and they have distinct characteristics and functionality:

1. **Dataframe.** A dataframe is a fundamental data structure in R. It is a two-dimensional tabular data structure where columns can contain different data types (numeric, character, factor, etc.). Dataframes are often used for storing and manipulating structured data, such as data imported from spreadsheets or databases. Dataframes are versatile and widely used for various data analysis tasks.
2. **Tibble.** A tibble is an enhanced dataframe introduced as part of the tidyverse ecosystem in R, available through the tibble package. Tibbles are designed to be more user-friendly and consistent than traditional dataframes. They have stricter rules, such as not automatically converting strings to factors, which can lead to fewer surprises. Tibbles are particularly well-suited for data wrangling and analysis with tools like dplyr and ggplot2.

⁷Chat GPT will also provide example codes on how to create dataframe, tibbles and time-series objects

- 3. Time-Series Object.** A time-series object is used specifically for handling time-series data, where data points are associated with specific timestamps or time intervals. Time-series objects can be created using various packages, such as the `ts` function from base R, or with packages like `xts` and `zoo` for more advanced time-series manipulation. Time-series objects have additional attributes and methods for time-based operations, like lagging, differencing, and time-based subsetting. They are essential for time-series analysis and forecasting tasks.

In summary, the choice between a dataframe, tibble, or time-series object depends on your specific data and analysis needs. Dataframes and tibbles are suitable for general data analysis, while time-series objects are specialized for handling time-dependent data. Tibbles, in particular, provide a more consistent and user-friendly experience for data manipulation tasks, which can be beneficial in many scenarios.

2.4.3 Data Handling in R

Data Handling involves getting the data, import them into R and pre-processing them.

Importing Data into R

To get data from the web in R one can use packages¹ for importing data, such as *quantmod*, that allow to get data from yahoo.finance or from the online Federal Reserve Database FRED (<https://fred.stlouisfed.org/>). Alternatively, data can be accessed from web provider by providing the appropriate URL or via Application Programme Interfaces (API) that can be run within R. Data from Excel or other foreign languages can also be read into R. Time-series Financial Data in Excel format used in this book have been downloaded Robert Shiller's webpage (<http://www.econ.yale.edu/~shiller/>) and Ken French's webpage (http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.htm). In the website associated to the book several example programmes to download and import data in R are made available.

An Illustrative R program

The following code, after preliminaries (such as setting the working directory and running all the relevant packages, after making sure that they are all available, downloads data from Yahoo Finance and the Fred Website illustrates how to change their frequency and how to save them locally in EXCEL format, it also shows how data available from a specific URL can be downloaded and organized.

¹
² `#clear the environment`

```

3  rm(list=ls())
4
5  setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
6  # packages used
7  listofpackages <- c("tidyverse","ellipse","reshape2","xts","xlsx","readxl",
8  "quantmod")
9  #installation of "xlsx" requires Java
10
11 for (j in listofpackages){
12   if(sum(installed.packages()[, 1] == j) == 0) {
13     install.packages(j)
14   }
15   library(j, character.only = T)
16 }
17
18 tickers <- c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
19             'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
20             'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','^DJI')
21             #,'^GSPC',,'^IRX')
22
23 #download the historical prices
24 getSymbols.yahoo(tickers,
25                 env = globalenv(),
26                 index.class = 'Date',
27                 from = "1985-01-31",
28                 to = "2023-07-31",
29                 periodicity = "monthly")
30
31 stocks =
32   merge(AXP[,6], AMGN[,6], AAPL[,6], BA[,6], CAT[,6], CSCO[,6], CVX[,6], GS[,6], HD[,6], HON[,6],
33         IBM[,6], INTC[,6], JNJ[,6], KO[,6], JPM[,6], MCD[,6], MMM[,6], MRK[,6], MSFT[,6], NKE[,6],
34         PG[,6], TRV[,6], UNH[,6], CRM[,6], VZ[,6], V[,6], WBA[,6], WMT[,6], DIS[,6], DOW[,6], DJI[,6])
35
36 colnames(stocks) <-
37   c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
38     'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
39     'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','DJI')
40
41 write.xlsx(as.data.frame(stocks), "2023_monthly_stocks.xlsx", row.names =
42           TRUE)
43
44 rm(list = c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
45            'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
46            'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','DJI'))
47
48 stocks_quarterly = to.quarterly(stocks$AXP)[,4]
49
50 for(i in 2:ncol(stocks)){
51   x = to.quarterly(stocks[, i])[,4]
52   stocks_quarterly = merge(stocks_quarterly, x)
53 }
54
55 colnames(stocks_quarterly) <-

```

```

51     c('AXP', 'AMGN', 'AAPL', 'BA', 'CAT', 'CSCO', 'CVX', 'GS', 'HD', 'HON',
52       'IBM', 'INTC', 'JNJ', 'KO', 'JPM', 'MCD', 'MMM', 'MRK', 'MSFT', 'NKE',
53       'PG', 'TRV', 'UNH', 'CRM', 'VZ', 'V', 'WBA', 'WMT', 'DIS', 'DOW', 'DJI')
54 write.xlsx(as.data.frame(stocks_quarterly), "2023_quarterly_stocks.xlsx",
55           row.names = TRUE)
56 # set tickers
57 tickers1 <- c("FEDFUNDS", "DGS10", "GDPPOT")
58 getSymbols.FRED(tickers1,
59               env = globalenv(),
60               return.class = "xts",
61               index.class = 'Date'
62             )
63 # import data from FRED database
64 quarterly_fedfunds <- apply.quarterly(FEDFUNDS, last)
65
66 urlfile="https://raw.githubusercontent.com/pcm-dpc/COVID-19/
67 master/dati-regioni/dpc-covid19-ita-regioni.csv"
68 COVID_all<-read_csv(url(urlfile)) #read_csv in package tidyverse
69 #COVID_all<-read.csv(url(urlfile)) #read.csv in package xls
70 LOMBARDIA=subset(COVID_all,denominazione_regione=="Lombardia",select=c(1:1,4:4,7:20))
71 VENETO=subset(COVID_all,denominazione_regione=="Veneto",select=c(1:1,4:4,7:20))

```

Pre-Processing Data

pre-processing data involves extracting subsets, combining dataframe, and constructing tables. Once the relevant database has been constructed data transformations are necessary to build returns from prices, to construct portfolios and to track their performance over time. R programming and the use of loops, control-flows and functions is very useful when the same operations are iterated for different series, think for example of constructing the returns on the 30 DJ stocks given their prices. Chapter 3 of [Singh and Allen \(2017\)](#) provides a good introduction to R programming and control flow.

2.4.4 Data Exploration and Graphics

Data Exploration and Graphics of the relevant transformed series complete the initial stage of the modelling process. Data exploration involves the use of descriptive univariate and multivariate statistics while graphics is implemented with the utilization of time-series plots, cross-sectional plots, histograms and QQ-plots. The packages *ggplot* and *highcharter* allow graphical analysis of data respectively formatted as tibble and time-series objects.

Ask Chat GPT

Here is the Chat GPT answer to the question "What can I do with ggplot in R?"

ggplot2 is a powerful data visualization package in R that allows you to create a wide range of high-quality plots and charts. It is based on the "Grammar of Graphics" concept, which means you can build complex plots by specifying their components in a structured and intuitive manner. Here are some of the things you can do with ggplot2 in R:

1. Scatter Plots: Create scatter plots to visualize the relationship between two continuous variables.

```
ggplot(data = df, aes(x = variable1, y = variable2)) + geom_point()
```

2. Line Plots: Generate line plots for time series or continuous data.

```
ggplot(data = df, aes(x = time, y = value)) + geom_line()
```

3. Bar Charts: Create bar charts to display the distribution of categorical data.

```
ggplot(data = df, aes(x = category, y = count)) + geom_bar(stat = "identity")
```

4. Histograms: Visualize the distribution of a single continuous variable.

```
ggplot(data = df, aes(x = variable)) + geom_histogram(binwidth = 1, fill = "blue", color = "black")
```

5. Box Plots: Generate box plots to display the distribution of a variable's quartiles and outliers.

```
ggplot(data = df, aes(x = category, y = value)) + geom_boxplot()
```

6. Density Plots: Create density plots to visualize the distribution of a variable as a smooth curve.

```
ggplot(data = df, aes(x = variable)) + geom_density()
```

7. Heatmaps: Generate heatmaps to display patterns in a matrix of data.

```
ggplot(data = df, aes(x = variable1, y = variable2, fill = value)) + geom_tile()
```

8. Faceting: Split your data into multiple smaller plots based on a categorical variable.

```
ggplot(data = df, aes(x = variable1, y = variable2)) + geom_point() + facet_wrap(category)
```

9. Customize Aesthetics: You can customize the appearance of your plots by modifying colors, labels, titles, and more using ggplot2's extensive options.

```
ggplot(data = df, aes(x = variable1, y = variable2)) + geom_point(color =
"red")+labs(title = "MyScatterPlot", x = "X-axislabel", y = "Y-axislabel")
```

10. Combine Layers: Combine multiple geometric layers to create complex plots.

```
ggplot(data = df, aes(x = variable1, y = variable2))+geom_point()+geom_smooth(method =
"lm")
```

These are just some of the things you can do with ggplot2. It offers a flexible and versatile framework for data visualization in R, making it a popular choice for creating informative and visually appealing graphs and charts.

An Illustrative R Code

```
1 rm(list=ls())
2 #Check the working directory before importing else provide full path
3 #setwd(path)
4 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
5
6 # packages used
7 listofpackages <- c("dygraphs",
8   "tidyverse", "ellipse", "reshape2", "highcharter", "xts", "xlsx", "readxl", "quantmod")
9
10 for (j in listofpackages){
11   if(sum(installed.packages()[, 1] == j) == 0) {
12     install.packages(j)
13   }
14   library(j, character.only = T)
15 }
16
17 raw_data = read_xlsx("../data/2023_monthly_stocks.xlsx")
18 names(raw_data)[1] = 'Date'
19 typeof(raw_data)
20 typeof(raw_data$Date)
21 typeof(raw_data$AXP)
22 typeof(raw_data$CSCO)
23
24 dates <- seq(as.Date("1985-02-01"), length=462, by="months")
25 params <- c("Date", "AXP", "AMGN", "AAPL", "BA", "CAT", "CSCO", "DJI")
26 data <- raw_data[, c(params)]
27 data <- na.omit(data)
28 data <- data %>%
29   mutate(Date = as.Date(Date, format = "%Y-%m-%d"))
30
31 params1 <- c("AXP", "AMGN", "AAPL", "BA", "CAT", "CSCO", "DJI")
32 tsdata <- xts(raw_data[, c(params1)], order.by=dates) # creates a time
33   series object
```

```

32 tsdata <- na.omit(tsdata) # omitting the rows with NA presence
33 data<- na.omit(data)
34 ## having created the database with all observation we generate a subset
35 #tsdata1 <- tsdata["1992-02-01/1993-02-01"]
36 #data=subset(data,select=c(1:12))
37
38 ## -----
39 # DATA TRANSFORMATIONS
40 ## -----
41 #1. from prices to returns
42 # exact monthly returns
43 t1<-nrow(data)
44 data$AXP_ret <- data$AMGN_ret <- array(data = NA, dim = t1)
45 for (i in 2:t1) {
46   data[i, "AMGN_ret"][[1]]=(data[i, "AMGN"][[1]]-data[i-1,
47     "AMGN"][[1]])/data[i-1, "AMGN"][[1]]
48   data[i, "AXP_ret"][[1]]=(data[i, "AXP"][[1]]-data[i-1,
49     "AXP"][[1]])/data[i-1, "AXP"][[1]]
50 }
51 # the following lines of R apply the same transformation to
52 # two series AXP and AMGN available in .xts format in a frame called tsdata.
53 # Could you do the same transformation in a more parsimonious way by having
54 # a loop over the serie names AXP and AMGN ?
55
56 series_names <- c("AAPL", "BA", "CAT", "CSCO", "DJI")
57
58 for (name in series_names) {
59   return_col_name <- paste0(name, "_ret")
60   data[, return_col_name] <- array(data = NA, dim = t1)
61   for (i in 2:nrow(data)) {
62     data[i, return_col_name] [[1]] <- (data[i, name] [[1]] - data[i - 1,
63       name] [[1]]) / data[i - 1, name] [[1]]
64   }
65 }
66
67 # same in .xts
68 t1<-nrow(tsdata)
69
70 tsdata$AXP_ret <- tsdata$AMGN_ret<- tsdata$AAPL_ret<- tsdata$BA_ret<-
71   array(data = NA, dim = t1)
72 tsdata$CAT_ret<-tsdata$CSCO_ret <- tsdata$DJI_ret<- array(data = NA, dim =
73   t1)
74 for (i in 2:t1) {
75   tsdata[i, "AMGN_ret"] [[1]]=(tsdata[i, "AMGN"] [[1]]-tsdata[i-1,
76     "AMGN"] [[1]])/data[i-1, "AMGN"] [[1]]
77   tsdata[i, "AXP_ret"] [[1]]=(tsdata[i, "AXP"] [[1]]-tsdata[i-1,
78     "AXP"] [[1]])/data[i-1, "AXP"] [[1]]
79   # tsdata[i, "AAPL_ret"] [[1]]=(tsdata[i, "AAPL"] [[1]]-tsdata[i-1,
80     "AAPL"] [[1]])/data[i-1, "AAPL"] [[1]]
81   # tsdata[i, "BA_ret"] [[1]]=(tsdata[i, "BA"] [[1]]-tsdata[i-1,
82     "BA"] [[1]])/data[i-1, "BA"] [[1]]

```

```

75 #   tsdata[i, "CAT_ret"][[1]]=(tsdata[i, "CAT"][[1]]-tsdata[i-1,
76   #   "CAT"][[1]])/data[i-1, "CAT"][[1]]
77 #   tsdata[i, "CSCO_ret"][[1]]=(tsdata[i, "CSCO"][[1]]-tsdata[i-1,
78   #   "CSCO"][[1]])/data[i-1, "CSCO"][[1]]
79 #   tsdata[i, "DJI_ret"][[1]]=(tsdata[i, "DJI"][[1]]-tsdata[i-1,
80   #   "DJI"][[1]])/data[i-1, "DJI"][[1]]
81 }
82 # the loop is a bit different in .xts
83 series_names <- c("AAPL", "BA", "CAT", "CSCO", "DJI")
84
85 for (name in series_names) {
86   return_col_name <- paste0(name, "_ret")
87   temporary_column <- array(data = NA, dim = t1)
88
89   tsdata <- merge.xts(tsdata, temporary_column) # add last column
90   colnames(tsdata)[ncol(tsdata)] = return_col_name # rename it
91
92   for (i in 2:nrow(data)) {
93     tsdata[i, return_col_name] <- (tsdata[i, name][[1]] - tsdata[i - 1,
94     name][[1]]) / tsdata[i - 1, name][[1]]
95   }
96 }
97
98 # buy and hold returns
99 ## what would happen had we invested $1 in the DJI and AXP at t0
100 ## initializing values
101 data$DJI_cum <- data$AXP_cum <- array(data = NA, dim = nrow(data))
102
103 data[1, c("DJI_cum", "AXP_cum")] <- 1
104 t1<-nrow(data)
105 for (i in 2:t1) {
106   data[i, "DJI_cum"][[1]]=data[i-1, "DJI_cum"][[1]]*(1+data[i,
107   "DJI_ret"][[1]])
108   data[i, "AXP_cum"][[1]]=data[i-1, "AXP_cum"][[1]]*(1+data[i,
109   "AXP_ret"][[1]])
110 }
111
112 tsdata$DJI_cum <- array(data = NA, dim = nrow(tsdata))
113 tsdata$AXP_cum <- array(data = NA, dim = nrow(tsdata))
114 tsdata[1, c("DJI_cum", "AXP_cum")] <- 1
115 t1<-nrow(data)
116 for (i in 2:t1) {
117   tsdata[i, "DJI_cum"][[1]]=tsdata[i-1, "DJI_cum"][[1]]*(1+tsdata[i,
118   "DJI_ret"][[1]])
119   tsdata[i, "AXP_cum"][[1]]=tsdata[i-1, "AXP_cum"][[1]]*(1+tsdata[i,
120   "AXP_ret"][[1]])
121 }
122
123 ## -----
124 # monthly log stock returns

```

```

119 ## -----
120 data$DJI_lp<-log(data$DJI_cum)
121 data$AXP_lp<-log(data$AXP_cum)
122 data$DJI_lret <- c(NA,diff(data$DJI_lp))
123 data$AXP_lret <- c(NA,diff(data$AXP_lp))
124 # value of a buy-and-hold portfolio using cumulative log returns
125 data$DJI_cum1 <- array(data = NA, dim = nrow(data))
126 data[1, c("DJI_cum1")] <- 1
127 for (i in 2:t1) {
128   data[i, "DJI_cum1"][[1]]=data[i-1, "DJI_cum1"][[1]]*(1+data[i,
129     "DJI_lret"][[1]])
130 }
131 tsdata$DJI_lp<-log(tsdata$DJI_cum)
132 tsdata$AXP_lp<-log(tsdata$AXP_cum)
133 tsdata$DJI_lret <- diff(tsdata$DJI_lp)
134 tsdata$AXP_lret <- diff(tsdata$AXP_lp)
135 tsdata$DJI_cum1 <- array(data = NA, dim = nrow(tsdata))
136 tsdata[1, c("DJI_cum1")] <- 1
137 for (i in 2:nrow(tsdata)) {
138   tsdata[i, "DJI_cum1"][[1]]=tsdata[i-1, "DJI_cum1"][[1]]*(1+tsdata[i,
139     "DJI_lret"][[1]])
140 }
141 tsdata.df <- as.data.frame(tsdata)
142 save(data, file='data.Rdata')
143 save(tsdata, file='tsdata.Rdata')
144 save(tsdata.df, file='tsdata.df.Rdata')
145
146 ## -----
147 # time-series plots
148 #-----
149 #(1) plot .xts series
150 plot(tsdata$DJI_ret, col = "blue", lwd = 2, main = "", ylab = "")
151 lines(tsdata$AXP_ret, col = "green", lwd = 2)
152 addLegend("topleft",
153   legend.names = c("DJI", "AXP"),
154   lty = c(1, 1), lwd = c(2, 2),
155   col = c("blue", "green"))
156 dev.copy2pdf(width = 5.72, out.type = "pdf",file="Figure_1xts.pdf")
157 dev.off()
158 #(2) use highchart with .xts series
159 highchart(type = "stock") %>%
160   hc_title(text = "Monthly Log Returns") %>%
161   hc_add_series(data=tsdata[, "DJI_ret"], name = "DJI_ret")%>%
162   hc_add_series(data=tsdata[, "AXP_ret"], name = "AXP_ret")%>%
163   hc_add_theme(hc_theme_flat()) %>%
164   hc_navigator(enabled = FALSE) %>%
165   hc_scrollbar(enabled = FALSE) %>%
166   hc_exporting(enabled = TRUE) %>%
167   hc_legend(enabled = TRUE)
168

```

```

169 # (2) use ggplot with the standard dataframe
170 plot <- ggplot(data, aes(x = Date)) +
171   geom_point(aes(y = DJI_ret, color = "DJI"), size = 2) +
172   geom_point(aes(y = AXP_ret, color = "AXP"), size = 2) +
173   labs(title = "Returns",
174        x = "Time", y = " Value") +
175   scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
176   theme_minimal() +
177   theme(axis.line = element_line(color = "black")) #+
178
179 print(plot)
180 ggsave(filename = "Figure_1.pdf", plot = plot, device = "pdf", width =
181         5.72, height = 3.12)
182 # dev.copy2pdf(width = 4, out.type = "pdf", file = "Figure_1.pdf")
183 # dev.off()
184 ## -----
185 # comparing returns and log-returns
186 #-----
187
188 plot(tsdata$DJI_ret, ylab = "Returns", main = "S&P500 ", col = "blue", lwd
189      = 2)
189 lines(tsdata$DJI_lret, col = "red")
190
191 # time-series plot of cumulative returns
192 plot(tsdata$DJI_cum,
193      type = "l", col = "red", ylim = c(0, 15),
194      ylab = "cumulative return mkt")
195 lines(tsdata$DJI_cuml, col = "blue", type = "l", ylab = "cumulative log
196        return mkt")
197
198 # cross-plot of exact and log-linearized returns
199 plot(x = data$DJI_ret, y = data$DJI_lret, col = "red")
200 lines(x = data$DJI_ret, y = data$DJI_lret, col = "blue")
201
202 # cross-plot of returns of AXP and their value predicted from the market
203 fm1 <- lm(AXP_ret ~ DJI_ret, data = data)
204 summary(fm1)
205 data$AXP_retfit <- c(NA, fitted(fm1))
206 plot(x = data$DJI_ret, y = data$AXP_ret, col = "red")
207 lines(x = data$DJI_ret, y = data$AXP_retfit, col = "blue")
208
209 plotactfit <- ggplot(data, aes(x = DJI_ret, y = AXP_ret)) +
210   geom_point(color = "red") +
211   geom_line(aes(x = DJI_ret, y = AXP_retfit), color = "blue") +
212   geom_hline(yintercept = 0, linetype = "dashed", color = "black") #
213   Adding the zero line
214
215 # Display the plot
216 print(plotactfit)

```

```

217 #-----
218 #plotting prices
219 #-----
220 sfDJI<- as.numeric(tsdata$DJI[1])
221 sfAXP<- as.numeric(tsdata$AXP[1])
222 plot(tsdata$DJI/sfDJI,col = "blue",lwd = 2)
223 lines(tsdata$AXP/sfAXP, col = "green",lwd = 2)
224 addLegend("topleft", on=1,
225           legend.names = c("DJIRs", "AXPrs"),
226           lty=c(1, 1), lwd=c(2, 1),
227           col=c("blue", "green", "red"))
228
229 # you can interact with Chat GPT to improve on this version of the graphs
230
231 #First Question When I run the following sequence in R I get a graph with
232   tsdata$DJI
233 #written at the top left of it. How do I remove this from the graph ?
234 #ANSWER
235
236 plot(tsdata$DJI/3267.70, col = "blue", lwd = 2, main = "", ylab = "")
237 lines(tsdata$AXP/3.277914, col = "green", lwd = 2)
238 addLegend("topleft",
239           legend.names = c("DJI", "AXPrs"),
240           lty = c(1, 1), lwd = c(2, 2),
241           col = c("blue", "green")) # Remove "red" from col argument
242
243 #Second Question> I would like to have the same graph in a double scale
244 # with DJI on the left hand scale and AXP on the right hand scale
245 combined_data <- data.frame(DJI = tsdata$DJI, AXP = tsdata$AXP )
246 dygraph(combined_data, main = "Double-Scale Time Series Graph") %>%
247   dySeries("DJI", label = "DJI", color = "blue") %>%
248   dySeries("AXP", label = "AXP", color = "green", axis = "y2") %>%
249   dyAxis("y", label = "DJI") %>%
250   dyAxis("y2", label = "AXP", independentTicks = TRUE) %>%
251   dyLegend(width = 250)
252
253
254 #-----
255 #plotting series from a list using GGPLOT
256 #-----
257 plot <- ggplot(data, aes(x = Date)) +
258   geom_line(aes(y = DJI/3267.70, color = "DJI"), size = 2) +
259   geom_line(aes(y = AXP/3.277914, color = "AXP"), size = 2) +
260   labs(title = "Trends",
261         x = "Time", y = " Value") +
262   scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
263   theme_minimal() +
264   theme(axis.line = element_line(color = "black")) +
265   scale_x_continuous(breaks = data$Date, labels = data$Date) # Add this
266     line for x-axis labels

```

```

267 # Print the plot
268 print(plot)
269
270 #Ask Chat GPT: When I run the following code in R I get "too many "
      labels on the x axis,
271 #how can I reduce the number of labels (say one every 5 years) ?
272 #Answer
273 #To reduce the number of x-axis labels in your ggplot, you can use the
      scale_x_date() function
274 #with the date_breaks argument to specify the intervals at which you want
      the labels to appear.
275 #In your case, you want to display labels every 5 years. Here's how you
      can modify your code to achieve this:
276 ggplot(data, aes(x = Date)) +
277   geom_line(aes(y = DJI/3267.70, color = "DJI"), size = 1) +
278   geom_line(aes(y = AXP/3.277914, color = "AXP"), size = 1) +
279   labs(title = "Trends",
280        x = "Time", y = " Value") +
281   scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
282   theme_minimal() +
283   theme(axis.line = element_line(color = "black")) +
284   scale_x_date(date_breaks = "5 years", date_labels = "%Y")
285 #In the code above:
286 #scale_x_date() is used to control the x-axis (date) scale.
287 #date_breaks = "5 years" specifies that you want to display labels every
      5 years.
288 #date_labels = "%Y" specifies the date format you want to use for the
      labels (in this case, the year only).
289 #This should result in a plot with x-axis labels appearing every 5 years,
      making the plot more readable when you have a large time series
      dataset. Adjust the date_breaks argument as needed to control the
      spacing of the labels according to your preferences.
290
291
292
293 ## -----
294 # combine several plots on one canvas
295 ## -----
296
297 par(mfrow = c(2, 2))
298
299 plot(tsdata$DJI_ret, ylab = "Returns", main = "DJ30 ", col = "blue", lwd
      = 2)
300
301 plot(x=data$DJI_ret, y=data$DJI_lret, col="red")
302 lines(x=data$DJI_ret, y=data$DJI_ret,col = "blue")
303
304 plot(x=data$AXP_ret, y=data$AXP_lret, col="red",ylim = c(-0.5, 1))
305 lines(x=data$AXP_ret, y=data$AXP_ret,col = "blue")
306
307 plot(tsdata$DJI_cum,
308      type = "l", col = "red", ylim = c(0, 12),main = "DJ30 ",

```

```

309     ylab = "cumulative return mkt")
310
311 par(mfrow = c(1, 1))
312
313
314 ## -----
315 # HISTOGRAMS AND QQ PLOTS
316 ## -----
317
318 ## Histograms
319 s1 <- na.omit(tpdata$DJI_ret)
320 hist(s1, breaks = seq(min(s1), max(s1), l = 20+1), prob=TRUE, main =
321     "histogram of monthly returns")
322 curve(dnorm(x, mean=mean(s1), sd=sd(s1)), col='darkblue', lwd=2, add=TRUE)
323
324 ## Histograms with Highcharter using .xts data
325
326 hc_hist <- hist(tpdata$DJI_lret, breaks = 50, plot = FALSE)
327 hchart(hc_hist, color = "cornflowerblue")%>%
328     hc_title(text =
329         paste("DJI",
330             "Log Returns Distribution",
331             sep = " ")) %>%
332     hc_add_theme(hc_theme_flat()) %>%
333     hc_exporting(enabled = TRUE) %>%
334     hc_legend(enabled = FALSE)
335
336 hc_hist <- hist(tpdata[, "DJI_lret"], breaks = 50, plot = FALSE)
337 hchart(hc_hist, color = "cornflowerblue")%>%
338     hc_title(text =
339         paste("DJI",
340             "Log Returns Distribution",
341             sep = " ")) %>%
342     hc_add_theme(hc_theme_flat()) %>%
343     hc_exporting(enabled = TRUE) %>%
344     hc_legend(enabled = FALSE)
345 ## -----
346
347 qqplot(tpdata.df$DJI_ret,
348     tpdata.df$DJI_lret,
349     ylim = c(-0.15, 0.15), xlim = c(-0.15, 0.15),
350     ylab = "monthly return. log approximation",
351     xlab = "monthly return. exact computation",
352     main = "Quantile-Quantile plot (Q-Q plot)")
353 mod5 <- lm(tpdata.df$DJI_ret ~ tpdata.df$DJI_lret)
354 abline(reg = mod5, col = "red")
355
356 ## qq-plot versus normal dist
357 qqnorm(tpdata$DJI_ret,
358     ylim = c(-0.15, 0.15), ylab = "monthly return. sample quantile",
359     xlab = "monthly return. theoretical quantiles",
360     main = "Normal (Q-Q plot)")

```

```

360 qqline(tsdata$DJI_ret, datax = FALSE, distribution = qnorm,
361         probs = c(0.25, 0.75), qtype = 7)
362
363 ## -----
364 # CORRELATION ANALYSIS
365 ## -----
366 tsdata.df <- as.data.frame(tsdata)
367 # Select specific columns and observations from the start date onward
368 selected_cols <- c("AMGN_ret", "AXP_ret", "AAPL_ret", "BA_ret",
369                  "CAT_ret", "CSCO_ret", "DJI_ret")
370 datashow <- subset(tsdata.df[, selected_cols])
371 datashow <- na.omit(datashow)
372 # Print the resulting subset
373 summary(datashow) # this is very useful to get a grip on the data
374 structure
375 mean(datashow[, "AMGN_ret"])
376 sd(datashow[, "AMGN_ret"])
377 var(datashow[, "AMGN_ret"])
378 cor(datashow)
379 cor.datacor = cor(datashow, use="complete.obs")
380 cor.datacor
381
382 ## -----
383 ord <- order(cor.datacor[1,])
384 ordered.cor.datacor <- cor.datacor[ord, ord]
385 plotcorr(ordered.cor.datacor, col=cm.colors(11)[5*ordered.cor.datacor +
386         6])
387
388 ## -----
389 cormat <- round(cor(datashow), 2)
390 head(cormat)
391 melted_cormat <- melt(cormat)
392 head(melted_cormat)
393 ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
394   geom_tile()
395 # Get lower triangle of the correlation matrix
396 get_lower_tri <- function(cormat){
397   cormat[upper.tri(cormat)] <- NA
398   return(cormat)
399 }
400 # Get upper triangle of the correlation matrix
401 get_upper_tri <- function(cormat){
402   cormat[lower.tri(cormat)] <- NA
403   return(cormat)
404 }
405 upper_tri <- get_upper_tri(cormat)
406 upper_tri
407 # Melt the correlation matrix
408 melted_cormat <- melt(upper_tri, na.rm = TRUE)
409 # Heatmap
410 ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+

```

```

409   geom_tile(color = "white")+
410   scale_fill_gradient2(low = "blue", high = "red", mid = "white",
411                       midpoint = 0, limit = c(-1,1), space = "Lab",
412                       name="Pearson\nCorrelation") +
413   theme_minimal()+
414   theme(axis.text.x = element_text(angle = 45, vjust = 1,
415                                   size = 12, hjust = 1))+
416   coord_fixed()

```

2.4.5 Interacting with Chat GPT

There many ways to use to use ChatGPT to learn R. The more precise is the query, the more precise will be the answer. But in any case interaction is fundamental for two reasons: either because chatCPT may not provide the exact answer to your question or because the snippet you receive in the answer might be "close" to the one that works but non quite there. One can think of three possible ways to interact with ChatGPT (1) Ask to generate a code snippet based on your query (2) Ask ChatGPT to explain a code snippet or a part of it that you do not understand (3) Ask ChatGPT to modify a code snippet of your or suggest improvements. In all of these three cases some interaction will be required before converging to a solution. Convergence will be much faster in case (3) than in case (1) , case (2) will be intermediate in that you will get a clear explanation but putting it at work in solving the specific problem at your hand will require some more effort. To illustrate a case of interaction with ChatGPT think of a generic USER who has found on the web the following R programme that computes the frontier and the efficient frontier for sample portfolio made of two and three assets.

```

1  #clear the environment
2  rm(list=ls())
3  ## -----
4  setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
5  library(data.table)
6  library(scales)
7  library(ggplot2)
8  library(xts)
9  link <-
    "https://raw.githubusercontent.com/DavZim/Efficient_Frontier/master/data/fin_data.csv"
10 dt <- fread(link)
11 dt[, date := as.Date(date)]
12
13 # create indexed values
14 dt[, idx_price := price/price[1], by = ticker]
15
16 # plot the indexed values
17 ggplot(dt, aes(x = date, y = idx_price, color = ticker)) +
18   geom_line() +

```

```

19   # Miscellaneous Formatting
20   theme_bw() + ggtitle("Price Developments") +
21   xlab("Date") + ylab("Price\n(Indexed 2000 = 1)") +
22   scale_color_discrete(name = "Company")
23   # calculate the arithmetic returns
24   dt[, ret := price / shift(price, 1) - 1, by = ticker]
25
26   # summary table
27   # take only non-na values
28   tab <- dt[!is.na(ret), .(ticker, ret)]
29
30   # calculate the expected returns (historical mean of returns) and
      volatility (standard deviation of returns)
31   tab <- tab[, .(er = round(mean(ret), 4),
32                     sd = round(sd(ret), 4)),
33               by = "ticker"]
34   ggplot(tab, aes(x = sd, y = er, color = ticker)) +
35     geom_point(size = 5) +
36     # Miscellaneous Formatting
37     theme_bw() + ggtitle("Risk-Return Tradeoff") +
38     xlab("Volatility") + ylab("Expected Returns") +
39     scale_y_continuous(label = percent, limits = c(0, 0.03)) +
40     scale_x_continuous(label = percent, limits = c(0, 0.1))
41
42
43
44   # load the data
45   link <-
      "https://raw.githubusercontent.com/DavZim/Efficient_Frontier/master/data/mult_assets.csv"
46   df <- data.table(read.csv(link))
47
48   df_table <- melt(df)[, .(mean = mean(value), sd = sd(value)), by = variable]
49
50   er_x <- mean(df$x)
51   er_y <- mean(df$y)
52   er_z <- mean(df$z)
53   sd_x <- sd(df$x)
54   sd_y <- sd(df$y)
55   sd_z <- sd(df$z)
56   cov_xy <- cov(df$x, df$y)
57   cov_xz <- cov(df$x, df$z)
58   cov_yz <- cov(df$y, df$z)
59
60   # two assets
61   two_assets_seq <- seq(from = 0, to = 1, length.out = 1000)
62
63   two <- data.table(wx = two_assets_seq,
64                     wy = 1 - two_assets_seq)
65
66   two[, ':=', (er_p = wx * er_x + wy * er_y,
67               sd_p = sqrt(wx^2 * sd_x^2 +
68                             wy^2 * sd_y^2 +

```

```

69             2 * wx * (1 - wx) * cov_xy))]
70
71 # plot_two <- ggplot() +
72 #   geom_point(data = two, aes(x = sd_p, y = er_p, color = wx)) +
73 #   geom_point(data = df_table[variable != "z"],
74 #             aes(x = sd, y = mean), color = "red", size = 3, shape = 18) +
75 #   theme_bw() + ggtitle("Possible Portfolios with Two Risky Assets") +
76 #   xlab("Volatility") + ylab("Expected Returns") +
77 #   scale_y_continuous(label = percent, limits = c(0, max(two$er_p) * 1.2))
78 #   +
79 #   scale_x_continuous(label = percent, limits = c(0, max(two$sd_p) * 1.2))
80 #   +
81 #   scale_color_continuous(name = expression(omega[x]), labels = percent)
82 #
83 # ggsave(plot_two, file = "two_assets.png", scale = 1, dpi = 600)
84
85 ggplot() +
86 geom_point(data = two, aes(x = sd_p, y = er_p, color = wx)) +
87 geom_point(data = df_table[variable != "z"],
88           aes(x = sd, y = mean), color = "red", size = 3, shape = 18) +
89 theme_bw() + ggtitle("Possible Portfolios with Two Risky Assets") +
90 xlab("Volatility") + ylab("Expected Returns") +
91 scale_y_continuous(label = percent, limits = c(0, max(two$er_p) * 1.2)) +
92 scale_x_continuous(label = percent, limits = c(0, max(two$sd_p) * 1.2)) +
93 scale_color_continuous(name = expression(omega[x]), labels = percent)
94
95 # three assets
96 three_assets_seq <- seq(from = 0, to = 1, length.out = 1000)
97
98 three <- data.table(wx = rep(three_assets_seq, each =
99                       length(three_assets_seq)),
100                    wy = rep(three_assets_seq, length(three_assets_seq)))
101
102 three[, wz := 1 - wx - wy]
103
104 three[, ':=', (er_p = wx * er_x + wy * er_y + wz * er_z,
105              sd_p = sqrt(wx^2 * sd_x^2 +
106                          wy^2 * sd_y^2 +
107                          wz^2 * sd_z^2 +
108                          2 * wx * wy * cov_xy +
109                          2 * wx * wz * cov_xz +
110                          2 * wy * wz * cov_yz))]
111
112 three <- three[wx >= 0 & wy >= 0 & wz >= 0]
113
114 plot_three <- ggplot() +
115   geom_point(data = three, aes(x = sd_p, y = er_p, color = wx - wz)) +
116   geom_point(data = df_table, aes(x = sd, y = mean), color = "red", size =
117     3, shape = 18) +
118   theme_bw() + ggtitle("Possible Portfolios with Three Risky Assets") +
119   xlab("Volatility") + ylab("Expected Returns") +
120   scale_y_continuous(label = percent, limits = c(0, max(three$er_p) * 1.2))

```

```

+
117 scale_x_continuous(label = percent, limits = c(0, max(three$sd_p) * 1.2))
+
118 scale_color_gradientn(colors = c("red", "blue", "yellow"),
119                        name = expression(omega[x] - omega[z]), labels =
                                percent)
120
121 ggsave(plot_three, file = "three_assets.png", scale = 1, dpi = 600)

```

The USER, after understanding the code, immediately realizes that the curse of dimensionality applies very rapidly and the procedure becomes very inefficient as the number of assets considered to form the portfolio gets larger.

So an interaction with Chat GPT begins to get a more general version of the code.

USER

I have a database containing sample of returns on five assets, say A,B,C,D,E. I would like to write an R code to derive the efficient frontier and plot it in the space of the mean and the standard deviation of the portfolio together with points identified by the mean and the standard deviation of the five assets. can you do it ?

Chat GPT

Certainly! To derive the efficient frontier and plot it in R, you can follow these steps. I'll provide you with a code template to get you started. Please make sure you have the necessary packages like quantmod and ggplot2 installed:

```

1 # Load necessary libraries
2 library(quantmod)
3 library(ggplot2)
4
5 # Sample returns for five assets (replace with your data)
6 returns <- data.frame(
7   A = c(0.05, 0.06, 0.07, 0.04, 0.03),
8   B = c(0.08, 0.09, 0.06, 0.07, 0.05),
9   C = c(0.10, 0.12, 0.11, 0.13, 0.09),
10  D = c(0.06, 0.05, 0.04, 0.03, 0.07),
11  E = c(0.07, 0.08, 0.09, 0.06, 0.05)
12 )
13
14 # Calculate mean and standard deviation of asset returns
15 mean_returns <- colMeans(returns)
16 sd_returns <- apply(returns, 2, sd)
17
18 # Number of random portfolios to simulate
19 n_portfolios <- 10000
20
21 # Create a function to simulate random portfolios
22 simulate_portfolios <- function(n_portfolios, returns) {
23   portfolio_returns <- numeric(n_portfolios)
24   portfolio_volatility <- numeric(n_portfolios)
25

```

```

26   for (i in 1:n_portfolios) {
27     weights <- runif(ncol(returns))
28     weights <- weights / sum(weights)
29     portfolio_returns[i] <- sum(weights * mean_returns)
30     portfolio_volatility[i] <- sqrt(t(weights) %% cov(returns) %% weights)
31   }
32
33   portfolio_data <- data.frame(Return = portfolio_returns, Volatility =
34     portfolio_volatility)
35   return(portfolio_data)
36 }
37 # Simulate random portfolios
38 set.seed(123) # For reproducibility
39 portfolios <- simulate_portfolios(n_portfolios, returns)
40
41 # Plot the efficient frontier
42 ggplot(portfolios, aes(x = Volatility, y = Return)) +
43   geom_point(alpha = 0.5) +
44   geom_point(aes(x = sd_returns, y = mean_returns), color = "red", size =
45     3) +
46   labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
47   ggtitle("Efficient Frontier") +
48   theme_minimal()

```

This code first calculates the mean and standard deviation of the returns for the five assets and then simulates a large number of random portfolios to derive the efficient frontier. It finally plots the efficient frontier along with points representing the mean and standard deviation of the five assets in the mean-risk space.

Make sure to replace the sample returns with your actual data for assets A, B, C, D, and E.

USER

this looks good but I doubt it will work with say 20 assets. What would you do if you need to derive exactly the efficient frontier with 20 assets ?

Chat GPT

When dealing with a larger number of assets, such as 20, you can still derive the efficient frontier using R, but you might need to use more advanced optimization techniques to find the optimal portfolio weights. Here's an outline of the steps you can take:

1. **Collect Data:** Make sure you have the historical returns of the 20 assets in your dataset.
2. **Define the Objective Function:** You'll need to define an objective function that calculates the portfolio return and risk (standard deviation) given a set of portfolio weights.

3. **Optimization:** Use optimization techniques like the quadratic programming solver in R to find the weights that maximize return for a given level of risk (standard deviation) while satisfying the constraint that the sum of weights equals 1.
4. **Generate a Range of Target Returns:** To plot the efficient frontier, you'll need to generate a range of target returns and find the portfolio with the minimum risk for each target return.

Here's a high-level code template to help you get started. This example uses the `quadprog` package for quadratic programming:

```

1 # Load necessary libraries
2 library(quadprog)
3 library(ggplot2)
4
5 # Sample returns for 20 assets (replace with your data)
6 # Ensure you have a matrix of returns with 20 columns
7 returns_matrix <- read.csv("your_returns_data.csv") # Replace with your
  data
8
9 # Calculate the mean returns and covariance matrix
10 mean_returns <- colMeans(returns_matrix)
11 cov_matrix <- cov(returns_matrix)
12
13 # Define the objective function to minimize risk (standard deviation)
14 portfolio.objective <- function(weights, cov_matrix) {
15   portfolio_return <- sum(weights * mean_returns)
16   portfolio_volatility <- sqrt(t(weights) %*% cov_matrix %*% weights)
17   return(portfolio_volatility)
18 }
19
20 # Set up constraints (sum of weights = 1)
21 A_eq <- matrix(1, 1, ncol(returns_matrix))
22 b_eq <- 1
23
24 # Generate a range of target returns
25 target_returns <- seq(min(mean_returns), max(mean_returns), length.out =
  100)
26
27 # Initialize vectors to store results
28 portfolio_returns <- numeric(length(target_returns))
29 portfolio_volatilities <- numeric(length(target_returns))
30
31 # Optimize for each target return
32 for (i in 1:length(target_returns)) {
33   target_return <- target_returns[i]
34
35   # Set up the quadratic programming problem
36   Dmat <- cov_matrix

```

```

37  dvec <- rep(0, ncol(returns_matrix))
38  Amat <- matrix(c(mean_returns, 1), nrow = 1)
39  bvec <- target_return
40
41  # Solve the optimization problem
42  weights <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)$solution
43
44  # Calculate portfolio risk (volatility)
45  portfolio_volatility <- portfolio.objective(weights, cov_matrix)
46
47  # Store results
48  portfolio_returns[i] <- target_return
49  portfolio_volatilities[i] <- portfolio_volatility
50 }
51
52 # Create a data frame for efficient frontier points
53 efficient_frontier <- data.frame(Return = portfolio_returns, Volatility =
   portfolio_volatilities)
54
55 # Plot the efficient frontier
56 ggplot(efficient_frontier, aes(x = Volatility, y = Return)) +
57   geom_line() +
58   labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
59   ggtitle("Efficient Frontier") +
60   theme_minimal()

```

Chat GPT

In this code, we use a loop to iterate over a range of target returns, solving the quadratic programming problem for each target return to find the corresponding portfolio weights and risk. Finally, we plot the efficient frontier. Make sure to replace “your_returns_data.csv” with the actual file path to your returns data.

Unfortunately, the code provided by Chat GPT has some issues. First, the A_eq and the B_eq are defined but they are never used in the code. Second, the matrices that define the optimization problem and its constraints to run the optimizer are not correctly defined. In fact, some of them are transposed and only one constraint is imposed rather than two (the sum of weight being equal to one and the return being equal to the target). You would need to check the working of the solve.QP function in r by digiting ?solve.QP in the R console and some iteration with Chat GPT to get to the following version of the code that we put at work on the data set of US stocks defined earlier in the chapter.

```

1  rm(list=ls())
2  #Check the working directory before importing else provide full path
3  #setwd(path)
4  setwd(dirname(rstudioapi::getActiveDocumentContext())$path))
5
6  # packages used
7  listofpackages <- c("dygraphs",

```

```

      "tidyverse", "ellipse", "reshape2", "highcharter", "xts", "xlsx", "readxl", "quantmod", "quadprog"
8
9 for (j in listofpackages){
10   if(sum(installed.packages()[, 1] == j) == 0) {
11     install.packages(j)
12   }
13   library(j, character.only = T)
14 }
15
16 raw_data          = read_xlsx("../data/2023_monthly_stocks.xlsx")
17 names(raw_data)[1] = 'Date'
18 typeof(raw_data)
19 typeof(raw_data$Date)
20 typeof(raw_data$AXP)
21 typeof(raw_data$CSCO)
22
23 dates <- seq(as.Date("1985-02-01"), length=462, by="months")
24 params <- c("Date", "AXP", "AMGN", "AAPL", "BA", "CAT", "CSCO", "CVX", "GS",
25            "HD", "HON", "IBM", "INTC", "JNJ", "KO", "JPM")
26 data <- raw_data[, c(params)]
27 data <- na.omit(data)
28 data <- data %>%
29   mutate(Date = as.Date(Date, format = "%Y-%m-%d"))
30
31 t1 <- nrow(data)
32 series_names <- c("AXP", "AMGN", "AAPL", "BA", "CAT", "CSCO", "CVX", "GS",
33                 "HD", "HON", "IBM", "INTC", "JNJ", "KO", "JPM")
34
35 for (name in series_names) {
36   return_col_name <- paste0(name, "_ret")
37   data[, return_col_name] <- array(data = NA, dim = t1)
38   for (i in 2:nrow(data)) {
39     data[i, return_col_name][[1]] <- (data[i, name][[1]] - data[i - 1,
40     name][[1]]) / data[i - 1, name][[1]]
41   }
42 }
43
44 params1 <-
45   c("AXP_ret", "AMGN_ret", "AAPL_ret", "BA_ret", "CAT_ret", "CSCO_ret", "CVX_ret", "GS_ret",
46     "HD_ret", "HON_ret", "IBM_ret", "INTC_ret", "JNJ_ret", "KO_ret", "JPM_ret")
47
48 returns_data <- data[, c(params1)]
49 returns_data <- na.omit(returns_data)
50
51 returns_matrix <- as.matrix(returns_data)
52
53 # Calculate the mean returns and covariance matrix
54 mean_returns <- colMeans(returns_matrix)
55 cov_matrix <- cov(returns_matrix)
56
57 # Define the objective function to minimize risk (standard deviation)
58 portfolio.objective <- function(weights, cov_matrix) {
59   portfolio_return <- sum(weights * mean_returns)

```

```

57   portfolio_volatility <- sqrt(t(weights) %*% cov_matrix %*% weights)
58   return(portfolio_volatility)
59 }
60
61 # Set up constraints (sum of weights = 1)
62 $A_eq <- matrix(1, nrow = 1, ncol = ncol(returns_matrix))
63 $b_eq <- matrix(1, nrow = 1)
64
65 # Generate a range of target returns
66 target_returns <- seq(min(mean_returns), max(mean_returns), length.out =
67   1000)
68
69 # Initialize vectors to store results
70 portfolio_returns <- numeric(length(target_returns))
71 portfolio_volatilities <- numeric(length(target_returns))
72
73 # Optimize for each target return
74 for (i in 1:length(target_returns)) {
75   target_return <- target_returns[i]
76
77   # Set up the quadratic programming problem
78   Dmat <- 2*cov_matrix
79   dvec <- matrix(rep(0, ncol(returns_matrix)),ncol=1)
80   a1mat <- matrix(mean_returns, nrow =ncol(returns_matrix))
81   a2mat <- matrix(rep(1, ncol(returns_matrix)), nrow =ncol(returns_matrix))
82   Amat <- cbind(a1mat, a2mat)
83   bvec <- matrix(c(target_return, 1),ncol=1)
84
85   # Solve the optimization problem
86   weights <- solve.QP(Dmat, dvec, Amat, bvec, meq = 2)$solution
87
88   # Calculate portfolio risk (volatility)
89   portfolio_volatility <- portfolio.objective(weights, cov_matrix)
90
91   # Store results
92   portfolio_returns[i] <- target_return
93   portfolio_volatilities[i] <- portfolio_volatility
94 }
95
96 # Create a data frame for efficient frontier points
97 efficient_frontier <- data.frame(Return = portfolio_returns, Volatility =
98   portfolio_volatilities)
99
100 # Plot the efficient frontier
101 ggplot(efficient_frontier, aes(x = Volatility, y = Return)) +
102   geom_line() +
103   labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
104   ggtitle("Efficient Frontier") +
105   theme_minimal()

```

2.5 Appendix: The Data

All empirical applications will be based on publicly available databases of US data observed at monthly (and therefore lower) frequency. They have been downloaded from Robert Shiller's webpage

(<http://www.econ.yale.edu/~shiller/>)

and Ken French's webpage

(http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html) and directly from yahoo finance.

The time series made available by Robert Shiller are saved in the successive columns of the EXCELworksheet DATA in the file **IE_DATA.XLS**

The time-series in the IE_DATA.XLS files	
identifier	description
P	S&P composite index
D	S&P dividend (at annual rate)
E	S&P earnings
CPI	US consumer price index
GS10	YTM of 10-year US Treasuries
CAPE	cyclically adjusted PE ratio

As described in the section "Online Data" of the webpage these stock market data are those used in the book, *Irrational Exuberance* [Princeton University Press 2000, Broadway Books 2001, 2nd ed., 2005] and cover the period 1871-Present. This data set consists of monthly stock price, dividends, and earnings data and the consumer price index (to allow conversion to real values), all starting January 1871. The price, dividend, and earnings series are from the same sources as described in Chapter 26 of the book *Market Volatility* [Cambridge, MA: MIT Press, 1989], although they are observed at monthly, rather than annual frequencies. Monthly dividend and earnings data are computed from the S&P four-quarter totals for the quarter since 1926, with linear interpolation to monthly figures. Dividend and earnings data before 1926 are from Cowles and associates (*Common Stock Indexes*, 2nd ed. [Bloomington, Ind.: Principia Press, 1939]), interpolated from annual data. The CPI-U (Consumer Price Index-All Urban Consumers) published by the U.S. Bureau of Labor Statistics begins in 1913; for years before 1913 it is spliced to the CPI Warren and Pearson's price index, by multiplying it by the ratio of the indexes in January 1913. December 1999 and January 2000 values for the CPI-U are extrapolated. See George F. Warren and Frank A. Pearson, *Gold and Prices* (New York: John Wiley and Sons, 1935). Data are from their Table 1, pp. 11-14.

The time series made available by Ken French are saved in the successive columns of the EXCELworksheet DATA in the file **FF_DATA.XLS**.

The time-series in the FF_Data.xls files	
identifier	description
EXRET_MKT	MKT excess ret
SMB	returns on SMB
HML	returns on HML
RF	returns on the risk-free asset
MOM	returns on MOM
RMW	returns on RMW
CMA	returns on CMA
PR(i,j)	returns on 25 FF portolios ($i=1,\dots,5,j=1,\dots,5$)

The construction of the Fama French factors is described at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_5_factors_2x3.html, while the construction of the FF portfolios is described at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/tw_5_ports.html. Finally, data on the components of the DJ30 and the index have been downloaded from yahoo.finance using the quantmod package in R.

Chapter 3

The Modelling Process at Work: the CER model

3.1 Introduction

In this chapter, we shall consider a very basic model for returns and illustrate how model specification, estimation and simulation can be applied to find optimal portfolio weights, measure the risk of a portfolio and backtest the portfolio performance

3.2 Model Specification: the Constant Expected Return Model

Our objective is the specification of a statistical model for asset prices and returns. To this end, consider the (naive) log random walk (LRW) hypothesis on the evolution of prices states that prices evolve approximately according to the stochastic difference equation:

$$\ln P_t = \mu + \ln P_{t-1} + \epsilon_t$$

where the 'innovations' ϵ_t are assumed to be uncorrelated across time ($cov(\epsilon_t; \epsilon_{t'}) = 0 \quad \forall t \neq t'$), with constant expected value 0 and constant variance σ^2 . Sometimes, a further hypothesis is added and the ϵ_t are assumed to be jointly normally distributed. In this case, the assumption of non correlation becomes equivalent to the assumption of independence.

Since $\ln P_t - \ln P_{t-1} = r_{t-1;t}^*$ the LRW is obviously equivalent to the assumption that log returns are uncorrelated random variables with constant expected value and variance.

A linear random walk in prices was sometimes considered in the earliest times of

quantitative financial research, but it does not seem a good model for prices since a sequence of negative innovations may result in negative prices. Moreover, while the hypothesis of constant variance for (log) returns may be a good first-order approximation of what we observe in markets, the same hypothesis for prices is not empirically sound: in general price changes tend to have a variance which is an increasing function of the price level.

If we take prices as inclusive of dividends, then we can write the following model for log-returns

$$\begin{aligned} r_{t,t+1} &= \mu + \sigma \epsilon_t \\ \epsilon_t &\sim i.i.d.(0, 1) \end{aligned}$$

This simple specification has some appealing properties for the n period returns $r_{t,t+n}$:

If we assume the LRW and consider a sequence of n log returns r_t^* at times $t, t-1, t-2, \dots, t-n+1$ (just for the sake of simplicity in notation we suppose each time interval Δ to be of length 1 and drop the generic Δ) we have the following:

$$E(r_{t,t+n}) = E\left(\sum_{i=1}^n r_{t+i,t+i-1}\right) = \sum_{i=1}^n E(r_{t+i,t+i-1}) = n\mu$$

$$Var(r_{t,t+n}) = Var\left(\sum_{i=1}^n r_{t+i,t+i-1}\right) = \sum_{i=1}^n Var(r_{t+i,t+i-1}) = n\sigma^2$$

This obvious result, which is a direct consequence of the assumption of constant expected value and variance and of non-correlation of innovations at different times is typically applied, for annualization purposes, also when the LRW is not considered to be valid.

So, for instance, given an evaluation of σ^2 on monthly data, this evaluation is annualized by multiplying it by 12

This is not a convention, but the correct procedure, if the LRW model holds. In this case, in fact, the variance over n time periods is equal to n times the variance over one time period. If the LRW model is not believed to hold, for instance, if the expected value and-or the variance of return are not constant over time or if we have correlation among the ϵ_t , this procedure becomes just as a convention.¹

¹Empirical computation of variances over different time intervals typically results in sequences which increase less than linearly wrt the increase of the time interval between consecutive observations. This could be interpreted as the existence of (small) on average negative correlations between returns.

3.2.1 Stocks for the long run

The fact that, under the LRW, the expected value grows linearly with the length of the time period while the standard deviation (square root of the variance) grows with the square root of the number of observations, has created a lot of discussion about the existence of some time horizon beyond which it is always proper to hold a stock portfolio. This problem, conventionally called 'time diversification', and more popularly 'stocks for the long run', has attracted some considerable attention.

We have three flavors of the "stocks for the long run" argument. The first and the second are a priori arguments depending on the log random walk hypothesis or something equivalent to it, the third is an a posteriori argument based on historical data.

The basic idea of the first version of the argument can be sketched as follows. Assume that single period (log) returns have (positive) expected value μ and variance σ^2 . Moreover, assume for simplicity that the investor requires a Sharpe ratio of say S . Under the above hypotheses, plus the log random walk hypothesis, the Sharpe ratio over n time periods is given by

$$S = \frac{n\mu}{\sqrt{n}\sigma} = \sqrt{n}\frac{\mu}{\sigma}$$

so that, if n is large enough, any required value can be reached. Another way of phrasing the same argument, when we add the hypothesis of normality on returns, is that, for any given probability α and any given required return C there is always an horizon for which the probability for n period return less than C is less than α .

$$\Pr(R^p < C) = \alpha.$$

$$\begin{aligned} \Pr(R^p < C) = \alpha &\iff \Pr\left(\frac{R^p - n\mu}{\sqrt{n}\sigma} < \frac{C - n\mu}{\sqrt{n}\sigma}\right) = \alpha \\ &\iff \Phi\left(\frac{C - n\mu}{\sqrt{n}\sigma}\right) = \alpha, \\ C &= n\mu + \Phi^{-1}(\alpha)\sqrt{n}\sigma \end{aligned}$$

But $n\mu + \Phi^{-1}(\alpha)\sqrt{n}\sigma$, for $\sqrt{n} > \frac{1}{2}\frac{\Phi^{-1}(\alpha)}{\mu}\sigma$ is an increasing function in n so that for any α and any chosen value C , there exists a n such that from that n onward, the probability for an n period return less than C is less than α .

The investment implication could be that for a time horizon of an undetermined number n of years, the investment that has the highest expected return per unit of standard deviation is optimal even if the standard deviation is very high. This

investment can be very risky in the "short run", but there is always a time horizon for which, the probability of any given loss is as small as you like or, that is the same, the Sharpe ratio as big as you like. Typically, such high return (and high volatility) investment are stocks, so: "stocks for the long run".

Note, however, that the value of n for which this lower bound crosses a given C level is the solution of

$$n\mu + \Phi^{-1}(\alpha) \sqrt{n}\sigma \geq C$$

In particular, for $C = 0$ the solution is

$$\sqrt{n} \geq -\frac{\Phi^{-1}(\alpha) \sigma}{\mu}$$

Consider now the case of a stock with σ/μ ratio for one year is of the order of 6. Even allowing for a large α , say 0.25, so that $\Phi^{-1}(\alpha)$ is near minus one, the required n shall be in the range of 36 which is only slightly shorter than the average working life.

As a matter of fact, based on the analysis of historical prices and risk adjusted returns, stocks have been almost always a good long-run investment. However, some care must be exercised in interpreting this evidence because history is what we have observed and one could doubt the possibility of an institution such as the stock market to survive without providing a sustainable impression of offering some opportunities. Unfortunately, the arrow of time is uni-directional and experimental data for financial time-series are not available.

3.3 Model Estimation

Model specification has led us to the following description for the vector of one-period returns on assets used to build a portfolio:

$$\begin{aligned} \mathbf{r}_{t,t+1} &= \boldsymbol{\mu} + \mathbf{H}\epsilon_{t+1} \\ \boldsymbol{\Sigma} &= \mathbf{H}\mathbf{H}' \\ \epsilon_{t+k} &\sim \mathcal{D}(\mathbf{0}, \mathbf{I}) \end{aligned}$$

where $\mathbf{r}_{t,t+k}$ is the vector of returns between time t and time $t+k$ in which we are interested, $\boldsymbol{\mu}$ is the vector of mean returns and the matrix \mathbf{H} determines the time invariance variance-covariance matrix of returns.

Model estimation allows to find values for $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$. In the case of CER this step is easily solved by n OLS regressions of the n returns on a constant.

$$\hat{\mu}^i = \frac{1}{T} \sum_{t=1}^T r_{t,t+1}^i$$

$$\hat{\sigma}_{ii} = \frac{1}{T} \sum_{t=1}^T \left(r_{t,t+1}^i - \hat{\mu}^i \right)^2$$

$$\hat{\sigma}_{ij} = \frac{1}{T-1} \sum_{t=1}^T \left(r_{t,t+1}^i - \hat{\mu}^i \right) \left(r_{t,t+1}^j - \hat{\mu}^j \right)$$

3.3.1 Parameters Estimation in a linear model

The CER is a special case of a linear model, consider the following general representation of a linear model :

$$\mathbf{y} = \mathbf{X}\beta + \epsilon,$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdot & \cdot & x_{1k} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{N1} & x_{N2} & \cdot & \cdot & x_{Nk} \end{pmatrix},$$

$$\beta = \begin{pmatrix} \beta_1 \\ \cdot \\ \cdot \\ \cdot \\ \beta_k \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_N \end{pmatrix}.$$

The simplest way to derive estimates of the parameters of interest is the ordinary least squares (OLS) method. Such a method chooses values for the unknown parameters to minimize the magnitude of the non-observable components. The best fit is obtained by minimizing the sum of squared vertical deviations of the data points from the fitted line.

Define the following quantity:

$$\mathbf{e}(\beta) = \mathbf{y} - \mathbf{X}\beta,$$

where $\mathbf{e}(\beta)$ is a $(n \times 1)$ vector. If we treat $\mathbf{X}\beta$, as a (conditional) prediction for \mathbf{y} , then we can consider $\mathbf{e}(\beta)$ as a forecasting error. The sum of the squared errors is then

$$\mathbf{S}(\beta) = \mathbf{e}(\beta)' \mathbf{e}(\beta).$$

The OLS method produces an estimator of β , $\hat{\beta}$, defined as follows:

$$\mathbf{S}(\hat{\beta}) = \min_{\beta} \mathbf{e}(\beta)' \mathbf{e}(\beta).$$

Given $\hat{\beta}$, we can define an associated vector of residual $\hat{\epsilon}$ as $\hat{\epsilon} = \mathbf{y} - \mathbf{X}\hat{\beta}$. The OLS estimator is derived by considering the necessary and sufficient conditions for $\hat{\beta}$ to be a unique minimum for \mathbf{S} :

1. $\mathbf{X}'\hat{\epsilon} = 0$;
2. $\text{rank}(\mathbf{X}) = k$.

Condition 1 imposes orthogonality between the \mathbf{X} variables and the OLS residuals, it ensures that residuals have zero mean when a constant is included among the regressors. Condition 2 requires that the columns of the \mathbf{X} matrix are linearly independent.

From 1. we derive an expression for the OLS estimates:

$$\begin{aligned} \mathbf{X}'\hat{\epsilon} &= \mathbf{X}'(\mathbf{y} - \mathbf{X}\hat{\beta}) = \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\hat{\beta} = 0, \\ \hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}. \\ \hat{\sigma}^2 &= \frac{\hat{\epsilon}'\hat{\epsilon}}{T - k} \end{aligned}$$

OLS in the CER

In the CER we have:

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\beta + \epsilon, \\ \mathbf{y} &= \begin{pmatrix} r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_T \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}, \\ \beta &= \mu, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \epsilon_T \end{pmatrix}. \end{aligned}$$

From one-period to multi-period returns in the CER

Notice that once one-step ahead returns are known, then also n-step ahead returns are known:

$$\begin{aligned} E_t(\mathbf{r}_{t,t+n}) &= n\hat{\mu} \\ \text{Var}(\mathbf{r}_{t,t+n}) &= n\hat{\Sigma} \end{aligned}$$

As a consequence of these properties of the data, weights in an optimal multi-horizon portfolio coincide with weights in a single-period horizon portfolio:

$$\begin{aligned} \hat{\mathbf{w}}^T &= \frac{\Sigma^{-1}(\mu - r^f \mathbf{e})}{\mathbf{e}'\Sigma^{-1}(\mu - r^f \mathbf{e})}, \\ &= \frac{\Sigma^{-1}(nn^{-1})(\mu - r^f \mathbf{e})}{\mathbf{e}'\Sigma^{-1}(nn^{-1})(\mu - r^f \mathbf{e})} \end{aligned}$$

3.4 Model Simulation: Monte-Carlo and Bootstrap Methods

Once parameters in the CER have been estimated the model can be simulated to derive the distribution of asset returns in the future, this is done by simulating pseudo data from the model. Model can be simply used to create the distribution of returns in the future and derive Value-at-Risk measures, but they can also be evaluated via the following procedure:

- split the sample into two parts, a training sample and a test sample.
- Use the training sample to estimate model parameters'.
- Use the model to simulate artificial observation for the test sample.
- Evaluate the model by comparing actual data in the test sample with model-simulated data over the same period.

We shall consider two ways of simulating pseudo-data: Monte-Carlo Simulation and Bootstrap. To use Monte-Carlo Simulation to generate pseudo data from the CER model, some estimates of μ σ are necessary. Given these estimates an assumption must be made on the distribution of ϵ_t . Then an artificial sample for ϵ_t of the length matching that of the available can be computer simulated. The simulated residuals are then mapped into simulated returns via μ , σ . This exercise can be replicated

N times (and therefore a Monte-Carlo simulation generates a matrix of computer-simulated returns whose dimensions are defined by the sample size T and by the number of replications N). The distribution of model-predicted returns can be then constructed and one can ask if the observed data can be considered as one draw from this distribution.

One of the possible limitations of the Monte-Carlo approach is the choice of a distribution from which the residuals are to be drawn. It might be very well the case that the model goes wrong because the choice of the statistical distribution is not the correct one. Bootstrap methods overcome this problem by sampling residuals from their empirical distribution. All the steps in a bootstrap simulation are the same as the Monte-Carlo simulation except that different observations for residuals are constructed by taking the deviation of returns from their sample mean putting them in an urn and resampling from the urn with replacement.

3.5 The CER model at work with R

In this section, we shall illustrate codes in R that apply model specification, estimation and simulation to the CER model to perform Optimal asset allocation and backtesting.

3.5.1 Asset Allocation with the CER

The following code runs after the usual preliminaries (setting working directory, upload relevant packages) uses the inbuilt database BERNDINVEST in the package Ecofin to perform optimal asset allocation adopting the CER model for US stocks.

First, Data transformation is applied via a loop to construct, from monthly returns monthly prices, i.e. the value over-time of a buy and hold portfolio in each stock, and monthly log-prices.

Second, descriptive graphical analysis is implemented using the facilities in the package ggplot.

Third, the relevant parameters in the CER are estimated and optimal asset allocation is found by computing weights for the tangency portfolio.

Lastly, the utilization of the package fPortfolio in R is described. [Research \(2023\)](#) is an excellent online guide to Fportfolio. The program illustrates how to get the data in the appropriate format, set constraints for the portfolio optimization, compute efficient frontiers and optimal portfolio weights and provide graphic illustration of the results.

¹ # Asset Allocation with CER
² # elaboration on the original code produced by E.Zivot by C. Favero

```

3 # author: Carlo Favero
4 # created: August, 2023
5 # comments: Original Examples are taken from chapter 11 in Zivot and Wang
   (2006)
6
7 rm(list=ls()) #Removes all items in Environment!
8 #setwd(path)
9 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
10
11 # set output options
12 options(width = 70, digits=4)
13
14 #install.packages("fEcofin", repos="http://R-Forge.R-project.org")
15 library(fEcofin)
16 # load required packages
17 listofpackages <- c("ellipse", "dygraphs", "ggplot2")
18
19 for (j in listofpackages){
20   if(sum(installed.packages()[, 1] == j) == 0) {
21     install.packages(j)
22   }
23   library(j, character.only = T)
24 }
25
26 install.packages(c("cluster", "mvoutlier", "pastecs", "fPortfolio"),
27 repos="http://cran.r-project.org")
28 # load required packages
29 library(cluster)
30 library(mvoutlier)
31 library(pastecs)
32 library(fPortfolio)
33
34 #####
35 # Data Loadings and Transform: Descriptive Analysis
36 #####
37
38 # create data frame with dates as rownames
39 berndt.df = berndtInvest[, -1]
40 berndt.df$date <- as.Date(berndtInvest[, 1])
41 rownames(berndt.df) = as.character(berndtInvest[, 1])
42 colnames(berndt.df)
43 dimnames(berndt.df)[[2]] #command alternative to the previous one
44
45 # transform the data and compute cumulative returns
46
47 t0 <- which(berndt.df$date == "1978-01-01")
48 t1 <- which(berndt.df$date == "1987-12-01")
49
50 series_names <-
   c("CITCRP", "CONED", "CONTIL", "DATGEN", "DEC", "DELTA", "GENMIL", "GERBER", "IBM",
51 "MARKET", "MOBIL", "PANAM", "PSNH", "TANDY", "TEXACO", "WEYER", "RKFREE")
52

```

70 CHAPTER 3. THE MODELLING PROCESS AT WORK: THE CER MODEL

```

53 for (name in series_names) {
54   P_col_name <- paste0(name, "_P")
55   LP_col_name <- paste0("L", P_col_name)
56   berndt.df[t0, P_col_name] <- 1
57   for (i in (t0+1):(t1)) {
58     berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] *
        (1+berndt.df[i, name][[1]] )
59   }
60   berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
61 }
62 # add a trend to the database
63 berndt.df$TREND <- array(data = NA, dim = nrow(berndt.df))
64 berndt.df[t0, c("TREND")] <- 1 # don't need to repeat the value to make the
    array being assigned be of the same length. be careful though as it is
    one of the few cases of exception
65
66 #####
67 # Descriptive Analysis
68 #####
69
70 #We can now plot, please note the difference with plotting from a
    time-series object
71
72 plot(berndt.df$date[t0:t1], berndt.df$TEXACO[t0:t1], ylab =
    "TEXACO", xlab="year", main = "Monthly Returns", col = "blue", lwd =
    2, type="l")
73
74
75 plot(berndt.df$date[t0:t1], berndt.df$TEXACO[t0:t1], col = 'blue', type =
    "l",
76       ylab = "returns TEXACO and MKT", xlab = "date", lwd = 2)
77 lines(y = rep(mean(berndt.df$TEXACO[t0:t1], na.rm = T),
    length(berndt.df$date[t0:t1])), x = berndt.df$date[t0:t1], col =
    "red")
78 lines(y = berndt.df$MARKET[t0:t1], x = berndt.df$date[t0:t1], col =
    "green", lwd = 2)
79 legend("topleft", legend = c("TEXACO", "MKT"),
80       col = c("blue", "green"), lty = 1)
81
82 plot(berndt.df$date[t0:t1], berndt.df$LTEXACO_P[t0:t1], col = 'blue', type =
    "l",
83       ylab = "portfolios TEXACO and MKT", xlab = "date", ylim = c(-0.5,
    2), lwd = 2)
84 lines(y = berndt.df$LMARKET_P[t0:t1], x = berndt.df$date[t0:t1], col =
    "green", lwd = 2)
85 legend("topleft", legend = c("TEXACO", "MKT"),
86       col = c("blue", "green"), lty = 1)
87
88 # Create the plot using ggplot, as generated by Chat GPT
89 ggplot(berndt.df, aes(x = date)) +
90   geom_line(aes(y = LTEXACO_P), color = "blue", size = 2, linetype =
    "solid") +

```

```

91 geom_line(aes(y = LMARKET_P), color = "green", size = 2, linetype =
    "solid") +
92 labs(x = "Date", y = "Portfolios TEXACO and MKT") +
93 ylim(-0.5, 2) +
94 theme_minimal() +
95 theme(
96   legend.position = "topleft",
97   legend.title = element_blank(),
98   legend.text = element_text(size = 12),
99   axis.text = element_text(size = 12),
100  axis.title = element_text(size = 14),
101  plot.title = element_text(size = 16, hjust = 0.5)
102 ) +
103 scale_color_manual(
104   values = c("blue", "green"),
105   guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
106 ) +
107 guides(fill = guide_legend(override.aes = list(size = 2)))
108
109 #####
110 # Asset Allocation with CER
111 #####
112 returns.df=berndt.df[, c(1:9,11:16)]
113 #returns.df = berndt.df[, c(-10, -17)]
114 exreturns.df=returns.df-berndt.df$RKFREE
115 returns.mat = as.matrix(exreturns.df)
116 n.obs = nrow(returns.mat)
117
118 #Estimation of CER model parameters
119 cov.sample=var(returns.mat)
120 mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
121
122 #
123 # compute tangency portfolio
124 #
125
126 e = matrix(1, nrow = nrow(cov.sample), ncol = 1) # unitary column vector e
127 w.tan.sample =
    (solve(cov.sample)%*(mu))/as.numeric(t(e)%*(solve(cov.sample)%*(mu)))
128
129 colnames(w.tan.sample) = "sample"
130 barplot(t(w.tan.sample), horiz=F, main="Weights", col="blue", cex.names =
    0.75, las=2)
131
132
133 #####
134 # Using the fportfolio package
135 #####
136
137 #returns.df=berndt.df[, c(1:9,11:16)]
138 #exreturns.df=returns.df-berndt.df$RKFREE
139 companies <- colnames(exreturns.df)

```

72 CHAPTER 3. THE MODELLING PROCESS AT WORK: THE CER MODEL

```

140 #ts
141 tsdata <- ts(exreturns.df, start = c(1978, 1), frequency = 12, names =
      companies)
142 s1 <- window(tsdata[, "TEXACO"], start = c(1978, 1), end = c(1987, 12))
143 dygraph(s1, ylab = "TEXACO", main = "monthly excess returns")
144 data01ts <- as.timeSeries(tsdata)
145 # financial data description
146 ddown<-drawdowns(data01ts)
147 ddowndata <- ts(ddown, start = c(1978, 1), frequency = 12, names =
      companies)
148 s1 <- window(ddowndata[, "TEXACO"], start = c(1978, 1), end = c(1987, 12))
149 dygraph(s1, ylab = "TEXACO", main = "drawdowns")
150 drawdownsStats(data01ts[, "TEXACO"])
151 #-----
152 # Portfolio Allocation
153 #-----
154
155 # Step 1 define the data in our case 15 excess returns data in data01ts
156 showClass("fPFOLIODATA")
157
158 lppData <- portfolioData(data = data01ts, spec = portfolioSpec())
159 # once the data have been defined we can get info on them
160 str(lppData, width = 65, strict.width = "cut")
161 print(lppData)
162 getData(portfolioData(lppData))[-1]
163 getStatistics(portfolioData(lppData))
164
165 # Step 2 Set Portfolio Constraints
166
167 showClass("fPFOLIOCON")
168 #default constraints: long-only
169 Data<-data01ts
170 Spec <- portfolioSpec()
171 setTargetReturn(Spec) <- mean(Data)
172 Constraints <- "LongOnly"
173 defaultConstraints <- portfolioConstraints(Data, Spec, Constraints)
174 str(defaultConstraints, width = 65, strict.width = "cut")
175 print(defaultConstraints)
176
177 # short constraints
178 shortConstraints <- "Short"
179 portfolioConstraints(Data, Spec, shortConstraints)
180
181 # box constraints
182 box.1 <- "minW[1:15] = 0.1"
183 box.2 <- "maxW[1:15] = 1" # you can have more boxes before combining them
184 boxConstraints <- c(box.1, box.2)
185 boxConstraints
186 portfolioConstraints(Data, Spec, boxConstraints)
187
188
189 # Step 3 Computing Optimal Portfolios

```

```

190
191 #3.0 A benchmark: equal weight portfolio
192 ewSpec <- portfolioSpec()
193 nAssets <- ncol(data01ts)
194 setWeights(ewSpec) <- rep(1/nAssets, times = nAssets)
195 ewPortfolio <- feasiblePortfolio(
196   data = data01ts,
197   spec = ewSpec,
198   constraints = "LongOnly")
199 print(ewPortfolio)
200
201 # Efficient Frontier plot
202 setNFrontierPoints(ewSpec) <- 25
203 eff_ew_frontier <- portfolioFrontier(data = data01ts, spec = ewSpec,
204   constraints = "LongOnly")
205 tailoredFrontierPlot(object = eff_ew_frontier)
206
207 #3.1 Long-Only
208 tgSpec <- portfolioSpec()
209 setRiskFreeRate(tgSpec) <- 0
210 constraints <- "longOnly"
211 tgPortfolio <- tangencyPortfolio(
212   data = data01ts,
213   spec = tgSpec, constraints = constraints)
214 print(tgPortfolio)
215
216 #printing the results
217 col <- seqPalette(ncol(data01ts), "BuPu")
218 weightsPie(tgPortfolio, box = FALSE, col = col)
219 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
220   font = 2, cex = 0.7, adj = 0)
221 weightedReturnsPie(tgPortfolio, box = FALSE, col = col)
222 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
223   font = 2, cex = 0.7, adj = 0)
224 covRiskBudgetsPie(tgPortfolio, box = FALSE, col = col)
225 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
226   font = 2, cex = 0.7, adj = 0)
227
228 efficient_frontier <- portfolioFrontier(data = data01ts, spec = tgSpec,
229   constraints = constraints)
230 print(efficient_frontier)
231 # Efficient Frontier plot
232 setNFrontierPoints(tgSpec) <- 25
233 efficient_frontier <- portfolioFrontier(data = data01ts, spec = tgSpec,
234   constraints = constraints)
235 tailoredFrontierPlot(object = efficient_frontier)
236
237 #-----
238 #3.2 Box-Constraints
239 #-----
240 boxSpec <- portfolioSpec()
241 setRiskFreeRate(boxSpec) <- 0

```

```

239 boxConstraints <- c("minW[1:15]=0.05", "maxW[1:15]=0.5")
240 tgPortfolio1 <- tangencyPortfolio(
241   data = data01ts,
242   spec = boxSpec, constraints = boxConstraints)
243 print(tgPortfolio1)
244
245 #printing the results
246 col <- seqPalette(ncol(data01ts), "BuPu")
247 weightsPie(tgPortfolio1, box = FALSE, col = col)
248 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
249       font = 2, cex = 0.7, adj = 0)
250 weightedReturnsPie(tgPortfolio, box = FALSE, col = col)
251 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
252       font = 2, cex = 0.7, adj = 0)
253 covRiskBudgetsPie(tgPortfolio, box = FALSE, col = col)
254 mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
255       font = 2, cex = 0.7, adj = 0)

```

3.5.2 Model Simulation with the CER: backtesting and VaR

To illustrate model simulation we consider the application to backtesting and Value at Risk. The following illustrative code starts by building the tangency and the minimum variance portfolio on the stocks considered in the previous section and by implementing within-sample evaluation of the properties of the optimized portfolios. This type of exercise suffers from the well-known problem of "look-ahead bias" as data not available in real time have been used to construct weights. True backtesting, when the available data are divided into a "training Sample" and a test sample, is then implemented using the package `fPortfolio`. A procedure is used according to which a rolling sample is used to build the allocation that reflects the information available in real-time at the end of the sample, allocations are then evaluated out-of-sample and then re-optimized. The possibility of smoothing optimal weights in the rolling procedure is also considered. Backtesting is then conducted by assessing ex-post the performance of each allocation. Finally, in the last part of the code, the CER model is applied to the Tangency portfolio to simulate, via bootstrap and Monte-Carlo procedures, the distribution of the returns and to produce one-month ahead Value-at-Risk.

```

1 # Asset Allocation with CER
2 # elaboration on the original code produced by E.Zivot by C. Favero
3 # author: Carlo Favero
4 # created: August, 2023
5 #
6 # comments: Original Examples are taken from chapter 11 in Zivot and Wang
7             (2006)
8 rm(list=ls()) #Removes all items in Environment!

```

```

 9 #setwd(path)
10 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
11
12 #install.packages("fEcofin", repos="http://R-Forge.R-project.org")
13 library(fEcofin)
14 # load required packages
15 listofpackages <- c("ellipse", "dygraphs", "ggplot2", "reshape2")
16
17 for (j in listofpackages){
18   if(sum(installed.packages()[, 1] == j) == 0) {
19     install.packages(j)
20   }
21   library(j, character.only = T)
22 }
23
24 install.packages(c("cluster", "mvoutlier", "pastecs", "fPortfolio"), repos="http://cran.r-project.org")
25 # load required packages
26 library(cluster)
27 library(mvoutlier)
28 library(pastecs)
29 library(fPortfolio)
30
31
32 # create data frame with dates as rownames
33 berndt.df = berndtInvest[, -1]
34 rownames(berndt.df) = as.character(as.Date(berndtInvest[, 1]))
35
36
37 #####
38 # Derive the optimal portfolio weights (i.e. the weights in the tangency
39 # portfolio)
40 # using the CER for (i) the Minimum Variance Portfolio , (ii) the tangency
41 # portfolio.
42 #####
43 returns.df=berndt.df[, c(1:9,11:16)]
44 #returns.df = berndt.df[, c(-10, -17)]
45 exreturns.df=returns.df-berndt.df$RKFREE
46 returns.mat = as.matrix(exreturns.df)
47 # using ggplot to plot series in returns
48 berndt.df$date <- as.Date(row.names(berndt.df))
49
50 # Create the time series plot using ggplot
51 ggplot(data = berndt.df, aes(x = date, y = WEYER)) +
52   geom_line() + # Add a line plot
53   labs(x = "Date", y = "WEYER") # Label the axes
54
55 #
56 # compute global min variance portfolio
57 #
58 # use CER model: estimate the relevant unknown parameters with the sample
59 # covariances

```

```

58 returns.mat = as.matrix(exreturns.df)
59 n.obs = nrow(returns.mat)
60 cov.sample=var(returns.mat)
61 mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
62 e = matrix(1, nrow = nrow(cov.sample), ncol = 1) # unitary column vector e
63 #
64 # compute GMIN portfolio
65 #
66 w.gmin.sample = solve(var(returns.mat))%*%rep(1,nrow(cov.sample))
67 w.gmin.sample = w.gmin.sample/sum(w.gmin.sample)
68 berndt.df$GMIN<-returns.mat%*%w.gmin.sample
69
70 barplot(t(w.gmin.sample), horiz=F, main="Weights", col="blue", cex.names =
    0.75, las=2)
71
72 ggplot(data = berndt.df, aes(x = date, y = GMIN)) +
73   geom_line() + # Add a line plot
74   labs(x = "Date", y = "GMIN") # Label the axes
75
76 #
77 # compute tangency portfolio
78 #
79 w.tan.sample = (solve(cov.sample)%*%as.numeric(mu))
80 w.tan.sample =w.tan.sample/as.numeric(t(e)%*(solve(cov.sample)%*(mu)))
81
82 berndt.df$TAN<-returns.mat%*%w.tan.sample
83
84 # visualize the differences
85 par(mfrow=c(1,2))
86 barplot(t(w.tan.sample), horiz=T, main="Tangency Port CER", col="blue",
    cex.names = 0.75, las=1)
87 barplot(t(w.gmin.sample), horiz=T, main="Min Var Port CER", col="red",
    cex.names = 0.75, las=1)
88 par(mfrow=c(1,1))
89
90 plot <- ggplot(data= berndt.df, aes(x = date)) +
91   geom_line(aes(y = TAN, color = "TAN"), size = 1) +
92   geom_line(aes(y = GMIN, color = "GMIN"), size = 1) +
93   labs(title = "Returns",
94     x = "Time", y = "Monthly Returns") +
95   scale_color_manual(values = c("TAN" = "red", "GMIN" = "blue")) +
96   theme_minimal() +
97   theme(axis.line = element_line(color = "black"))
98
99 print(plot)
100
101
102 #####
103 # Graphs the value over-time of 1 dollar invested in 1978:1 until the end
    of the
104 # available sample in the two alternative tangency portfolios and in the
    market

```

```

105 #####
106 berndt.df$Port_mkt <- berndt.df$Port_TAN<- berndt.df$Port_GMIN <-
      array(data = NA, dim = nrow(berndt.df))
107
108 berndt.df[1, c("Port_mkt", "Port_TAN", "Port_GMIN")] <- 1
109 t1<-nrow(berndt.df)
110 for (i in 2:t1) {
111   berndt.df[i, "Port_mkt"][[1]]=berndt.df[i-1,
      "Port_mkt"][[1]]*(1+berndt.df[i, "MARKET"][[1]])
112   berndt.df[i, "Port_TAN"][[1]]=berndt.df[i-1,
      "Port_TAN"][[1]]*(1+berndt.df[i, "TAN"][[1]])
113   berndt.df[i, "Port_GMIN"][[1]]=berndt.df[i-1,
      "Port_GMIN"][[1]]*(1+berndt.df[i, "GMIN"][[1]])
114 }
115
116
117 # time series Plot of the three Portfolios
118
119 plot <- ggplot(data= berndt.df, aes(x = date)) +
120   geom_line(aes(y = Port_mkt, color = "Port_mkt"), size = 1) +
121   geom_line(aes(y = Port_GMIN, color = "Port_GMIN"), size = 1) +
122   geom_line(aes(y = Port_TAN, color = "Port_TAN"), size = 1) +
123   labs(title = "Returns",
124        x = "Time", y = "Monthly Returns") +
125   scale_color_manual(values = c("Port_mkt" = "red", "Port_GMIN" =
      "blue", "Port_TAN" = "green")) +
126   theme_minimal() +
127   theme(axis.line = element_line(color = "black"))
128
129
130 # compare means and sd values on global min variance portfolios
131
132 mu.gmin.sample = as.numeric(colMeans(berndt.df$GMIN))
133 mu.tan.sample = as.numeric(colMeans(berndt.df$TAN))
134 sd.gmin.sample = as.numeric(apply(berndt.df$GMIN,2,sd))
135 sd.tan.sample = as.numeric(apply(berndt.df$TAN,2,sd))
136 cbind(mu.tan.sample,mu.gmin.sample, sd.tan.sample, sd.gmin.sample)
137
138 ## -----
139 # BACKTESTING with fPortfolio
140 ## -----
141 companies <- colnames(berndt.df)
142 #getting the data in ts format
143 tsdata <- ts(berndt.df, start = c(1978, 1), frequency = 12, names =
      companies)
144 data01ts <- as.timeSeries(tsdata)
145 ddown<-drawdowns(data01ts)
146 ddowndata <- ts(ddown, start = c(1978, 1), frequency = 12, names =
      companies)
147 s1 <- window(ddowndata[, "TAN"], start = c(1978, 1), end = c(1987, 12))
148 dygraph(s1, ylab = "TAN", main = "drawdowns")
149 drawdownsStats(data01ts[, "TAN"])

```

```

150 ## -----
151 # out-of-sample BACKTESTING
152 ## -----
153
154 Data <- data01ts
155 Spec <- portfolioSpec()
156 Constraints <- "LongOnly"
157 Backtest <- portfolioBacktest()
158 setWindowsHorizon(Backtest) <- "60m"
159 equidistWindows(data = Data, backtest = Backtest)
160
161
162 #Specify assets for backtesting
163 #Formula <- MARKET ~ CITCRP + CONED + CONTIL + DATGEN + DEC + DELTA +
164 # + GENMIL + GERBER + IBM+MOBIL+PANAM+PSNH+TANDY+TEXACO+WEYER
165 Formula <- MARKET ~ CITCRP + CONED + CONTIL + DATGEN + DEC + DELTA +
166 GENMIL + GERBER + IBM + MOBIL + PANAM + PSNH + TANDY + TEXACO + WEYER
167
168 #Optimize rolling portfolios and run backtests
169 #btportfolios <- portfolioBacktesting(formula = Formula,
170 # +data = data01ts, spec = Spec,
171 # constraints = Constraints,
172 # + backtest = Backtest, trace = FALSE)
173
174 btportfolios <- portfolioBacktesting(formula = Formula,
175 data = data01ts, spec = Spec,
176 constraints = Constraints,
177 backtest = Backtest, trace = FALSE)
178
179 #Weights are rebalanced on a monthly basis
180 Weights <- round(100 * btportfolios$weights, 2)[1:60, ]
181 Weights
182
183 setSmootherLambda(btportfolios$backtest) <- "1m"
184 SmoothPortfolios <- portfolioSmoothing(object = btportfolios, trace = FALSE)
185 smoothWeights <- round(100 * SmoothPortfolios$smoothWeights, 2)[1:60, ]
186 smoothWeights
187
188 backtestPlot(SmoothPortfolios, cex = 0.6, font = 1, family = "mono")
189
190 netPerformance(SmoothPortfolios)
191
192 ## -----
193 # MODEL-SIMULATION
194 ## -----
195
196 ## -----
197 # model specification and estimation
198 mod_TAN <- lm(berndt.df$TAN ~ 1)
199 summary(mod_TAN)

```

```

200
201 ## -----
202 # parameter calibration and choice of the number of replications in the
      simulation and of the sample size for simulated data
203 vol <- sd(mod_TAN$residuals)
204 alpha <- mod_TAN$coefficients[[1]]
205 nrep <- 1000
206 TT <- nrow(berndt.df)
207 # here I create the containers to be filled with the generated data.
208 y_bt <- y_mc <- array(1, c(TT, nrep))
209 x_bt <- x_mc <- array(alpha, c(TT, nrep))
210
211 # now, the loop
212
213 for (i in 1:nrep){
214   u <- rnorm(TT)
215   res <- sample(mod_TAN$residuals, replace = T) # this (re)samples from the
      data
216
217   x_mc[, i] <- alpha + vol * u # the Monte Carlo way
218   x_bt[, i] <- alpha+res      # the bootstrap way
219
220   # now we simply construct and store the bootstrapped and MC cumulative
      returns
221   for (j in 2:TT){
222     y_mc[j, i] <- y_mc[j-1, i] * (1 + x_mc[j, i])
223     y_bt[j, i] <- y_bt[j-1, i] * (1 + x_bt[j, i])
224   }
225 }
226
227 # now we want to construct the series of means and quantiles of the
      resulting collection of drawn series
228 for (i in 1:TT){
229   # obtaining the means
230   berndt.df$y_bt_mean[i] <- mean(y_bt[i, ])
231   berndt.df$x_bt_mean[i] <- mean(x_bt[i, ])
232   berndt.df$y_mc_mean[i] <- mean(y_mc[i, ])
233   berndt.df$x_mc_mean[i] <- mean(x_mc[i, ])
234
235   # and the quantiles
236   berndt.df$y_bt_q05[i] <- quantile(y_bt[i, ], 0.05)
237   berndt.df$x_bt_q05[i] <- quantile(x_bt[i, ], 0.05)
238   berndt.df$y_mc_q05[i] <- quantile(y_mc[i, ], 0.05)
239   berndt.df$x_mc_q05[i] <- quantile(x_mc[i, ], 0.05)
240
241   berndt.df$y_bt_q95[i] <- quantile(y_bt[i, ], 0.95)
242   berndt.df$x_bt_q95[i] <- quantile(x_bt[i, ], 0.95)
243   berndt.df$y_mc_q95[i] <- quantile(y_mc[i, ], 0.95)
244   berndt.df$x_mc_q95[i] <- quantile(x_mc[i, ], 0.95)
245
246 }
247

```

```

248 ## -----
249 # plotting
250 plot <- ggplot(data= berndt.df, aes(x = date)) +
251   geom_line(aes(y = TAN, color = "TAN"), size = 1) +
252   geom_line(aes(y = x_mc_mean, color = "x_mc_mean"), size = 1) +
253   geom_line(aes(y = x_mc_q05, color = "x_mc_q05"), size = 1) +
254   geom_line(aes(y = x_mc_q95, color = "x_mc_q95"), size = 1) +
255   labs(title = "Simulation",
256        x = "Time", y = "Monthly Returns") +
257   scale_color_manual(values = c("TAN" = "blue", "x_mc_q05" =
258     "red", "x_mc_q95" = "red", "x_mc_mean" = "green")) +
259   theme_minimal() +
260   theme(axis.line = element_line(color = "black"))
261 ## -----
262 # Value at Risk via Monte Carlo simulation
263 ## -----
264 s1_mc=x_mc[2,]
265 hist(s1_mc, breaks = seq(min(s1_mc), max(s1_mc), l = 20+1),prob=TRUE, main
266   = "histogram of monthly returns")
267 curve(dnorm(x,mean=mean(s1_mc),sd=sd(s1_mc)),col='darkblue',lwd=2,add=TRUE)
268 VaR_mc <- quantile(s1_mc, 0.05)
269 VaR_mc

```

Chapter 4

Factor Models for Asset Prices and Returns

4.1 Introduction: Factor Models and Reduction in Dimensionality

The traditional approach to asset allocation among N risky assets requires the prediction of their future distribution $\mathbf{r} \sim \mathcal{D}(\mu, \Sigma)$. One of the most relevant problems in the implementation of the traditional approach to portfolio allocation is dimensionality. The implementation of asset allocation and risk measurement among n assets requires the estimation of a very large number of parameters: $\frac{n(n+1)}{2} + n$. The relevant dimension for the use of factor models in asset allocation is the time-series as Factor models allows to reduce of the dimensionality of the number of parameters to be estimated to derive the predictive distribution of returns. Moreover, linear multi-factor models (e.g., [Fama and French, 1993](#); [Fama and French, 2015](#); [Ang, 2014](#); [Hou et al., 2018](#)) represent the workhorse of empirical asset pricing. These models have been also successfully employed to parsimoniously characterize the cross-section of average one-period (often monthly) returns.

4.2 Factor Models: Time-Series Representation

The statistical distribution of excess returns on N assets ($i=1\dots n$) can be conditioned on a vector of K factors \mathbf{f} (where N is large and K is small)

$$\begin{aligned}
 r_{t,t+k}^i &= \gamma_0^i + \gamma_1^{i'} \mathbf{f}_{t,t+k} + v_{t,t+k}^i & (4.1) \\
 \mathbf{f}_{t,t+k} &= \boldsymbol{\mu}^f + \mathbf{H}^f \boldsymbol{\epsilon}_{t,t+k} \\
 \boldsymbol{\Sigma}^f &= \mathbf{H}^f \mathbf{H}^{f'} \\
 \mathbf{E}(v_{t,t+k}^i, v_{t,t+k}^j) &= 0 \\
 \mathbf{E}(v_{t,t+k}^i, \boldsymbol{\epsilon}_{t,t+k}^j) &= 0 \\
 \boldsymbol{\epsilon}_{t,t+k} &\sim \mathcal{D}(\mathbf{0}, \mathbf{I})
 \end{aligned}$$

Note that the projection of the large number of N excess returns on the small number K of factors allows decomposing the compensation for risk into two orthogonal components: a common risk component captured by the factors $\gamma_1^{i'} \mathbf{f}$ and an idiosyncratic component captured by the residuals of the projection of returns on factors $v_{t,t+k}^i$. By their nature, idiosyncratic components are not correlated with each other and therefore while the variance-covariance matrix of N excess returns contains $N(N+1)/2$ parameters the variance-covariance matrix of the residuals of the projections of excess returns on factors is diagonal and contains only N parameters to be estimated. The application of the CER model for asset allocation to select a portfolio from N assets requires the estimation of $N+N(N+1)/2$ parameters, while the adoption of a structure of K factors requires the estimation of $(2N+NK) + (K+K(K+1)/2)$ parameters. Think, for example, of an asset allocation problem with 30 assets and 4 factors. The CER would require the estimation of 505 parameters, the factor model would reduce that number to 194. The traditional factor model results from a combination of the application of the CER to factors and of the projection of returns to factors. the constancy of conditional expectations of factors implies the absence of predictability for them which immediately translates into the absence of predictability for returns. In the traditional factor model, we have:

$$\begin{aligned}
 E(\mathbf{f}_{t,t+k}) &= \boldsymbol{\mu}^f & (4.2) \\
 E(r_{t,t+k}^i) &= \gamma_0^i + \gamma_1^{i'} E(\mathbf{f}_{t,t+k}) \\
 &= \gamma_0^i + \gamma_1^{i'} \boldsymbol{\mu}^f
 \end{aligned}$$

and the model rules out predictability both for factors and returns.

4.3 Factor Models: Cross-Sectional representation

If we consider the cross-section of returns rather than their time-series, the multifactor model has the following cross-sectional representation for the $(Nx1)$ vector of returns between time t and time $t + k$ as a linear projection of factors between time t and time $t + k$

$$\begin{aligned}
 \mathbf{r}_{t,t+k} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B} \underset{(Kx1)}{\mathbf{f}_{t,t+k}} + \underset{(Nx1)}{\mathbf{v}_t} & (4.3) \\
 \mathbf{f}_{t,t+k} &= \underset{(Kx1)}{\mu^f} + \underset{(KxK)}{\mathbf{H}^f} \underset{(Kx1)}{\epsilon^f} \\
 \Sigma^v &= \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma_n \end{bmatrix} \\
 \Sigma^f &= \mathbf{H}^f \mathbf{H}^{f'}.
 \end{aligned}$$

The specification and estimation of a factor model allow to parsimoniously compute optimal portfolio weights. In fact, we have

$$\begin{aligned}
 \underset{(Nx1)}{E} \mathbf{r}_{t,t+k} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B} \underset{(Kx1)}{\mu^f} & (4.4) \\
 \underset{(NxN)}{\Sigma}^r &= \underset{(NxK)}{B} \underset{(KxK)}{\Sigma}^f \underset{(KxN)}{B'} + \underset{(NxN)}{\Sigma}^v
 \end{aligned}$$

from which optimal weights are derived for the different specifications of the optimal portfolio.

4.4 Factor-based Portfolios and Factor Exposures

After optimal portfolio weights have been set using a specific criterion, the exposure of portfolios to factors can be assessed by computing the share of the total portfolio variance attributable to each factor. Define the returns of an optimal portfolio obtained by combining n assets as $r_{t+1}^p = \sum_{i=1}^N w_i r_{t+1}^i$

$$r_{t+1}^p = \alpha_1 + \beta_{f^1} f_{t+1}^1 + \beta_{f^2} f_{t+1}^2 + \cdots + \beta_{f^k} f_{t+1}^k + v_{t+1}$$

$$\begin{aligned}
\text{Var}(r_{t+1}^p) &= \text{Cov}(r_{t+1}^p, r_{t+1}^p) \\
&= \beta_{f^1} \text{Cov}(f_{t+1}^1, r_{t+1}^p) + \dots + \beta_{f^k} \text{Cov}(f_{t+1}^k, r_{t+1}^p) \\
&\quad + \text{Cov}(v_{t+1}, r_{t+1}^p)
\end{aligned}$$

The factor exposure can then be computed as the share of the total variance portfolio attributable to each factor:

$$EXP_{f^i}^p = \frac{\beta_{f^i} \text{Cov}(f_{t+1}^i, r_{t+1}^p)}{\text{Var}(r_{t+1}^p)}$$

The above decomposition resembles the risk parity approach that we have seen in the first chapter. In fact, as risk parity can be considered as an alternative method to allocate assets, "smart beta" strategies can be implemented through alternative weighting methods that emphasize the exposures to specific factors.

4.5 A single factor model: The CAPM

We shall illustrate factor models with the most famous single factor model for asset returns: the CAPM (Sharpe, 1964; Lintner, 1965). In the CAPM the common factor to all asset returns is identified with the market. The CAPM has the following time-series representation for the return the i -th assets to be included in the portfolio

$$\begin{aligned}
\left(r_t^i - r_t^{rf} \right) &= \beta_{0,i} + \beta_{1,i} \left(r_t^m - r_t^{rf} \right) + u_{i,t} \\
\left(r_t^m - r_t^{rf} \right) &= \mu_m + u_{m,t} \\
u_{i,t} &\sim n.i.d. \left(0, \sigma_i^2 \right) \\
\begin{pmatrix} u_{i,t} \\ u_{m,t} \end{pmatrix} &\sim n.i.d. \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{ii} & 0 \\ 0 & \sigma_{mm} \end{pmatrix} \right]
\end{aligned}$$

The hypothesis of crucial importance for the validity of the factor representation is that of orthogonality between the common shock $u_{m,t}$ and all the idiosyncratic shocks $u_{i,t}$. The cross-sectional representation of the vectors of N returns in the CAPM is then:

$$\begin{aligned}
 \mathbf{r}_t &= \beta_0 + \beta_1 r_t^m + \mathbf{u}_t \\
 r_t^m &= E(r^m) + \sigma_m \mathbf{u}_{m,t} \\
 \Sigma &= \beta_1 \beta_1' \sigma_m^2 + \Sigma_u \\
 \mu &= \beta_0 + \beta_1 E(r^m)
 \end{aligned}$$

Note that while if a CER model is adopted for all returns the total number of parameters to be estimated is $N + \frac{N(N+1)}{2}$ (the parameters in the mean vector+the parameters in the variance-covariance matrix of returns), while μ, Σ can be obtained with the estimation of $3N+2$ parameters when the CAPM is adopted.

4.5.1 Asset Allocation with the CER and the CAPM in R

The following R code allows uploading a data set of US stock market returns, performing descriptive and graphical analysis of the performance of the single index model applied to returns and tracking the capability of the model for returns to track prices in the case of a specific stock, implementing optimal portfolio allocation with the CER model, implementing optimal portfolio allocation with the CAPM model, comparing the results, and checking the validity of the CAPM model by comparing the correlation matrix of returns with the correlation matrix of their estimated idiosyncratic components. The code also exploits alternative approaches to run CAPM regressions for many assets using first multivariate least squares and then iterating OLS regressions for all available returns.

```

1 # elaboration on the original produced by E.Zivot by C. Favero
2 # author: Carlo Favero
3 # created: July, 2021
4 #
5 # comments: Original Examples follow chapter 11 in Zivot and Wang (2006)
6
7
8 rm(list=ls()) #Removes all items in Environment!
9 #setwd(path)
10 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
11
12 # set output options
13 # options(width = 70, digits=4)
14 listofpackages <- c("dygraphs",
15                    "dplyr", "ellipse", "reshape2", "ggplot2", "PerformanceAnalytics", "zoo")
16
17 for (j in listofpackages){
18   if(sum(installed.packages()[, 1] == j) == 0) {
19     install.packages(j)
20   }

```

```

20   library(j, character.only = T)
21 }
22 install.packages("fEcofin", repos="http://R-Forge.R-project.org")
23 # load required packages
24 library(fEcofin)           # various data sets
25
26 #####
27 # Data Loadings and Transform Descriptive Analysis
28 #####
29
30 # create data frame with dates as rownames
31 berndt.df = berndtInvest[, -1]
32 berndt.df$date <- as.Date(berndtInvest[, 1])
33 rownames(berndt.df) = as.character(berndtInvest[, 1])
34 colnames(berndt.df)
35 dimnames(berndt.df)[[2]] #command alternative to the previous one
36
37 # transform the data and compute cumulative returns
38
39 t0 <- which(berndt.df$date == "1978-01-01")
40 t1 <- which(berndt.df$date == "1987-12-01")
41
42 series_names <-
43   c("CITCRP", "CONED", "CONTIL", "DATGEN", "DEC", "DELTA", "GENMIL", "GERBER", "IBM",
44     "MARKET", "MOBIL", "PANAM", "PSNH", "TANDY", "TEXACO", "WEYER", "RKFREE")
45
46 for (name in series_names) {
47   P_col_name <- paste0(name, "_P")
48   LP_col_name <- paste0("L", P_col_name)
49   berndt.df[t0, P_col_name] <- 1
50   for (i in (t0+1):(t1)) {
51     berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] *
52       (1+berndt.df[i, name][[1]] )
53   }
54   berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
55 }
56 # add a trend to the database
57 berndt.df$TREND <- array(data = NA, dim = nrow(berndt.df))
58 berndt.df[t0, c("TREND")] <- 1 # don't need to repeat the value to make the
59   array being assigned be of the same length. be careful though as it is
60   one of the few cases of exception
61
62 #####
63 # Descriptive Analysis of prices and returns
64 #####
65 # plot log prices
66 ggplot(berndt.df, aes(x = date)) +
67   geom_line(aes(y = LTEXACO_P), color = "blue", size = 1, linetype =
68     "solid") +
69   geom_line(aes(y = LMARKET_P), color = "green", size = 1, linetype =
70     "solid") +
71   labs(x = "Date", y = "Portfolios TEXACO and MKT") +

```

```

66   ylim(-0.5, 2) +
67   theme_minimal() +
68   theme(
69     legend.position = "topleft",
70     legend.title = element_blank(),
71     legend.text = element_text(size = 12),
72     axis.text = element_text(size = 12),
73     axis.title = element_text(size = 14),
74     plot.title = element_text(size = 16, hjust = 0.5)
75   ) +
76   scale_color_manual(
77     values = c("blue", "green"),
78     guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
79   ) +
80   guides(fill = guide_legend(override.aes = list(size = 2)))
81 # plot returns
82 ggplot(berndt.df, aes(x = date)) +
83   geom_line(aes(y = TEXACO), color = "blue", size = 1, linetype = "solid") +
84   geom_line(aes(y = MARKET), color = "green", size = 1, linetype = "solid")
85   +
86   labs(x = "Date", y = "Portfolios TEXACO and MKT") +
87   ylim(-0.45, 0.45) +
88   theme_minimal() +
89   theme(
90     legend.position = "topleft",
91     legend.title = element_blank(),
92     legend.text = element_text(size = 12),
93     axis.text = element_text(size = 12),
94     axis.title = element_text(size = 14),
95     plot.title = element_text(size = 16, hjust = 0.5)
96   ) +
97   scale_color_manual(
98     values = c("blue", "green"),
99     guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
100  ) +
101  guides(fill = guide_legend(override.aes = list(size = 2)))
102
103
104 #####
105 # CAPM FOR TEXACO
106 #####
107 capm_tex<-lm(TEXACO ~ MARKET, data=berndt.df)
108 summary(capm_tex)
109 berndt.df$TEXACO_fitted<-capm_tex$fitted.values
110
111 #fitting returns
112
113 plot(berndt.df$date[t0:t1],berndt.df$TEXACO[t0:t1], col = 'blue', type =
114       "l",
115       ylab = " actual and fitted returns", xlab = "date",lwd = 2,)
116 lines(y = rep(mean(berndt.df$TEXACO[t0:t1], na.rm = T),

```

```

length(berndt.df$TEXACO[t0:t1])), x = berndt.df$date[t0:t1], col =
"red")
116 lines(y = berndt.df$TEXACO_fitted[t0:t1], x = berndt.df$date[t0:t1], col =
"green",lwd = 2)
117 legend("topright", legend = c("TEXACO ACTUAL", "TEXACO FITTED"),
118 col = c("blue", "green"), lty = 1)
119 grid(nx = 6, ny = 7, col = "lightgray", lty = "dotted",
120 lwd = par("lwd"), equilogs = TRUE)
121
122 #fitting prices
123 berndt.df$TEXACO_P_FITTED <- array(data = NA, dim = nrow(berndt.df))
124 berndt.df$TEXACO_P_FITTED[t0] <- 1
125 for (i in (t0+1):(t1)) {
126 berndt.df$TEXACO_P_FITTED[i] <- berndt.df$TEXACO_P_FITTED[i-1] * (1 +
berndt.df$TEXACO_fitted[i])}
127
128 plot(berndt.df$date[t0:t1],berndt.df$TEXACO_P[t0:t1], col = 'blue', type =
"l",
129 ylab = " actual and fitted prices", xlab = "date",lwd =
2,ylim=c(0.9,5))
130 #lines(y = rep(mean(berndt.df$TEXACO[t0:t1], na.rm = T),
length(berndt.df$TEXACO[t0:t1])), x = berndt.df$date[t0:t1], col =
"red")
131 lines(y = berndt.df$TEXACO_P_FITTED[t0:t1], x = berndt.df$date[t0:t1], col
= "green",lwd = 2)
132 legend("topleft", legend = c("TEXACO ACTUAL", "TEXACO FITTED"),
133 col = c("blue", "green"), lty = 1)
134 grid(nx = 6, ny = 7, col = "lightgray", lty = "dotted",
135 lwd = par("lwd"), equilogs = TRUE)
136 #dev.copy2pdf(width = 8.5, out.type = "pdf",file="CAPM.pdf")
137 #dev.off()
138 #####
139 # Optimal Portfolio weights with the CER approach
140 #####
141
142 returns.df=berndt.df[, c(1:9,11:16)]
143 #returns.df = berndt.df[, c(-10, -17)]
144 exreturns.df=returns.df-berndt.df$RKFREE
145 returns.mat = as.matrix(exreturns.df)
146 n.obs = nrow(returns.mat)
147
148 #Estimation of CER model parameters
149 cov.sample=var(returns.mat)
150 mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
151 #
152 # compute global min variance portfolio
153 #
154 w.gmin.sample = solve(var(returns.mat))%*%rep(1,nrow(cov.sample))
155 w.gmin.sample = w.gmin.sample/sum(w.gmin.sample)
156 colnames(w.gmin.sample) = "sample"
157 barplot(t(w.gmin.sample), horiz=F, main="Weights", col="blue", cex.names =
0.75, las=2)

```

```

158 #####
159 # A Single index model: the CAPM
160 #####
161
162 ##
163 ## use multivariate regression and matrix algebra
164 ##
165 returnsmkt.df=berndt.df[, c(10:10,17:17)]
166 #returns.df = berndt.df[, c(-10, -17)]
167 returnsmkt.df$EXRETMKT=returnsmkt.df$MARKET-returnsmkt.df$RKFREE
168 market.mat = as.matrix(returnsmkt.df[,3, drop=F])
169 n.obs = nrow(returns.mat)
170 X.mat = cbind(rep(1,n.obs),market.mat)
171 colnames(X.mat)[1] = "intercept"
172 XX.mat = crossprod(X.mat)
173
174 # multivariate least squares
175 G.hat = solve(XX.mat)%%crossprod(X.mat,returns.mat)
176 # can also use solve(qr(X.mat), returns.mat)
177 beta.hat = G.hat[2,]
178 E.hat = returns.mat - X.mat%%G.hat
179 #D.hat=crossprod(E.hat)
180 diagD.hat = diag(crossprod(E.hat)/(n.obs-2))
181 # compute R2 values from multivariate regression
182 sumSquares = apply(returns.mat, 2, function(x) {sum( (x - mean(x))^2 )})
183 R.square = 1 - (n.obs-2)*diagD.hat/sumSquares
184
185 # print and plot results
186 cbind(beta.hat, diagD.hat, R.square)
187
188 par(mfrow=c(1,2))
189 barplot(beta.hat, horiz=T, main="Beta values", col="blue", cex.names =
190       0.75, las=1)
191 barplot(R.square, horiz=T, main="R-square values", col="blue", cex.names =
192       0.75, las=1)
193 par(mfrow=c(1,1))
194
195 # compute single index model covariance/correlation matrices
196 cov.si = as.numeric(var(market.mat))*beta.hat%%t(beta.hat) +
197       diag(diagD.hat)
198 cor.si = cov2cor(cov.si)
199 #
200 # COMPARE CORRELATIONS
201 #
202 # FACTOR MODEL BASED CORRELATION MATRIX using plotcorr() from ellipse
203 # package
204 #
205 rownames(cor.si) = colnames(cor.si)
206 ord <- order(cor.si[1,])
207 ordered.cor.si <- cor.si[ord, ord]
208 plotcorr(ordered.cor.si, col=cm.colors(11)[5*ordered.cor.si + 6])
209 plotcorr(cor.si, col=cm.colors(11)[5*cor.si + 6])

```

```

206 #
207 # SAMPLE CORRELATION MATRIX
208 #
209 cor.sample = cor(returns.mat)
210 ord <- order(cor.sample[1,])
211 ordered.cor.sample <- cor.sample[ord, ord]
212 plotcorr(ordered.cor.sample, col=cm.colors(11)[5*ordered.cor.sample + 6])
213 plotcorr(cor.sample, col=cm.colors(11)[5*cor.sample + 6])
214 #
215 # CAPM residuals CORRELATION MATRIX
216 #
217 cor.resid = cor(E.hat)
218 ord <- order(cor.resid[1,])
219 ordered.cor.resid <- cor.resid[ord, ord]
220 plotcorr(ordered.cor.resid, col=cm.colors(11)[5*ordered.cor.resid + 6])
221 #
222 # compute global min variance portfolio
223 #
224 # use CAPM covariance (1-factor model)
225 w.gmin.si = solve(cov.si)%*%rep(1,nrow(cov.si))
226 w.gmin.si = w.gmin.si/sum(w.gmin.si)
227 colnames(w.gmin.si) = "single.index"
228
229
230 #par(mfrow=c(2,1))
231 #barplot(t(w.gmin.si), horiz=F, main="Single Index Weights", col="blue",
232 #        cex.names = 0.75, las=2)
233 #barplot(t(w.gmin.sample), horiz=F, main="Sample Weights", col="blue",
234 #        cex.names = 0.75, las=2)
235 #par(mfrow=c(1,1))
236
237 #compare weights delivered by the two alternative methods
238 pdf("output.pdf", width = 10, height = 8)
239 par(mfrow = c(2, 1))
240 barplot(t(w.gmin.si), horiz=F, main="Single Index Weights", col="blue",
241 #        cex.names = 0.75, las=2)
242 barplot(t(w.gmin.sample), horiz=F, main="Sample Weights", col="blue",
243 #        cex.names = 0.75, las=2)
244 par(mfrow = c(1, 1))
245 dev.off()
246
247 # compare means and sd values on global min variance portfolios
248 mu.vals = colMeans(returns.mat)
249 mu.gmin.si = as.numeric(crossprod(w.gmin.si, mu.vals))
250 sd.gmin.si = as.numeric(sqrt(t(w.gmin.si)%*%cov.si)%*%w.gmin.si))
251 mu.gmin.sample = as.numeric(crossprod(w.gmin.sample, mu.vals))
252 sd.gmin.sample =
253   as.numeric(sqrt(t(w.gmin.sample)%*%var(returns.mat)%*%w.gmin.sample))
254 cbind(mu.gmin.si,mu.gmin.sample, sd.gmin.si, sd.gmin.sample)

```

```

253
254
255 #####
256 ## AN ALTERNATIVE APPROACH to compute parameters in CAPM:
257 ## use lm function to compute single index model regressions for each asset
258 #####
259
260 asset.names = colnames(returns.mat)
261 asset.names
262
263 # initialize list object to hold regression objects
264
265 reg.list = list()
266 # loop over all assets and estimate time series regression
267 for (i in asset.names) {
268   reg.df = berndt.df[, c(i, "MARKET")]
269   si.formula = as.formula(paste(i, "~", "MARKET", sep=" "))
270   reg.list[[i]] = lm(si.formula, data=reg.df)
271 }
272
273 # examine the elements of reg.list - they are lm objects!
274 names(reg.list)
275 class(reg.list$CITCRP)
276 reg.list$CITCRP
277 summary(reg.list$CITCRP)
278
279 # plot actual vs. fitted over time
280 # use chart.TimeSeries() function from PerformanceAnalytics package
281
282 dataToPlot = cbind(fitted(reg.list$CITCRP), berndt.df$CITCRP)
283 colnames(dataToPlot) = c("Fitted", "Actual")
284 dev.off()
285
286 # Verify the data
287 str(dataToPlot)
288 summary(dataToPlot)
289
290 # Create the time series chart
291 chart.TimeSeries(dataToPlot, main = "Single Index Model for CITCRP",
292                 colorset = c("black", "blue"), legend.loc = "bottomleft")
293
294
295 # scatterplot of the single index model regression
296 plot(berndt.df$MARKET, berndt.df$CITCRP, main="SI model for CITCRP",
297      type="p", pch=16, col="blue",
298      xlab="MARKET", ylab="CITCRP")
299 abline(h=0, v=0)
300 abline(reg.list$CITCRP, lwd=2, col="red")
301
302 ## extract beta values, residual sd's and R2's from list of regression
303 ## brute force loop

```

```

304 reg.vals = matrix(0, length(asset.names), 3)
305 rownames(reg.vals) = asset.names
306 colnames(reg.vals) = c("beta", "residual.sd", "r.square")
307 for (i in names(reg.list)) {
308   tmp.fit = reg.list[[i]]
309   tmp.summary = summary(tmp.fit)
310   reg.vals[i, "beta"] = coef(tmp.fit)[2]
311   reg.vals[i, "residual.sd"] = tmp.summary$sigma
312   reg.vals[i, "r.square"] = tmp.summary$r.squared
313 }
314 reg.vals
315
316 # alternatively use R apply function for list objects - lapply or sapply
317 extractRegVals = function(x) {
318   # x is an lm object
319   beta.val = coef(x)[2]
320   residual.sd.val = summary(x)$sigma
321   r2.val = summary(x)$r.squared
322   ret.vals = c(beta.val, residual.sd.val, r2.val)
323   names(ret.vals) = c("beta", "residual.sd", "r.square")
324   return(ret.vals)
325 }
326 reg.vals = sapply(reg.list, FUN=extractRegVals)
327 t(reg.vals)

```

4.6 Validating Factor Models

In the previous section, we have seen that a first validation of a factor model can be implemented by exploiting the fact that the diagonality of the variance-covariance matrix of the residuals coming from projecting asset returns on factors is a necessary—and testable—requirement for the validity of any factor model. However, further validation can be based on testing restrictions on the estimated coefficients in any given factor model.

Consider once again the time-series representation of a factor model

$$r_{t+1}^i = \alpha_1 + \beta_i^{f^1} f_{t+1}^1 + \beta_i^{f^2} f_{t+1}^2 + \cdots + \beta_i^{f^k} f_{t+1}^k + v_{t+1} \quad (4.5)$$

After having estimated N equations for the N assets you have available the following k vectors of coefficients, each of length N :

$\beta^{f^1}, \beta^{f^2}, \dots, \beta^{f^k}$. Using the sample of t observations on the returns of the N assets you can compute the vector of length N of average sample returns for the assets: $E(\mathbf{r})$.

You can now run the affine expected return-beta cross-sectional regression:

$$E(\mathbf{r}) = \gamma_0 + \gamma_1 \beta_{f_1} + \gamma_2 \beta_{f_2} + \cdots + \gamma_k \beta_{f_k} + \mathbf{u}$$

A two-step test [Fama and MacBeth \(1973\)](#) for the validity of any factor model can be run by considering the following null hypothesis:

$$\hat{\gamma}_0 = \bar{r}^f, \quad \hat{\gamma}_i = E(f^i)$$

care must be exercised in the test as the variance-covariance matrix of the residuals in the cross-sectional regression will not be diagonal and corrections for heteroscedasticity should be implemented. Note also that, if both test assets and factors are excess returns, the validity of the model can be simply tested by evaluating the null that all intercepts in the time-series model for excess returns are zero. This null is inevitably rejected in the single-factor CAPM model. Two industries have emerged (i) the factors "zoo", that looks for omitted factors (ii) the performance evaluation industry that classifies fund manager performance according to their alphas.

4.6.1 Which Factors ?

Many different set of factors have been considered in the literature :

- Fundamental Factors
 - Fama-French five factors with observable characteristics and estimated betas (MKT, SMB, HML, RMW, CMA and momentum MOM)
 - BARRA factors with known time-invariant betas and unobservable factor realizations estimated by cross-sectional regressions.
- Macroeconomic Factors (inflation, growth and uncertainty)
- Statistical Factors (for example principal components)

4.7 Factor Models with Predictability

Factor models are commonly used to characterize parsimoniously the predictive distribution of asset returns. Specifically, multi-factor models in which k factors characterize in a lower parametric dimension the distribution of n asset returns, have the following general form:

$$r_{i,t+1} = \alpha_i + \beta_i' \mathbf{f}_{t+1} + v_{i,t+1}, \quad (4.6)$$

$$\mathbf{f}_{t+1} = E(\mathbf{f}_{t+1} | I_t) + \epsilon_{t+1} \quad \text{with } \epsilon_{t+1} \sim \mathcal{D}(\mathbf{0}, \Sigma) \quad (4.7)$$

where $Cov(v_{i,t+1}, v_{j,t+1}) = 0$ for $i \neq j$, \mathbf{f}_{t+1} is a k -dimensional vector of factors at time $t + 1$, $r_{i,t+1}$ is the return on the i -th of the n assets at time $t + 1$, and the vector β_i'

contains the loadings for asset i on the k factors. Equation (4.6) specifies the conditional distribution of returns on factors, while equation (4.7) specifies the predictive distribution for factors at time $t+1$ conditioning on information available at time t . A baseline specification for this system assumes away factors predictability thus implying that conditional expectations of factors have no variance (i.e., $E(\mathbf{f}_{t+1} | I_t) = \mu$).

In equation (4.6) it is often assumed without further qualification that returns and factors are stationary variables. The model, however, leaves prices undetermined: the long-run forecast for asset prices is independent from the long-run forecast of factors. A factor model that leaves asset prices undetermined does not exploit information in the data that can be used for (i) factor selection, and (ii) asset allocation.

Consider an asset i and denote its log one-period return by $r_{i,t}$. We define the log price of this asset as:

$$\ln P_{i,t} = \ln P_{i,t-1} + \mathbf{r}_{i,t} , \quad (4.8)$$

i.e., prices of any asset are cumulative returns. The analogous of the (log) price for an asset can be constructed for any given factor. We define as factor (log) price the cumulative returns of a portfolio investing in standard factors (e.g., the aggregate market return). The generic prices associated to factors with a log period returns of \mathbf{f}_t evolve according to the following process:

$$\ln \mathbf{F}_t = \ln \mathbf{F}_{t-1} + \mathbf{f}_t . \quad (4.9)$$

If returns to test assets and factors are stationary, then portfolio prices and factor-prices are non-stationary. In fact, imagine simulating data using the model given by equations (4.6), (4.7), (4.8), (4.9). The simulated data will deliver a linear relationship between returns and factors but no relationship between asset prices and factor prices. Asset prices and factor prices will follow two *unrelated* stochastic trends. In technical jargon, the model given by (4.6)–(4.9) rules out the hypothesis of the existence of a long-run relation (cointegration) between asset prices and factor prices by assumption. The presence of co-integration which is, at least in some cases, borne out by the data it is not tested for, nor it is reflected in the factor model specification when appropriate. This has two implications. First, in the absence of cointegration, the opportunity of discarding factor models that do not explain the long-run trends in prices is not exploited. Second, in the presence of cointegration, its implications for portfolio returns predictability are left unexplored.

In fact, if factor prices are the non-stationary variables that drive the non-stationary dynamics of portfolio prices, then a linear combination of prices and risk drivers should be stationary, i.e., asset and factor prices should be cointegrated.

Consider the following model describing the exposure of a given portfolio price

$P_{i,t}$ to factor prices \mathbf{F}_t :

$$\ln P_{i,t} = \alpha_{0,i} + \alpha_{1,i}t + \beta'_i \ln \mathbf{F}_t + u_{i,t} .$$

The estimation of such regression delivers stationary residuals $u_{i,t}$ anytime the chosen set of factor prices captures the stochastic trend that determines the long-run dynamics of prices. In this case, the linear combination of the right-hand side variables of the equation defines the long-run equilibrium value determined by the factor prices and $u_{i,t}$ captures temporary deviations of asset prices from it. Thus, it is natural to refer to the residuals $u_{i,t}$ as the “Equilibrium Correction Term” (henceforth, *ECT*) associated with asset i at time t . Formally, we define the residual from the long-run cointegrating relationship as:

$$ECT_{i,t} \equiv \ln P_{i,t} - \hat{\alpha}_{0,i} - \hat{\alpha}_{1,i}t - \hat{\beta}'_i \ln \mathbf{F}_t . \quad (4.10)$$

For expository purposes, it is useful to specify the error term $u_{i,t}$ as an AR(1) process. In sum, we model the joint distribution of asset prices, factor prices, asset returns and factors as follows:

$$\begin{aligned} \ln P_{i,t+1} &= \alpha_{0,i} + \alpha_{1,i}t + \beta'_i \ln \mathbf{F}_{t+1} + u_{i,t+1} & (4.11) \\ u_{i,t+1} &= \rho_i u_{i,t} + v_{i,t+1} \\ \mathbf{f}_{t+1} &= E(\mathbf{f}_{t+1} | I_t) + \epsilon_{t+1} \\ \ln P_{i,t} &= \ln P_{i,t-1} + r_{i,t} \\ \ln \mathbf{F}_t &= \ln \mathbf{F}_{t-1} + \mathbf{f}_t \end{aligned}$$

where $\epsilon_{t+1} \sim \mathcal{D}(\mathbf{0}, \Sigma)$, $u_{i,t+1}$ and $v_{i,t+1}$ have zero mean and variance $\sigma_{u,i}^2$ and $\sigma_{v,i}^2$, respectively, and $Cov(v_{i,t+1}, v_{j,t+1}) = 0$ for $i \neq j$.

By taking first differences of our model in (4.11) we obtain a novel specification for returns and factors, where asset returns relate to factors *plus* the *ECT*:

$$r_{i,t+1} = \alpha_{1,i} + \beta'_i \mathbf{f}_{t+1} + \underbrace{(\rho_i - 1)}_{\delta_i} \underbrace{u_{i,t}}_{\equiv ECT_{i,t}} + v_{i,t+1}. \quad (4.12)$$

Eq. (4.12) represents the Factor Error Correction Model (FECM).¹

Two comments are in order. First, we include a linear trend in Eq. (4.11) since it allows us to recover the standard short-run specification—returns are regressed on factors plus a constant—when taking first-differences. In other words, a positive α_1

¹The equilibrium correction representation (4.12) of cointegrated time-series (see the system of equations in (4.11)) is warranted by the [Engle and Granger \(1987\)](#) representation theorem.

in the long-run relation (4.11) generates “alpha” in returns.²

Second, when $ECT_{i,t}$ is stationary, then asset and factor prices are cointegrated. The stationarity of $ECT_{i,t}$ implies that, in the relation (4.12) linking returns to factors, this term appears with a coefficient δ_i capturing the speed with which the system eliminates disequilibria with respect to the long-run relationship. Indeed, δ_i is related to the persistence ρ_i of $ECT_{i,t}$, see Eq. (4.12).

When factor prices explain the buy-and-hold value of a portfolio, cointegration implies that portfolio returns respond to the Equilibrium Correction Term so far omitted in the empirical asset pricing literature. The inclusion of the ECT ensures that the specification for returns is consistent with the long-run relationship between asset and factor prices. The omission of the ECT leads to a misspecification of the factor model, in the sense that the factor model leaves price dynamics undetermined.

Interestingly, a traditional factor model would not be affected by omitting the disequilibrium term only if factor prices and asset prices are not cointegrated (i.e., when $|\rho_i| = 1$). However, this case also implies that a given factor model is unable to price the buy-and-hold portfolios since asset prices do not track factor prices in the long-run. The significance of the ECM terms generates predictability that is relevant for computing optimal portfolio weights. The standard cross-sectional representation of 1-period ahead returns becomes now

$$\begin{aligned} \mathbf{r}_{t,t+1} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B} \underset{(Kx1)}{\mathbf{f}_{t,t+1}} + \underset{(NxN)}{\Gamma} \underset{(Nx1)}{\mathbf{u}_t} + \underset{(Nx1)}{\mathbf{v}_t} & (4.13) \\ \mathbf{f}_{t,t+1} &= \underset{(Kx1)}{\mu^f} + \underset{(KxK)}{\mathbf{H}^f} \underset{(Kx1)}{\epsilon^f} \\ \Sigma^v &= \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \sigma_n \end{bmatrix} \\ \Sigma^f &= \mathbf{H}^f \mathbf{H}^{f'}. \end{aligned}$$

Predictability emerges as the conditional expectations of one-period ahead expected returns is time varying, the relevant conditional variance-covariance matrix of predicted asset returns also changes as the variance of the one-period ahead predictive

²Moreover, as discussed by Engle and Yoo (1987) and MacKinnon (2010), the inclusion of a trend is a simple way to avoid the dependence of the distribution of test statistics for residuals on α_1 .

error is different for the variance of asset returns. In fact, we have

$$\begin{aligned} E_{(Nx1)} \mathbf{r}_{t,t+1} &= \alpha_{(Nx1)} + B_{(NxK)} \mu^f_{(Kx1)} + \Gamma_{(NxN)(Nx1)} \mathbf{u}_t \\ \Sigma_{(NxN)}^r &= B_{(NxK)} \Sigma^f_{(KxK)} B'_{(KxN)} + \Sigma^v_{(NxN)} \end{aligned} \quad (4.14)$$

Where Γ is a diagonal matrix when asset returns depend exclusively on their own price disequilibria. The analysis of the long-run (cointegrating) relationship between asset prices and factor prices provides an opportunity to validate factor models that is left unexploited by the standard factor model specification in equation (4.6)-(4.7). Furthermore, looking at the short-run FECM specification in (4.12), the omission of the *ECT* omits a source of predictability of the conditional distribution of test assets returns that has relevant consequences for asset allocation and risk management. For example, consider the situation in which the portfolio price is aligned with the long-run value determined by the risk drivers, and assume a negative shock (to price) occurs. The returns predictive distribution based on the *ECT* is then shifted to the right. This shift represents an opportunity to be exploited for asset allocation and relevant information for risk measurement.

4.7.1 An illustration with R

The following R code considers the assets in the previous asset allocation example runs the long runs regressions of asset prices and factor prices and concentrates on a case study on TEXACO to show the relevance of predictability and illustrate how the CAPM can be modified to derive a factor model with returns predictability

```

1
2 # The effect of omitting long-run trends from factor models
3
4 rm(list=ls()) #Removes all items in Environment!
5 #setwd(path)
6 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
7
8 # set output options
9 options(width = 70, digits=4)
10 listofpackages <- c("dygraphs",
11                    "dplyr","ellipse","reshape2","ggplot2","PerformanceAnalytics","zoo")
12
13 for (j in listofpackages){
14   if(sum(installed.packages()[, 1] == j) == 0) {
15     install.packages(j)
16   }
17   library(j, character.only = T)
18 }
19 install.packages("fEcofin", repos="http://R-Forge.R-project.org")

```

```

19
20 # load required packages
21 library(fEcofin)                      # various data sets
22
23 #####
24 # Data Loadings and Transform Descriptive Analysis
25 #####
26
27 # create data frame with dates as rownames
28 berndt.df = berndtInvest[, -1]
29 berndt.df$date <- as.Date(berndtInvest[, 1])
30 rownames(berndt.df) = as.character(berndtInvest[, 1])
31 colnames(berndt.df)
32 dimnames(berndt.df)[[2]] #command alternative to the previous one
33
34 # transform the data and compute cumulative returns
35
36 t0 <- which(berndt.df$date == "1978-01-01")
37 t1 <- which(berndt.df$date == "1987-12-01")
38
39 series_names <-
40     c("CITCRP", "CONED", "CONTIL", "DATGEN", "DEC", "DELTA", "GENMIL", "GERBER",
41     "IBM", "MARKET", "MOBIL", "PANAM", "PSNH", "TANDY", "TEXACO", "WEYER", "RKFREE")
42
43 for (name in series_names) {
44     P_col_name <- paste0(name, "_P")
45     LP_col_name <- paste0("L", P_col_name)
46     berndt.df[t0, P_col_name] <- 1
47     for (i in (t0+1):(t1)) {
48         berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] *
49             (1+berndt.df[i, name][[1]] )
50     }
51     berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
52 }
53
54 # add a trend to the database
55 berndt.df$TREND <- array(data = NA, dim = nrow(berndt.df))
56 berndt.df[t0, c("TREND")] <- 1 # don't need to repeat the value to make the
57     array being assigned be of the same length. be careful though as it is
58     one of the few cases of exception
59
60 for (i in (t0+1):(t1)) {
61     berndt.df[i, "TREND"][[1]] <- berndt.df[i-1, "TREND"][[1]] +1
62 }
63
64 #####
65 # Descriptive Analysis
66 #####
67 ggplot(berndt.df, aes(x = date)) +
68     geom_line(aes(y = LTEXACO_P, color = "TEXACO"), size = 1, linetype =
69         "solid") +
70     geom_line(aes(y = LMARKET_P, color = "MARKET"), size = 1, linetype =
71         "solid") +
72     labs(x = "Date", y = "Portfolios TEXACO and MKT") +

```

```

65   ylim(-0.5, 2) +
66   theme_minimal() +
67   theme(
68     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
69     legend.title = element_blank(),
70     legend.text = element_text(size = 8),
71     axis.text = element_text(size = 8),
72     axis.title = element_text(size = 10),
73     plot.title = element_text(size = 12, hjust = 0.5)
74   ) +
75   scale_color_manual(
76     values = c("blue", "green"),
77     labels = c("TEXACO", "MARKET")
78   )
79
80
81 # plot returns
82 ggplot(berndt.df, aes(x = date)) +
83   geom_line(aes(y = TEXACO, color = "TEXACO"), size = 1, linetype =
84     "solid") +
85   geom_line(aes(y = MARKET, color = "MARKET"), size = 1, linetype =
86     "solid") +
87   labs(x = "Date", y = "Returns TEXACO and MKT") +
88   ylim(-0.45, 0.45) +
89   theme_minimal() +
90   theme(
91     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
92     legend.title = element_blank(),
93     legend.text = element_text(size = 8),
94     axis.text = element_text(size = 8),
95     axis.title = element_text(size = 8),
96     plot.title = element_text(size = 12, hjust = 0.5)
97   ) +
98   scale_color_manual(
99     values = c("blue", "green"),
100    labels = c("TEXACO", "MARKET")
101  )
102
103 #####
104 # Standard CAPM Factor Models
105 #####
106
107 ## use lm function to compute single index model regressions for each asset
108 ##
109 returns.mat = as.matrix(berndt.df[, c(1:9, 11:16)])
110 asset.names = colnames(returns.mat)
111
112 # initialize list object to hold regression objects
113 reg.list = list()
114 # loop over all assets and estimate time series regression

```

```

115 for (i in asset.names) {
116   reg.df = berndt.df[, c(i, "MARKET")]
117   si.formula = as.formula(paste(i,"~", "MARKET", sep=" "))
118   reg.list[[i]] = lm(si.formula, data=reg.df)
119 }
120
121 # examine the elements of reg.list - they are lm objects!
122 names(reg.list)
123 class(reg.list$TEXACO)
124 reg.list$TEXACO
125 summary(reg.list$TEXACO)
126
127 # plot actual vs. fitted over time
128 # use chart.TimeSeries() function from PerformanceAnalytics package
129 dataToPlot = cbind(fitted(reg.list$TEXACO), berndt.df$TEXACO)
130 colnames(dataToPlot) = c("Fitted","Actual")
131 chart.TimeSeries(dataToPlot, main="Single Index Model for TEXACO",
132                 colorset=c("black","blue"), legend.loc="bottomleft")
133
134 # scatterplot of the single index model regression
135 plot(berndt.df$MARKET, berndt.df$TEXACO, main="SI model for CITCRP",
136      type="p", pch=16, col="blue",
137      xlab="MARKET", ylab="TEXACO")
138 abline(h=0, v=0)
139 abline(reg.list$TEXACO, lwd=2, col="red")
140
141 ## extract beta values, residual sd's and R2's from list of regression
142     objects
143 reg.vals = matrix(0, length(asset.names), 3)
144 rownames(reg.vals) = asset.names
145 colnames(reg.vals) = c("beta", "residual.sd", "r.square")
146 for (i in names(reg.list)) {
147   tmp.fit = reg.list[[i]]
148   tmp.summary = summary(tmp.fit)
149   reg.vals[i, "beta"] = coef(tmp.fit)[2]
150   reg.vals[i, "residual.sd"] = tmp.summary$sigma
151   reg.vals[i, "r.square"] = tmp.summary$r.squared
152 }
153 reg.vals
154
155 # print regression results
156
157 par(mfrow=c(1,2))
158 barplot(reg.vals[,1], horiz=T, main="Beta values", col="blue", cex.names =
159         0.75, las=1)
159 barplot(reg.vals[,3], horiz=T, main="R-square values", col="blue",
160         cex.names = 0.75, las=1)
160 par(mfrow=c(1,1))
161
162
163 #####

```

```

164 # CAPM in levels
165 #####
166
167 ## use lm function to compute single index model regressions for each asset
168 ##
169 selected_columns <-
170   c("LCITCRP_P", "LCONED_P", "LCONTIL_P", "LDATGEN_P", "LDEC_P", "LDELTA_P", "LGENMIL_P",
171     "LGERBER_P", "LIBM_P", "LMOBIL_P", "LPANAM_P", "LPSNH_P", "LTANDY_P", "LTEXACO_P", "LWEYER_P")
172
173 # Extract the specified columns and store them in a matrix
174 lprices.mat <- as.matrix(berndt.df[, selected_columns])
175
176 asset.names = colnames(lprices.mat)
177
178 # initialize list object to hold regression objects
179
180 reg1.list = list()
181 # loop over all assets and estimate time series regression
182 for (i in asset.names) {
183   #reg.df = berndt.df[, c(i, "LMARKET_P")]
184   si.formula = as.formula(paste(i, "~", "LMARKET_P+TREND", sep=" "))
185   reg1.list[[i]] = lm(si.formula, data=berndt.df)
186 }
187
188 # examine the elements of reg.list - they are lm objects!
189 names(reg1.list)
190 class(reg1.list$LTEXACO_P)
191 reg1.list$LTEXACO_P
192 summary(reg1.list$LTEXACO_P)
193
194 # plot actual vs. fitted over time
195 # use chart.TimeSeries() function from PerformanceAnalytics package
196 dataToPlot = cbind(fitted(reg1.list$LTEXACO_P), berndt.df$LTEXACO_P)
197 colnames(dataToPlot) = c("Fitted", "Actual")
198 chart.TimeSeries(dataToPlot, main="Single Index Model for price TEXACO",
199                 colorset=c("black", "blue"), legend.loc="bottomleft")
200
201 # scatterplot of the single index model regression
202 plot(berndt.df$LMARKET_P, berndt.df$LTEXACO_P, main="SI model for
203       LTEXACO_P",
204      type="p", pch=16, col="blue",
205      xlab="MARKET", ylab="TEXACO")
206 abline(h=0, v=0)
207 abline(reg.list$TEXACO, lwd=2, col="red")
208
209 ## extract beta values, residual sd's and R2's from list of regression
210 objects
211
212 reg.vals1 = matrix(0, length(asset.names), 3)
213 rownames(reg.vals1) = asset.names
214 colnames(reg.vals1) = c("beta", "residual.sd", "r.square")

```

```

213 for (i in names(reg1.list)) {
214   tmp.fit = reg1.list[[i]]
215   tmp.summary = summary(tmp.fit)
216   reg.vals1[i, "beta"] = coef(tmp.fit)[2]
217   reg.vals1[i, "residual.sd"] = tmp.summary$sigma
218   reg.vals1[i, "r.square"] = tmp.summary$r.squared
219 }
220 reg.vals1
221
222 # print regression results
223
224 par(mfrow=c(1,2))
225 barplot(reg.vals1[,1], horiz=T, main="Beta values", col="blue", cex.names =
      0.75, las=1)
226 barplot(reg.vals1[,3], horiz=T, main="R-square values", col="blue",
      cex.names = 0.75, las=1)
227 par(mfrow=c(1,1))
228
229 #####
230 # a Single Factor Model with Predictability : an illustration with TEXACO
231 #####
232
233 #Log Level linear model between LCITCRP_P TREND an MARKET
234 model_TEXACO_P=lm(berndt.df$LTEXACO_P ~ berndt.df$LMARKET_P+berndt.df$TREND)
235 summary(model_TEXACO_P)
236
237 ggplot(berndt.df, aes(x = date)) +
238   geom_line(aes(y = LTEXACO_P, color = "TEXACO"), size = 1, linetype =
      "solid") +
239   geom_line(aes(y = fitted(model_TEXACO_P), color = "Fitted"), size = 1,
      linetype = "solid") +
240   labs(x = "Date", y = "Actual and Fitted") +
241   ylim(-0.5, 2) +
242   theme_minimal() +
243   theme(
244     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
245     legend.title = element_blank(),
246     legend.text = element_text(size = 8),
247     axis.text = element_text(size = 8),
248     axis.title = element_text(size = 10),
249     plot.title = element_text(size = 12, hjust = 0.5)
250   ) +
251   scale_color_manual(
252     values = c("blue", "green"),
253     labels = c("TEXACO", "Fitted")
254   )
255
256
257 #store log level residuals as u and test for their stationarity
258 u_TEXACO=as.matrix(model_TEXACO_P$residuals)
259 DuTEXACO=diff(u_TEXACO, lag=1)
260 model_DuTEXACO=lm(DuTEXACO ~ u_TEXACO[1:(nrow(u_TEXACO)-1)]-1)

```

```
261 summary(model_DuTEXACO)
262 D12uTEXACO=diff(u_TEXACO,lag=12)
263 model_D12uTEXACO=lm(D12uTEXACO ~ u_TEXACO[1:(nrow(u_TEXACO)-12)]-1)
264 summary(model_D12uTEXACO)
265 #####
266
267 #Compute Log Returns for 1M ahead timespan (1 months)
268 logret1M_TEXACO=diff(berndt.df$LTEXACO_P,lag=1)
269 logret1M_MARKET=diff(berndt.df$LMARKET_P,lag=1)
270
271 #Compute Log Returns for 1Y ahead timespan (12 months)
272 logret1Y_TEXACO=diff(berndt.df$LTEXACO_P,lag=12)
273 logret1Y_MARKET=diff(berndt.df$LMARKET_P,lag=12)
274
275 #####
276
277 #model the regression on log returns appending the u residuals as another
    variable
278 model_d_TEXACO_1M=lm(logret1M_TEXACO ~
    logret1M_MARKET+u_TEXACO[1:(nrow(u_TEXACO)-1)])
279 summary(model_d_TEXACO_1M)
280
281 model_d_TEXACO_1Y=lm(logret1Y_TEXACO ~
    logret1Y_MARKET+u_TEXACO[1:(nrow(u_TEXACO)-12)])
282 summary(model_d_TEXACO_1Y)
283
284 model_d_TEXACO_1Y_CAPM=lm(logret1Y_TEXACO ~ logret1Y_MARKET)
285 summary(model_d_TEXACO_1Y_CAPM)
```

Chapter 5

Models for Risk Measurement

5.1 Risk Measurement

Once the portfolio weights ($\hat{\mathbf{w}}$) are chosen, possibly exploiting the predictability of the distribution of the relevant future returns, the distribution of portfolio returns can be described as follows:

$$\begin{aligned} R^p &\sim \mathcal{D}(\mu_p, \sigma_p^2) \\ \mu_p &= \boldsymbol{\mu}'\hat{\mathbf{w}} \quad \sigma_p^2 = \hat{\mathbf{w}}'\boldsymbol{\Sigma}\hat{\mathbf{w}} \end{aligned}$$

Having solved the portfolio problem and having committed to a given allocation described by $\hat{\mathbf{w}}$, there is a further role that econometrics can play : measuring portfolio risk [Christoffersen \(2011\)](#). Note that even if portfolio weights can be decided at low frequency with a horizon of, say, one or more years, risk is run at high frequency and therefore what matters for risk measurement is the predictive distribution of returns at high frequencies. The question "What is the risk of my portfolio tomorrow?" is relevant even if the portfolio is built with a ten-year perspective

5.1.1 Value at Risk (VaR)

A natural measure of risk is Value at Risk (VaR) . The VaR is the percentage loss obtained with a probability at most of α per cent:

$$\Pr(R_{t+1}^p < -VaR_\alpha) = \alpha.$$

VaR depends on the predictive distribution of returns at high frequency, once α is chosen, VaR_α is defined by the predictive distribution of returns

5.2 VaR without predictability

We start our discussion of risk measurement by illustrating how the standard models used so far, which imply no predictability for the distribution of returns, can be used to compute VaR.

5.2.1 VaR with the CER

Applying the CER model to the univariate distribution of portfolio returns, we have

$$\begin{aligned} \mathbf{r}_{t,t+1}^P &= \mu + \sigma \epsilon_{t+1} \\ \epsilon_{t+1} &\sim \mathcal{D}(0, 1) \end{aligned}$$

Given some estimates of the unknown parameters in the model (μ σ in our case), the distribution of returns at $t + 1$ (say tomorrow) can be simulated either by making an assumption on the distribution of $\hat{\epsilon}_{t+1}$ and resampling from it (Monte-Carlo), or by re-sampling from the estimated residuals of the model (Bootstrap). In both cases an artificial sample for $\hat{\epsilon}_{t+1}$ of the desired length can be generated. Simulated residuals are then mapped into simulated returns via the model's parameters. This exercise can be replicated N times to construct the distribution of model-predicted returns. Once the distribution is derived, then VaR is available

5.2.2 VaR with the CAPM:

Factor models can also be simulated to derive VaR. Suppose you are invested in a specific portfolio and apply the CAPM to derive the distribution of its future returns

$$\begin{aligned} R_{t+1}^{Port} &= \gamma_0 + \gamma_1 R_{t+1}^{Mkt} + \sigma^{Port} v_{1,t+1} \\ R_{t+1}^{Mkt} &= \mu + \sigma^{Mkt} z_{t+1} \\ v_{i,t+1} &\sim \text{IID } \mathcal{N}(0, 1), \\ z_{t+1} &\sim \text{IID } \mathcal{N}(0, 1), \end{aligned}$$

After parameters' estimation, get residuals for a training sample. Then, at each point in time after your training sample generate an artificial sample for the residuals. The model has two residuals: capturing the common risk component and the idiosyncratic risk component. By their nature these two residuals can be simulated independently, drawing them independently from their marginal distribution rather than drawing them simultaneously from their joint distribution. Simulated residuals can then be

mapped into simulated returns via the model, to construct the distribution of model-predicted returns and derive the VaR for the portfolio. Note that both in the case of the VaR with CER and the VaR with CAPM the absence of predictability will imply that the VaR is constant over time. A model with no predictability rules out variability and/or persistence in the VaR measures.

5.3 The Evidence from high-frequency data

Figure 5.1 illustrated the behaviour of one-day returns and squared returns for the SP500. Data at high-frequency show:

- very little or no persistence in the first moments
- persistence in the variance
- non-normality
- Volatility “clusters” in time: high (low) volatility tends to be followed by high (low) volatility

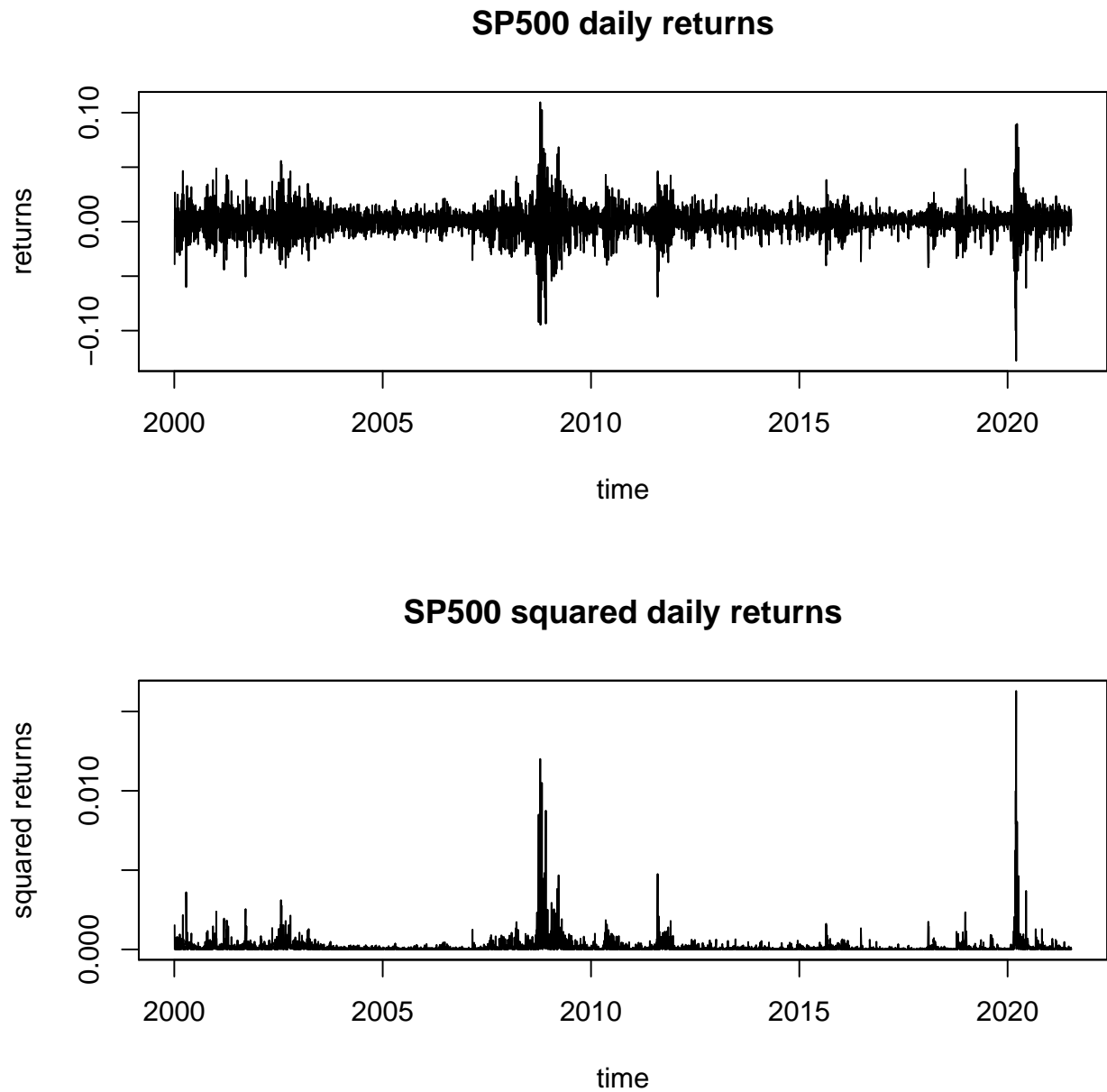


FIGURE 5.1. Daily SP500 Returns and Squared Returns

These features of the data can be used to build appropriate models with predictability in the distribution of future returns driven by the predictability in the second moments and use them to construct time-varying measures of VaR.

5.4 A general model for high-frequency data

The data at high frequency suggest a different modelling framework from the standard models with no-predictability :

$$\begin{aligned} R_{t+1} &= \sigma_{t+1} u_{t+1} \\ \sigma_{t+1}^2 &= f(\mathcal{I}_t) \quad u_{t+1} \sim IID \mathcal{D}(0, 1). \end{aligned}$$

The following features of the model are noteworthy:

1. The distribution of returns is centered around a mean of zero, and the zero mean model dominates any alternative model based on predictors.
2. The variance is time-varying and predictable, given the information set, \mathcal{I}_t , available at time t .
3. The distribution of returns at high frequency is not normal, i.e., $\mathcal{D}(0, 1)$ may often differ from $\mathcal{N}(0, 1)$

5.4.1 GARCH Models for Heteroscedasticity.

Generalizing the seminal contribution of modelling time-varying volatility by [Engle \(1982\)](#), [Bollerslev \(1986\)](#) proposed a parsimonious model capable of capturing all the features of high-frequency returns:

$$\begin{aligned} R_{t+1} &= \mu_t + \sigma_{t+1} z_{t+1} \quad z_{t+1} \sim IID \mathcal{N}(0, 1), \\ \sigma_{t+1}^2 &= \omega + \alpha (R_t - \mu_t)^2 + \beta \sigma_t^2 \\ \alpha + \beta &< 1 \end{aligned}$$

where returns have a constant mean (that is usually zero) and a time varying GARCH(1,1) structure.

In a model like this the innovation $\epsilon_t \equiv \sigma_t z_t$ has zero mean and is serially uncorrelated at all lags $j \geq 1$. Where μ_t is often, but not necessarily, set to 0.

5.4.2 GARCH Properties

R_{t+1} has a finite unconditional long-run variance of $\frac{\omega}{1-\alpha-\beta}$

$$\begin{aligned}
\sigma^2 &= E(\sigma_{t+1}^2) = \omega + \alpha E(R_t - \mu)^2 + \beta \sigma^2 \\
&= \omega + \alpha \sigma^2 + \beta \sigma^2 \\
&= \frac{\omega}{1 - \alpha - \beta}
\end{aligned}$$

Substituting ω out of the GARCH expression:

$$\begin{aligned}
\sigma_{t+1}^2 &= (1 - \alpha - \beta) \sigma^2 + \alpha R_t^2 + \beta \sigma_t^2 \\
&= \sigma^2 + \alpha ((R_t - \mu)^2 - \sigma^2) + \beta (\sigma_t^2 - \sigma^2)
\end{aligned}$$

which illustrates the relation between predicted variance and long-run variance in a GARCH model.

5.4.3 GARCH Forecasting

$$\begin{aligned}
\sigma_{t+1|t}^2 &= \sigma^2 + \alpha [(R_t - \mu_t)^2 - \sigma^2] + \beta (\sigma_t^2 - \bar{\sigma}^2), \\
\sigma_{t+2|t}^2 &= \sigma^2 + (\alpha + \beta) \sigma_{t+1|t}^2 \\
\sigma_{t+n+1|t}^2 &= \sigma^2 + (\alpha + \beta)^n \sigma_{t+1|t}^2
\end{aligned}$$

5.4.4 Testing for GARCH

The presence of a (G)ARCH in returns/disturbances can be tested via the Lagrange multiplier test proposed by Engle (1982) the test is implemented the following two steps: First, use simple OLS to estimate the most appropriate regression equation or ARMA model on asset returns and let $\{\hat{z}_t^2\}$ denote the squares of the standardized returns (residuals), for instance, coming from a homoskedastic model, $\hat{z}_t^2 = R_t^2 / \hat{\sigma}^2$; Second, regress these squared residuals on a constant and on q lagged values $\hat{z}_{t-1}^2, \hat{z}_{t+2}^2, \dots, \hat{z}_{t-q}^2$ (e_t is a white noise shock):

$$\hat{z}_t^2 = \xi_0 + \xi_1 \hat{z}_{t-1}^2 + \xi_2 \hat{z}_{t-2}^2 + \dots + \xi_q \hat{z}_{t-q}^2 + e_t.$$

If there are no ARCH effects, the estimated values of ξ_1 through ξ_q should be zero, $\xi_1 = \xi_2 = \dots = \xi_q$.

5.5 Estimation of GARCH Models

Standard OLS estimation cannot be applied to GARCH models as σ_{t+1} is not observed. Maximum Likelihood methods are necessary in this case. These methods are promptly available in R and we shall describe their working in a simple case. Think of the following Data Generating Process for returns

$$\begin{aligned} R_{t+1} &= \sigma_{t+1} z_{t+1} & z_{t+1} &\sim \text{IID } \mathcal{N}(0, 1), \\ \sigma_{t+1}^2 &= \omega + \alpha R_t^2 + \beta \sigma_t^2 \\ \alpha + \beta &< 1 \end{aligned}$$

The assumption of IID normal shocks (z_t), implies (from normality and identical distribution of z_{t+1}) that the density of the time t observation is:

$$l_t \equiv \Pr(R_t; \boldsymbol{\theta}) = \frac{1}{\sigma_t(\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{R_t^2}{\sigma_t^2(\boldsymbol{\theta})}\right),$$

where the notation $\sigma_t^2(\boldsymbol{\theta})$ emphasizes that conditional variance depends on $\boldsymbol{\theta} \in \Theta$, $\boldsymbol{\theta} = (\alpha, \beta, \omega)$.

Because each shock is independent of the others (from independence over time of z_{t+1}), the total probability density function (PDF) of the entire sample is then the product of T such densities:

$$L(R_1, R_2, \dots, R_T; \boldsymbol{\theta}) \equiv \prod_{t=1}^T l_t = \prod_{t=1}^T \frac{1}{\sigma_t(\boldsymbol{\theta})\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{R_t^2}{\sigma_t^2(\boldsymbol{\theta})}\right).$$

taking logs

$$\mathcal{L}(R_1, R_2, \dots, R_T; \boldsymbol{\theta}) = -\frac{T}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^T \log \sigma_t^2(\boldsymbol{\theta}) - \frac{1}{2} \sum_{t=1}^T \frac{R_t^2}{\sigma_t^2(\boldsymbol{\theta})}$$

Substituting an expression for $\sigma_t^2(\boldsymbol{\theta})$ (given by the chosen GARCH specification) given the observations on the returns and given an initial observation for variance

$$\begin{aligned} \mathcal{L}(R_1, R_2, \dots, R_T; \boldsymbol{\theta}) &= -\frac{T}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^T \log [\omega + \alpha R_{t-1}^2 + \beta \sigma_{t-1}^2] \\ &\quad - \frac{1}{2} \sum_{t=1}^T \frac{R_t^2}{\omega + \alpha R_{t-1}^2 + \beta \sigma_{t-1}^2}, \\ \sigma_0^2 &= \frac{\omega}{1 - \alpha - \beta} \end{aligned}$$

maximizing the log-likelihood to select the unknown parameters will deliver the MLE, denoted as $\hat{\boldsymbol{\theta}}_T^{ML}$

5.5.1 Quasi MLE Estimation

The QMLE result says that we can still use MLE estimation *based on normality assumptions* even when the shocks are not normally distributed, if our choices of conditional mean and variance functions are defensible, at least in empirical terms (i.e. conditional mean and conditional variance are correctly specified). However, because the maintained model still has that $R_{t+1} = \sigma_{t+1} z_{t+1}$ with $z_{t+1} \sim \text{IID } \mathcal{D}(0, 1)$, the shocks will have to be anyway IID: you can just do without normality, but the convenience of $z_{t+1} \sim \text{IID } \mathcal{D}(0, 1)$ To illustrate QMLE consider the following example.

Because we know that the long-run (ergodic) variance from a GARCH(1,1) is $\bar{\sigma}^2 = \omega / (1 - \alpha - \beta)$, instead of jointly estimating ω , α , and β , you simply set

$$\tilde{\omega} = (1 - \alpha - \beta) \left[\frac{1}{T} \sum_{t=1}^T R_t^2 \right]$$

for whatever values of α and β . Note that (i) you impose the long-run variance estimate on the GARCH model directly and avoid that the model may yield nonsensical estimates; (ii) you have reduced the number of parameters to be estimated in the model by one. These benefits must be carefully contrasted with the well-known costs, the loss of efficiency caused by QMLE.

5.6 From GARCH to VaR

After estimation a GARCH model can be simulated using bootstrap or Monte-Carlo to derive the distribution of returns and the relevant VaR

$$\begin{aligned}
R_{t+1} &= \mu + \sigma_{t+1}z_{t+1} & z_{t+1} &\sim \text{IID } \mathcal{N}(0, 1), \\
\sigma_{t+1}^2 &= \omega + \alpha (R_t - \mu_t)^2 + \beta\sigma_t^2 \\
\alpha + \beta &< 1
\end{aligned}$$

Given estimation, derive $\hat{z}_t = \frac{R_t}{\hat{\sigma}_t}$. At time t you can now predict σ_{t+1}^2 and the distribution of R_{t+1} can now be simulated via the preferred method.

Recursion can then be applied to derive the distribution of R_{t+n} with $n > 1$.

5.6.1 GARCH with factors

Think of modelling the returns of many assets at a high frequency with a (single) factor model

$$\begin{aligned}
R_{t+1}^i &= \gamma_0 + \gamma_1 f_{t+1} + \sigma^i v_{i,t+1} \\
f_{t+1} &= \mu_t + \sigma_{t+1}z_{t+1} \\
\sigma_{t+1}^2 &= \omega + \alpha (R_t - \mu_t)^2 + \beta\sigma_t^2 \\
v_{i,t+1} &\sim \text{IID } \mathcal{N}(0, 1), \\
z_{t+1} &\sim \text{IID } \mathcal{N}(0, 1), \\
\alpha + \beta &< 1
\end{aligned}$$

one GARCH estimation will allow to model many returns distribution. Again factor models allow parsimonious representation.

5.7 Measuring risk: an illustration with R.

The following programme illustrates how to construct VaR in models with and without predictability. A data set on monthly returns on the Dow Jones index and Bank of America is extracted to estimate the model for a training sample up to December 2005 and the to compute one-step ahead Var over the period 2006:1 2015:1. The Var is computed using the CAPM model with a CER for the market and the CAPM model with a GARCH for the market. While the first measure of risk is constant through the sample the second one reflects the predictability of the volatility fitted with a GARCH(1,1).

```

1 rm(list=ls())
2 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
3 # packages used

```

```

4 listofpackages <- c("tidyverse","dygraphs", "rugarch",
  "forecast","dplyr","ellipse","reshape2","ggplot2","xts","xlsx","readxl")
5
6 for (j in listofpackages){
7   if(sum(installed.packages()[, 1] == j) == 0) {
8     install.packages(j)
9   }
10  library(j, character.only = T)
11 }
12
13 # setting the seed for replication
14 set.seed(77)
15
16 raw_data      = read_xlsx("../data/2023_monthly_stocks.xlsx")
17 names(raw_data)[1] = 'Date'
18 typeof(raw_data)
19 typeof(raw_data$Date)
20 typeof(raw_data$AXP)
21 typeof(raw_data$CSC0)
22
23 dates <-seq(as.Date("1985-02-01"),length=462, by="months")
24 params <- c("Date","BA", "DJI")
25 data <- raw_data[, c(params)]
26 data<- na.omit(data)
27 data <- data %>%
28   mutate(Date = as.Date(Date, format = "%Y-%m-%d"))
29
30 params1 <- c("BA", "DJI")
31 tsdata <- xts(raw_data[, c(params1)], order.by=dates) # creates a time
  series object
32 tsdata <- na.omit(tsdata) # omitting the rows with NA presence
33 data<- na.omit(data)
34 ## having created the database with all observation we generate a subset
35 #tsdata1 <- tsdata["1992-02-01/1993-02-01"]
36 #data=subset(data,select=c(1:12))
37
38 ## -----
39 # DATA TRANSFORMATIONS
40 ## -----
41 #1. from prices to returns
42 # exact monthly returns
43 t1<-nrow(data)
44 data$BA_ret <- data$DJI_ret <- array(data = NA, dim = t1)
45 for (i in 2:t1) {
46   data[i, "BA_ret"][[1]]=(data[i, "BA"][[1]]-data[i-1,
  "BA"][[1]])/data[i-1, "BA"][[1]]
47   data[i, "DJI_ret"][[1]]=(data[i, "DJI"][[1]]-data[i-1,
  "DJI"][[1]])/data[i-1, "DJI"][[1]]
48 }
49
50
51 # same in .xts

```

```

52 t1<-nrow(tpdata)
53
54 tpdata$BA_ret <- tpdata$DJI_ret<- array(data = NA, dim = t1)
55
56 for (i in 2:t1) {
57   tpdata[i, "BA_ret"][[1]]=(tpdata[i, "BA"][[1]]-tpdata[i-1,
58     "BA"][[1]])/data[i-1, "BA"][[1]]
59   tpdata[i, "DJI_ret"][[1]]=(tpdata[i, "DJI"][[1]]-tpdata[i-1,
60     "DJI"][[1]])/data[i-1, "DJI"][[1]]
61 }
62 ## -----
63 ## VAR with CER-CAPM
64 ## -----
65
66
67 ## -----
68 ## MODEL SPECIFICATION AND ESTIMATION
69 ## -----
70 start_date <- as.Date("1992-03-01") # Replace with your start date
71 end_date <- as.Date("2005-12-01") # Replace with your end date
72
73 # Extract observations between 'start_date' and 'end_date'
74
75 data_est <- subset(x = data, Date >= start_date & Date <= end_date)
76
77 # estimation
78 cer_mkt <- lm(data_est$DJI_ret ~ 1)
79 capm_BA <- lm(data_est$BA_ret ~ data_est$DJI_ret)
80 summary(cer_mkt)
81 summary(capm_BA)
82
83 ## -----
84 ## MODEL SIMULATION
85 ## -----
86
87 tt <- as.Date("2006-01-01")
88 tT <- as.Date("2015-12-01")
89 data_sim <- subset(x = data, Date >= tt & Date <= tT)
90
91 # creating the containers
92 nrep <- 1000
93 BA_bt_2 <- mkt_bt_2 <- array(0, c(length(data_sim$DJI_ret), nrep))
94
95 # resampling the residuals
96 res_mkt_bt_2 <- matrix(sample(resid(cer_mkt), size =
97   length(data_sim$DJI_ret) * nrep, replace = T),
98   nrow = length(data_sim$DJI_ret), ncol = nrep)
99 res_BA_bt_2 <- matrix(sample(resid(capm_BA), size =
100   length(data_sim$DJI_ret) * nrep, replace = T),
101   nrow = length(data_sim$DJI_ret), ncol = nrep)

```

```

100
101
102 # the loop
103 for (i in 1:nrep){
104   for (j in 1:length(data_sim$DJI_ret)){
105     mkt_bt_2[j, i] <- coef(cer_mkt)[1] + res_mkt_bt_2[j, i]
106     BA_bt_2[j, i] <- coef(capm_BA)[1] + coef(capm_BA)[2] * mkt_bt_2[j, i] +
       res_BA_bt_2[j, i]
107   }
108 }
109
110 # the quantiles
111 var_BA_capm <- array(0, length(data_sim$DJI_ret))
112 for (j in 1:length(data_sim$DJI_ret)){
113   var_BA_capm[j] <- quantile(BA_bt_2[j, ], probs = 0.05)
114 }
115 data_sim$var_BA_capm<-var_BA_capm
116 # plotting
117 ggplot(data_sim, aes(x = Date)) +
118   geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
119   geom_line(aes(y = var_BA_capm, color = "VaR"), size = 1, linetype =
       "solid") +
120   labs(x = "Date", y = "Returns and VaR") +
121   ylim(-0.15, 0.15) +
122   theme_minimal() +
123   theme(
124     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
125     legend.title = element_blank(),
126     legend.text = element_text(size = 8),
127     axis.text = element_text(size = 8),
128     axis.title = element_text(size = 10),
129     plot.title = element_text(size = 12, hjust = 0.5)
130   ) +
131   scale_color_manual(
132     values = c("blue", "green"),
133     labels = c("BA", "VaR")
134   )
135
136
137
138 ## -----
139 ## GARCH MODELLING
140 ## -----
141
142 # the market GARCH regression
143 ## specification
144 mkt_garch <- ugarchspec(variance.model = list(garchOrder = c(1, 1)),
145                        mean.model = list(armaOrder = c(0, 0)))
146 ## estimation
147 mkt_garchfit <- ugarchfit(mkt_garch, data = data_est$DJI_ret)
148 mkt_garchfit
149

```

```

150
151 # forecasting and plotting the results
152 horizon <- 10*12 # ten years
153 mygarchforecast <- ugarchforecast(mkt_garchfit, n.ahead = 10*12)
154
155 plotdata <- cbind(mygarchforecast@forecast$seriesFor,
156                 mygarchforecast@forecast$seriesFor +
157                 mygarchforecast@forecast$sigmaFor*1.96,
158                 mygarchforecast@forecast$seriesFor -
159                 mygarchforecast@forecast$sigmaFor*1.96)
158 colnames(plotdata) <- c("mean", "upper", "lower")
159 dygraph(ts(plotdata, start = c(2006,1), frequency = 12), main = "Forecast
160         of the mean") %>%
161     dySeries(c("lower", "mean", "upper"))
162
162 plotdata2 <- as.matrix(mygarchforecast@forecast$sigmaFor^2)
163 colnames(plotdata2) <- "var"
164 dygraph(ts(plotdata2, start = c(2006, 1), frequency = 12), main = "Forecast
165         of the variance")
166
167 ## -----
168 ## GARCH SIMULATION
169 ## -----
170
171 ## coefficients
172 gamma0 <- coef(mkt_garchfit)[1]
173 omega0 <- coef(mkt_garchfit)[2]
174 omega1 <- coef(mkt_garchfit)[3]
175 omega2 <- coef(mkt_garchfit)[4]
176 sigma2 <- sigma(mkt_garchfit) # this constructs the series of standard
177     deviations conditional on information at "t-1". Is thus a vector.
178
178 # the CAPM
179 ## estimation
180 capm_BA <- lm(data_est$BA_ret ~ data_est$DJI_ret)
181 summary(capm_BA)
182
183 beta0 <- coef(capm_BA)[1]
184 beta1 <- coef(capm_BA)[2]
185
186
187
188 ## -----
189 # simulation
190 # output containers
191 nrep <- 1000
192 BA_bt <- mkt_bt<- sigma<- array(0, c(length(data_sim$DJI_ret), nrep))
193
194 # extracting the errors and resampling
195
196 res_mkt <- as.numeric(residuals(mkt_garchfit, standardize = T)) # the

```

```

    standardized residuals from the market equation
197
198 res_mkt_bt <- matrix(sample(res_mkt, size = length(data_sim$DJI_ret) *
    nrep, replace = T),
199                       nrow = length(data_sim$DJI_ret), ncol = nrep)
200 res_BA_bt <- matrix(sample(resid(capm_BA), size = length(data_sim$DJI_ret)
    * nrep, replace = T),
201                       nrow = length(data_sim$DJI_ret), ncol = nrep)
202
203 # initial values
204 mkt_bt[1, ] <- data_sim$DJI_ret[1]
205 BA_bt[1, ] <- beta0 + beta1*mkt_bt[1, ]
206 sigma[1, ] <- ugarchforecast(mkt_garchfit)@forecast$sigmaFor[1] #takes the
    first value (the one step ahead)
207 # the loop
208 for (i in 1:nrep){
209   for (j in 2:length(data_sim$DJI_ret)){
210     sigma[j, i] <- sqrt(omega0+omega1*( data_sim$DJI_ret[j-1]- gamma0)^2 +
        omega2*(sigma[j-1, i])^2)
211     mkt_bt[j, i] <- gamma0 + res_mkt_bt[j, i] * sigma[j,i]
212     BA_bt[j, i] <- beta0 + beta1 * mkt_bt[j, i] + res_BA_bt[j, i]
213   }
214 }
215
216 # getting the quantiles
217 var_BA_garch <- array(0, length(data_sim$DJI_ret))
218 for (j in 1:length(data_sim$DJI_ret)){
219   var_BA_garch[j] <- quantile(BA_bt[j, ], probs = 0.05)
220 }
221
222 data_sim$var_BA_garch<-var_BA_garch
223
224 # plotting
225 tt <- as.Date("2006-02-01")
226 tT <- as.Date("2015-12-01")
227 data_simplot <- subset(x = data_sim, Date >= tt & Date <= tT)
228 ggplot(data_simplot, aes(x = Date)) +
229   geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
230   geom_line(aes(y = var_BA_capm, color = "Var CAPM"), size = 1, linetype =
    "solid") +
231   geom_line(aes(y = var_BA_garch, color = "VaR GARCH"), size = 1, linetype
    = "solid") +
232   labs(x = "Date", y = "Returns and VaR") +
233   ylim(-0.30, 0.20) +
234   theme_minimal() +
235   theme(
236     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
237     legend.title = element_blank(),
238     legend.text = element_text(size = 8),
239     axis.text = element_text(size = 8),
240     axis.title = element_text(size = 10),
241     plot.title = element_text(size = 12, hjust = 0.5)

```

```

242 ) +
243 scale_color_manual(
244   values = c("blue", "green", "red"),
245   labels = c("BA", "VaR CAPM", "Var GARCH")
246 )
247
248 save(data_simplot, file="VaRdata.Rdata")

```

5.8 Backtesting VaR

How do we test the validity of a VaR model ? The relevant evidence to judge a VaR model are violations:

$$\text{Min}(R_{t+1} - \text{VaR}_{t+1}^p, 0)$$

- (a) A good VaR model should not feature neither too few nor too many violations.
- (b) We have too few violations when a VaR at the confidence level of alpha shows less than $100 \cdot \alpha$ violations in a sample of 100 observations. In this case, the VaR model is too conservative.
- (c) when we have violations there are two interesting aspects of that: their number and their timing. A five per cent VaR that features 5 violations in five successive periods cannot be taken as a valid VaR model as violations are not independent. Clustering of violations is a problem that should lead to reject specific VaR models. [Kupiec \(2002\)](#) proposed a formal test of VaR validity based on these two aspects.

5.8.1 Unconditional Coverage Testing

Given a time-series of VaR and observed returns the "hit sequence" of VaR violations is defined as follows:

$$\begin{aligned}
 I_{t+1} &= 1, \text{ if } R_{t+1} > \text{VaR}_{t+1}^p \\
 I_{t+1} &= 0, \text{ if } R_{t+1} \leq \text{VaR}_{t+1}^p
 \end{aligned}$$

If the VaR is a valid model violations should not be predictable: the probability of a VaR violation should be p every day. The hit sequence in this case should be distributed over time as a Bernoulli variable that takes the value 1 with probability p and the value 0 with probability $1 - p$. So

$$\begin{aligned}
 H_0 & : I_{t+1} \sim i.i.d. \text{ Bernoulli}(p) \\
 f(I_{t+1}, p) & = (1-p)^{1-I_{t+1}} p^{I_{t+1}}
 \end{aligned}$$

The first test of the validity of a VaR is therefore constructed as follows. Take a Bernoulli distribution (I_{t+1}, x) for the that the number of violations, derive a maximum likelihood estimator \hat{x} of x , and test using a likelihood ratio test that \hat{x} is not statistically different from p .

$$\begin{aligned}
 L(I_{t+1}, x) & = \prod_{i=1}^T (1-x)^{1-I_{t+1}} x^{I_{t+1}} \\
 & = (1-x)^{T_0} x^{T_1}
 \end{aligned}$$

where T_1 is the number of violations of the VaR observed in the sample, and $T_0 = T - T_1$.

The maximum likelihood estimator $\hat{x} = \frac{T_1}{T}$.

A likelihood ratio test of the null hypothesis $\hat{x} = p$, can then be constructed as follows:

$$LR_{uc} = -2 \ln \left[\frac{L(p)}{L(\hat{x})} \right]$$

which is distributed as a χ^2 with one degree of freedom.

Note that usually the number of violations and the number of observations available will not be large, so rather than relying upon the χ^2 distribution, it is advisable to use Monte-Carlo simulations to build the relevant distribution to conduct the test. In this case the simulated P-values would be obtained by drawing an artificial sample of the relevant size from the null, and using as a P-value the share of simulated test that are larger than the observed ones.

5.8.2 Independence Testing

We concentrate now on a test able to reject a VaR with clustered violations. In this case the hit sequence is dependent over time and its evolution over time can be described by a so-called Markov sequence where the transition from the relevant states (violation and no violation) can be described by the following transition probability matrix

$$X_1 = \begin{bmatrix} x_{00} & 1 - x_{00} \\ 1 - x_{11} & x_{11} \end{bmatrix}$$

where:

$$\begin{aligned} x_{00} &= \Pr(I_{t+1} = 0 \mid I_t = 0) \\ 1 - x_{00} &= \Pr(I_{t+1} = 1 \mid I_t = 0) \\ x_{11} &= \Pr(I_{t+1} = 1 \mid I_t = 1) \\ 1 - x_{11} &= \Pr(I_{t+1} = 0 \mid I_t = 1) \end{aligned}$$

If we observe a sample of T observations the likelihood function of the first order Markov process can be written as follows:

$$L(X_1, I_{t+1}) = x_{00}^{T_{00}} (1 - x_{00})^{T_{01}} (1 - x_{11})^{T_{10}} x_{11}^{T_{11}}$$

The maximum likelihood estimates of the relevant parameters are then

$$\begin{aligned} \hat{x}_{00} &= \frac{T_{00}}{T_{00} + T_{01}} \\ \hat{x}_{11} &= \frac{T_{11}}{T_{10} + T_{11}} \end{aligned}$$

and so

$$\hat{X}_1 = \begin{bmatrix} \frac{T_{00}}{T_{00} + T_{01}} & \frac{T_{01}}{T_{00} + T_{01}} \\ \frac{T_{10}}{T_{10} + T_{11}} & \frac{T_{11}}{T_{10} + T_{11}} \end{bmatrix}$$

Independence Testing

Under independence

$$\hat{X}_1^{id} = \begin{bmatrix} 1 - \hat{x} & \hat{x} \\ 1 - \hat{x} & \hat{x} \end{bmatrix}$$

and therefore the independence hypothesis $(1 - \hat{x}_{00}) = \hat{x}_{11}$ can be tested using a likelihood ratio test

$$LR_{ind} = -2 \ln \left[\frac{L(\hat{X}_1^{id})}{L(\hat{X}_1)} \right] \sim \chi_1^2$$

As for the unconditional coverage test, small sample problems can be fixed by Monte Carlo simulation of the critical values, moreover samples in which $T_{11} = 0$ are often observed. In this cases, the likelihood function is computed as

$$L(X_1, I_{t+1}) = x_{00}^{T_{00}} (1 - x_{00})^{T_{01}}$$

5.8.3 Conditional Coverage Testing

Conditional Coverage Testing

Having constructed the test for independence we can test jointly the hypothesis of conditional coverage and independence via the following likelihood ratio test:

$$LR_{cc} = -2 \ln \left[\frac{L(p)}{L(\hat{X}_1)} \right] \sim \chi_2^2$$

note that

$$LR_{cc} = LR_{uc} + LR_{ind}$$

5.8.4 Backtesting VaR in R

The following programme implements the [Kupiec \(2002\)](#) test on the Var measures derived in Section 7.

```

1 rm(list=ls())
2 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
3 # packages used
4 listofpackages <- c("tidyverse", "dygraphs", "rugarch",
5   "forecast", "dplyr", "ellipse", "reshape2", "ggplot2", "xts", "xlsx", "readxl")
6
7 for (j in listofpackages){
8   if(sum(installed.packages()[, 1] == j) == 0) {
9     install.packages(j)
10    }
11  library(j, character.only = T)
12 }
13 # loading the databases
14 load("VaRdata.Rdata")
15 ggplot(data_simplot, aes(x = Date)) +
16   geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
17   geom_line(aes(y = var_BA_capm, color = "Var CAPM"), size = 1, linetype =
18     "solid") +

```

```

18   geom_line(aes(y = var_BA_garch, color = "VaR GARCH"), size = 1, linetype
      = "solid") +
19   labs(x = "Date", y = "Returns and VaR") +
20   ylim(-0.30, 0.20) +
21   theme_minimal() +
22   theme(
23     legend.position = c(0.15, 0.95), # Set the legend position (top-left)
24     legend.title = element_blank(),
25     legend.text = element_text(size = 8),
26     axis.text = element_text(size = 8),
27     axis.title = element_text(size = 10),
28     plot.title = element_text(size = 12, hjust = 0.5)
29   ) +
30   scale_color_manual(
31     values = c("blue", "green", "red"),
32     labels = c("BA", "VaR CAPM", "Var GARCH")
33   )
34   ## -----
35   # VaR tail
36   alpha <- 0.1
37
38
39   ## -----
40   violations <- (data_simplot$BA_ret - data_simplot$var_BA_capm) < 0
41   table(violations)
42   plot(y = violations*runif(length(violations), min = 0.99, max = 1.01), x =
      data_simplot$Date, main = "VaR violations",
43     ylab = "violations", xlab = "time") # adding jitter to make sure that
      adjacent observations don't overlap
44
45   ## testing
46   ### Unconditional coverage
47   p <- alpha
48   T1 <- sum(violations)
49   T0 <- length(violations) - sum(violations)
50
51   x <- T1/(T1+T0) # violations as fraction of sample length, which is also
      the test statistic estimate
52
53   L_p <- (1-p)^T0 * p^T1
54   L_x <- (1-x)^T0 * x^T1
55
56   LR_uc <- -2 * log(L_p/L_x) # test statistic
57   critical <- qchisq(p = 0.95, df = 1) # the 5% critical value
58
59   LR_uc; critical
60   LR_uc > critical
61
62   ### Independence testing
63   temp1 <- abs(diff(violations)) # to identify moments of change
64   temp2 <- violations[2:length(violations)] # to identify the ending points
65   T01 <- sum(temp2 * temp1) # those that finish with 1 and had a change

```

```

66 T11 <- sum(temp2 * (1-temp1)) # those that finish with 1 and had no change
67 T10 <- sum((1 - temp2) * temp1) # those finishing with 0 and having a change
68 T00 <- sum((1-temp2) * (1-temp1)) # finishing with 0 and no change
69
70 xhat <- x # from before
71 x00 <- T00/(T00 + T01)
72 x11 <- T11/(T11 + T10)
73
74 L_x_different <- x00^T00 * (1-x00)^T01 * (1-x11)^T10 * x11^T11
75 L_x_equal <- (1-xhat)^T00 * xhat^T01 * (1-xhat)^T10 * xhat^T11
76
77 LR_ind <- -2 * log(L_x_equal/L_x_different)
78 critical <- qchisq(p = 0.95, df = 1) # the 5% critical value
79 LR_ind; critical
80 LR_ind > critical

```

5.9 Beyond GARCH: non-linear and multivariate models

GARCH models can be extended in many ways ([Christoffersen \(2011\)](#), [Zivot and Wang \(2006\)](#)) A number of empirical papers have emphasized that for many assets and sample periods, a negative return increases conditional variance by more than a positive return of the same magnitude does, the so-called *leverage effect*.

A way of capturing the leverage effect is to directly build a model that exploits the possibility to define an indicator variable, I_t , to take on the value 1 if on day t the return is negative and zero otherwise. For concreteness, in the simple (1,1) case, variance dynamics can now be specified as:

$$\sigma_{t+1}^2 = \omega + \alpha R_t^2 + \alpha \theta I_t R_t^2 + \beta \sigma_t^2 \quad I_t \equiv \begin{cases} 1 & \text{if } R_t < 0 \\ 0 & \text{if } R_t \geq 0 \end{cases} \quad \text{or}$$

$$\sigma_{t+1}^2 = \begin{cases} \omega + \alpha(1 + \theta)R_t^2 + \beta\sigma_t^2 & \text{if } R_t < 0 \\ \omega + \alpha R_t^2 + \beta\sigma_t^2 & \text{if } R_t \geq 0 \end{cases} .$$

A $\theta > 0$ will capture the leverage effect.

This model is sometimes referred to as the GJR-GARCH model—from [Glosten et al. \(1993\)](#) paper—or threshold GARCH (TGARCH) model.

In this model, because when 50% of the shocks are assumed to be negative and

the other 50% positive, so that $E[I_t] = 1/2$, the long-run variance equals:

$$\begin{aligned}\bar{\sigma}^2 &\equiv E[\sigma_{t+1}^2] = \omega + \alpha E[R_t^2] + \alpha\theta E[I_t R_t^2] + \beta E[\sigma_t^2] \\ &= \omega + \alpha\bar{\sigma}^2 + \alpha\theta E[I_t]\bar{\sigma}^2 + \beta\bar{\sigma}^2 \\ &= \omega + \alpha\bar{\sigma}^2 + \frac{1}{2}\alpha\theta\bar{\sigma}^2 + \beta\bar{\sigma}^2 \implies \bar{\sigma}^2 = \frac{\omega}{1 - \alpha(1 + 0.5\theta) - \beta}.\end{aligned}$$

Visibly, in this case the persistence index is $\alpha(1 + 0.5\theta) + \beta$

Another important dimension of extension of GARCH modelling is from the univariate to the multivariate framework, see, for example, chapter 13 in [Zivot and Wang \(2006\)](#). When considering multivariate volatility modelling an important aspect is the parsimonious parameterization, to this end a factor structure might again be helpful and the approach presented in this chapter of a factor structure in which the time-varying volatility is driven by the common risk component only while the idiosyncratic components are homoscedastic can prove very useful to handle portfolio allocation problems with many assets (large N) and few factors (small K).

Bibliography

- Ang, Andrew (2014) *Asset management: A systematic approach to factor investing*: Oxford University Press.
- Bollerslev, Tim (1986) “Generalized autoregressive conditional heteroskedasticity,” *Journal of econometrics*, 31 (3), 307–327.
- Campbell, John Y and Robert J Shiller (1987) “Cointegration and tests of present value models,” *Journal of political economy*, 95 (5), 1062–1088.
- Christoffersen, Peter (2011) *Elements of financial risk management*: Academic press.
- Cieslak, Anna, Adair Morse, and Annette Vissing-Jorgensen (2019) “Stock returns over the FOMC cycle,” *The Journal of Finance*, 74 (5), 2201–2248.
- Cochrane, John H (1999) “New facts in finance.”
- Engle, Robert F (1982) “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica: Journal of the econometric society*, 987–1007.
- Engle, Robert F and Clive WJ Granger (1987) “Co-integration and error correction: Representation, estimation, and testing,” *Econometrica*, 55 (2), 251–276.
- Engle, Robert F and Byung Sam Yoo (1987) “Forecasting and testing in co-integrated systems,” *Journal of Econometrics*, 35 (1), 143–159.
- Fama, Eugene F (1970) “Efficient capital markets: A review of theory and empirical work,” *The journal of Finance*, 25 (2), 383–417.
- Fama, Eugene F. and Kenneth R. French (1993) “Common risk factors in the returns on stocks and bonds,” *Journal of Financial Economics*, 33 (1), 3 – 56, [https://doi.org/10.1016/0304-405X\(93\)90023-5](https://doi.org/10.1016/0304-405X(93)90023-5).
- Fama, Eugene F and Kenneth R French (2015) “A five-factor asset pricing model,” *Journal of Financial Economics*, 116 (1), 1–22.

- Fama, Eugene F and James D MacBeth (1973) “Risk, return, and equilibrium: Empirical tests,” *Journal of political economy*, 81 (3), 607–636.
- Glosten, Lawrence R, Ravi Jagannathan, and David E Runkle (1993) “On the relation between the expected value and the volatility of the nominal excess return on stocks,” *The journal of finance*, 48 (5), 1779–1801.
- Hou, Kewei, Haitao Mo, Chen Xue, and Lu Zhang (2018) “Which factors?” *Review of Finance*, 23 (1), 1–35.
- Jegadeesh, Narasimhan and Sheridan Titman (2011) “Momentum,” *Annu. Rev. Financ. Econ.*, 3 (1), 493–509.
- Kupiec, Paul (2002) “Stress-testing in a Value at Risk Framework,” *Risk management: value at risk and beyond*, 10, 76–99.
- Lintner, John (1965) “Security prices, risk, and maximal gains from diversification,” *The Journal of Finance*, 20 (4), 587–615.
- Lucca, David O and Emanuel Moench (2015) “The pre-FOMC announcement drift,” *The Journal of finance*, 70 (1), 329–371.
- MacKinnon, James G (2010) “Critical values for cointegration tests,” Technical report, Queen’s Economics Department.
- Research, Milton Financial Market (2023) “The Complete Guide to Portfolio Optimization in R,” <https://miltonfmr.com/the-complete-guide-to-portfolio-optimization-in-r-part1/>, Accessed on 2023/09.
- Sharpe, William F (1964) “Capital asset prices: A theory of market equilibrium under conditions of risk,” *The Journal of finance*, 19 (3), 425–442.
- Shiller, Robert J (1979) “The volatility of long-term interest rates and expectations models of the term structure,” *Journal of political Economy*, 87 (6), 1190–1219.
- (1981) “Do stock prices move too much to be justified by subsequent changes in dividends?”
- Singh, Abhay Kumar and David Edmund Allen (2017) *R in finance and economics: A beginner’s guide*: World Scientific.
- Taleb, Nassim Nicholas (2012) *Antifragile: how to live in a world we don’t understand*, 3: Allen Lane London.

- Torfs, P and Claudia Brauer (2014) “A (very) short introduction to R,” <https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf>, Accessed on 2023/09.
- Valkanov, Rossen (2003) “Long-horizon regressions: theoretical results and applications,” *Journal of Financial Economics*, 68 (2), 201–232.
- Zivot, Eric and Jiahui Wang (2006) *Modeling financial time series with S-PLUS*, 2: Springer.

Index

- anomalies, 11
- asset allocation, 81
- backtesting, 74
 - VaR models, 119
- Chat GPT, 33–36, 40, 50, 53, 54, 56
- cointegration, 95
- duration, 31
- efficient frontier, 50
- efficient market hypothesis, 8
- Equilibrium Correction Term, 95
- estimation
 - Maximum Likelihood, 111
 - OLS, 66
 - Quasi Maximum Likelihood, 112
- factor exposures, 84
- factor models, 81
 - cross-sectional representation, 83
 - time-series representation, 82
 - validation, 92
- Factor Models with predictability, 93
- factor-based portfolios, 83
- look-ahead-bias, 74
- maturity, 31
- model
 - CAPM, 8
 - Constant Expected Return, 9
 - dynamic dividend growth, 29
 - Factor Error Correction (FECM), 95
 - GARCH, 109
 - GARCH with factors, 113
 - lognormal, 22
 - random walk, 61
 - Threshold GARCH, 124
- models for asset returns, 3
- portfolio
 - smart beta, 84
 - minimum variance, 5
 - risk parity, 5
 - tangency, 5
- predictability, 17
- R
 - Optimal Portfolio Allocation, 85
 - Backtesting VaR models, 122
 - data-handling, 37
 - dataframes, 36
 - datatypes, 36
 - estimation, 68
 - measuring risk, 113
 - model simulation, 74
 - models with predictability, 97
 - package fPortfolio, 68
- rational expectations, 17
- returns
 - continuously compounded, 21
 - coupon-bonds, 32
 - linear, 21
 - multi-period, 23
 - Stock and Bond, 26
 - Zero-Coupon Bonds, 30

Risk Measurement, 105

Sharpe ratio, 63

simulation, 67

 Bootstrap, 67

 Monte-Carlo, 67

time-diversification, 63

Value at Risk with factor models, 106

Value at risk with predictability, 109

Value at Risk without predictability, 106

Value-at-Risk, 10, 74, 105