Chapter 2

Empirical Models in Finance

2.1 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 the 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 $r^i_{t,t+k}$ 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.

2.2 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.

2.2.1 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 the interval $[t, t+k]$, $\mathbf{r} \sim \mathcal{D}(\mu, \Sigma)$, $\mathbf{w}$ is the vector of weights given to the N risky assets in the portfolio and $\mathbf{e}$ is a (Nx1) column vector of ones.

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

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

where $E[\mathbf{r}] = \left(1 - \mathbf{w}'\mathbf{e}\right) r^f + \mathbf{w}'\mu = r^f + \mathbf{w}'(\mu - r^f\mathbf{e})$ 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:

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

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:

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

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

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.

2.2.2 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:

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

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:

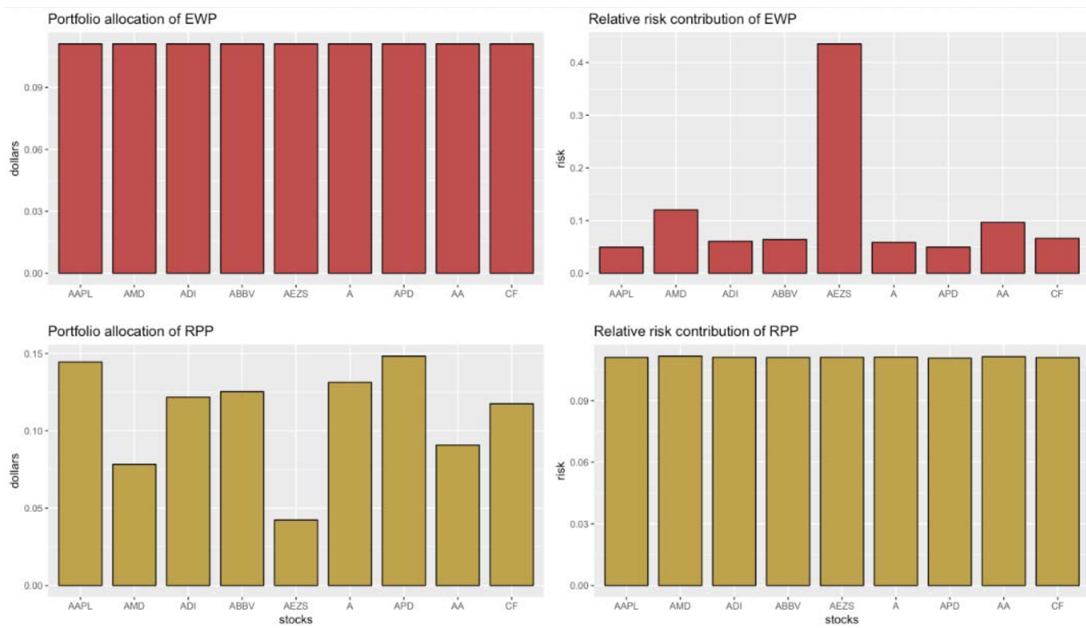

FIGURE 2.1. **Portfolio allocation in Equal Weights Portfolios(EWP) and Risk Parity Portfolios(RPP)**

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

2.3 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 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

2.3.1 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 assessing the importance of 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 simulation and forecasting, i.e. to estimate the unknown parameters, while the test sample is used for model evaluation,simulation and forecasting.

2.4 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.

2.4.1 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:

$$r^i_{t,t+1} = \mu^i + \sigma^i \epsilon_{it} \qquad \epsilon_{it} \sim NID\,(0,1)$$
$$Cov\,(\epsilon_{it}, \epsilon_{js}) = \begin{cases} \sigma_{ij} & t = s \\ 0 & t \neq s \end{cases} .$$

Note that the absence of predictability of excess returns is not a 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 Contant 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.

$$E(r^i_{t,t+n}) = E(\sum_{k=1}^{n} r^i_{t+k,t+k-1}) = \sum_{k=1}^{n} E(r^i_{t+k,t+k-1}) = n\mu$$

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

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:

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

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^i_t and the market return r^m_t,can be described as follows:

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

where r^{rf}_t is the return on the risk-free asset. $\sigma_{im} = 0$ 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:

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

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 $\alpha \in (0,1)$):

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

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

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

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*:

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

2.5 Empirical Challenges to the traditional model

Over 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

2.5.1 The time-series evidence

Practitioners implementing portfolio allocation based on the CER model experienced rather soon a number of problems that made evident 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:

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

where P_t is the stock price at time t and D_t is the dividend paid at time t, $p_t = \ln(P_t), d_t = \ln(D_t), \kappa$ 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:

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

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:

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

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:

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

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

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

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.

Table 2.1. 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

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.

2.5.2 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 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.

2.5.3 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:

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

A number of features of this model at high frequency is 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 $\mathcal{N}(0,1)$

2.5.4 The cross-section evidence

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

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

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.

2.6 The Implications of the new evidence

2.6.1 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:

$$\begin{aligned} P_{s,t} &= X_{i,t+1}\sum_{s=1}^{k}\pi_{t+1}(s)m_{t+1}(s) \\ 1 + R_{s,t+1} &= \frac{1}{\sum_{j=1}^{m}\pi_{t+1}(s)m_{t+1}(s)} \end{aligned}$$

In general, we can write:

$$\begin{aligned} P_{i,t} &= E_t\left(m_{t+1}X_{i,t+1}\right) \\ 1 + R_{s,t+1} &= \frac{1}{E_t\left(m_{t+1}\right)} \end{aligned}$$

consider now a risky asset :

$$\begin{aligned} E_t\left(m_{t+1}\left(1 + R_{i,t+1}\right)\right) &= 1 \\ Cov\left(m_{t+1}R_{i,t+1}\right) &= 1 - E_t\left(m_{t+1}\right)E_t\left(1 + R_{i,t+1}\right) \\ E_t\left(1 + R_{i,t+1}\right) &= -\frac{Cov\left(m_{t+1}R_{i,t+1}\right)}{E_t\left(m_{t+1}\right)} + \left(1 + R_{s,t+1}\right) \end{aligned}$$

Turning now to excess returns we can write:

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

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 on returns 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 discriminating between the possible sources of predictability.

2.7 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:

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

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:

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

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.

2.8 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:

$$\begin{aligned}
\mathbf{r}_{t,t+k} &= f\left(X_t^{\mu},\Theta_t^{\mu}\right)+\mathbf{H}_{t+k}\epsilon_{t+k} && (2.1)\\
\mathbf{\Sigma}_{t+k} &= \mathbf{H}_{t+k}\mathbf{H}_{t+k}^{\prime}. \\
\mathbf{\Sigma}_{t+k} &= g\left(X_t^{\sigma},\Theta_t^{\sigma}\right)+\sum\nolimits_{j=1}^{q}\mathbf{B}_j\mathbf{\Sigma}_{t+k-j}\mathbf{B}_j^{\prime}, && (2.2)\\
\epsilon_{t+k} &\sim \mathcal{D}\left(\mathbf{0},\mathbf{I}\right)
\end{aligned}$$

where $\mathbf{r}_{t,t+k}$ is the vector of returns between time t and time t+k in which we are interested, 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]

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

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:

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

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.

[1] The 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 $N(0,1)$ and n.i.d.(0, 1) have identical meaning.

Chapter 3

Asset Prices and Returns

3.1 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.

3.2 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.

3.2.1 Simple and log Returns

The linear or simple return between times t and $t-1$ is defined as[1]:

$$R_t = P_t/P_{t-1} - 1 \tag{3.1}$$

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

[1]Note that (3.1) 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:

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

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.

3.2.2 Statistical models for asset prices and returns.

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

$$\begin{aligned} \ln P_t &= \alpha_0 + \ln P_{t-1} + u_t \\ u_t &\sim N.I.D.[0, \sigma^2] \end{aligned} \tag{3.2}$$

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

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

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

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

3.2.3 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:

$$\begin{aligned}
R_{t,t+n} &= P_{t+n}/P_t - 1 \\
&= \frac{P_{t+n}}{P_{t+n-1}} \frac{P_{t+n-1}}{P_{t+n-2}} ... \frac{P_{t+1}}{P_t} - 1 \\
&= \prod_{i=1}^{n} \left(1 + R_{t+i,t+i-1}\right) - 1
\end{aligned} \tag{3.3}$$

in the case of log returns we have instead:

$$\begin{aligned}
r_{t,t+n} &= \ln\left(P_{t+n}/P_t\right) \\
&= \ln\left(\frac{P_{t+n}}{P_{t+n-1}} \frac{P_{t+n-1}}{P_{t+n-2}} ... \frac{P_{t+1}}{P_t}\right) \\
&= \sum_{i=1}^{n} r_{t+i,t+i-1}
\end{aligned} \tag{3.4}$$

Consider the case in which the length of our period in 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,... t+n.

In the case of simple returns, we have

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

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:

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

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

3.2.4 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.

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

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

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:

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

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.

3.3 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.

3.3.1 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]

$$H_{t+1}^{s} \equiv \frac{P_{t+1} + D_{t+1}}{P_t} - 1 = \frac{P_{t+1} - P_t + D_{t+1}}{P_t} = \frac{\Delta P_{t+1}}{P_t} + \frac{D_{t+1}}{P_t}, \tag{3.5}$$

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

[2]The use of '≡' emphasizes that (3.5) 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:

$$r^s_{t+1} \equiv \log\left(1 + H^s_{t+1}\right) = \log\left(\frac{P_{t+1} + D_{t+1}}{P_t}\right) = \log\left(P_{t+1} + D_{t+1}\right) - \log\left(P_t\right). \quad (3.6)$$

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 (3.5) by $\left(1 + H^s_{t+1}\right)$ and multiplying both sides by P_t/D_t we have:

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

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

$$p_t - d_t = -r^s_{t+1} + \Delta d_{t+1} + \ln\left(1 + e^{p_{t+1} - d_{t+1}}\right) \quad (3.7)$$

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] $-r^s_{t+1}$ follows from

$$\begin{aligned}\log\frac{1}{\left(1 + H^s_{t+1}\right)} &= \log 1 - \log\left(1 + H^s_{t+1}\right) \\ &= -\log\left(1 + H^s_{t+1}\right) = -r^s_{t+1}\end{aligned}$$

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:

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

where

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

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 (3.7), we obtain that the log price-dividend ratio is defined as:[4]

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

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

$$r^s_{t+1} = \kappa + \rho\left(p_{t+1}-d_{t+1}\right) + \Delta d_{t+1} - \left(p_t - d_t\right),$$

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, $\rho\left(p_{t+1}-d_{t+1}\right) - \left(p_t-d_t\right) = \Delta(p_{t+1} - d_{t+1}) - (1-\rho)\left(p_{t+1}-d_{t+1}\right)$. Moreover, by *forward* recursive substitution one obtains:

$$\begin{aligned}
(p_t - d_t) &= \kappa - r^s_{t+1} + \Delta d_{t+1} + \rho(p_{t+1}-d_{t+1}) \\
&= \kappa - r^s_{t+1} + \Delta d_{t+1} + \rho\left(\kappa - r^s_{t+2} + \Delta d_{t+2} + \rho(p_{t+2}-d_{t+2})\right) \\
&= (\kappa+\rho\kappa) - (r^s_{t+1}+\rho r^s_{t+2}) + (\Delta d_{t+1} + \rho\Delta d_{t+2}) + \rho^2(p_{t+2}-d_{t+2}) \\
&= (\kappa+\rho\kappa) - (r^s_{t+1}+\rho r^s_{t+2}) + (\Delta d_{t+1} + \rho\Delta d_{t+2}) + \\
&\quad + \rho^2(\kappa - r^s_{t+3} + \Delta d_{t+3} + \rho(p_{t+3}-d_{t+3})) \\
&= \kappa(1+\rho+\rho^2) - (r^s_{t+1}+\rho r^s_{t+2}+\rho^2 r^s_{t+3}) + (\Delta d_{t+1}+\rho\Delta d_{t+2}+\rho^2\Delta d_{t+3}) + \rho^3(p_{t+3}-d_{t+3}) \\
&= \ldots = \kappa\sum_{j=1}^{m}\rho^{j-1} + \sum_{j=1}^{m}\rho^{j-1}(\Delta d_{t+j} - r^s_{t+j}) + \rho^m\left(p_{t+m}-d_{t+m}\right).
\end{aligned}$$

[4]The 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., that[5]

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

from

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

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

$$(p_t - d_t) = \frac{\kappa}{1-\rho} + \sum_{j=1}^{m} \rho^{j-1} \left(\Delta d_{t+j} - r^s_{t+j}\right).$$

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, $r^s_{t+j} \equiv r^f + (r^s_{t+j} - r^f)$.[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., Δd_{t+1}, Δd_{t+2}, ..., Δd_{t+m} as $m \longrightarrow \infty$, 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., $(r^s_{t+1} - r^f)$, $(r^s_{t+2} - r^f)$, ..., $(r^s_{t+m} - r^f)$ as $m \longrightarrow \infty$. 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:

[5] This 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 $\rho^m \left(p_{t+m} - d_{t+m}\right)$ is zero.

[6] Here 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.

$$r_{t+1}^{s} = \kappa + \rho E_t\left(p_{t+1} - d_{t+1}\right) + E_t \Delta d_{t+1} - \left(p_t - d_t\right) + \rho u_{t+1}^{pd} + u_{t+1}^{\Delta d}$$

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

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.

3.3.2 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:

$$P_{t,T} = \frac{1}{\left(1 + Y_{t,T}\right)^{T-t}}, \tag{3.8}$$

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:

$$p_{t,T} = -\left(T - t\right) y_{t,T}, \tag{3.9}$$

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, $r^T_{t,t+1}$, is then defined as follows:

$$
\begin{aligned}
r^T_{t,t+1} &\equiv p_{t+1,T} - p_{t,T} = -\left(T-t-1\right) y_{t+1,T} + \left(T-t\right) y_{t,T} && (3.10)\\
&= y_{t,T} - \left(T-t-1\right)\left(y_{t+1,T} - y_{t,T}\right), \\
&= \left(T-t\right) y_{t,T} - \left(T-t-1\right) y_{t+1,T}, && (3.11)
\end{aligned}
$$

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:

$$
P^c_{t,T} = \frac{C}{\left(1+Y^c_{t,T}\right)} + \frac{C}{\left(1+Y^c_{t,T}\right)^2} + ... + \frac{1+C}{\left(1+Y_{t,T}\right)^{T-t}}.
$$

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:

$$
\begin{aligned}
D^c_{t,T} &= \frac{\frac{C}{\left(1+Y^c_{t,T}\right)} + 2\frac{C}{\left(1+Y^c_{t,T}\right)^2} + ... + \left(T-t\right)\frac{1+C}{\left(1+Y_{t,T}\right)^{T-t}}}{P^c_{t,T}} \\
&= \frac{C\sum\limits_{i=1}^{T-t}\frac{i}{\left(1+Y^c_{t,T}\right)^i} + \frac{(T-t)}{\left(1+Y_{t,T}\right)^{T-t}}}{P^c_{t,T}}.
\end{aligned}
$$

Note that when a bond is floating at par we have

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

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

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

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

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

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

$$E\left[\frac{y_{t,T} - \gamma y_{t+1,T}}{1-\gamma} \mid I_t\right] = r_t + \phi_{t,T} \tag{3.12}$$

From the above expression, by recursive substitution, under the terminal condition that at maturity the price equals the principal, we obtain:

$$y_{t,T} = y^*_{t,T} + E[\Phi_T \mid I_t] = \frac{1-\gamma}{1-\gamma^{T-t}} \sum_{j=0}^{T-t-1} \gamma^j E[r_{t+j} \mid I_t] + E[\Phi_T \mid I_t] \tag{3.13}$$

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 :

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

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

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

3.4 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 experience.

- 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.

3.4.1 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.

3.4.2 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.

[7] Chat 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.

3.4.3 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 packagesR!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.

```

#clear the environment
```

```r
rm(list=ls())

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
# packages used
listofpackages <- c("tidyverse","ellipse","reshape2","xts","xlsx","readxl",
"quantmod")
#installation of "xlsx" requires Java

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

tickers <- c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
             'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
             'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','^DJI')
                 #,'^GSPC',,'^IRX')

#download the historical prices
getSymbols.yahoo(tickers,
                 env = globalenv(),
                 index.class ='Date',
                 from = "1985-01-31",
                 to = "2023-07-31",
                 periodicity ="monthly")

stocks =
    merge(AXP[,6],AMGN[,6],AAPL[,6],BA[,6],CAT[,6],CSCO[,6],CVX[,6],GS[,6],HD[,6],HON[,6],
         IBM[,6],INTC[,6],JNJ[,6],KO[,6],JPM[,6],MCD[,6],MMM[,6],MRK[,6],MSFT[,6],NKE[,6],
         PG[,6],TRV[,6],UNH[,6],CRM[,6],VZ[,6],V[,6],WBA[,6],WMT[,6],DIS[,6],DOW[,6],DJI[,6]

colnames(stocks) <-
    c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
                        'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
                        'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','DJI')

write.xlsx(as.data.frame(stocks), "2023_monthly_stocks.xlsx", row.names =
    TRUE)
rm(list = c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
            'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
            'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','DJI'))

stocks_quarterly = to.quarterly(stocks$AXP)[,4]

for(i in 2:ncol(stocks)){
  x = to.quarterly(stocks[, i])[,4]
  stocks_quarterly = merge(stocks_quarterly, x)
}

colnames(stocks_quarterly) <-
```

```
    c('AXP','AMGN','AAPL','BA','CAT','CSCO','CVX','GS','HD','HON',
                       'IBM','INTC','JNJ','KO','JPM','MCD','MMM','MRK','MSFT','NKE',
                       'PG','TRV','UNH','CRM','VZ','V','WBA','WMT','DIS','DOW','DJI')

write.xlsx(as.data.frame(stocks_quarterly), "2023_quarterly_stocks.xlsx",
    row.names = TRUE)

# set tickers
tickers1 <- c("FEDFUNDS", "DGS10", "GDPPOT")
getSymbols.FRED(tickers1,
                 env = globalenv(),
                 return.class = "xts",
                 index.class ='Date'
                 )
# import data from FRED database
quarterly_fedfunds <- apply.quarterly(FEDFUNDS, last)

urlfile="https://raw.githubusercontent.com/pcm-dpc/COVID-19/
master/dati-regioni/dpc-covid19-ita-regioni.csv"
COVID_all<-read_csv(url(urlfile)) #read_csv in package tidyverse
#COVID_all<-read.csv(url(urlfile)) #read.csv in package xlxs
LOMBARDIA=subset(COVID_all,denominazione_regione=="Lombardia",select=c(1:1,4:4,7:20))
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.

3.4.4 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

```r
rm(list=ls())
#Check the working directory before importing else provide full path
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# packages used
listofpackages <- c("dygraphs",
    "tidyverse","ellipse","reshape2","highcharter","xts","xlsx","readxl","quantmod")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

raw_data            = read_xlsx("../data/2023_monthly_stocks.xlsx")
names(raw_data)[1] = 'Date'
typeof(raw_data)
typeof(raw_data$Date)
typeof(raw_data$AXP)
typeof(raw_data$CSCO)

dates <-seq(as.Date("1985-02-01"),length=462, by="months")
params <- c("Date","AXP","AMGN","AAPL","BA","CAT","CSCO", "DJI")
data <- raw_data[, c(params)]
data<- na.omit(data)
data <- data %>%
  mutate(Date = as.Date(Date, format = "%Y-%m-%d"))

params1 <- c("AXP","AMGN","AAPL","BA","CAT","CSCO", "DJI")
tsdata <- xts(raw_data[, c(params1)], order.by=dates) # creates a time
    series object
```

```
tsdata <- na.omit(tsdata) # omitting the rows with NA presence
data<- na.omit(data)
## having created the database with all observation we generate a subset
#tsdata1 <- tsdata["1992-02-01/1993-02-01"]
#data=subset(data,select=c(1:12))

## --------------------
# DATA TRANSFORMATIONS
## --------------------
#1. from prices to returns
# exact monthly returns
t1<-nrow(data)
data$AXP_ret <- data$AMGN_ret <- array(data = NA, dim = t1)
for (i in 2:t1) {
  data[i, "AMGN_ret"][[1]]=(data[i, "AMGN"][[1]]-data[i-1,
      "AMGN"][[1]])/data[i-1, "AMGN"][[1]]
  data[i, "AXP_ret"][[1]]=(data[i, "AXP"][[1]]-data[i-1,
      "AXP"][[1]])/data[i-1, "AXP"][[1]]
}

# the following lines of R apply the same transfromation to
# two series AXP and AMGN available in .xts format in a frame called tsdata.
# Could you do the same transformation in a more parsimonious way  by having
# a loop over the serie names  AXP and AMGN ?

series_names <- c("AAPL","BA","CAT","CSCO","DJI")

for (name in series_names) {
  return_col_name <- paste0(name, "_ret")
  data[, return_col_name] <- array(data = NA, dim = t1)
  for (i in 2:nrow(data)) {
    data[i, return_col_name][[1]] <- (data[i, name][[1]] - data[i - 1,
        name][[1]]) / data[i - 1, name][[1]]
  }
}

# same in .xts
t1<-nrow(tsdata)

tsdata$AXP_ret <- tsdata$AMGN_ret<- tsdata$AAPL_ret<- tsdata$BA_ret<-
    array(data = NA, dim = t1)
tsdata$CAT_ret<-tsdata$CSCO_ret <- tsdata$DJI_ret<- array(data = NA, dim =
    t1)
for (i in 2:t1) {
  tsdata[i, "AMGN_ret"][[1]]=(tsdata[i, "AMGN"][[1]]-tsdata[i-1,
      "AMGN"][[1]])/data[i-1, "AMGN"][[1]]
  tsdata[i, "AXP_ret"][[1]]=(tsdata[i, "AXP"][[1]]-tsdata[i-1,
      "AXP"][[1]])/data[i-1, "AXP"][[1]]
  #   tsdata[i, "AAPL_ret"][[1]]=(tsdata[i, "AAPL"][[1]]-tsdata[i-1,
      "AAPL"][[1]])/data[i-1, "AAPL"][[1]]
  #   tsdata[i, "BA_ret"][[1]]=(tsdata[i, "BA"][[1]]-tsdata[i-1,
      "BA"][[1]])/data[i-1, "BA"][[1]]
```

```
  #   tsdata[i, "CAT_ret"][[1]]=(tsdata[i, "CAT"][[1]]-tsdata[i-1, "CAT"][[1]])/data[i-1, "CAT"][[1]]
  #   tsdata[i, "CSCO_ret"][[1]]=(tsdata[i, "CSCO"][[1]]-tsdata[i-1, "CSCO"][[1]])/data[i-1, "CSCO"][[1]]
  #   tsdata[i, "DJI_ret"][[1]]=(tsdata[i, "DJI"][[1]]-tsdata[i-1, "DJI"][[1]])/data[i-1, "DJI"][[1]]
}

# the loop is a bit different in .xts
series_names <- c("AAPL","BA","CAT","CSCO","DJI")

for (name in series_names) {
  return_col_name <- paste0(name, "_ret")
  temporary_column  <- array(data = NA, dim = t1)

  tsdata <- merge.xts(tsdata, temporary_column) # add last column
  colnames(tsdata)[ncol(tsdata)] = return_col_name # rename it

  for (i in 2:nrow(data)) {
    tsdata[i, return_col_name] <- (tsdata[i, name][[1]] - tsdata[i - 1, name][[1]]) / tsdata[i - 1, name][[1]]
  }
}

# buy and hold returns
## what would happen had we invested $1 in the DJI and AXP at t0
## initializing values
data$DJI_cum <- data$AXP_cum <- array(data = NA, dim = nrow(data))

data[1, c("DJI_cum", "AXP_cum")] <- 1
t1<-nrow(data)
for (i in 2:t1) {
  data[i, "DJI_cum"][[1]]=data[i-1, "DJI_cum"][[1]]*(1+data[i, "DJI_ret"][[1]])
  data[i, "AXP_cum"][[1]]=data[i-1, "AXP_cum"][[1]]*(1+data[i, "AXP_ret"][[1]])
}


tsdata$DJI_cum <- array(data = NA, dim = nrow(tsdata))
tsdata$AXP_cum <- array(data = NA, dim = nrow(tsdata))
tsdata[1, c("DJI_cum", "AXP_cum")] <- 1
t1<-nrow(data)
for (i in 2:t1) {
  tsdata[i, "DJI_cum"][[1]]=tsdata[i-1, "DJI_cum"][[1]]*(1+tsdata[i, "DJI_ret"][[1]])
  tsdata[i, "AXP_cum"][[1]]=tsdata[i-1, "AXP_cum"][[1]]*(1+tsdata[i, "AXP_ret"][[1]])
}

## -------------------------
# monthly log stock returns
```

```
## ------------------------
data$DJI_lp<-log(data$DJI_cum)
data$AXP_lp<-log(data$AXP_cum)
data$DJI_lret <- c(NA,diff(data$DJI_lp))
data$AXP_lret <- c(NA,diff(data$AXP_lp))
# value of a buy-and-hold portfolio using cumulative log returns
data$DJI_cuml  <- array(data = NA, dim = nrow(data))
data[1, c("DJI_cuml")]  <- 1
for (i in 2:t1) {
  data[i, "DJI_cuml"][[1]]=data[i-1, "DJI_cuml"][[1]]*(1+data[i,
      "DJI_lret"][[1]])
      }

tsdata$DJI_lp<-log(tsdata$DJI_cum)
tsdata$AXP_lp<-log(tsdata$AXP_cum)
tsdata$DJI_lret <- diff(tsdata$DJI_lp)
tsdata$AXP_lret <- diff(tsdata$AXP_lp)
tsdata$DJI_cuml  <- array(data = NA, dim = nrow(tsdata))
tsdata[1, c("DJI_cuml")]  <- 1
for (i in 2:nrow(tsdata)) {
  tsdata[i, "DJI_cuml"][[1]]=tsdata[i-1, "DJI_cuml"][[1]]*(1+tsdata[i,
      "DJI_lret"][[1]])
}

tsdata.df <- as.data.frame(tsdata)
save(data, file='data.Rdata')
save(tsdata, file='tsdata.Rdata')
save(tsdata.df, file='tsdata.df.Rdata')

## ---------------------------
# time-series plots
#-----------------------------
#(1) plot .xts series
plot(tsdata$DJI_ret, col = "blue", lwd = 2, main = "", ylab = "")
lines(tsdata$AXP_ret, col = "green", lwd = 2)
addLegend("topleft",
          legend.names = c("DJI", "AXP"),
          lty = c(1, 1), lwd = c(2, 2),
          col = c("blue", "green"))
dev.copy2pdf(width = 5.72, out.type = "pdf",file="Figure_1xts.pdf")
dev.off()
#(2) use highchart with .xts series
highchart(type = "stock") %>%
  hc_title(text = "Monthly Log Returns") %>%
  hc_add_series(data=tsdata[, "DJI_ret"], name = "DJI_ret")%>%
  hc_add_series(data=tsdata[, "AXP_ret"], name = "AXP_ret")%>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_navigator(enabled = FALSE) %>%
  hc_scrollbar(enabled = FALSE) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = TRUE)
```

```r
#(2) use ggplot with the standard dataframe
plot <- ggplot(data, aes(x = Date)) +
  geom_point(aes(y = DJI_ret, color = "DJI"), size = 2) +
  geom_point(aes(y = AXP_ret, color = "AXP"), size = 2) +
  labs(title = "Returns",
       x = "Time", y = " Value") +
  scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black")) #+

print(plot)
ggsave(filename = "Figure_1.pdf", plot = plot, device = "pdf",width = 5.72,height=3.12)
# dev.copy2pdf(width = 4, out.type = "pdf",file="Figure_1.pdf")
# dev.off()

## ---------------------------
# comparing returns and log-returns
#-----------------------------

plot(tsdata$DJI_ret, ylab = "Returns", main = "S&P500 ", col = "blue", lwd = 2)
lines(tsdata$DJI_lret, col = "red")

# time-series plot of cumulative returns
plot(tsdata$DJI_cum,
     type = "l", col = "red", ylim = c(0, 15),
     ylab = "cumulative return mkt")
lines(tsdata$DJI_cuml, col = "blue",type = "l",ylab = "cumulative log return mkt")

# cross-plot of exact and log-linearized returns
plot(x=data$DJI_ret, y=data$DJI_lret, col="red")
lines(x=data$DJI_ret, y=data$DJI_ret,col = "blue")

#cross-plot of returns of AXP and their value predicted from the market
fm1 <- lm(AXP_ret ~ DJI_ret, data=data)
summary(fm1)
data$AXP_retfit<-c(NA,fitted(fm1))
plot(x=data$DJI_ret, y=data$AXP_ret, col="red")
lines(x=data$DJI_ret, y=data$AXP_retfit,col = "blue")

plotactfit <- ggplot(data, aes(x = DJI_ret, y = AXP_ret)) +
  geom_point(color = "red") +
  geom_line(aes(x = DJI_ret, y = AXP_retfit), color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black")   # Adding the zero line

  # Display the plot
  print(plotactfit)
```

```r
#--------------------
#plotting prices
#--------------------
sfDJI<- as.numeric(tsdata$DJI[1])
sfAXP<- as.numeric(tsdata$AXP[1])
plot(tsdata$DJI/sfDJI,col = "blue",lwd = 2)
lines(tsdata$AXP/sfAXP, col = "green",lwd = 2)
addLegend("topleft", on=1,
          legend.names = c("DJIrs", "AXPrs"),
          lty=c(1, 1), lwd=c(2, 1),
          col=c("blue", "green", "red"))

# you can interact with Chat GPT to improve on this version of the graphs

#First Question When I run the following sequence in R I get a graph with tsdata$DJI
#written at the top left of it. How do I remove this from the graph ?
#ANSWER

plot(tsdata$DJI/3267.70, col = "blue", lwd = 2, main = "", ylab = "")
lines(tsdata$AXP/3.277914, col = "green", lwd = 2)
addLegend("topleft",
          legend.names = c("DJI", "AXPrs"),
          lty = c(1, 1), lwd = c(2, 2),
          col = c("blue", "green"))  # Remove "red" from col argument

#Second Question> I would like to have the same graph in a double scale
# with DJI on the left hand scale and AXP on the right hand scale
combined_data <- data.frame(DJI = tsdata$DJI, AXP = tsdata$AXP )
dygraph(combined_data, main = "Double-Scale Time Series Graph") %>%
  dySeries("DJI", label = "DJI", color = "blue") %>%
  dySeries("AXP", label = "AXP", color = "green", axis = "y2") %>%
  dyAxis("y", label = "DJI") %>%
  dyAxis("y2", label = "AXP", independentTicks = TRUE) %>%
  dyLegend(width = 250)



#--------------------
#plotting series from a list using GGPLOT
#--------------------
plot <- ggplot(data, aes(x = Date)) +
  geom_line(aes(y = DJI/3267.70, color = "DJI"), size = 2) +
  geom_line(aes(y = AXP/3.277914, color = "AXP"), size = 2) +
  labs(title = "Trends",
       x = "Time", y = " Value") +
  scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black")) +
  scale_x_continuous(breaks = data$Date, labels = data$Date)  # Add this line for x-axis labels
```

```
# Print the plot
print(plot)

#Ask Chat GPT: When I run the following code in R I get "too many " labels on the x axis,
#how can I reduce the number of labels (say one every 5 years) ?
#Answer
#To reduce the number of x-axis labels in your ggplot, you can use the scale_x_date() function
#with the date_breaks argument to specify the intervals at which you want the labels to appear.
#In your case, you want to display labels every 5 years. Here's how you can modify your code to achieve this:
ggplot(data, aes(x = Date)) +
  geom_line(aes(y = DJI/3267.70, color = "DJI"), size = 1) +
  geom_line(aes(y = AXP/3.277914, color = "AXP"), size = 1) +
  labs(title = "Trends",
       x = "Time", y = " Value") +
  scale_color_manual(values = c("DJI" = "red", "AXP" = "blue")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black")) +
  scale_x_date(date_breaks = "5 years", date_labels = "%Y")
#In the code above:
#scale_x_date() is used to control the x-axis (date) scale.
#date_breaks = "5 years" specifies that you want to display labels every 5 years.
#date_labels = "%Y" specifies the date format you want to use for the labels (in this case, the year only).
#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.



## ---------------------------------------------------------
#  combine  several plots on one canvas
## ---------------------------------------------------------

par(mfrow = c(2, 2))

plot(tsdata$DJI_ret, ylab = "Returns", main = "DJ30 ", col = "blue", lwd = 2)

plot(x=data$DJI_ret, y=data$DJI_lret, col="red")
lines(x=data$DJI_ret, y=data$DJI_ret,col = "blue")

plot(x=data$AXP_ret, y=data$AXP_lret, col="red",ylim = c(-0.5, 1))
lines(x=data$AXP_ret, y=data$AXP_ret,col = "blue")

plot(tsdata$DJI_cum,
     type = "l", col = "red", ylim = c(0, 12),main = "DJ30 ",
```

```
      ylab = "cumulative return mkt")

par(mfrow = c(1, 1))


## -------------------------
#  HISTOGRAMS AND QQ PLOTS
## -------------------------

## Histograms
s1 <- na.omit(tsdata$DJI_ret)
hist(s1, breaks = seq(min(s1), max(s1), l = 20+1),prob=TRUE, main =
    "histogram of monthly returns")
curve(dnorm(x,mean=mean(s1),sd=sd(s1)),col='darkblue',lwd=2,add=TRUE)

## Histograms with Highcharter using .xts data

hc_hist <- hist(coredata(tsdata$DJI_lret), breaks = 50, plot = FALSE)
hchart(hc_hist, color = "cornflowerblue")%>%
  hc_title(text =
             paste("DJI",
                   "Log Returns Distribution",
                   sep = " ")) %>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = FALSE)

hc_hist <- hist(tsdata[, "DJI_lret"], breaks = 50, plot = FALSE)
hchart(hc_hist, color = "cornflowerblue")%>%
  hc_title(text =
             paste("DJI",
                   "Log Returns Distribution",
                   sep = " ")) %>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = FALSE)
## ------------------------

qqplot(tsdata.df$DJI_ret,
       tsdata.df$DJI_lret,
       ylim = c(-0.15,0.15), xlim = c(-0.15,0.15),
       ylab = "monthly return. log approximation",
       xlab = "monthly return. exact computation",
       main = "Quantile-Quantile plot (Q-Q plot)")
mod5 <- lm(tsdata.df$DJI_ret ~ tsdata.df$DJI_lret)
abline(reg = mod5, col = "red")

# qq-plot versus normal dist
qqnorm(tsdata$DJI_ret,
       ylim = c(-0.15,0.15),ylab = "monthly return. sample quantile",
       xlab = "monthly return. theoretical quantiles",
       main = "Normal (Q-Q plot)")
```

```r
qqline(tsdata$DJI_ret, datax = FALSE, distribution = qnorm,
       probs = c(0.25, 0.75), qtype = 7)

## ---------------------
#  CORRELATION ANALYSIS
## ---------------------
tsdata.df <- as.data.frame(tsdata)
# Select specific columns and observations from the start date onward
selected_cols <- c("AMGN_ret", "AXP_ret", "AAPL_ret", "BA_ret", "CAT_ret", "CSCO_ret", "DJI_ret")
datashow <- subset(tsdata.df[, selected_cols])
datashow<-na.omit(datashow)
  # Print the resulting subset
summary(datashow) # this is very useful to get a grip on the data structure
mean(datashow[,"AMGN_ret"])
sd(datashow[,"AMGN_ret"])
var(datashow[,"AMGN_ret"])
cor(datashow)
cor.datacor = cor(datashow, use="complete.obs")
cor.datacor

## ---------------------
ord <- order(cor.datacor[1,])
ordered.cor.datacor <- cor.datacor[ord, ord]
plotcorr(ordered.cor.datacor, col=cm.colors(11)[5*ordered.cor.datacor + 6])


## ---------------------
cormat <- round(cor(datashow),2)
head(cormat)
melted_cormat <- melt(cormat)
head(melted_cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
# Get lower triangle of the correlation matrix
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}
upper_tri <- get_upper_tri(cormat)
upper_tri
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Heatmap
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
```

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

3.4.5 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 becasue 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.

```
#clear the environment
rm(list=ls())
## ----------------------------------------------------------------------
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
library(data.table)
library(scales)
library(ggplot2)
library(xts)
link <-
    "https://raw.githubusercontent.com/DavZim/Efficient_Frontier/master/data/fin_data.csv"
dt <- fread(link)
dt[, date := as.Date(date)]

# create indexed values
dt[, idx_price := price/price[1], by = ticker]

# plot the indexed values
ggplot(dt, aes(x = date, y = idx_price, color = ticker)) +
  geom_line() +
```

```
  # Miscellaneous Formatting
  theme_bw() + ggtitle("Price Developments") +
  xlab("Date") + ylab("Price\n(Indexed 2000 = 1)") +
  scale_color_discrete(name = "Company")
# calculate the arithmetic returns
dt[, ret := price / shift(price, 1) - 1, by = ticker]

# summary table
# take only non-na values
tab <- dt[!is.na(ret), .(ticker, ret)]

# calculate the expected returns (historical mean of returns) and volatility (standard deviation of returns)
tab <- tab[, .(er = round(mean(ret), 4),
               sd = round(sd(ret), 4)),
           by = "ticker"]
ggplot(tab, aes(x = sd, y = er, color = ticker)) +
  geom_point(size = 5) +
  # Miscellaneous Formatting
  theme_bw() + ggtitle("Risk-Return Tradeoff") +
  xlab("Volatility") + ylab("Expected Returns") +
  scale_y_continuous(label = percent, limits = c(0, 0.03)) +
  scale_x_continuous(label = percent, limits = c(0, 0.1))



# load the data
link <- "https://raw.githubusercontent.com/DavZim/Efficient_Frontier/master/data/mult_assets.csv"
df <- data.table(read.csv(link))

df_table <- melt(df)[, .(mean = mean(value), sd = sd(value)), by = variable]

er_x <- mean(df$x)
er_y <- mean(df$y)
er_z <- mean(df$z)
sd_x <- sd(df$x)
sd_y <- sd(df$y)
sd_z <- sd(df$z)
cov_xy <- cov(df$x, df$y)
cov_xz <- cov(df$x, df$z)
cov_yz <- cov(df$y, df$z)

# two assets
two_assets_seq <- seq(from = 0, to = 1, length.out = 1000)

two <- data.table(wx = two_assets_seq,
                  wy = 1 - two_assets_seq)

two[, ':=' (er_p = wx * er_x + wy * er_y,
            sd_p = sqrt(wx^2 * sd_x^2 +
                          wy^2 * sd_y^2 +
```

```
                          2 * wx * (1 - wx) * cov_xy))]

# plot_two <- ggplot() +
#   geom_point(data = two, aes(x = sd_p, y = er_p, color = wx)) +
#   geom_point(data = df_table[variable != "z"],
#              aes(x = sd, y = mean), color = "red", size = 3, shape = 18) +
#   theme_bw() + ggtitle("Possible Portfolios with Two Risky Assets") +
#   xlab("Volatility") + ylab("Expected Returns") +
#   scale_y_continuous(label = percent, limits = c(0, max(two$er_p) * 1.2))
    +
#   scale_x_continuous(label = percent, limits = c(0, max(two$sd_p) * 1.2))
    +
#   scale_color_continuous(name = expression(omega[x]), labels = percent)
#
# ggsave(plot_two, file = "two_assets.png", scale = 1, dpi = 600)

  ggplot() +
  geom_point(data = two, aes(x = sd_p, y = er_p, color = wx)) +
  geom_point(data = df_table[variable != "z"],
             aes(x = sd, y = mean), color = "red", size = 3, shape = 18) +
  theme_bw() + ggtitle("Possible Portfolios with Two Risky Assets") +
  xlab("Volatility") + ylab("Expected Returns") +
  scale_y_continuous(label = percent, limits = c(0, max(two$er_p) * 1.2)) +
  scale_x_continuous(label = percent, limits = c(0, max(two$sd_p) * 1.2)) +
  scale_color_continuous(name = expression(omega[x]), labels = percent)

# three assets
three_assets_seq <- seq(from = 0, to = 1, length.out = 1000)

three <- data.table(wx = rep(three_assets_seq, each =
    length(three_assets_seq)),
                    wy = rep(three_assets_seq, length(three_assets_seq)))

three[, wz := 1 - wx - wy]

three[, ':=' (er_p = wx * er_x + wy * er_y + wz * er_z,
              sd_p = sqrt(wx^2 * sd_x^2 +
                          wy^2 * sd_y^2 +
                          wz^2 * sd_z^2 +
                          2 * wx * wy * cov_xy +
                          2 * wx * wz * cov_xz +
                          2 * wy * wz * cov_yz))]

three <- three[wx >= 0 & wy >= 0 & wz >= 0]

plot_three <- ggplot() +
  geom_point(data = three, aes(x = sd_p, y = er_p, color = wx - wz)) +
  geom_point(data = df_table, aes(x = sd, y = mean), color = "red", size =
      3, shape = 18) +
  theme_bw() + ggtitle("Possible Portfolios with Three Risky Assets") +
  xlab("Volatility") + ylab("Expected Returns") +
  scale_y_continuous(label = percent, limits = c(0, max(three$er_p) * 1.2))
```

```
      +
  scale_x_continuous(label = percent, limits = c(0, max(three$sd_p) * 1.2))
      +
  scale_color_gradientn(colors = c("red", "blue", "yellow"),
                        name = expression(omega[x] - omega[z]), labels =
                            percent)

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 bythe 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:

```
# Load necessary libraries
library(quantmod)
library(ggplot2)

# Sample returns for five assets (replace with your data)
returns <- data.frame(
  A = c(0.05, 0.06, 0.07, 0.04, 0.03),
  B = c(0.08, 0.09, 0.06, 0.07, 0.05),
  C = c(0.10, 0.12, 0.11, 0.13, 0.09),
  D = c(0.06, 0.05, 0.04, 0.03, 0.07),
  E = c(0.07, 0.08, 0.09, 0.06, 0.05)
)

# Calculate mean and standard deviation of asset returns
mean_returns <- colMeans(returns)
sd_returns <- apply(returns, 2, sd)

# Number of random portfolios to simulate
n_portfolios <- 10000

# Create a function to simulate random portfolios
simulate_portfolios <- function(n_portfolios, returns) {
  portfolio_returns <- numeric(n_portfolios)
  portfolio_volatility <- numeric(n_portfolios)

```

```
  for (i in 1:n_portfolios) {
    weights <- runif(ncol(returns))
    weights <- weights / sum(weights)
    portfolio_returns[i] <- sum(weights * mean_returns)
    portfolio_volatility[i] <- sqrt(t(weights) %*% cov(returns) %*% weights)
  }

  portfolio_data <- data.frame(Return = portfolio_returns, Volatility =
      portfolio_volatility)
  return(portfolio_data)
}

# Simulate random portfolios
set.seed(123)  # For reproducibility
portfolios <- simulate_portfolios(n_portfolios, returns)

# Plot the efficient frontier
ggplot(portfolios, aes(x = Volatility, y = Return)) +
  geom_point(alpha = 0.5) +
  geom_point(aes(x = sd_returns, y = mean_returns), color = "red", size =
      3) +
  labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
  ggtitle("Efficient Frontier") +
  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:

```r
# Load necessary libraries
library(quadprog)
library(ggplot2)

# Sample returns for 20 assets (replace with your data)
# Ensure you have a matrix of returns with 20 columns
returns_matrix <- read.csv("your_returns_data.csv")  # Replace with your data

# Calculate the mean returns and covariance matrix
mean_returns <- colMeans(returns_matrix)
cov_matrix <- cov(returns_matrix)

# Define the objective function to minimize risk (standard deviation)
portfolio.objective <- function(weights, cov_matrix) {
  portfolio_return <- sum(weights * mean_returns)
  portfolio_volatility <- sqrt(t(weights) %*% cov_matrix %*% weights)
  return(portfolio_volatility)
}

# Set up constraints (sum of weights = 1)
A_eq <- matrix(1, 1, ncol(returns_matrix))
b_eq <- 1

# Generate a range of target returns
target_returns <- seq(min(mean_returns), max(mean_returns), length.out = 100)

# Initialize vectors to store results
portfolio_returns <- numeric(length(target_returns))
portfolio_volatilities <- numeric(length(target_returns))

# Optimize for each target return
for (i in 1:length(target_returns)) {
  target_return <- target_returns[i]

  # Set up the quadratic programming problem
  Dmat <- cov_matrix
```

```
  dvec <- rep(0, ncol(returns_matrix))
  Amat <- matrix(c(mean_returns, 1), nrow = 1)
  bvec <- target_return

  # Solve the optimization problem
  weights <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)$solution

  # Calculate portfolio risk (volatility)
  portfolio_volatility <- portfolio.objective(weights, cov_matrix)

  # Store results
  portfolio_returns[i] <- target_return
  portfolio_volatilities[i] <- portfolio_volatility
}

# Create a data frame for efficient frontier points
efficient_frontier <- data.frame(Return = portfolio_returns, Volatility =
    portfolio_volatilities)

# Plot the efficient frontier
ggplot(efficient_frontier, aes(x = Volatility, y = Return)) +
  geom_line() +
  labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
  ggtitle("Efficient Frontier") +
  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 ÿour_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.

```
rm(list=ls())
#Check the working directory before importing else provide full path
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# packages used
listofpackages <- c("dygraphs",
```

```
    "tidyverse","ellipse","reshape2","highcharter","xts","xlsx","readxl","quantmod","quadprog

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

raw_data            = read_xlsx("../data/2023_monthly_stocks.xlsx")
names(raw_data)[1] = 'Date'
typeof(raw_data)
typeof(raw_data$Date)
typeof(raw_data$AXP)
typeof(raw_data$CSCO)

dates <-seq(as.Date("1985-02-01"),length=462, by="months")
params <- c("Date","AXP","AMGN","AAPL","BA","CAT","CSCO","CVX","GS",
            "HD","HON","IBM","INTC","JNJ","KO","JPM")
data <- raw_data[, c(params)]
data<- na.omit(data)
data <- data %>%
  mutate(Date = as.Date(Date, format = "%Y-%m-%d"))

t1<-nrow(data)
series_names <- c("AXP","AMGN","AAPL","BA","CAT","CSCO","CVX","GS",
                  "HD","HON","IBM","INTC","JNJ","KO","JPM")

for (name in series_names) {
  return_col_name <- paste0(name, "_ret")
  data[, return_col_name] <- array(data = NA, dim = t1)
  for (i in 2:nrow(data)) {
    data[i, return_col_name][[1]] <- (data[i, name][[1]] - data[i - 1,
       name][[1]]) / data[i - 1, name][[1]]
  }
}

params1 <-
    c("AXP_ret","AMGN_ret","AAPL_ret","BA_ret","CAT_ret","CSCO_ret","CVX_ret","GS_ret",
            "HD_ret","HON_ret","IBM_ret","INTC_ret","JNJ_ret","KO_ret","JPM_ret")
returns_data <- data[, c(params1)]
returns_data <- na.omit(returns_data)

returns_matrix<-as.matrix(returns_data)

# Calculate the mean returns and covariance matrix
mean_returns <- colMeans(returns_matrix)
cov_matrix <- cov(returns_matrix)

# Define the objective function to minimize risk (standard deviation)
portfolio.objective <- function(weights, cov_matrix) {
  portfolio_return <- sum(weights * mean_returns)
```

```
  portfolio_volatility <- sqrt(t(weights) %*% cov_matrix %*% weights)
  return(portfolio_volatility)
}

# Set up constraints (sum of weights = 1)
$A_eq <- matrix(1, nrow = 1, ncol = ncol(returns_matrix))
$b_eq <- matrix(1, nrow = 1)

# Generate a range of target returns
target_returns <- seq(min(mean_returns), max(mean_returns), length.out =
    1000)

# Initialize vectors to store results
portfolio_returns <- numeric(length(target_returns))
portfolio_volatilities <- numeric(length(target_returns))

# Optimize for each target return
for (i in 1:length(target_returns)) {
  target_return <- target_returns[i]

  # Set up the quadratic programming problem
  Dmat <- 2*cov_matrix
  dvec <- matrix(rep(0, ncol(returns_matrix)),ncol=1)
  a1mat<- matrix(mean_returns, nrow =ncol(returns_matrix))
  a2mat<-matrix(rep(1, ncol(returns_matrix)), nrow =ncol(returns_matrix))
  Amat <- cbind(a1mat, a2mat)
  bvec <- matrix(c(target_return, 1),ncol=1)

  # Solve the optimization problem
  weights <- solve.QP(Dmat, dvec, Amat, bvec, meq = 2)$solution

  # Calculate portfolio risk (volatility)
  portfolio_volatility <- portfolio.objective(weights, cov_matrix)

  # Store results
  portfolio_returns[i] <- target_return
  portfolio_volatilities[i] <- portfolio_volatility
}

# Create a data frame for efficient frontier points
efficient_frontier <- data.frame(Return = portfolio_returns, Volatility =
    portfolio_volatilities)

# Plot the efficient frontier
ggplot(efficient_frontier, aes(x = Volatility, y = Return)) +
  geom_line() +
  labs(x = "Standard Deviation (Risk)", y = "Mean Return") +
  ggtitle("Efficient Frontier") +
  theme_minimal()
```

3.5 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 form 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 1 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-Uare 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,...5,j=1,...,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 4

The Modelling Process at Work: the CER model

4.1 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

4.2 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

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

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 $t, t-1, t-2, ..., t-n+1$ (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:

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

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

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]

[1] Empirical 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.

4.2.1 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.

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

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

But $n\mu + \Phi^{-1}\left(\alpha\right)\sqrt{n}\sigma$, 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.

4.3 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:

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

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

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

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

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

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

4.3.1 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 :

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

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

$$\beta = \begin{pmatrix} \beta_1 \\ . \\ . \\ . \\ \beta_k \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \varepsilon_1 \\ . \\ . \\ . \\ \varepsilon_N \end{pmatrix}.$$

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, $\widehat{\beta}$, defined as follows:

$$\mathbf{S}\left(\widehat{\beta}\right) = \min_{\beta} \mathbf{e}\left(\beta\right)' \mathbf{e}\left(\beta\right).$$

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

1. $\mathbf{X}'\widehat{\epsilon} = 0$;

2. rank$(\mathbf{X}) = k$.

Condition 1 imposes orthogonality between the **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 **X** matrix are linearly independent.

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

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

OLS in the CER

In the CER we have:

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

$$\begin{aligned}
\mathbf{y} &= \begin{pmatrix} r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_T \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix}, \\
\beta &= \mu, \quad \epsilon = \begin{pmatrix} \varepsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_T \end{pmatrix}.
\end{aligned}$$

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:

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

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:

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

4.4 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 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.

4.5 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.

4.5.1 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.

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

```
# author: Carlo Favero
# created: August, 2023
# comments: Original Examples are taken from  chapter 11 in Zivot and Wang (2006)

rm(list=ls()) #Removes all items in Environment!
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# set output options
options(width = 70, digits=4)

#install.packages("fEcofin", repos="http://R-Forge.R-project.org")
library(fEcofin)
# load required packages
listofpackages <- c("ellipse","dygraphs","ggplot2")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

install.packages(c("cluster","mvoutlier","pastecs","fPortfolio"),
repos="http://cran.r-project.org")
# load required packages
library(cluster)
library(mvoutlier)
library(pastecs)
library(fPortfolio)

####################################################
# Data Loadings and Transform: Descriptive Analysis
####################################################

# create data frame with dates as rownames
berndt.df = berndtInvest[, -1]
berndt.df$date <- as.Date(berndtInvest[, 1])
rownames(berndt.df) = as.character(berndtInvest[, 1])
colnames(berndt.df)
dimnames(berndt.df)[[2]] #command alternative to the previous one

# transform the data and compute cumulative returns

t0 <- which(berndt.df$date == "1978-01-01")
t1 <- which(berndt.df$date == "1987-12-01")

series_names <- c("CITCRP","CONED","CONTIL","DATGEN","DEC","DELTA","GENMIL","GERBER","IBM",
"MARKET","MOBIL","PANAM","PSNH","TANDY","TEXACO","WEYER","RKFREE")

```

```
for (name in series_names) {
  P_col_name <- paste0(name,"_P")
  LP_col_name<- paste0("L",P_col_name)
  berndt.df[t0, P_col_name] <- 1
  for (i in (t0+1):(t1)) {
    berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] *
        (1+berndt.df[i, name][[1]] )
  }
  berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
}
# add a trend to the database
berndt.df$TREND<- array(data = NA, dim = nrow(berndt.df))
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

############################
# Descriptive Analysis
############################

  #We can now plot, please note the difference with plotting from a
     time-series object

plot(berndt.df$date[t0:t1],berndt.df$TEXACO[t0:t1],ylab =
    "TEXACO",xlab="year", main = "Monthly Returns", col = "blue", lwd =
    2,type="l")


plot(berndt.df$date[t0:t1],berndt.df$TEXACO[t0:t1], col = 'blue', type =
    "l",
      ylab = "returns TEXACO and MKT", xlab = "date",lwd = 2)
lines(y = rep(mean(berndt.df$TEXACO[t0:t1], na.rm = T),
    length(berndt.df$TEXACO[t0:t1])), x = berndt.df$date[t0:t1], col =
    "red")
lines(y = berndt.df$MARKET[t0:t1], x = berndt.df$date[t0:t1], col =
    "green",lwd = 2)
legend("topleft", legend = c("TEXACO", "MKT"),
       col = c("blue", "green"), lty = 1)

plot(berndt.df$date[t0:t1],berndt.df$LTEXACO_P[t0:t1], col = 'blue', type =
    "l",
      ylab = "portfolios TEXACO and MKT", xlab = "date",ylim = c(-0.5,
         2),lwd = 2)
lines(y = berndt.df$LMARKET_P[t0:t1], x = berndt.df$date[t0:t1], col =
    "green",lwd = 2)
legend("topleft", legend = c("TEXACO", "MKT"),
       col = c("blue", "green"), lty = 1)

# Create the plot using ggplot, as generated by Chat GPT
ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = LTEXACO_P), color = "blue", size = 2, linetype =
     "solid") +
```

```r
  geom_line(aes(y = LMARKET_P), color = "green", size = 2, linetype = "solid") +
  labs(x = "Date", y = "Portfolios TEXACO and MKT") +
  ylim(-0.5, 2) +
  theme_minimal() +
  theme(
    legend.position = "topleft",
    legend.title = element_blank(),
    legend.text = element_text(size = 12),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    plot.title = element_text(size = 16, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
  ) +
  guides(fill = guide_legend(override.aes = list(size = 2)))
############################
# Asset Allocation with CER
############################
returns.df=berndt.df[, c(1:9,11:16)]
#returns.df = berndt.df[, c(-10, -17)
exreturns.df=returns.df-berndt.df$RKFREE
returns.mat = as.matrix(exreturns.df)
n.obs = nrow(returns.mat)
#Estimation of CER model parameters
cov.sample=var(returns.mat)
mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
#
# compute tangency portfolio
#
e = matrix(1, nrow = nrow(cov.sample), ncol = 1) # unitary column vector e
w.tan.sample = (solve(cov.sample)%*%(mu))/as.numeric(t(e)%*%(solve(cov.sample)%*%(mu)))
colnames(w.tan.sample) = "sample"
barplot(t(w.tan.sample), horiz=F, main="Weights", col="blue", cex.names = 0.75, las=2)
################################
# Using the fportfolio package
################################
#returns.df=berndt.df[, c(1:9,11:16)]
#exreturns.df=returns.df-berndt.df$RKFREE
companies <- colnames(exreturns.df)
```

```
#ts
tsdata <- ts(exreturns.df, start = c(1978, 1), frequency = 12, names =
    companies)
s1 <- window(tsdata[, "TEXACO"], start = c(1978, 1), end = c(1987, 12))
dygraph(s1, ylab = "TEXACO", main = "monthly excess returns")
data01ts <- as.timeSeries(tsdata)
# financial data description
ddown<-drawdowns(data01ts)
ddowndata <- ts(ddown, start = c(1978, 1), frequency = 12, names =
    companies)
s1 <- window(ddowndata[, "TEXACO"], start = c(1978, 1), end = c(1987, 12))
dygraph(s1, ylab = "TEXACO", main = "drawdowns")
drawdownsStats(data01ts[, "TEXACO"])
#-----------------------
# Portfolio Allocation
#-----------------------

# Step 1 define the data in our case 15 excess returns data in data01ts
showClass("fPFOLIODATA")

lppData <- portfolioData(data = data01ts, spec = portfolioSpec())
# once the data have been defined we can get info on them
str(lppData, width = 65, strict.width = "cut")
print(lppData)
getData(portfolioData(lppData))[-1]
getStatistics(portfolioData(lppData))

# Step 2 Set Portfolio Constraints

showClass("fPFOLIOCON")
#default constraints: long-only
Data<-data01ts
Spec <- portfolioSpec()
setTargetReturn(Spec) <- mean(Data)
Constraints <- "LongOnly"
defaultConstraints <- portfolioConstraints(Data, Spec, Constraints)
str(defaultConstraints, width = 65, strict.width = "cut")
print(defaultConstraints)

# short constraints
shortConstraints <- "Short"
portfolioConstraints(Data, Spec, shortConstraints)

# box constraints
box.1 <- "minW[1:15] = 0.1"
box.2 <- "maxW[1:15] = 1" # you can have more boxes before combining them
boxConstraints <- c(box.1, box.2)
boxConstraints
portfolioConstraints(Data, Spec, boxConstraints)


# Step 3 Computing Optimal Portfolios
```

```r

#3.0 A benchmark: equal weight portfolio
ewSpec <- portfolioSpec()
nAssets <- ncol(data01ts)
setWeights(ewSpec) <- rep(1/nAssets, times = nAssets)
ewPortfolio <- feasiblePortfolio(
  data = data01ts,
  spec = ewSpec,
  constraints = "LongOnly")
print(ewPortfolio)

# Efficient Frontier plot
setNFrontierPoints(ewSpec) <- 25
eff_ew_frontier <- portfolioFrontier(data = data01ts, spec = ewSpec,
    constraints = "LongOnly")
tailoredFrontierPlot(object = eff_ew_frontier)

#3.1 Long-Only
tgSpec <- portfolioSpec()
setRiskFreeRate(tgSpec) <- 0
constraints <- "longOnly"
tgPortfolio <- tangencyPortfolio(
  data = data01ts,
  spec = tgSpec, constraints = constraints)
print(tgPortfolio)

#printing the results
col <- seqPalette(ncol(data01ts), "BuPu")
weightsPie(tgPortfolio, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)
weightedReturnsPie(tgPortfolio, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)
covRiskBudgetsPie(tgPortfolio, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)

efficient_frontier <- portfolioFrontier(data = data01ts, spec = tgSpec,
    constraints = constraints)
print(efficient_frontier)
# Efficient Frontier plot
setNFrontierPoints(tgSpec) <- 25
efficient_frontier <- portfolioFrontier(data = data01ts, spec = tgSpec,
    constraints = constraints)
tailoredFrontierPlot(object = efficient_frontier)

#---------------------
#3.2 Box-Constraints
#---------------------
boxSpec <- portfolioSpec()
setRiskFreeRate(boxSpec) <- 0
```

```
boxConstraints <- c("minW[1:15]=0.05", "maxW[1:15]=0.5")
tgPortfolio1 <- tangencyPortfolio(
  data = data01ts,
  spec = boxSpec, constraints = boxConstraints)
print(tgPortfolio1)

#printing the results
col <- seqPalette(ncol(data01ts), "BuPu")
weightsPie(tgPortfolio1, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)
weightedReturnsPie(tgPortfolio, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)
covRiskBudgetsPie(tgPortfolio, box = FALSE, col = col)
mtext(text = "Tangency MV Portfolio", side = 3, line = 1.5,
      font = 2, cex = 0.7, adj = 0)
```

4.5.2 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.

```
# Asset Allocation with CER
# elaboration on the original code produced by E.Zivot by C. Favero
# author: Carlo Favero
# created: August, 2023
#
# comments: Original Examples are taken from  chapter 11 in Zivot and Wang
    (2006)

rm(list=ls()) #Removes all items in Environment!
```

```r
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

#install.packages("fEcofin", repos="http://R-Forge.R-project.org")
library(fEcofin)
# load required packages
listofpackages <- c("ellipse","dygraphs","ggplot2","reshape2")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

install.packages(c("cluster","mvoutlier","pastecs","fPortfolio"),repos="http://cran.r-project
# load required packages
library(cluster)
library(mvoutlier)
library(pastecs)
library(fPortfolio)


# create data frame with dates as rownames
berndt.df = berndtInvest[, -1]
rownames(berndt.df) = as.character(as.Date(berndtInvest[, 1]))


################################################################################
# Derive the optimal portfolio weights (i.e. the weights in the tangency portfolio)
# using the CER for (i) the Minimun Variance Portfolio , (ii) the tangency portfolio.
###############################################################################
returns.df=berndt.df[, c(1:9,11:16)]
#returns.df = berndt.df[, c(-10, -17)
exreturns.df=returns.df-berndt.df$RKFREE
returns.mat = as.matrix(exreturns.df)
# using ggplot to plot series in returns
berndt.df$date <- as.Date(row.names(berndt.df))

# Create the time series plot using ggplot
ggplot(data = berndt.df, aes(x = date, y = WEYER)) +
  geom_line() +                   # Add a line plot
  labs(x = "Date", y = "WEYER")  # Label the axes


#
# compute global min variance portfolio
#
# use CER model: estimate the relevant unknown parameters with the sample covariances
```

```r
returns.mat = as.matrix(exreturns.df)
n.obs = nrow(returns.mat)
cov.sample=var(returns.mat)
mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
e = matrix(1, nrow = nrow(cov.sample), ncol = 1) # unitary column vector e
#
# compute GMIN portfolio
#
w.gmin.sample = solve(var(returns.mat))%*%rep(1,nrow(cov.sample))
w.gmin.sample = w.gmin.sample/sum(w.gmin.sample)
berndt.df$GMIN<-returns.mat%*%w.gmin.sample

barplot(t(w.gmin.sample), horiz=F, main="Weights", col="blue", cex.names = 0.75, las=2)

ggplot(data = berndt.df, aes(x = date, y = GMIN)) +
  geom_line() +                     # Add a line plot
  labs(x = "Date", y = "GMIN")  # Label the axes

#
# compute tangency portfolio
#
w.tan.sample = (solve(cov.sample)%*%as.numeric(mu))
w.tan.sample =w.tan.sample/as.numeric(t(e)%*%(solve(cov.sample)%*%(mu)))

berndt.df$TAN<-returns.mat%*%w.tan.sample

# visualize the differences
par(mfrow=c(1,2))
barplot(t(w.tan.sample), horiz=T, main="Tangency Port CER", col="blue", cex.names = 0.75, las=1)
barplot(t(w.gmin.sample), horiz=T, main="Min Var Port CER", col="red", cex.names = 0.75, las=1)
par(mfrow=c(1,1))

plot <- ggplot(data= berndt.df, aes(x = date)) +
  geom_line(aes(y = TAN, color = "TAN"), size = 1) +
  geom_line(aes(y = GMIN, color = "GMIN"), size = 1) +
  labs(title = "Returns",
       x = "Time", y = "Monthly Returns") +
  scale_color_manual(values = c("TAN" = "red", "GMIN" = "blue")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black"))

print(plot)


#################################################
# Graphs the value over-time of 1 dollar invested in 1978:1 until the end of the
# available sample in the two alternative tangency portfolios and in the market
```

```r
##################################################
berndt.df$Port_mkt <- berndt.df$Port_TAN<- berndt.df$Port_GMIN <- array(data = NA, dim = nrow(berndt.df))

berndt.df[1, c("Port_mkt", "Port_TAN","Port_GMIN")] <- 1
t1<-nrow(berndt.df)
for (i in 2:t1) {
  berndt.df[i, "Port_mkt"][[1]]=berndt.df[i-1, "Port_mkt"][[1]]*(1+berndt.df[i, "MARKET"][[1]])
  berndt.df[i, "Port_TAN"][[1]]=berndt.df[i-1, "Port_TAN"][[1]]*(1+berndt.df[i, "TAN"][[1]])
  berndt.df[i, "Port_GMIN"][[1]]=berndt.df[i-1, "Port_GMIN"][[1]]*(1+berndt.df[i, "GMIN"][[1]])
}


# time series Plot of the three Portfolios

plot <- ggplot(data= berndt.df, aes(x = date)) +
  geom_line(aes(y = Port_mkt, color = "Port_mkt"), size = 1) +
  geom_line(aes(y = Port_GMIN, color = "Port_GMIN"), size = 1) +
  geom_line(aes(y = Port_TAN, color = "Port_TAN"), size = 1) +
  labs(title = "Returns",
       x = "Time", y = "Monthly Returns") +
  scale_color_manual(values = c("Port_mkt" = "red", "Port_GMIN" = "blue","Port_TAN" = "green")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black"))


# compare means and sd values on global min variance portfolios

mu.gmin.sample = as.numeric(colMeans(berndt.df$GMIN))
mu.tan.sample = as.numeric(colMeans(berndt.df$TAN))
sd.gmin.sample = as.numeric(apply(berndt.df$GMIN,2,sd))
sd.tan.sample = as.numeric(apply(berndt.df$TAN,2,sd))
cbind(mu.tan.sample,mu.gmin.sample, sd.tan.sample, sd.gmin.sample)

## ------------------------------
#  BACKTESTING   with fPortfolio
## ------------------------------
companies <- colnames(berndt.df)
#getting the data in ts format
tsdata <- ts(berndt.df, start = c(1978, 1), frequency = 12, names = companies)
data01ts <- as.timeSeries(tsdata)
ddown<-drawdowns(data01ts)
ddowndata <- ts(ddown, start = c(1978, 1), frequency = 12, names = companies)
s1 <- window(ddowndata[, "TAN"], start = c(1978, 1), end = c(1987, 12))
dygraph(s1, ylab = "TAN", main = "drawdowns")
drawdownsStats(data01ts[, "TAN"])
```

```
## ------------------------------
#  out-of-sample BACKTESTING
## ------------------------------

Data <- data01ts
Spec <- portfolioSpec()
Constraints <- "LongOnly"
Backtest <- portfolioBacktest()
setWindowsHorizon(Backtest) <- "60m"
equidistWindows(data = Data, backtest = Backtest)


#Specify assets for backtesting
#Formula <- MARKET ~ CITCRP + CONED + CONTIL + DATGEN + DEC + DELTA +
# +     GENMIL + GERBER +IBM+MOBIL+PANAM+PSNH+TANDY+TEXACO+WEYER
Formula <- MARKET ~ CITCRP + CONED + CONTIL + DATGEN + DEC + DELTA +
  GENMIL + GERBER + IBM + MOBIL + PANAM + PSNH + TANDY + TEXACO + WEYER

#Optimize rolling portfolios and run backtests
#btportfolios <- portfolioBacktesting(formula = Formula,
#                                       +data = data01ts, spec = Spec, constraints = Constraints,
#                                      + backtest = Backtest, trace = FALSE)

btportfolios <- portfolioBacktesting(formula = Formula,
                                    data = data01ts, spec = Spec, constraints = Constraints,
                                    backtest = Backtest, trace = FALSE)


#Weights are rebalanced on a monthly basis
Weights <- round(100 * btportfolios$weights, 2)[1:60, ]
Weights

setSmootherLambda(btportfolios$backtest) <- "1m"
SmoothPortfolios <- portfolioSmoothing(object = btportfolios,trace = FALSE)
smoothWeights <- round(100 * SmoothPortfolios$smoothWeights,2)[1:60, ]
smoothWeights

backtestPlot(SmoothPortfolios, cex = 0.6, font = 1, family = "mono")

netPerformance(SmoothPortfolios)

## --------------------
#  MODEL-SIMULATION
## --------------------

## ----------------------------------
# model specification and estimation
mod_TAN <- lm(berndt.df$TAN ~ 1)
summary(mod_TAN)
```

```r

## ----------------------------------
# parameter calibration and choice of the number of replications in the simulation and of the sample size for simulated data
vol <- sd(mod_TAN$residuals)
alpha <- mod_TAN$coefficients[[1]]
nrep <- 1000
TT <- nrow(berndt.df)
# here I create the containers to be filled with the generated data.
y_bt <- y_mc <- array(1, c(TT, nrep))
x_bt <- x_mc <- array(alpha, c(TT, nrep))

# now, the loop

for (i in 1:nrep){
  u <- rnorm(TT)
  res <- sample(mod_TAN$residuals, replace = T) # this (re)samples from the data

  x_mc[, i] <- alpha + vol * u # the Monte Carlo way
  x_bt[, i] <- alpha+res             # the bootstrap way

  # now we simply construct and store the bootstrapped and MC cumulative returns
  for (j in 2:TT){
    y_mc[j, i] <- y_mc[j-1, i] * (1 + x_mc[j, i])
    y_bt[j, i] <- y_bt[j-1, i] * (1 + x_bt[j, i])
  }
}

# now we want to construct the series of means and quantiles of the resulting collection of drawn series
for (i in 1:TT){
  # obtaining the means
  berndt.df$y_bt_mean[i] <- mean(y_bt[i, ])
  berndt.df$x_bt_mean[i] <- mean(x_bt[i, ])
  berndt.df$y_mc_mean[i] <- mean(y_mc[i, ])
  berndt.df$x_mc_mean[i] <- mean(x_mc[i, ])

  # and the quantiles
  berndt.df$y_bt_q05[i] <- quantile(y_bt[i, ], 0.05)
  berndt.df$x_bt_q05[i] <- quantile(x_bt[i, ], 0.05)
  berndt.df$y_mc_q05[i] <- quantile(y_mc[i, ], 0.05)
  berndt.df$x_mc_q05[i] <- quantile(x_mc[i, ], 0.05)

  berndt.df$y_bt_q95[i] <- quantile(y_bt[i, ], 0.95)
  berndt.df$x_bt_q95[i] <- quantile(x_bt[i, ], 0.95)
  berndt.df$y_mc_q95[i] <- quantile(y_mc[i, ], 0.95)
  berndt.df$x_mc_q95[i] <- quantile(x_mc[i, ], 0.95)

}

```

```
## ---------------------------------------------
# plotting
plot <- ggplot(data= berndt.df, aes(x = date)) +
  geom_line(aes(y = TAN, color = "TAN"), size = 1) +
  geom_line(aes(y = x_mc_mean, color = "x_mc_mean"), size = 1) +
  geom_line(aes(y = x_mc_q05, color = "x_mc_q05"), size = 1) +
  geom_line(aes(y = x_mc_q95, color = "x_mc_q95"), size = 1) +
  labs(title = "Simulation",
       x = "Time", y = "Monthly Returns") +
  scale_color_manual(values = c("TAN" = "blue", "x_mc_q05" = "red","x_mc_q95" = "red","x_mc_mean" = "green")) +
  theme_minimal() +
  theme(axis.line = element_line(color = "black"))

## -------------------------------------------
# Value at Risk  via Monte Carlo simulation
## -------------------------------------------
s1_mc=x_mc[2,]
hist(s1_mc, breaks = seq(min(s1_mc), max(s1_mc), l = 20+1),prob=TRUE, main = "histogram of monthly returns")
curve(dnorm(x,mean=mean(s1_mc),sd=sd(s1_mc)),col='darkblue',lwd=2,add=TRUE)
VaR_mc <- quantile(s1_mc, 0.05)
VaR_mc
```

Chapter 5

Factor Models for Asset Prices and Returns

5.1 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.

5.2 Factor Models:Time-Series Representation

The statistical distribution of excess returns on N assets (i=1...n) can be conditioned on a vector of K factors **f** (where N is large and K is small)

$$\begin{aligned} r^i_{t,t+k} &= \gamma^i_0 + \gamma^{i\prime}_1 \mathbf{f}_{t,t+k} + v^i_{t,t+k} \\ \mathbf{f}_{t,t+k} &= \mu^f + \mathbf{H}^f \epsilon_{t,t+k} \\ \Sigma^f &= \mathbf{H}^f \mathbf{H}^{f\prime}. \\ \mathbf{E}\left(v^i_{t,t+k}, v^j_{t,t+k}\right) &= 0 \\ \mathbf{E}\left(v^i_{t,t+k}, \epsilon^j_{t,t+k}\right) &= 0 \\ \epsilon_{t+k} &\sim \mathcal{D}\left(\mathbf{0}, \mathbf{I}\right) \end{aligned} \tag{5.1}$$

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 $\gamma^{i\prime}_1 \mathbf{f}$ and an idiosyncratic component captured by the residuals of the projection of returns on factors $v^i_{t,t+k}$. 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:

$$\begin{aligned} E\left(\mathbf{f}_{t,t+k}\right) &= \mu^f \\ E\left(r^i_{t,t+k}\right) &= \gamma^i_0 + \gamma^{i\prime}_1 E\left(\mathbf{f}_{t,t+k}\right) \\ &= \gamma^i_0 + \gamma^{i\prime}_1 \mu^f \end{aligned} \tag{5.2}$$

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

5.3 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 $t+k$

$$\begin{aligned}
\underset{(Nx1)}{\mathbf{r}_{t,t+k}} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B}\underset{(Kx1)}{\mathbf{f}_{t,t+k}} + \underset{(Nx1)}{\mathbf{v}_t} \\
\underset{(Kx1)}{\mathbf{f}_{t,t+k}} &= \underset{(Kx1)}{\mu^f} + \underset{(KxK)}{\mathbf{H}^f}\underset{(Kx1)}{\epsilon^f} \\
\Sigma^v &= \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ .. & .. & .. & .. \\ 0 & 0 & 0 & \sigma_n \end{bmatrix} \\
\Sigma^f &= \mathbf{H}^f\mathbf{H}^{f\prime}.
\end{aligned} \tag{5.3}$$

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

$$\begin{aligned}
\underset{(Nx1)}{E}\mathbf{r}_{t,t+k} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B}\underset{(Kx1)}{\mu^f} \\
\underset{(NxN)}{\Sigma}{}^r &= \underset{(NxK)}{B}\underset{(KxK)}{\Sigma}{}^f\underset{(KxN)}{B'} + \underset{(NxN)}{\Sigma}{}^v
\end{aligned} \tag{5.4}$$

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

5.4 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 $r_{t+1}^p = \sum_{i=1}^N w_i r_{t+1}^i$

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

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

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

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

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.

5.5 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:

$$
\begin{aligned}
\mathbf{r}_t &= \beta_{\mathbf{0}} + \beta_{\mathbf{1}} r_t^m + \mathbf{u}_t \\
r_t^m &= E\left(r^m\right) + \sigma_m \mathbf{u}_{m,t} \\
\Sigma &= \beta_{\mathbf{1}} \beta_{\mathbf{1}}{}' \sigma_m^2 + \Sigma_u \\
\mu &= \beta_0 + \beta_1 E\left(r^m\right)
\end{aligned}
$$

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 and μ, Σ can be obtained with the estimation of 3N+2 parameters when the CAPM is adopted.

5.5.1 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.

```
# elaboration on the original produced by E.Zivot by C. Favero
# author: Carlo Favero
# created: July, 2021
#
# comments: Original Examples follow chapter 11 in Zivot and Wang (2006)


rm(list=ls()) #Removes all items in Environment!
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# set output options
# options(width = 70, digits=4)
listofpackages <- c("dygraphs",
    "dplyr","ellipse","reshape2","ggplot2","PerformanceAnalytics","zoo")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
```

```
  library(j, character.only = T)
}
install.packages("fEcofin", repos="http://R-Forge.R-project.org")
# load required packages
library(fEcofin)                # various data sets

#########################
# Data Loadings and Transform Descriptive Analysis
##########################

# create data frame with dates as rownames
berndt.df = berndtInvest[, -1]
berndt.df$date <- as.Date(berndtInvest[, 1])
rownames(berndt.df) = as.character(berndtInvest[, 1])
colnames(berndt.df)
dimnames(berndt.df)[[2]] #command alternative to the previous one

# transform the data and compute cumulative returns

t0 <- which(berndt.df$date == "1978-01-01")
t1 <- which(berndt.df$date == "1987-12-01")

series_names <- c("CITCRP","CONED","CONTIL","DATGEN","DEC","DELTA","GENMIL","GERBER","IBM",
"MARKET","MOBIL","PANAM","PSNH","TANDY","TEXACO","WEYER","RKFREE")

for (name in series_names) {
  P_col_name <- paste0(name,"_P")
  LP_col_name<- paste0("L",P_col_name)
  berndt.df[t0, P_col_name] <- 1
  for (i in (t0+1):(t1)) {
    berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] * (1+berndt.df[i, name][[1]] )
  }
  berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
}
# add a trend to the database
berndt.df$TREND<- array(data = NA, dim = nrow(berndt.df))
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

####################################
# Descriptive Analysis of prices and returns
####################################
# plot log prices
ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = LTEXACO_P), color = "blue", size = 1, linetype = "solid") +
  geom_line(aes(y = LMARKET_P), color = "green", size = 1, linetype = "solid") +
  labs(x = "Date", y = "Portfolios TEXACO and MKT") +
```

```
  ylim(-0.5, 2) +
  theme_minimal() +
  theme(
    legend.position = "topleft",
    legend.title = element_blank(),
    legend.text = element_text(size = 12),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    plot.title = element_text(size = 16, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
  ) +
  guides(fill = guide_legend(override.aes = list(size = 2)))
# plot returns
ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = TEXACO), color = "blue", size = 1, linetype = "solid") +
  geom_line(aes(y = MARKET), color = "green", size = 1, linetype = "solid")
      +
  labs(x = "Date", y = "Portfolios TEXACO and MKT") +
  ylim(-0.45, 0.45) +
  theme_minimal() +
  theme(
    legend.position = "topleft",
    legend.title = element_blank(),
    legend.text = element_text(size = 12),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    plot.title = element_text(size = 16, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    guide = guide_legend(override.aes = list(size = 2, linetype = "solid"))
  ) +
  guides(fill = guide_legend(override.aes = list(size = 2)))



################################
# CAPM FOR TEXACO
################################
capm_tex<-lm(TEXACO ~ MARKET, data=berndt.df)
summary(capm_tex)
berndt.df$TEXACO_fitted<-capm_tex$fitted.values

#fitting returns

plot(berndt.df$date[t0:t1],berndt.df$TEXACO[t0:t1], col = 'blue', type =
    "l",
      ylab = " actual and fitted returns", xlab = "date",lwd = 2,)
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")
lines(y = berndt.df$TEXACO_fitted[t0:t1], x = berndt.df$date[t0:t1], col =
    "green",lwd = 2)
legend("topright", legend = c("TEXACO ACTUAL", "TEXACO FITTED"),
       col = c("blue", "green"), lty = 1)
grid(nx = 6, ny = 7, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)

#fitting prices
berndt.df$TEXACO_P_FITTED <-  array(data = NA, dim = nrow(berndt.df))
berndt.df$TEXACO_P_FITTED[t0] <- 1
for (i in (t0+1):(t1)) {
  berndt.df$TEXACO_P_FITTED[i] <- berndt.df$TEXACO_P_FITTED[i-1] * (1 +
      berndt.df$TEXACO_fitted[i])}

plot(berndt.df$date[t0:t1],berndt.df$TEXACO_P[t0:t1], col = 'blue', type =
    "l",
     ylab = " actual and fitted prices", xlab = "date",lwd =
         2,ylim=c(0.9,5))
#lines(y = rep(mean(berndt.df$TEXACO[t0:t1], na.rm = T),
    length(berndt.df$TEXACO[t0:t1])), x = berndt.df$date[t0:t1], col =
    "red")
lines(y = berndt.df$TEXACO_P_FITTED[t0:t1], x = berndt.df$date[t0:t1], col
    = "green",lwd = 2)
legend("topleft", legend = c("TEXACO ACTUAL", "TEXACO FITTED"),
       col = c("blue", "green"), lty = 1)
grid(nx = 6, ny = 7, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
#dev.copy2pdf(width = 8.5, out.type = "pdf",file="CAPM.pdf")
#dev.off()
####################
# Optimal Portfolio weights with the CER approach
###################

returns.df=berndt.df[, c(1:9,11:16)]
#returns.df = berndt.df[, c(-10, -17)
exreturns.df=returns.df-berndt.df$RKFREE
returns.mat = as.matrix(exreturns.df)
n.obs = nrow(returns.mat)

#Estimation of CER model parameters
cov.sample=var(returns.mat)
mu = matrix(colMeans(returns.mat), nrow = ncol(returns.mat), ncol = 1)
#
# compute global min variance portfolio
#
w.gmin.sample = solve(var(returns.mat))%*%rep(1,nrow(cov.sample))
w.gmin.sample = w.gmin.sample/sum(w.gmin.sample)
colnames(w.gmin.sample) = "sample"
barplot(t(w.gmin.sample), horiz=F, main="Weights", col="blue", cex.names =
    0.75, las=2)
```

```
################################
# A Single index model: the CAPM
################################

##
## use multivariate regression and matrix algebra
##
returnsmkt.df=berndt.df[, c(10:10,17:17)]
#returns.df = berndt.df[, c(-10, -17)
returnsmkt.df$EXRETMKT=returnsmkt.df$MARKET-returnsmkt.df$RKFREE
market.mat = as.matrix(returnsmkt.df[,3, drop=F])
n.obs = nrow(returns.mat)
X.mat = cbind(rep(1,n.obs),market.mat)
colnames(X.mat)[1] = "intercept"
XX.mat = crossprod(X.mat)

# multivariate least squares
G.hat = solve(XX.mat)%*%crossprod(X.mat,returns.mat)
# can also use solve(qr(X.mat), returns.mat)
beta.hat = G.hat[2,]
E.hat = returns.mat - X.mat%*%G.hat
#D.hat=crossprod(E.hat)
diagD.hat = diag(crossprod(E.hat)/(n.obs-2))
# compute R2 values from multivariate regression
sumSquares = apply(returns.mat, 2, function(x) {sum( (x - mean(x))^2 )})
R.square = 1 - (n.obs-2)*diagD.hat/sumSquares

# print and plot results
cbind(beta.hat, diagD.hat, R.square)

par(mfrow=c(1,2))
barplot(beta.hat, horiz=T, main="Beta values", col="blue", cex.names = 0.75, las=1)
barplot(R.square, horiz=T, main="R-square values", col="blue", cex.names = 0.75, las=1)
par(mfrow=c(1,1))

# compute single index model covariance/correlation matrices
cov.si = as.numeric(var(market.mat))*beta.hat%*%t(beta.hat) + diag(diagD.hat)
cor.si = cov2cor(cov.si)
#
# COMPARE CORRELATIONS
#
# FACTOR MODEL BASED CORRELATION MATRIX using plotcorr() from ellipse package
#
rownames(cor.si) = colnames(cor.si)
ord <- order(cor.si[1,])
ordered.cor.si <- cor.si[ord, ord]
plotcorr(ordered.cor.si, col=cm.colors(11)[5*ordered.cor.si + 6])
plotcorr(cor.si, col=cm.colors(11)[5*cor.si + 6])
```

```
#
# SAMPLE CORRELATION MATRIX
#
cor.sample = cor(returns.mat)
ord <- order(cor.sample[1,])
ordered.cor.sample <- cor.sample[ord, ord]
plotcorr(ordered.cor.sample, col=cm.colors(11)[5*ordered.cor.sample + 6])
plotcorr(cor.sample, col=cm.colors(11)[5*cor.sample + 6])
#
# CAPM residuals  CORRELATION MATRIX
#
cor.resid = cor(E.hat)
ord <- order(cor.resid[1,])
ordered.cor.resid <- cor.resid[ord, ord]
plotcorr(ordered.cor.resid, col=cm.colors(11)[5*ordered.cor.resid + 6])
#
# compute global min variance portfolio
#
# use CAPM covariance (1-factor model)
w.gmin.si = solve(cov.si)%*%rep(1,nrow(cov.si))
w.gmin.si = w.gmin.si/sum(w.gmin.si)
colnames(w.gmin.si) = "single.index"


#par(mfrow=c(2,1))
#barplot(t(w.gmin.si), horiz=F, main="Single Index Weights", col="blue",
    cex.names = 0.75, las=2)
#barplot(t(w.gmin.sample), horiz=F, main="Sample Weights", col="blue",
    cex.names = 0.75, las=2)
#par(mfrow=c(1,1))


#compare weights delivered by the two alternative methods
pdf("output.pdf", width = 10, height = 8)
par(mfrow = c(2, 1))
barplot(t(w.gmin.si), horiz=F, main="Single Index Weights", col="blue",
    cex.names = 0.75, las=2)
barplot(t(w.gmin.sample), horiz=F, main="Sample Weights", col="blue",
    cex.names = 0.75, las=2)
par(mfrow = c(1, 1))
dev.off()



# compare means and sd values on global min variance portfolios
mu.vals = colMeans(returns.mat)
mu.gmin.si = as.numeric(crossprod(w.gmin.si, mu.vals))
sd.gmin.si = as.numeric(sqrt(t(w.gmin.si)%*%cov.si%*%w.gmin.si))
mu.gmin.sample = as.numeric(crossprod(w.gmin.sample, mu.vals))
sd.gmin.sample =
    as.numeric(sqrt(t(w.gmin.sample)%*%var(returns.mat)%*%w.gmin.sample))
cbind(mu.gmin.si,mu.gmin.sample, sd.gmin.si, sd.gmin.sample)
```

```
########################################
## AN ALTERNATIVE APPROACH to compute parameters in CAPM:
## use lm function to compute single index model regressions for each asset
#######################################

asset.names = colnames(returns.mat)
asset.names

# initialize list object to hold regression objects

reg.list = list()
# loop over all assets and estimate time series regression
for (i in asset.names) {
 reg.df = berndt.df[, c(i, "MARKET")]
 si.formula = as.formula(paste(i,"~", "MARKET", sep=" "))
 reg.list[[i]] = lm(si.formula, data=reg.df)
}

# examine the elements of reg.list  - they are lm objects!
names(reg.list)
class(reg.list$CITCRP)
reg.list$CITCRP
summary(reg.list$CITCRP)

# plot actual vs. fitted over time
# use chart.TimeSeries() function from PerformanceAnalytics package

dataToPlot = cbind(fitted(reg.list$CITCRP), berndt.df$CITCRP)
colnames(dataToPlot) = c("Fitted","Actual")
dev.off()

# Verify the data
str(dataToPlot)
summary(dataToPlot)

# Create the time series chart
chart.TimeSeries(dataToPlot, main = "Single Index Model for CITCRP",
                 colorset = c("black", "blue"), legend.loc = "bottomleft")


# scatterplot of the single index model regression
plot(berndt.df$MARKET, berndt.df$CITCRP, main="SI model for CITCRP",
     type="p", pch=16, col="blue",
     xlab="MARKET", ylab="CITCRP")
abline(h=0, v=0)
abline(reg.list$CITCRP, lwd=2, col="red")

## extract beta values, residual sd's and R2's from list of regression objects
## brute force loop
```

```r
reg.vals = matrix(0, length(asset.names), 3)
rownames(reg.vals) = asset.names
colnames(reg.vals) = c("beta", "residual.sd", "r.square")
for (i in names(reg.list)) {
    tmp.fit = reg.list[[i]]
    tmp.summary = summary(tmp.fit)
    reg.vals[i, "beta"] = coef(tmp.fit)[2]
    reg.vals[i, "residual.sd"] = tmp.summary$sigma
    reg.vals[i, "r.square"] = tmp.summary$r.squared
}
reg.vals

# alternatively use R apply function for list objects - lapply or sapply
extractRegVals = function(x) {
# x is an lm object
 beta.val = coef(x)[2]
 residual.sd.val = summary(x)$sigma
 r2.val = summary(x)$r.squared
 ret.vals = c(beta.val, residual.sd.val, r2.val)
 names(ret.vals) = c("beta", "residual.sd", "r.square")
 return(ret.vals)
}
reg.vals = sapply(reg.list, FUN=extractRegVals)
t(reg.vals)
```

5.6 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

$$r^i_{t+1} = \alpha_1 + \beta_i^{f^1} f^1_{t+1} + \beta_i^{f^2} f^2_{t+1} + \cdots + \beta_i^{f^k} f^k_{t+1} + v_{t+1} \tag{5.5}$$

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

$\boldsymbol{\beta}^{f1}$, $\boldsymbol{\beta}^{f2}$,..., $\boldsymbol{\beta}^{fk}$. 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:

$$E(\mathbf{r}) = \gamma_0 + \gamma_1 \boldsymbol{\beta}_{f1} + \gamma_2 \boldsymbol{\beta}_{f2} + \cdots + \gamma_k \boldsymbol{\beta}_{fk} + \mathbf{u}$$

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

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

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.

5.6.1 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)

5.7 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:

$$r_{i,t+1} = \alpha_i + \beta_i' \mathbf{f}_{t+1} + v_{i,t+1}, \tag{5.6}$$

$$\mathbf{f}_{t+1} = E\left(\mathbf{f}_{t+1} \mid I_t\right) + \epsilon_{t+1} \qquad \text{with } \epsilon_{t+1} \sim \mathcal{D}\left(\mathbf{0}, \mathbf{\Sigma}\right) \tag{5.7}$$

where $Cov\left(v_{i,t+1}, v_{j,t+1}\right) = 0$ 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 (5.6) specifies the conditional distribution of returns on factors, while equation (5.7) 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., $E(\mathbf{f}_{t+1} \mid I_t) = \mu$).

In equation (5.6) 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:

$$\ln P_{i,t} = \ln P_{i,t-1} + \mathbf{r}_{i,t} \ , \tag{5.8}$$

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:

$$\ln \mathbf{F}_t = \ln \mathbf{F}_{t-1} + \mathbf{f}_t \ . \tag{5.9}$$

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 (5.6), (5.7), (5.8), (5.9). 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 (5.6)–(5.9) 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:

$$ECT_{i,t} \equiv \ln P_{i,t} - \hat{\alpha}_{0,i} - \hat{\alpha}_{1,i}t - \hat{\beta}_i' \ln \mathbf{F}_t \ . \tag{5.10}$$

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:

$$\begin{aligned} \ln P_{i,t+1} &= \alpha_{0,i} + \alpha_{1,i}t + \beta_i' \ln \mathbf{F}_{t+1} + u_{i,t+1} \qquad (5.11) \\ u_{i,t+1} &= \rho_i u_{i,t} + v_{i,t+1} \\ \mathbf{f}_{t+1} &= E\left(\mathbf{f}_{t+1} \mid I_t\right) + \epsilon_{t+1} \\ \ln P_{i,t} &= \ln P_{i,t-1} + r_{i,t} \\ \ln \mathbf{F}_t &= \ln \mathbf{F}_{t-1} + \mathbf{f}_t \end{aligned}$$

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

By taking first differences of our model in (5.11) we obtain a novel specification for returns and factors, where asset returns relate to factors *plus* the ECT:

$$r_{i,t+1} = \alpha_{1,i} + \beta_i' \mathbf{f}_{t+1} + \underbrace{(\rho_i - 1)}_{\delta_i} \underbrace{u_{i,t}}_{\equiv ECT_{i,t}} + v_{i,t+1}. \tag{5.12}$$

Eq. (5.12) represents the Factor Error Correction Model (FECM).[1]

Two comments are in order. First, we include a linear trend in Eq. (5.11) 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

[1]The equilibrium correction representation (5.12) of cointegrated time-series (see the system of equations in (5.11)) is warranted by the Engle and Granger (1987) representation theorem.

in the long-run relation (5.11) 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 (5.12) 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. (5.12).

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 $\mid \rho_i \mid= 1$). 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

$$\begin{aligned}
\underset{(Nx1)}{\mathbf{r}_{t,t+1}} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B}\underset{(Kx1)}{\mathbf{f}_{t,t+1}} + \underset{(NxN)}{\Gamma}\underset{(Nx1)}{\mathbf{u}_t} + \underset{(Nx1)}{\mathbf{v}_t} \qquad (5.13)\\
\underset{(Kx1)}{\mathbf{f}_{t,t+1}} &= \underset{(Kx1)}{\mu^f} + \underset{(KxK)}{\mathbf{H}^f}\underset{(Kx1)}{\epsilon^f}\\
\Sigma^v &= \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ .. & .. & .. & .. \\ 0 & 0 & 0 & \sigma_n \end{bmatrix}\\
\Sigma^f &= \mathbf{H}^f\mathbf{H}^{f\prime}.
\end{aligned}$$

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 asthe variance of the one-period ahead predictive

[2]Moreover, 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

$$\begin{aligned} \underset{(Nx1)}{E}\,\mathbf{r}_{t,t+1} &= \underset{(Nx1)}{\alpha} + \underset{(NxK)}{B}\,\underset{(Kx1)}{\mu^f} + \underset{(NxN)}{\Gamma}\,\underset{(Nx1)}{\mathbf{u}_t} \\ \underset{(NxN)}{\Sigma}{}^r &= \underset{(NxK)}{B}\,\underset{(KxK)}{\Sigma}{}^f\,\underset{(KxN)}{B'} + \underset{(NxN)}{\Sigma}{}^v \end{aligned} \tag{5.14}$$

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 (5.6)-(5.7). Furthermore, looking at the short-run FECM specification in (5.12), 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.

5.7.1 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

```

# The effect of omitting long-run trends from factor models

rm(list=ls()) #Removes all items in Environment!
#setwd(path)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# set output options
options(width = 70, digits=4)
listofpackages <- c("dygraphs",
    "dplyr","ellipse","reshape2","ggplot2","PerformanceAnalytics","zoo")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}
install.packages("fEcofin", repos="http://R-Forge.R-project.org")
```

```
# load required packages
library(fEcofin)                # various data sets

########################
# Data Loadings and Transform Descriptive Analysis
########################

# create data frame with dates as rownames
berndt.df = berndtInvest[, -1]
berndt.df$date <- as.Date(berndtInvest[, 1])
rownames(berndt.df) = as.character(berndtInvest[, 1])
colnames(berndt.df)
dimnames(berndt.df)[[2]] #command alternative to the previous one

# transform the data and compute cumulative returns

t0 <- which(berndt.df$date == "1978-01-01")
t1 <- which(berndt.df$date == "1987-12-01")

series_names <- c("CITCRP","CONED","CONTIL","DATGEN","DEC","DELTA","GENMIL","GERBER",
"IBM","MARKET","MOBIL","PANAM","PSNH","TANDY","TEXACO","WEYER","RKFREE")

for (name in series_names) {
  P_col_name <- paste0(name,"_P")
  LP_col_name<- paste0("L",P_col_name)
  berndt.df[t0, P_col_name] <- 1
  for (i in (t0+1):(t1)) {
    berndt.df[i, P_col_name][[1]] <- berndt.df[i-1, P_col_name][[1]] * (1+berndt.df[i, name][[1]] )
  }
  berndt.df[, LP_col_name] <- log(berndt.df[, P_col_name])
}
# add a trend to the database
berndt.df$TREND<- array(data = NA, dim = nrow(berndt.df))
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
for (i in (t0+1):(t1)) {
  berndt.df[i, "TREND"][[1]] <- berndt.df[i-1, "TREND"][[1]] +1
}

########################
# Descriptive Analysis
########################
ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = LTEXACO_P, color = "TEXACO"), size = 1, linetype = "solid") +
  geom_line(aes(y = LMARKET_P, color = "MARKET"), size = 1, linetype = "solid") +
  labs(x = "Date", y = "Portfolios TEXACO and MKT") +
```

```
  ylim(-0.5, 2) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    labels = c("TEXACO", "MARKET")
  )


# plot returns
ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = TEXACO, color = "TEXACO"), size = 1, linetype =
      "solid") +
  geom_line(aes(y = MARKET, color = "MARKET"), size = 1, linetype =
      "solid") +
  labs(x = "Date", y = "Returns TEXACO and MKT") +
  ylim(-0.45, 0.45) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 8),
    plot.title = element_text(size = 12, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    labels = c("TEXACO", "MARKET")
  )

##############################
# Standard CAPM Factor Models
##############################

## use lm function to compute single index model regressions for each asset
##
returns.mat = as.matrix(berndt.df[, c(1:9,11:16)])
asset.names = colnames(returns.mat)
asset.names

# initialize list object to hold regression objects

reg.list = list()
# loop over all assets and estimate time series regression
```

```
for (i in asset.names) {
 reg.df = berndt.df[, c(i, "MARKET")]
 si.formula = as.formula(paste(i,"~", "MARKET", sep=" "))
 reg.list[[i]] = lm(si.formula, data=reg.df)
}

# examine the elements of reg.list  - they are lm objects!
names(reg.list)
class(reg.list$TEXACO)
reg.list$TEXACO
summary(reg.list$TEXACO)

# plot actual vs. fitted over time
# use chart.TimeSeries() function from PerformanceAnalytics package
dataToPlot = cbind(fitted(reg.list$TEXACO), berndt.df$TEXACO)
colnames(dataToPlot) = c("Fitted","Actual")
chart.TimeSeries(dataToPlot, main="Single Index Model for TEXACO",
                 colorset=c("black","blue"), legend.loc="bottomleft")

# scatterplot of the single index model regression
plot(berndt.df$MARKET, berndt.df$TEXACO, main="SI model for CITCRP",
     type="p", pch=16, col="blue",
     xlab="MARKET", ylab="TEXACO")
abline(h=0, v=0)
abline(reg.list$TEXACO, lwd=2, col="red")

## extract beta values, residual sd's and R2's from list of regression objects

reg.vals = matrix(0, length(asset.names), 3)
rownames(reg.vals) = asset.names
colnames(reg.vals) = c("beta", "residual.sd", "r.square")
for (i in names(reg.list)) {
    tmp.fit = reg.list[[i]]
    tmp.summary = summary(tmp.fit)
    reg.vals[i, "beta"] = coef(tmp.fit)[2]
    reg.vals[i, "residual.sd"] = tmp.summary$sigma
    reg.vals[i, "r.square"] = tmp.summary$r.squared
}
reg.vals

# print regression results

par(mfrow=c(1,2))
barplot(reg.vals[,1], horiz=T, main="Beta values", col="blue", cex.names = 0.75, las=1)
barplot(reg.vals[,3], horiz=T, main="R-square values", col="blue", cex.names = 0.75, las=1)
par(mfrow=c(1,1))


#######################
```

```
# CAPM in levels
#######################

## use lm function to compute single index model regressions for each asset
##
selected_columns <- c("LCITCRP_P","LCONED_P","LCONTIL_P","LDATGEN_P","LDEC_P","LDELTA_P","LGENMIL_P",
"LGERBER_P","LIBM_P","LMOBIL_P","LPANAM_P","LPSNH_P","LTANDY_P","LTEXACO_P","LWEYER_P")

# Extract the specified columns and store them in a matrix
lprices.mat <- as.matrix(berndt.df[, selected_columns])

asset.names = colnames(lprices.mat)
asset.names

# initialize list object to hold regression objects

reg1.list = list()
# loop over all assets and estimate time series regression
for (i in asset.names) {
  #reg.df = berndt.df[, c(i, "LMARKET_P")]
  si.formula = as.formula(paste(i,"~", "LMARKET_P+TREND", sep=" "))
  reg1.list[[i]] = lm(si.formula, data=berndt.df)
}

# examine the elements of reg.list  - they are lm objects!
names(reg1.list)
class(reg1.list$LTEXACO_P)
reg1.list$LTEXACO_P
summary(reg1.list$LTEXACO_P)

# plot actual vs. fitted over time
# use chart.TimeSeries() function from PerformanceAnalytics package
dataToPlot = cbind(fitted(reg1.list$LTEXACO_P), berndt.df$LTEXACO_P)
colnames(dataToPlot) = c("Fitted","Actual")
chart.TimeSeries(dataToPlot, main="Single Index Model for price TEXACO",
                 colorset=c("black","blue"), legend.loc="bottomleft")

# scatterplot of the single index model regression
plot(berndt.df$LMARKET_P, berndt.df$LTEXACO_P, main="SI model for LTEXACO_P",
     type="p", pch=16, col="blue",
     xlab="MARKET", ylab="TEXACO")
abline(h=0, v=0)
abline(reg.list$TEXACO, lwd=2, col="red")

## extract beta values, residual sd's and R2's from list of regression objects

reg.vals1 = matrix(0, length(asset.names), 3)
rownames(reg.vals1) = asset.names
colnames(reg.vals1) = c("beta", "residual.sd", "r.square")
```

```
for (i in names(reg1.list)) {
  tmp.fit = reg1.list[[i]]
  tmp.summary = summary(tmp.fit)
  reg.vals1[i, "beta"] = coef(tmp.fit)[2]
  reg.vals1[i, "residual.sd"] = tmp.summary$sigma
  reg.vals1[i, "r.square"] = tmp.summary$r.squared
}
reg.vals1

# print regression results

par(mfrow=c(1,2))
barplot(reg.vals1[,1], horiz=T, main="Beta values", col="blue", cex.names =
    0.75, las=1)
barplot(reg.vals1[,3], horiz=T, main="R-square values", col="blue",
    cex.names = 0.75, las=1)
par(mfrow=c(1,1))

########################################
# a Single Factor Model with Predictability : an illustration with TEXACO
#######################################

#Log Level linear model between LCITCRP_P TREND an MARKET
model_TEXACO_P=lm(berndt.df$LTEXACO_P ~ berndt.df$LMARKET_P+berndt.df$TREND)
summary(model_TEXACO_P)

ggplot(berndt.df, aes(x = date)) +
  geom_line(aes(y = LTEXACO_P, color = "TEXACO"), size = 1, linetype =
      "solid") +
  geom_line(aes(y = fitted(model_TEXACO_P), color = "Fitted"), size = 1,
      linetype = "solid") +
  labs(x = "Date", y = "Actual and Fitted") +
  ylim(-0.5, 2) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    labels = c("TEXACO", "Fitted")
  )


#store log level residuals as u and test for their stationarity
u_TEXACO=as.matrix(model_TEXACO_P$residuals)
DuTEXACO=diff(u_TEXACO,lag=1)
model_DuTEXACO=lm(DuTEXACO ~  u_TEXACO[1:(nrow(u_TEXACO)-1)]-1)
```

```
summary(model_DuTEXACO)
D12uTEXACO=diff(u_TEXACO,lag=12)
model_D12uTEXACO=lm(D12uTEXACO ~  u_TEXACO[1:(nrow(u_TEXACO)-12)]-1)
summary(model_D12uTEXACO)
##############################

#Compute Log Returns for 1M ahead timespan (1 months)
logret1M_TEXACO=diff(berndt.df$LTEXACO_P,lag=1)
logret1M_MARKET=diff(berndt.df$LMARKET_P,lag=1)

#Compute Log Returns for 1Y ahead timespan (12 months)
logret1Y_TEXACO=diff(berndt.df$LTEXACO_P,lag=12)
logret1Y_MARKET=diff(berndt.df$LMARKET_P,lag=12)

#############################

#model the regression on log returns appending the u residuals as another
    variable
model_d_TEXACO_1M=lm(logret1M_TEXACO ~
    logret1M_MARKET+u_TEXACO[1:(nrow(u_TEXACO)-1)])
summary(model_d_TEXACO_1M)

model_d_TEXACO_1Y=lm(logret1Y_TEXACO ~
    logret1Y_MARKET+u_TEXACO[1:(nrow(u_TEXACO)-12)])
summary(model_d_TEXACO_1Y)

model_d_TEXACO_1Y_CAPM=lm(logret1Y_TEXACO ~ logret1Y_MARKET)
summary(model_d_TEXACO_1Y_CAPM)
```

Chapter 6

Models for Risk Measurement

6.1 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:

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

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 riskChristoffersen (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

6.1.1 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:

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

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

6.2 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.

6.2.1 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

6.2.2 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.

6.3 The Evidence from high-frequency data

Figure 6.1 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

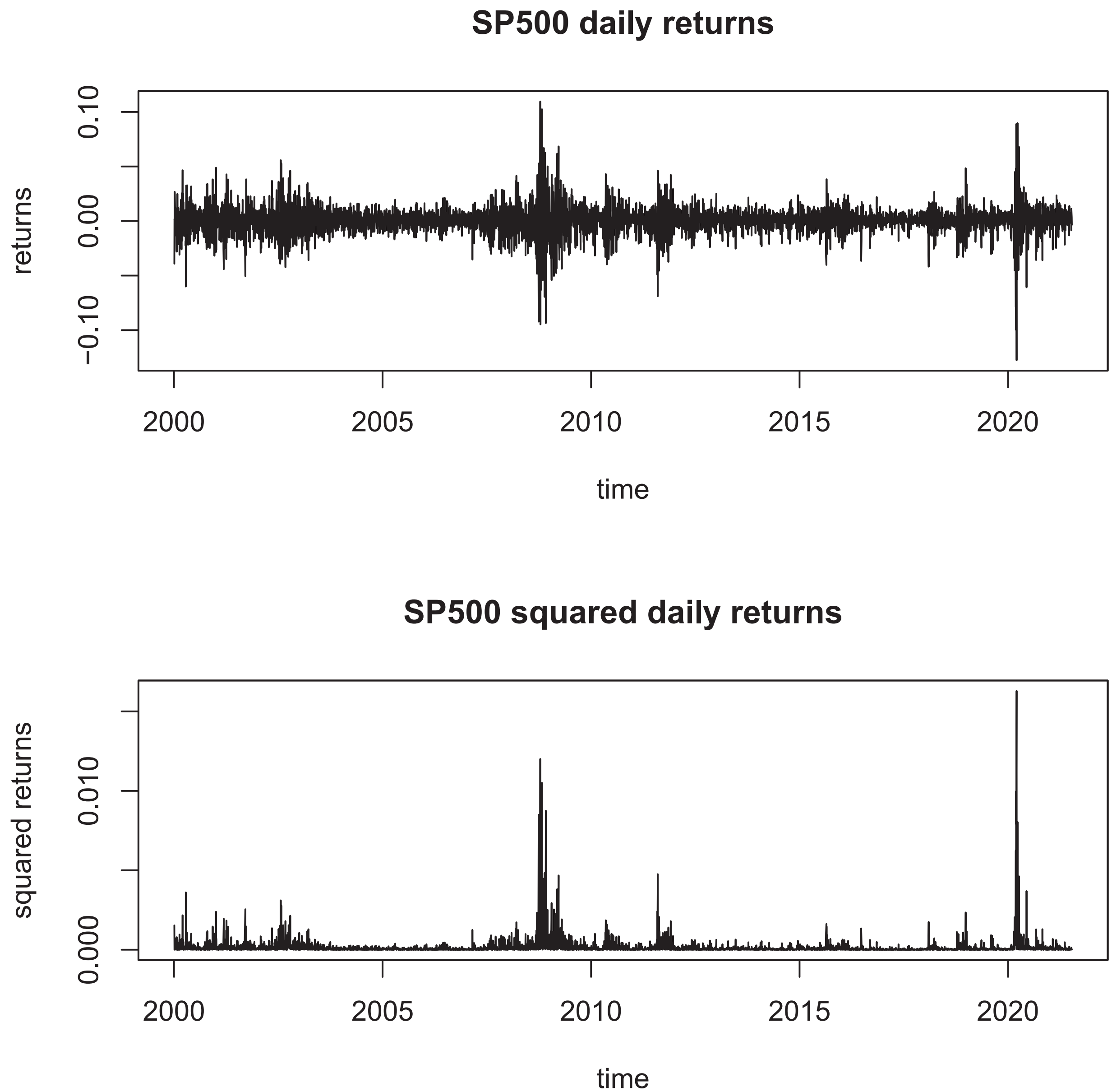

FIGURE 6.1. **Daily SP500 Returns and Squared Returns**

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.

6.4 A general model for high-frequency data

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

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

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 $\mathcal{N}(0,1)$

6.4.1 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:

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

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 $j \geqslant 1$. Where μ_t is often, but not necessarily, set to 0.

6.4.2 GARCH Properties

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

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

Substituting ω out of the GARCH expression:

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

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

6.4.3 GARCH Forecasting

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

6.4.4 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 $\hat{z}_{t-1}^2$, $\hat{z}_{t+2}^2$, ..., $\hat{z}_{t-q}^2$ (e_t is a white noise shock):

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

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

6.5 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

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

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:

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

taking logs

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

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

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

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

6.5.1 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 defendable, 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 $z_{t+1} \sim$ IID $\mathcal{D}(0,1)$, the shocks will have to be anyway IID: you can just do without normality, but the convenience of $z_{t+1} \sim$ 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.

6.6 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

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

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.

6.6.1 GARCH with factors

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

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

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

6.7 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).

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

```r
listofpackages <- c("tidyverse","dygraphs", "rugarch",
    "forecast","dplyr","ellipse","reshape2","ggplot2","xts","xlsx","readxl")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

# setting the seed for replication
set.seed(77)

raw_data            = read_xlsx("../data/2023_monthly_stocks.xlsx")
names(raw_data)[1] = 'Date'
typeof(raw_data)
typeof(raw_data$Date)
typeof(raw_data$AXP)
typeof(raw_data$CSCO)

dates <-seq(as.Date("1985-02-01"),length=462, by="months")
params <- c("Date","BA", "DJI")
data <- raw_data[, c(params)]
data<- na.omit(data)
data <- data %>%
  mutate(Date = as.Date(Date, format = "%Y-%m-%d"))

params1 <- c("BA", "DJI")
tsdata <- xts(raw_data[, c(params1)], order.by=dates) # creates a time
    series object
tsdata <- na.omit(tsdata) # omitting the rows with NA presence
data<- na.omit(data)
## having created the database with all observation we generate a subset
#tsdata1 <- tsdata["1992-02-01/1993-02-01"]
#data=subset(data,select=c(1:12))

## --------------------
# DATA TRANSFORMATIONS
## --------------------
#1. from prices to returns
# exact monthly returns
t1<-nrow(data)
data$BA_ret <- data$DJI_ret <- array(data = NA, dim = t1)
for (i in 2:t1) {
  data[i, "BA_ret"][[1]]=(data[i, "BA"][[1]]-data[i-1,
      "BA"][[1]])/data[i-1, "BA"][[1]]
  data[i, "DJI_ret"][[1]]=(data[i, "DJI"][[1]]-data[i-1,
      "DJI"][[1]])/data[i-1, "DJI"][[1]]
}


# same in .xts
```

```r
t1<-nrow(tsdata)

tsdata$BA_ret <- tsdata$DJI_ret<- array(data = NA, dim = t1)

for (i in 2:t1) {
  tsdata[i, "BA_ret"][[1]]=(tsdata[i, "BA"][[1]]-tsdata[i-1, "BA"][[1]])/data[i-1, "BA"][[1]]
  tsdata[i, "DJI_ret"][[1]]=(tsdata[i, "DJI"][[1]]-tsdata[i-1, "DJI"][[1]])/data[i-1, "DJI"][[1]]

}

## ----------------------------------
## VAR with CER-CAPM
## ----------------------------------


## -----------------------------------
## MODEL SPECIFICATION AND ESTIMATION
## -----------------------------------
start_date <- as.Date("1992-03-01")  # Replace with your start date
end_date <- as.Date("2005-12-01")    # Replace with your end date

# Extract observations between 'start_date' and 'end_date'

data_est <- subset(x = data,  Date >= start_date & Date <= end_date)

# estimation
cer_mkt <- lm(data_est$DJI_ret ~ 1)
capm_BA <- lm(data_est$BA_ret ~ data_est$DJI_ret)
summary(cer_mkt)
summary(capm_BA)

## ------------------
## MODEL SIMULATION
## ------------------

tt <- as.Date("2006-01-01")
tT <- as.Date("2015-12-01")
data_sim <- subset(x = data,  Date >= tt & Date <= tT)

# creating the containers
nrep <- 1000
BA_bt_2 <- mkt_bt_2 <- array(0, c(length(data_sim$DJI_ret), nrep))

# resampling the residuals
res_mkt_bt_2 <- matrix(sample(resid(cer_mkt), size = length(data_sim$DJI_ret) * nrep, replace = T),
                     nrow = length(data_sim$DJI_ret), ncol = nrep)
res_BA_bt_2 <- matrix(sample(resid(capm_BA), size = length(data_sim$DJI_ret) * nrep, replace = T),
                    nrow = length(data_sim$DJI_ret), ncol = nrep)
```

```


# the loop
for (i in 1:nrep){
  for (j in 1:length(data_sim$DJI_ret)){
    mkt_bt_2[j, i] <- coef(cer_mkt)[1] + res_mkt_bt_2[j, i]
    BA_bt_2[j, i] <- coef(capm_BA)[1] + coef(capm_BA)[2] * mkt_bt_2[j, i] +
        res_BA_bt_2[j, i]
  }
}

# the quantiles
var_BA_capm <- array(0, length(data_sim$DJI_ret))
for (j in 1:length(data_sim$DJI_ret)){
  var_BA_capm[j] <- quantile(BA_bt_2[j, ], probs = 0.05)
}
data_sim$var_BA_capm<-var_BA_capm
# plotting
ggplot(data_sim, aes(x = Date)) +
  geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
  geom_line(aes(y = var_BA_capm, color = "VaR"), size = 1, linetype =
      "solid") +
  labs(x = "Date", y = "Returns and VaR") +
  ylim(-0.15, 0.15) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green"),
    labels = c("BA", "VaR")
  )



## -------------------
## GARCH MODELLING
## -------------------

# the market GARCH regression
## specification
mkt_garch <- ugarchspec(variance.model = list(garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 0)))
## estimation
mkt_garchfit <- ugarchfit(mkt_garch, data = data_est$DJI_ret)
mkt_garchfit

```

```

# forecasting and plotting the results
horizon <- 10*12 # ten years
mygarchforecast <- ugarchforecast(mkt_garchfit, n.ahead = 10*12)

plotdata <- cbind(mygarchforecast@forecast$seriesFor,
                  mygarchforecast@forecast$seriesFor +
                      mygarchforecast@forecast$sigmaFor*1.96,
                  mygarchforecast@forecast$seriesFor -
                      mygarchforecast@forecast$sigmaFor*1.96)
colnames(plotdata) <- c("mean", "upper", "lower")
dygraph(ts(plotdata, start = c(2006,1), frequency = 12), main = "Forecast
    of the mean") %>%
  dySeries(c("lower", "mean", "upper"))

plotdata2 <- as.matrix(mygarchforecast@forecast$sigmaFor^2)
colnames(plotdata2) <- "var"
dygraph(ts(plotdata2, start = c(2006, 1), frequency = 12), main = "Forecast
    of the variance")


## ---------------------
## GARCH SIMULATION
## ---------------------

## coefficients
gamma0 <- coef(mkt_garchfit)[1]
omega0 <- coef(mkt_garchfit)[2]
omega1 <- coef(mkt_garchfit)[3]
omega2 <- coef(mkt_garchfit)[4]
sigma2 <- sigma(mkt_garchfit) # this constructs the series of standard
    deviations conditional on information at "t-1". Is thus a vector.

# the CAPM
## estimation
capm_BA <- lm(data_est$BA_ret ~ data_est$DJI_ret)
summary(capm_BA)

beta0 <- coef(capm_BA)[1]
beta1 <- coef(capm_BA)[2]



## -----------------
# simulation
# output containers
nrep <- 1000
BA_bt <- mkt_bt<- sigma<- array(0, c(length(data_sim$DJI_ret), nrep))

# extracting the errors and resampling

res_mkt <- as.numeric(residuals(mkt_garchfit, standardize = T)) # the
```

```
standardized residuals from the market equation

res_mkt_bt <- matrix(sample(res_mkt, size = length(data_sim$DJI_ret) * nrep, replace = T),
                     nrow = length(data_sim$DJI_ret), ncol = nrep)
res_BA_bt <- matrix(sample(resid(capm_BA), size = length(data_sim$DJI_ret) * nrep, replace = T),
                    nrow = length(data_sim$DJI_ret), ncol = nrep)

# initial values
mkt_bt[1, ] <- data_sim$DJI_ret[1]
BA_bt[1, ] <- beta0 + beta1*mkt_bt[1, ]
sigma[1, ]<- ugarchforecast(mkt_garchfit)@forecast$sigmaFor[1] #takes the first value (the one step ahead)
# the loop
for (i in 1:nrep){
  for (j in 2:length(data_sim$DJI_ret)){
    sigma[j, i] <- sqrt(omega0+omega1*( data_sim$DJI_ret[j-1]- gamma0)^2 + omega2*(sigma[j-1, i])^2)
    mkt_bt[j, i] <- gamma0 + res_mkt_bt[j, i] * sigma[j,i]
    BA_bt[j, i] <- beta0 + beta1 * mkt_bt[j, i] + res_BA_bt[j, i]
  }
}

# getting the quantiles
var_BA_garch <- array(0, length(data_sim$DJI_ret))
for (j in 1:length(data_sim$DJI_ret)){
  var_BA_garch[j] <- quantile(BA_bt[j, ], probs = 0.05)
}

data_sim$var_BA_garch<-var_BA_garch

# plotting
tt <- as.Date("2006-02-01")
tT <- as.Date("2015-12-01")
data_simplot <- subset(x = data_sim,  Date >= tt & Date <= tT)
ggplot(data_simplot, aes(x = Date)) +
  geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
  geom_line(aes(y = var_BA_capm, color = "Var CAPM"), size = 1, linetype = "solid") +
  geom_line(aes(y = var_BA_garch, color = "VaR GARCH"), size = 1, linetype = "solid") +
  labs(x = "Date", y = "Returns and VaR") +
  ylim(-0.30, 0.20) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
```

```
  ) +
  scale_color_manual(
    values = c("blue", "green", "red"),
    labels = c("BA", "VaR CAPM", "Var GARCH")
  )

save(data_simplot,file="VaRdata.Rdata")
```

6.8 Backtesting VaR

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

$$Min(R_{t+1} - VaR^p_{t+1}, 0)$$

(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*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.

6.8.1 Unconditional Coverage Testing

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

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

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.

6.8.2 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:

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

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

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

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

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

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

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

6.8.3 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:

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

note that

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

6.8.4 Backtesting VaR in R

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

```
rm(list=ls())
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
# packages used
listofpackages <- c("tidyverse","dygraphs", "rugarch",
    "forecast","dplyr","ellipse","reshape2","ggplot2","xts","xlsx","readxl")

for (j in listofpackages){
  if(sum(installed.packages()[, 1] == j) == 0) {
    install.packages(j)
  }
  library(j, character.only = T)
}

# loading the  databases
load("VaRdata.Rdata")
ggplot(data_simplot, aes(x = Date)) +
  geom_line(aes(y = BA_ret, color = "BA"), size = 1, linetype = "solid") +
  geom_line(aes(y = var_BA_capm, color = "Var CAPM"), size = 1, linetype =
      "solid") +
```

```
  geom_line(aes(y = var_BA_garch, color = "VaR GARCH"), size = 1, linetype = "solid") +
  labs(x = "Date", y = "Returns and VaR") +
  ylim(-0.30, 0.20) +
  theme_minimal() +
  theme(
    legend.position = c(0.15, 0.95),  # Set the legend position (top-left)
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 10),
    plot.title = element_text(size = 12, hjust = 0.5)
  ) +
  scale_color_manual(
    values = c("blue", "green", "red"),
    labels = c("BA", "VaR CAPM", "Var GARCH")
  )
## ---------------
# VaR tail
alpha <- 0.1


## --------------
violations <- (data_simplot$BA_ret - data_simplot$var_BA_capm) < 0
table(violations)
plot(y = violations*runif(length(violations), min = 0.99, max = 1.01), x = data_simplot$Date, main = "VaR violations",
     ylab = "violations", xlab = "time") # adding jitter to make sure that adjacent observations don’t overlap

## testing
### Unconditional coverage
p <- alpha
T1 <- sum(violations)
T0 <- length(violations) - sum(violations)

x <- T1/(T1+T0) # violations as fraction of sample length, which is also the test statistic estimate

L_p <- (1-p)^T0 * p^T1
L_x <- (1-x)^T0 * x^T1

LR_uc <- -2 * log(L_p/L_x) # test statistic
critical <- qchisq(p = 0.95, df = 1) # the 5% critical value

LR_uc; critical
LR_uc > critical

### Independence testing
temp1 <- abs(diff(violations)) # to identify moments of change
temp2 <- violations[2:length(violations)] # to identify the ending points
T01 <- sum(temp2 * temp1) # those that finish with 1 and had a change
```

```
T11 <- sum(temp2 * (1-temp1)) # those that finish with 1 and had no change
T10 <- sum((1 - temp2) * temp1) # those finishing with 0 and having a change
T00 <- sum((1-temp2) * (1-temp1)) # finishing with 0 and no change

xhat <- x # from before
x00 <- T00/(T00 + T01)
x11 <- T11/(T11 + T10)

L_x_different <- x00^T00 * (1-x00)^T01 * (1-x11)^T10 * x11^T11
L_x_equal <- (1-xhat)^T00 * xhat^T01 * (1-xhat)^T10 * xhat^T11

LR_ind <- -2 * log(L_x_equal/L_x_different)
critical <- qchisq(p = 0.95, df = 1) # the 5% critical value
LR_ind; critical
LR_ind > critical
```

6.9 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:

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

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

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:

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

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.

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