University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Industrial Economics and Technology Management	Spring semester, 2020. Open
Author: Terje Berg	
Programme coordinator: Supervisor(s): Harald Haukås	
Title of master's thesis: Quality Minus Junk – A Study Across 44 Countries	
Credits: 30	
Keywords: Empirical asset pricing, QMJ, Factor model, Quality factor	Number of pages: 44 + supplemental material/other: 52 Stavanger, 15.June 2020

Abstract

This thesis seeks to further investigate the quality factor which is reported in the research literature on financial market anomalies and related to systematic investment strategies. The research is done first by a literature study, then by examining whether the findings from Asness, Frazzini and Pedersen's paper "quality minus junk" [1] can be replicated when increasing the dataset from 24 countries to 44. The quality factor is a proxy based on an asset pricing model where the future discounted payoffs is split into separate terms relating to profitability, growth and safety. I provide evidence confirming two research hypotheses, namely that 1: There is a positive and significant relationship between price and quality, and 2: an abnormal risk-adjusted return can be earned by investing in high-quality stocks and shorting low-quality stocks.

The first finding on the positive relationship between price and quality shows that the model specification based on modern asset pricing has explanatory power on stock prices, but most of the cross-sectional variation in prices is still unexplained. This finding is true in 40 out of the 44 countries examined for the sample data between 2005 and 2019.

The second finding shows that a factor-mimicking portfolio (QMJ) going long on the highest quality firms and shorting the low-quality stocks earns a significant risk-adjusted return with a Sharpe ratio after hedging for other factor exposures just above 1. The risk-adjusted alpha was positive for 43 out of the 44 countries in the sample.

This thesis contributes the empirical asset pricing field by confirming the results from Asness et al. using a broad sample of 44 countries obtained from a different data provider and with all the factors built from scratch. In their paper they conclude the abnormal risk-adjusted returns of quality stock are due to mispricing and they are unable to find a risk-based explanation. This study supports those conclusions and I find that quality deliver consistent returns during times of distress as well as in times of boom. It is difficult to find a risk-based explanation of the abnormal returns or a behavior-based story for why investors underweight high-quality stocks. Rather, the intelligent investor should add the QMJ factor to his or her toolbox of factors which can be used to create a portfolio aligned with the investor's goals and preferences.

> ***An investment operation is one which, upon thorough analysis, promises safety of principal and a satisfactory return. Operations not meeting these requirements are speculative.***
>
> ***–Benjamin Graham (1934)***

Contents

List of Figures

List of Tables

1 Introduction

The goal of this thesis is to examine factor models used in the asset pricing of stocks. More specifically, I wish to determine whether the findings relating to the quality factor proposed by Asness, Frazzini and Pedersen [1] can be replicated using an extensive data set from 44 countries, available through the novel Quantopian cloud platform. Quality is defined as "a characteristic that investors, all else held equal, should be willing to pay a higher price for". The overarching question they ask is: "Do the highest quality firms command the highest prices?" They conclude that the quality factor presents a puzzle to asset pricing theory, because they are unable to explain the high returns of "quality firms" based on a risk story or to demonstrate that prices in the cross-section vary "enough" with quality measures.

This puzzle intrigued me to dig into the field of empirical asset pricing and learn the skills to put factor models to the test. In order to prove or disprove the findings from Asness et al. I have formulated these two hypotheses that need to be tested:

1) There is a positive, but weak, correlation between asset price and a firm's quality.
2) A significant risk-adjusted return can be earned by investing in (going long) high-quality stocks and shorting the low-quality stocks.

2 Background

2.1 A Brief History of Asset Pricing

The problem of efficient allocation of limited resources is a fundamental issue to understand in economics. As student we are introduced to utility theory as model to understand how people use their resources efficiently by taking into account preferences, e.g. to risk or other factors. Utility theory is used to expand into the modern portfolio theory. Markowitz's minimum variance problem and the capital asset pricing model (CAPM) are presented as frameworks to understand pricing of individual assets under market equilibrium. This framework works well as an academic model, but given its many unrealistic assumptions the traditional derivations do not hold up against empirical data [2]. Still, surveys show that more than 70% of companies use the simple CAPM for determining cost of capital [3] (and for which it may be good enough).

Since the CAPM was developed in the 1960s alternatives and improvements have been proposed by several researchers. Ross developed the alternative arbitrage pricing theory (APT) [4] and perhaps most notably Fama and French who published their three-factor model [5] (and revised it with a five-factor asset pricing model in 2015 [6]). This work has been of special interest for security analysts and investors who try to predict the future price movement of securities or take advantage of certain market behaviour.

The typical textbook economic theory teaches us that stock prices in the market fully reflect all available information, sometimes referred to as the efficient market hypothesis [7]. This should imply that new information quickly affects asset prices and that the current available information cannot be used to predict future returns. What investors with decades of practical experience teach us on the other hand is a more nuanced perspective. Benjamin Graham, both an academic and practitioner writes together with David Dodd in their landmark textbook from 1934, "Security analysis" [8]:

> *"The market is not a **weighing machine**, on which the value of each issue is recorded by an exact and impersonal mechanism, in accordance with its specific qualities. Rather should we say that the market is a **voting machine**, whereon countless individuals register choices which are the product partly of reason and partly of emotion."*

Graham & Dodd were also some of the first authors to break down factors that affect the market prices and in doing so they made a sharp distinction between what they called speculative factors and analytical (or investment) factors. The purpose of including Figure 2-1 is to show that there is long timeline from those early "experience-based" observations of market factors to the more quantitative studies of recent years and attempts to explain them for instance by behavioural economics and factor models. Some of these explanations of how humans don't act like the rational Homo Economicus, like prospect theory, have have been popularized through the book "Thinking, fast and slow" [9] and has won researchers like Kahneman and Vernon the Nobel Memorial Price in Economics.

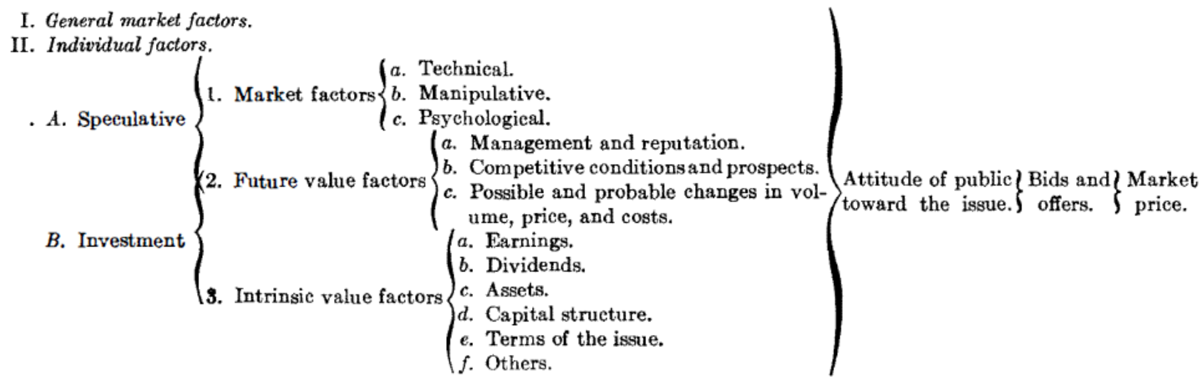


FIGURE 2-1: RELATIONSHIP OF INTRINSIC VALUE FACTORS TO MARKET PRICE. FROM THE 1934 TEXTBOOK SECURITY ANALYSIS [8].

2.2 Modern Asset Pricing

2.2.1 Consumption-Based Model

In this thesis I reference the textbooks by two leading scholars, John Cochrane of University of Chicago [10] and John Y. Campbell [11] of Harvard University, extensively on the topic of asset pricing and I will avoid repeating the references unnecessarily. They both view asset pricing theory as a framework to understand the price of some claim to an uncertain (future) payment. The concepts described here are generalizations of the theory developed from Markowitz and onwards. I present it here in a top-down approach to arrive at the models used in this thesis and that form the basis of modern asset pricing.

The fundamental concept in all asset pricing, from stocks to bonds to options, is this: price equals the expected discounted payoff. The investor must choose how much to consume now and how much to save for tomorrow. The marginal utility loss of consuming less today and buying some asset should equal the marginal utility gain of consuming more of the asset's payoff in the future. If the price and the future payoff does not satisfy this condition the investor will either buy more of or sell the asset. The investor's first order condition for optimizing that choice leads to the *consumption-based asset pricing model*:

$$p_t = E_t\left[\beta \frac{u'(c_{t+1})}{u'(c_t)} x_{t+1}\right] \quad \text{, where} \quad \begin{array}{l} p \text{ is the price at time } t \\ u \text{ is the utility as a function of consumption} \\ x \text{ is the asset's payoff} \\ \beta \text{ is the subjective discount factor} \end{array} \qquad (1)$$

In the case of a stock the expected payoff x at a given time is the expected price plus the expected dividend payment at this given time. We treat the payoff as a random variable which can take many possible outcomes. The utility function u may take any form we'd want, e.g. $u(c) = \ln(c)$, and describes the benefit, worth or value the investor gets from consumption at a given time (now or future). U is also treated as a random variable because we don't know how much money we have tomorrow and thus how much we want to consume it. The beta is here used as a subjective discount factor to correct for the fact that investors are risk averse and impatient, preferring money now over a risky and delayed cash flow. We typically separate certain terms into a *stochastic discount factor* m which measures the investor's "hunger"; the marginal utility of consumption in the future instead of today (or how much he values additional wealth tomorrow):

$$m_{t+1} = \beta \frac{u'(c_{t+1})}{u'(c_t)} \tag{2}$$

As a side note, most asset pricing models, like CAPM, ICAPM or APT, can be derived as special cases from the pricing equation (1) by imposing different constraints and form to the stochastic discount factor. For example, in the CAPM the discount factor $m = a + bR^W$ is assumed to be a linear function of the return on a "wealth portfolio" R^w (often proxied by a stock market portfolio like S&P500).

Consider that we have a certain risk-free rate; then the discount factor becomes $E(m) = \frac{1}{R^f}$, which is the more "standard" discount factor often used. We can write the price of a specific asset *i* as below in equation (3) using the definition of covariance. And using the fact that expected discounted excess return should be equal to zero, derive the expected excess return R^e (4):

$$p_t^i = E_t\left(m_{t+1}x_{t+1}^i\right) = \frac{E_t\left(x_{t+1}^i\right)}{R_t^f} + Cov_t\left(m_{t+1}, x_{t+1}^i\right) \tag{3}$$

$$E_t\left(R_{t+1}^{ei}\right) = -\,Cov_t\left(R_{t+1}^{ei}, m_{t+1}\right) \tag{4}$$

These derivations lead to the fundamental insight that the asset's price and its expected excess return (risk premium) depend whether the payoff/return covary positively or negatively with the investor's stochastic discount factor. As consumption c increase, the marginal utility m declines (diminishing return). If the asset payoffs x also declines together with m it implies a higher asset price. If payoffs covary negative with m investors will be willing to pay a lower price.

The insight can be explained from risk aversion; if the investor holds an asset that has a positive covariance with consumption, i.e. pays off well when you feel rich and pays less when you feel poor, it will make the consumption stream more volatile. A negative covariance between returns on the other hand will reduce consumption volatility and the investor can keep a steady consumption even in bad times. Essentially, we value assets that pay us when we are most "hungry" for money. The variance of the asset payoffs themselves are irrelevant and does not generate a risk premium; the investor cares only about volatility in his own consumption.

2.2.2 Expected Return-Beta Model

Traditional asset pricing models, like CAPM, ICAPM and APT, often measure the investor's "hunger" by evaluating the behavior of large asset portfolios. This evaluation is done by manipulating the pricing equation above to allow representing expected return by betas which are suitable for linear regression (note removal of excess return and that time subscript is removed):

$$E\left(R^i\right) = R^f + \left(\frac{Cov\left(R^{ei}, m\right)}{var(m)}\right)\left(-\frac{var(m)}{E(m)}\right) = \gamma + \beta_{i,m}\lambda_m \tag{5}$$

Worth noting here is that the λ is not asset specific and often interpreted as the price of risk, whilst the β is asset specific and the quantity of risk in each asset. Gamma is the inverse of E(m). For practical purposes we often wish to use factors that are not direct measurements of consumption growth. This goal can be achieved by introducing the concept of "factor-mimicking portfolios", in which we select a portfolio of assets whose payoff or return correlate closely with the discount factor m. The payoff space X is the set of all payoffs that investors can invest in and where investors can form any portfolio of traded assets or linear combinations of payoff vectors. Thus, a portfolio can be represented as vector of payoffs $\vec{x}$ (e.g. return on S&P500 stocks) and the payoff space consist of $X = \{\vec{c}^T\vec{x}\}$, where c is a vector of portfolio weights. To mimic the stochastic discount factor, we must choose a vector $\overrightarrow{x^*}$ which should be the orthogonal projection of the m onto X. We do this by choosing $\overrightarrow{x^*} = \vec{c}^T\vec{x}$, such that (dropping vector notation) $p = E(mx) = E(x^*x)$. This implies $x^* = p^T E(xx^T)^{-1}x$ is our discount factor to price the basis assets in x.

This discount factor is called the mimicking portfolio for m and is holds the same pricing implications as m, i.e. we can substitute all the m's in equation 5 with x*. Using the same arguments, we can create any factor f in which the factor-mimicking excess returns is the orthogonal projection of vector f onto the excess return space R^e. We can use the model form of equation 5 with the betas being regression coefficients of the returns on the factor-mimicking portfolio (not the factor itself). When using return as a factor the model becomes very elegant, since the factor risk premium is also the expected excess return.

By expand the model and saying β and λ is a linear combination of a sum of β_k and λ_k we end up with the expected return-beta model form we will use in this thesis:

$$E(R^{ei}) = \sum_{k=1}^{M} \lambda_k \beta_{k,i} \,, \qquad i = [1, N] \tag{6}$$

In the cross-sectional regression above the β_k's are the exposure of asset *i* to risk factor *k* and λ_k is the expected return for each unit of this exposure. We can also run a time-series regression for each asset i where beta are the coefficients we get when running a regression of return on factors. The factors i are (or should be) proxies for marginal utility growth:

$$R_t^i = \alpha_i + \sum_{k=1}^{M} (\beta_{k,i} f_{k,t}) + \varepsilon_{i,t} \,, \qquad t = [1, T] \tag{7}$$

2.2.3 Selecting factors – market anomalies

When generating a factor-mimicking portfolio, the analyst will look for factors that have predictive power on future returns. As described above we can trivially fit an unlimited number of factors to suit the return space data. So, to quote Cochrane the challenge is that:

> *"Most empirical asset pricing research posits an ad hoc pond of factors, fishes around a bit in that pond, and reports statistical measures that show ''success,'' in that the model is not statistically rejected in pricing a set of portfolios"*

So how do we combat this? According to Cochrane the best advice is to understand fundamental macroeconomic sources of risk, use economic theory to carefully specify the factors applied and use cross-sample and out-of-sample checking of your model's stability. The purpose is not necessarily to have a perfect data fit, but to describe how the investor's "hunger" varies along axes of interest in the cross-section and/or in time.

Factors that do not fit into the effective market hypothesis or CAPM framework have historically been called market anomalies, styles or risk factors. William Sharpe [12] was one of the many early researchers who suggested that the main differences in portfolio (mutual fund) returns could be attributed to differences in exposure to the four asset classes value/growth and large/small cap and he used factor models to show it. Most famous is of course the three Fama-French factors (market, size and value) and the later addition of momentum by Carhart [13], which have become the benchmark models used in empirical research.

Hou et al. [14] studied market anomalies extensively using cross-sectional data in a critical review, where they replicated 447 anomaly variables reported in financial literature. This research field is very prone to data mining (e.g. statistical overfitting or using of illiquid stocks), so after replicating the cross-sectional regression analyses for each factor the researchers found that only 161 were significant at a "typical" 5% significance level and only 46 when applying a more strict criteria as recommended for this type of analysis [15]. There is no commonly accepted classification of the market anomalies, but some of the most significant cross-sectional categories include:

Factor	**Description**
Momentum	The phenomenon that securities which have performed well relative to peers (winners) on average continue to outperform, and securities that have performed relatively poorly (losers) tend to continue to underperform [16].
Value	Value is the phenomenon that securities which appear "cheap" on average outperform securities which appear to be "expensive" [17].
Investment	A negative relation between capital investments for a firm and its future returns.
Profitability	An observation that more profitable firms have higher expected returns than less profitable profitable firms.

It is interesting to note that even with all the developments in data availability and computing power it seems like many of the experience-based guidelines Benjamin Graham gives in his book "The intelligent investor" [18] are still valid and actually resemble many of the significant predictive factors found. Graham advises the intelligent investor to select stocks that have a low risk of default, low debt-to-asset ratio, high asset-to-liability ratio, at least five years of earnings growth, low price-to-earnings ratio, low price-to-book ratio and who are paying dividends.

In order to avoid the pitfalls of selecting factor models that result from data mining or other misuse of statistics and data several authors have in recent years provided guidelines and heuristics. Both Hsu et al. [19] and Arnott et al. [20] provide guidelines for good which can be summarized as:

- Establish an ex-ante economic foundation
- Factors should be robust across definitions and geographies (cross-validation)
- Do not ignore trading costs and fees
- Ensure good data quality and document data transformations

As ever, it pays to listen to the advice of the old masters, here from Albert Einstein (1933):

> *"It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience."*

2.3 Quality as a Factor

2.3.1 Overview

In this thesis I will focus on investigating the factor, or investing style, called quality. It refers to a hypothesis (and finding) that investing in highly profitable, operationally efficient, safe and stable companies tend to outperform the market over time. There is no good common definition of quality across the literature, but Asness et al. [1] made a good attempt by defining quality as "the characteristics that investors, all else held equal, should be willing to pay a higher price for".

Then, what would those characteristics be? NBIM [21] review the quality factor and find that the factors considered in the literature can typically be grouped into three categories; profitability, safety and quality of earnings. Asness et al. also review existing literature and arrive on a very similar grouping. Because Asness et al. justify their quality model with a more theoretical first principles approach that fits into a modern asset pricing framework I will use their grouping in the following:

- Profitability
  The economic reasoning is that, all else equal, highly profitable companies should command a higher stock price. Profitability refers to the ability to generate earnings compared to expenses and can be measured by many accounting ratios. Hou et. Al [14] found 41 significant profitability-factors, typically related to ROE, ROA, gross profit to assets, asset turnover, operating profits to assets and cash-based operating profits to assets. Generally, these accounting factors indicate how well the company is deploying its capital to generate return and how well it manages its expenses.
- Safety
  The basis of safety characteristic is that, all else qual, investors should a higher price for companies with a lower required return (when looking at the companies' discounted cash flow). Risk of default by for instance excessive leverage would by economic theory increase the financing cost of a firm and thus the required return. Typical factors that describe the safety of a company are often related to a strong balance sheet, like low debt-to-assets, high current ratios, or volatility of profitability factors.
- Growth
  Growing profits are considered a characteristic that investors should pay a premium for, all else equal. This growth can typically indicate that the company has a sustaining competitive advantage over the competition. It can be measured as X-year growth in profitability (measured as above) or considering volatility over time.

Additionally, it is important to bring up research that failed to find a statistically significant "quality factor". Beck et al [22] found that the individual constituents of a typical quality factor does not have any explanatory power on return and risk of stocks. This result either disproves quality as a factor or it could be that it is the interaction between the quality variables that makes it a predictor of future returns.

2.3.2 Model for Quality

Asness et al. derive a mathematical model for quality using the firm value (price) described as the present value of all future dividends as a starting point (as described in section 2.2):

$$p_t = \sum_{s=1}^{\infty} E_t(m_{t+s}x_{t+s}) = \sum_{s=1}^{\infty} E_t\left(\frac{1+\varepsilon_{t+s}^{M}}{1+r_f}x_{t+s}\right) \tag{8}$$

The important point to note here is their choice of stochastic discount factor (also called pricing kernel), in which ε_{t+1}^{M} is the zero-mean innovation to the discount factor. When computing the conditional expectation of the discount factor they end up with $E_t(m_{t+1}) = \frac{1}{1+r_f}$. This implies that firm value is priced using a discount factor which is constant across time, which is an assumption we know to be an important simplification (see Cochrane [23] for a great review of the time-varying discount rate). We know that interest rates vary a lot over time and the expectation of future interest rates will greatly affect the discount rate and subsequently the prices. This choice of discount factor could lead to a model that does not account for changing interest rate regimes.

In the following I use value *V* instead of price *p* to keep the same terminology as in the paper. After some derivations and using the residual income valuation model Asness et al. compute the fundamental firm value as the sum of the book value and all future discounted residual incomes, which as a fraction of book value becomes the following:

$$\underbrace{\frac{V_t}{B_t}}_{\text{Value}} = 1 + \underbrace{\frac{v^e e_t + v - v^a \varepsilon_t^a}{B_t}}_{\text{profitability}} + \underbrace{v^g \frac{g_t - \bar{g}}{B_t}}_{\text{growth}} - \underbrace{v^\pi \frac{\pi_t - \bar{\pi}}{B_t}}_{\text{safety (negative risk)}} \qquad (9)$$

V_t - the firm's value	ε_t^a - the zero-mean random shock to residual income
B_t - the book value	π_t - risk premium due to covariation with discount factor
e_t - sustainable residual income	g - growth term
	v^i - valuation coefficients

Using this model founded on "first principles", we see that the firm's value can be explained by factors relating to its ability to generate profit, growth and avoid negative risk, which was hypothesized in the previous chapter. Other authors, e.g. Frama and French (2014) [24], starting from the same dividend growth model end up with slightly different factor models although the reasoning is similar.

The next step in developing the factor model is to find representative proxies for the variables in the equation above. We are not looking to price assets correct in absolute values, but to compare prices relative to each other. Asness et al. use the result from robust studies to select available fundamental data points for companies and to construct proxies for profitability, growth and safety. For instance, the profitability measures used have been mostly selected from a highly cited study by Novy-Marx [25]. Each variable is ranked in the cross-section universe, normalized (z-scored) and given equal weight when averaged into a factor, which is normalized on its own.

$$Quality = z(Profitability + Growth + Safety) \qquad (10)$$

$$Profitability = z(\mathrm{GPOA}_z + \mathrm{ROE}_z + \mathrm{ROA}_z + \mathrm{CFOA}_z + \mathrm{GMAR}_z + \mathrm{ACC}_z) \qquad (11)$$

$$Growth = z(\Delta\mathrm{GPOA}_z + \Delta\mathrm{ROE}_z + \Delta\mathrm{ROA}_z + \Delta\mathrm{CFOA}_z + \Delta\mathrm{GMAR}_z) \qquad (12)$$

$$Safety = z(\mathrm{BAB}_z + \mathrm{LEV}_z + O_z + Z_z + \mathrm{EVOL}_z) \qquad (13)$$

The following table provides a summary of the variables. The growth variables with a delta-prefix indicates the 5-year change of the variable. To see how I have constructed these measures in the analysis I refer to the Python notebook's section 2.2 in Appendix 3.

TABLE 2-1: VARIABLE DEFINITIONS

Variable	Description	Variable	Description
GPOA	Gross profit over assets	**BAB**	Market beta
ROE	Return on equity	**LEV**	Leverage (debt over assets)
ROA	Return on assets	**O**	Ohlson's O-score
CFOA	Cash flow over assets	**Z**	Altman's Z-score
GMAR	Gross margin	**EVOL**	Earnings volatility
ACC	Fraction of cash earnings		

3 Data

3.1 Quantopian Research Platform

For all the data analysis in this thesis I have used Quantopian.com, which provide a cloud-based data science platform for performing quantitative financial analysis using the Python programming language. The platform provides an IDE (interactive development environment) to perform research on equity data and an engine to perform backtesting of trading algorithms.

3.2 Data Sources

The data available through Quantopian contain quality checked equity data from 44 countries from 2004-2019. For example, to avoid survivorship bias they contain data as stocks are listed and delisted and stored point-in-time so that the backtesting simulation engine avoids any lookahead bias. As an example, in asset pricing research it is fairly customary to construct factors based on accounting data with a six months' time lag to avoid lookahead bias. With point-in-time data this is not necessary because data in the backtesting engine will only become available in the simulation at the historical filing date of each company's financial reporting.

The data sources used in this thesis are from FactSet and contain fundamentals data, equity pricing and metadata and RBICS (business industry classification). To expand the analysis the available data also range from analyst estimates to insider trade transactions and news sentiment.

3.3 Data Processing

Although the data sources are of high quality, they still must be processed in order to obtain sample data that can be analyzed and that are relevant. The first data screening is to remove any non-tradable assets and to only include primary shares. The primary share is defined as the first share/ticker that a company has at IPO and is still actively trading. If this share is no longer trading, the share with the highest volume is denoted as the primary share. After this initial filtering, we typically see that the datasets contain up to 10-30% of missing data for some of the accounting/fundamental data that we use to build the quality factor. This is typically the smaller stocks that we anyway almost disregard when value-weighting the components later in the analysis work. Some data is critical, so any stock with missing market to book ratio has been completely removed.

The next step of processing is to build all the quality factors given in section 2.3.2. First, we perform a winsorization to limit the effect of extreme values and possible spurious outliers (especially relevant for accounting data and ratios). 95% winsorization is done by setting all values outside the lower and upper bound (2.5 and 97.5 percentile) equal to the boundary value.

Next, we know that accounting data is different when comparing across industries. For example, the profit margin is typically much higher for companies in the financial sector than in the utilities sector (although this tells us little about the returns from investing in either industry). To normalize accounting data across industries we demean each factor my sector. This means that the most profitable investment bank is ranked equal as the most profitable utility company when constructing the profitability factor. There are several choices on methodology for industry demeaning and each one has its positive and negative sides. Some researchers eliminate all financial services firms from the analysis, they cluster regressions by industry or they enforce sector neutrality by weighting methods. For this work I have chosen to demean by grouping all stocks within each sector together and then normalizing each of the factor components within sectors. The dataset groups each asset in one of 13 industries, but unfortunately industry classification is missing for a lot of the smaller markets (again, this is most widespread in small stocks). Comparing results before and after industry demeaning showed a marked improvement when inferring statistical significance of the regressions.

When normalizing (z-scoring) the fundamental data we assign the stocks with missing data to have a score of zero, such that when we aggregate the normalized factors the missing data for each stock is effectively ignored. The method may not be perfect, but for empirical research it serves the purpose of not having to delete every stock which is missing some data which would reduce our sample data. Finally, the stocks are ranked and then normalized as suggest by Asness et al. When the procedure in this section has been performed, we are left with zero missing data for the analysis work.

3.4 Summary Statistics

The 44 countries analyzed are listed below, sorted by size along with some summary statistics. All the countries have been analyzed from 2005 until June 2019 and consist of a total of 49 003 companies.

The market capitalization time series is converted to US dollars using London Market spot exchange rates at close of each day and the mean value here is across the entire time series. The number of stocks per month is the number included in each monthly calculation and the total stocks is the total number of unique assets over the entire time period. The global market weight is a naive calculation of each countries relative size as the weighted average of the mean market cap multiplied by number of stocks per month. For comparison I have also made a column for each countries weight based on the number of stocks in each universe divided by total number of stocks.

TABLE 3-1: SUMMARY STATISTICS OF DATA SAMPLE

Country	Mean Market Cap	Stocks per month	Total Stocks	Global Market Weighted Size	Global Equal Weight Size
United States	4.15E+09	4502	6866	34.9 %	14.0 %
Great Britain	1.78E+09	1723	3820	8.3 %	7.8 %
Japan	1.17E+09	3723	5092	7.3 %	10.4 %
Hong Kong	2.32E+09	1454	2493	7.1 %	5.1 %
China	1.30E+09	2143	3575	5.7 %	7.3 %
Canada	6.59E+08	2554	4887	3.9 %	10.0 %
Germany	1.80E+09	846	1503	3.3 %	3.1 %
Australia	6.99E+08	1680	2866	2.5 %	5.8 %
South Korea	6.54E+08	1571	2620	2.1 %	5.3 %
Switzerland	4.40E+09	242	374	2.0 %	0.8 %
Russia	2.78E+09	236	543	1.9 %	1.1 %
Spain	4.67E+09	158	319	1.8 %	0.7 %
Sweden	1.17E+09	485	1132	1.6 %	2.3 %
Taiwan	4.88E+08	1647	2379	1.4 %	4.9 %
Brazil	2.92E+09	174	283	1.0 %	0.6 %
Netherlands	4.10E+09	107	201	1.0 %	0.4 %
South Africa	1.28E+09	312	602	0.9 %	1.2 %
Singapore	6.94E+08	678	1018	0.9 %	2.1 %
Mexico	3.17E+09	116	196	0.8 %	0.4 %

Country	Mean Market Cap	Stocks per month	Total Stocks	Global Market Weighted Size	Global Equal Weight Size
Norway	1.07E+09	241	500	0.7 %	1.0 %
Indonesia	7.42E+08	428	660	0.6 %	1.3 %
Malaysia	3.77E+08	943	1294	0.6 %	2.6 %
Denmark	1.61E+09	169	280	0.6 %	0.6 %
Thailand	5.23E+08	564	834	0.5 %	1.7 %
Finland	1.67E+09	128	203	0.4 %	0.4 %
Turkey	6.68E+08	321	472	0.4 %	1.0 %
Poland	3.46E+08	442	760	0.3 %	1.6 %
Colombia	2.64E+09	47	82	0.3 %	0.2 %
Austria	1.69E+09	76	127	0.3 %	0.3 %
Philippines	6.98E+08	230	303	0.3 %	0.6 %
Argentina	1.41E+09	78	116	0.2 %	0.2 %
Ireland	2.48E+09	33	64	0.2 %	0.1 %
Greece	3.78E+08	243	363	0.2 %	0.7 %
Portugal	1.44E+09	52	89	0.2 %	0.2 %
Peru	6.16E+08	120	191	0.1 %	0.4 %
Czech Republic	1.97E+09	19	55	0.1 %	0.1 %
New Zealand	4.26E+08	126	223	0.1 %	0.5 %
Pakistan	2.14E+08	246	342	0.1 %	0.7 %
Hungary	7.23E+08	35	64	0.1 %	0.1 %

4 Analysis

4.1 Overview

In order to answer my three main research questions and to replicate the "Quality minus junk" study I will perform the following analyses to test each hypothesis:

1) There is a positive correlation between price and quality
    a. Persistence of quality
    b. Regression of price on quality
2) There is a positive risk-adjusted return from investing in (going long) high-quality stocks and shorting the low-quality stocks
    a. Regression of excess return on quality sorted portfolios
    b. Regressions of the QMJ factor-mimicking portfolio returns on risk factors.

4.2 Correlations Between Factor Components

The following plots show the relationship between the quality factor and its sub-factors profitability, growth and safety and the relationship between each sub-factor and their individual components. The pairwise correlations for the global sample mean weighted by market size is found in Table 4-1. The main finding you can see graphically from Figure 4-1 is the strong pairwise correlation coefficients between the quality components. The correlation between profitability and growth of 0.67 across the whole global value weighted sample indicate that profitability is persistent and this is in line with findings from Novy-Marx [25]. It is less intuitive that profitability and growth is correlated with the safety factor and from these initial plots and the average numbers across the sample it difficult to see any patterns (on average profitability and safety have a correlation coefficient of 0.11).

From Figure 4-2 to Figure 4-4 we see that there looks to be a positive relationship between all the components (fundamental data) that make up a factor, which tells us that Asness et al. found a robust set of proxies for their factors. By robust I mean that if some data constituting e.g. the safety factor is missing or has measurement error it will have less effect on the aggregated main factors. It should also be noted that the factor sub-components are not shown in their normalized form and that the underlying sample distributions of fundamental data are very much non-normal.

TABLE 4-1: PEARSON PRODUCT-MOMENT CORRELATIONS BETWEEN QUALITY AND SUB-FACTORS

	Quality	Profitability	Growth	Safety
Quality	1.00			0
Profitability	0.82	1.00		
Growth	0.73	0.62	1.00	
Safety	0.48	0.12	0.07	1.00

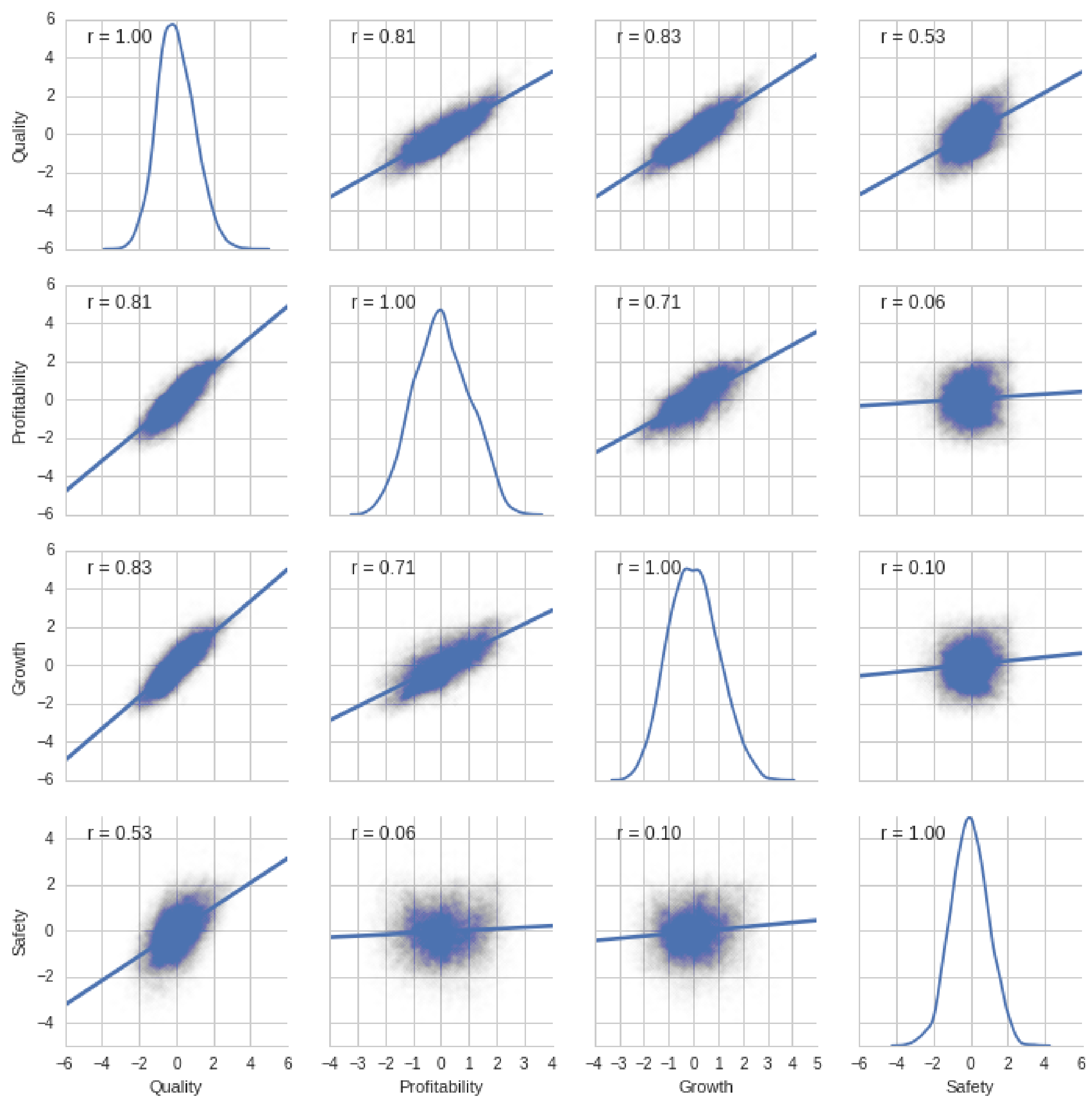


FIGURE 4-1: QUALITY PLOTTED AGAINST ITS SUB-FACTORS PROFITABILITY, GROWTH AND SAFETY. THE DIAGONAL SHOWS THE SAMPLE DISTRIBUTION OF EACH FACTOR ALONG WITH A FITTED REGRESSION LINE TO INDICATE DIRECTION OF RELATIONSHIP. PEARSON PRODUCT-MOMENT CORRELATIONS ARE DENOTED BY R-VALUE. DATA IS US SAMPLE FROM '05-'19.

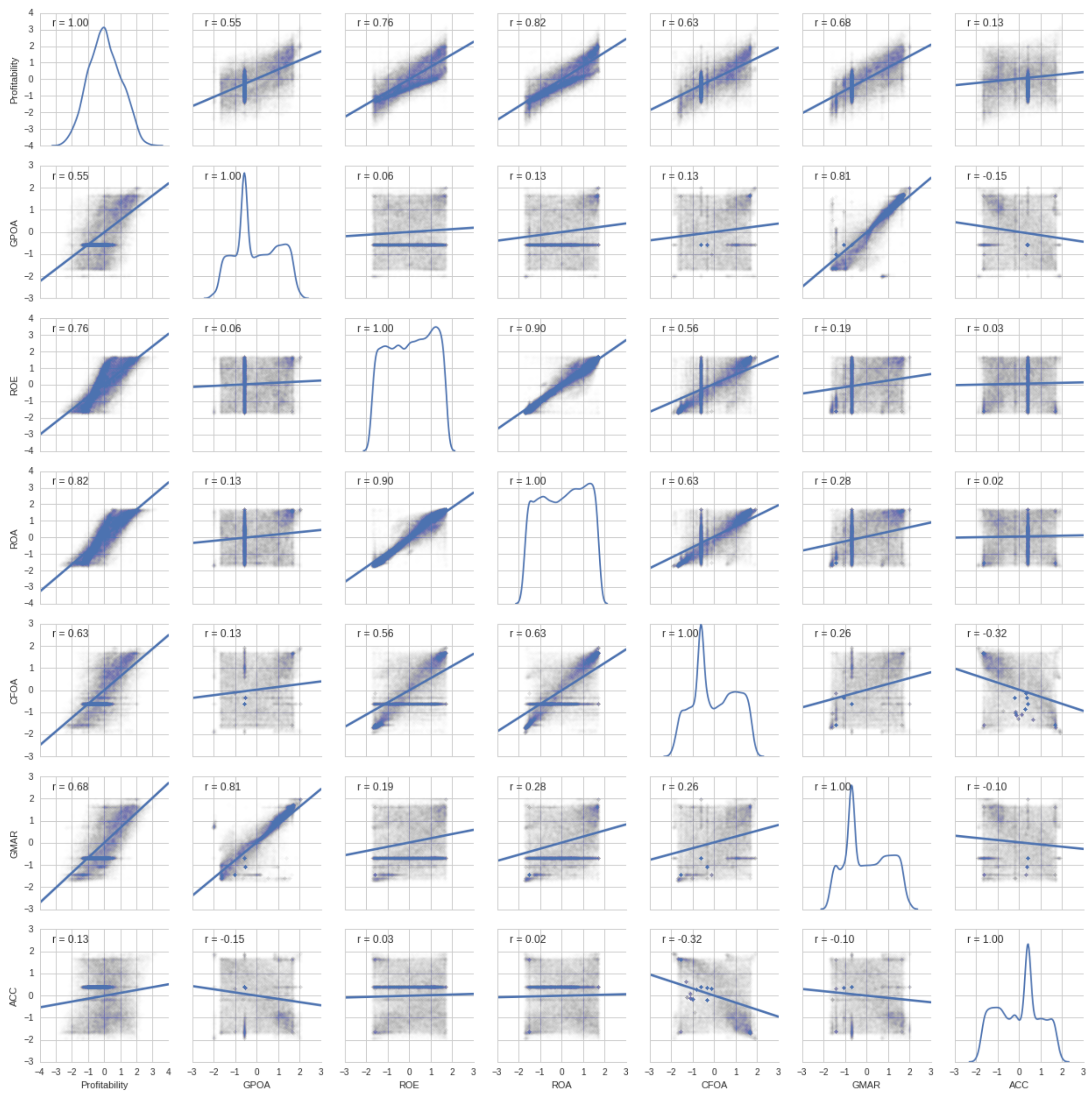


FIGURE 4-2: PAIR-PLOTS OF THE PROFITABILITY FACTOR AND ITS SUB-COMPONENTS. THE DIAGONAL PLOTS ARE THE SAMPLE DISTRIBUTIONS AND THE OFF-DIAGONAL SCATTER PLOTS ALSO SHOW A FITTED REGRESSION-LINE TO INDICATE DIRECTION OF RELATIONSHIP. PEARSON PRODUCT-MOMENT CORRELATIONS ARE DENOTED BY R-VALUE. THE SAMPLE IS US STOCKS '05-'19

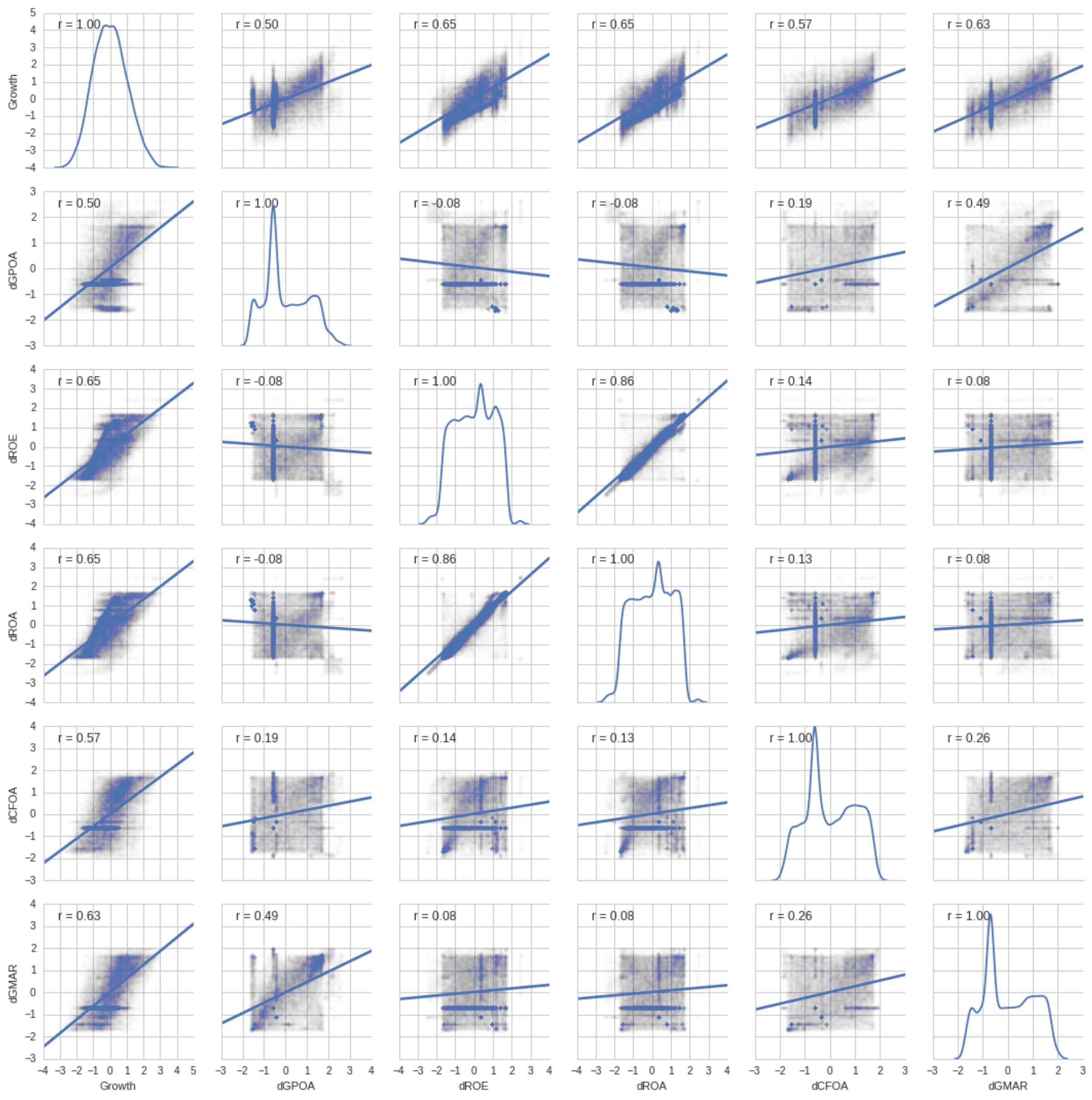


Figure 4-3: Pair-plots of the growth factor and its sub-components. The diagonal plots are the sample distributions and the off-diagonal scatter plots also show a fitted regression-line to indicate direction of relationship. Pearson product-moment correlations are denoted by R-value. The sample is US stocks '05-'19

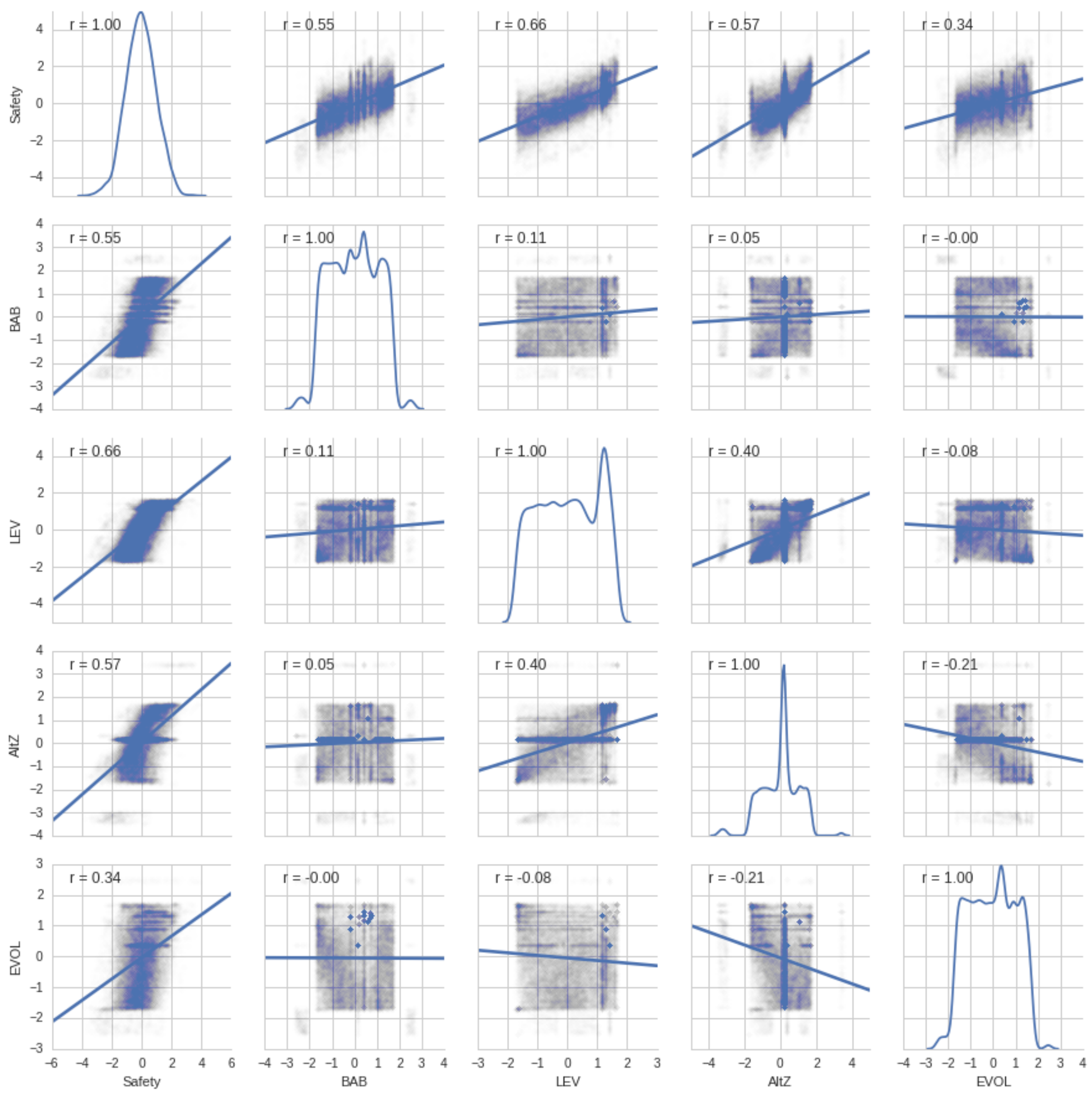


FIGURE 4-4: PAIR-PLOTS OF THE SAFETY FACTOR AND ITS SUB-COMPONENTS. THE DIAGONAL PLOTS ARE THE SAMPLE DISTRIBUTIONS AND THE OFF-DIAGONAL SCATTER PLOTS ALSO SHOW A FITTED REGRESSION-LINE TO INDICATE DIRECTION OF RELATIONSHIP. THE SAMPLE IS US STOCKS '05-'19

4.3 Regression Analysis

In the following sections we run regression analyses on price and on returns. Our main tool is ordinary least-squares regression, but we use the procedures of Fama-Macbeth to obtain corrected standard errors and correct for autocorrelation using the method of Newey-West. Although these are the traditional tools used in asset pricing research (and by Asness et al.) the more modern approach, summarized nicely by Peterson [26], would be a panel regression with clustered standard error estimates for firm effects and time. Thompson [27] builds on this approach and provides simple formulas for firm and time effect corrected standard errors. The bias of the Fama-Macbeth standard error is most severe in cases where a persistent dependent variable is regressed on persistent independent variables, for example when we regress the market-to-book ratio on firm characteristics.

Despite this, because I struggled to implement clustered errors in the Python Statsmodels regression library (and to stay true to Asness et al.'s methodology) I chose to use the Fama-Macbeth procedure described here.

The procedure is slightly different depending on what our factors are. For observable characteristics the first step below is often omitted or taken as a separate analysis. In our case we do regressions on quality, which is calculated separately for each asset and each time step. So, there is no need to estimate asset-specific betas using equation 14 below.

The Fama-Macbeth two-step regression procedure is often used in analysis of factors that explain asset returns. It is a practical "two-pass" way of testing how the factors describe portfolio or asset returns by finding the return premium from exposure to the factors. First, each portfolio's or asset's return is regressed against the factor time series f_t (e.g. MKT or QMJ) to determine how exposed it is to each one using equation 14.

$$R_t^{ei} = a_i + \beta_i' f_t + \epsilon_t^i \,, \quad \text{t=1,2...,T for each i} \tag{14}$$

$$E_T\left(R^{ei}\right) = \beta_i' \lambda + \alpha_i \,, \quad \text{i=1,2,...,N} \tag{15}$$

$$R_t^{ei} = \beta_i' \lambda_t + \alpha_{it} \,, \quad \text{i=1,2,...,N for each t} \tag{16}$$

In a "traditional" two-step regression, we would then use equation (15) to estimate a single cross-sectional regression with the sample averages, but Fama-MacBeth suggested that instead we run a cross-sectional regression *at each time period*. The main advantage of Fama-MacBeth is to then average these coefficients, once for each factor, to give the premium expected for a unit exposure to each factor (19) and alpha (17) over time. This method splits the sample into T smaller samples and we can deduce the variation across samples (time), assuming no autocorrelation. Our estimators simply become the average across time (sample mean) and the sampling errors are generated from the standard deviation of the sample means:

$$\hat{\alpha}_i = \frac{1}{T}\sum_{t=1}^{T} \hat{\alpha}_{it} \quad , \quad \sigma^2(\hat{\alpha}_i) = \frac{1}{T^2}\sum_{t=1}^{T}(\hat{\alpha}_{it} - \hat{\alpha}_i)^2 \tag{17,18}$$

$$\hat{\lambda} = \frac{1}{T}\sum_{t=1}^{T} \hat{\lambda}_t \quad , \quad \sigma^2(\hat{\lambda}) = \frac{1}{T^2}\sum_{t=1}^{T}\left(\hat{\lambda}_t - \hat{\lambda}\right)^2 \tag{19,20}$$

If the prices are independent and identically distributed (iid) normally over time, then the t-statistic can be used to test the null hypothesis that the regression coefficients are zero. See Cochrane [10] chapter 12.3 for a detailed treatment of the procedure. The practical implementation of this second step is to run a regression of the periodic estimates against a constant and use the software option for Newey-West heteroskedasticity and autocorrelation adjusted standard errors. The estimated coefficient of such a regression is simply the sample mean (as above) and corrected standard errors. The Newey-West procedure requires the user to set a number of time lags to use in the correction and according to literature [28, p. 7] a guideline for choice is to follow this equation when using the Bartlett kernel (which is what Statsmodels uses as default):

$$t_{lags} = 4\left(\frac{T}{100}\right)^{\frac{2}{9}} \xrightarrow{in\ our\ sample} t_{lags} = 5 \tag{21}$$

4.4 Portfolio Forming

Before digging into the analysis of quality-sorted portfolios, I will briefly explain the method used. Portfolio analysis is traditionally a very commonly used method in empirical asset pricing to examine the cross-sectional relationship between some variable(s). It is essentially a non-parametric cross-sectional regression using non-overlapping histogram weights, as illustrated in Figure 4-5. The big motivation for creating portfolios is to remove the "noise" (idiosyncratic volatility) of each individual asset by bundling them into portfolios of assets that have relatively similar exposure to a factor. The univariate portfolio analysis procedure has, as detailed nicely by Bali et al. [28], four steps:

1) Calculate the factor breakpoints that will be used to divide the sample into portfolios.
2) Use these breakpoints to form the portfolios.
3) Calculate the average value of the outcome variable Y within each portfolio for each period t and present the time series average with corrected standard errors.
4) Examine the variation in these average values of Y across the different portfolios.
    a. Examine if the time-series mean of the portfolios, especially the difference portfolio (H-L), is statistically different from a null hypothesis mean value (often zero). A non-zero mean is evidence that a cross-sectional relation exists between the sort variable and outcome variable.

There are also methods for creating bivariate (double-sorted) portfolios, which is what Asness et al. does when creating the QMJ factor by sorting on size (market capitalization) and then sorting on the quality factor. This bivariate sort is similar to a regression on quality controlled for size. According to Cochrane [10] you can get the same results whether you perform portfolio sorts or multivariate regression in time and cross-section (panel data regression). This is what Figure 4-5 shows by comparing a regression slope and the portfolio mean values in a factor model on returns with log(book/market cap) as the only factor variable. The challenge with portfolio sorting is when your factor models starts to have more than 2-3 factors which should all be sorted on. Ang, Liu and Schwarz [29] actually find that the practice of portfolio forming leads to larger standard errors of cross-sectional coefficient estimates because it reduces the information (beta dispersion). Nevertheless, because Asness et al. use double sorted portfolios in their paper I have done the same.

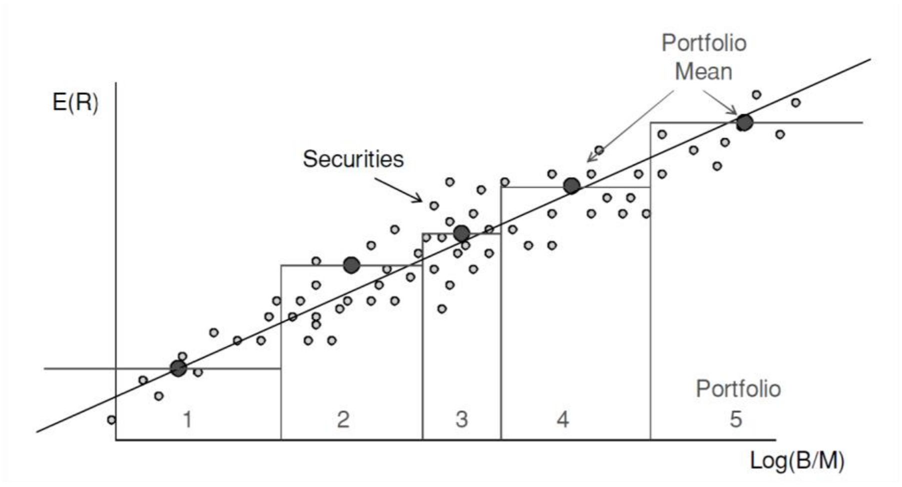


FIGURE 4-5: PORTFOLIO MEAN RETURN VERSUS CROSS-SECTIONAL REGRESSIONS [23].

4.5 The Price of Quality Stocks

4.5.1 Persistence of Quality

In order to determine the price of quality stocks I first perform a univariate portfolio sort on quality to split each stock universe into ten equal-sized quality portfolios. This is the first step of testing whether high quality firms command higher prices. If quality is persistent it means the market can predict future quality and take this into account when determining the prices today.

The results from Table A-1 in Appendix 1 show us that the quality score is consistent over time for the entire sample. This is also illustrated visually in the figures below. Figure 4-6 shows the difference portfolio (high minus low) for each country in the sample at the time of portfolio formation and three and ten years after formation. Each month we form the quality sorted portfolios and record the quality scores of the same portfolio three and ten years later. We examine if the time series mean of the difference portfolio after three and ten years is statistically distinguishable from zero as an indication/evidence that a cross-sectional relationship is persistent. This is done by regressing the time series means on a constant and implementing the Newey-West adjusted standard errors to correct for heteroskedasticity and autocorrelation.

We also want to know if there is a monotonic pattern in the quality sorted portfolios. Figure 4-7 shows the average portfolio quality means for the entire sample. At portfolio formation the monotonicity is by construction, but we also see that the monotonic pattern is persistent even after ten years. There is a regression towards the mean, but we clearly see that on average a significant number of the high-quality companies at portfolio formation are still winners even ten years later and vice versa.

The combined results allow us to conclude that quality is persistent is every country of the sample and that it is possible to select companies which will exhibit high quality in the future by looking at their recent past. In theory that should mean that the market has the necessary information to correctly reflect future quality in today’s prices.

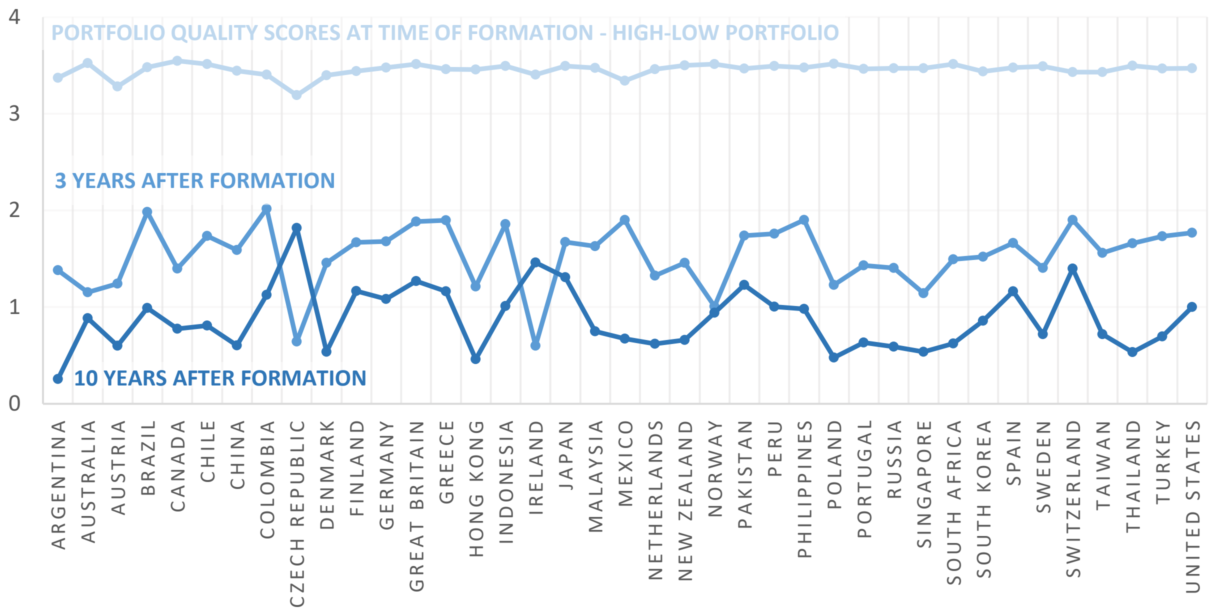


Figure 4-6: This chart shows the high-minus low of quality sorted portfolios at time of portfolio formation and the corresponding mean portfolio quality scores 3 and 10 years after portfolio formation. All estimates are statistically significant; the average t-statistics for the 3-year lagged means is 32.61 and for the 10-year lagged means it is 15.72.

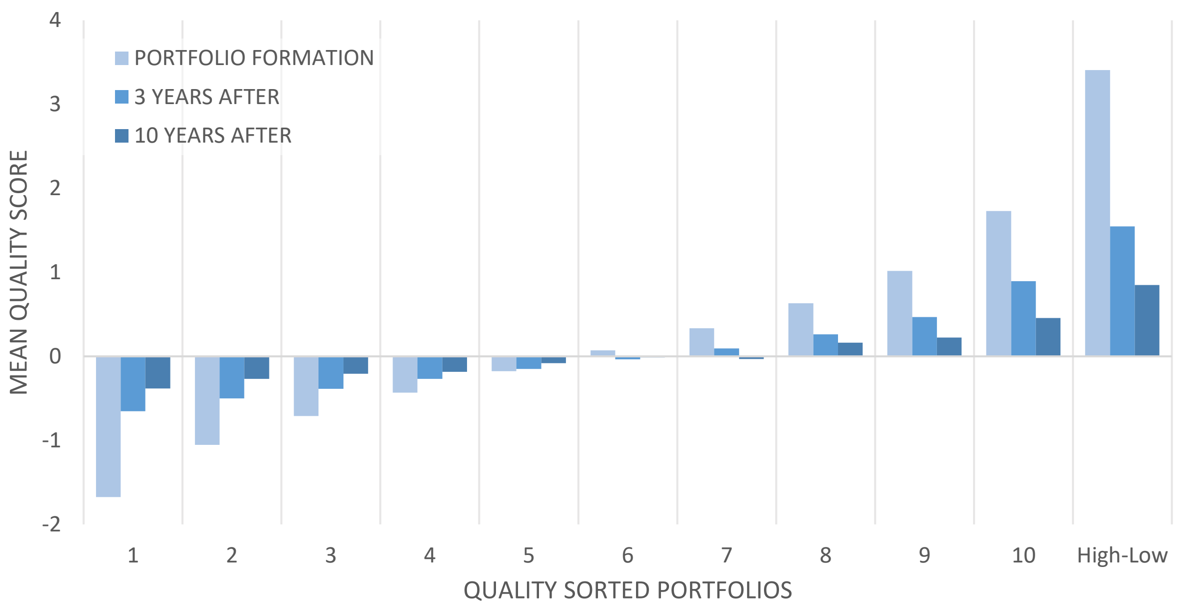


Figure 4-7: Change in mean portfolio quality scores from time of formation and after 3 years and 10 years. Portfolio scores are the equal weight mean of the entire global sample.

4.5.2 Price of Quality in the Cross-Section

We now run a cross-sectional regression of price on quality. Although it varies from the expected return-beta model its equivalence can be seen by examining equation 3 to 5.

$$p_t^i = \frac{E_t\left(x_{t+1}^i\right)}{R_t^f} + Cov_t\left(m_{t+1}, x_{t+1}^i\right) \tag{22}$$

$$\log\left(\frac{ME}{BE} + 1\right)_{it} = \alpha_{it} + \lambda_t \beta_i = \alpha_{it} + \lambda_t Quality_i \tag{23}$$

We perform the natural logarithmic transformation of highly skewed ME/BE data into “close to” normally distributed data, but to avoid numerical issues I applied $\log(1+x)$ to the market equity to book equity (the regressand). The interpretation is, according to Woolridge’s textbook on econometrics [30, p. 193], approximately same as in a standard log-linear regression model where a unit change in X leads to 100×β% change in Y. Because our factors are z-scored a unit change in our factors is equal to a standard deviation change, e.g. moving up from the mean quality score to the 84th percentile. I have run regression models without the log-transformed dependent variable and the results are similar (coefficients still strongly significant). The cross-sectional regressions are performed using the Fama-Macbeth procedure as described in section 4.3. We run five different model specifications: four regressions of price on quality, profitability, growth and safety individually and a fifth multivariate regression of price on profitability, growth and safety. To get a visual understanding of the sample I have included Figure 4-8 which shows the underlying sample data scatterplot of price and quality along with a fitted regression line and the sample distributions for the US market.

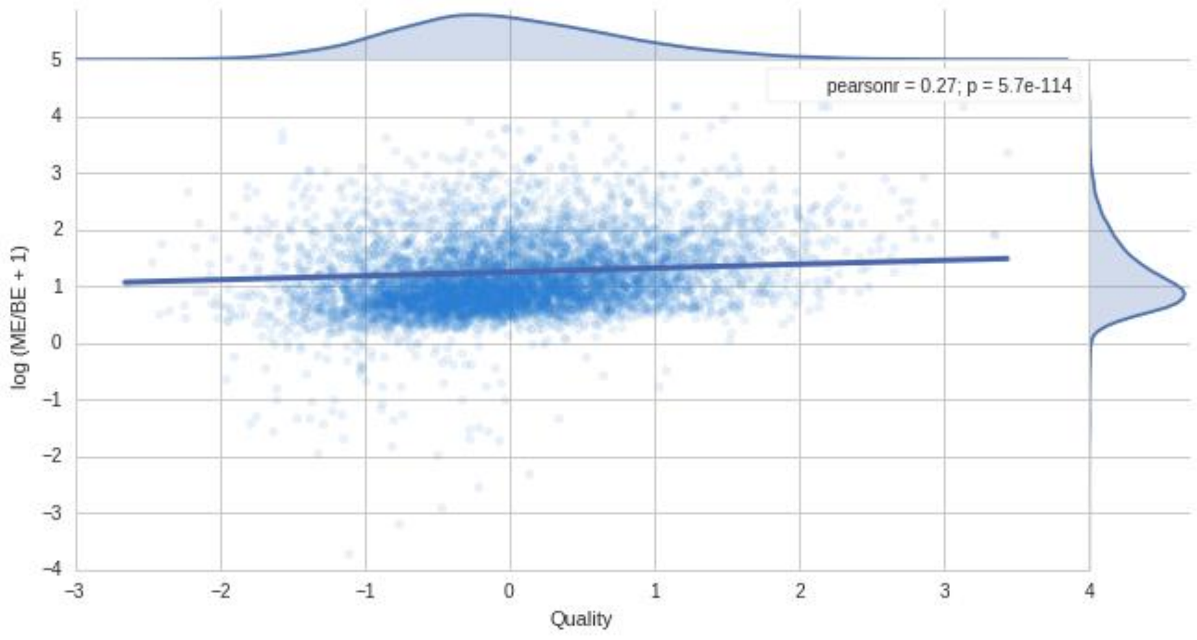


FIGURE 4-8: TYPICAL SCATTERPLOT OF TIME AVERAGED SAMPLE DATA AND FITTED REGRESSION LINE (US ’05-’19). DISTRIBUTION OF SAMPLES SHOWN ON TOP AND RIGHT ALONG WITH PEARSON’S CORRELATION NUMBER (0.27) BETWEEN THE TWO VARIABLES

From Table 4-2 we see that our hypothesis stating that there is a positive correlation between price and quality cannot be rejected for the big majority of countries. Our regressions show significant coefficient, all our sub-components of quality have the same sign and the results are in general in alignment with Asness et al. Although I have not included all the controlling factors that they have I find the explanatory power of the regressions to be in the same range as their study, with about 8% of variation explained in the US sample and 8% in the value weighted global sample. The magnitudes of the coefficients are in the same order of magnitude and the aggregation of sub-components into the quality score retains both magnitude and statistical significance. Given that our starting point was a theoretically plausible formulation of a firm's fundamental value it is perhaps surprising that a regression model made of the components profitability, growth and safety explain very little of the cross-sectional variation in prices. But nonetheless, we can conclude that higher quality companies do command higher prices than "junk" companies.

Looking at the magnitudes, Figure 4-9 zooms in on the quality slop estimate for the sample. First, we see that the magnitude varies, but the majority of countries exhibit a statistically significant positive relation with a value weighted sample mean of 0.11. This is interpreted such that if the quality score of a company moves one standard deviation higher, ceteris paribus, the market to book value (ME/BE) ratio only increases 11% from the total sample. Or put differently; companies having a quality score in the 99th percentile (2σ) can enjoy only a 22% higher stock price than the average company (all else equal, particularly book equity). The slope coefficient estimated for quality is much lower than the 0.22 and 0.24 coefficient that Asness et al. finds, but their sample is much longer and could indicate that relationship between price and quality is lower in more recent years. For the Norwegian reader it is worth noting that the slope coefficient for the Oslo Stock Exchange is on the lower side at 0.05 (with a t-statistic of 12.57), i.e. the price of quality in Norway is low compared to the global sample.

Our analysis on the relationship between price and quality leaves us to conclude that our model specification for quality does explain prices, i.e. our first hypothesis that high-quality firms exhibit higher prices cannot be rejected for 40 of our 44 countries. The findings here confirm and support the results from Asness et al.'s study on the quality factor.

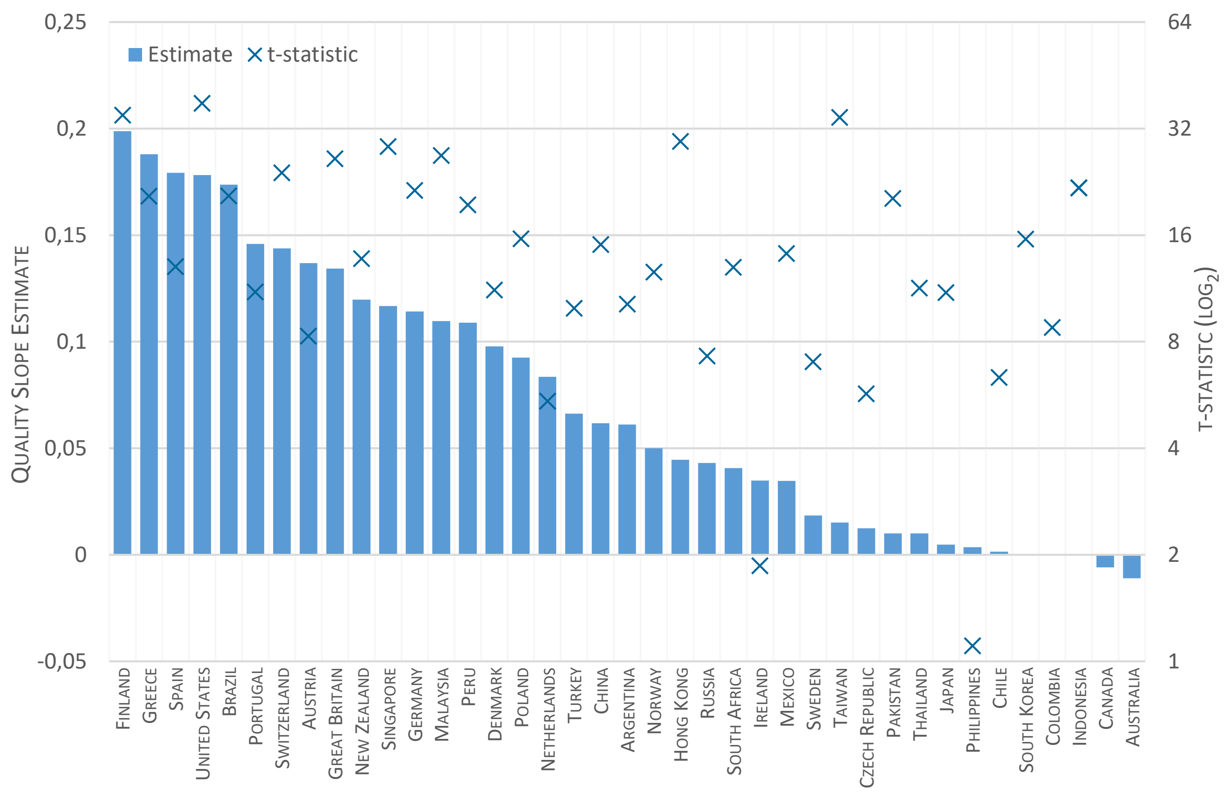


Figure 4-9: This chart's left-hand axis shows the slope coefficient estimates for a regression of price on quality for each country. The crosses indicate the t-statistic for each regression and the t-values are given on the right-hand axis scaled by log$_2$, which means that all the estimates with crosses below the main horizontal axis have a t-statistic less than 2.0 (Ireland and Phillipines).

Table 4-2: The price of quality - Cross sectional regressions

This table presents the results from monthly Fama-MacBeth regressions. The dependent variable is the natural logarithms of a firm's market-to-book ratio plus one at time t. The explanatory variables are the quality scores at time t. AdjR^2 is the time series average of the adjusted R-squared of the cross-sectional regressions of price on quality (model 1). Standard errors are adjusted for heteroskedasticity and autocorrelation using a lag length of 5 periods. T-statistics are shown below the coefficient estimates in paranthesis. The top row shows the global sample mean by market size weights. Countries as sorted by market size from big to small.

$\log\left(\frac{ME}{BE}+1\right)$	(1)	(2)	(3)	(4)		(5)				
	Quality	Profitability	Growth	Safety	Profitability	Growth	Safety	Adj R^2	N_{time}	$\bar{N}_{assets}$
Sample mean	**0.11**	**0.08**	**0.06**	**0.11**	**0.06**	**0.02**	**0.10**	**0.06**	**171**	**2363**
United States	0.19	0.15	0.15	0.12	0.10	0.07	0.10	0.079	119	4133
	(54.74)	(28.84)	(22.13)	(23.84)	(20.64)	(7.28)	(14.71)			
Great Britain	0.13	0.06	0.00	0.20	0.10	-0.04	0.20	0.028	172	1652
	(26.29)	(7.25)	(-0.73)	(46.75)	(6.73)	(-3.14)	(40.65)			
Japan	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.062	163	3723
	(11.00)	(13.37)	(0.00)	(0.00)	(17.56)	(-8.50)	(0.00)			
Hong Kong	0.04	0.01	0.02	0.07	-0.01	0.01	0.07	0.022	168	1440
	(29.47)	(3.58)	(19.38)	(24.14)	(-2.12)	(2.22)	(23.50)			
China	0.06	0.03	0.04	0.07	-0.03	0.05	0.08	0.034	165	2126

$\log\left(\frac{ME}{BE}+1\right)$	**(1)**	**(2)**	**(3)**	**(4)**		**(5)**				
	Quality	Profitability	Growth	Safety	Profitability	Growth	Safety	Adj R^2	N_{time}	$\overline{N}_{assets}$
	(15.05)	(7.82)	(11.44)	(10.25)	(-5.58)	(10.32)	(10.23)			
Canada	-0.01	-0.11	-0.08	0.18	-0.06	0.01	0.18	0.000	172	2334
	(-0.37)	(-7.23)	(-5.81)	(11.70)	(-3.51)	(0.75)	(16.06)			
Germany	0.11	0.06	0.02	0.16	0.04	0.00	0.16	0.028	173	817
	(21.38)	(10.72)	(3.83)	(25.81)	(5.28)	(-0.23)	(24.69)			
Australia	-0.01	-0.10	-0.07	0.15	-0.07	0.02	0.13	0.001	172	1620
	(-2.13)	(-12.78)	(-10.96)	(18.67)	(-8.86)	(2.52)	(20.44)			
South Korea	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.015	167	1571
	(15.59)	(0.40)	(9.53)	(13.79)	(-5.74)	(8.26)	(11.70)			
Switzerland	0.14	0.13	0.07	0.11	0.15	-0.05	0.10	0.077	171	239
	(24.00)	(17.69)	(10.55)	(20.13)	(13.43)	(-3.87)	(17.12)			
Russia	0.04	-0.01	0.05	0.06	-0.08	0.08	0.06	0.010	168	234
	(7.30)	(-2.09)	(8.28)	(8.79)	(-8.51)	(9.68)	(6.34)			
Spain	0.18	0.17	0.09	0.14	0.17	-0.04	0.10	0.069	174	154
	(13.05)	(11.40)	(6.61)	(16.22)	(9.57)	(-2.61)	(13.06)			
Sweden	0.02	-0.03	-0.03	0.09	-0.01	-0.01	0.08	0.003	171	482
	(7.02)	(-7.74)	(-7.60)	(19.24)	(-1.70)	(-1.70)	(19.35)			
Malaysia	0.08	0.07	0.04	0.07	0.07	-0.02	0.05	0.115	164	943
	(37.94)	(36.00)	(23.09)	(38.65)	(29.25)	(-15.81)	(28.57)			
Taiwan	0.02	0.01	0.00	0.02	0.01	0.00	0.01	0.117	163	1647
	(34.43)	(23.15)	(14.21)	(26.99)	(18.84)	(-8.31)	(16.05)			
Brazil	0.17	0.16	0.06	0.15	0.19	-0.07	0.12	0.109	168	170
	(20.66)	(12.88)	(7.53)	(15.30)	(11.55)	(-5.64)	(10.41)			
Netherlands	0.08	0.03	-0.02	0.16	0.08	-0.07	0.16	0.016	174	105
	(5.43)	(0.00)	(-1.04)	(8.28)	(4.36)	(-2.31)	(7.77)			
South Africa	0.04	0.04	0.02	0.03	0.06	-0.03	0.02	0.029	170	312
	(12.99)	(14.41)	(3.43)	(8.12)	(15.99)	(-4.61)	(6.18)			
Singapore	0.12	0.09	0.06	0.11	0.08	-0.01	0.10	0.066	171	669
	(28.46)	(18.07)	(14.29)	(22.76)	(11.11)	(-0.93)	(16.53)			
Mexico	0.03	0.04	0.02	0.02	0.04	-0.01	0.01	0.085	171	116
	(14.20)	(14.68)	(0.00)	(8.31)	(13.61)	(-4.12)	(0.00)			
Norway	0.05	0.01	0.02	0.07	0.02	0.01	0.07	0.027	171	240
	(12.57)	(3.63)	(4.13)	(14.28)	(2.78)	(1.76)	(15.21)			
Indonesia	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.062	165	428
	(21.75)	(20.17)	(18.42)	(11.93)	(6.36)	(5.29)	(9.60)			
Denmark	0.10	0.08	0.05	0.09	0.08	-0.02	0.09	0.112	170	168
	(11.19)	(9.61)	(7.31)	(13.14)	(9.47)	(-2.89)	(15.12)			
Thailand	0.01	0.01	0.00	0.01	0.01	-0.01	0.01	0.033	166	564
	(11.34)	(13.69)	(5.26)	(7.96)	(14.37)	(-8.31)	(6.56)			
Finland	0.20	0.17	0.09	0.20	0.16	-0.04	0.17	0.142	171	126
	(34.89)	(23.79)	(10.51)	(43.00)	(13.33)	(-3.98)	(34.40)			
Turkey	0.07	0.05	0.01	0.09	0.03	-0.01	0.08	0.020	171	317
	(9.94)	(8.75)	(1.14)	(14.78)	(7.45)	(-1.23)	(12.20)			
Poland	0.09	0.06	0.01	0.12	0.05	-0.01	0.11	0.000	170	437
	(15.64)	(8.21)	(2.58)	(29.70)	(5.74)	(-1.83)	(27.61)			

$\log\left(\frac{ME}{BE}+1\right)$	**(1)** Quality	**(2)** Profitability	**(3)** Growth	**(4)** Safety	**(5)** Profitability	Growth	Safety	Adj R^2	N_{time}	$\bar{N}_{assets}$
Colombia	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.058	166	47
	(8.77)	(10.28)	(5.30)	(7.42)	(9.65)	(-5.60)	(1.37)			
Austria	0.14	0.12	0.05	0.15	0.14	-0.06	0.13	0.076	169	75
	(8.29)	(5.42)	(4.34)	(8.34)	(4.53)	(-3.05)	(9.59)			
Philippines	0.00	-0.03	-0.01	0.05	-0.04	0.01	0.05	0.000	166	228
	(1.11)	(-7.22)	(-4.07)	(9.05)	(-7.86)	(3.61)	(9.52)			
Argentina	0.06	0.07	0.04	0.04	0.07	-0.02	0.02	0.037	164	78
	(10.22)	(11.22)	(5.48)	(5.97)	(7.51)	(-2.02)	(3.15)			
Ireland	0.03	0.05	0.02	0.00	0.11	-0.06	-0.02	0.000	173	32
	(1.86)	(2.43)	(1.27)	(-0.00)	(4.05)	(-2.80)	(-0.56)			
Greece	0.19	0.16	0.08	0.19	0.12	-0.01	0.14	0.108	167	238
	(20.64)	(17.34)	(8.56)	(31.41)	(10.97)	(-1.11)	(30.07)			
Portugal	0.15	0.15	0.09	0.09	0.15	-0.02	0.06	0.047	173	50
	(11.06)	(9.99)	(5.72)	(5.24)	(7.12)	(-0.85)	(3.75)			
Peru	0.11	0.10	0.06	0.10	0.07	0.00	0.07	0.103	171	120
	(19.47)	(15.70)	(10.20)	(19.00)	(7.54)	(-0.40)	(10.66)			
Czech Republic	0.01	0.02	0.01	0.00	0.02	0.00	0.00	0.000	171	19
	(5.70)	(6.73)	(6.08)	(-0.30)	(4.13)	(-0.38)	(-0.92)			
New Zealand	0.12	0.02	0.06	0.16	0.02	0.04	0.16	0.043	171	122
	(13.74)	(2.08)	(6.01)	(11.25)	(1.55)	(4.78)	(10.34)			
Chile	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.001	170	190
	(6.33)	(-1.97)	(-2.43)	(11.43)	(-1.62)	(-1.39)	(9.64)			
Pakistan	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.070	167	246
	(20.33)	(14.38)	(13.22)	(23.94)	(7.74)	(0.31)	(23.69)			

4.6 The Return of Quality Stocks

4.6.1 The Return of Univariate Portfolio Sorts on Quality

In the analysis on prices above we found that the price of stocks varies monotonically with quality. We now perform a similar portfolio analysis on excess return. Although higher quality firms command higher prices, higher quality should not provide any excess return. We use the method of section 4.4 to create the 10 quality-sorted portfolios. The results of the different regressions are reported in Table A-2 of Appendix 1. Because of computer memory restrictions I had to perform regressions on one country sample at a time and it is thus not possible to infer statistical significance of a global sample.

First, I calculated the value-weighted excess return of the portfolios averaged over time and the Fama-Macbeth method for estimates and standard errors. Asness et al. do not have any lag between the measured returns and time of portfolio formation, i.e. they make no assumption on the hypothetical investor's holding period. Because we have verified the persistence of quality, I have not included a sensitivity study on the time lag between measured return and portfolio formation.

Secondly, I have performed regressions of the portfolio excess returns using three models: the CAPM market factor, the Fama-French 3-factor model and the Fama-French-Carhart 4-factor model.

For each regression I have reported the regression constant alpha which is the average excess return that is not due to sensitivity to the risk factors in the model. The t-statistic of the alpha tells us whether the portfolio generates statistically significant average abnormal returns. We also report the adjusted R^2 for the 4-factor model regression, the Sharpe ratio of the portfolio's excess returns and the information ratio for the 4-factor alpha – defined as the annualized alpha divided by its annualized standard deviation and which can be interpreted as the Sharpe ratio adjusted for hedging the four factor exposures.

The excess return for each portfolio is calculated using the Fama-Macbeth procedure by regressing the time series means on a constant and implementing the Newey-West adjusted standard errors to correct for autocorrelation. To obtain the alpha estimates on the CAPM, 3-factor and 4-factor specifications I run regressions of excess return on the factor combinations and report the intercept after controlling for the CAPM, Frama-French [5] and Fama-French-Carhart [13] risk factors in separate regression models. The risk factors are constructed for each country in line with the methodology of Asness et al. and the original authors.

$$R_{p,t} = \alpha_{CAPM} + \beta_{1,p} MKT_t + \varepsilon_{p,t} \tag{24}$$

$$R_{p,t} = \alpha_{3-factor} + \beta_{1,p} MKT_t + \beta_{2,p} SMB_t + \beta_{3,p} HML_t + \varepsilon_{p,t} \tag{25}$$

$$R_{p,t} = \alpha_{4-factor} + \beta_{1,p} MKT_t + \beta_{2,p} SMB_t + \beta_{3,p} HML_t + \beta_{4,p} MOM_t + \varepsilon_{p,t} \tag{26}$$

Where the factor-mimicking portfolios are made up as follows:

$$MKT = R_m - R_f \tag{27}$$

$$SMB = \frac{1}{3}(\text{Small Value} + \text{Small Neutral} + \text{Small Growth}) - \frac{1}{3}(\text{Big Value} + \text{Big Neutral} + \text{Big Growth}) \tag{28}$$

$$HML = \frac{1}{2}(\text{Small Value} + \text{Big Value}) - \frac{1}{2}(\text{Small Growth} + \text{Big Growth}) \tag{29}$$

$$MOM = \frac{1}{2}(\text{Small High} + \text{Big High}) - \frac{1}{2}(\text{Small Low} + \text{Big Low}) \tag{30}$$

Because the full panel data results in Table A-2 is quite big I have summarized the main findings into charts and included a short excerpt as Table 4-3. From Figure 4-10 we see the global average of mean portfolio excess returns for all countries which exhibit statistically significant difference portfolios(changing to an equal weight of country results does not change the interpretation). The very clear finding from the portfolio analysis confirms our research hypothesis that investing in high quality companies and/or shorting low quality earns a significantly positive excess return.

The difference portfolio earns an excess of 0.98% per month (12.4% annualized) and the results are monotonically increasing. Of our 44 countries we see statistically significant result in 21 countries, accounting for 64% of the global market capitalization. When controlling for market risk and the other common risk factors we see that the monotonicity across portfolios remain, but the abnormal excess returns of the difference portfolio is reduced from 98 to 70 basis points per month.

Figure 4-11 shows the 4-factor alpha for our entire sample, split into those with statistically significant estimates and not. The number of countries with significant alpha estimates for the difference portfolio has decreased to 16 countries so we can only draw inference for those shown in blue in the chart. Of the countries with significant alphas the abnormal return varies from 68 to 242 basis points per month, which is a very large magnitude.

The Sharpe ratio is a performance measure of risk-adjusted return and is calculated as the excess return divided by its standard deviation. Table A-2, exemplified in Table 4-3, show that this ratio increases with quality, The information ratio (IR) is usually a measure of portfolio returns in excess of some benchmark divided by its standard deviation, but Asness et al. calculate the IR as the 4-factor alpha divided by the standard deviation of residuals. Figure 4-12 shows the global market size weighted mean values for Sharpe ratio and information ratio across the quality sorted portfolios. The chart tells us that not only do quality stocks provide a higher return, but they are also safer in the meaning that we get higher returns per unit risk when investing in quality stocks over "junky" stocks.

The results are in line with what Asness et al. finds in their table 3 and it supports our research hypothesis 2 – that there is a positive risk-adjusted return from investing in high-quality stocks and shorting the low-quality stocks. Asness et al. use these findings to support their argument that limited market efficiency explains why quality only explain asset prices to a limited extent. They argue that if high quality stocks earn a higher risk-adjusted return than low quality stocks it must imply that market prices fail to reflect the quality characteristics. Alternatively, quality is linked to risk in a way which is not fully captured by the safety sub-factor of quality. We will explore this in the next section.

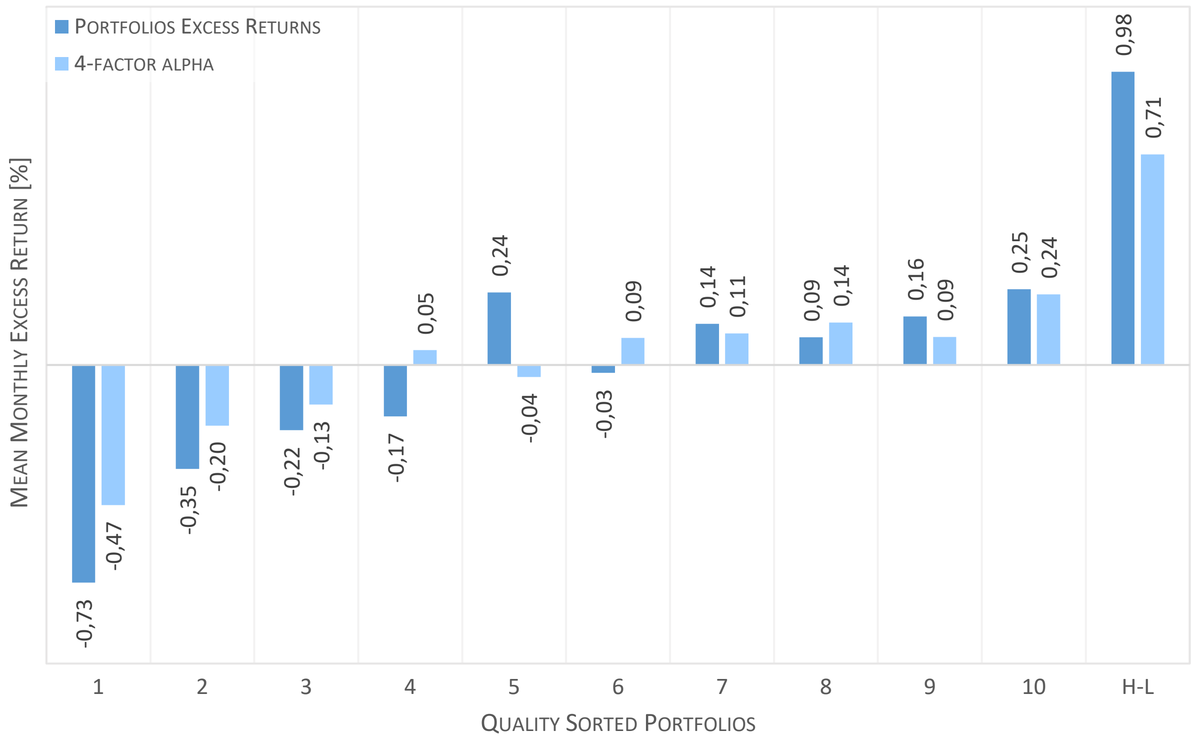


Figure 4-10: The chart shows market-weighted means of the countries with statistically significant difference portfolios for excess returns and 4-factor alphas. The blue bars represent the global average excess return of the quality-sorted portfolios along with the difference portfolios high-low. The peach bars represent the global average abnormal alpha returns when regressing portfolio excess returns on the 4 Fama-French-Carhart factors.

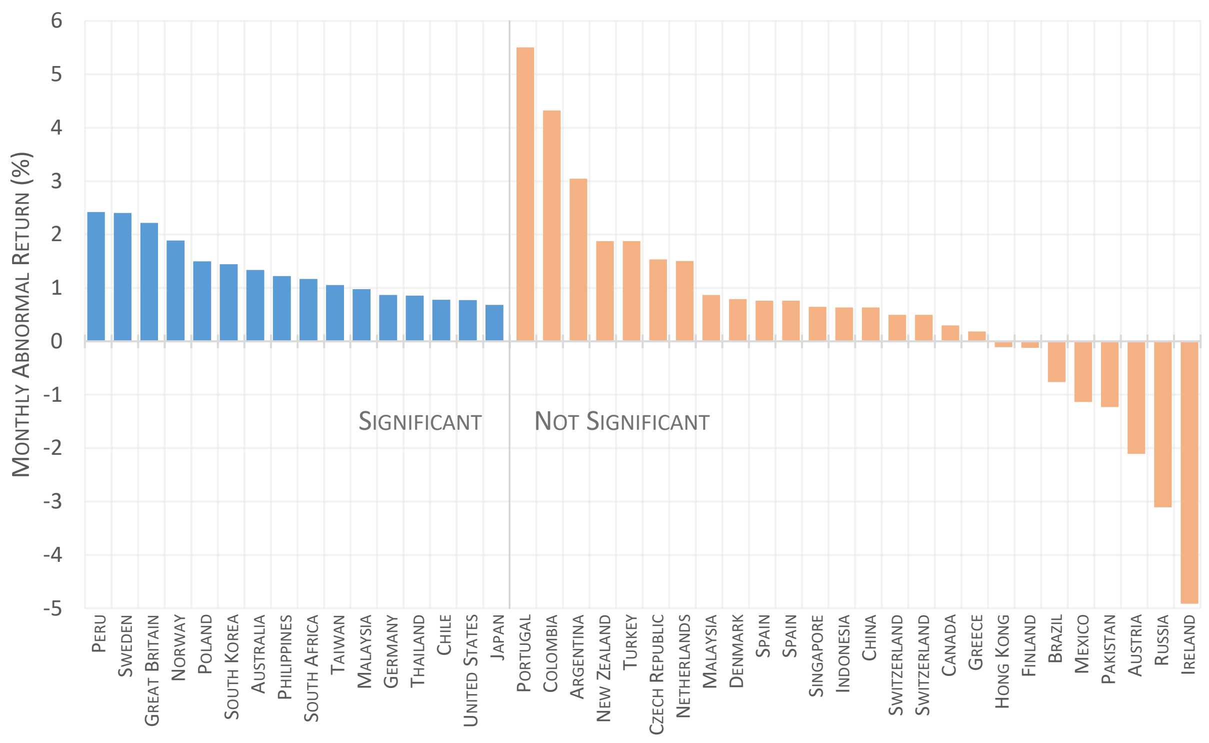


FIGURE 4-11: THIS CHART SHOWS THE FAMA-FRENCH-CARHART ABNORMAL RETURNS (ALPHA) FOR THE REGRESSIONS OF THE QUALITY-SORTED DIFFERENCE PORTFOLIO'S MONTHLY EXCESS RETURN ON THE FAMA-FRENCH-CARHART FOUR FACTORS. THE COUNTRIES IS BLUE HAVE STATISTICALLY SIGNIFICANT ALPHA ESTIMATES, WHILST THE COUNTRIES IN PEACH ARE INCLUDED ONLY FOR REFERENCE AS THEIR ESTIMATES ARE NOT SIGNIFICANTLY DIFFERENT FROM ZERO.

TABLE 4-3: RETURN ON QUALITY - EXCERPT FROM TABLE A-2 SHOWING TWO COUNTRIES OF INTEREST

THE TABLE SHOWS QUALITY SORTED PORTFOLIO EXCESS MONTHLY RETURNS ALONG WITH THE DIFFERENCE PORTFOLIO HIGH-LOW. THE ALPHAS ARE THE INTERCEPT FROM TIME SERIES REGRESSIONS OF THE MONTHLY EXCESS RETURNS USING THE CAPM, FAMA-FRENCH 3-FACTOR MODEL AND THE FAMA-FRENCH 4-FACTOR REGRESSION MODELS. RETURNS AND ALPHAS IN MONTHLY PERCENTAGE, T-STATISTICS ARE SHOWN BELOW ESTIMATES IN PARANTHESIS AND 5% STATISTICAL SIGNIFICANCE IS SHOWN IN BOLD. SHARPE RATIOS (RUN ON EXCESS RETURNS) AND INFORMATION RATIOS (RUN ON 4-FACTOR ALPHAS) ARE ANNUALIZED.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
Panel A: US Sample											
Excess Return	-1.86	-1.13	-0.88	-0.90	0.33	-0.60	-0.22	-0.39	-0.20	-0.11	**1.75**
	(-1.73)	(-1.14)	(-0.97)	(-0.99)	(0.26)	(-0.74)	(-0.26)	(-0.50)	(-0.26)	(-0.14)	(3.68)
CAPM α	**-1.18**	-0.76	-0.29	-0.32	0.03	-0.07	0.19	0.06	-0.10	0.38	**1.55**
	(-2.36)	(-1.31)	(-0.69)	(-0.74)	(0.04)	(-0.20)	(0.54)	(0.16)	(-0.36)	(1.45)	(3.34)
3-factor α	**-1.06**	-0.25	-0.48	-0.26	-0.48	-0.07	-0.01	0.02	-0.19	0.08	**1.14**
	(-2.55)	(-0.49)	(-1.13)	(-0.67)	(-0.97)	(-0.24)	(-0.03)	(0.04)	(-0.58)	(0.27)	(3.60)
4-factor α	**-0.95**	-0.68	-0.52	0.00	-0.41	-0.07	-0.15	-0.12	-0.31	-0.18	**0.77**
	(-2.57)	(-1.08)	(-1.37)	(-0.02)	(-0.94)	(-0.22)	(-0.42)	(-0.39)	(-0.91)	(-0.67)	(3.23)
Sharpe Ratio	-0.63	-0.43	-0.36	-0.37	0.11	-0.27	-0.10	-0.19	-0.10	-0.05	1.96
Information Ratio	-1.14	-0.75	-0.77	-0.01	-0.55	-0.10	-0.21	-0.16	-0.42	-0.30	1.04
Panel B: Norwegian Sample											
Excess Return	-0.35	-0.51	0.34	0.25	0.22	0.33	0.70	0.64	0.67	0.92	**1.27**
	(-0.54)	(-0.83)	(0.58)	(0.46)	(0.41)	(0.72)	(1.34)	(1.46)	(1.48)	(1.89)	(3.38)

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
CAPM α	-1.45	-3.84	0.87	0.64	0.57	0.47	**0.84**	**1.17**	0.52	**1.31**	**2.76**
	(-1.24)	(-1.35)	(1.71)	(1.22)	(1.42)	(1.49)	(2.21)	(2.60)	(0.72)	(3.54)	(2.28)
3-factor α	-0.36	-0.39	0.90	0.36	**1.04**	**0.77**	**1.02**	**0.85**	**0.96**	**1.32**	**1.68**
	(-0.59)	(-0.84)	(1.96)	(0.77)	(2.77)	(2.28)	(2.86)	(1.94)	(2.22)	(3.20)	(2.38)
4-factor α	-0.64	-0.41	0.86	0.19	0.97	0.39	**1.00**	**0.85**	0.82	**1.25**	**1.89**
	(-1.11)	(-0.81)	(1.71)	(0.42)	(2.64)	(1.04)	(2.45)	(2.00)	(1.82)	(3.42)	(2.87)
Sharpe Ratio	-0.18	-0.28	0.22	0.16	0.16	0.26	0.48	0.49	0.50	0.72	0.93
Information Ratio	-0.26	-0.21	0.50	0.12	0.75	0.31	0.64	0.54	0.51	0.95	0.73

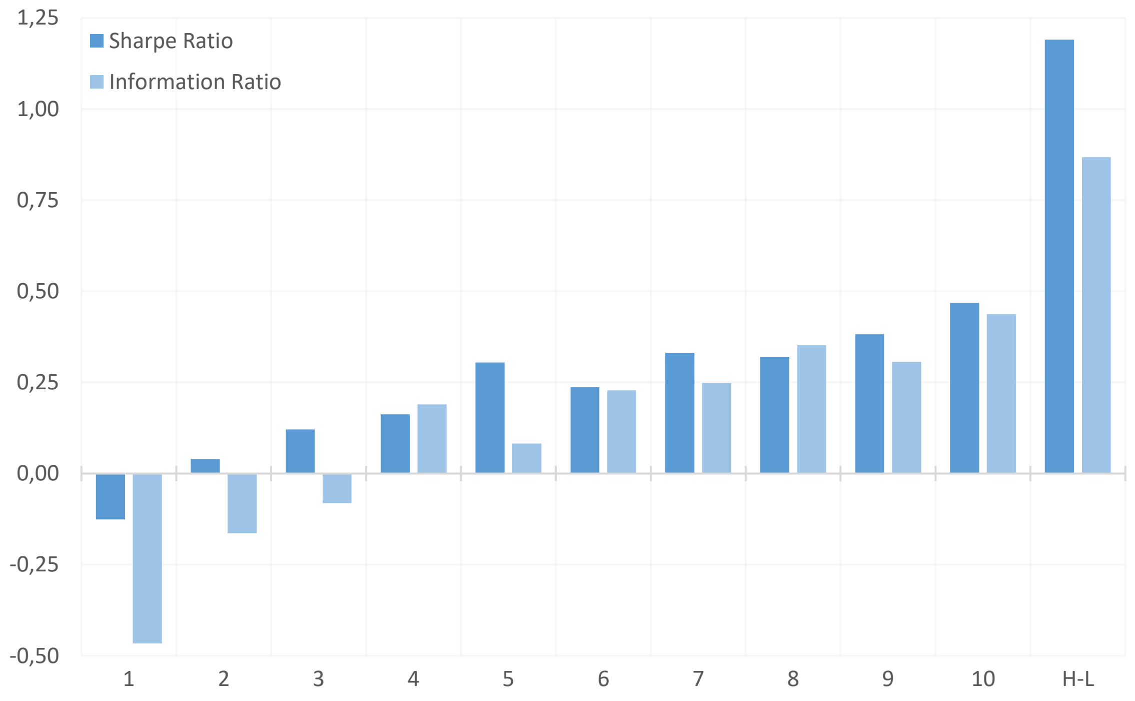


FIGURE 4-12: THIS CHART SHOWS THE GLOBAL MARKET WEIGHT MEAN OF THE SHARPE RATIOS AND THE INFORMATION RATIOS OF QUALITY-SORTED PORTFOLIOS ALONG WITH THE DIFFERENCE PORTFOLIO (HIGH-LOW). THE SHARPE RATIO IS CALCULATED ON THE MONTHLY MEAN EXCESS RETURN OF THE PORTFOLIO AND THE INFORMATION RATIO IS CALCULATED ON THE INTERCEPT (ALPHA) OF REGRESSIONS OF PORTFOLIO RETURNS ON THE FAMA-FRENCH-CARHART FOUR FACTORS.

4.6.2 The Return on the QMJ Factor

We now create the QMJ (quality minus junk) factor in the same way as Asness et al. The QMJ factor is created by taking the value weighted returns of the intersection of six portfolios formed on size and quality. Each month we form two portfolios, small and large, based on the available asset's market capitalization. In their study they use a different breakpoint for US and global stocks, but I have used the 80th percentile of market equity as breakpoint for all countries. Quality is sorted into ten portfolios and we define the top 30% as quality stocks and the bottom 30% as junk. QMJ then becomes:

$$QMJ = \frac{1}{2}(Small\ Quality - Small\ Junk) + \frac{1}{2}(Big\ Quality - Big\ Junk) \tag{31}$$

With the factor returns we run a regression of QMJ on the Fama-French-Carhart risk factors and report the alpha from regressions on the CAPM model and Fama-French 3-factor model. Regressions are run for each time period and standard errors corrected as per the Fama-Macbeth procedure and corrected for autocorrelation using the Newey-West method.

$$QMJ_t = \alpha_t + MKT_t + SMB_t + HML_t + MOM_t + \varepsilon_t \tag{32}$$

The results are shown in Table 4-4 for the full sample of countries. The key takeaway is that the QMJ factor delivers significant excess return and alpha with respect to the various risk factors. The alpha is the average excess return that is not due to sensitivity to the risk factors in the model. The t-statistic of the alpha tells us whether the QMJ portfolio generates statistically significant average abnormal returns. The excess monthly return is on average 0.24% (2.9% per year) among the countries with significant estimates and the abnormal returns (alpha) averaging at 0.31% and significantly non-zero for 32 of the 44 countries.

From the risk-factor loadings on MKT, SMB, HML and MOM we see that quality has significant negative exposures to all except the momentum factor (MOM). The negative exposure to market (MKT) and size (SMB) tells us that QMJ is long on low-beta and large stocks and/or short high-beta small cap stocks. QMJ is negatively loaded on the value factor (HML) which can be explained because quality is positively related to price whilst HML is long on cheap stocks. The exposure to momentum is small and not significant, but I kept it in the model for comparison with Asness et al. The Sharpe ratio is calculated on the excess returns, but the more interesting comparison is the information ratio which can be interpreted as the Sharpe ratio adjusted for hedging the other factor exposures. Figure 4-13 shows us that across our entire sample the QMJ factor delivers a positive alpha (except Ireland) and information ratio, even considering our short time series of data.

Generally, the results on the QMJ factor match very well with what Asness et al.'s study on 24 countries found. A QMJ portfolio that is long high-quality and short junk stocks earn a large and significant abnormal return (alpha) when controlling for some of the most used risk-factors used in literature. From the factor loadings the QMJ portfolio appear safer and this is also supported by the high information ratio (Sharpe ratio after hedging for other factor exposures) above 1.

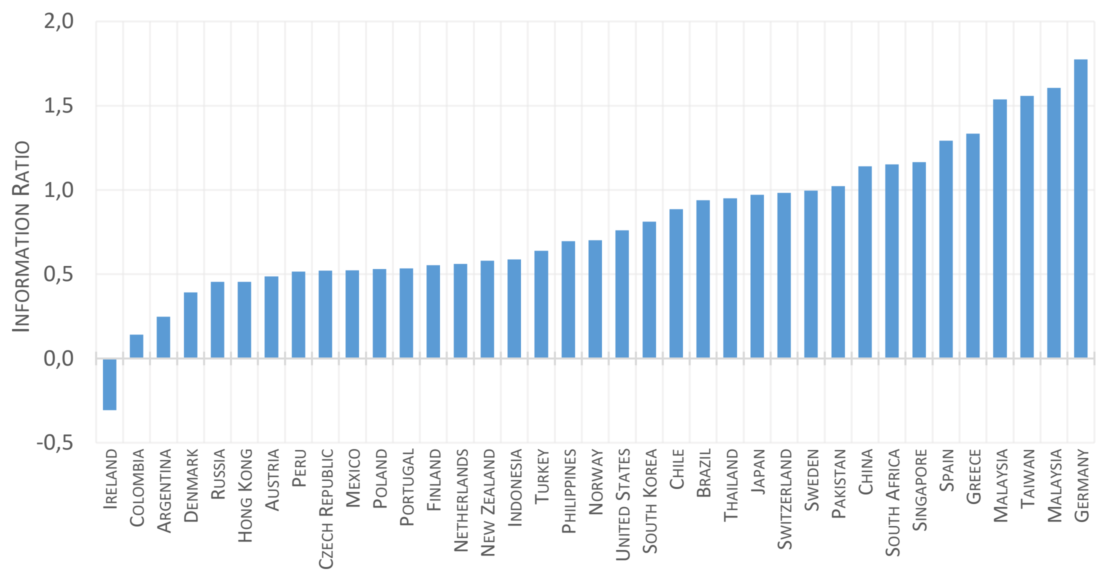


Figure 4-13: QMJ 4-factor alpha information ratios. This chart plots the Fama-French-Carhart adjusted information ratio of the Quality minus Junk (QMJ) factor.

Table 4-4: Regressions of QMJ returns on risk factors.

This table shows the QMJ portfolio returns and factor loadings on the Fama-French-Carhart risk factors along with the intercept (alphas) of time-series regressions of monthly returns on the CAPM and Fama-French 3-factor model. Returns and alphas are monthly percentages and t-statistics are shown in parentheses under the coefficient estimates.Annualized Sharpe ratio is calculated on the QMJ portfolio excess return and the information ratio on the 4-factor alpha.

QMJ	Excess Return	CAPM α	3-factor α	4-factor α	MKT	SMB	HML	MOM	Sharpe Ratio	IR	Adj. R^2
Sample Mean	**0.24**	**0.30**	**0.33**	**0.31**	**-0.06**	**-0.10**	**-0.16**	**0.09**	**0.67**	**1.06**	**0.38**
United States	0.25	0.38	0.33	0.32	-0.10	-0.19	-0.27	0.04	0.58	1.39	0.71
	(1.97)	(3.95)	(4.75)	(4.23)	(-5.77)	(-6.01)	(-4.74)	(0.70)			
Great Britain	0.64	0.66	0.71	0.62	-0.03	-0.01	-0.17	0.19	2.13	2.52	0.31
	(7.27)	(8.95)	(8.46)	(6.51)	(-0.99)	(-0.12)	(-5.07)	(3.04)			
Japan	0.07	0.05	0.18	0.18	0.01	0.05	-0.32	-0.10	0.21	0.97	0.59
	(0.84)	(0.68)	(3.34)	(3.28)	(0.78)	(2.05)	(-12.52)	(-2.68)			
Hong Kong	0.04	0.08	0.15	0.16	-0.08	-0.06	-0.16	-0.01	0.09	0.46	0.23
	(0.29)	(0.76)	(1.49)	(1.52)	(-3.35)	(-3.39)	(-3.38)	(-0.14)			
China	0.13	0.22	0.36	0.40	-0.03	-0.08	-0.29	0.20	0.24	1.14	0.56
	(1.00)	(1.99)	(3.21)	(3.82)	(-2.46)	(-2.29)	(-6.90)	(2.23)			
Canada	0.16	0.25	0.70	0.77	-0.17	-0.19	0.05	0.06	0.30	1.98	0.40
	(0.86)	(1.72)	(4.50)	(4.21)	(-5.09)	(-4.79)	(0.65)	(0.82)			
Germany	0.42	0.44	0.54	0.52	-0.12	-0.23	-0.07	0.05	1.28	1.77	0.25
	(4.52)	(4.72)	(6.09)	(5.57)	(-4.10)	(-5.56)	(-1.73)	(1.04)			
Australia	0.38	0.42	0.55	0.56	-0.12	-0.17	0.12	0.03	1.05	1.99	0.37
	(3.61)	(4.49)	(0.00)	(6.85)	(-5.94)	(-5.70)	(4.11)	(0.78)			
South Korea	0.15	0.21	0.33	0.32	-0.06	-0.01	-0.17	0.07	0.34	0.81	0.17
	(1.16)	(1.68)	(2.55)	(2.54)	(-2.65)	(-0.37)	(-3.67)	(1.00)			
Switzerland	0.35	0.33	0.31	0.32	0.03	-0.04	-0.14	-0.02	1.02	0.98	0.08

	Excess		3-factor	4-factor					Sharpe		
QMJ	Return	CAPM α	α	α	MKT	SMB	HML	MOM	Ratio	IR	Adj. R^2
	(4.91)	(4.56)	(4.53)	(4.75)	(0.92)	(-0.88)	(-3.14)	(-0.36)			
Switzerland	0.35	0.33	0.31	0.32	0.03	-0.04	-0.14	-0.02	1.02	0.98	0.08
	(4.91)	(4.56)	(4.53)	(4.75)	(0.92)	(-0.88)	(-3.14)	(-0.36)			
Russia	0.22	0.29	0.36	0.38	-0.08	-0.01	0.13	0.31	0.24	0.45	0.14
	(0.70)	(0.94)	(1.44)	(1.68)	(-1.79)	(-0.10)	(2.35)	(2.17)			
Spain	0.46	0.56	0.55	0.57	-0.20	-0.24	-0.26	-0.03	0.80	1.29	0.40
	(2.40)	(3.27)	(3.70)	(3.91)	(-4.96)	(-4.67)	(-5.90)	(-0.43)			
Spain	0.46	0.56	0.55	0.57	-0.20	-0.24	-0.26	-0.03	0.80	1.29	0.40
	(2.40)	(3.27)	(3.70)	(3.91)	(-4.96)	(-4.67)	(-5.90)	(-0.43)			
Sweden	0.51	0.56	0.49	0.38	-0.09	-0.07	-0.02	0.24	1.19	1.00	0.18
	(4.78)	(5.36)	(4.76)	(3.29)	(-3.60)	(-1.65)	(-0.41)	(3.44)			
Malaysia	0.57	0.62	0.63	0.57	-0.09	-0.12	-0.40	0.18	0.94	1.54	0.63
	(3.84)	(4.51)	(5.47)	(4.16)	(-3.69)	(-4.31)	(-10.15)	(2.60)			
Malaysia	0.30	0.40	0.48	0.46	-0.08	-0.04	-0.21	0.10	0.82	1.60	0.35
	(2.83)	(4.37)	(4.62)	(4.66)	(-2.21)	(-1.17)	(-3.87)	(1.46)			
Taiwan	0.18	0.17	0.40	0.46	0.03	-0.05	-0.45	-0.14	0.40	1.56	0.54
	(1.61)	(1.53)	(4.29)	(4.29)	(1.88)	(-1.53)	(-9.84)	(-2.07)			
Brazil	0.25	0.37	0.60	0.61	-0.09	-0.07	-0.09	-0.02	0.33	0.94	0.04
	(1.37)	(1.95)	(2.51)	(2.42)	(-2.06)	(-1.11)	(-0.98)	(-0.20)			
Netherlands	0.37	0.48	0.43	0.31	-0.12	-0.08	-0.12	0.19	0.58	0.56	0.27
	(2.34)	(3.05)	(2.82)	(2.01)	(-2.87)	(-1.72)	(-2.64)	(2.18)			
South Africa	0.30	0.33	0.44	0.43	-0.08	-0.09	-0.17	0.12	0.71	1.15	0.16
	(2.48)	(2.74)	(4.32)	(4.13)	(-2.36)	(-2.68)	(-4.80)	(1.48)			
Singapore	0.25	0.31	0.43	0.46	-0.09	-0.18	-0.01	0.13	0.50	1.16	0.33
	(1.81)	(2.40)	(3.53)	(3.56)	(-2.95)	(-5.13)	(-0.27)	(2.37)			
Mexico	0.25	0.26	0.36	0.26	0.00	-0.05	-0.16	0.15	0.48	0.52	0.11
	(1.79)	(1.72)	(2.64)	(1.73)	(-0.06)	(-0.77)	(-2.83)	(2.00)			
Norway	0.43	0.48	0.43	0.38	-0.14	-0.15	-0.10	0.08	0.73	0.70	0.12
	(2.29)	(2.74)	(2.66)	(2.40)	(-3.60)	(-1.96)	(-2.10)	(1.02)			
Indonesia	0.17	0.15	0.33	0.33	-0.03	-0.13	-0.03	0.03	0.30	0.59	0.02
	(0.94)	(0.78)	(1.97)	(1.96)	(-0.58)	(-2.13)	(-0.46)	(0.43)			
Denmark	0.46	0.55	0.37	0.21	-0.05	-0.11	-0.26	0.26	0.72	0.39	0.27
	(1.86)	(2.34)	(2.02)	(1.18)	(-1.46)	(-1.82)	(-3.55)	(3.74)			
Thailand	0.03	0.15	0.26	0.29	-0.16	-0.13	-0.18	-0.05	0.09	0.95	0.33
	(0.30)	(1.47)	(2.67)	(2.71)	(-6.00)	(-3.86)	(-4.71)	(-1.11)			
Finland	0.44	0.44	0.43	0.26	0.04	0.02	-0.13	0.25	0.88	0.55	0.12
	(3.14)	(3.18)	(3.35)	(1.80)	(1.01)	(0.33)	(-2.33)	(2.49)			
Turkey	0.28	0.31	0.38	0.29	0.00	-0.08	-0.02	0.26	0.56	0.64	0.13
	(2.14)	(2.48)	(3.41)	(2.53)	(-0.14)	(-1.46)	(-0.42)	(2.58)			
Chile	0.25	0.36	0.42	0.34	-0.12	0.02	-0.08	0.13	0.61	0.89	0.13
	(2.59)	(4.29)	(4.31)	(2.95)	(-1.90)	(0.30)	(-1.47)	(1.55)			
Poland	0.19	0.29	0.33	0.25	-0.12	-0.07	-0.04	0.18	0.34	0.53	0.25
	(1.08)	(1.72)	(2.06)	(1.57)	(-3.23)	(-2.24)	(-0.68)	(2.87)			
Colombia	0.24	0.13	0.18	0.15	0.07	-0.05	0.02	0.18	0.22	0.14	0.02
	(0.89)	(0.47)	(0.82)	(0.70)	(0.67)	(-0.32)	(0.15)	(0.86)			

QMJ	Excess Return	CAPM α	3-factor α	4-factor α	MKT	SMB	HML	MOM	Sharpe Ratio	IR	Adj. R^2
Austria	0.15	0.20	0.34	0.29	-0.11	-0.11	-0.14	0.27	0.21	0.49	0.32
	(0.82)	(1.27)	(2.23)	(2.08)	(-2.78)	(-2.13)	(-2.72)	(4.53)			
Philippines	0.14	0.27	0.48	0.48	-0.12	-0.11	-0.19	0.00	0.18	0.70	0.10
	(0.66)	(1.37)	(2.35)	(2.34)	(-2.68)	(-1.66)	(-3.13)	(-0.03)			
Argentina	-0.02	0.33	0.35	0.25	-0.12	0.05	-0.07	0.33	-0.02	0.25	0.17
	(-0.08)	(1.52)	(1.56)	(1.19)	(-3.35)	(0.80)	(-1.35)	(2.42)			
Ireland	-0.46	-0.47	-0.42	-0.49	0.01	-0.07	-0.06	0.28	-0.28	-0.31	0.05
	(-1.01)	(-1.03)	(-1.01)	(-1.19)	(0.09)	(-0.66)	(-0.88)	(2.39)			
Greece	0.56	0.50	0.77	0.89	-0.09	-0.07	-0.26	0.14	0.60	1.33	0.46
	(2.35)	(2.40)	(4.20)	(4.51)	(-1.85)	(-1.27)	(-4.87)	(1.87)			
Portugal	0.47	0.54	0.40	0.54	-0.20	0.09	0.12	0.33	0.41	0.53	0.20
	(1.53)	(1.98)	(1.45)	(1.85)	(-2.43)	(1.46)	(1.87)	(3.83)			
Peru	0.26	0.14	0.42	0.42	0.18	0.03	-0.30	0.00	0.28	0.52	0.17
	(1.07)	(0.69)	(2.13)	(2.13)	(1.51)	(0.34)	(-3.94)	(-0.01)			
Czech Republic	0.26	0.34	0.44	0.51	-0.03	0.07	-0.06	0.37	0.25	0.52	0.10
	(0.83)	(1.08)	(1.39)	(1.64)	(-0.40)	(0.81)	(-0.87)	(3.06)			
New Zealand	0.24	0.31	0.33	0.35	-0.18	-0.14	0.05	0.21	0.36	0.58	0.14
	(1.42)	(1.68)	(1.80)	(1.89)	(-3.06)	(-3.01)	(1.24)	(2.26)			
Pakistan	0.40	0.42	0.64	0.63	-0.04	-0.19	-0.14	0.02	0.57	1.02	0.17
	(1.87)	(2.02)	(3.47)	(3.48)	(-1.46)	(-3.46)	(-1.94)	(0.20)			

4.7 Time-varying Return of Quality

I also look at the intertemporal properties to see what they tell us about how quality varies over time. Although the sample data only stretches from 2005 to 2019 it does include very shifting economic periods. The boom time leading up to the financial crisis in 2008 is one environment and the following downturn lasting until 2009 another. And finally, the ten-year bull rally up until 2019. From Figure 4-14 we see the average cumulative factor returns for the global sample. The market ups and down can be seen on the black line and we also see how quality on average performs very stable during the time sample. Asness et al. describe a "flight to quality", i.e. when the market goes down investors flock to high quality stocks and drive their returns up. The same phenomenon is proven for the value factor and we see that the HML returns heavily influenced by high returns during the financial crisis. My data does not show an equally strong effect on QMJ, but we do see that the QMJ returns are in fact positive during the financial crisis. From Figure 4-15 we can see the time series of the price of quality (which we analyzed in section 4.5.2). The chart shows the that the slope coefficient of quality in regressions of price on quality increase by 100% during the financial crisis compared to the pre-crisis values.

The "slow and steady" performance of a long quality, sort junk strategy is also shown in Figure 4-16. It displays, for the US sample, the cumulative abnormal returns using the 4-factor alpha for the quality sorted difference (H-L) portfolio. We see that through bull and bear markets the investment in high quality stocks yield an almost perfectly linear return when hedging for the other factors. The figure is very much in line with the equivalent figures in Asness et al. [1]. Figure 4-18 shows the QMJ factor returns for all the countries in the sample to give an indication of the variation.

As a curiosity, I have also included the cumulative 4-factor alpha of the quality sorted portfolio for the Norwegian sample in Figure 4-17. Here we se a clear shift in abnormal returns after the financial crisis, with quality (H-L) earning huge returns after hedging for other risk factors.

In conclusion, we observe the same time-varying properties of quality as found by Asness et al. QMJ and the quality difference portfolio deliver consistent positive risk-adjusted returns over time and it looks to be robust to varying economic environments.

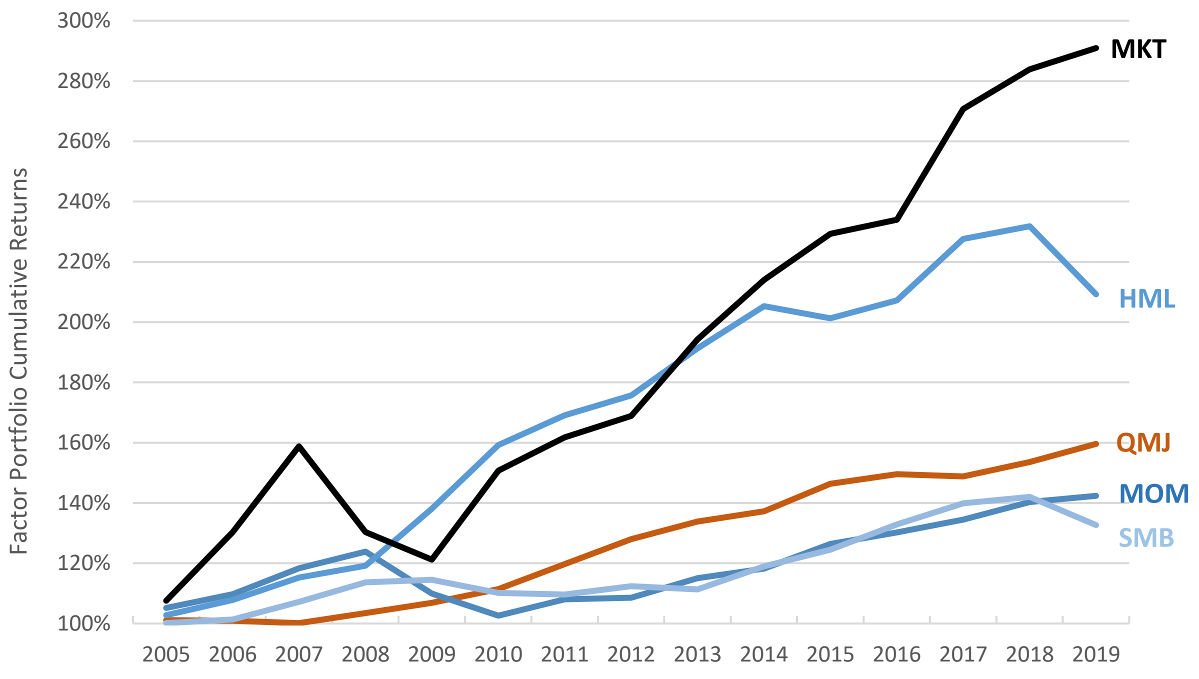


FIGURE 4-14: THE CHART SHOWS THE GLOBAL AVERAGE CUMULATIVE EXCESS RETURNS FROM THE FACTOR PORTFOLIOS FORMED EACH MONTH FOR THE FACTORS MKT (MARKET), HML (VALUE), QMJ (QUALITY), MOM (MOMENTUM) AND SMB (SIZE).

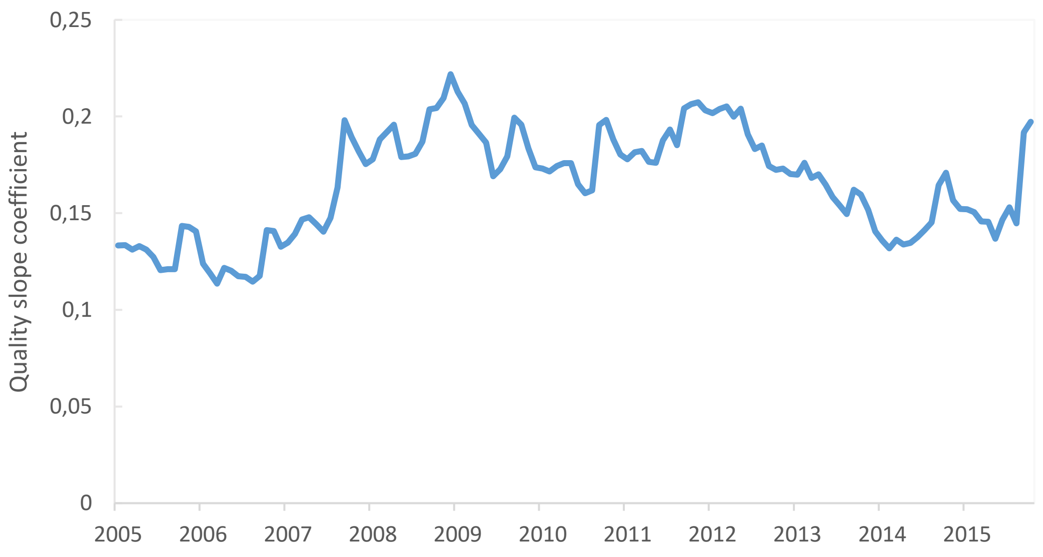


FIGURE 4-15: THE TIME-VARYING PRICE OF QUALITY. THIS CHART SHOWS THE SLOPE COEFFICIENTS FROM MONTHLY CROSS-SECTIONAL REGRESSIONS OF PRICE ON QUALITY AS DEFINED IN SECTION 4.5.2. THE SAMPLE IS US STOCKS.

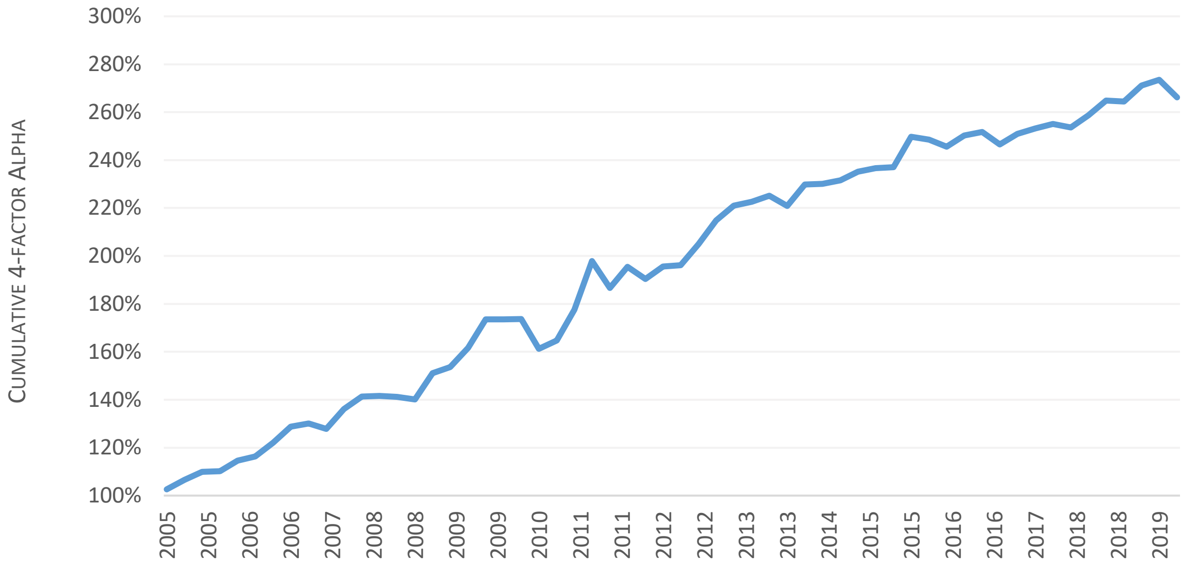


FIGURE 4-16: THIS CHART SHOWS THE CUMULATIVE 4-FACTOR ALPHA FACTOR ABNORMAL RETURNS FOR THE DIFFERENCE PORTFOLIO (HIGH-LOW) SORTED ON QUALITY. SAMPLE IS THE US STOCK MARKET.

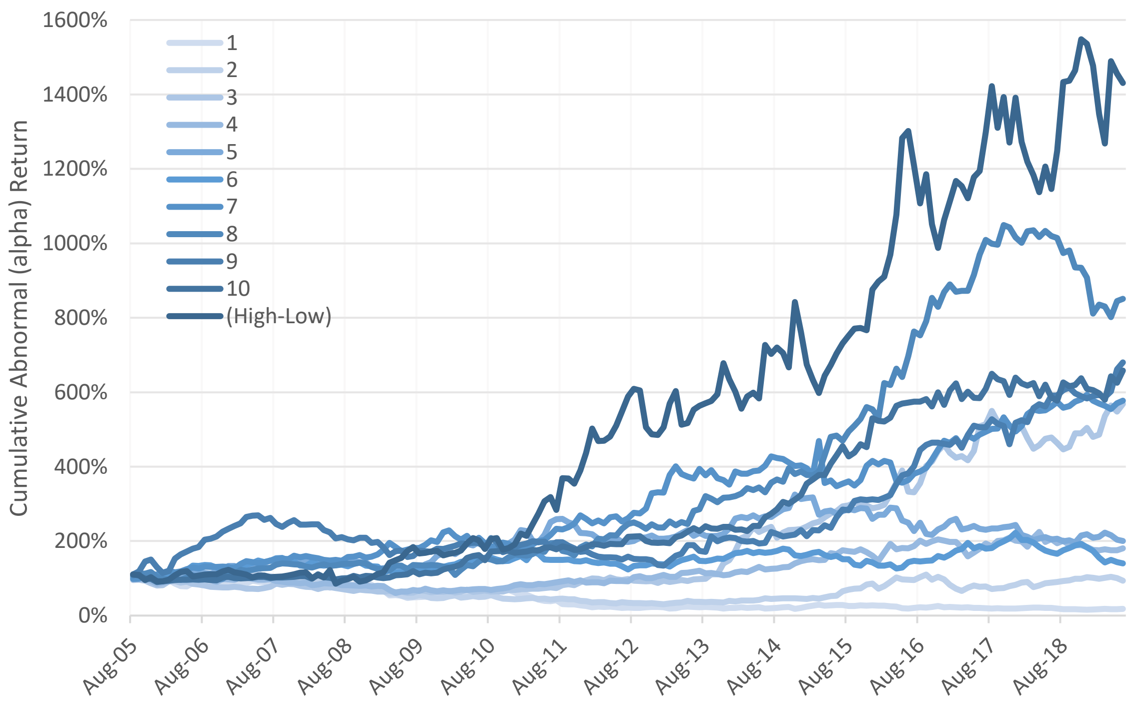


FIGURE 4-17: ABNORMAL RETURNS (ALPHA) OF THE QUALITY SORTED PORTFOLIOS WHEN ADJUSTED FOR FAMA-FRENCH-CARHART RISK FACTORS. SAMPLE IS NORWEGIAN STOCKS AND THE ESTIMATES ARE SIGNIFICANT FOR ALL PORTFOLIOS FROM 6 AND ABOVE.

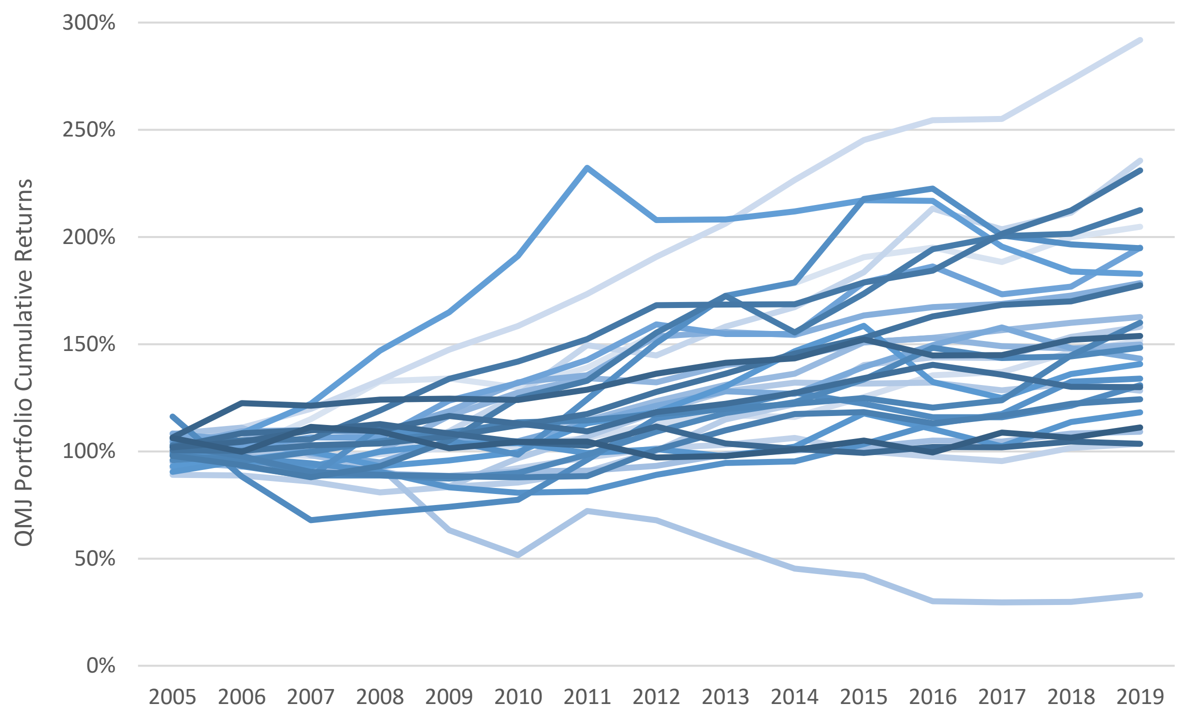


FIGURE 4-18: CUMULATIVE EXCESS RETURNS OF A QMJ PORTFOLIO FORMED EVERY MONTH. EACH LINE REPRESENTS ONE OF THE 44 COUNTRIES IN THE GLOBAL SAMPLE. IRELAND IS THE ONLY LOSS-MAKING COUNTRY.

4.8 Algorithmic Trading

In this section I explain how the QMJ factor can be used in modern investment practice. With the rise of computing power and accessible data over the last decades, we have also seen a rapid development in the vaguely defined field of quantitative finance. This involves the use of mathematical models on large datasets of financial and can refer to algorithmic trading methods, data driven research and analysis or other way of applying math to draw conclusions from datasets. The workflow of developing a quantitative investment strategy, shown in Figure 4-19, can be explained by breaking it into several models [31]. First and foremost, a quantitative investor needs large datasets of primarily financial data. Once the raw data is obtained, it must be cleansed and organized so the models can use it.

The return model, somethings called the alpha model, is the heart of the strategy. This model will build on some investment hypothesis that can be used to predict the relative movements of future returns for the asset class being analyzed. For example, we could hypothesize that the future return of a firm's stock correlates with how often it is mentioned on Twitter, by the historic stock prices or some other pattern. The return model is then tested on the data and its effectiveness can be analyzed by various statistical measures. In the analysis work we have shown how the QMJ alpha looks to be robust and could form the basis of a suitable return model.

The purpose of the risk model is to enable constructing a portfolio in line with the investors risk profile and to minimize risk of losses. This is done by evaluating the performance of the constructed portfolio against some measure of risk. The traditional measure used by Markowitz was portfolio return variance, but more modern risk models use measures like the Sharpe ratio, exposure to factors (e.g. style and sector) or portfolio drawdown to understand the portfolio risk and set limits to exposure.

The portfolio construction model combines and perform trade-off studies between the return model and the risk model combined with a model to take into account trading cost and constraints like the cost of buying a stock, slippage effects or avoiding illiquid assets. Portfolio construction can be done using for instance mean variance methods, but often the investor will optimize the portfolio construction and execution model by performing back tests. Back testing means that you run a simulation of how your algorithm would have performed using historic data. The portfolio construction model describes the rules for selecting assets and trade them at an acceptable cost and risk level and the execution model simulates the actual trading on historic data.

Using this framework, we could create a trading strategy based on the QMJ factor or include the QMJ factor in an existing strategy to take advantage of the abnormal returns we seem to achieve.

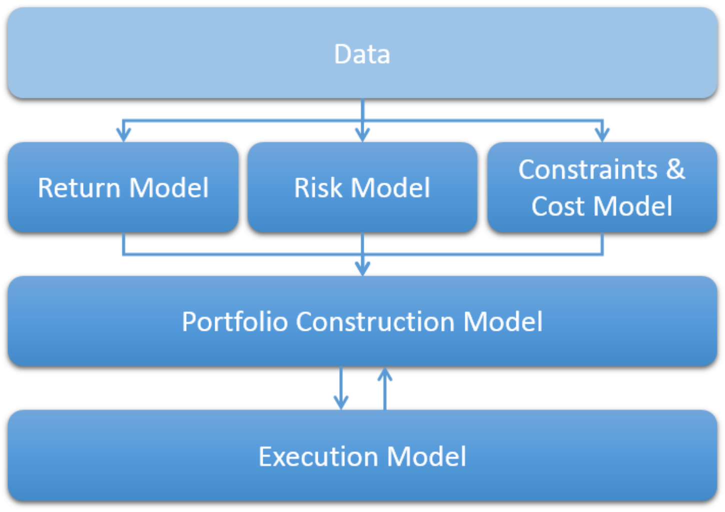


FIGURE 4-19: WORKFLOW FOR QUANTITATIVE INVESTMENT, ADOPTED FROM [31]

5 Conclusion

The goal of this thesis is to examine whether the findings for Asness et al.'s paper "quality minus junk" [1] could be replicated when increasing the dataset from 24 countries to 44. The quality factor is a proxy based on an asset pricing model where the future discounted payoffs is split into separate terms relating to profitability, growth and safety. I provide evidence confirming the two research hypothesis, namely that 1: There is a positive and significant relationship between price and quality, and 2: an abnormal risk-adjusted return can be earned by investing in high-quality stocks and shorting low-quality stocks.

The first finding on the positive relationship between price and quality shows that the model specification based on modern asset pricing has explanatory power on stock prices, but most of the cross-sectional variation in prices is still unexplained. This finding is the case in 40 out of the 44 countries examined for the sample data between 2005 and 2019.

The second finding shows that a factor-mimicking portfolio (QMJ) going long on the highest quality firms and shorting the low-quality stocks earns a significant risk-adjusted return with a Sharpe ratio after hedging for other factor exposures just above 1. The risk-adjusted alpha was positive for 43 out of the 44 countries in the sample.

The third finding of this study is that the abnormal quality returns are consistent across the time period and appears robust to changing economic environments. The price of quality stocks do increase during times of distress, indicating a "flight to quality", but the risk-adjusted alpha (when hedging for other risk factors) does not fall during the financial crisis.

This thesis confirms the results from Asness et al. using a broader sample of countries, but with a short time horizon. In their paper they conclude the abnormal returns of quality stock are due to mispricing and they are unable to find a risk-based explanation. If anything, quality stocks are less risky than lower-quality stocks as measured by Sharpe ratio. The other avenue to pursue would be preference or behavior-based explanation to why the average investor does not wish to hold quality stocks. I find that quality deliver consistent returns during times of distress as well as in times of boom. It is therefore difficult to argue that investors shy away from these stocks to avoid negative returns when the marginal utility of consumption is high. It has been proposed that investor bias like lottery preference and overconfidence [21] could be an explanation why the demand for high-quality stocks is not higher, but they are difficult to prove or disprove. Asness et al. analysis on analysts' estimates indicate (if analysts' expectations are a good proxy for market expectations) that the average investor tends to overestimate junk stocks and underestimate quality stocks.

Recommendations for further research would be to perform backtesting of a trading strategy based on QMJ to test whether the strategy delivers abnormal returns also when controlling for constraints like commissions, constraints on short trading, etc. I would also dig deeper into the price of quality. We found quality to explain very little of the cross-sectional variation in price, but by analyzing the relationship between price and the individual accounting components that make up the quality factor or regression of quality on other variables we might learn more to understand why a theoretically sound valuation model explains so little of cross-sectional variation in prices.

References

[1] C. S. Asness, A. Frazzini and L. H. Pedersen, "Quality minus junk," *Review of Accounting Studies,* pp. 34-112, March 2019.

[2] J. Montier, Value investing - tools and techniques for intelligent investment, Wiley, 2009.

[3] J. R. Graham and C. R. Harvey, "The theory and practice of corporate finance: evidence from the field," *Journal of Financial Economics,* pp. 187-243, 2001.

[4] S. Ross, "The arbitrage theory of capital asset pricing," *Journal of Economic Theory,* pp. 341-360, 1976.

[5] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *Journal of Financial Economics,* pp. 3-56, 1993.

[6] E. F. Fama and K. R. French, "A five-factor asset pricing model," *Journal of Financial Economics,* pp. 1-22, 2015.

[7] E. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," *Journal of Finance,* pp. 383-417, 1970.

[8] B. Graham and D. L. Dodd, Security Analysis, McGraw-Hill, 1934.

[9] D. Kahneman, Thinking, fast and slow, Farrar, Straus and Giroux, 2011.

[10] J. H. Cochrane, Asset Pricing, Princeton University Press, 2005.

[11] J. Y. Campbell, Financial Decisions and Markets: A Course in Asset Pricing, Princeton: Princeton University Press, 2018.

[12] W. F. Sharpe, "Determining a fund's effective asset mix," *Investment Management Review,* pp. 59-69, February 1988.

[13] M. M. Carhart, "On Persistence in Mutual Fund Performance," *The Journal of Finance,* pp. 57-82, March 1997.

[14] K. Hou, C. Xue and L. Zhang, "Replicating Anomalies," *Fisher College of Business Working Paper No. 2017-03-010,* 2017.

[15] C. R. Harvey and L. Yan, "Backtesting," *Journal of Portfolio Management,* pp. 13-28, 2015.

[16] C. S. Asness, A. Frazzini, R. Israel and T. J. Moskowitz, "Fact, Fiction and Momentum Investing," *Journal of Portfolio Management,* 2014.

[17] C. S. Asness, A. Frazzini, R. Israel and T. J. Moskowitz, "Fact, Fiction, and Value Investing," *Journal of Portfolio Management,* 2015.

[18] B. Graham, The intelligent investor, Harper, 1973.

[19] J. Hsu, V. Kalesnik and V. Viswanathan, "A framework for assessing factors and implementing smart beta strategies," *Journal of Index Investing,* pp. 89-97, 2015.

[20] R. Arnott, C. R. Harvey and H. Markowitz, "A backtesting protocol in the era of machine learning," Working Paper, 2018.

[21] Norges Bank Investment Management, "The quality factor," 2015.

[22] N. Beck, J. Hsu, V. Kalesnik and H. Kostka, "Will your factor deliver? An examination of factor robustness and implementation cost," *Financial Analysts Journal,* pp. 58-82, 2016.

[23] J. H. Cochrane, "Presidential Address: Discount Rates," *THE JOURNAL OF FINANCE,* vol. LXVI, no. 4, 2011.

[24] E. F. Fama and K. R. French, "A Five-Factor Asset Pricing Model," *Fama-Miller Working Paper Series,* 2014.

[25] R. Novy-Marx, "The other side of value: The gross profitability premium," *Journal of Financial Economics,* vol. 108, pp. 1-28, 2013.

[26] M. A. Peterson, "Estimating Standard Errors in Finance Panel Data Sets: Comparing Approaches," *The Review of Financial Studies,* pp. 435-480, January 2009.

[27] S. B. Thompson, "Simple Formulas for Standard Errors that Cluster by Both Firm and Time," *Journal of Financial Economics,* pp. 1-10, January 2011.

[28] T. G. Bali, R. F. Engle and S. Murray, Empirical Asset Pricing, Wiley & Sons, 2016.

[29] A. Ang, J. Liu and K. Schwarz, "Using Stocks or Portfolios in Tests of Factor Models," *Journal of Financial and Quantitative Analysis,* vol. 55, no. 3, pp. 709-750, 2020.

[30] J. M. Woolridge, Introductory Econometrics: A Modern Approach (5th edition), Cengage Learning, 2012.

[31] R. K. Narang, Inside the black box - the simple truth about quantitative trading, Wiley, 2009.

Appendix 1 – Table A-1

TABLE A-1: PERSISTENCE OF QUALITY MEASURE ACROSS QUALITY SORTED PORTFOLIOS.

Country	Time	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L	t-stat
Argentina	t	-1.75	-1.13	-0.69	-0.36	-0.08	0.19	0.46	0.71	1.05	1.63	3.37	(177.64)
	t + 3Y	-0.68	-0.47	-0.45	-0.27	-0.11	0.06	0.13	0.13	0.45	0.70	1.38	(14.79)
	t + 10Y	0.08	-0.12	-0.39	-0.38	-0.11	-0.17	-0.19	0.13	0.21	0.33	0.26	(1.27)
Australia	t	-1.74	-1.05	-0.67	-0.37	-0.11	0.12	0.37	0.65	1.02	1.79	3.52	(335.83)
	t + 3Y	-0.43	-0.29	-0.18	-0.13	-0.07	-0.01	0.14	0.29	0.42	0.73	1.15	(53.89)
	t + 10Y	-0.36	-0.17	-0.05	-0.04	-0.01	0.07	0.11	0.18	0.23	0.52	0.88	(29.16)
Austria	t	-1.61	-1.09	-0.76	-0.45	-0.15	0.15	0.45	0.76	1.07	1.67	3.28	(155.02)
	t + 3Y	-0.69	-0.59	-0.27	-0.22	-0.17	0.29	0.10	0.38	0.59	0.55	1.24	(12.90)
	t + 10Y	-0.65	-0.16	0.04	0.04	0.20	0.14	-0.11	0.47	0.16	-0.08	0.60	(5.97)
Brazil	t	-1.66	-0.99	-0.67	-0.43	-0.20	0.04	0.33	0.66	1.10	1.82	3.48	(205.09)
	t + 3Y	-0.76	-0.59	-0.50	-0.34	-0.26	-0.07	0.06	0.30	0.68	1.22	1.98	(39.08)
	t + 10Y	-0.48	-0.44	-0.14	-0.12	-0.23	-0.20	-0.23	-0.01	0.24	0.53	0.99	(8.21)
Canada	t	-1.83	-1.03	-0.63	-0.33	-0.09	0.15	0.39	0.65	1.00	1.72	3.55	(190.69)
	t + 3Y	-0.61	-0.29	-0.16	-0.06	0.00	0.08	0.19	0.31	0.51	0.79	1.40	(23.90)
	t + 10Y	-0.24	-0.05	0.00	-0.01	0.06	0.19	0.27	0.28	0.50	0.54	0.78	(9.26)
Chile	t	-1.58	-0.98	-0.68	-0.43	-0.19	0.04	0.27	0.59	1.05	1.93	3.51	(291.95)
	t + 3Y	-0.71	-0.50	-0.37	-0.24	-0.14	0.00	0.14	0.30	0.56	1.03	1.74	(26.93)
	t + 10Y	-0.18	-0.27	-0.06	-0.13	-0.13	0.09	0.08	0.18	0.20	0.63	0.81	(5.61)
China	t	-1.63	-1.06	-0.73	-0.44	-0.18	0.09	0.37	0.68	1.09	1.81	3.45	(506.64)
	t + 3Y	-0.74	-0.67	-0.52	-0.40	-0.30	-0.16	-0.02	0.15	0.42	0.85	1.59	(57.92)
	t + 10Y	-0.40	-0.39	-0.34	-0.38	-0.27	-0.23	-0.18	-0.22	-0.16	0.20	0.60	(11.61)
Colombia	t	-1.76	-1.01	-0.60	-0.32	-0.09	0.14	0.41	0.68	1.03	1.64	3.40	(261.99)
	t + 3Y	-0.99	-0.50	-0.51	-0.24	-0.19	-0.06	0.17	0.42	0.47	1.03	2.02	(13.21)
	t + 10Y	-0.72	-0.21	-0.22	-0.21	-0.01	-0.45	-0.54	-0.37	-0.14	0.41	1.13	(5.68)

Country	Time	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L	t-stat
Czech Republic	t	-1.55	-1.02	-0.72	-0.41	-0.13	0.12	0.39	0.70	1.00	1.65	3.19	(78.37)
	t + 3Y	-0.35	-0.17	-0.14	-0.29	-0.08	-0.01	0.37	0.31	0.19	0.28	0.64	(3.19)
	t + 10Y	-1.20	-0.23	0.29	-0.60	0.48	0.65	0.09	0.09	-0.23	-0.06	1.82	(41.56)
Denmark	t	-1.48	-0.96	-0.73	-0.53	-0.31	-0.02	0.31	0.67	1.14	1.92	3.40	(246.93)
	t + 3Y	-0.29	-0.40	-0.50	-0.38	-0.20	-0.18	0.07	0.22	0.41	1.16	1.46	(21.32)
	t + 10Y	-0.06	-0.17	-0.41	-0.42	-0.13	-0.19	0.00	0.11	0.14	0.48	0.54	(8.24)
Finland	t	-1.61	-1.03	-0.71	-0.43	-0.16	0.09	0.35	0.65	1.04	1.83	3.44	(231.98)
	t + 3Y	-0.69	-0.46	-0.34	-0.31	-0.17	-0.06	0.19	0.41	0.54	0.98	1.67	(27.84)
	t + 10Y	-0.43	-0.10	-0.22	-0.21	-0.07	0.02	-0.02	0.36	0.70	0.73	1.17	(12.31)
Germany	t	-1.72	-1.06	-0.70	-0.41	-0.14	0.12	0.40	0.69	1.06	1.76	3.48	(591.68)
	t + 3Y	-0.74	-0.46	-0.33	-0.22	-0.08	0.00	0.17	0.33	0.54	0.94	1.68	(55.67)
	t + 10Y	-0.40	-0.34	-0.20	-0.23	-0.07	0.08	0.13	0.28	0.46	0.68	1.08	(21.31)
Great Britain	t	-1.73	-1.07	-0.68	-0.38	-0.12	0.12	0.38	0.66	1.04	1.78	3.51	(678.87)
	t + 3Y	-0.85	-0.47	-0.26	-0.15	-0.10	0.05	0.16	0.35	0.56	1.03	1.88	(79.38)
	t + 10Y	-0.68	-0.25	-0.13	-0.07	-0.07	0.05	0.12	0.18	0.35	0.59	1.27	(30.70)
Greece	t	-1.63	-1.06	-0.73	-0.43	-0.15	0.11	0.37	0.66	1.04	1.83	3.46	(350.98)
	t + 3Y	-0.81	-0.60	-0.46	-0.31	-0.23	-0.04	0.13	0.40	0.47	1.09	1.90	(34.84)
	t + 10Y	-0.49	-0.50	-0.50	-0.24	-0.05	-0.04	0.02	0.26	0.28	0.67	1.16	(18.73)
Hong Kong	t	-1.64	-1.06	-0.72	-0.44	-0.18	0.08	0.37	0.68	1.10	1.82	3.46	(537.86)
	t + 3Y	-0.68	-0.57	-0.43	-0.35	-0.27	-0.18	-0.09	0.08	0.27	0.53	1.21	(56.55)
	t + 10Y	-0.38	-0.34	-0.33	-0.20	-0.25	-0.34	-0.20	-0.15	0.04	0.08	0.46	(19.61)
Hungary	t	-1.99	-1.73	-1.44	-1.28	-1.17	-1.04	-0.94	-0.85	-0.77	-0.68	1.32	(20.61)
	t + 3Y	0.24	-0.42	-0.34	-0.22	0.02	0.15	-1.14	-0.12	-0.13	-0.17	-0.93	(-4.75)
Indonesia	t	-1.77	-1.07	-0.66	-0.38	-0.12	0.15	0.41	0.69	1.04	1.73	3.49	(666.84)
	t + 3Y	-0.81	-0.68	-0.46	-0.34	-0.27	-0.10	0.01	0.14	0.38	1.05	1.86	(57.31)
	t + 10Y	-0.42	-0.39	-0.18	-0.30	-0.22	-0.03	-0.06	-0.04	0.12	0.59	1.01	(11.19)
Ireland	t	-1.71	-0.98	-0.64	-0.37	-0.13	0.11	0.39	0.66	1.03	1.70	3.40	(117.26)
	t + 3Y	-0.27	-0.22	-0.31	0.02	0.23	-0.02	-0.11	-0.05	0.07	0.33	0.60	(4.03)

Country	Time	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L	t-stat
	t + 10Y	-0.82	-0.38	-0.24	-0.09	0.20	-0.13	-0.07	0.24	-0.09	0.27	1.46	(8.53)
Japan	t	-1.64	-1.05	-0.71	-0.43	-0.17	0.08	0.35	0.66	1.06	1.85	3.49	(716.21)
	t + 3Y	-0.71	-0.58	-0.45	-0.34	-0.22	-0.09	0.06	0.24	0.49	0.96	1.67	(79.45)
	t + 10Y	-0.58	-0.48	-0.44	-0.35	-0.28	-0.14	0.01	0.10	0.33	0.73	1.31	(56.19)
Malaysia	t	-1.66	-1.02	-0.66	-0.41	-0.18	0.07	0.32	0.65	1.09	1.83	3.50	(442.96)
	t + 3Y	-0.66	-0.41	-0.33	-0.28	-0.20	-0.05	0.11	0.25	0.55	1.05	1.71	(38.23)
	t + 10Y	-0.15	-0.05	-0.11	0.02	-0.07	-0.05	0.21	0.20	0.12	0.56	0.72	(16.92)
Malaysia	t	-1.71	-1.08	-0.71	-0.41	-0.14	0.12	0.40	0.70	1.09	1.74	3.45	(582.26)
	t + 3Y	-0.63	-0.58	-0.48	-0.35	-0.17	-0.10	0.11	0.25	0.49	0.92	1.55	(31.71)
	t + 10Y	-0.26	-0.29	-0.38	-0.23	-0.22	-0.12	0.05	0.14	0.24	0.52	0.78	(40.70)
Mexico	t	-1.59	-1.07	-0.72	-0.45	-0.20	0.06	0.40	0.74	1.10	1.75	3.34	(217.21)
	t + 3Y	-0.72	-0.75	-0.62	-0.33	-0.25	-0.16	0.23	0.34	0.60	1.18	1.90	(29.80)
	t + 10Y	-0.10	-0.33	-0.64	-0.47	-0.23	-0.28	-0.11	0.12	0.28	0.58	0.67	(4.85)
Netherlands	t	-1.70	-1.01	-0.65	-0.39	-0.16	0.06	0.33	0.69	1.10	1.76	3.46	(293.37)
	t + 3Y	-0.55	-0.37	-0.24	-0.11	-0.11	-0.04	0.06	0.23	0.44	0.78	1.33	(12.56)
	t + 10Y	-0.54	0.06	0.27	0.04	-0.05	-0.05	0.27	0.21	0.38	0.11	0.62	(5.76)
New Zealand	t	-1.67	-1.04	-0.66	-0.37	-0.14	0.09	0.33	0.62	1.02	1.83	3.50	(277.97)
	t + 3Y	-0.62	-0.38	-0.37	-0.35	-0.08	0.06	0.04	0.22	0.40	0.85	1.46	(17.71)
	t + 10Y	-0.09	-0.17	-0.45	-0.18	-0.20	0.05	0.10	0.02	0.14	0.58	0.66	(3.97)
Norway	t	-1.72	-1.03	-0.65	-0.37	-0.13	0.09	0.34	0.65	1.04	1.79	3.51	(407.33)
	t + 3Y	-0.32	-0.37	-0.27	-0.15	-0.14	-0.07	0.02	0.22	0.38	0.69	1.01	(17.56)
	t + 10Y	-0.39	0.00	-0.15	-0.29	-0.08	-0.05	-0.21	-0.10	0.26	0.55	0.94	(4.83)
Pakistan	t	-1.71	-1.05	-0.69	-0.40	-0.13	0.11	0.36	0.67	1.09	1.76	3.47	(207.29)
	t + 3Y	-0.77	-0.60	-0.36	-0.34	-0.21	-0.04	0.02	0.25	0.49	0.97	1.74	(19.71)
	t + 10Y	-0.59	-0.54	-0.20	-0.21	-0.14	0.12	-0.04	0.18	0.57	0.65	1.23	(30.93)
Peru	t	-1.57	-1.00	-0.68	-0.44	-0.19	0.04	0.27	0.57	1.08	1.92	3.49	(211.13)
	t + 3Y	-0.57	-0.64	-0.26	-0.27	-0.14	-0.01	0.10	0.23	0.59	1.18	1.76	(33.59)
	t + 10Y	-0.38	-0.38	-0.40	-0.35	-0.27	-0.15	-0.06	-0.01	0.43	0.63	1.00	(5.06)
Philippines	t	-1.59	-1.07	-0.71	-0.42	-0.16	0.08	0.34	0.62	1.03	1.88	3.48	(220.86)

Country	Time	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L	t-stat
	t + 3Y	-0.84	-0.68	-0.37	-0.23	-0.14	0.05	0.24	0.28	0.52	1.06	1.90	(24.23)
	t + 10Y	-0.56	-0.45	-0.09	-0.18	-0.01	-0.04	0.00	0.37	0.42	0.42	0.98	(13.67)
Poland	t	-1.71	-1.00	-0.65	-0.38	-0.15	0.07	0.33	0.65	1.05	1.81	3.52	(260.77)
	t + 3Y	-0.39	-0.33	-0.32	-0.25	-0.18	-0.14	-0.05	0.05	0.35	0.84	1.23	(32.13)
	t + 10Y	0.05	-0.03	-0.07	-0.13	0.08	0.09	0.06	0.23	0.24	0.52	0.48	(3.85)
Portugal	t	-1.66	-0.97	-0.63	-0.37	-0.14	0.07	0.31	0.61	1.01	1.80	3.46	(164.00)
	t + 3Y	-0.63	-0.35	-0.37	-0.24	0.02	0.04	0.01	0.25	0.33	0.80	1.43	(12.26)
	t + 10Y	-0.32	-0.20	-0.48	-0.02	0.22	0.26	-0.30	0.05	0.17	0.31	0.63	(6.24)
Russia	t	-1.76	-1.01	-0.67	-0.38	-0.11	0.13	0.39	0.68	1.04	1.71	3.47	(243.41)
	t + 3Y	-0.76	-0.69	-0.49	-0.42	-0.17	-0.17	-0.04	0.25	0.28	0.66	1.41	(19.32)
	t + 10Y	-0.15	-0.26	0.00	-0.20	-0.23	0.09	-0.15	0.81	0.32	0.45	0.59	(3.66)
Singapore	t	-1.72	-1.06	-0.69	-0.40	-0.15	0.11	0.39	0.70	1.08	1.75	3.47	(488.51)
	t + 3Y	-0.47	-0.43	-0.42	-0.32	-0.19	-0.06	0.00	0.12	0.34	0.67	1.14	(27.17)
	t + 10Y	-0.29	-0.20	-0.17	-0.08	-0.10	-0.01	0.04	-0.01	0.22	0.24	0.54	(8.69)
South Africa	t	-1.71	-1.06	-0.67	-0.38	-0.12	0.11	0.36	0.65	1.03	1.80	3.51	(422.57)
	t + 3Y	-0.68	-0.57	-0.39	-0.19	-0.10	-0.02	0.15	0.24	0.42	0.82	1.50	(39.46)
	t + 10Y	-0.38	-0.24	-0.25	-0.27	-0.18	-0.07	0.07	0.23	0.28	0.27	0.62	(8.49)
South Korea	t	-1.67	-1.08	-0.73	-0.44	-0.17	0.10	0.39	0.71	1.11	1.77	3.44	(715.74)
	t + 3Y	-0.79	-0.69	-0.59	-0.42	-0.31	-0.20	-0.08	0.14	0.31	0.73	1.52	(26.20)
	t + 10Y	-0.52	-0.47	-0.32	-0.33	-0.36	-0.22	-0.19	-0.01	0.01	0.33	0.86	(47.10)
Spain	t	-1.67	-1.04	-0.69	-0.40	-0.12	0.12	0.35	0.64	1.03	1.81	3.48	(269.31)
	t + 3Y	-0.65	-0.51	-0.38	-0.23	-0.13	-0.03	0.12	0.34	0.60	1.02	1.66	(21.39)
	t + 10Y	-0.49	-0.14	0.13	0.07	-0.12	0.08	0.09	0.46	0.25	0.67	1.16	(12.20)
Spain	t	-1.67	-1.04	-0.69	-0.40	-0.12	0.12	0.35	0.64	1.03	1.81	3.48	(269.31)
	t + 3Y	-0.65	-0.51	-0.38	-0.23	-0.13	-0.03	0.12	0.34	0.60	1.02	1.66	(21.39)
	t + 10Y	-0.49	-0.14	0.13	0.07	-0.12	0.08	0.09	0.46	0.25	0.67	1.16	(12.20)
Sweden	t	-1.68	-1.06	-0.71	-0.40	-0.13	0.12	0.37	0.66	1.03	1.81	3.49	(459.65)
	t + 3Y	-0.41	-0.32	-0.30	-0.22	-0.05	0.06	0.24	0.39	0.59	0.99	1.40	(26.67)
	t + 10Y	0.04	-0.13	-0.09	0.11	0.31	0.30	0.37	0.44	0.39	0.76	0.72	(11.67)

Country	Time	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L	t-stat
Switzerland	t	-1.56	-1.03	-0.74	-0.47	-0.21	0.06	0.34	0.67	1.09	1.87	3.43	(226.51)
	t + 3Y	-0.76	-0.52	-0.45	-0.20	-0.14	-0.03	0.15	0.36	0.64	1.14	1.90	(30.43)
	t + 10Y	-0.66	-0.51	-0.32	-0.17	-0.09	0.03	0.13	0.34	0.34	0.74	1.40	(20.48)
Switzerland	t	-1.56	-1.03	-0.74	-0.47	-0.21	0.06	0.34	0.67	1.09	1.87	3.43	(226.51)
	t + 3Y	-0.76	-0.52	-0.45	-0.20	-0.14	-0.03	0.15	0.36	0.64	1.14	1.90	(30.43)
	t + 10Y	-0.66	-0.51	-0.32	-0.17	-0.09	0.03	0.13	0.34	0.34	0.74	1.40	(20.48)
Taiwan	t	-1.64	-1.08	-0.73	-0.45	-0.17	0.10	0.39	0.71	1.09	1.79	3.43	(741.48)
	t + 3Y	-0.63	-0.58	-0.48	-0.37	-0.28	-0.07	0.08	0.27	0.50	0.93	1.56	(59.58)
	t + 10Y	-0.32	-0.33	-0.28	-0.35	-0.26	-0.05	-0.03	0.03	0.20	0.40	0.72	(17.31)
Thailand	t	-1.74	-1.06	-0.67	-0.38	-0.14	0.12	0.39	0.69	1.05	1.76	3.50	(557.28)
	t + 3Y	-0.74	-0.61	-0.45	-0.27	-0.22	-0.02	0.06	0.20	0.46	0.92	1.66	(33.17)
	t + 10Y	-0.35	-0.23	-0.21	-0.22	-0.20	-0.20	-0.05	0.09	0.05	0.19	0.53	(6.10)
Turkey	t	-1.71	-1.06	-0.69	-0.42	-0.12	0.15	0.40	0.67	1.05	1.75	3.47	(242.05)
	t + 3Y	-0.81	-0.43	-0.41	-0.29	-0.07	0.09	0.18	0.38	0.57	0.92	1.73	(46.10)
	t + 10Y	-0.29	-0.12	-0.21	0.02	0.00	0.11	0.16	0.29	0.27	0.41	0.70	(11.88)
United States	t	-1.60	-1.01	-0.73	-0.47	-0.21	0.06	0.34	0.67	1.08	1.87	3.47	(313.63)
	t + 3Y	-0.76	-0.62	-0.49	-0.36	-0.20	-0.07	0.10	0.30	0.56	1.01	1.77	(56.42)
	t + 10Y	-0.42	-0.46	-0.43	-0.27	-0.10	-0.05	0.00	0.17	0.30	0.58	1.00	(58.65)

Appendix 2 – Table A-2

TABLE A-2: REGRESSIONS OF QUALITY-SORTED PORTFOLIO EXCESS RETURNS ON RISK FACTORS

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
United States	Excess Return	-1.86	-1.13	-0.88	-0.90	0.33	-0.60	-0.22	-0.39	-0.20	-0.11	1.75
		(-1.73)	(-1.14)	(-0.97)	(-0.99)	(0.26)	(-0.74)	(-0.26)	(-0.50)	(-0.26)	(-0.14)	(3.68)
	CAPM α	-1.18	-0.76	-0.29	-0.32	0.03	-0.07	0.19	0.06	-0.10	0.38	1.55
		(-2.36)	(-1.31)	(-0.69)	(-0.74)	(0.04)	(-0.20)	(0.54)	(0.16)	(-0.36)	(1.45)	(3.34)
	3-factor α	-1.06	-0.25	-0.48	-0.26	-0.48	-0.07	-0.01	0.02	-0.19	0.08	1.14
		(-2.55)	(-0.49)	(-1.13)	(-0.67)	(-0.97)	(-0.24)	(-0.03)	(0.04)	(-0.58)	(0.27)	(3.60)
	4-factor α	-0.95	-0.68	-0.52	0.00	-0.41	-0.07	-0.15	-0.12	-0.31	-0.18	0.77
		(-2.57)	(-1.08)	(-1.37)	(-0.02)	(-0.94)	(-0.22)	(-0.42)	(-0.39)	(-0.91)	(-0.67)	(3.23)
	Sharpe Ratio	-0.63	-0.43	-0.36	-0.37	0.11	-0.27	-0.10	-0.19	-0.10	-0.05	1.96
	Information Ratio	(-1.14)	(-0.75)	(-0.77)	(-0.01)	(-0.55)	(-0.10)	(-0.21)	(-0.16)	(-0.42)	(-0.30)	(1.04)
Great Britain	Excess Return	-0.77	-0.41	-0.22	-0.03	0.05	0.15	0.42	0.58	0.46	0.81	1.58
		(-1.64)	(-0.98)	(-0.50)	(-0.06)	(0.11)	(0.34)	(0.99)	(1.34)	(1.21)	(2.21)	(8.59)
	CAPM α	-1.11	-0.56	-0.58	-0.53	-0.21	-0.31	0.42	0.34	0.23	1.12	2.23
		(-3.77)	(-2.22)	(-1.99)	(-1.68)	(-0.80)	(-1.14)	(1.78)	(1.49)	(0.97)	(4.18)	(8.97)
	3-factor α	-1.04	-0.37	-0.42	-0.47	-0.10	-0.24	0.47	0.34	0.35	1.19	2.23
		(-4.11)	(-1.63)	(-1.60)	(-1.65)	(-0.44)	(-0.81)	(1.73)	(1.33)	(1.43)	(4.57)	(9.18)
	4-factor α	-1.05	-0.52	-0.49	-0.55	-0.18	-0.33	0.47	0.38	0.33	1.17	2.22
		(-4.07)	(-2.25)	(-1.91)	(-1.93)	(-0.72)	(-1.10)	(1.81)	(1.70)	(1.40)	(4.50)	(9.68)
	Sharpe Ratio	-0.55	-0.36	-0.18	-0.02	0.04	0.13	0.37	0.48	0.44	0.79	2.32
	Information Ratio	(-1.25)	(-0.66)	(-0.66)	(-0.69)	(-0.23)	(-0.42)	(0.63)	(0.53)	(0.47)	(1.67)	(2.90)
Japan	Excess Return	0.58	0.68	0.65	0.71	0.61	0.70	0.64	0.80	0.70	0.74	0.16
		(1.15)	(1.49)	(1.43)	(0.00)	(1.26)	(1.46)	(1.33)	(0.00)	(0.00)	(1.38)	(1.24)
	CAPM α	0.14	0.32	0.47	0.40	0.37	0.62	0.45	0.55	0.48	0.73	0.59
		(0.51)	(1.29)	(1.81)	(1.55)	(1.53)	(2.37)	(1.96)	(2.50)	(2.21)	(2.84)	(0.00)
	3-factor α	0.20	0.35	0.52	0.46	0.42	0.64	0.51	0.59	0.61	0.71	0.51
		(0.79)	(1.47)	(2.10)	(1.88)	(1.75)	(2.54)	(2.26)	(2.82)	(2.95)	(3.05)	(2.96)

Country		**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**H-L**
	4-factor α	-0.06	0.30	0.37	0.35	0.27	0.43	0.39	0.43	0.46	0.63	0.68
		(-0.23)	(1.24)	(1.75)	(1.41)	(1.13)	(1.70)	(1.79)	(1.98)	(2.04)	(2.77)	(3.42)
	Sharpe Ratio	0.38	0.49	0.47	0.50	0.42	0.49	0.44	0.53	0.45	0.45	0.28
	Information Ratio	(-0.08)	(0.43)	(0.61)	(0.53)	(0.40)	(0.65)	(0.59)	(0.64)	(0.68)	(0.95)	(1.08)
Hong Kong	Excess Return	0.96	0.91	0.94	0.79	1.00	0.97	0.90	0.81	0.93	0.87	-0.10
		(1.22)	(1.15)	(1.20)	(1.00)	(1.29)	(1.27)	(1.23)	(1.10)	(1.37)	(1.26)	(-0.38)
	CAPM α	1.23	0.92	0.72	-0.03	0.97	1.80	0.87	0.93	0.74	1.15	-0.08
		(1.68)	(1.88)	(1.45)	(-0.04)	(1.63)	(1.59)	(2.08)	(2.25)	(1.70)	(2.54)	(-0.14)
	3-factor α	0.62	0.68	0.77	0.29	0.71	0.86	0.69	0.79	0.44	0.64	0.02
		(1.85)	(1.68)	(2.08)	(0.58)	(1.83)	(1.93)	(1.81)	(2.25)	(1.21)	(1.39)	(0.06)
	4-factor α	0.60	0.46	0.76	0.66	0.69	0.66	0.72	0.68	0.46	0.49	-0.11
		(1.74)	(1.05)	(1.95)	(1.50)	(1.88)	(1.58)	(1.84)	(1.92)	(1.26)	(1.09)	(-0.25)
	Sharpe Ratio	0.46	0.43	0.47	0.38	0.49	0.47	0.49	0.43	0.52	0.46	-0.11
	Information Ratio	(0.45)	(0.35)	(0.60)	(0.50)	(0.57)	(0.57)	(0.67)	(0.67)	(0.43)	(0.42)	(-0.06)
China	Excess Return	1.54	1.54	1.70	1.74	1.70	1.72	1.68	1.72	1.80	1.90	0.36
		(1.58)	(1.68)	(1.82)	(1.89)	(1.79)	(1.85)	(1.77)	(1.82)	(1.98)	(2.21)	(1.43)
	CAPM α	0.10	-0.09	0.08	0.44	0.39	0.78	0.63	1.46	0.97	1.29	1.20
		(0.22)	(-0.20)	(0.19)	(0.92)	(0.93)	(1.54)	(1.45)	(2.83)	(2.77)	(2.52)	(2.11)
	3-factor α	0.36	-0.10	0.40	0.58	0.97	0.26	0.09	1.27	1.21	0.79	0.43
		(0.76)	(-0.25)	(1.01)	(1.42)	(2.32)	(0.55)	(0.15)	(2.91)	(2.69)	(1.92)	(0.79)
	4-factor α	0.70	-1.81	0.53	0.63	1.04	0.04	-0.93	1.11	2.26	1.34	0.63
		(1.12)	(-1.01)	(1.33)	(1.51)	(2.47)	(0.05)	(-0.70)	(2.54)	(1.70)	(2.05)	(1.31)
	Sharpe Ratio	0.54	0.55	0.62	0.64	0.62	0.64	0.61	0.64	0.68	0.78	0.32
	Information Ratio	(0.28)	(-0.25)	(0.32)	(0.40)	(0.56)	(0.01)	(-0.18)	(0.66)	(0.52)	(0.62)	(0.31)
Canada	Excess Return	2.04	1.79	1.50	1.39	1.37	1.08	1.20	1.30	1.15	1.35	-0.69
		(2.61)	(2.27)	(1.74)	(1.80)	(1.86)	(1.55)	(1.77)	(2.01)	(1.98)	(2.68)	(-1.64)
	CAPM α	1.70	1.69	1.13	1.32	0.95	1.16	1.02	1.14	1.05	1.39	-0.32
		(3.00)	(2.87)	(2.23)	(2.80)	(2.20)	(3.20)	(3.39)	(3.94)	(4.00)	(5.30)	(-0.66)
	3-factor α	0.89	1.06	0.55	1.04	0.68	0.70	0.76	0.82	0.89	1.14	0.26

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
		(2.07)	(2.97)	(1.42)	(3.25)	(2.06)	(2.29)	(2.46)	(2.59)	(3.31)	(5.06)	(0.67)
	4-factor α	0.84	0.93	0.47	0.85	0.77	0.75	0.84	0.99	0.88	1.14	0.30
		(1.95)	(2.55)	(1.26)	(2.58)	(2.42)	(2.47)	(2.97)	(3.43)	(3.38)	(4.93)	(0.80)
	Sharpe Ratio	1.05	0.90	0.70	0.67	0.68	0.57	0.62	0.73	0.70	0.92	-0.58
	Information Ratio	(0.67)	(0.75)	(0.40)	(0.83)	(0.69)	(0.81)	(0.98)	(1.17)	(1.09)	(1.52)	(0.23)
Germany	Excess Return	-0.11	0.22	0.43	0.80	0.30	0.62	0.75	0.79	0.91	0.96	1.07
		(-0.23)	(0.57)	(1.12)	(2.15)	(0.86)	(1.64)	(2.01)	(2.22)	(2.30)	(3.06)	(3.73)
	CAPM α	-0.50	0.58	0.48	0.82	0.11	0.88	0.90	0.05	0.82	0.72	1.21
		(-0.91)	(1.86)	(1.38)	(2.95)	(0.40)	(2.10)	(4.13)	(0.08)	(2.81)	(3.49)	(2.22)
	3-factor α	0.06	0.44	0.51	0.52	0.54	0.70	1.11	0.57	0.85	1.09	1.02
		(0.18)	(1.79)	(1.88)	(2.31)	(2.10)	(2.81)	(5.82)	(3.00)	(3.27)	(6.05)	(2.71)
	4-factor α	0.22	0.31	0.63	0.54	0.50	0.60	0.99	0.54	0.87	1.09	0.87
		(0.65)	(1.18)	(2.42)	(2.42)	(1.92)	(2.30)	(5.48)	(3.05)	(3.31)	(6.14)	(2.49)
	Sharpe Ratio	-0.07	0.19	0.39	0.77	0.29	0.60	0.74	0.84	0.86	0.98	0.98
	Information Ratio	(0.19)	(0.34)	(0.73)	(0.66)	(0.61)	(0.74)	(1.34)	(0.71)	(1.04)	(1.36)	(0.65)
Australia	Excess Return	0.36	0.82	0.60	0.83	0.62	0.95	0.67	0.84	1.01	0.81	0.45
		(0.67)	(1.29)	(0.90)	(0.90)	(0.91)	(1.59)	(1.20)	(1.53)	(1.93)	(1.72)	(1.93)
	CAPM α	-0.36	0.40	-0.16	0.10	0.52	0.90	0.72	0.74	0.87	0.62	0.99
		(-1.09)	(1.09)	(-0.42)	(0.22)	(1.33)	(2.47)	(2.47)	(2.21)	(2.64)	(1.98)	(2.87)
	3-factor α	-0.63	0.41	-0.49	-0.09	0.35	0.69	0.59	0.78	0.91	0.72	1.34
		(-1.94)	(1.15)	(-1.41)	(-0.22)	(1.01)	(2.27)	(2.05)	(2.47)	(2.73)	(2.34)	(4.17)
	4-factor α	-0.68	0.33	-0.48	-0.07	0.38	0.75	0.53	0.74	0.90	0.65	1.34
		(-2.11)	(0.93)	(-1.41)	(-0.20)	(1.11)	(2.50)	(1.90)	(2.32)	(2.76)	(2.12)	(4.11)
	Sharpe Ratio	0.23	0.48	0.34	0.47	0.35	0.56	0.44	0.57	0.75	0.60	0.57
	Information Ratio	(-0.57)	(0.26)	(-0.39)	(-0.06)	(0.33)	(0.72)	(0.60)	(0.85)	(1.04)	(0.76)	(1.08)
South Korea	Excess Return	0.47	0.93	1.24	1.32	1.22	1.22	1.34	1.41	1.46	1.12	0.65
		(0.86)	(1.74)	(2.37)	(2.50)	(2.42)	(2.44)	(2.64)	(2.75)	(2.93)	(2.34)	(2.18)
	CAPM α	-0.09	-0.07	0.19	0.91	0.75	0.90	0.25	0.88	0.65	1.49	1.58
		(-0.13)	(-0.12)	(0.35)	(2.25)	(1.52)	(1.87)	(0.60)	(2.31)	(1.88)	(3.44)	(2.50)

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
	3-factor α	-0.07	0.07	0.57	1.19	1.09	1.11	0.08	0.91	0.67	1.33	1.40
		(-0.11)	(0.14)	(1.29)	(3.39)	(2.95)	(3.03)	(0.26)	(2.70)	(2.00)	(3.38)	(2.26)
	4-factor α	-0.27	0.04	0.39	1.03	1.05	1.01	-0.15	0.56	0.52	1.17	1.44
		(-0.51)	(0.08)	(0.92)	(3.70)	(3.68)	(3.22)	(-0.50)	(1.84)	(1.55)	(3.46)	(2.34)
	Sharpe Ratio	0.24	0.51	0.65	0.75	0.68	0.66	0.74	0.78	0.80	0.64	0.73
	Information Ratio	(-0.15)	(0.03)	(0.26)	(0.98)	(0.98)	(0.93)	(-0.12)	(0.52)	(0.40)	(0.98)	(0.77)
Switzerland	Excess Return	-0.03	-0.07	0.17	0.65	0.13	0.48	0.45	0.44	0.60	0.91	0.95
		(-0.09)	(-0.20)	(0.50)	(1.81)	(0.34)	(1.41)	(1.28)	(1.22)	(1.53)	(2.27)	(3.24)
	CAPM α	-0.58	-0.20	0.01	0.20	0.22	0.20	0.10	0.33	0.40	0.51	1.09
		(-1.72)	(-0.73)	(0.06)	(0.86)	(0.66)	(0.87)	(0.40)	(1.06)	(1.49)	(1.91)	(2.81)
	3-factor α	-0.02	0.48	0.48	0.40	0.69	0.41	0.35	0.53	0.54	0.38	0.40
		(-0.06)	(1.87)	(2.26)	(1.65)	(2.46)	(1.70)	(1.21)	(1.56)	(2.02)	(1.54)	(1.13)
	4-factor α	-0.23	0.38	0.31	0.28	0.62	0.62	-0.18	0.36	0.43	0.26	0.50
		(-0.65)	(1.24)	(1.45)	(1.24)	(2.30)	(2.34)	(-0.54)	(1.17)	(1.53)	(1.04)	(1.21)
	Sharpe Ratio	-0.03	-0.07	0.17	0.60	0.11	0.47	0.38	0.39	0.51	0.73	1.02
	Information Ratio	(-0.15)	(0.39)	(0.39)	(0.32)	(0.62)	(0.56)	(-0.14)	(0.30)	(0.43)	(0.20)	(0.27)
Russia	Excess Return	0.89	1.13	1.64	1.36	1.52	1.05	1.50	1.28	1.59	1.39	0.50
		(0.92)	(1.29)	(2.04)	(1.45)	(2.23)	(1.69)	(1.78)	(1.97)	(2.41)	(2.01)	(0.74)
	CAPM α	0.71	0.50	1.21	43.50	1.62	1.33	2.36	1.64	1.09	2.52	1.80
		(0.62)	(0.60)	(1.73)	(1.14)	(2.65)	(2.62)	(2.17)	(2.12)	(1.78)	(1.47)	(0.77)
	3-factor α	-1.20	1.05	1.57	-1.36	30.26	1.03	1.50	3.99	0.87	0.94	2.14
		(-1.47)	(0.77)	(1.90)	(-1.12)	(1.15)	(1.18)	(1.88)	(1.28)	(1.07)	(0.51)	(0.98)
	4-factor α	-1.14	3.50	0.89	-0.39	1.19	-1.21	0.74	1.10	-0.21	-4.25	-3.11
		(-1.07)	(1.33)	(1.76)	(-0.77)	(1.93)	(-0.76)	(1.39)	(1.43)	(-0.23)	(-0.97)	(-0.85)
	Sharpe Ratio	0.37	0.55	0.87	0.63	0.83	0.61	0.76	0.74	0.95	0.79	0.25
	Information Ratio	(-0.30)	(0.36)	(0.36)	(-0.20)	(0.43)	(-0.19)	(0.23)	(0.33)	(-0.04)	(-0.25)	(-0.21)
Spain	Excess Return	-0.72	-0.14	0.41	0.14	0.21	0.46	0.62	0.33	0.73	1.01	1.73
		(-1.32)	(-0.25)	(0.87)	(0.32)	(0.44)	(1.07)	(1.46)	(0.84)	(2.10)	(2.96)	(4.76)
	CAPM α	-1.15	-0.32	0.66	0.12	-0.09	0.60	0.30	0.18	0.09	1.33	2.48

Country		**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**H-L**
		(-3.04)	(-0.78)	(1.77)	(0.38)	(-0.22)	(1.55)	(0.89)	(0.61)	(0.30)	(4.72)	(5.98)
	3-factor α	-0.38	0.18	0.48	0.28	0.17	0.38	0.33	0.09	0.03	0.65	1.03
		(-0.78)	(0.46)	(1.56)	(0.85)	(0.45)	(1.24)	(1.17)	(0.31)	(0.08)	(1.86)	(1.79)
	4-factor α	-0.37	0.29	0.53	0.11	0.09	0.45	0.20	0.34	0.05	0.39	0.76
		(-0.75)	(0.79)	(1.53)	(0.30)	(0.23)	(1.83)	(0.65)	(1.06)	(0.15)	(1.00)	(1.35)
	Sharpe Ratio	-0.44	-0.08	0.29	0.10	0.14	0.36	0.55	0.29	0.68	0.94	1.56
	Information Ratio	(-0.18)	(0.18)	(0.41)	(0.07)	(0.07)	(0.33)	(0.20)	(0.36)	(0.04)	(0.30)	(0.32)
Sweden	Excess Return	-0.72	0.38	0.24	0.29	0.66	0.74	0.95	1.02	1.10	1.05	1.76
		(-1.42)	(0.66)	(0.43)	(0.57)	(1.47)	(1.84)	(2.17)	(2.44)	(2.60)	(2.58)	(5.45)
	CAPM α	-1.02	1.14	-0.75	-0.47	0.68	0.12	0.59	0.84	0.74	1.09	2.12
		(-1.78)	(1.85)	(-1.35)	(-0.93)	(1.54)	(0.27)	(1.37)	(2.01)	(1.49)	(2.98)	(4.52)
	3-factor α	-0.84	1.02	0.27	0.13	0.58	0.75	1.04	1.22	1.37	1.60	2.44
		(-1.53)	(2.13)	(0.51)	(0.24)	(1.33)	(1.62)	(2.00)	(3.72)	(3.33)	(4.37)	(4.53)
	4-factor α	-1.13	0.84	0.24	0.17	0.49	0.57	0.50	0.84	1.27	1.27	2.40
		(-1.93)	(1.94)	(0.45)	(0.35)	(1.13)	(1.22)	(0.90)	(2.45)	(3.27)	(3.64)	(4.39)
	Sharpe Ratio	-0.41	0.25	0.14	0.21	0.47	0.56	0.72	0.80	0.89	0.83	1.44
	Information Ratio	(-0.56)	(0.54)	(0.13)	(0.10)	(0.31)	(0.34)	(0.30)	(0.61)	(0.92)	(0.88)	(1.23)
Malaysia	Excess Return	0.01	0.42	0.62	0.83	0.69	0.67	0.95	0.84	1.01	0.84	0.83
		(0.03)	(1.00)	(1.45)	(2.10)	(0.00)	(1.66)	(2.39)	(2.08)	(2.69)	(2.31)	(3.96)
	CAPM α	-0.12	0.48	0.84	0.90	0.74	0.98	0.97	0.90	1.22	1.05	1.17
		(-0.34)	(1.19)	(2.91)	(2.53)	(2.31)	(3.14)	(4.02)	(3.28)	(4.51)	(0.00)	(2.76)
	3-factor α	-0.23	0.26	0.97	0.96	0.88	1.13	0.98	1.05	1.25	0.85	1.08
		(-0.63)	(0.89)	(3.26)	(3.26)	(0.00)	(4.23)	(4.29)	(4.19)	(4.99)	(0.00)	(2.49)
	4-factor α	-0.29	0.25	1.04	0.83	0.77	1.06	0.75	0.92	1.09	0.69	0.98
		(-0.82)	(0.87)	(3.54)	(0.00)	(2.34)	(4.07)	(3.19)	(0.00)	(4.46)	(3.08)	(2.27)
	Sharpe Ratio	0.01	0.27	0.42	0.60	0.48	0.50	0.72	0.65	0.81	0.68	1.07
	Information Ratio	(-0.24)	(0.25)	(1.10)	(0.87)	(0.79)	(1.30)	(1.03)	(1.18)	(1.45)	(0.98)	(0.77)
Taiwan	Excess Return	0.36	0.65	0.81	0.92	0.92	0.97	0.92	0.98	0.90	0.95	0.58
		(0.63)	(1.16)	(1.49)	(1.72)	(1.79)	(1.74)	(1.69)	(1.76)	(1.63)	(1.77)	(2.26)

Country		**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**H-L**
	CAPM α	-0.33	-0.37	-0.01	0.65	0.21	0.70	0.12	0.49	0.71	0.50	0.83
		(-1.16)	(-1.45)	(-0.03)	(1.81)	(0.67)	(1.92)	(0.53)	(1.66)	(2.72)	(1.61)	(2.47)
	3-factor α	-0.40	-0.31	-0.01	0.55	0.15	0.81	0.08	0.48	0.69	0.58	0.98
		(-1.74)	(-1.47)	(-0.05)	(1.81)	(0.58)	(2.44)	(0.35)	(1.87)	(2.79)	(2.05)	(3.28)
	4-factor α	-0.36	-0.29	-0.06	0.50	0.15	0.75	0.02	0.45	0.66	0.70	1.05
		(-1.54)	(-1.41)	(-0.25)	(1.72)	(0.58)	(2.55)	(0.07)	(1.95)	(2.86)	(2.91)	(3.75)
	Sharpe Ratio	0.21	0.38	0.48	0.56	0.56	0.57	0.56	0.58	0.53	0.55	0.59
	Information Ratio	(-0.38)	(-0.37)	(-0.07)	(0.51)	(0.17)	(0.90)	(0.02)	(0.54)	(0.83)	(0.86)	(1.04)
Brazil	Excess Return	0.10	1.37	0.78	1.02	1.49	1.64	1.46	1.04	1.47	1.36	1.26
		(0.17)	(1.91)	(1.21)	(1.71)	(2.36)	(2.56)	(2.46)	(1.78)	(2.61)	(2.69)	(3.87)
	CAPM α	0.75	2.80	0.70	1.68	1.15	2.28	1.13	0.72	1.44	1.24	0.49
		(1.14)	(5.10)	(1.08)	(3.96)	(2.28)	(3.39)	(1.75)	(1.33)	(2.73)	(4.07)	(0.64)
	3-factor α	1.22	2.02	1.67	1.54	1.54	1.21	2.20	0.71	1.49	1.23	0.01
		(2.24)	(3.08)	(1.14)	(3.30)	(2.71)	(1.49)	(3.64)	(1.38)	(3.20)	(2.96)	(0.02)
	4-factor α	1.69	1.65	1.12	1.27	1.27	0.37	1.88	0.55	1.25	0.92	-0.76
		(3.21)	(2.11)	(0.79)	(2.31)	(2.29)	(0.40)	(3.39)	(1.05)	(3.12)	(2.25)	(-1.13)
	Sharpe Ratio	0.05	0.66	0.43	0.58	0.92	0.94	0.88	0.63	1.00	1.02	0.84
	Information Ratio	(0.71)	(0.63)	(0.22)	(0.68)	(0.65)	(0.12)	(1.02)	(0.30)	(0.84)	(0.58)	(-0.24)
Netherlands	Excess Return	-0.70	0.21	0.27	0.45	0.27	0.90	0.64	1.13	0.31	1.08	1.78
		(-1.23)	(0.38)	(0.51)	(0.97)	(0.62)	(2.32)	(1.49)	(2.51)	(0.72)	(2.40)	(4.46)
	CAPM α	-0.84	0.05	0.43	0.83	0.09	1.12	0.59	1.58	1.12	0.37	1.21
		(-1.50)	(0.09)	(0.76)	(1.30)	(0.16)	(2.52)	(1.05)	(2.71)	(2.07)	(0.61)	(1.42)
	3-factor α	-0.40	1.10	1.00	0.99	0.41	0.93	0.44	0.78	1.37	0.08	0.48
		(-0.46)	(1.49)	(1.75)	(1.37)	(0.76)	(1.77)	(0.64)	(1.33)	(2.64)	(0.15)	(0.49)
	4-factor α	-1.17	0.87	0.09	1.81	0.12	1.39	0.20	-0.26	0.83	0.34	0.00
		(-1.13)	(0.94)	(0.15)	(2.23)	(0.22)	(1.72)	(0.26)	(-0.34)	(1.36)	(0.51)	(1.30)
	Sharpe Ratio	-0.37	0.11	0.15	0.27	0.17	0.63	0.44	0.74	0.24	0.81	1.39
	Information Ratio	(-0.27)	(0.22)	(0.03)	(0.68)	(0.06)	(0.49)	(0.07)	(-0.08)	(0.39)	(0.14)	(0.37)
South Africa	Excess Return	0.60	0.64	1.06	0.95	0.76	1.01	0.88	1.19	1.12	1.38	0.78

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
		(1.20)	(1.76)	(3.03)	(2.62)	(1.94)	(2.79)	(2.52)	(3.59)	(3.27)	(4.32)	(2.04)
	CAPM α	0.32	1.01	1.43	0.94	0.95	0.53	0.99	1.97	0.97	1.52	1.20
		(0.53)	(2.83)	(2.87)	(2.82)	(1.95)	(1.19)	(2.45)	(4.80)	(3.09)	(2.99)	(2.09)
	3-factor α	-0.02	1.25	0.98	0.88	1.22	-0.01	0.76	1.90	1.07	1.18	1.20
		(-0.05)	(3.00)	(2.20)	(2.57)	(2.97)	(-0.01)	(2.07)	(4.33)	(3.50)	(3.09)	(2.59)
	4-factor α	0.00	1.34	1.00	0.77	1.20	0.07	0.86	1.87	0.89	1.17	1.17
		(0.01)	(3.40)	(2.15)	(2.19)	(2.93)	(0.15)	(2.43)	(4.28)	(2.61)	(3.22)	(2.68)
	Sharpe Ratio	0.45	0.54	1.11	0.87	0.66	1.03	0.88	1.22	1.17	1.54	0.63
	Information Ratio	(0.00)	(0.70)	(0.59)	(0.47)	(0.90)	(0.05)	(0.69)	(1.38)	(0.77)	(1.04)	(0.62)
Singapore	Excess Return	0.40	0.58	0.71	0.68	0.61	0.66	0.86	0.87	0.74	0.96	0.56
		(0.57)	(0.78)	(1.13)	(1.09)	(1.00)	(0.96)	(1.44)	(1.46)	(1.33)	(1.67)	(1.52)
	CAPM α	0.46	0.60	1.16	0.86	0.15	0.84	0.77	1.17	0.74	1.41	0.95
		(1.19)	(1.62)	(2.42)	(2.11)	(0.43)	(2.17)	(2.65)	(3.99)	(3.16)	(4.24)	(1.93)
	3-factor α	0.23	1.52	0.91	0.74	0.14	0.87	0.72	1.27	0.65	1.29	1.06
		(0.65)	(1.60)	(2.09)	(2.08)	(0.38)	(2.56)	(2.52)	(3.48)	(2.91)	(4.21)	(2.31)
	4-factor α	0.44	0.66	0.68	0.71	-0.27	0.59	0.74	0.90	0.60	1.08	0.65
		(0.99)	(1.76)	(1.81)	(2.08)	(-0.71)	(1.93)	(2.54)	(2.69)	(2.45)	(3.66)	(1.38)
	Sharpe Ratio	0.18	0.27	0.38	0.37	0.35	0.36	0.51	0.53	0.47	0.59	0.42
	Information Ratio	(0.24)	(0.40)	(0.50)	(0.56)	(-0.18)	(0.50)	(0.62)	(0.83)	(0.56)	(0.97)	(0.32)
Mexico	Excess Return	0.18	0.17	0.37	0.82	0.50	1.07	1.20	0.87	1.00	1.09	0.91
		(0.43)	(0.43)	(0.89)	(2.66)	(1.38)	(3.58)	(3.95)	(2.91)	(3.00)	(0.00)	(2.83)
	CAPM α	0.68	0.30	0.55	0.91	1.01	1.24	0.93	0.76	0.39	0.95	0.27
		(1.34)	(0.91)	(1.67)	(3.56)	(2.63)	(0.00)	(3.34)	(2.44)	(0.94)	(3.28)	(0.47)
	3-factor α	1.38	0.78	0.83	0.80	1.45	1.12	0.78	0.88	0.61	0.74	-0.64
		(2.92)	(2.65)	(0.00)	(3.11)	(3.92)	(0.00)	(2.18)	(2.79)	(2.75)	(2.26)	(-1.11)
	4-factor α	1.29	-0.64	0.31	0.81	1.44	0.95	0.75	0.77	0.44	0.15	-1.14
		(2.27)	(-1.17)	(0.85)	(2.53)	(2.51)	(2.82)	(1.77)	(2.85)	(1.61)	(0.37)	(-1.58)
	Sharpe Ratio	0.14	0.14	0.33	0.87	0.49	1.19	1.41	0.94	0.94	0.99	0.78
	Information Ratio	(0.66)	(-0.35)	(0.21)	(0.57)	(0.97)	(0.78)	(0.66)	(0.61)	(0.39)	(0.11)	(-0.44)

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
Norway	Excess Return	-0.35	-0.51	0.34	0.25	0.22	0.33	0.70	0.64	0.67	0.92	1.27
		(-0.54)	(-0.83)	(0.58)	(0.46)	(0.41)	(0.72)	(1.34)	(1.46)	(1.48)	(1.89)	(3.38)
	CAPM α	-1.45	-3.84	0.87	0.64	0.57	0.47	0.84	1.17	0.52	1.31	2.76
		(-1.24)	(-1.35)	(1.71)	(1.22)	(1.42)	(1.49)	(2.21)	(2.60)	(0.72)	(3.54)	(2.28)
	3-factor α	-0.36	-0.39	0.90	0.36	1.04	0.77	1.02	0.85	0.96	1.32	1.68
		(-0.59)	(-0.84)	(1.96)	(0.77)	(2.77)	(2.28)	(2.86)	(1.94)	(2.22)	(3.20)	(2.38)
	4-factor α	-0.64	-0.41	0.86	0.19	0.97	0.39	1.00	0.85	0.82	1.25	1.89
		(-1.11)	(-0.81)	(1.71)	(0.42)	(2.64)	(1.04)	(2.45)	(2.00)	(1.82)	(3.42)	(2.87)
	Sharpe Ratio	-0.18	-0.28	0.22	0.16	0.16	0.26	0.48	0.49	0.50	0.72	0.93
	Information Ratio	(-0.26)	(-0.21)	(0.50)	(0.12)	(0.75)	(0.31)	(0.64)	(0.54)	(0.51)	(0.95)	(0.73)
Indonesia	Excess Return	1.03	1.68	1.88	1.61	1.83	1.64	1.38	1.60	1.94	1.82	0.79
		(2.15)	(3.36)	(3.52)	(3.86)	(3.70)	(3.39)	(3.08)	(3.07)	(3.57)	(3.14)	(2.08)
	CAPM α	1.38	1.32	1.14	1.46	2.06	1.61	1.53	1.47	2.81	1.80	0.42
		(2.85)	(2.82)	(3.50)	(4.84)	(4.25)	(4.39)	(4.00)	(3.54)	(3.54)	(4.13)	(0.89)
	3-factor α	1.30	0.95	0.85	0.99	1.75	1.55	1.45	1.19	2.17	1.95	0.64
		(3.35)	(2.46)	(2.46)	(3.21)	(4.39)	(4.01)	(4.17)	(3.03)	(5.13)	(5.49)	(1.42)
	4-factor α	1.17	0.97	0.69	0.98	1.61	1.51	1.37	1.19	2.19	1.80	0.63
		(3.16)	(2.46)	(1.87)	(3.09)	(4.42)	(3.82)	(3.53)	(3.47)	(5.45)	(5.73)	(1.34)
	Sharpe Ratio	0.73	1.18	1.27	1.20	1.22	1.16	0.94	1.11	1.19	1.19	0.67
	Information Ratio	(0.95)	(0.65)	(0.53)	(0.83)	(1.23)	(1.20)	(0.99)	(0.99)	(1.79)	(1.41)	(0.35)
Denmark	Excess Return	-0.64	0.28	-0.10	-0.13	0.19	0.27	0.70	0.58	0.69	0.96	1.61
		(-1.06)	(0.44)	(-0.18)	(-0.23)	(0.35)	(0.60)	(1.46)	(1.13)	(1.55)	(2.08)	(3.97)
	CAPM α	-0.80	0.38	-0.22	-0.14	0.25	0.87	0.65	0.26	0.79	0.85	1.65
		(-1.53)	(0.66)	(-0.42)	(-0.29)	(0.61)	(1.63)	(1.27)	(0.56)	(1.74)	(1.87)	(2.90)
	3-factor α	-0.37	1.38	0.48	-0.01	0.39	0.46	1.23	0.49	1.07	0.51	0.89
		(-0.44)	(2.20)	(1.22)	(-0.02)	(0.79)	(0.78)	(2.72)	(1.09)	(2.88)	(1.06)	(0.92)
	4-factor α	-0.36	1.19	0.77	-0.28	0.35	0.65	0.86	0.31	0.54	0.43	0.79
		(-0.40)	(1.93)	(1.49)	(-0.48)	(0.61)	(1.18)	(1.63)	(0.65)	(1.27)	(0.86)	(0.76)
	Sharpe Ratio	-0.35	0.17	-0.07	-0.10	0.14	0.19	0.54	0.44	0.57	0.74	1.24

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
	Information Ratio	(-0.12)	(0.51)	(0.36)	(-0.13)	(0.21)	(0.33)	(0.49)	(0.18)	(0.33)	(0.25)	(0.25)
Thailand	Excess Return	0.57	0.76	0.78	0.82	0.93	1.04	1.07	0.90	0.93	1.02	0.45
		(1.28)	(1.55)	(1.69)	(1.85)	(1.88)	(2.25)	(2.18)	(1.98)	(2.21)	(2.38)	(2.13)
	CAPM α	0.30	0.65	0.62	0.29	1.02	0.80	0.99	0.97	0.92	1.03	0.73
		(1.03)	(1.80)	(2.22)	(1.10)	(4.01)	(3.12)	(3.87)	(3.92)	(3.89)	(4.39)	(2.46)
	3-factor α	0.22	0.60	0.83	0.75	1.02	0.83	0.86	1.36	1.26	1.23	1.00
		(0.84)	(2.18)	(2.97)	(2.73)	(4.20)	(3.48)	(3.24)	(5.48)	(4.88)	(5.47)	(2.96)
	4-factor α	0.29	0.58	0.72	0.61	0.87	0.74	0.63	1.24	1.20	1.15	0.85
		(1.06)	(2.05)	(2.45)	(2.17)	(3.61)	(3.37)	(2.51)	(5.12)	(5.05)	(5.38)	(2.67)
	Sharpe Ratio	0.41	0.52	0.56	0.63	0.72	0.84	0.83	0.73	0.80	0.89	0.59
	Information Ratio	(0.32)	(0.61)	(0.75)	(0.66)	(0.94)	(0.91)	(0.78)	(1.48)	(1.67)	(1.59)	(0.82)
Finland	Excess Return	-0.55	0.02	-0.01	0.40	0.77	0.81	0.63	0.95	0.65	0.92	1.47
		(-1.04)	(0.05)	(-0.02)	(0.84)	(1.65)	(1.87)	(1.48)	(2.22)	(1.47)	(2.35)	(3.95)
	CAPM α	-0.72	-1.29	-1.16	-0.75	0.54	0.79	0.69	1.07	-0.97	0.65	1.38
		(-0.85)	(-1.47)	(-1.46)	(-1.38)	(1.11)	(1.34)	(1.77)	(2.20)	(-0.95)	(1.29)	(1.39)
	3-factor α	0.91	-0.07	0.45	0.39	0.87	0.76	1.42	1.08	0.07	1.10	0.19
		(0.93)	(-0.10)	(0.53)	(0.69)	(1.75)	(1.09)	(3.10)	(2.01)	(0.13)	(2.09)	(0.17)
	4-factor α	1.01	0.17	1.05	0.21	1.37	0.46	1.24	1.32	0.23	0.89	-0.12
		(1.05)	(0.24)	(1.33)	(0.32)	(1.73)	(0.54)	(2.24)	(2.11)	(0.34)	(1.69)	(-0.12)
	Sharpe Ratio	-0.31	0.01	-0.01	0.26	0.54	0.60	0.44	0.64	0.44	0.69	1.22
	Information Ratio	(0.27)	(0.06)	(0.40)	(0.09)	(0.66)	(0.19)	(0.62)	(0.64)	(0.11)	(0.42)	(-0.03)
Turkey	Excess Return	0.86	1.08	1.36	1.35	1.46	1.38	1.63	1.51	1.58	1.86	1.00
		(1.40)	(1.68)	(2.25)	(2.30)	(2.60)	(2.11)	(2.58)	(2.46)	(2.52)	(3.23)	(3.04)
	CAPM α	-0.48	0.42	1.63	-1.07	3.85	0.50	0.96	1.63	0.53	2.83	3.31
		(-0.70)	(0.72)	(2.33)	(-0.45)	(2.09)	(0.80)	(1.63)	(2.32)	(0.77)	(2.10)	(2.31)
	3-factor α	-0.14	0.42	2.37	1.45	2.16	0.73	2.04	1.90	0.16	2.73	2.88
		(-0.22)	(0.64)	(3.52)	(2.22)	(3.26)	(1.29)	(3.21)	(2.67)	(0.25)	(2.79)	(2.61)
	4-factor α	0.07	0.46	2.74	0.90	2.42	0.24	1.55	1.78	0.16	1.94	1.87
		(0.11)	(0.62)	(3.39)	(1.30)	(3.50)	(0.38)	(2.49)	(2.62)	(0.24)	(3.13)	(1.96)

Country		**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**H-L**
	Sharpe Ratio	0.46	0.55	0.72	0.77	0.75	0.70	0.86	0.80	0.86	1.08	0.86
	Information Ratio	(0.03)	(0.17)	(1.16)	(0.43)	(1.13)	(0.09)	(0.66)	(0.68)	(0.07)	(0.96)	(0.58)
Chile	Excess Return	0.49	0.70	0.71	0.69	0.75	0.81	0.77	0.75	0.91	1.22	0.73
		(1.46)	(2.00)	(2.66)	(2.35)	(2.90)	(3.39)	(3.51)	(3.28)	(3.89)	(4.54)	(3.41)
	CAPM α	0.27	0.31	0.49	0.52	0.64	0.69	0.73	0.70	0.92	1.07	0.80
		(1.30)	(1.38)	(2.03)	(2.81)	(3.24)	(3.30)	(3.56)	(4.22)	(5.68)	(3.97)	(2.93)
	3-factor α	0.26	0.25	0.46	0.20	0.83	0.52	0.75	0.89	1.08	1.16	0.90
		(1.35)	(1.08)	(1.85)	(1.21)	(3.42)	(2.02)	(4.52)	(5.55)	(6.73)	(4.89)	(3.21)
	4-factor α	0.38	0.22	0.47	0.06	0.82	0.45	0.64	0.94	1.01	1.15	0.78
		(1.73)	(0.95)	(1.62)	(0.34)	(3.42)	(1.69)	(3.76)	(5.67)	(5.91)	(4.57)	(2.48)
	Sharpe Ratio	0.50	0.74	0.86	0.88	0.97	1.05	1.17	1.06	1.31	1.54	0.83
	Information Ratio	(0.37)	(0.24)	(0.48)	(0.09)	(1.06)	(0.61)	(0.97)	(1.43)	(1.31)	(1.38)	(0.61)
Poland	Excess Return	0.47	0.57	0.74	0.61	0.56	0.88	0.49	0.63	0.82	0.76	0.30
		(0.53)	(0.78)	(1.09)	(0.82)	(0.92)	(1.35)	(0.71)	(0.93)	(1.40)	(1.33)	(0.60)
	CAPM α	-1.09	0.42	0.56	0.55	0.36	0.60	0.73	-0.10	0.87	0.93	2.01
		(-1.50)	(0.66)	(1.27)	(0.90)	(0.78)	(1.29)	(1.35)	(-0.24)	(2.09)	(2.11)	(2.77)
	3-factor α	-0.94	0.85	0.61	0.52	0.20	0.41	1.04	0.50	0.86	0.68	1.61
		(-1.46)	(1.58)	(1.33)	(0.81)	(0.46)	(0.84)	(1.76)	(1.08)	(1.84)	(2.41)	(2.40)
	4-factor α	-1.09	0.86	0.67	-2.19	0.06	0.01	0.70	0.47	0.69	0.41	1.50
		(-1.71)	(1.72)	(1.38)	(-0.79)	(0.11)	(0.01)	(1.14)	(0.88)	(1.61)	(1.30)	(2.22)
	Sharpe Ratio	0.19	0.29	0.40	0.31	0.30	0.51	0.27	0.38	0.50	0.52	0.20
	Information Ratio	(-0.39)	(0.51)	(0.37)	(-0.20)	(0.03)	(0.00)	(0.37)	(0.25)	(0.40)	(0.34)	(0.58)
Colombia	Excess Return	1.45	1.24	1.02	1.02	0.71	0.81	1.09	1.42	1.48	1.48	0.03
		(3.08)	(2.19)	(2.17)	(2.57)	(2.00)	(1.70)	(1.84)	(3.85)	(3.25)	(2.96)	(0.07)
	CAPM α	1.35	0.01	0.30	1.42	0.08	0.30	1.77	3.42	2.00	1.78	0.44
		(1.63)	(0.02)	(0.61)	(1.57)	(0.13)	(0.68)	(2.36)	(1.46)	(3.02)	(3.02)	(0.43)
	3-factor α	-56.24	1.09	0.87	0.78	-7.33	-0.67	0.61	-1.90	4.11	-1.13	55.11
		(-0.98)	(1.01)	(1.37)	(1.54)	(-0.75)	(-0.57)	(0.57)	(-0.63)	(1.66)	(-0.60)	(0.96)
	4-factor α	-3.18	-0.75	0.60	4.02	5.93	9.80	2.37	1.07	4.07	1.15	4.32

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
		(-0.47)	(-0.37)	(0.36)	(1.29)	(0.94)	(1.45)	(1.54)	(0.62)	(1.62)	(1.70)	(0.62)
	Sharpe Ratio	0.91	0.77	0.68	0.68	0.55	0.52	0.74	1.16	1.01	0.94	0.02
	Information Ratio	(-0.08)	(-0.09)	(0.07)	(0.42)	(0.39)	(0.74)	(0.41)	(0.21)	(0.58)	(0.48)	(0.16)
Austria	Excess Return	0.06	0.41	-0.06	0.56	0.52	0.81	0.68	0.63	1.05	0.35	0.29
		(0.09)	(0.86)	(-0.12)	(1.09)	(0.96)	(1.56)	(1.42)	(1.45)	(2.79)	(0.79)	(0.59)
	CAPM α	-0.24	-0.36	-0.14	1.28	1.08	2.82	0.52	0.58	0.78	0.76	1.00
		(-0.31)	(-0.75)	(-0.28)	(1.31)	(1.70)	(1.17)	(0.99)	(1.05)	(1.99)	(1.29)	(1.25)
	3-factor α	1.83	-0.42	0.17	0.36	1.64	1.38	-0.22	0.06	-0.06	0.28	-1.55
		(1.54)	(-0.97)	(0.22)	(0.47)	(1.56)	(2.24)	(-0.27)	(0.09)	(-0.11)	(0.54)	(-1.29)
	4-factor α	1.05	-4.74	3.20	-1.39	0.30	2.11	1.11	-1.66	0.18	-1.05	-2.10
		(0.94)	(-1.10)	(1.24)	(-1.47)	(1.05)	(2.04)	(1.25)	(-1.76)	(0.17)	(-1.24)	(-1.55)
	Sharpe Ratio	0.03	0.26	-0.04	0.32	0.32	0.49	0.43	0.44	0.74	0.25	0.15
	Information Ratio	(0.28)	(-0.23)	(0.50)	(-0.35)	(0.29)	(0.61)	(0.34)	(-0.33)	(0.05)	(-0.26)	(-0.35)
Philippines	Excess Return	1.29	1.80	1.58	1.46	0.84	1.70	1.76	1.76	1.39	1.74	0.45
		(2.04)	(3.19)	(3.04)	(2.91)	(1.39)	(3.07)	(3.06)	(3.18)	(3.27)	(3.32)	(1.02)
	CAPM α	1.04	1.72	1.94	1.21	0.31	1.83	4.18	1.22	2.10	1.67	0.63
		(2.07)	(2.97)	(3.50)	(2.02)	(0.57)	(3.15)	(2.01)	(2.11)	(4.44)	(3.98)	(0.96)
	3-factor α	0.37	1.45	1.74	1.27	-0.33	1.83	1.90	1.50	1.94	1.56	1.19
		(1.01)	(2.81)	(3.55)	(2.24)	(-0.41)	(2.85)	(3.31)	(3.09)	(3.91)	(3.72)	(2.20)
	4-factor α	0.33	1.55	1.64	1.79	0.00	1.66	1.81	1.40	2.00	1.56	1.22
		(0.99)	(2.77)	(3.11)	(2.25)	(0.01)	(2.53)	(2.88)	(2.73)	(4.04)	(3.32)	(2.15)
	Sharpe Ratio	0.70	1.06	1.06	0.95	0.49	1.09	1.11	1.15	1.07	1.28	0.27
	Information Ratio	(0.21)	(0.76)	(0.80)	(0.66)	(0.00)	(0.71)	(0.91)	(0.72)	(1.23)	(0.87)	(0.54)
Argentina	Excess Return	1.92	1.82	2.49	1.79	2.67	2.69	2.68	2.51	1.87	2.33	0.40
		(2.27)	(2.49)	(3.20)	(2.34)	(3.40)	(3.76)	(3.52)	(4.08)	(2.95)	(4.03)	(0.68)
	CAPM α	-0.24	-0.78	0.24	1.95	1.94	1.14	3.73	1.92	1.95	1.79	2.03
		(-0.19)	(-0.71)	(0.24)	(2.39)	(1.96)	(1.56)	(3.93)	(1.86)	(2.42)	(2.33)	(1.29)
	3-factor α	0.06	-1.71	-0.62	1.45	2.52	0.76	5.44	2.08	4.94	1.57	1.52
		(0.03)	(-1.38)	(-0.46)	(1.19)	(1.42)	(0.89)	(3.98)	(1.40)	(1.96)	(1.65)	(0.63)

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
	4-factor α	-0.33	-1.08	0.15	3.67	3.43	1.09	5.31	3.55	0.94	2.71	3.05
		(-0.17)	(-0.76)	(0.10)	(2.35)	(1.70)	(0.99)	(3.72)	(1.98)	(0.60)	(2.14)	(1.14)
	Sharpe Ratio	0.71	0.80	1.13	0.81	1.27	1.42	1.32	1.27	0.98	1.33	0.17
	Information Ratio	(-0.05)	(-0.20)	(0.03)	(0.86)	(0.57)	(0.29)	(1.44)	(0.80)	(0.16)	(0.73)	(0.42)
Ireland	Excess Return	0.46	1.49	0.97	-0.19	0.03	-0.34	0.34	-0.55	-0.06	0.75	0.28
		(0.54)	(1.71)	(1.32)	(-0.19)	(0.03)	(-0.34)	(0.40)	(-0.78)	(-0.07)	(1.06)	(0.34)
	CAPM α	2.17	14.90	3.14	2.91	0.37	0.00	-1.31	-3.55	8.18	-0.33	-2.50
		(0.92)	(1.08)	(1.25)	(1.45)	(0.30)	(1.12)	(-0.31)	(-2.05)	(1.07)	(-0.24)	(-0.88)
	3-factor α	2.22	4.98	1.95	20.17	4.16	-1.36	17.18	-0.76	3.94	10.02	7.80
		(0.58)	(2.28)	(0.50)	(1.13)	(0.63)	(-0.26)	(1.33)	(-0.35)	(0.69)	(1.49)	(1.04)
	4-factor α	-0.33	1.24	-0.47	3.40	1.28	1.56	-10.30	-0.36	-1.05	-5.25	-4.91
		(-0.17)	(1.02)	(-0.27)	(1.90)	(0.70)	(0.00)	(-0.95)	(-0.30)	(-0.59)	(-0.98)	(-0.88)
	Sharpe Ratio	0.16	0.54	0.40	-0.06	0.01	-0.11	0.13	-0.22	-0.02	0.35	0.09
	Information Ratio	(-0.04)	(0.25)	(-0.06)	(0.61)	(0.15)	(0.34)	(-0.13)	(-0.06)	(-0.15)	(-0.22)	(-0.20)
Greece	Excess Return	-0.33	0.54	0.02	0.38	0.22	0.12	0.44	0.65	0.42	0.60	0.92
		(-0.45)	(0.88)	(0.03)	(0.57)	(0.33)	(0.19)	(0.64)	(1.21)	(0.81)	(1.13)	(2.32)
	CAPM α	-0.23	0.49	-0.27	0.80	0.22	0.28	0.88	0.78	0.52	0.40	0.62
		(-0.33)	(0.80)	(-0.50)	(1.19)	(0.36)	(0.51)	(1.73)	(2.08)	(1.23)	(1.05)	(0.96)
	3-factor α	0.10	-0.92	0.22	0.44	-0.20	0.14	1.23	0.23	0.24	0.50	0.40
		(0.13)	(-1.19)	(0.33)	(0.62)	(-0.41)	(0.31)	(2.54)	(0.60)	(0.58)	(1.42)	(0.45)
	4-factor α	0.16	-0.70	0.66	0.66	-0.48	0.06	0.92	-0.01	0.07	0.34	0.18
		(0.18)	(-1.07)	(0.89)	(0.90)	(-0.87)	(0.12)	(1.64)	(-0.02)	(0.18)	(1.00)	(0.19)
	Sharpe Ratio	-0.14	0.26	0.01	0.17	0.10	0.06	0.22	0.41	0.26	0.38	0.55
	Information Ratio	(0.04)	(-0.22)	(0.25)	(0.25)	(-0.17)	(0.03)	(0.46)	(-0.00)	(0.04)	(0.25)	(0.05)
Portugal	Excess Return	0.40	-0.09	0.35	1.07	0.11	0.78	1.53	0.76	0.61	1.40	1.00
		(0.57)	(-0.13)	(0.62)	(1.32)	(0.19)	(1.16)	(2.46)	(1.44)	(1.30)	(2.46)	(1.43)
	CAPM α	1.69	-0.42	0.36	1.59	0.16	1.45	1.02	0.86	0.39	2.60	0.91
		(1.30)	(-0.40)	(0.35)	(1.49)	(0.17)	(1.25)	(1.05)	(0.92)	(0.38)	(2.93)	(0.66)
	3-factor α	1.04	-9.34	-0.64	2.98	-0.49	12.82	3.40	22.08	6.13	0.58	-0.46

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
		(0.68)	(-1.27)	(-0.34)	(0.91)	(-0.07)	(1.48)	(0.61)	(0.91)	(1.42)	(0.30)	(-0.17)
	4-factor α	-1.05	0.08	46.63	-87.03	15.06	-16.29	0.13	-13.50	85.62	4.45	5.51
		(-0.62)	(0.00)	(1.26)	(-1.53)	(1.18)	(-1.27)	(0.01)	(-1.36)	(0.86)	(0.41)	(0.47)
	Sharpe Ratio	0.15	-0.04	0.17	0.42	0.06	0.37	0.74	0.42	0.34	0.63	0.33
	Information Ratio	(-0.15)	(0.00)	(3.24)	(-0.04)	(0.74)	(-0.13)	(0.00)	(-0.24)	(37.06)	(0.10)	(0.13)
Peru	Excess Return	1.44	1.35	1.33	1.59	1.84	1.96	1.39	1.63	1.88	1.84	0.40
		(2.85)	(2.67)	(2.54)	(3.69)	(4.28)	(3.54)	(2.49)	(4.45)	(4.48)	(2.51)	(0.58)
	CAPM α	0.52	1.46	1.75	1.68	1.64	1.48	1.38	1.65	1.50	2.16	1.64
		(1.49)	(3.09)	(2.49)	(4.60)	(3.55)	(2.81)	(3.23)	(4.13)	(4.20)	(4.80)	(3.19)
	3-factor α	0.32	1.05	0.64	0.95	1.44	1.08	1.11	1.27	0.79	2.26	1.94
		(0.64)	(2.18)	(1.69)	(1.83)	(2.34)	(3.39)	(2.69)	(2.93)	(1.43)	(5.18)	(3.34)
	4-factor α	0.13	0.88	0.45	0.95	1.22	0.90	0.98	1.15	0.70	2.55	2.42
		(0.24)	(1.85)	(1.16)	(2.01)	(2.28)	(2.71)	(2.56)	(2.52)	(1.21)	(4.17)	(3.54)
	Sharpe Ratio	0.93	0.95	0.93	1.35	1.32	1.43	1.06	1.25	1.44	0.95	0.18
	Information Ratio	(0.06)	(0.50)	(0.29)	(0.58)	(0.62)	(0.59)	(0.61)	(0.68)	(0.41)	(0.75)	(0.68)
Czech Republic	Excess Return	1.09	-0.88	0.93	0.54	0.91	0.59	0.57	1.38	1.06	1.01	-0.08
		(1.84)	(-1.72)	(1.71)	(1.05)	(2.96)	(1.63)	(1.32)	(3.37)	(1.86)	(1.80)	(-0.10)
	CAPM α	-0.19	-20.14	42.77	-81.77	57.81	-19.06	-80.13	12.42	13.36	2.36	2.55
		(-0.19)	(-0.58)	(0.99)	(-0.78)	(1.34)	(-0.57)	(-0.86)	(0.85)	(0.15)	(2.30)	(1.84)
	3-factor α	0.34	-0.15	0.94	0.97	-2.35	1.05	0.46	-0.01	0.77	2.57	2.23
		(0.54)	(-0.25)	(2.14)	(1.97)	(-1.03)	(1.85)	(1.53)	(-0.02)	(1.04)	(2.74)	(1.81)
	4-factor α	0.43	-0.15	0.67	1.07	0.30	0.61	0.39	0.37	0.78	1.97	1.53
		(1.01)	(-0.50)	(1.75)	(2.46)	(0.75)	(1.55)	(1.40)	(0.96)	(1.28)	(2.78)	(1.72)
	Sharpe Ratio	0.53	-0.38	0.42	0.21	0.62	0.42	0.26	0.92	0.55	0.60	-0.03
	Information Ratio	(0.30)	(-0.11)	(0.50)	(0.66)	(0.17)	(0.42)	(0.36)	(0.24)	(0.40)	(0.77)	(0.51)
New Zealand	Excess Return	0.67	0.44	0.26	0.29	0.68	0.77	0.49	0.67	0.72	1.36	0.69
		(1.20)	(1.13)	(0.71)	(0.76)	(1.88)	(2.07)	(1.33)	(1.96)	(1.96)	(4.34)	(1.26)
	CAPM α	0.74	0.80	-0.29	1.14	0.84	0.72	0.44	0.65	0.19	1.48	0.74
		(1.05)	(1.29)	(-0.42)	(2.59)	(1.60)	(1.03)	(0.55)	(1.08)	(0.34)	(2.84)	(0.87)

Country		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	H-L
	3-factor α	-0.28	0.40	-0.35	0.58	0.28	0.80	0.94	1.31	0.23	2.65	2.93
		(-0.36)	(0.65)	(-0.70)	(0.72)	(0.63)	(1.26)	(1.59)	(1.77)	(0.42)	(3.67)	(2.68)
	4-factor α	-0.14	-0.19	-0.21	0.91	0.30	0.80	0.57	0.62	0.14	1.74	1.88
		(-0.17)	(-0.30)	(-0.31)	(1.66)	(0.62)	(1.45)	(1.04)	(0.98)	(0.26)	(3.25)	(1.83)
	Sharpe Ratio	0.37	0.30	0.19	0.21	0.58	0.57	0.37	0.60	0.70	1.33	0.37
	Information Ratio	(-0.04)	(-0.06)	(-0.08)	(0.40)	(0.13)	(0.40)	(0.26)	(0.25)	(0.07)	(0.90)	(0.54)
Pakistan	Excess Return	1.22	1.11	1.50	1.02	1.61	1.49	1.54	1.64	1.88	1.46	0.24
		(1.63)	(1.60)	(2.40)	(1.51)	(2.32)	(2.04)	(2.13)	(2.49)	(2.93)	(2.36)	(0.46)
	CAPM α	1.90	1.03	1.87	1.67	1.59	1.31	2.63	1.80	2.77	1.78	-0.12
		(2.68)	(1.79)	(2.91)	(2.79)	(2.23)	(2.25)	(3.40)	(3.41)	(4.22)	(3.58)	(-0.19)
	3-factor α	2.04	0.84	1.94	1.55	1.97	0.39	2.95	0.75	2.63	1.29	-0.75
		(2.60)	(1.40)	(3.37)	(2.88)	(2.58)	(0.71)	(3.42)	(1.32)	(3.80)	(2.71)	(-0.89)
	4-factor α	2.13	0.56	1.49	1.17	1.64	0.70	3.49	0.22	2.65	0.90	-1.23
		(2.24)	(0.99)	(2.34)	(2.10)	(2.38)	(0.99)	(3.17)	(0.33)	(4.14)	(1.84)	(-1.17)
	Sharpe Ratio	0.55	0.56	0.82	0.54	0.82	0.81	0.80	0.88	1.10	0.86	0.14
	Information Ratio	(0.71)	(0.25)	(0.73)	(0.52)	(0.68)	(0.28)	(1.42)	(0.10)	(1.26)	(0.49)	(-0.28)

Appendix 3 – Python notebook with code used to perform analysis

Coding Notebook

Table of Contents

1. Introduction

The we want to use the Pipeline library, which contain functions that allow us to perform cross-sectional trailing-window computations. "Cross-sectional" because one value is computed for each asset, and "trailing-window" because data is fetched over a trailing window. An example would be: "Every day, fetch the last two days of closing prices for all assets. For each asset, calculate the percent change between the asset’s previous close price and its current close price."

The pipeline function does the following steps for us:

1. Imports data
2. Defines the computations to be done on the data
3. Instantiates the pipeline (i.e. creates an object we can use later)

We then run the pipeline on data between our start and end date and perform the defined computations.

2. Analysis Setup

2.1. Import Libraries

The first thing we do in a Python code is to import the libraries containing functions we will use.

In [5]:
```
# Define all the countries we will loop over:
countries = (
'AR_EQUITIES',
'AT_EQUITIES',
'AU_EQUITIES',
'BE_EQUITIES',
'BR_EQUITIES',
'CA_EQUITIES',
'CH_EQUITIES',
'CL_EQUITIES',
'CN_EQUITIES',
'CO_EQUITIES',
'CZ_EQUITIES',
'DE_EQUITIES',
'DK_EQUITIES',
'ES_EQUITIES',
'FI_EQUITIES',
'FR_EQUITIES',
'GB_EQUITIES',
'GR_EQUITIES',
'HK_EQUITIES',
'HU_EQUITIES',
'ID_EQUITIES',
'IE_EQUITIES',
'IN_EQUITIES',
'IT_EQUITIES',
'JP_EQUITIES',
'MY_EQUITIES',
'MX_EQUITIES',
'NL_EQUITIES',
'NZ_EQUITIES',
'NO_EQUITIES',
'PK_EQUITIES',
'PE_EQUITIES',
'PH_EQUITIES',
'PL_EQUITIES',
'PT_EQUITIES',
'RU_EQUITIES',
'SG_EQUITIES',
'ZA_EQUITIES',
'KR_EQUITIES',
'ES_EQUITIES',
'SE_EQUITIES',
'CH_EQUITIES',
'TW_EQUITIES',
'TH_EQUITIES',
'TR_EQUITIES',
'US_EQUITIES',
    )

domains = np.array([AR_EQUITIES, # Argentina
    AT_EQUITIES, # Austria
    AU_EQUITIES, # Australia
    BE_EQUITIES, # Belgium
    BR_EQUITIES, # Brazil
    CA_EQUITIES, # Canada
    CH_EQUITIES, # Switzerland
    CL_EQUITIES, # Chile
    CN_EQUITIES, # China
    CO_EQUITIES, # Colombia
    CZ_EQUITIES, # Czech Republic
    DE_EQUITIES, # Germany
    DK_EQUITIES, # Denmark
    ES_EQUITIES, # Spain
    FI_EQUITIES, # Finland
    FR_EQUITIES, # France
    GB_EQUITIES, # Great Britain
    GR_EQUITIES, # Greece
    HK_EQUITIES, # Hong Kong
    HU_EQUITIES, # Hungary
    ID_EQUITIES, # Indonesia
    IE_EQUITIES, # Ireland
    IN_EQUITIES, # India
    IT_EQUITIES, # Italy
    JP_EQUITIES, # Japan
    MY_EQUITIES, # Malaysia
    MX_EQUITIES, # Mexico
    NL_EQUITIES, # Netherlands
    NZ_EQUITIES, # New Zealand
    NO_EQUITIES, # Norway
    PK_EQUITIES, # Pakistan
    PE_EQUITIES, # Peru
    PH_EQUITIES, # Philippines
    PL_EQUITIES, # Poland
    PT_EQUITIES, # Portugal
    RU_EQUITIES, # Russia
    SG_EQUITIES, # Singapore
    ZA_EQUITIES, # South Africa
    KR_EQUITIES, # South Korea
    ES_EQUITIES, # Spain
    SE_EQUITIES, # Sweden
    CH_EQUITIES, # Switzerland
    TW_EQUITIES, # Taiwan
    TH_EQUITIES, # Thailand
    TR_EQUITIES, # Turkey
    US_EQUITIES])
```

In [6]:
```
# define tables we will collect data in
table4_0 = pd.DataFrame()
table4_1 = pd.DataFrame(columns=range(13))
table4_1.columns = ['country', '1','2','3','4','5','6','7','8','9','10','H-L','t-stat']
table4_2 = pd.DataFrame()
table4_3 = pd.DataFrame()
table4_4 = pd.DataFrame()
AppTable_1 = pd.DataFrame()
```

In [10]:
```
c=45
countries[c]
```

Out[10]:
```
'US_EQUITIES'
```

In [11]:
```
domains[c]
```

Out[11]:
```
EquityCalendarDomain('US', 'XNYS')
```

In [12]:
```
from quantopian.pipeline.domain import (
    AR_EQUITIES, # Argentina
    AT_EQUITIES, # Austria
    AU_EQUITIES, # Australia
    BE_EQUITIES, # Belgium
    BR_EQUITIES, # Brazil
    CA_EQUITIES, # Canada
    CH_EQUITIES, # Switzerland
    CL_EQUITIES, # Chile
    CN_EQUITIES, # China
    CO_EQUITIES, # Colombia
    CZ_EQUITIES, # Czech Republic
    DE_EQUITIES, # Germany
    DK_EQUITIES, # Denmark
    ES_EQUITIES, # Spain
    FI_EQUITIES, # Finland
    FR_EQUITIES, # France
    GB_EQUITIES, # Great Britain
    GR_EQUITIES, # Greece
    HK_EQUITIES, # Hong Kong
    HU_EQUITIES, # Hungary
    ID_EQUITIES, # Indonesia
    IE_EQUITIES, # Ireland
    IN_EQUITIES, # India
    IT_EQUITIES, # Italy
    JP_EQUITIES, # Japan
    MY_EQUITIES, # Malaysia
    MX_EQUITIES, # Mexico
    NL_EQUITIES, # Netherlands
    NZ_EQUITIES, # New Zealand
    NO_EQUITIES, # Norway
    PK_EQUITIES, # Pakistan
    PE_EQUITIES, # Peru
    PH_EQUITIES, # Philippines
    PL_EQUITIES, # Poland
    PT_EQUITIES, # Portugal
    RU_EQUITIES, # Russia
    SG_EQUITIES, # Singapore
    ZA_EQUITIES, # South Africa
    KR_EQUITIES, # South Korea
    ES_EQUITIES, # Spain
    SE_EQUITIES, # Sweden
    CH_EQUITIES, # Switzerland
    TW_EQUITIES, # Taiwan
    TH_EQUITIES, # Thailand
    TR_EQUITIES, # Turkey
    US_EQUITIES, # United States
)
```

In [13]:
```
from quantopian.pipeline import Pipeline
from quantopian.pipeline.data import EquityPricing, factset
from quantopian.pipeline.domain import DE_EQUITIES
from quantopian.research import run_pipeline
from quantopian.pipeline.factors import CustomFactor, Returns, SimpleBeta
from quantopian.pipeline.classifiers import Classifier
from quantopian.pipeline.data.factset import EquityMetadata, RBICSFocus
from quantopian.research import run_pipeline, returns
from quantopian.pipeline import Pipeline, CustomFactor
from quantopian.research.experimental import get_factor_returns, get_factor_loadings
from quantopian.pipeline.filters import QTradableStocksUS


import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

import empyrical as ep
import alphalens as al
import pyfolio as pf

import statsmodels.api as sm
from statsmodels import regression,stats

#save_list = dir()
```

In [ ]:

In [ ]:

2.2 Define Factors

2.2.1 Profitability

Before we start running the pipeline we will define the factors that we want the pipeline to calculate for all of our assets. We start of with the factors that define "Profitability"

$$GPOA = \frac{REVT-COGS}{AT}$$

$$ROE = \frac{IB}{BE}$$

$$ROA = \frac{IB}{AT}$$

$$CFOA = \frac{IB+DP-\Delta WC-CAPX}{AT}$$

$$GMAR = \frac{REVT-COGS}{SALE}$$

$$ACC = \frac{DP-\Delta WC}{AT}$$

$$Profitability = z(z_{GPOA} + z_{ROE} + z_{ROA} + z_{CFOA} + z_{GMAR} + z_{ACC})$$

In [14]:
```
def zscores(x):
    return (x - np.nanmean(x, axis=0)) / np.nanstd(x, axis=0)

# Gross profit over assets
class GPOA(CustomFactor):
    window_length = 1
    def compute(self, today, asset_ids, out, REVT, COGS, AT):
        GPOA = (REVT - COGS) / AT
        out[:] = GPOA

# cash flow over assets
class CFOA(CustomFactor):
    window_length=1
    def compute(self, today, asset_ids, out, IB, DP, dWC, CAPX, AT):
        CFOA = (IB + DP + dWC - CAPX) / AT
        out[:] = CFOA

# Gross margin
class GMAR(CustomFactor):
    window_length=1
    def compute(self, today, asset_ids, out, REVT, COGS, SALE):
        GMAR = (REVT.astype(np.float64) - COGS.astype(np.float64)) / SALE.astype(np.float64)
        out[:] = GMAR

# low accruals
class ACC(CustomFactor):
    window_length=1
    def compute(self, today, asset_ids, out, DP, dWC, AT):
        ACC = (DP - dWC)/AT
        out[:] = ACC
```

2.2.2 Growth

We now define the factors required to calculate the "Growth"-factor. Here, every acccounting measure is taken per share outstanding

$$\Delta GPOA = \frac{(REVT-COGS)_t-(REVT-COGS)_{t-5}}{AT_{t-5}}$$

$$\Delta ROE = \frac{IB_t-IB_{t-5}}{BE_{t-5}}$$

$$\Delta ROA = \frac{IB_t-IB_{t-5}}{AT_{t-5}}$$

$$\Delta CFOA = \frac{(IB+DP-\Delta WC-CAPX)_t-(IB+DP-\Delta WC-CAPX)_{t-5}}{AT_{t-5}}$$

$$\Delta GMAR = \frac{(REVT-COGS)_t-(REVT-COGS)_{t-5}}{SALE_{t-5}}$$

$$Growth = z(z_{\Delta GPOA} + z_{\Delta ROE} + z_{\Delta ROA} + z_{\Delta CFOA} + z_{\Delta GMAR} + z_{\Delta ACC})$$

In [15]:
```
#[0] refers to the first data in the window (5 years ago), [-1] refers to latest data
win_len = 252*5  # Five year window

# Change in Gross Profits over Assets
class dGPOA(CustomFactor):
    window_length = win_len      #5 year window
    window_safe = True
    def compute(self, today, asset_ids, out, REVT, COGS, AT):
        GPOA = ((REVT[-1] - REVT[0]) - (COGS[-1] - COGS[0]) )/ AT[0]
        out[:] = GPOA

# change in cash flow over assets
class dCFOA(CustomFactor):
    window_length = win_len
    window_safe = True
    def compute(self, today, asset_ids, out, IB, DP, dWC, CAPX, AT):
        CFOA = ((IB[-1] + DP[-1] + dWC[-1] - CAPX[-1]) - (IB[0] + DP[0] + dWC[0] - CAPX[0]) )/ AT[0]
        out[:] = CFOA

#Change in Gross Margin
class dGMAR(CustomFactor):
    window_length = win_len
    window_safe = True
    def compute(self, today, asset_ids, out, REVT, COGS, SALE):
        GMAR = ((REVT[-1] - COGS[-1]) - (REVT[0] - COGS[0]))/ SALE[0]
        out[:] = GMAR

# Change in ROE
class dROE(CustomFactor):
    window_length = win_len
    window_safe = True
    def compute(self, today, asset_ids, out, ROE):
        out[:] = ROE[-1] - ROE[0]

# Change in ROA
class dROA(CustomFactor):
    window_length = win_len
    window_safe = True
    def compute(self, today, asset_ids, out, ROA):
        out[:] = ROA[-1] - ROA[0]
```

2.2.3 Safety

The safety factor is computed by averaging z-scores of low beta (BAB), low leverage (LEV) and low bankruptcy risk (Ohlson's O and Altman's Z) and low earnings volatility (EVOL). First we define the required customfactors, which are: $BAB = -\beta$

$$LEV = -\frac{(DLTT+DLC+MIBT+PSTK)}{AT} = -\frac{Debt}{Assets}$$

$$O = "\ Ohlson's O - score\ "$$

$$Z = "\ AltmansZ\ "$$

$$EVOL = \sigma(ROE)_{5year}$$

Functions used to create other factors

In [16]:
```
def vectorized_beta(dependents, independent, allowed_missing, out=None):
    """
    Compute slopes of linear regressions between columns of ``dependents`` and
    ``independent``.
    Parameters
    ----------
    dependents : np.array[N, M]
        Array with columns of data to be regressed against ``independent``.
    independent : np.array[N, 1]
        Independent variable of the regression
    allowed_missing : int
        Number of allowed missing (NaN) observations per column. Columns with
        more than this many non-nan observations in either ``dependents`` or
        ``independents`` will output NaN as the regression coefficient.
    out : np.array[M] or None, optional
        Output array into which to write results.  If None, a new array is
        created and returned.
    Returns
    -------
    slopes : np.array[M]
        Linear regression coefficients for each column of ``dependents``.
    """
    # Cache these as locals since we're going to call them multiple times.
    nan = np.nan
    isnan = np.isnan
    N, M = dependents.shape

    if out is None:
        out = np.full(M, nan)

    # Copy N times as a column vector and fill with nans to have the same
    # missing value pattern as the dependent variable.
    #
    # PERF_TODO: We could probably avoid the space blowup by doing this in
    # Cython.

    # shape: (N, M)
    independent = np.where(
        isnan(dependents),
        nan,
        independent,
    )

    # Calculate beta as Cov(X, Y) / Cov(X, X).
    # https://en.wikipedia.org/wiki/Simple_linear_regression#Fitting_the_regression_line  # noqa
    #
    # NOTE: The usual formula for covariance is::
    #
    #    mean((X - mean(X)) * (Y - mean(Y)))
    #
    # However, we don't actually need to take the mean of both sides of the
    # product, because of the folllowing equivalence::
    #
    # Let X_res = (X - mean(X)).
    # We have:
    #
    #     mean(X_res * (Y - mean(Y))) = mean(X_res * (Y - mean(Y)))
    #                             (1) = mean((X_res * Y) - (X_res * mean(Y)))
    #                             (2) = mean(X_res * Y) - mean(X_res * mean(Y))
    #                             (3) = mean(X_res * Y) - mean(X_res) * mean(Y)
    #                             (4) = mean(X_res * Y) - 0 * mean(Y)
    #                             (5) = mean(X_res * Y)
    #
    #
    # The tricky step in the above derivation is step (4). We know that
    # mean(X_res) is zero because, for any X:
    #
    #     mean(X - mean(X)) = mean(X) - mean(X) = 0.
    #
    # The upshot of this is that we only have to center one of `independent`
    # and `dependent` when calculating covariances. Since we need the centered
    # `independent` to calculate its variance in the next step, we choose to
    # center `independent`.

    # shape: (N, M)
    ind_residual = independent - np.nanmean(independent, axis=0)

    # shape: (M,)
    covariances = np.nanmean(ind_residual * dependents, axis=0)

    # We end up with different variances in each column here because each
    # column may have a different subset of the data dropped due to missing
    # data in the corresponding dependent column.
    # shape: (M,)
    independent_variances = np.nanmean(ind_residual ** 2, axis=0)

    # shape: (M,)
    np.divide(covariances, independent_variances, out=out)

    # Write nans back to locations where we have more then allowed number of
    # missing entries.
    nanlocs = np.isnan(independent).sum(axis=0) > allowed_missing
    out[nanlocs] = np.nan

    return out
```

def vectorized_beta(spy, assets): """Calculate beta between every column of ``assets`` and ``spy``. Parameters ---------- spy : np.array An (n x 1) array of returns for SPY. assets : np.array An (n x m) array of returns for m assets. """ assert len(spy.shape) == 2 and spy.shape[1] == 1, "Expected a column vector for spy." asset_residuals = assets - assets.mean(axis=0) spy_residuals = spy - spy.mean() covariances = (asset_residuals * spy_residuals).sum(axis=0) spy_variance = (spy_residuals ** 2).sum() return covariances / spy_variance

In [17]:
```
# this factor creates a value-weighted fraction for each asset in the universe
# and will be used when calculating the BAB factor
class cap_weight(CustomFactor):
    window_length=2
    inputs = [factset.Fundamentals.mkt_val]

    def compute(self,today, asset_ids, out, mkt_value):
        mkt_sum = np.nansum(mkt_value, axis=1)[:,None]
        mkt_value = mkt_value
        out[:] = (mkt_value.T/mkt_sum.T)


# Leverage factor
class LEV(CustomFactor):
    window_length = 1

    def compute(self, today, asset_ids, out, DEBT, MIBT, PSTK, AT):
        out[:] = -(DEBT + MIBT + PSTK) / AT

# Low earnings volatility
class EVOL(CustomFactor):
    window_length = 252*5
    window_safe = True

    def compute(self, today, asset_ids, out, ROE):
        out[:] = np.std(ROE, axis=0)


class BAB(CustomFactor):
    window_length=21
    window_safe=True

    def compute(self,today,asset_ids,out, returns):
        mkt_returns = returns.mean(axis=1)[:, np.newaxis]
        #BAB = vectorized_beta(mkt_returns, returns.values)
        out[:] = vectorized_beta(returns, mkt_returns, 10)
```

2.2.4 Payout

This is an alternative additional factor which can be included in the quality factor. It consist of the following sub-factors:

$$EISS = -log\frac{SHROUTadj_t}{SHROUTadj_{t-1}}$$

$$DISS = -log\frac{TOTD_t}{TOTD_{t-1}}$$

2.3 Data Collection

2.3.1 Define data sets to be used

In [18]:
```
# Define some fundamental data that we will use in our factors
# Profitability
REVT = factset.Fundamentals.gross_inc_af.latest   # gross income /ltm
COGS = factset.Fundamentals.cogs_af.latest        # cost of goods sold _ltm
ROE = factset.Fundamentals.roe_af.latest           # return on equity
ROA = factset.Fundamentals.roa_af.latest           # return on assets
SALE = factset.Fundamentals.sales_af.latest        # total sales
AT = factset.Fundamentals.assets.latest            # total assets
IB = factset.Fundamentals.net_inc_af.latest       # net income /_ltm
DP = factset.Fundamentals.dep_exp_af.latest       # depreciation /_ltm
dWC = factset.Fundamentals.wkcap_chg_af.latest    # change in working capital /_ltm
CAPX = factset.Fundamentals.capex_af.latest       # capital expenditure /_ltm

# Same fundamental data defined so that a lookback function can be used to calculate growth over time
REVTa = factset.Fundamentals.gross_inc_af
COGSa = factset.Fundamentals.cogs_af
ATa = factset.Fundamentals.assets
ROEa = factset.Fundamentals.roe_af
ROAa = factset.Fundamentals.roa_af
SALEa = factset.Fundamentals.sales_af
ATa = factset.Fundamentals.assets
IBa = factset.Fundamentals.net_inc_af
DPa = factset.Fundamentals.dep_exp_af
dWCa = factset.Fundamentals.wkcap_chg_af
CAPXa = factset.Fundamentals.capex_af

# Safety
ALT_Z = factset.Fundamentals.zscore_af.latest            # Altman's Z-score
DEBT = factset.Fundamentals.debt.latest                  # Total debt
MIBT = factset.Fundamentals.min_int_accum.latest         # Total accumulated minority interest
PSTK = factset.Fundamentals.pfd_stk.latest               # Preferred stock

#Others
MKT_VAL = factset.Fundamentals.mkt_val.fx('USD').latest          # Market capitalization
PB = factset.Fundamentals.pbk_af.latest                # Price to book ratio
Sector = RBICSFocus.l1_name.latest                    # sector classification code based business focus
BE = factset.Fundamentals.shldrs_eq.latest             # Book value defined as shareholders equity
```

2.3.2 Run the Main Data Pipeline

2.3.2.1 Factors used in Carhart 4-factor model

This code defines what we need for the Carhart (Fama-French + momentum) 4 factor model

In [19]:
```
class Momentum(CustomFactor):
    # will give us the price from last month minus last year divided by last year
    inputs = [EquityPricing.close]
    window_length = 252

    def compute(self, today, assets, out, prices):
        out[:] = (prices[-21] - prices[-252])/prices[-252]

# market cap and book-to-price data gets fed in here
# MKT_VAL already defined
# Note here that we do not use Book-to-Market as Fama, but Book-to-Price due to availability of data
BE_ME = BE/MKT_VAL
# and momentum as lagged returns
momentum = Momentum()

# Create a pipeline filter for 'tradable' stocks.
is_tradable = (EquityMetadata.security_type.latest.eq('SHARE') & EquityMetadata.is_primary.latest)

# Grab the the factors based on Fama/French + Carhart(momentum) breakpoints
big = MKT_VAL.percentile_between(80, 100, mask=is_tradable)
small = MKT_VAL.percentile_between(0, 50, mask=is_tradable)

value = BE_ME.percentile_between(70, 100, mask=is_tradable)
neutral = BE_ME.percentile_between(30, 70, mask=is_tradable)
growth = BE_ME.percentile_between(0, 30, mask=is_tradable)

highmom = momentum.percentile_between(70, 100, mask=is_tradable)
lowmom = momentum.percentile_between(0, 30, mask=is_tradable)

sv = BE_ME.percentile_between(70, 100, mask=small)
sn = BE_ME.percentile_between(30, 70, mask=small)
sg = BE_ME.percentile_between(0, 30, mask=small)
bv = BE_ME.percentile_between(70, 100, mask=big)
bn = BE_ME.percentile_between(30, 70, mask=big)
bg = BE_ME.percentile_between(0, 30, mask=big)
```

2.3.2.2 Factors used in Quality and running pipeline

In [20]:
```
# Start and End time of our analysis
start_date = pd.Timestamp("2007-06-06")  #2004-06-06
end_date = pd.Timestamp("2018-06-06")


# Instantiate the Profitability factors
GPOA_ = GPOA(inputs=[REVT, COGS, AT])
ROE_ = ROE
ROA_ = ROA
CFOA_ = CFOA(inputs=[IB, DP, dWC, CAPX, AT])
GMAR_ = GMAR(inputs=[REVT, COGS, SALE])
ACC_ = ACC(inputs=[DP, dWC, AT])

# Growth factors
dGPOA_  = dGPOA(inputs=[REVTa, COGSa, ATa])
dROE_ = dROE(inputs=[ROEa])
dROA_ = dROA(inputs=[ROAa])
dCFOA_ = CFOA(inputs=[IBa, DPa, dWCa, CAPXa, ATa])
dGMAR_ = GMAR(inputs=[REVTa, COGSa, SALEa])

# Safety Factors
#VW = value_weights(inputs=[MKT_VAL])
ALT_Z_ = ALT_Z
LEV_ = LEV(inputs=[DEBT, MIBT, PSTK, AT])
EVOL_ = EVOL(inputs=[ROEa])
BAB_ = BAB(inputs=[Returns(window_length=21)])



def make_pipeline():
    #insert factors

    yesterday_close = EquityPricing.close.latest
# Note taht we are reporting MONTHLY returns each day
    returns_d = Returns(window_length=2)
    returns_m = Returns(window_length=21)
    returns_a = Returns(window_length=252)
    mkt_val = factset.Fundamentals.mkt_val.fx('USD').latest

    return Pipeline(
    columns={
        # basics
        'MKT_VAL':MKT_VAL,
        'Price':yesterday_close,
        'Returns_m':returns_m,
        'Returns_d':returns_d,
        'Returns_a':returns_a,
        'Sector':Sector,
        # profitability
        'ACC':ACC_,
        'GPOA':GPOA_,
        'ROE':ROE_,
        'ROA':ROA_,
        'CFOA':CFOA_,
        'GMAR':GMAR_,
        # growth
        'dROA':dROA_,
        'dGPOA':dGPOA_,
        'dCFOA':dCFOA_,
        'dROE':dROE_,
        'dGMAR':dGMAR_,
        # safety
        'AltZ':ALT_Z_,
        'EVOL':EVOL_,
        'LEV':LEV_,
        'BAB': BAB_,    #-(SimpleBeta(target=symbols('SPY'), regression_length=252*5)),
        # Carhart 4-factors
        'sv':sv,
        'sn':sn,
        'sg':sg,
        'bv':bv,
        'bn':bn,
        'bg':bg,
        'highmom':highmom,
        'lowmom':lowmom,
        'ME_BE':1/BE_ME,
        'small':small,
        'big':big
    },
    domain = domains[c],
    screen = is_tradable         #  QTradableStocksUS()
    )

pipe = make_pipeline()

df = run_pipeline(pipe, start_date, end_date)

print "There are %d assets in this universe." % len(df.index.levels[1])
```

```
/venvs/py35/lib/python3.5/site-packages/numpy/lib/nanfunctions.py:703: RuntimeWarning: Mean of empty slice
  warnings.warn("Mean of empty slice", RuntimeWarning)
```

Pipeline Execution Time: 23 Minutes, 20.47 Seconds

```
There are 7621 assets in this universe.
```

In [ ]:

The next piece of code check the fraction of NaN entries per column

In [21]:
```
df = df.assign(BAB = 0)
```

In [22]:
```
columns = list(df)
a = pd.DataFrame(index=[0], columns=columns)
for x in columns:
    i= float(len(df[x])-df[x].count())
    j = float(len(df[x]))
    a[x]=i/j

a
```

Out[22]:

```
        ACC      AltZ  BAB      CFOA      EVOL     GMAR      GPOA       LEV     ME_BE  MKT_VAL ...    dGMAR     dGPOA      dROA      dROE highmom lowmom   sg small   sn   sv
0  0.187574  0.269086  0.0  0.187599  0.196017  0.18428  0.179738  0.014109  0.015571   0.0153 ...  0.18428  0.316026  0.187003  0.196017     0.0    0.0  0.0   0.0  0.0  0.0
```

1 rows × 32 columns

In [23]:
```
# Use this snippet do display dataframe as a grid

#import qgrid
#qgrid.show_grid(df)
```

To save computation time we resample the data into monthly frequency

In [24]:
```
# Convert our data into monthly instead of daily data

days = df.index.levels[0].unique()
d21s = days.values[::21][:-1]
d252 = days.values[::252][:-1]   # yearly dates

df = df.loc[list(d21s)]
#df_1m = df.resample('1M').first()

# code snippet to mask dataframe using monthly data
#mkt_bear_mask_d21 = mkt_bear_mask[mkt_bear_mask.index.get_level_values(0).isin(d21s)]
#df_m.shape
```

2.4 Data Cleansing

On the treatment of NaN's

Financial data are notoriously subject to outliers (extreme data points). In many statistical analyses, such data points may exert an undue influence on the results, making the results unreliable. Thus, if these outliers are not adjusted or accounted for, it is possible that they may lead to a failure to detect a phenomenon that does exist (a type II error), or even worse, results that indicate a phenomenon where no such phenomenon is actually present (a type I error). While there are several statistical methods that are designed to assess the effect of outliers or ameliorate their effect on results, empirical asset pricing researchers usually take a more ad hoc approach to dealing with the effect of outliers. There are two techniques that are commonly used in empirical asset pricing research to deal with the effect of outliers. The first technique, known as winsorization, simply sets the values of a given variable that are above or below a certain cutoff to that cutoff. The second technique, known as truncation, simply takes values of a given variable that are deemed extreme to be missing. Because we analyse data both with respect to time and cross-section we apply the winsorization at each time period.

In the event (quite likely for fundamental data) that there is an outlier, say outlier=70000 (unit), and the true mean is 1(unit)..... "a = np.nan_to_num((a-np.nanmean(a))) " might result in np.nanmean(a) =say 25 (unit) as opposed to 1(unit, being the true mean), then "a-25" pushes most of the values to say -24 (unit). This is fine so far (as we will scale things again at the end), however, we then replace NaN with 0 by method of np.nan_to_num, so the NaN names become good alpha names (right of the distribution). These NaN names are more likely than not going to stay as good alpha numbers in the remaining of the preprocessing, despite the subsequent winzorization and final scaling.

Ideally, the winsorization should be done before the first normalization, so that the first normalised numbers are more likely centred around zero, then it is fine to assign zero to NaN names. But, since scipy's winsorize method doesn't work well with NaNs in the array, the current method of replacing NaN with zero is right, in a sense that it ensures the next step of "winsorizing" will work.

The ideal way to do the whole processing is to ignore NaN, then winsorize, then normalize, then throw the NaN back into the distribution as zero.

Pseudo code:

1) Extract columns with quality factors

2) Replace +/-Inf values with NaN

3) Replace NaN values with mean value for the given date

4) Apply 95% winsorization on each factor for each date

5) Z-score normalize each factor for each day

Sector Neutralizing

Before presenting the results, we should step back and discuss how to deal with sectors and industries. When working with accounting data, there could be large differences between various accounting measures across different sectors. For example, profit margin, a quality factor, may differ considerably among financial, utility, technology, and consumer discretionary stocks. The same applies to other fundamental factors: dividend yield, profit margins, patents, change in employees, R&D normalized by assets or sales, employee utilization, accruals, stock option expensing (and many others) all vary widely across sectors. When selecting stocks, it may be more relevant to compare companies with their peers.

There are numerous ways of dealing with sector differences:

1. Demean the factors by sector
2. Standardize the factors by z-scoring within sectors
3. Forcing strict sector neutrality (for example, computing quantiles within each sector)
4. Eliminate certain sectors (financial stocks and utilities, for example)
5. Selectively eliminate sectors based on performance
6. All of the above but for industry groups rather than sectors

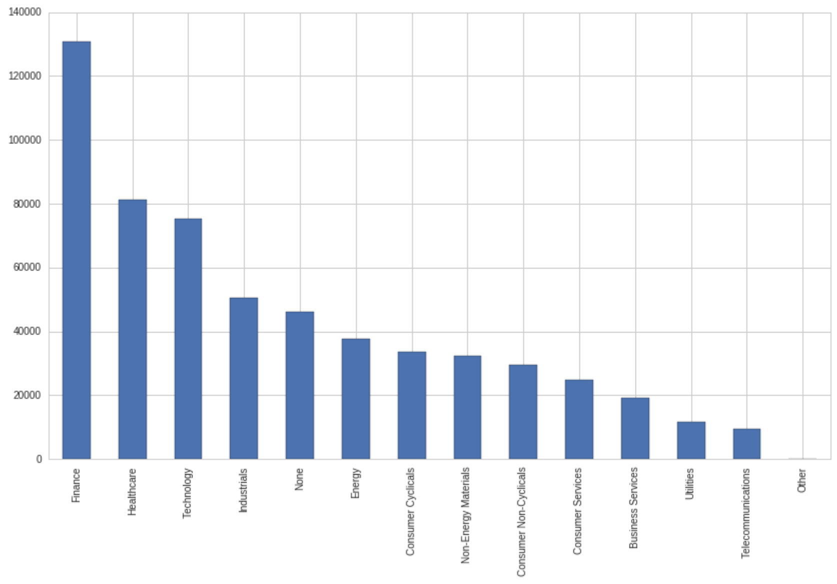
In this thesis I use method #2, grouping each sector together and normalizing values within each sector

Most academic papers demean by industries rather than sectors. Morningstar has 69 industry groups, larger than the 24 GICS industry groups or the 47 Fama-French Industries. The plot below shows the industry sector distribution in our dataset.

In [25]:
```
df['Sector'].value_counts(dropna=False).plot(kind='bar')
```

```
/venvs/py35/lib/python3.5/site-packages/pandas/indexes/category.py:121: FutureWarning:
Setting NaNs in `categories` is deprecated and will be removed in a future version of pandas.
  data = data.set_categories(categories)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f63d401cda0>
```



In [26]:
```
# remove entries without ME_BE values, they can not be regressed on

df2=df[['ACC', 'GPOA','dGPOA', 'ROE', 'dROE', 'ROA', 'dROA', 'CFOA', 'dCFOA', \
       'GMAR', 'dGMAR', 'AltZ', 'EVOL', 'LEV','BAB']]
columns = list(df2)
from scipy import stats

# In this for loop we replace NaN values with the column mean value
for x in columns:
    df2[x] = df2[x].replace([np.inf, -np.inf], np.nan)
    fd = df2[x].unstack().fillna(df2[x].unstack().mean(axis=1)[0])
    df2[x] = fd.stack()

# Use clip to apply a 95% winsorization window, i.e. outliers are replace by 2.5th or 97.5th percentile value
for x in columns:
    df2[x] = df2[x].clip(df2[x].quantile(0.025), df2[x].quantile(0.975))

# Use zscore to create column-wise z-scored values of our factors
df2['Sector']=df['Sector']

# We now perform z-scoring of each factor grouped by sector, i.e. z-scoring within each sector
#df2 = df2.groupby("Sector").transform(lambda x: stats.zscore(x))

#################################################################################
#     TO BE CHECKED
##################################################################################

# Optional method using ranked factors
# Before calculating z-score the factor is transformed to it's rank within the group in line with
# Asness et al. methodology. This makes the winsorization un-necessary, but you loose data granularity
df2 = df2.groupby("Sector").transform(lambda x: stats.zscore(x.rank()))

#################################################################################
#     /TO BE CHECKED
##################################################################################

# The previous operation removed the "sector" column so we re-introduce it
df2['Sector']=df['Sector']
#df2.dropna(subset=['Sector'], inplace=True)

# Add market/book ratio and remove equities with NaN and Inf ratios (typically also without market cap data)
# and value-weights (VW) of each asset
df2['ME_BE'] = df['ME_BE']
df2['ME_BE'].replace([np.inf, -np.inf], np.nan, inplace=True)
df2.dropna(subset=['ME_BE'], inplace=True)
df2['ME_BE'] = df2['ME_BE'].clip(df2['ME_BE'].quantile(0.005), df2['ME_BE'].quantile(0.995))
# Remove 0.5% outliers in each end of the data set (99% winsorized)
columns.append('ME_BE')
```

```
/venvs/py35/lib/python3.5/site-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  if __name__ == '__main__':
/venvs/py35/lib/python3.5/site-packages/ipykernel_launcher.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  # This is added back by InteractiveShellApp.init_path()
/venvs/py35/lib/python3.5/site-packages/ipykernel_launcher.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  from ipykernel import kernelapp as app
/venvs/py35/lib/python3.5/site-packages/ipykernel_launcher.py:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
/venvs/py35/lib/python3.5/site-packages/pandas/core/groupby.py:2193: FutureWarning:
Setting NaNs in `categories` is deprecated and will be removed in a future version of pandas.
  ordered=self.grouper.ordered))
```

In [27]:
```
# Calculate the fraction of non-NaN data for the new dataframe
a = pd.DataFrame(index=[0], columns=columns)
for x in columns:
    i= float(df2[x].count())
    j = float(len(df2[x]))
    a[x]=i/j

a
```

Out[27]:

	ACC	GPOA	dGPOA	ROE	dROE	ROA	dROA	CFOA	dCFOA	GMAR	dGMAR	AltZ	EVOL	LEV	BAB	ME_BE
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0

In [ ]:

2.5 Combining variables into our main factors

Below we create the factors for profitability, growth and safety and combine them into the single quality score

In [28]:
```
from scipy import stats
prof_col = ['GPOA', 'ROE', 'ROA', 'CFOA', 'GMAR', 'ACC']
# 1-Sum the colums, 2-Group dataframe by each date, 3-Zscore factor across assets for each date
Profitability = df2[prof_col].sum(axis=1).to_frame().groupby(level=0).transform(stats.zscore)
Profitability.columns = ['Profitability']

growth_col = ['dGPOA', 'dROE', 'dROA', 'dCFOA', 'dGMAR']
Growth = df2[growth_col].sum(axis=1).to_frame().groupby(level=0).transform(stats.zscore)
Growth.columns = ['Growth']

safety_col = ['BAB', 'LEV', 'AltZ', 'EVOL']
Safety = df2[safety_col].sum(axis=1).to_frame().groupby(level=0).transform(stats.zscore)
Safety.columns = ['Safety']

Quality = Profitability.join(Growth.join(Safety)).sum(axis=1).to_frame().groupby(level=0).transform(stats.zscore)
Quality.columns = ['Quality']

# add the new factors to our dataframe
df2['Quality']=Quality
df2['Profitability']=Profitability
df2['Safety']=Safety
df2['Growth']=Growth
```

In [29]:
```
#fx('GBP')

# We create some summary data which we will add to the thesis

tot_assets = len(df.index.levels[1])
used_assets = len(df2.index.levels[1])
assets_per_period = df2['Quality'].groupby(level=0).count().mean()
avg_mkt_cap = df2.join(df['MKT_VAL'])['MKT_VAL'].mean()
```

In [30]:
```
table4_0 = table4_0.append(pd.DataFrame({'Country': countries[c] ,
              'Total Stokcs': [tot_assets],
              'Used Stocks': [used_assets],
              'Stocks per month': [assets_per_period],
              'Mean Market Cap': [avg_mkt_cap]
             }))
```

In [31]:
```
table4_0
```

Out[31]:

	Country	Mean Market Cap	Stocks per month	Total Stokcs	Used Stocks
0	US_EQUITIES	4.455770e+09	4372.687023	7621	7621

3. Analysis

3.1 Overview

The analysis section will seek to understand the Quality factor in our dataset. This is done by answering the following questions:

1. What is the price of quality?
2. How has the price of quality changed over time?
3. What is the excess return of quality?

In [ ]:

3.1.1 Summary Statistics

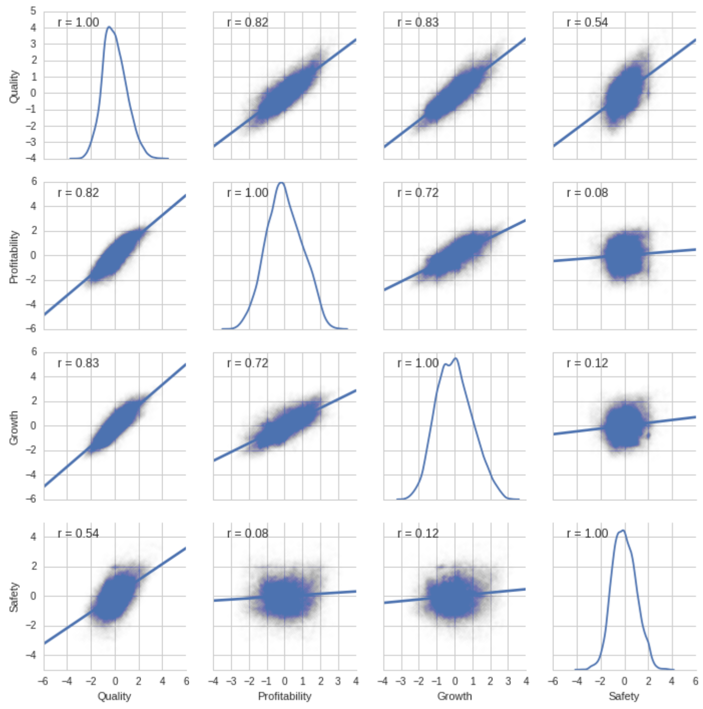
Pair-plots

The following figures show the distributions of each of the quality components, grouped by sector, and also the scatter plots of their relation to each other. The data points are for each stock resampled to annual data.

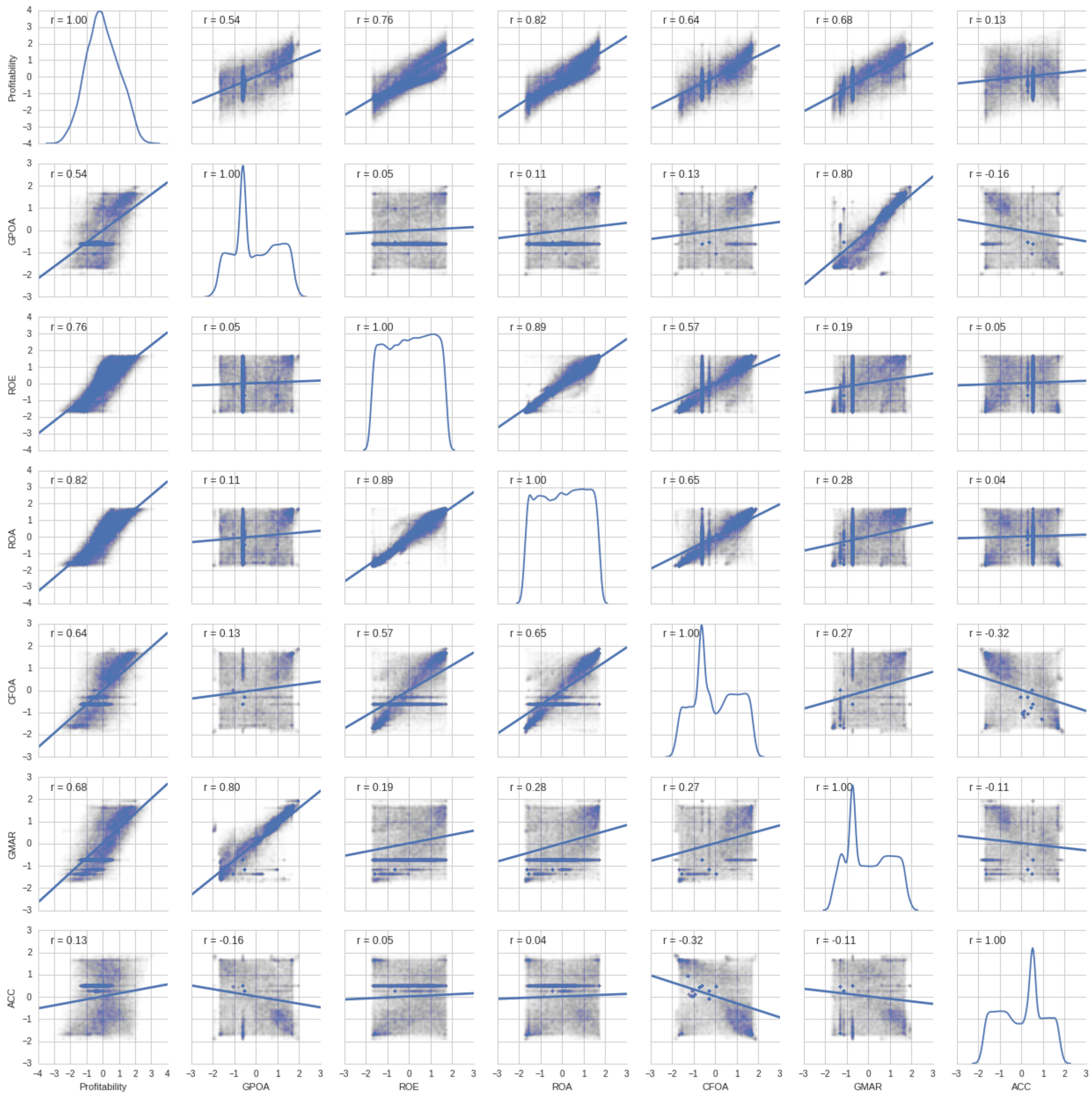
In [32]:
```
from scipy.stats import pearsonr

def corrfunc(x, y, **kws):
    (r, p) = pearsonr(x, y)
    ax = plt.gca()
    ax.annotate("r = {:.2f} ".format(r),
                xy=(.1, .9), xycoords=ax.transAxes)
```

```
In [33]: plt.figure(dpi=600)
         g1 = sns.pairplot(df2[df2.index.get_level_values(0).isin(d252)], diag_kind='kde', \
                      kind='reg', plot_kws = dict(ci=None, scatter_kws=dict(alpha=0.002)),\
                      vars=['Quality' ,'Profitability', 'Growth', 'Safety'])
         g1.map(corrfunc)
         plt.show()
```

```
/venvs/py35/lib/python3.5/site-packages/statsmodels/nonparametric/kdetools.py:20: VisibleDeprecationWarning: using a non-integer number instead of an integer will resul
t in an error in the future
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j

<matplotlib.figure.Figure at 0x7f63d62d94a8>
```

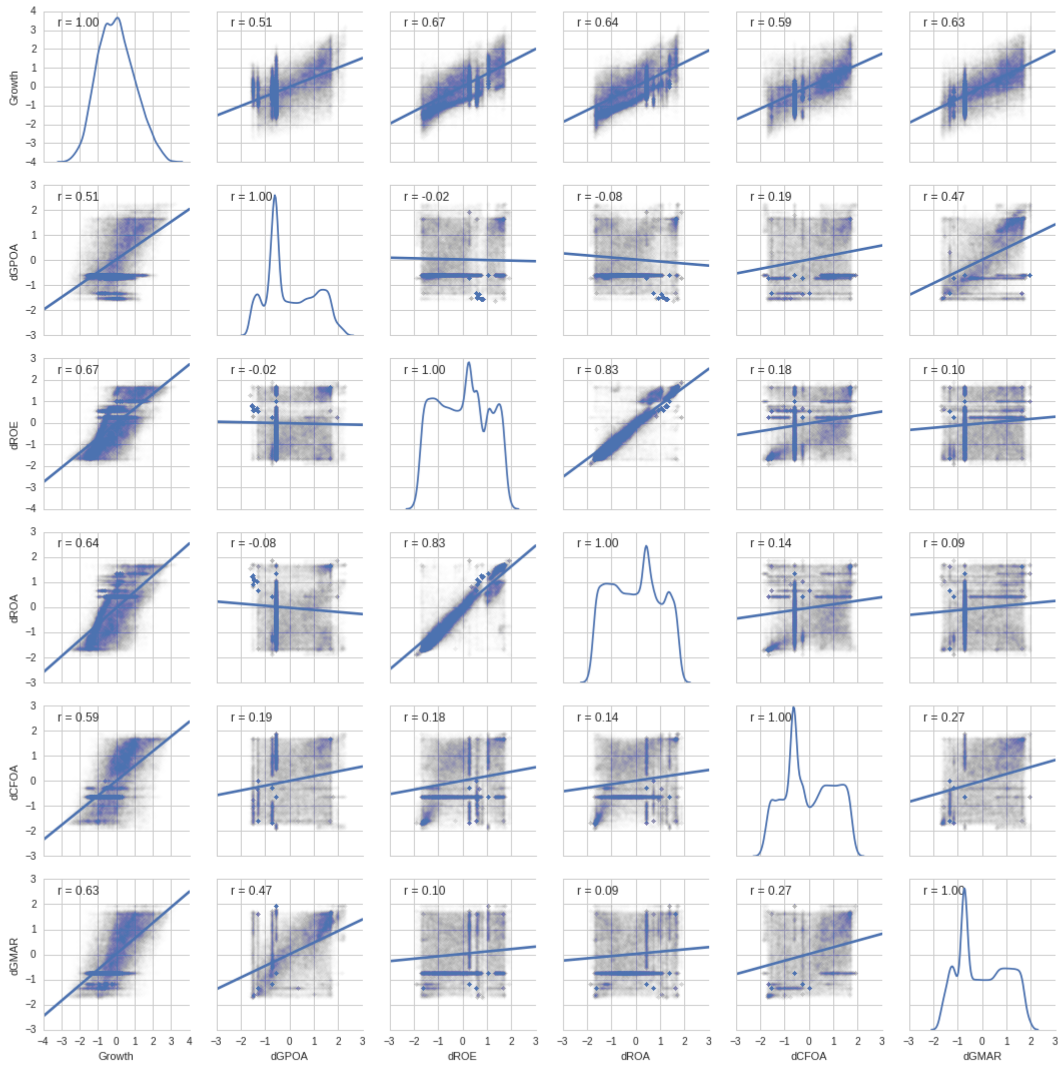


```
In [34]: plt.figure(dpi=600)
         g2=sns.pairplot(df2[df2.index.get_level_values(0).isin(d252)],  \
                     diag_kind='kde', kind='reg', plot_kws = dict(ci=None, scatter_kws=dict(alpha=0.005)),\
                     vars=['Profitability' ,'GPOA', 'ROE', 'ROA', 'CFOA', 'GMAR', 'ACC'])
         g2.map(corrfunc)
         plt.show()

<matplotlib.figure.Figure at 0x7f63b3a520b8>
```

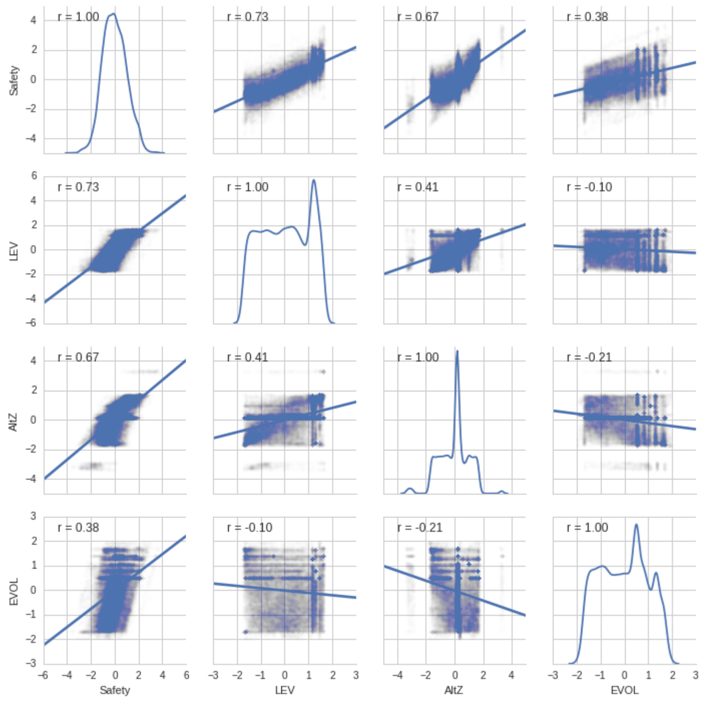


```
In [35]: plt.figure(dpi=600)
         g3 = sns.pairplot(df2[df2.index.get_level_values(0).isin(d252)],  \
                     diag_kind='kde', kind='reg', plot_kws = dict(ci=None, scatter_kws=dict(alpha=0.005)),\
                     vars=['Growth', 'dGPOA', 'dROE', 'dROA', 'dCFOA', 'dGMAR'])

         g3.map(corrfunc)
         plt.show()
```

```
<matplotlib.figure.Figure at 0x7f63b245a320>
```



```
In [36]: plt.figure(dpi=600)
         g4 = sns.pairplot(df2[df2.index.get_level_values(0).isin(d252)],  \
                     diag_kind='kde', kind='reg', plot_kws = dict(ci=None, scatter_kws=dict(alpha=0.002)),\
                     vars=['Safety' , 'LEV', 'AltZ', 'EVOL'])
         g4.map(corrfunc)
         plt.show()

         <matplotlib.figure.Figure at 0x7f639fa40d68>
```

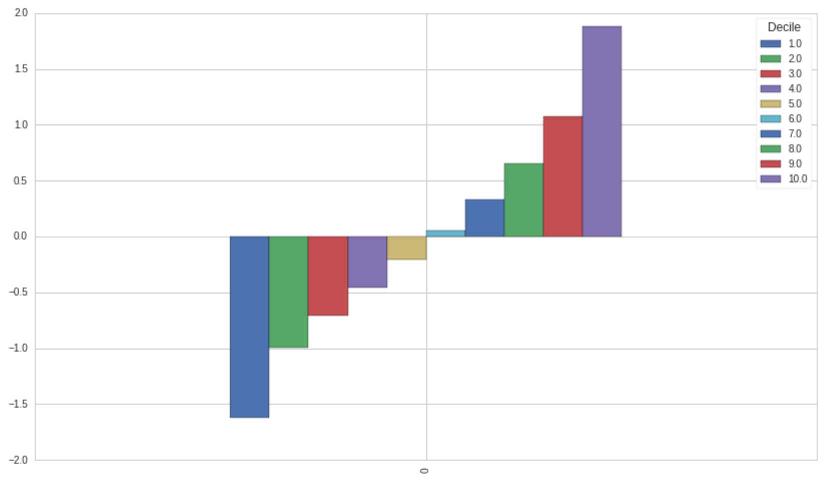


```
In [37]: # create factor correlation matrix and store to results

         apptable = df2[['Quality', 'Profitability', 'Growth', 'Safety']].corr(method='pearson')
         apptable['Country'] = countries[c]

         AppTable_1 = AppTable_1.append(apptable)
```

3.1.1 Creating 10 portfolios ranked on quality

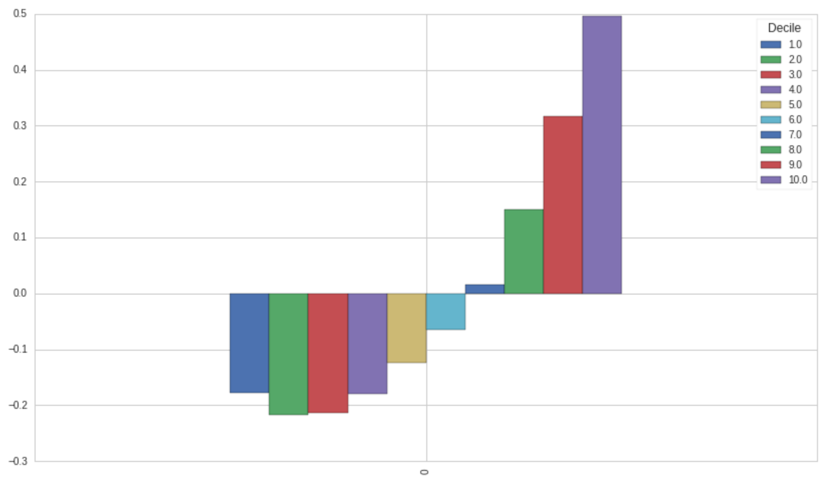
First we create a new column 'Decile' where we first rank stocks on quality for each day (level=0) and then we transform this rank into a quantile from 1-10. That means each portfolio 1-10 has an equal amount of assets, but the bin size/spacing of each portfolio varies from decile to decile.

```
In [ ]:

In [ ]:

In [38]: num_deciles = 10

         df2['Decile'] = df2.groupby(level=0)['Quality'].rank(method='first')
         df2['Decile'] = df2.groupby(level=0)['Decile'].transform(pd.qcut,num_deciles,labels=False)+1

         # Create a column with Market Cap weight-values

         df2['VW'] = df2.join(df['MKT_VAL'])['MKT_VAL'].groupby(level=0).transform(lambda x: (x/x.sum()))

In [39]: # Create a value-weighted quality column

         df2['Quality_w'] = (df2['Quality']*df2['VW']).to_frame().groupby(level=0).transform(stats.zscore)

In [40]: # Plot the portfolio mean values of Quality for each decile

         df2.groupby([df2.index.get_level_values(0), 'Decile'])['Quality'].mean().unstack().mean().to_frame().T.plot(kind='bar')

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x7f639fda42e8>
```



In [41]:
```
df2.groupby([df2.index.get_level_values(0), 'Decile'])['Quality_w'].mean().unstack().mean().to_frame().T.plot(kind='bar')
```

Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f639f985710>
```



Now, we make a table showing the persistence of Quality.

1. For each time step we collect the mean Quality for each decile at time t and at time t+time_lag
2. We iterate through our dataset and calculate the mean value across time for time t and t+time_lag. This is stored in the dataframe 'df_pers'. The values are calculated using the portfolio of stocks in each decile at time t.
3. We take the average value of each decile and return in the table. Rows 0-9 represent mean quality for deciles 1-10 at time t and rows 10-19 are the same for time t+time_lag. This is used to check persistence of quality.

In [42]:
```
#asset = df2.index.levels[1].unique()
#time = df2.index.levels[0].unique()
df_pers = pd.DataFrame(index=range(len(d21s)),columns=range(20))
time_lag = 36 # months to compare for persistence


for t in range(len(d21s)-time_lag):
    for x in range(num_deciles):
        df_pers.iloc[t,x] = ((df2.xs(d21s[t], level=0)['Decile'] == x+1)* \
                             (df2.xs(d21s[t], level=0)['Quality'])).replace(0, np.nan).mean()
        df_pers.iloc[t,x+num_deciles] = ((df2.xs(d21s[t], level=0)['Decile'] == x+1)* \
                             (df2.xs(d21s[t+time_lag], level=0)['Quality'])).replace(0, np.nan).mean()

# portfolio means at time t=0 + H-L
time1=df_pers.iloc[:,:10].mean(axis=0).append(pd.Series((df_pers.iloc[:,9]-df_pers.iloc[:,0]).mean(axis=0)))


# Calculate difference portfolio H-L t-statistics which is reported in table
reg=regression.linear_model.OLS((df_pers.iloc[:,9]-df_pers.iloc[:,0]).astype(float),\
           pd.Series([1]*len(df_pers.iloc[:,9])), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
time1 = time1.append(reg.tvalues)
time1.reset_index(drop=True, inplace=True)

# portfolio means at time lag + H-L

time2=df_pers.iloc[:,10:20].mean(axis=0).append(pd.Series((df_pers.iloc[:,19]-df_pers.iloc[:,10]).mean(axis=0)))

# Calculate difference portfolio H-L t-statistics which is reported in table
reg=regression.linear_model.OLS((df_pers.iloc[:,19]-df_pers.iloc[:,10]).astype(float),\
           pd.Series([1]*len(df_pers.iloc[:,19])), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
time2 = time2.append(reg.tvalues)
time2.reset_index(drop=True, inplace=True)

# Step two - running the same with different time lag
time_lag = 120

df_pers = pd.DataFrame(index=range(len(d21s)),columns=range(20))
for t in range(len(d21s)-time_lag):
    for x in range(num_deciles):
        df_pers.iloc[t,x] = ((df2.xs(d21s[t], level=0)['Decile'] == x+1)* \
                             (df2.xs(d21s[t], level=0)['Quality'])).replace(0, np.nan).mean()
        df_pers.iloc[t,x+num_deciles] = ((df2.xs(d21s[t], level=0)['Decile'] == x+1)* \
                             (df2.xs(d21s[t+time_lag], level=0)['Quality'])).replace(0, np.nan).mean()

# portfolio means at time lag + H-L

time3=df_pers.iloc[:,10:20].mean(axis=0).append(pd.Series((df_pers.iloc[:,19]-df_pers.iloc[:,10]).mean(axis=0)))

# Calculate difference portfolio H-L t-statistics which is reported in table
reg=regression.linear_model.OLS((df_pers.iloc[:,19]-df_pers.iloc[:,10]).astype(float),\
           pd.Series([1]*len(df_pers.iloc[:,19])), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
time3 = time3.append(reg.tvalues)
time3.reset_index(drop=True, inplace=True)

table = pd.concat([time1,time2,time3], axis=1).T
table.insert(loc=0, column='country', value=countries[c])
table.columns = ['country', '1','2','3','4','5','6','7','8','9','10','H-L','t-stat']

table4_1 = pd.concat([table4_1, table], axis=0)
```

In [43]:
```
table
```

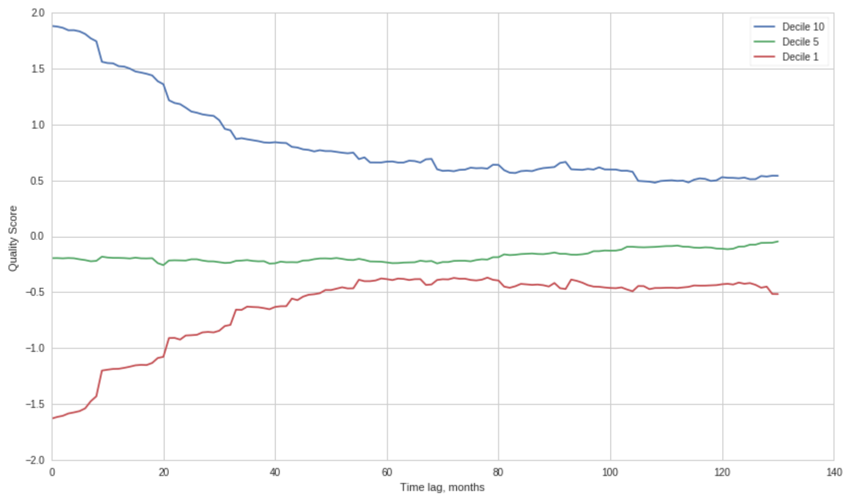
Out[43]:

	country	1	2	3	4	5	6	7	8	9	10	H-L	t-stat
0	US_EQUITIES	-1.600421	-0.991795	-0.714636	-0.471784	-0.218303	0.043986	0.328642	0.655801	1.077768	1.890787	3.491207	491.659326
1	US_EQUITIES	-0.713970	-0.541918	-0.453090	-0.361003	-0.213619	-0.042345	0.084985	0.285672	0.540355	0.980004	1.693974	57.943815
2	US_EQUITIES	-0.429558	-0.345658	-0.273342	-0.153566	-0.057271	-0.015952	0.035076	0.127851	0.348011	0.539534	0.969091	228.750323

In the next section of code we plot the persistence of Quality for Decile 1,5 & 10. These are plots that graphically summarize how quality changes over time for the portfolios formed at the start period. We see that the best and worst companies at time=0 measured by quality, on average, regress towards the mean, but the 'average' companies of decile 5 remain, on average, at the same quality score.

In [44]:
```python
decile_10=[]
decile_5=[]
decile_1=[]
for lag in range(len(d21s)):
    x = ((df2.xs(d21s[0], level=0)['Decile'] == 10)* (df2.xs(d21s[lag], level=0)['Quality'])).replace(0, np.nan).mean()
    decile_10.append(x)
    y = ((df2.xs(d21s[0], level=0)['Decile'] == 5)* (df2.xs(d21s[lag], level=0)['Quality'])).replace(0, np.nan).mean()
    decile_5.append(y)
    z = ((df2.xs(d21s[0], level=0)['Decile'] == 1)* (df2.xs(d21s[lag], level=0)['Quality'])).replace(0, np.nan).mean()
    decile_1.append(z)

fig=plt.figure()

ax=fig.add_subplot(111)
ax.plot(decile_10, label="Decile 10")
ax.plot(decile_5, label="Decile 5")
ax.plot(decile_1, label="Decile 1")
plt.xlabel('Time lag, months')
plt.ylabel('Quality Score')
plt.legend()
```

Out[44]:
```
<matplotlib.legend.Legend at 0x7f639dc61160>
```



3.2 The Price of Quality

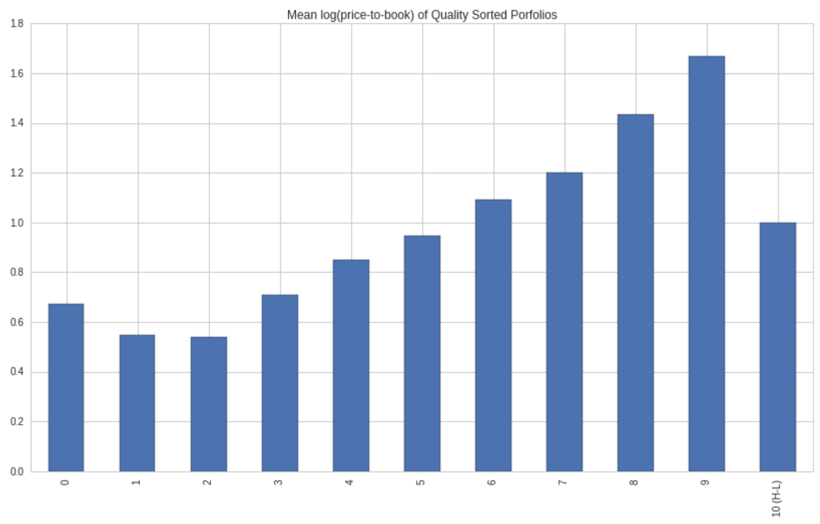
In this section we run the following regressions to find out if quality is related to the price of a company. We do this by running the following regressions:

1. $log(\frac{ME}{BE}) = a + b * Quality$
2. $log(\frac{ME}{BE}) = a + b_1 * Profitability + b_2 * Growth + b_3 * Safety$

In [ ]:

In [45]:
```
##############################################################
############################### CHECK VALUE WEIGHTING
##############################################################

def val_weight_avg(Y,W,D):
    num=(D*Y*W).sum()
    denom=(D*W).sum()
    return(num/denom)

t_p = d21s
data = pd.DataFrame(index=range(len(t_p)),columns=range(10))
t_hold = 0 # time period investor holds portfolio

for t in range(len(t_p)-t_hold):
    for x in range(num_deciles):
        try:
            Y = (df2.xs(t_p[t+t_hold], level=0)['ME_BE'].apply(np.log))
            W = df2.xs(t_p[t], level=0)['VW']
            D = (df2.xs(t_p[t], level=0)['Decile'] == x+1)
            data.iloc[t,x] = val_weight_avg(Y,W,D)
        except:
            pass

data['10 (H-L)'] =  (data.iloc[:,-1]-data.iloc[:,0])

# This bar plot shows the time series mean value of each portfolio
data.mean(axis=0).plot(kind='bar', title="Mean log(price-to-book) of Quality Sorted Porfolios")
```

Out[45]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f639dceaba8>
```



Cross-Sectional Regression

We now run a cross-sectional regression. This can also be done in different ways, but I find this Fama-Macbeth regression of price on quality approach easiest to implement. Start with a dataset and standard regression notation:

$$y_{it} = \beta x_{it} + \varepsilon$$
$$i = 1, 2, \dots, N$$
$$t = 1, 2, \dots, T$$

In an expected return-beta asset pricing model the x_{it} stands for β_i and β stands for λ. This can be confusing, but it's just how the discipline has always done it.

Step 1: The first step is a time series regression to get your right-hand variable x_{it}, i.e. the beta coefficients. In our case this is our Quality-factor.

Step 2: The lambda coefficients (here: λ_t) are estimates for the slope/risk-premium of your (risk-)factors β_t. What does this mean? We apply a cross-sectional regression at each point of time t. If there is a (linear) relationship between your risk factors β_t and stock price in period t, we would obtain a well-measured (i.e. statistical significant) positive factor risk-premium at t. The economic interpretation of λ_t is how much the expected stock price would rise, if this stocks risk-factor increases one unit.

We get estimates for the risk-premia λ´t at each point of time t.

For each time period t we do a cross-section regression on all assets i to get estimates for the risk-premia λ_t. Due to limited computational power (and statistical methodologies) in 1973, we simply use the variation in λ´t over time to deduce its variation across samples. β′iλt+ait

$$P_{it} = \alpha_{it} + \lambda_t \beta_i = \alpha_{it} + \lambda_t Quality_i$$

The regression we run is:

$$log(\frac{ME}{BE} + 1)_{it} = \alpha_{it} + \lambda_t Quality_i + \varepsilon_t$$

We then calculate the cross-sectional estimates for lambda with Fama-Macbeth sampling errors:

$$\hat{\lambda} = \frac{1}{T} \sum_{t=1}^{T} \hat{\lambda}_t$$

$$\sigma^2(\hat{\lambda}) = \frac{1}{T^2} \sum_{t=1}^{T} (\hat{\lambda}_t - \hat{\lambda})^2$$

The hypothesis test of α or $\lambda = 0$ then becomes:

$$\frac{\hat{\lambda}}{\sqrt{\sigma^2(\hat{\lambda})}} \sim t_{statistic}$$

3.2.1 Univariate regressions on Quality, Profitability, Growth and Safety

I first implemented the Fama-Macbeth procedure without correction for autocorrelation using the above formulas. This is shown in the code section below which is now inactive. The active code section below implements the F-M procedure slightly differently and which allows us to use the built-in Newey-West HAC (heteroskedasticity auto correlation) corrected standard errors.

# this code is retired - does not include autocorrelation corrected errors factors = ('Quality', 'Profitability', 'Growth', 'Safety') data = pd.DataFrame(columns=factors, index=range(6)) n=0 for s in factors: # setting up some variables to be used reg_1 = pd.DataFrame() betas = [] alphas = [] nobs = [] adjR = [] t_val = [] # use d21s for monthly, d252 for yearly # we run the cross-sectional regression for each year for t in d21s: X = df2[s][t] Y = (df2['ME_BE'][t]).apply(np.log1p) # note use of log1p = log(1+x) to avoid taking log of zero reg_1 =regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop').fit() alphas.append(reg_1.params[0]) betas.append(reg_1.params[1]) nobs.append(reg_1.nobs) adjR.append(reg_1.rsquared_adj) t_val.append(reg_1.tvalues[1]) n_obs = float(len(betas)) # calculation of Fama-Macbeth estimators, errors and t-statistic lambda_e = np.power(n_obs,-1)*(np.array(betas).sum()) lambda_err = np.power(n_obs,-2)*np.power((np.array(betas)-lambda_e),2).sum() lambda_t = lambda_e/np.sqrt(lambda_err) data.iloc[0,n] = lambda_e data.iloc[1,n] = np.power(lambda_err,0.5) data.iloc[2,n] = lambda_t data.iloc[3,n] = np.array(adjR).mean() data.iloc[4,n] = n_obs data.iloc[5,n] = np.array(nobs).mean() n=n+1 data.insert(loc=0, column=' ', value=['coefficient', 'std.error', 't-stat', 'adjR2', 'n_time', 'n_assets']) data

This implementation of the Fama-Macbeth procedure is like the above one in step 1 & 2, but in step 3 we do the following:

- Regress the $\hat{\lambda}_t = const + \varepsilon_t$ using Newey-West corrected standard errors
- The estimated intercept is the mean of $\hat{\lambda}$ and $se(const) = se(\hat{\lambda})$ corrected for heteroskedasticity and autocorrelation

  We verify the results by setting Newey-West lags equal to zero and see that the results are same as using the above approach

In [46]:
```
#################################################################
### Fama-Macbeth Procedure with HAC corrected standard errors ##
#################################################################

factors = ('Quality', 'Profitability', 'Growth', 'Safety')

data_stat = pd.DataFrame(columns=factors, index=range(6))
n=0
for s in factors:
    # setting up some variables to be used
    reg_1 = pd.DataFrame()
    betas = []
    alphas = []
    nobs = []
    adjR = []
    t_val = []

    # use d21s for monthly, d252 for yearly
    # we run the cross-sectional regression for each year
    for t in d21s:
        X = df2[s][t]
        Y = (df2['ME_BE'][t]).apply(np.log1p)    # note use of log1p = log(1+x) to avoid taking log of zero
        try:
            reg_1 =regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop').fit()
            alphas.append(reg_1.params[0])
            betas.append(reg_1.params[1])
            nobs.append(reg_1.nobs)
            adjR.append(reg_1.rsquared_adj)
            t_val.append(reg_1.tvalues[1])
        except:
            pass

    n_obs = float(len(betas))

    # calculation of Fama-Macbeth estimators, errors and t-statistic

    reg2 = regression.linear_model.OLS(betas, pd.Series([1]*len(betas)),\
                        missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
    data_stat.iloc[0,n] = reg2.params[0]
    data_stat.iloc[1,n] = reg2.bse[0]
    data_stat.iloc[2,n] = reg2.tvalues[0]
    data_stat.iloc[3,n] = np.array(adjR).mean()
    data_stat.iloc[4,n] = n_obs
    data_stat.iloc[5,n] = np.array(nobs).mean()
    n=n+1

data_stat.insert(loc=0, column=' ', value=['coefficient', 'std.error', 't-stat', 'adjR2', 'n_time', 'n_assets'])

table1 = pd.concat([data_stat.iloc[0,:], data_stat.iloc[2,:]], axis=1).T
table3 = (data_stat.iloc[3:,1]).to_frame().T
table3.columns = ['AdjR2', 'n_time', 'n_assets']


data_stat
```

Out[46]:

		Quality	Profitability	Growth	Safety
0	coefficient	0.183504	0.146308	0.14337	0.10835
1	std.error	0.00446812	0.00581031	0.00747079	0.00434272
2	t-stat	41.0697	25.1808	19.1908	24.9497
3	adjR2	0.0772477	0.0504276	0.0507488	0.0268854
4	n_time	131	131	131	131
5	n_assets	4235.98	4235.98	4235.98	4235.98

In [ ]:

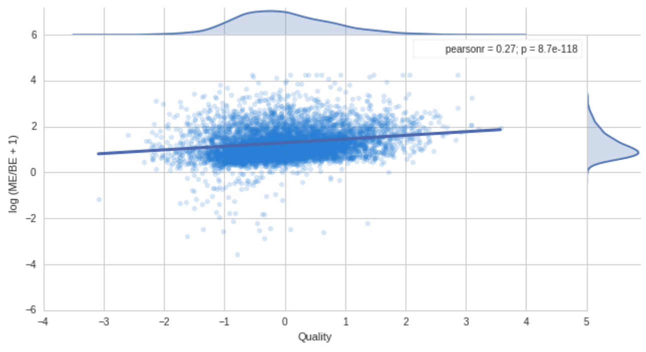
In [ ]:

3.2.2 Plotting univariate Price-Quality regression on sample data

We now make a regression plot of the sample data scattered and the fitted regression line for Quality. Sample distributions are plotted in the top and right margin.

In [47]:
```
from scipy import stats

df2['Quality'].mean(axis=0, level=1).min()
df2['Quality'].mean(axis=0, level=1).max()
x1 = np.linspace(df2['Quality'].mean(axis=0, level=1).min(), df2['Quality'].mean(axis=0, level=1).max(), 50)
y1 = reg_1.params[0] + reg_1.params[1]*x1

plt.figure(dpi=600)
g = sns.JointGrid(x=df2['Quality'].mean(axis=0, level=1), \
                  y=df2['ME_BE'].mean(axis=0, level=1).apply(np.log1p), \
                  space=0, size=5, ratio=10)
g.plot_joint(plt.scatter, alpha=0.2, color='#297FD5')
plt.plot(x1, y1, color='#4A66AC', linewidth=3)
g.plot_marginals(sns.kdeplot, shade=True)
g.annotate(stats.pearsonr)

g.fig.set_figwidth(10)
g.ax_joint.set_xlabel('Quality')
g.ax_joint.set_ylabel('log (ME/BE + 1)')
```

Out[47]:
```
<matplotlib.text.Text at 0x7f639daf00f0>

<matplotlib.figure.Figure at 0x7f639dcfb8d0>
```



3.2.3 Multivariate Regression of Price on Profitability, Growth and Safety

Here we run the combined regression of price on profitability, growth and safety

In [ ]:

In [48]:
```
data_stat = pd.DataFrame(columns=['const', 'Profitability', 'Growth', 'Safety'], index=range(6))


# setting up some variables to be used

n=0
nobs = []
adjR = []

betas=pd.DataFrame(index=range(len(d21s)), columns=range(4))

# we run the cross-sectional regression for each year
for t in d21s:
    X1 = df2['Profitability'][t]
    X2 = df2['Growth'][t]
    X3 = df2['Safety'][t]
    X = pd.concat([X1, X2, X3], axis=1)
    Y = (df2['ME_BE'][t]).apply(np.log1p)    # note use of log1p = log(1+x) to avoid taking log of zero
    try:
        reg_1 =regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop').fit()
        betas.iloc[n,:] = np.array(reg_1.params)
        nobs.append(reg_1.nobs)
        adjR.append(reg_1.rsquared_adj)
        n = n+1
    except:
        pass


t_obs = float(len(betas))

# calculation of Fama-Macbeth estimators, errors and t-statistic

for i in range(4):
    reg2 = regression.linear_model.OLS(np.asarray(betas.iloc[:,i], dtype=np.float64), \
                pd.Series([1]*len(betas.iloc[:,i])), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
    data_stat.iloc[0,i] = reg2.params[0]
    data_stat.iloc[1,i] = reg2.bse[0]
    data_stat.iloc[2,i] = reg2.tvalues[0]
    data_stat.iloc[3,i] = np.array(adjR).mean()
    data_stat.iloc[4,i] = t_obs
    data_stat.iloc[5,i] = np.array(nobs).mean()


data_stat.insert(loc=0, column=' ', value=['coefficient', 'std.error', 't-stat', 'adjR2', 'n_time', 'n_assets'])

table2 = pd.concat([data_stat.iloc[0,2:], data_stat.iloc[2,2:]], axis=1).T


data_stat
```

Out[48]:

		const	Profitability	Growth	Safety
0	coefficient	1.15329	0.0938384	0.067226	0.0973663
1	std.error	0.0264361	0.006163	0.00928325	0.00506223
2	t-stat	43.6254	15.2261	7.24165	19.2339
3	adjR2	0.0806242	0.0806242	0.0806242	0.0806242
4	n_time	131	131	131	131
5	n_assets	4235.98	4235.98	4235.98	4235.98

```
In [49]: # Create data table 4.2 going into the thesis

         table = pd.concat([table1.reset_index(drop=True),\
                            table2.reset_index(drop=True), \
                            table3.reset_index(drop=True)], axis=1)
         table.insert(loc=0, column='country', value=countries[c])

         table4_2 = table4_2.append(table)
```

Comparison using panel regression

For comparison I run a pooled time-series and cross-sectional regression with Newey-West corrected standard errors. This is just to get a sense-check that the above "manual" regressesions are within the correct order of magnitude. We know that pooled regression in cross-section and time will not be correct unless we perform clustering - which is difficult to achieve in the Python library statsmodels.

We first run a regression using quality alone as independant variable and then a new regression using the three factors Profitability, Growth and Safety. The the R-squared is very low, so we know that Quality definitely is not sufficent alone to explain the return variation. But quality has a significant beta coefficient, which is good.

Other things we look for in the model are our residuals, which are "analyzed" by:

- Omnibus/Prob(Omnibus) – a test of the skewness and kurtosis of the residual. The Prob (Omnibus) performs a statistical test indicating the probability that the residuals are normally distributed. We hope to see something less than 1, zero would would indicate perfect normalcy.
- Skew – a measure of data symmetry. We want to see something close to zero, indicating the residual distribution is normal. Note that this value also drives the Omnibus.
- Kurtosis – a measure of "peakiness", or curvature of the data. Higher peaks lead to greater Kurtosis. Greater Kurtosis can be interpreted as a tighter clustering of residuals around zero, implying a better model with few outliers.
- Durbin-Watson – tests for autocorrelation which takes values between 0 and 4. Two means there is no autocorrelation, more indicates positive correlation and vice versa.
- Jarque-Bera (JB)/Prob(JB) – like the Omnibus test in that it tests both skew and kurtosis. We hope to see in this test a confirmation of the Omnibus test and the number should be below 1. High numbers indicate that we reject the H0 hypothesis that the data is normally distributed.
- Condition Number – This test measures the sensitivity of a function's output as compared to its input. When we have multicollinearity, we can expect much higher fluctuations to small changes in the data, hence, we hope to see a relatively small number, something below 30. In this case our Quality model only has two variables and one is a constant so we are well below 30, which we would expect.

```
In [50]: X = df2['Quality']
         Y = (df2['ME_BE']).apply(np.log1p)
         reg_1 = regression.linear_model.OLS(Y, sm.add_constant(X),\
                                         missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
         reg_1.summary()
         # Newey-West parameter
         # maxlags = 4*(T/100)^a, a=0.22 or 0.16  [Ref. Newey-West(1987, 1994)]

         # Alternative library using pandas.OLS
         # pd.stats.ols.OLS(Y,X, nw_lags=6)
```

```
Out[50]: OLS Regression Results

              Dep. Variable:            ME_BE        R-squared:        0.072
                     Model:              OLS    Adj. R-squared:        0.072
                    Method:    Least Squares       F-statistic:    3.119e+04
                      Date: Sat, 13 Jun 2020 Prob (F-statistic):        0.00
                      Time:         21:37:18    Log-Likelihood: -5.5032e+05
         No. Observations:           554914               AIC:    1.101e+06
              Df Residuals:           554912               BIC:    1.101e+06
                  Df Model:                1
          Covariance Type:              HAC

                     coef  std err         z  P>|z|  [95.0% Conf. Int.]
            const  1.1529    0.001  1163.294  0.000         1.151 1.155
          Quality  0.1838    0.001   176.609  0.000         0.182 0.186

                Omnibus:  159343.675    Durbin-Watson:      1.894
          Prob(Omnibus):       0.000  Jarque-Bera (JB): 965082.289
                   Skew:       1.246          Prob(JB):       0.00
               Kurtosis:       8.961          Cond. No.       1.02
```

```
In [51]: X = df2['Profitability'].to_frame().join(df2['Growth']).join(df2['Safety'])
         Y = (df2['ME_BE']).apply(np.log1p)
         reg_1 = regression.linear_model.OLS(Y, sm.add_constant(X),\
                                 missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
         reg_1.summary()
         # Newey-West parameter inside fit() cov_type='HAC',cov_kwds={'maxlags':6}
         # maxlags = 4*(T/100)^a, a=0.22 or 0.16  [Ref. Newey-West(1987, 1994)]
```

```
Out[51]: OLS Regression Results

              Dep. Variable:            ME_BE        R-squared:        0.073
                     Model:              OLS    Adj. R-squared:        0.073
                    Method:    Least Squares       F-statistic:    1.074e+04
                      Date: Sat, 13 Jun 2020 Prob (F-statistic):        0.00
                      Time:         21:37:18    Log-Likelihood: -5.5009e+05
         No. Observations:           554914               AIC:    1.100e+06
              Df Residuals:           554910               BIC:    1.100e+06
                  Df Model:                3
          Covariance Type:              HAC

                          coef  std err         z  P>|z|  [95.0% Conf. Int.]
                 const  1.1525    0.001  1166.420  0.000         1.151 1.154
         Profitability  0.0921    0.002    60.149  0.000         0.089 0.095
                Growth  0.0687    0.001    49.656  0.000         0.066 0.071
                Safety  0.0977    0.001    95.915  0.000         0.096 0.100

                Omnibus:  159521.533    Durbin-Watson:      1.895
          Prob(Omnibus):       0.000  Jarque-Bera (JB): 959806.216
                   Skew:       1.250          Prob(JB):       0.00
               Kurtosis:       8.938          Cond. No.       2.47
```

3.3 The Return of Quality Stocks

3.3.1 Return of quality-sorted portfolios

Calculating Excess Return

In order to analyse returns of portfolios we first have to import the risk-free rate and return data. The risk free rate is taken as the 3month US Treasury Bill (T-bill) which we have downloaded manually from the Federal Reserve Bank data via Quandl. We create a column for excess return which is the Re = R - Rf

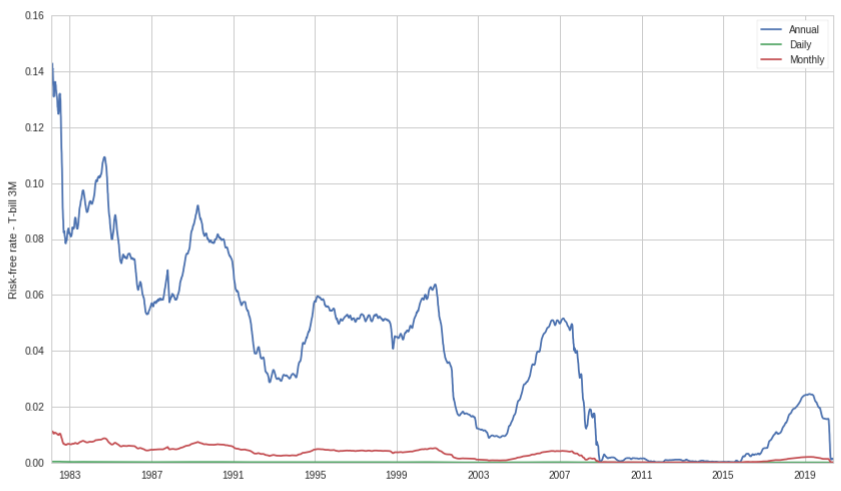
In [52]:
```
################################################
############ Utility Functions #################
################################################

# convert returns between time scales

def annual_monthly(x):
    return((1 + x)**(1/12) - 1)

def monthly_annual(x):
    return(((1+ x)**12)-1)

def daily_monthly(x):
    return(((1+ x)**21)-1)

def monthly_daily(x):
    return((1 + x)**(1/21) - 1)

def annual_daily(x):
    return((1 + x)**(1/365) - 1)

# winsorize-function, input array and percentage winsorization
def winsorize(x,p):
    q = 0.5*(1-p)
    x = x.clip(x.quantile(q), x.quantile(1-q))
    return(x)

# market-cap weighted average function - input is a series or frame
def mkt_weight(x):
    w = x.to_frame().join(df['MKT_VAL'])['MKT_VAL']
    return((x*w).sum()/w.sum())
```

In [ ]:
```

```

In [53]:
```
# Quandl download FRED-DGS3MO.csv: www.quandl.com/data/FRED/DGS3MO-3-Month-Treasury-Constant-Maturity-Rate
# Then upload FRED-DGS3MO.csv to Research for use as local_csv()
# Data is given in percent

fredData = local_csv('FRED-DGS3MO.csv', date_column='Date')
fredData['Rf_21'] = fredData[::-1].rolling(window=21).mean().dropna().div(100) # Approx Rf

# Add the risk free rate to df2 - for each date all assets will have the same Rf
df2['Rf'] = df2.reset_index(level=1).join(fredData['Rf_21']).set_index('level_1', \
                                    append=True)['Rf_21']
df2['Rf'].ffill(inplace=True)


## Add the asset returns as a column in our dataframe and remove data points which have no return value
df2 = df2.join(df['Returns_d'])
df2.dropna(axis=0, subset=['Returns_d'], inplace=True)
df2 = df2.join(df['Returns_m'])
df2.dropna(axis=0, subset=['Returns_m'], inplace=True)
df2 = df2.join(df['Returns_a'])
df2.dropna(axis=0, subset=['Returns_a'], inplace=True)
df2['Returns_d'] = winsorize(df2['Returns_d'],0.99)
df2['Returns_m'] = winsorize(df2['Returns_m'],0.99)
df2['Returns_a'] = winsorize(df2['Returns_a'],0.99)

# Re-create the column with value-weights because we have dropped some data
df2['VW'] = df2.join(df['MKT_VAL'])['MKT_VAL'].groupby(level=0).transform(lambda x: (x/x.sum()))
```

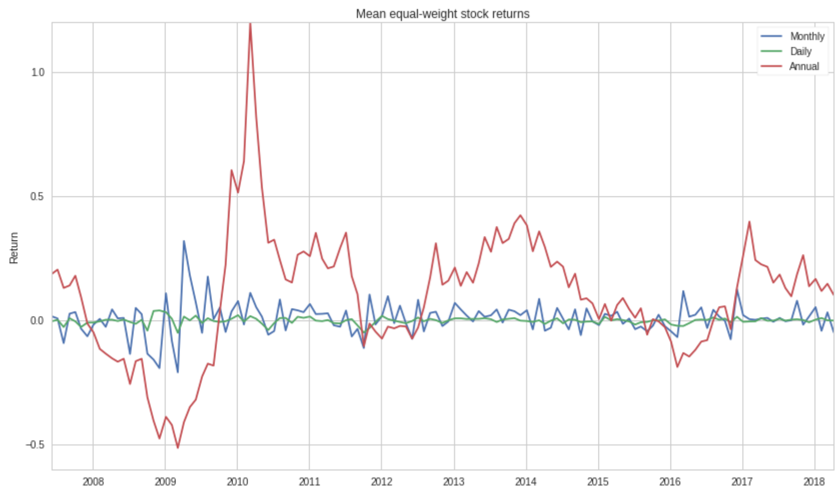
In [54]:
```
# Plot the daily, monthly and annual risk free rate to understand the data

fig=plt.figure()

ax=fig.add_subplot(111)
ax.plot(fredData['Rf_21'], label="Annual")
ax.plot(fredData['Rf_21'].apply(annual_daily), label="Daily")
ax.plot(fredData['Rf_21'].apply(annual_monthly), label="Monthly")
ax.set_ylabel("Risk-free rate - T-bill 3M")
plt.legend()
```

Out[54]:
```
<matplotlib.legend.Legend at 0x7f639cb4acc0>
```

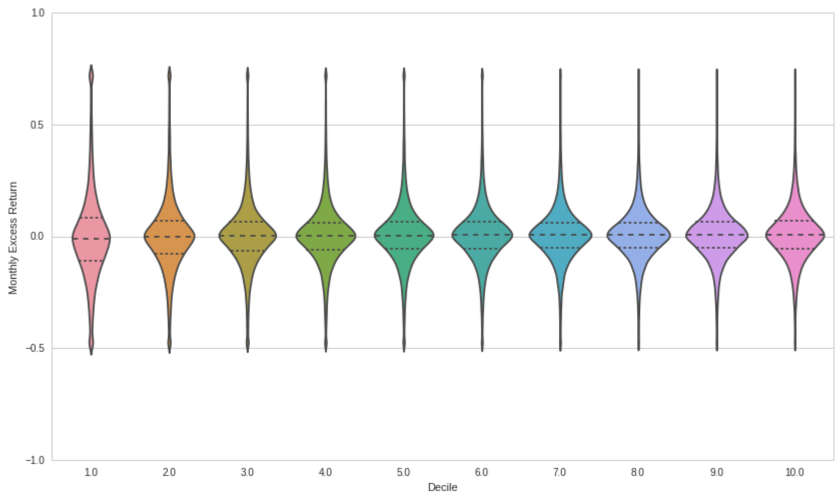


In [55]:
```
## Plot the mean Returns to get and understanding of the data
fig=plt.figure()

ax=fig.add_subplot(111)
ax.plot(df2['Returns_m'].groupby(level=0).mean(), label="Monthly")
ax.plot(df2['Returns_d'].groupby(level=0).mean(), label="Daily")
ax.plot(df2['Returns_a'].groupby(level=0).mean(), label="Annual")

plt.legend()
plt.title('Mean equal-weight stock returns')
ax.set_ylabel("Return")
```

Out[55]:
```
<matplotlib.text.Text at 0x7f639d957240>
```



In [56]:
```
Y=(df2['Returns_m']-df2['Rf'].apply(annual_monthly))
#Y=Y.clip(Y.quantile(0.005),Y.quantile(0.995))

sns.violinplot(x=df2['Decile'], \
               y=Y, \
               inner="quartile")
plt.ylabel("Monthly Excess Return")
plt.ylim(-1,1);
```



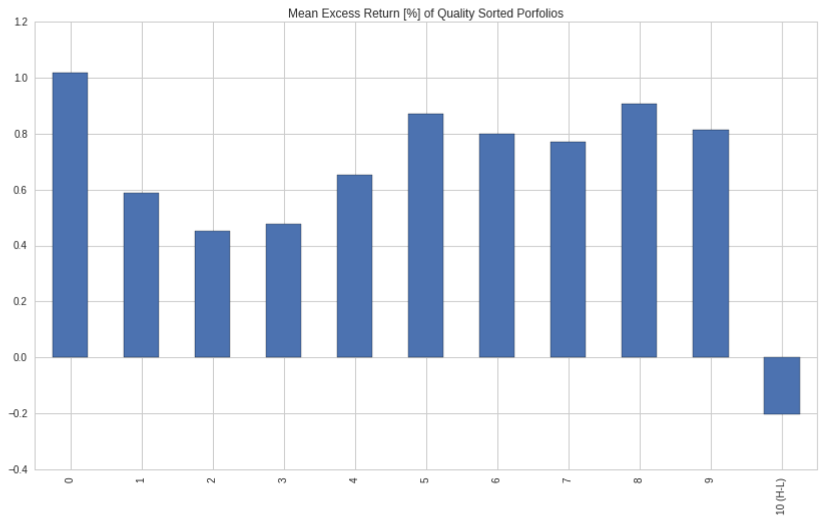
Create Quality Sorted Portfolios

In this section of code we do a portfolio analysis of the excess return on the 10 quality-sorted portfolios.

For each time step we collect the mean excess return for each decile at time t We iterate through our dataset and calculate the mean value across time for time t. This is stored in the dataframe 'data'. The values are calculated using the portfolio of stocks in each decile at time t. We take the average value of each decile and return in the table. Each column represent portfolios 1 (low) to 10 (high). We also include the 10-1 (high minus low) portfolio.

In [57]:
```
def val_weight_avg(Y,W,D):
    num=(D*Y*W).sum()
    denom=(D*W).sum()
    return(num/denom)

t_p = d21s
data = pd.DataFrame(index=range(len(t_p)),columns=range(10))
t_hold = 1 # time period investor holds portfolio

for t in range(len(t_p)-t_hold):
    for x in range(num_deciles):
        try:
            Y = (df2.xs(t_p[t+t_hold], level=0)['Returns_m']) \
            - df2.xs(t_p[t], level=0)['Rf'].apply(annual_monthly)
            W = df2.xs(t_p[t], level=0)['VW']
            D = (df2.xs(t_p[t], level=0)['Decile'] == x+1)
            data.iloc[t,x] = val_weight_avg(Y,W,D)
        except:
            pass

data['10 (H-L)'] =  (data.iloc[:,-1]-data.iloc[:,0])

# This bar plot shows the time series mean value of each portfolio
(100*data.mean(axis=0)).plot(kind='bar', title="Mean Excess Return [%] of Quality Sorted Porfolios")
```

Out[57]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f639d85b0b8>
```



Calculate Estimators and Statistical Inference

I know calculate the estimators, std.errors and t-statistic for the portfolios under null hypothesis of zero excess return. I calculate it in two ways to check my methods. First method is as per Cochrane - Asset Prices chapter 12.3, and the second method add Newey-West corrected standard errors using method of Woolridge p.351 and Empirical Asset Pricing chapter 1.3 & 5.1.5

I also test a third way using the pandas fama_macbeth method implementation.

I conclude that the second and third method are almost identical in standard errors and I will use the built-in method of statsmodels as it is more robust than my own implementation

In [58]:
```
# My implementation of Cochrane's Fama-Macbeth procedure -
# which only corrects for cross-correlation, but not autocorrelation

n_obs = float(len(data))
data_stat = pd.DataFrame(index=range(3), columns=data.columns)

# estimator "beta" (multiplied by 100 to get percentage values)
data_stat.iloc[0,:] = data.mean(axis=0)*100
# estimator variance
data_stat.iloc[1,:] = np.power(np.power(n_obs,-2)*np.power(data-data.mean(axis=0),2).sum(axis=0),0.5)*100
# estimator t-statistic
data_stat.iloc[2,:] = data_stat.iloc[0,:]/(data_stat.iloc[1,:])

data_stat.insert(loc=0, column=' ', value=['estimator', 'std.error', 't-stat'])
data_stat
```

Out[58]:

		0	1	2	3	4	5	6	7	8	9	10 (H-L)
0	estimator	1.01805	0.590086	0.45258	0.476049	0.652149	0.872899	0.799146	0.772923	0.908772	0.81272	-0.205326
1	std.error	0.739419	0.710461	0.588676	0.515707	0.539849	0.461505	0.478714	0.400339	0.413202	0.403794	0.476292
2	t-stat	1.37682	0.830568	0.76881	0.923099	1.20802	1.89142	1.66936	1.93067	2.19934	2.01271	-0.431092

In [59]:
```
# My implementation of Empirical Asset Pricing Ch 5.1.5 - which includes HAC corrected errors

data_stat2 = pd.DataFrame(index=range(3), columns=['0', '1','2','3','4','5','6','7','8','9','10 (H-L)'])

for i in range(11):
    reg1=regression.linear_model.OLS(data.iloc[:,i].astype(float),\
            pd.Series([1]*len(data.iloc[:,i])), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
    data_stat2.iloc[0,i] = reg1.params[0]*100  #monthly excess return in percent
    data_stat2.iloc[1,i] = reg1.bse[0]*100
    data_stat2.iloc[2,i] = reg1.tvalues[0]

data_stat2.insert(loc=0, column='_', value=['estimator', 'std.error', 't-stat'])
data_stat2
```

Out[59]:

	_	0	1	2	3	4	5	6	7	8	9	10 (H-L)
0	estimator	1.01805	0.590086	0.45258	0.476049	0.652149	0.872899	0.799146	0.772923	0.908772	0.81272	-0.205326
1	std.error	0.736016	0.636246	0.531978	0.525502	0.533112	0.44868	0.469951	0.393833	0.440156	0.405554	0.503948
2	t-stat	1.38318	0.927449	0.85075	0.905894	1.22329	1.94548	1.70049	1.96257	2.06466	2.00398	-0.407434

In [60]:
```python
# Using the method from pd.stats.fama_macbeth to calculate estimates

# input betas from pd.stats.ols.OLS, but be regression where beta is matrix
def calc_fm_stat(beta, nw_lags_beta):
    beta = np.array(beta.dropna(), dtype=np.float64)
    N = len(beta)
    B = beta - beta.mean(0)
    C = np.dot(B.T, B) / N

    if nw_lags_beta is not None:
        for i in range(nw_lags_beta + 1):

            cov = np.dot(B[i:].T, B[:(N - i)]) / N
            weight = i / (nw_lags_beta + 1)
            C += 2 * (1 - weight) * cov

    mean_beta = beta.mean(0)
    std_beta = np.sqrt(np.diag(C)) / np.sqrt(N)
    t_stat = mean_beta / std_beta

    return mean_beta, std_beta, t_stat

data_stat = pd.DataFrame(index=range(3), columns=data.columns)

data_stat.iloc[0,:] = calc_fm_stat(data, nw_lags_beta=5)[0]*100
data_stat.iloc[1,:] = calc_fm_stat(data, nw_lags_beta=5)[1]*100
data_stat.iloc[2,:] = calc_fm_stat(data, nw_lags_beta=5)[2]
data_stat.insert(loc=0, column=' ', value=['estimator', 'std.error', 't-stat'])
data_stat
```

Out[60]:

		0	1	2	3	4	5	6	7	8	9	10 (H-L)
0	estimator	1.01805	0.590086	0.45258	0.476049	0.652149	0.872899	0.799146	0.772923	0.908772	0.81272	-0.205326
1	std.error	1.29692	1.21057	1.00034	0.909026	0.939003	0.802545	0.833737	0.696221	0.737184	0.707176	0.848524
2	t-stat	0.784969	0.487445	0.452428	0.523691	0.694513	1.08766	0.958511	1.11017	1.23276	1.14925	-0.24198

3.3.2 Risk-Adjusted Returns of Quality-Sorted Portfolios

In this section we want to examine whether patterns in the average portfolio returns are driven by cross-sectional variation in portfolio sensitivities to systematic risk factors, i.e. if the patterns from previous section persist when controlling for known systematic risk factors.

$$r_{et} = \alpha + \beta_{MKT} MKT_t + \beta_{SMB} SMB_t + \beta_{HML} HML_t + \beta_{MOM} MOM_t + \varepsilon_t$$

Creating the Factor Mimicking Portfolios

The Fama-French-Carhart factor mimicking portfolios are:

$$MKT = R_{mkt} - R_f$$
$$SMB = \frac{R_{sv} + R_{sn} + R_{sg}}{3} - \frac{R_{bv} + R_{bn} + R_{bg}}{3}$$
$$HML = \frac{R_{sv} + R_{bv}}{2} - \frac{(R_{sg} + R_{bg}}{2}$$
$$MOM = \frac{R_{HighMom} + R_{LowMom}}{2}$$

And the Quality factor QMJ is built as per Asness et al at the intersection of six value-weighted portfolios formed on size and quality:

$$QMJ = \frac{R_{sq} - R_{sj}}{2} + \frac{(R_{bq} - R_{bj}}{2}$$

In [ ]:

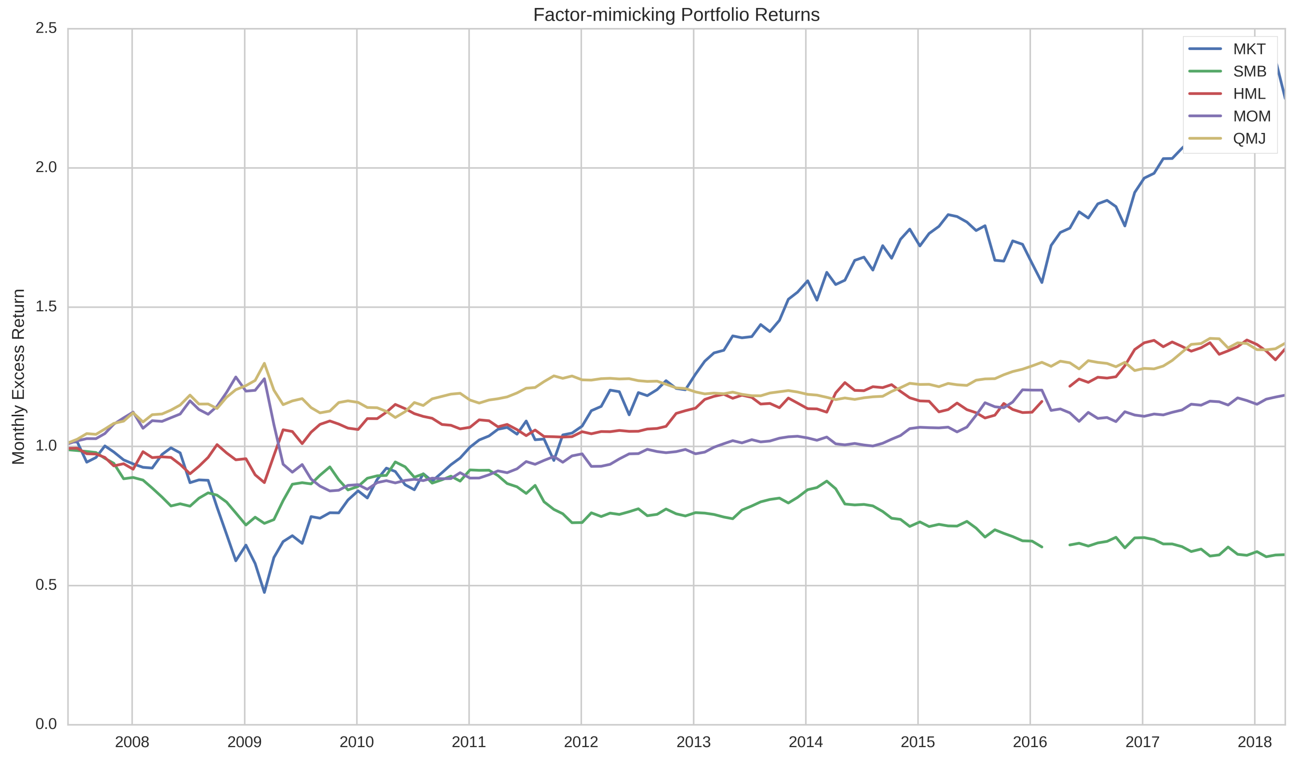
```
In [61]: # Create masks for the double-sort on size and quality
         # using 80th quantile as break-points for size and quality as per QMJ paper

         size = df2.join((pd.qcut(df['MKT_VAL'], 10, labels=False)+1))['MKT_VAL']
         large = size.isin([8, 9, 10])
         small = size.isin([1, 2, 3, 4, 5, 6, 7])
         highQ = df2['Decile'].isin([8, 9, 10])
         lowQ = df2['Decile'].isin([1, 2, 3, 4, 5, 6, 7])

         ### Masks to create SMV, HML and MOM portfolios
         #small value
         sv = df2.join(df.sv)['sv']
         #small neutral
         sn = df2.join(df.sn)['sn']
         #small growth
         sg = df2.join(df.sg)['sg']
         #big value
         bv = df2.join(df.bv)['bv']
         #big neutral
         bn = df2.join(df.bn)['bn']
         #big growth
         bg = df2.join(df.bg)['bg']
         #high momentum
         highmom = df2.join(df.highmom)['highmom']
         #low momentum
         lowmom = df2.join(df.lowmom)['lowmom']

         ### Masks to create the QMJ portfolio
         #small quality
         sq = highQ[small].align(df2)[0].fillna(False)
         #small junk
         sj = lowQ[small].align(df2)[0].fillna(False)
         #big quality
         bq = highQ[large].align(df2)[0].fillna(False)
         #big junk
         bj = lowQ[large].align(df2)[0].fillna(False)

         ## Create the factor returns
         # Monthly returns minus risk free rate - weighted by market cap weights for each month
         df2['Rexcess_m'] = (df2['Returns_m'] - df2['Rf'].apply(annual_monthly)) #*df2['VW']

         # group_by(level=0).mean() gives you the average return of each day for a particular group of stocks
         R_sv = df2[sv]['Rexcess_m'].groupby(level=0).mean()
         R_sn = df2[sn]['Rexcess_m'].groupby(level=0).mean()
         R_sg = df2[sg]['Rexcess_m'].groupby(level=0).mean()

         R_bv = df2[bv]['Rexcess_m'].groupby(level=0).mean()
         R_bn = df2[bn]['Rexcess_m'].groupby(level=0).mean()
         R_bg = df2[bg]['Rexcess_m'].groupby(level=0).mean()

         R_highmom = df2[highmom]['Rexcess_m'].groupby(level=0).mean()
         R_lowmom = df2[lowmom]['Rexcess_m'].groupby(level=0).mean()

         R_sq = df2[sq]['Rexcess_m'].groupby(level=0).mean()
         R_sj = df2[sj]['Rexcess_m'].groupby(level=0).mean()
         R_bq = df2[bq]['Rexcess_m'].groupby(level=0).mean()
         R_bj = df2[bj]['Rexcess_m'].groupby(level=0).mean()

         # Market return - value weighted market return for each day
         df2['Rm'] = df2.reset_index(level=1).join(df2['Returns_m'].groupby(level=0)\
                                  .apply(mkt_weight).to_frame(name='Rm'))\
                                  .set_index('level_1', append=True)['Rm']

         # Defining our final factor mimicking portfolio returns
         MKT = (df2['Rm'] - df2['Rf'].apply(annual_monthly)).groupby(level=0).mean()
         SMB = (R_sv + R_sn + R_sg)/3 - (R_bv + R_bn + R_bg)/3
         HML = (R_sv + R_bv)/2 - (R_sg + R_bg)/2
         MOM = 0.5*(R_highmom - R_lowmom)
         QMJ = 0.5*(R_sq - R_sj)+0.5*(R_bq - R_bj)

         # Add factor returns to dataframe - one value for all assets per date
         df2['MKT'] = df2.reset_index(level=1).join(MKT.to_frame(name='MKT'))\
                      .set_index('level_1', append=True)['MKT']
         df2['SMB'] = df2.reset_index(level=1).join(SMB.to_frame(name='SMB'))\
                      .set_index('level_1', append=True)['SMB']
         df2['HML'] = df2.reset_index(level=1).join(HML.to_frame(name='HML'))\
                      .set_index('level_1', append=True)['HML']
         df2['MOM'] = df2.reset_index(level=1).join(MOM.to_frame(name='MOM'))\
                      .set_index('level_1', append=True)['MOM']
         df2['QMJ'] = df2.reset_index(level=1).join(QMJ.to_frame(name='QMJ'))\
                      .set_index('level_1', append=True)['QMJ']
```

```
In [ ]:
```

Here we show some summary data on the created factor-mimicking portfolio returns. Plot of the factor returns and the correlation matrix

In [62]:
```
fig=plt.figure(dpi=600)

ax=fig.add_subplot(111)
ax.plot((1+MKT).cumprod(), label="MKT")
ax.plot((1+SMB).cumprod(), label="SMB")
ax.plot((1+HML).cumprod(), label="HML")
ax.plot((1+MOM).cumprod(), label="MOM")
ax.plot((1+QMJ).cumprod(), label="QMJ")
plt.ylabel("Monthly Excess Return")
plt.title("Factor-mimicking Portfolio Returns")
plt.legend()
```

Out[62]:
```
<matplotlib.legend.Legend at 0x7f639b7078d0>
```



In [ ]:

pd.DataFrame(np.transpose([MKT, SMB, HML, MOM, QMJ]), \ columns=['MKT', 'SMB', 'HML', 'MOM', 'QMJ']).corr(method='pearson')

Regressions on Quality-sorted portfolios

In this section we want to examine whether patterns in the average portfolio returns are driven by cross-sectional variation in portfolio sensitivities to systematic risk factors, i.e. if the patterns from previous section persist when controlling for known systematic risk factors. Now that we have created the factor-mimicking portfolios we can run the regressions for each portfolio

$$r_{et} = \alpha + \beta_{MKT} MKT_t + \beta_{SMB} SMB_t + \beta_{HML} HML_t + \beta_{MOM} MOM_t + \varepsilon_t$$

In [63]:
```
########## FAMA-MACBETH 2-pass regression of excess returns on risk factors ################
############################################################################################

## Step 1: Estimating Betas for each asset

assets = df2.index.levels[1].unique()

betas=pd.DataFrame(index=range(len(assets)), columns=range(7))
betas.columns = ['Asset', 'Const', 'MKT', 'SMB', 'HML', 'MOM', 'QMJ']

betasCAPM=pd.DataFrame(index=range(len(assets)), columns=range(3))
betasCAPM.columns = ['Asset', 'Const', 'MKT']

betasFF=pd.DataFrame(index=range(len(assets)), columns=range(5))
betasFF.columns = ['Asset', 'Const', 'MKT', 'SMB', 'HML']

betasFFC=pd.DataFrame(index=range(len(assets)), columns=range(6))
betasFFC.columns = ['Asset', 'Const', 'MKT', 'SMB', 'HML', 'MOM']

nobs = []
adjR = []

for i in range(len(assets)):
    Y = df2.xs(assets[i], level=1)['Rexcess_m']

    X1 = df2.xs(assets[i], level=1)['MKT']
    X2 = df2.xs(assets[i], level=1)['SMB']
    X3 = df2.xs(assets[i], level=1)['HML']
    X4 = df2.xs(assets[i], level=1)['MOM']
    X5 = df2.xs(assets[i], level=1)['QMJ']
    X = pd.concat([X1, X2, X3, X4, X5], axis=1)    # FFC + QMJ
    X_FF = pd.concat([X1, X2, X3], axis=1)         #fama-french 3-factors
    X_FFC = pd.concat([X1, X2, X3, X4], axis=1)    #fama-french-carhart 4-factors

    if not(Y.empty or X.empty):   #this excludes assets which do not have data
        try:
            reg_1 =regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop').fit()
            betas.iloc[i,0] = assets[i]
            betas.iloc[i,1:] = np.array(reg_1.params)
            # regression on CAPM (market factor)
            reg_2 =regression.linear_model.OLS(Y, sm.add_constant(X1), missing='drop').fit()
            betasCAPM.iloc[i,0] = assets[i]
            betasCAPM.iloc[i,1:] = np.array(reg_2.params)
            # regression on FF 3-factors
            reg_3 =regression.linear_model.OLS(Y, sm.add_constant(X_FF), missing='drop').fit()
            betasFF.iloc[i,0] = assets[i]
            betasFF.iloc[i,1:] = np.array(reg_3.params)
            # regression on FFC 4-factor
            reg_4 =regression.linear_model.OLS(Y, sm.add_constant(X_FFC), missing='drop').fit()
            betasFFC.iloc[i,0] = assets[i]
            betasFFC.iloc[i,1:] = np.array(reg_4.params)
            adjR.append(reg_4.rsquared_adj)
            nobs.append(reg_4.nobs)
        except:
            pass

betas.dropna(inplace=True)
betasCAPM.dropna(inplace=True)
betasFF.dropna(inplace=True)
betasFFC.dropna(inplace=True)
```

In [76]:
```
# Step 2: Estimate risk premia for each quality-sorted portfolio
n=0
nobs2 = []
adjR2 = []

# the alpha DF will store regression intercepts for the 3 models (CAPM, FF, FFC) for each period
alphasCAPM = pd.DataFrame(index=range(len(d21s)), columns=range(10))
alphasFF = pd.DataFrame(index=range(len(d21s)), columns=range(10))
alphasFFC = pd.DataFrame(index=range(len(d21s)), columns=range(10))
port_return = pd.DataFrame(index=range(len(d21s)), columns=range(10))
port_beta = pd.DataFrame(index=range(len(d21s)), columns=range(10))

t_p = d21s
data = pd.DataFrame(index=range(len(t_p)),columns=range(10))


for t in range(len(t_p)):
    for x in range(num_deciles):
        try:
            Y = ((df2.xs(t_p[t], level=0)['Decile'] == x+1) \
                *df2.xs(t_p[t], level=0)['Rexcess_m']).replace(0, np.nan).dropna()
            X1 = Y.to_frame().join(betasCAPM.set_index('Asset')).iloc[:,2:]  #CAPM betas per asset
            X2 = Y.to_frame().join(betasFF.set_index('Asset')).iloc[:,2:]    #FF betas per asset
            X3 = Y.to_frame().join(betasFFC.set_index('Asset')).iloc[:,2:]   #FFC betas per asset

            reg_1 =regression.linear_model.OLS(Y, \
                    sm.add_constant(np.array(X1, dtype=np.float64)), missing='drop').fit()
            reg_2 =regression.linear_model.OLS(Y, \
                    sm.add_constant(np.array(X2, dtype=np.float64)), missing='drop').fit()
            reg_3 =regression.linear_model.OLS(Y, \
                    sm.add_constant(np.array(X3, dtype=np.float64)), missing='drop').fit()
            alphasCAPM.iloc[t,x] = reg_1.params[0]
            alphasFF.iloc[t,x] = reg_2.params[0]
            alphasFFC.iloc[t,x] = reg_3.params[0]
            nobs2.append(reg_3.nobs)
            adjR2.append(reg_3.rsquared_adj)
            port_return.iloc[t,x] = Y.mean()
            port_beta.iloc[t,x] = reg_3.params[1]
            n = n+1
        except:
            pass



alphasCAPM['10 (H-L)'] =  (alphasCAPM.iloc[:,-1]-alphasCAPM.iloc[:,0])
alphasFF['10 (H-L)'] =  (alphasFF.iloc[:,-1]-alphasFF.iloc[:,0])
alphasFFC['10 (H-L)'] =  (alphasFFC.iloc[:,-1]-alphasFFC.iloc[:,0])
port_return['10 (H-L)'] =  (port_return.iloc[:,-1]-port_return.iloc[:,0])
```

In [65]:
```
### Inference ###
# My implementation of Empirical Asset Pricing Ch 5.1.5 - which includes HAC corrected errors

data_stat = pd.DataFrame(index=range(10), columns=['0', '1','2','3','4','5','6','7','8','9','10 (H-L)'])

for i in range(11):
    # regression of CAPM alpha
    Y = alphasCAPM.iloc[:,i].astype(float)
    reg1=regression.linear_model.OLS(Y,\
            pd.Series([1]*len(Y)), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})

    Y = alphasFF.iloc[:,i].astype(float)
    reg2=regression.linear_model.OLS(Y,\
            pd.Series([1]*len(Y)), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})

    Y = alphasFFC.iloc[:,i].astype(float)
    reg3=regression.linear_model.OLS(Y,\
            pd.Series([1]*len(Y)), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
    # regression of mean portfolio returns
    Y = port_return.iloc[:,i].astype(float)
    reg4=regression.linear_model.OLS(Y,\
            pd.Series([1]*len(Y)), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})

    data_stat.iloc[0,i] = reg4.params[0]*100
    data_stat.iloc[1,i] = reg4.tvalues[0]

    data_stat.iloc[2,i] = reg1.params[0]*100  #monthly alpha in percent
    data_stat.iloc[3,i] = reg1.tvalues[0]

    data_stat.iloc[4,i] = reg2.params[0]*100
    data_stat.iloc[5,i] = reg2.tvalues[0]

    data_stat.iloc[6,i] = reg3.params[0]*100
    data_stat.iloc[7,i] = reg3.tvalues[0]

    data_stat.iloc[8,i] = ((1+reg4.params[0])**12 -1)/(reg4.resid.std()*np.sqrt(12))
    data_stat.iloc[9,i] = ((1+reg3.params[0])**12 -1)/(reg3.resid.std()*np.sqrt(12))

data_stat.insert(loc=0, column='_', value=['Excess Return', 't-stat', 'CAPM alpha', 't-stat', \
                                           'Fama-French alpha', 't-stat', 'FFC alpha', 't-stat', \
                                             'Sharpe Ratio', 'Information Ratio'])
#data_stat2.append(data_stat)
data_stat['Country'] = countries[c]
```

In [91]:
```
alphasFFC.apply(lambda x: (1+x).cumprod()).iloc[120:,:]
```

Out[91]:

	0	1	2	3	4	5	6	7	8	9	10 (H-L)
120	0.704868	1.256325	1.644912	2.260492	2.797866	2.804400	3.240667	2.206529	3.713175	2.917486	3.752644
121	0.689643	1.272269	1.687431	2.364420	2.760376	2.790931	3.256542	2.240481	3.715125	2.927333	3.846367
122	0.691412	1.287948	1.689873	2.357322	2.791911	2.841769	3.268995	2.276725	3.780268	2.954420	3.872090
123	0.697083	1.323625	1.717553	2.362648	2.829850	2.822727	3.277184	2.278417	3.873478	2.995926	3.894728
124	0.678276	1.368602	1.761479	2.414902	2.938883	2.909527	3.342193	2.339538	3.906976	3.035422	4.051152
125	0.683524	1.350447	1.724380	2.392932	2.935087	2.863577	3.326930	2.347347	3.847156	3.023102	4.003365
126	0.722112	1.370219	1.743104	2.454590	2.993673	2.830955	3.445877	2.371715	3.936717	3.038371	3.797578
127	0.718575	1.365171	1.737397	2.433271	2.933816	2.766757	3.406215	2.382905	3.924993	2.969197	3.729716
128	0.691826	1.347377	1.726431	2.373512	2.922785	2.780400	3.409542	2.318394	3.774602	2.920680	3.807616
129	0.707672	1.329867	1.739530	2.424268	2.950193	2.791981	3.483752	2.299084	3.864684	2.941242	3.747205
130	0.722789	1.325993	1.731964	2.432799	2.993411	2.854041	3.551977	2.333479	3.904884	2.961567	3.693057

In [ ]:

From the resulting table above we see that excess return increases with quality

In [66]:
```
# this table is used in the thesis:

table4_3 = table4_3.append(data_stat)
```

I also run the second step on the whole sample (not quality-sorted). The results are not used in the thesis

In [67]:
```
# Step 2: Estimating risk premia using entire cross-sectional sample using the FFC 4-factors + QMJ

n=0
nobs = []
adjR = []
alpha = []

betas2=pd.DataFrame(index=range(len(d21s)), columns=range(6))

# we run the cross-sectional regression for each year
for t in d21s:
    try:
        Y = df2['Rexcess_m'][t]
        X = Y.to_frame().join(betas.set_index('Asset')).iloc[:,2:]  # add estimated betas from previous step
        X = np.array(X, dtype=np.float64)
        reg_1 =regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop').fit()
        betas2.iloc[n,:] = np.array(reg_1.params)
        nobs.append(reg_1.nobs)
        adjR.append(reg_1.rsquared_adj)
        n = n+1
    except:
        pass

t_obs = float(len(betas2))

### Inferring results from periodic cross-sectional regressions

data_stat = pd.DataFrame(columns=['alpha', 'MKT', 'SMB', 'HML', 'MOM', 'QMJ'], index=range(6))

# calculation of Fama-Macbeth estimators, errors and t-statistic

for i in range(6):
    reg2 = regression.linear_model.OLS(np.asarray(betas2.iloc[:,i], dtype=np.float64), \
                pd.Series([1]*len(betas2)), missing='drop').fit(cov_type='HAC',cov_kwds={'maxlags':6})
    data_stat.iloc[0,i] = reg2.params[0]*100
    data_stat.iloc[1,i] = reg2.bse[0]*100
    data_stat.iloc[2,i] = reg2.tvalues[0]
    data_stat.iloc[3,i] = np.array(adjR).mean()
    data_stat.iloc[4,i] = t_obs
    data_stat.iloc[5,i] = np.array(nobs).mean()


data_stat.insert(loc=0, column='_', value=['coefficient', 'std.error', 't-stat', 'adjR2', 'n_time', 'n_assets'])

data_stat
```

Out[67]:

	_	alpha	MKT	SMB	HML	MOM	QMJ
0	coefficient	0.66343	0.211168	-0.252986	-0.101139	0.19587	0.0219661
1	std.error	0.256042	0.410081	0.225862	0.153122	0.22245	0.0999381
2	t-stat	2.5911	0.514942	-1.12009	-0.66051	0.880512	0.219798
3	adjR2	0.125912	0.125912	0.125912	0.125912	0.125912	0.125912
4	n_time	131	131	131	131	131	131
5	n_assets	3935.51	3935.51	3935.51	3935.51	3935.51	3935.51

3.3.3 Returns of the QMJ Portfolio

We now do a regression of QMJ on the risk factors

$$QMJ_t = \alpha + \beta MKT_t + \beta SMB_t + \beta HML_t + \beta MOM_t$$

The resulting table is similar to Table 3 of Asness et al.'s QMJ paper.

In [68]:
```
# Data for regressions
Y=QMJ
X=pd.concat([MKT, SMB, HML, MOM], axis=1)
X.columns = ['MKT', 'SMB', 'HML', 'MOM']

# reg 1: CAPM alpha
reg1 = regression.linear_model.OLS(Y, sm.add_constant(X['MKT']), missing='drop')\
                .fit(cov_type='HAC', cov_kwds={'maxlags':6}, use_t=True)

# reg 2: FF 3-factor alpha
reg2 = regression.linear_model.OLS(Y, sm.add_constant(X[['MKT', 'SMB', 'HML']]), missing='drop')\
                .fit(cov_type='HAC', cov_kwds={'maxlags':6}, use_t=True)

# reg 3: FFC 4-factor alpha and loadings
reg3 = regression.linear_model.OLS(Y, sm.add_constant(X), missing='drop')\
                .fit(cov_type='HAC', cov_kwds={'maxlags':6}, use_t=True)

# reg 4: Excess return of QMJ
reg4 = regression.linear_model.OLS(np.array(QMJ), pd.Series([1]*len(QMJ)), missing='drop')\
                .fit(cov_type='HAC', cov_kwds={'maxlags':6}, use_t=True)


# Display results in table
data_table4 = pd.DataFrame(index=range(2), columns=['Excess Return',  'CAPM-alpha',  \
                                             '3-factor alpha',  '4-factor alpha',\
                                             'MKT', 'SMB',  'HML',  'MOM',  \
                                             'Sharpe Ratio', 'IR', 'AdjR2'])

data_table4.iloc[0,0] = reg4.params[0]*100
data_table4.iloc[1,0] = reg4.tvalues[0]
data_table4.iloc[0,1] = reg1.params[0]*100
data_table4.iloc[1,1] = reg1.tvalues[0]
data_table4.iloc[0,2] = reg2.params[0]*100
data_table4.iloc[1,2] = reg2.tvalues[0]
data_table4.iloc[0,3] = reg3.params[0]*100
data_table4.iloc[1,3] = reg3.tvalues[0]
data_table4.iloc[0,4] = reg3.params[1]
data_table4.iloc[1,4] = reg3.tvalues[1]
data_table4.iloc[0,5] = reg3.params[2]
data_table4.iloc[1,5] = reg3.tvalues[2]
data_table4.iloc[0,6] = reg3.params[3]
data_table4.iloc[1,6] = reg3.tvalues[3]
data_table4.iloc[0,7] = reg3.params[4]
data_table4.iloc[1,7] = reg3.tvalues[4]
data_table4.iloc[0,8] = ((reg4.params[0]+1)**12-1) / (reg4.resid.std()*np.sqrt(12))
data_table4.iloc[0,9] = ((reg3.params[0]+1)**12-1) / (reg3.resid.std()*np.sqrt(12))
data_table4.iloc[0,10] = reg3.rsquared_adj

data_table4
data_table4.insert(loc=0, column='_', value=['coefficient', 't-stat'])
data_table4.insert(loc=0, column='Country', value=countries[c])

table4_4 = table4_4.append(data_table4)
```

```
In [ ]:
```

```
In [ ]:
```

4. Results

We'll now print the tables that will be used in the thesis

```
In [69]: # summary statistics
         table4_0
```

Out[69]:

	Country	Mean Market Cap	Stocks per month	Total Stokcs	Used Stocks
0	US_EQUITIES	4.455770e+09	4372.687023	7621	7621

```
In [70]: # Price of quality-sorted portfolios and persistence

         table4_1
```

Out[70]:

	country	1	2	3	4	5	6	7	8	9	10	H-L	t-stat
0	US_EQUITIES	-1.600421	-0.991795	-0.714636	-0.471784	-0.218303	0.043986	0.328642	0.655801	1.077768	1.890787	3.491207	491.659326
1	US_EQUITIES	-0.713970	-0.541918	-0.453090	-0.361003	-0.213619	-0.042345	0.084985	0.285672	0.540355	0.980004	1.693974	57.943815
2	US_EQUITIES	-0.429558	-0.345658	-0.273342	-0.153566	-0.057271	-0.015952	0.035076	0.127851	0.348011	0.539534	0.969091	228.750323

```
In [71]: # Regressions of price on quality, P, G & S
         table4_2
```

Out[71]:

	country		Quality	Profitability	Growth	Safety	Profitability	Growth	Safety	AdjR2	n_time	n_assets
0	US_EQUITIES	coefficient	0.183504	0.146308	0.14337	0.10835	0.0938384	0.067226	0.0973663	0.0772477	131	4235.98
1	US_EQUITIES	t-stat	41.0697	25.1808	19.1908	24.9497	15.2261	7.24165	19.2339	NaN	NaN	NaN

```
In [72]: # Returns of quality sorted portfolios and risk-adjusted returns
         table4_3
```

Out[72]:

	_	0	1	2	3	4	5	6	7	8	9	10 (H-L)	Country
0	Excess Return	0.0673576	0.417252	0.543829	0.695894	0.829649	0.876736	0.930652	0.928255	0.926384	0.93795	0.870593	US_EQUITIES
1	t-stat	0.0781559	0.610886	0.920765	1.18966	1.44903	1.53674	1.71569	1.73908	1.77792	1.8888	2.04419	US_EQUITIES
2	CAPM alpha	-0.227111	0.393517	0.401278	0.589979	0.840627	0.738867	1.0863	0.837306	1.12756	0.942859	1.16997	US_EQUITIES
3	t-stat	-0.589954	1.10707	1.40061	2.02546	3.65401	2.97195	5.9858	4.74537	4.95971	4.07286	3.28063	US_EQUITIES
4	Fama-French alpha	-0.147971	0.345595	0.494673	0.704804	0.907279	0.895915	1.03251	0.743044	1.0649	0.771266	0.919237	US_EQUITIES
5	t-stat	-0.420465	1.24979	1.83814	2.3997	4.47384	4.12286	5.45873	4.31125	5.35265	3.6637	3.10086	US_EQUITIES
6	FFC alpha	-0.196993	0.242772	0.448243	0.706019	0.859483	0.821777	0.990579	0.669302	1.06426	0.850802	1.0478	US_EQUITIES
7	t-stat	-0.536659	0.900514	2.13515	2.84525	5.08045	4.18331	5.72724	3.67147	5.92375	4.06755	3.53021	US_EQUITIES
8	Sharpe Ratio	0.0268358	0.193928	0.270845	0.368288	0.44556	0.489639	0.548546	0.559272	0.558777	0.585857	0.816248	US_EQUITIES
9	Information Ratio	-0.212374	0.364807	0.668641	1.13021	1.58629	1.55921	1.87721	1.18245	1.99578	1.59095	1.25896	US_EQUITIES

```
In [73]: # Returns on QMJ portfolio regressed on risk factors
         table4_4
```

Out[73]:

	Country	_	Excess Return	CAPM-alpha	3-factor alpha	4-factor alpha	MKT	SMB	HML	MOM	Sharpe Ratio	IR	AdjR2
0	US_EQUITIES	coefficient	0.252649	0.381082	0.332712	0.3193	-0.098942	-0.190041	-0.268593	0.040034	0.579068	1.38507	0.711057
1	US_EQUITIES	t-stat	1.96964	3.95476	4.74715	4.22637	-5.77038	-6.01318	-4.73552	0.695137	NaN	NaN	NaN

```
In [74]: # Correlation matrix of Quality and sub-factors
         AppTable_1
```

Out[74]:

	Quality	Profitability	Growth	Safety	Country
Quality	1.000000	0.814614	0.833582	0.530603	US_EQUITIES
Profitability	0.814614	1.000000	0.716814	0.058666	US_EQUITIES
Growth	0.833582	0.716814	1.000000	0.099704	US_EQUITIES
Safety	0.530603	0.058666	0.099704	1.000000	US_EQUITIES

In [75]:
```
AppTable_2 = pd.DataFrame([MKT,SMB,HML,MOM,QMJ]).T.apply(lambda x:(1+x).cumprod()).resample('Q')
AppTable_2
```

```
/venvs/py35/lib/python3.5/site-packages/IPython/utils/dir2.py:65: FutureWarning: .resample() is now a deferred operation
use .resample(...).mean() instead of .resample(...)
  canary = getattr(obj, '_ipython_canary_method_should_not_exist_', None)
/venvs/py35/lib/python3.5/site-packages/IPython/utils/dir2.py:74: FutureWarning: .resample() is now a deferred operation
use .resample(...).mean() instead of .resample(...)
  m = getattr(obj, name, None)
```

Out[75]:

	Unnamed 0	Rexcess_m	Rexcess_m	Rexcess_m	Rexcess_m
2007-06-30 00:00:00+00:00	1.014280	0.987660	0.992668	1.009085	1.012212
2007-09-30 00:00:00+00:00	0.973015	0.981372	0.979731	1.024983	1.038083
2007-12-31 00:00:00+00:00	0.977511	0.926449	0.942477	1.075897	1.078325
2008-03-31 00:00:00+00:00	0.927881	0.872358	0.952703	1.093292	1.107210
2008-06-30 00:00:00+00:00	0.980654	0.799040	0.952474	1.102799	1.131539
2008-09-30 00:00:00+00:00	0.875620	0.810565	0.929966	1.137006	1.162635
2008-12-31 00:00:00+00:00	0.684757	0.795165	0.978031	1.195501	1.171873
2009-03-31 00:00:00+00:00	0.566479	0.728746	0.907629	1.214059	1.250921
2009-06-30 00:00:00+00:00	0.645860	0.801930	1.026623	0.973579	1.171498
2009-09-30 00:00:00+00:00	0.713644	0.876932	1.046408	0.891345	1.143545
2009-12-31 00:00:00+00:00	0.776610	0.883124	1.078842	0.847265	1.149078
2010-03-31 00:00:00+00:00	0.845181	0.878514	1.086544	0.859164	1.145336
2010-06-30 00:00:00+00:00	0.897822	0.922008	1.136112	0.874286	1.117861
2010-09-30 00:00:00+00:00	0.873683	0.885910	1.108521	0.881399	1.158198
2010-12-31 00:00:00+00:00	0.932490	0.882554	1.072287	0.891068	1.185814
2011-03-31 00:00:00+00:00	1.019041	0.914424	1.085081	0.889997	1.162648
2011-06-30 00:00:00+00:00	1.057598	0.872023	1.069624	0.912122	1.180040
2011-09-30 00:00:00+00:00	1.046931	0.830478	1.044081	0.943432	1.217524
2011-12-31 00:00:00+00:00	1.012596	0.752319	1.034133	0.957285	1.249939
2012-03-31 00:00:00+00:00	1.114708	0.745153	1.050249	0.943110	1.239877
2012-06-30 00:00:00+00:00	1.170421	0.760248	1.054466	0.954419	1.243075
2012-09-30 00:00:00+00:00	1.192910	0.760769	1.060134	0.981463	1.234326
2012-12-31 00:00:00+00:00	1.215594	0.760830	1.106035	0.982302	1.213444
2013-03-31 00:00:00+00:00	1.300223	0.759124	1.161741	0.983107	1.191650
2013-06-30 00:00:00+00:00	1.377159	0.752379	1.181078	1.014081	1.190064
2013-09-30 00:00:00+00:00	1.414517	0.798548	1.160574	1.019643	1.184935
2013-12-31 00:00:00+00:00	1.511510	0.809044	1.155828	1.033248	1.197131
2014-03-31 00:00:00+00:00	1.581470	0.857129	1.130796	1.028406	1.182213
2014-06-30 00:00:00+00:00	1.615204	0.810017	1.207228	1.008241	1.170006
2014-09-30 00:00:00+00:00	1.677757	0.781190	1.208283	1.005615	1.177336
2014-12-31 00:00:00+00:00	1.733112	0.730569	1.197765	1.042687	1.211377
2015-03-31 00:00:00+00:00	1.758058	0.720096	1.149628	1.067253	1.219729
2015-06-30 00:00:00+00:00	1.821010	0.719406	1.139712	1.063292	1.221943
2015-09-30 00:00:00+00:00	1.745216	0.693566	1.111456	1.136819	1.240896
2015-12-31 00:00:00+00:00	1.709576	0.674728	1.135864	1.166912	1.267577
2016-03-31 00:00:00+00:00	1.655669	0.649072	1.141866	1.177583	1.292595
2016-06-30 00:00:00+00:00	1.798089	0.648876	1.228784	1.114731	1.294637
2016-09-30 00:00:00+00:00	1.858111	0.651151	1.241005	1.108567	1.302343
2016-12-31 00:00:00+00:00	1.854753	0.659930	1.295795	1.108496	1.286607
2017-03-31 00:00:00+00:00	1.992407	0.662211	1.370070	1.112255	1.282171
2017-06-30 00:00:00+00:00	2.068280	0.637112	1.358446	1.134677	1.337509
2017-09-30 00:00:00+00:00	2.124888	0.615654	1.352248	1.156720	1.381056
2017-12-31 00:00:00+00:00	2.252166	0.619716	1.361015	1.162257	1.364806
2018-03-31 00:00:00+00:00	2.369765	0.611618	1.339570	1.165785	1.348135
2018-06-30 00:00:00+00:00	2.248685	0.610954	1.350379	1.183824	1.370540

In [ ]:

Alphalens tearsheet

betas.set_index('Asset')['QMJ'].to_frame(name="Beta_QMJ") df2.reset_index(level=1).join(fredData['Rf_21']).set_index('level_1', \ append=True)['Rf_21']from alphalens.utils import get_clean_factor_and_forward_returns factor_data = df2['Quality'] pricing_data = get_pricing(symbols=df2.index.levels[1], # Finds all assets that appear at least once in "factor_data" start_date=start_date, end_date=pd.Timestamp("2019-01-01"), # must be after run_pipeline()'s end date. Explained more in lesson 4 fields='open_price' # Generally, you should use open pricing. Explained more in lesson 4) merged_data = get_clean_factor_and_forward_returns(factor=factor_data, prices=pricing_data) from alphalens.tears import create_full_tear_sheet create_full_tear_sheet(merged_data)Y=data.iloc[:,0].astype(float) X=pd.Series([1]*107) pd.stats.ols.OLS(Y,X,nw_lags=6, intercept=False).t_stat # attributes .beta, .std_err, .t_stat Attributes for stats.OLS 'beta', 'df', 'df_model', 'df_resid', 'f_stat', 'p_value', 'r2', 'r2_adj', 'resid', 'rmse', 'std_err', 'summary_as_matrix', 't_stat', 'var_beta', 'x', 'y', 'y_fitted', 'y_predict'