On Prediction and Filtering of Stock Index Returns

Fredrik Hallgren

Department of Mathematics,

KTH, Stockholm, Sweden,

May, 2011

Abstract

The predictability of asset returns is a much debated and investigated subject in academia as well as in the financial services industry. In this thesis we study the predictability of the returns of European stock indices, using time series and regression based forecasting methods, as well as filtering techniques, specifically the Hodrick-Prescott filter. In disagreement with the Efficient Market Hypothesis, which claims that asset prices incorporate all information embedded in historical prices, indications of predictability based on historical returns are found. Predictability was further improved by filtering the data before applying the forecasting methods.

Acknowledgements. Many thanks to my supervisor Boualem Djehiche for the idea to this thesis and valuable input, comments and support along the way. I am also very grateful to the employees at Archipel Asset Management and Brummer & Partners for making the process of writing this thesis an invaluable learning experience and among the most fun and interesting times of my studies at KTH.

Contents

1 Introduction

The purpose of this project is in a broad sense to investigate the statistical properties of financial time series. More specifically, various methods of time series analysis will be used to attempt to forecast the future behaviour of these series, including filtering the data before applying the methods of prediction. The dataset we analyze in this paper comprise daily returns of six European stock indices - the Amsterdam Exchange Index, CAC40, DAX, IBEX, FTSE and the Swiss Market Index, for the period 1993-04-07 to 2005-12-30. It is vital to exclude a part of the full sample, to guard against data snooping in the construction of any models and make sure that the models created exhibit similar characteristics out-of-sample. Numerical analysis will be carried out in the programming language Python, using the package NumPy for numerical calculations.

If the Efficient Market Hypothesis were true, then no predictability could be found in the time series used in this study. According to the Efficient Market Hypothesis, stock indices should be modeled like a random walk, i.e. for a stock index price series $\{X_t\}$ the representation $X_t = X_{t-1} + \varepsilon_t$ should be used, where $\{\varepsilon_t\}$ is a white noise process. In this representation obviously there is no dependency across time and the best prediction of X_t is X_{t-1}. The random walk model of asset prices has been disputed though, e.g. in Lo and MacKinlay (1999), and it seems that there indeed is some predictability in stock index returns. However it is unclear whether predictability is large enough to allow for profitable exploitation above the risk-free rate, e.g. when including transaction costs. This matter is not delved into further in this study.

The report is divided into two parts. In the first part (chapters 3, 4 and 5) we use standard methods of time series analysis to predict future returns given past returns over some window, based on autocovariance and regression. The second part (Chapter 6) uses filtering methods to analyze the data and improve predictability, specifically through the Hodrick-Prescott filter.

Chapter 3 uses the univariate projection approach to calculate the best linear prediction of a time series, equivalent to the conditional expectation given previous returns. The best linear prediction is the one that minimizes the squared distance between the prediction and the outcome. Some predictability indeed seems to be present. Chapter 4 attempts to normalize the data with a view to increase predictability. Instead of predicting the raw returns, the data is normalized by subtracting each day's mean from the returns, and predicting the deviations from the mean, thereby reducing some common variability. In this way, performance of the models in chapter 3 was improved. When using multivariate prediction, normalizing the data becomes unnecessary, since the multivariate prediction implicitly calculates the prediction of deviations from a linear combination of returns. Also, the minimum-variance portfolio is created and the deviations from this portfolio are predicted, as well as the portfolio itself. A simple trading implementation is performed, to see how the predictability translates into returns of a trading portfolio. Chapter 5 investigates multivariate models of prediction, specifically multivariate regression on returns over arbitrary time periods. Predictability was somewhat improved over previous models. Chapter 6 is dedicated to the Hodrick-Prescott (HP) filter. Three ways to determine the smoothing parameter are investigated – a maximum-likelihood estimate derived in e.g. Schlicht (2004), a consistent estimator in e.g. Dermoune, Djehiche and Rahmania (2008) and a Generalized Cross-Validation estimate (see e.g. Weinert (2007)). The maximum-likelihood estimate turned out to be computationally impractical and was not used in any implementation. A regression was performed on the slope of the trend extracted by the HP filter, and the explanatory power of the HP filter turned out to be good when using the consistent estimator of the smoothing parameter. The best prediction of all models was obtained when performing a regression on both the HP filter slope and previous returns.

2 Initial data analysis

The data is made up of 2842 data points per index, comprising daily log returns.

We calculated the correlation matrix for the indices to get an idea of the dependence between them, see Table 1. As can be easily seen, the indices are heavily correlated.

Table 1: Index correlation matrix

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
AEX	1	0.8464	0.7962	0.7988	0.7637	0.7922
FCHI	...	1	0.7967	0.7918	0.7936	0.7521
FTSE	...	...	1	0.7142	0.7154	0.7325
GDAXI	...	...	...	1	0.7266	0.7282
IBEX	...	...	...	...	1	0.7027
SSMI	...	...	...	...	...	1

We also calculated the standard deviation for each index, see Table 2 below.

Table 2: Standard deviations

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
Standard deviation	0.01394	0.01372	0.01064	0.01491	0.01345	0.01184

3 Projection of returns on previous returns

The first model will make use of standard Hilbert space theory to construct a prediction based on the projection of one day's return on earlier days' returns.

Suppose we have a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. The space $L^2(\Omega, \mathcal{A}, \mathbb{P})$ is then defined as the collection of all square integrable random variables X on $(\Omega, \mathcal{A}, \mathbb{P})$, i.e. the random variables for which

$$\mathbb{E}[X^2] = \int_\Omega X^2 \mathrm{d}\mathbb{P} < +\infty$$

The space L^2 is a vector space, and

$$\langle X, \bar{Y} \rangle = \mathbb{E}[X\bar{Y}] = \int_\Omega X\bar{Y} \mathrm{d}\mathbb{P}$$

defines an inner product on L^2. Equipped with this inner product the space is complete and thus a Hilbert space.

Suppose our observations $\{x_1, x_2, ..., x_n\}$ are outcomes of random variables $\{X_1, X_2, ..., X_n\}$ belonging to L^2. The random variables are part of a stationary process $\{X_t\}_{t\in\mathbb{Z}}$, i.e. a process with constant mean and constant autocovariance function. Further assume $\gamma(h) \to 0$ as $h \to +\infty$. Then in this first model the prediction will be the projection on the linear span of earlier random variables

$$\hat{X}_{n+1} = Proj_{span\{X_1,...X_n\}} X_{n+1}$$

i.e. the element $\hat{X}_{n+1}$ in $span\{X_1, ...X_n\} = \{\phi_{n1}X_n + ... + \phi_{nn}X_1 : \bar{\phi} \in \mathbb{R}^n\}$ that minimizes the distance to this subspace.

$$\|X_{n+1} - \hat{X}_{n+1}\| = \inf_{y \in span\{X_1,...,X_n\}} \|X_{n+1} - y\|$$

where $\|X\|^2 = \mathbb{E}[X^2]$. By the orthogonal projection theorem such a smallest element exists, provided the span is a closed subspace of L^2. Note that the projection of a random variable on the space of all random variables that are functions of some random variables $X_1, ..., X_n$ equals the conditional expectation given the random variables $X_1, ..., X_n$

$$Proj_{\{Z:Z=f(X_1,...,X_n)\}} X_{n+1} = \mathbb{E}[X_{n+1}|X_1, ...X_n]$$

The difference $X_{n+1} - \hat{X}_{n+1}$ is orthogonal to the span, which gives us the projection equations

$$\langle X_{n+1} - \hat{X}_{n+1}, Y \rangle = \langle X_{n+1} - (\phi_{n1}X_n + ... + \phi_{nn}X_1), Y \rangle = 0$$

for all elements Y in the span, which is equivalent to

$$\langle X_{n+1} - (\phi_{n1}X_n + ... + \phi_{nn}X_1), X_i \rangle = 0, \quad i = 1, ..., n.$$

Hence,

$$\mathbb{E}[X_{n+1}X_i] = \mathbb{E}[(\phi_{n1}X_n + ... + \phi_{nn}X_1)X_i], \quad i = 1, ..., n,$$

or

$$\gamma(i) = \sum_{j=1}^{n} \phi_{nj}\gamma(i-j), \quad i = 1, ..., n,$$

where, $\gamma(h)$ is the covariance function. We have here assumed that we have a zero-mean process. In matrix form the above expression becomes

$$\Gamma_n \bar{\phi}_n = \bar{\gamma}_n,$$

where $(\Gamma_n)_{i,j} = \gamma(i-j)$, $i, j = 1, ..., n$ and $\gamma_n = (\gamma(1), ..., \gamma(n))'$.

3.1 Yule-Walker Estimation of an AR(p) process

A more concise way to arrive at the same results is through Yule-Walker estimation of autoregressive processes, see e.g. Brockwell and Davis (1991). Suppose that our observations are generated by a stationary zero-mean AR(p) process $\{X_t\}_t$, i.e.

$$X_t = \phi_1 X_{t-1} + ... + \phi_p X_{t-p} + Z_t, \tag{3.1}$$

where $\{Z_t\}_t$ is a white noise, i.e. a sequence of uncorrelated, zero-mean random variables with equal variances σ^2, written $\{Z_t\}_t \sim \mathrm{WN}(0, \sigma^2)$. The coefficients $\phi_1, ..., \phi_p$ are real numbers. We thus assume that each return is a linear combination of previous returns plus an uncorrelated term.

As a prediction we will use

$$\hat{X}_{t+1} = \phi_1 X_t + ... + \phi_p X_{t-p+1}, \tag{3.2}$$

since the white noise is impossible to predict, being uncorrelated with the previous observations, but will be zero on average.

To find the coefficients $\bar{\phi}$ we multiply both sides of (1) by X_{t-j}, for each $j = 1, ..., p$, and take expectations

$$\mathbb{E}[X_t X_{t-j}] = \phi_1 \mathbb{E}[X_{t-1}X_{t-j}] + ... + \phi_p \mathbb{E}[X_{t-p}X_{t-j}], \quad j = 1, ..., p$$

or

$$\mathbb{E}[X_t \bar{X}] = \bar{\phi}\mathbb{E}[\bar{X}\bar{X}'], \quad \bar{X} = (X_{t-1}, ..., X_{t-p})', \tag{3.3}$$

or in matrix form

$$\Gamma_p \bar{\phi} = \bar{\gamma}_p,$$

with $(\Gamma_p)_{i,j} = \gamma(i-j)$ and $\bar{\gamma} = (\gamma(1), ..., \gamma(p))'$.

We then estimate the autocovariances and solve the system of equations to obtain the estimated coefficients $\hat{\bar{\phi}}$.

3.2 Durbin-Levinson algorithm

To increase computational efficiency one can turn to recursive algorithms for calculation of $\bar{\phi}_n$. One such algorithm is the Durbin-Levinson algorithm. For further details see e.g. Brockwell and Davis (1991).

Initializing the algorithm with $\phi_{11} = \gamma(1)/\gamma(0)$ and $v_0 = \gamma(0)$, where $v = \mathbb{E}[(X_{n+1} - \hat{X}_{n+1})^2]$, the coefficients $\bar{\phi}_n$ are given by

$$\phi_{nn} = \left(\gamma(n) - \sum_{j=1}^{n-1} \phi_{n-1,j}\gamma(n-j) \right) / v_{n-1},$$
$$\phi_{n,j} = \phi_{n-1,j} - \phi_{nn} \cdot \phi_{n,n-j}, \quad j = 1, ..., n-1,$$
$$v_n = v_{n-1}(1 - \phi_{nn}^2).$$

3.3 Ordinary Least Squares multiple regression

The projection approach is equivalent to performing an Ordinary least squares (OLS) multiple regression. In the OLS multiple regression, one wants to explain a random variable with some other random variables, through the model

$$Y = \beta_1 X_1 + ... + \beta_n X_n + \epsilon.$$

The goal of the OLS regression is to minimize

$$\sum_{i=1}^{N} (y_i - \bar{\beta}' \bar{x}_i)^2,$$

for a set of observations $\{\bar{x}_t\}$ and $\{y_t\}$. But this is minimized exactly by the empirical estimate of the projection coefficients of X onto $\bar{Y}$. The projection in theory minimizes the norm

$$\|Y - \hat{Y}\|^2 = \|Y - \bar{\beta}'\bar{X}\|^2 = \mathbb{E}[(Y - \bar{\beta}'\bar{X})^2].$$

Furthermore, the solution to the OLS multiple regression problem is given by

$$\bar{\beta} = (X'X)^{-1}X'\bar{y},$$

where, X is the matrix of observations of the independent variables, and likewise $\bar{y}$ is a vector of observations of the dependent variable. $\frac{1}{N}X'X$ and $\frac{1}{N}X'\bar{y}$ are exactly (one of) the empirical estimates of $\mathbb{E}[X_t\bar{X}]$ and $\mathbb{E}[\bar{X}\bar{X}']$ derived above.

3.4 Implementation

The model takes in a sample of size N and the number of previous returns n used to predict the current return. Next, the autocovariance function is estimated using the estimator

$$\hat{\gamma}(h) = \frac{1}{N-h} \sum_{i=1}^{N-h} (x_i - \bar{x})(x_{i+h} - \bar{x}), \quad 0 \leq h < N,$$

where $\bar{x}$ is the sample mean. The Durbin-Levinson algorithm is used to calculate the coefficients $\bar{\phi}$.

The models will be evaluated using the *correlation mean.* Given a window, the code loops through the data, using the window backward to estimate the model, producing index predictions for the next day. This is repeated for all days in the sample. Next, the correlation between the predictions for the indices and the outcomes is calculated, and then we take the mean of all correlations.

When choosing the sample size, there is a trade-off between increased precision in the estimation of the covariance function, and validity of the assumption of stationarity. We want a large enough sample size to get accurate estimates of the autocovariance function, eliminating as much noise as possible. However the larger the time span, the autocovariance function is more likely to have changed.

Another important question is how many previous returns $X_t, X_{t-1}, ..., X_{t-n}$ to use in the prediction. Below we see that this has a large effect on the performance.

3.5 Performance

The performance of the model was generally poor, with little predictive power for the index returns. See Table 3 for a summary of the results.

Table 3: Correlation mean

	N = 50	N = 100	N = 500	N = 1000
n = 1	-0.0014	0.0163	0.0063	-0.0259
n = 2	-0.0008	-0.0022	0.0160	-0.0105
n = 3	-0.0054	-0.0020	0.0185	0.0030
n = 4	-0.0047	-0.0001	0.0171	0.0101
n = 5	-0.0041	0.0033	0.0176	0.0130
n = 6	-0.0101	0.0028	0.0209	0.0082
n = 10	-0.0018	0.0107	0.0145	0.0079

The best results are obtained for $N = 500$, with the highest correlation for lags $n = 6$.

The model is based solely on the autocovariance function, so weak values of the autocovariance between different days' returns lead to the model not being able to make very accurate predictions. This lead to the projection coefficients being close to zero and their values seemingly mostly due to noise in the calculation of the autocovariance function.

3.6 Testing statistical significance

To determine whether any of the autocovariances were statistically significant, we perform a statistical test by use of the following theorem, see e.g. Brockwell and Davis (1991) for further details.

Theorem 1. *If $\{X_t\}_t$ is the stationary process*

$$X_t - \mu = \sum_{j=-\infty}^{+\infty} \psi_j Z_{t-j}, \quad \{Z_t\}_t \sim IID(0, \sigma^2),$$

where $\sum_{j=-\infty}^{+\infty} |\psi_j| < \infty$ and $\mathbb{E}[Z_t^4] < +\infty$, then $\forall h \in \{1, 2, ...\}$ we have approximately for large N

$$\hat{\boldsymbol{\rho}}(h) \sim \mathcal{N}(\boldsymbol{\rho}(h), N^{-1}W),$$

where

$$\hat{\boldsymbol{\rho}}(h) = (\hat{\rho}(1), ..., \hat{\rho}(h)),$$

$$\boldsymbol{\rho}(h) = (\rho(1), ..., \rho(h)),$$

and W is the covariance matrix.

Under the null hypothesis that $\{X_t\}_t \sim IID(0, \sigma^2)$ then $W = I_n$ and the $\rho(i)'s$ are independent and normally distributed with variance N^{-1}. So we could reject the null hypothesis of no autocorrelation and consider an estimate of the autocorrelation statistically significant if it is outside of the interval $\pm 1.96N^{-1/2}$, at a significance level of 95%. At N = 50, 100, 500 and 1000 the corresponding intervals are $\pm 0.277, \pm 0.196, \pm 0.0877$ and ± 0.0620. Basically none of the estimates were statistically significant up to and including $N = 500$.

4 Eliminating common variability

In the previous model the time series were all considered separately, when in reality they are highly correlated. This can be taken advantage of to eliminate some of the noise that is common to the series.

This general idea could be implemented in a number of ways. One way is to construct a linear combination of the indices where the white noises in different series partly offset each other, by for example minimizing the variance. One risk though is that the linear combination also removes a predictable trend component.

4.1 Normalizing

One possibility is to subtract a day's mean return from each index and look at the new transformed indices, i.e. each day trying to predict the deviations from that day's mean across all indices.

4.1.1 Prediction of deviations from the mean

We begin by transforming the data by subtracting the mean for each day from each index. We thus get a transformed data series with deviations from the day's mean, which we can evaluate using the model above, to see if we get a better performance than before. To see how much variability was removed we computing the standard deviation of the new series.

Below the standard deviations of the normalized series

Table 4: Standard deviations of deviations from mean

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
Standard deviation	0.005235	0.005200	0.005700	0.006889	0.006469	0.006005

The transformed series are a lot less volatile, less than half the previous values in almost all cases.

The performance of the projection model was greatly improved when applying it to normalized data instead. See below a summary of results for different lags.

Table 5: Correlation mean for normalized data

	N = 50	N = 100	N = 500	N = 1000
n = 1	0.0350	0.0564	0.0333	0.0224
n = 2	0.0327	0.0513	0.0341	0.0205
n = 3	0.0320	0.0471	0.0269	0.0183
n = 4	0.0259	0.0403	0.0315	0.0145
n = 5	0.0205	0.0355	0.0375	0.0244
n = 6	0.0227	0.0350	0.0390	0.0264
n = 10	0.0228	0.0275	0.0389	0.0240

The best correlation was achieved for $N = 100$ and $n = 1$, with a correlation of 0.0564, compared to a maximum correlation of 0.0209 in the previous model. Also note that we never get a negative correlation, unlike the previous model. It is encouraging to see that at least some predictiability seems to be present in the series.

Also note that the best results are generally obtained for N = 100, instead of N = 500. This might be due to the fact that since there is less noise, a smaller sample size is enough to make the statistical characteristics appear, and the advantages of a closer to constant autocovariance function overtakes the increased noise in its estimation due to a smaller sample.

The predictive performance varies substantially among the different indices. Noteworthy is that the correlation between the series of predictions and the series of actual returns is significantly higher for the DAX index than the rest. For example with one lag and a sample size of 130 we get a correlation of 0.187, whilst for the second best index the correlation is only 0.0598.

To investigate this further I make scatter plots of the predicted versus actual returns for the case $N = 130$, $n = 1$, for the different series. Please see the six figures below. For the DAX index there are a few outliers.

I also divided the data into two subsets to test the performance in each subset (for $N = 130$, $n = 1$), to make sure that the correlation was not some earlier phenomen which since has disappeared and to see whether the performance has remained fairly constant, which would be desirable. The performane turned out to be similar, with a correlation mean of 0.0548 for the first subset and 0.0589 for the second one.

Dividing the data into four subsets, we obtain 0.0856, 0.0081, 0.0565 and 0.0422 for the first, second, third and fourth period, respectively. Note the poor result for the second period, 1996-09-09 to 1999-11-23.

4.1.2 Prediction of the mean

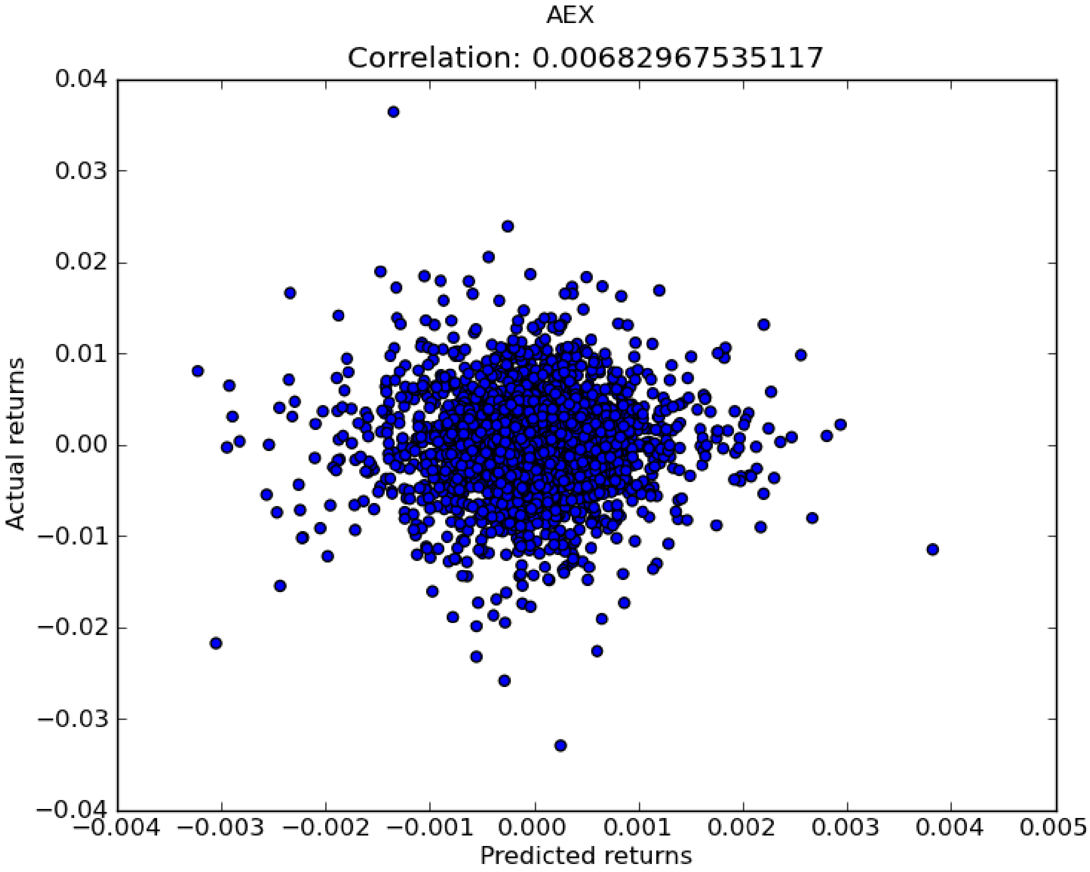
I also tried to predict the time series of the daily mean of the log index returns. The performance seems to be slightly improved as opposed to predicting the indices themselves (the actual/predicted correlation of the mean series seems slightly higher than the average of the actual/predicted series for the constituent indices). However in general the performance was poor. This indicates that the mean process contains mostly common noise, and the use of forming the mean proces lies in being able to form the deviations from it. Indeed, if there is more noise in the mean process itself, then more noise has been removed from the deviations, improving predictability for that model. The reason the performance seems to be slightly better might be due to that the idiosyncratic noise is averaged out over the indices, and the idiosyncratic noises are uncorrelated by assumption.

Table 6: Correlation actual/predicted series for the mean process

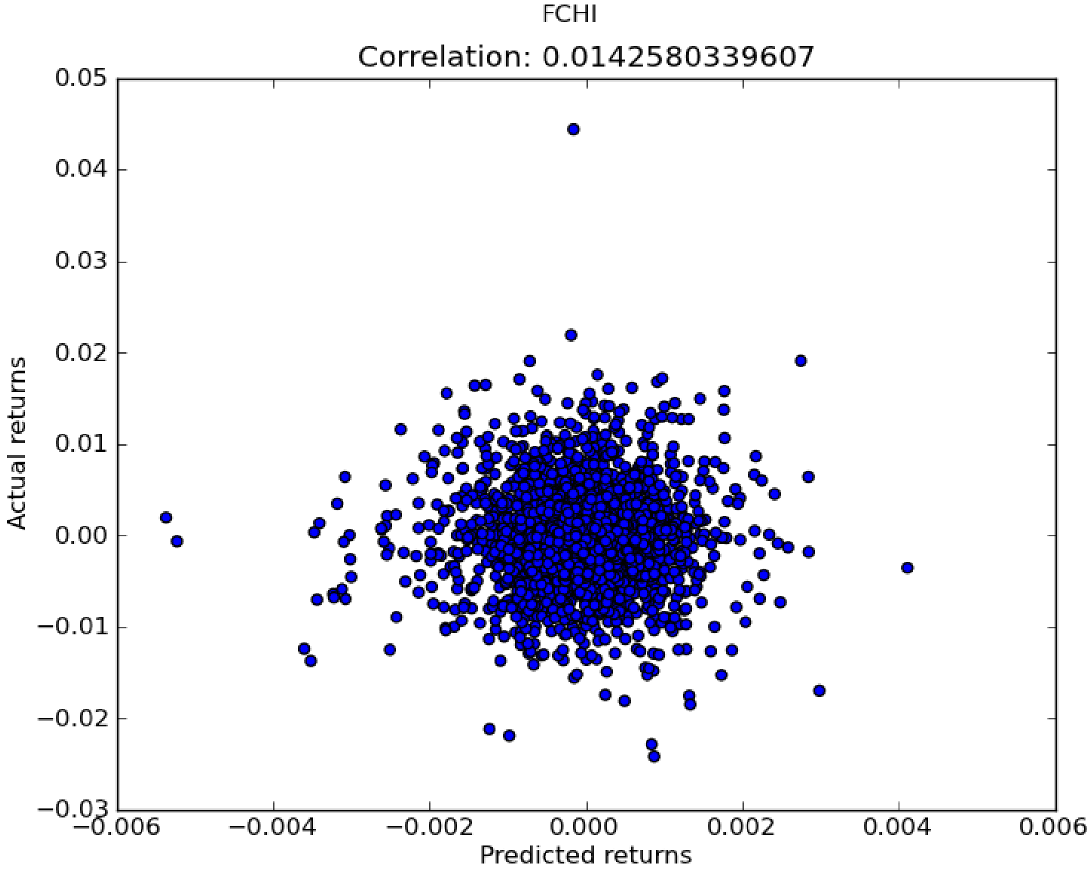
	N = 50	N = 100	N = 500	N = 1000
n = 1	-0.0130	-0.00824	0.00252	0.00516
n = 2	-0.0128	0.00063	0.0188	0.0284
n = 3	0.00061	0.0237	0.0418	0.0616
n = 4	-0.00839	0.0123	0.0288	0.0609
n = 5	-0.00724	0.00064	0.0229	0.0571
n = 6	0.00879	0.0105	0.0400	0.0661
n = 10	0.00888	0.0106	0.0419	0.0689

With data snooping, i.e. knowing which parameters yield the best results, the model performs well, however the result is quite different for different parameter values.

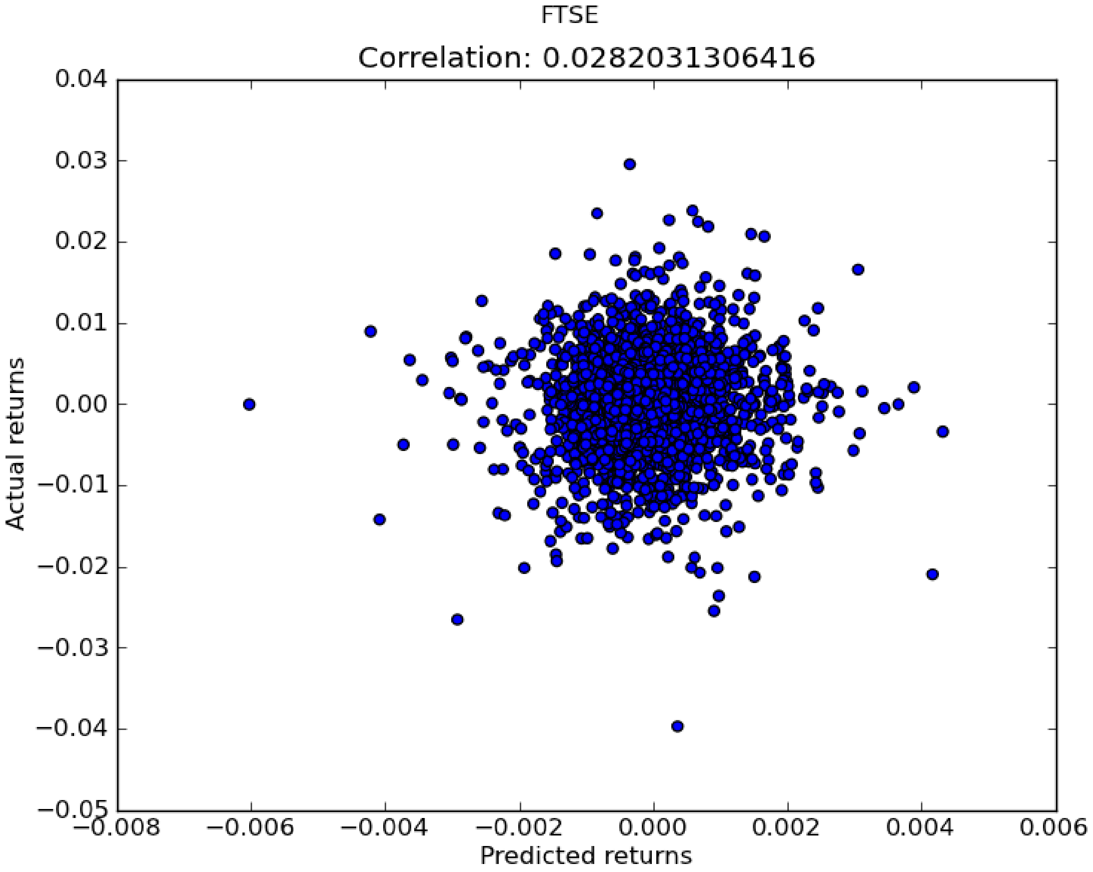
Report/Report/Plot 1 scatter.png



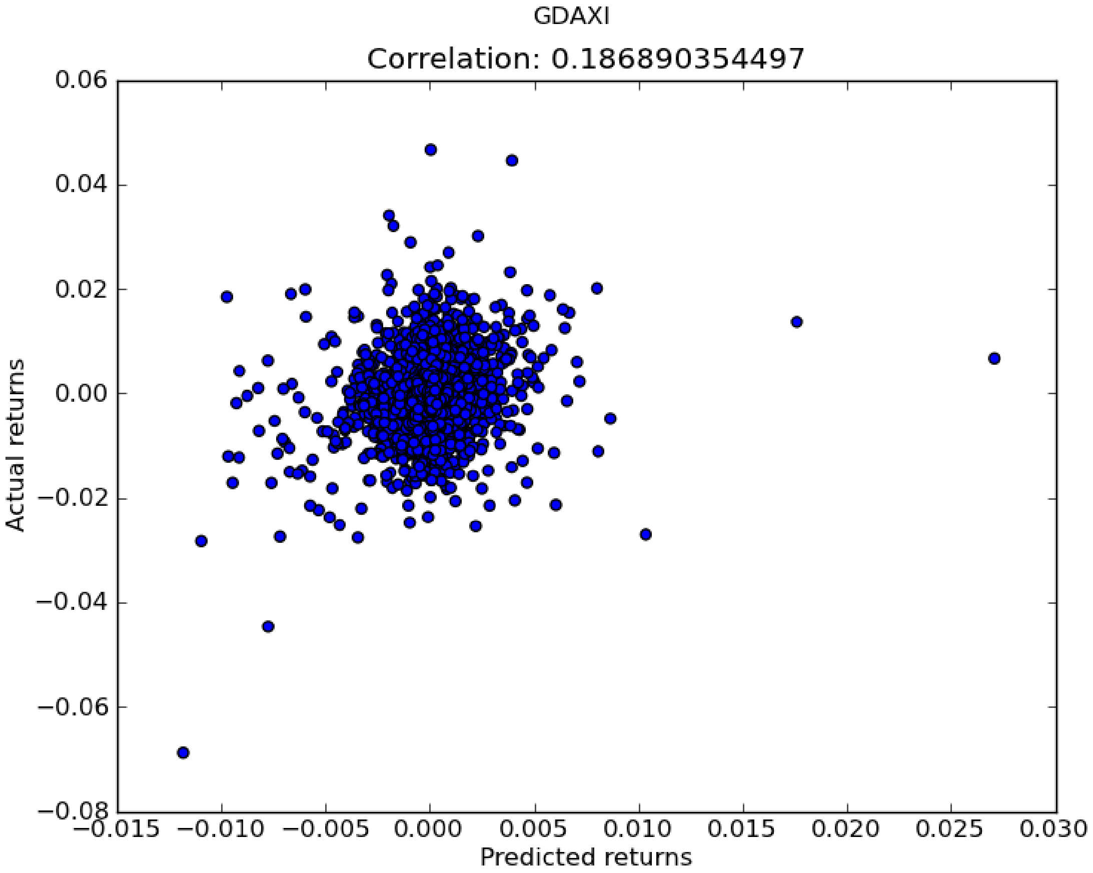
Report/Report/Plot 2 scatter.png



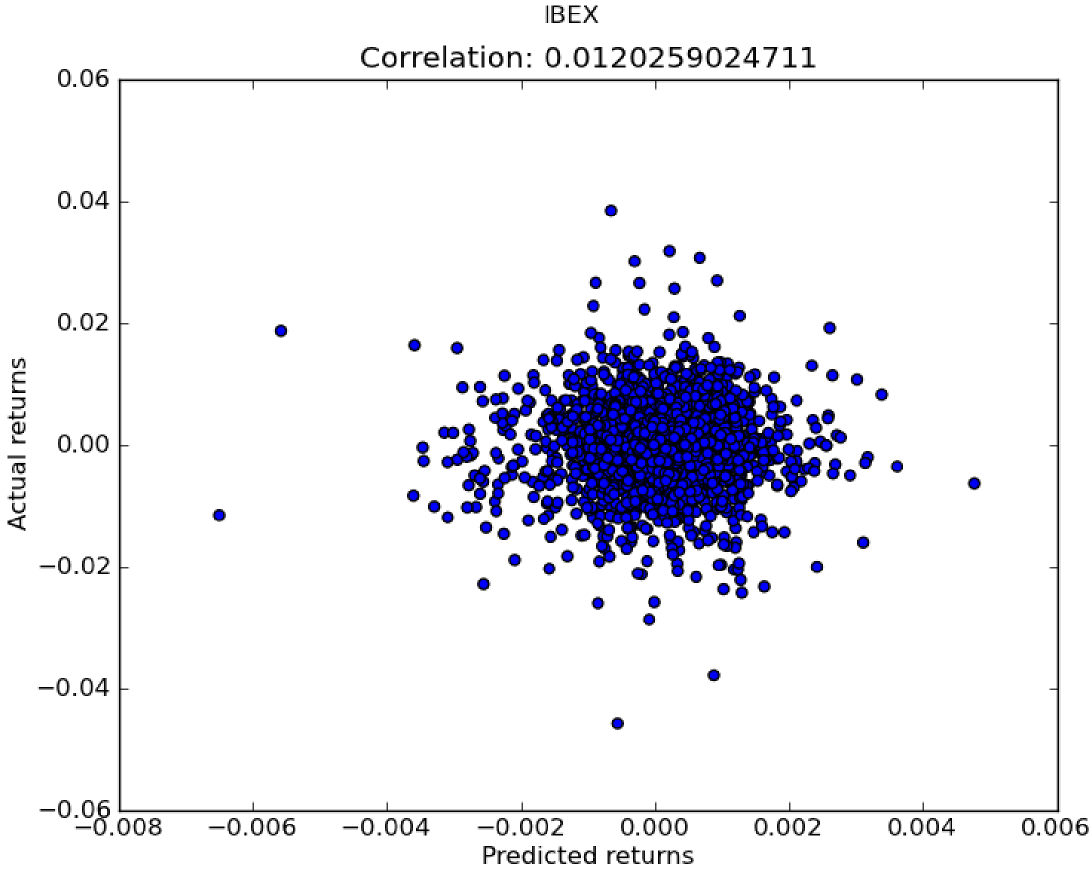
Report/Report/Plot 3 scatter.png



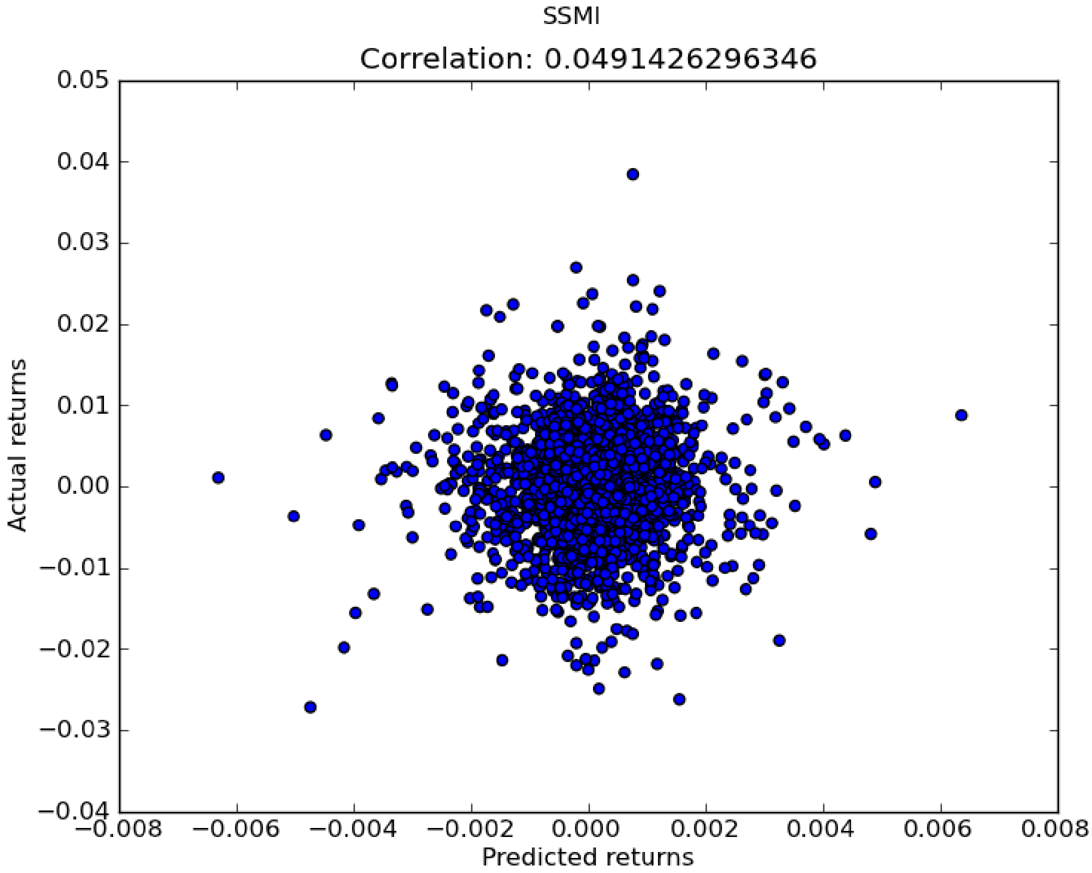
Report/Report/Plot 4 scatter.png



Report/Report/Plot 5 scatter.png



Report/Report/Plot 6 scatter.png



Note that the mechanisms generating improved predictability are different in the two cases - in the first it was thanks to elimination of common noise in the second it was supposedly due to the idiosyncratic noise averaging out.

4.1.3 A note on the prediction coefficients

In the context of the mean deviation predictions, it would be interesting to look at not only the prediction, but also the actual projection coefficients, $\hat{\hat{\phi}}$. As mentioned above, these are directly derivable from the autocovariance function, through e.g. the Durbin-Levinson algorithm.

See Table 7 for some values of the autocorrelation function. I used the last 100 returns in the estimations. For all the covariances statistical significance at 95 % is achieved if the value is outside the interval $\pm 1.96/\sqrt{N} = \pm 0.196$ (see above). Achieved by few (but indeed some) estimates, clearly there is valuable information in the estimates despite not necessarily being statistically significant.

Table 7: Autocorrelation function

	$\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$
AEX	0.2276	-0.0178	-0.0427	0.1406
FCHI	-0.0507	-0.1838	-0.0053	-0.0955
FTSE	0.0236	0.0665	-0.0055	0.0524
GDAXI	-0.1667	0.1626	-0.1775	-0.0267
IBEX	0.2483	0.1496	0.1206	0.0351
SSMI	-0.0239	0.0335	-0.0370	0.0037

The autocorrelation is quite large. Note though that some of the correlation further back might already be captured by the correlation with more recent days. For this reason it would be interesting to also look at the partial autocorrelation function. The partial autocorrelation function is also given by ϕ_{nn}, as seen in the last column in Table 8 below, showing some values of the projection coefficients, for the last 100 days, with $n = 4$.

Table 8: Projection coefficients

	ϕ_{14}	ϕ_{24}	ϕ_{34}	ϕ_{44}
AEX	0.2463	-0.0566	-0.0628	0.1644
FCHI	-0.0688	-0.2143	-0.0357	-0.1371
FTSE	0.0230	0.0629	-0.0097	0.0486
GDAXI	-0.1379	0.1306	-0.1495	-0.0973
IBEX	0.2202	0.0799	0.0732	-0.0216
SSMI	-0.0219	0.0321	-0.0355	0.0010

4.1.4 Weighting the mean

In the above model we have use the simple arithmetic mean, however increased performance might be obtained by using some weighted mean, and predicting the deviations from this mean instead, or predicting the mean itself.

We might weight by variance, standard deviation or covariance, or some other measure. The intuition behind weighting the mean is that if one index has higher variance (or covariance, or standard deviation) then it is more affected by common noise factors, and thus its deviation is better predicted by a deviation from a mean with higher weight in this index.

However it seemed difficult to achieve better performance by weighting the mean. The deviations from a mean weighted by variance or standard deviation seem to be only marginally improved in some cases, and in other cases performing worse.

4.2 Minimum-variance portfolio (MVP)

Now a similar strategy to the above one will be implemented. To remove as much noise as possible, leaving only completely idiosyncratic noise and (hopefully) some trend component, a minimum-variance portfolio will be constructed. Then we will (1) attempt to predict this series instead, and (2) attempt to predict the deviations from it.

One danger with this model is that when creating a model trying to eliminate all common noise, any predictability or trend is eliminated as well, and the only thing left is idiosyncratic noise, decreasing predictability.

I wrote a function that takes in a data series and returns a new data series that dynamically calculates the minimum-variance portfolio for each day.

I calculated the standard deviation of the minimum-variance portfolio and obtained 0.01045, which was not much lower than the constituent indices, due to the high correlation between the series. The standard deviation of the mean portfolio was 0.01174.

I also calculated the standard deviations of the deviations from the minimum-variance portfolio.

Table 9: Standard deviations of deviations from min-var series

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
Standard deviation	0.01040	0.00977	0.00469	0.01179	0.00974	0.00715

4.2.1 Prediction of deviations from the MVP

I first tried to predict the deviations from the mean-variance process. See the results above. I have used 500 past returns when calculating the weights in the minimum-variance portfolio.

Table 10: Correlation mean for prediction of deviations from minimum variance portfolio

	N = 50	N = 100	N = 500	N = 1000
n = 1	0.0182	0.0384	0.0012	-0.0022
n = 2	0.0121	0.0283	0.0090	-0.0033
n = 3	0.0144	0.0375	0.0135	0.0043
n = 4	0.0192	0.0309	0.0125	-0.0014
n = 5	0.0185	0.0283	0.0257	0.0184
n = 6	0.0236	0.0276	0.0240	0.0154
n = 10	0.0249	0.0306	0.0230	0.0138

As can be seen the performance turned out to be fairly good, especially compared to the original linpred model, however not as good as the simpler mean deviation prediction.

4.2.2 Prediction of the MVP

I also tried to predict the minimum variance series itself. See below the results. As can be seen quite a large sample size is needed for good results.

Table 11: Correlation actual/predicted for the minimum variance portfolio

	N = 50	N = 100	N = 500	N = 1000
n = 1	-0.0239	-0.0126	0.0096	-0.0459
n = 2	-0.0299	-0.0104	0.0327	-0.0083
n = 3	-0.0263	0.0025	0.0514	0.0386
n = 4	-0.0415	-0.0145	0.0386	0.0340
n = 5	-0.0468	-0.0207	0.0324	0.0321
n = 6	-0.0502	-0.0311	0.0375	0.0339
n = 10	-0.0683	-0.0441	0.0277	0.0298

We get good performance for $N = 500$ but very poor performance for $N = 50$. With data snooping (i.e. knowing the values of the parameters N, n that produce good results) the model performs well.

4.3 A trading implementation

We will now implement a simple trading strategy, as follows. We will use the mean deviations with parameter values $N = 100$ and $n = 1$, i.e. those with the best results above. For each day we will look at the predictions of the projection model and take short and long positions depending on the predictions. We will first use equal weights on all indices, with long or short positions depending on the sign of the prediction.

See below Figure 1 for the results over the entire time period.

Avergae yearly return was $\sim$ 4.6 %. An important question is how much disappears when taking into account transaction costs, since this model requires daily rebalancing.

We will now extend the model and weight by the absolute value of the prediction. However some initial tests indicate no improvement, quite the contrary. Please see Figure 2.

As can be seen absolute performance is worse, but variance is significantly reduced. Please see Table 12 for summary statistics.

Table 12: Daily mean and standard deviation

	Equal weight	Weighting by prediction
Standard deviation	0.002576	0.000615
Mean	0.000184	0.000067
Ratio	0.0713	0.1097

Indeed, return per standard deviation is actually higher in the second approach. However with transaction costs of e.g. 2 basis points, .0002, the daily mean is negative, assuming the whole position has to be rebalanced each day.

Report/Report/figptradingmeandev100-1.png

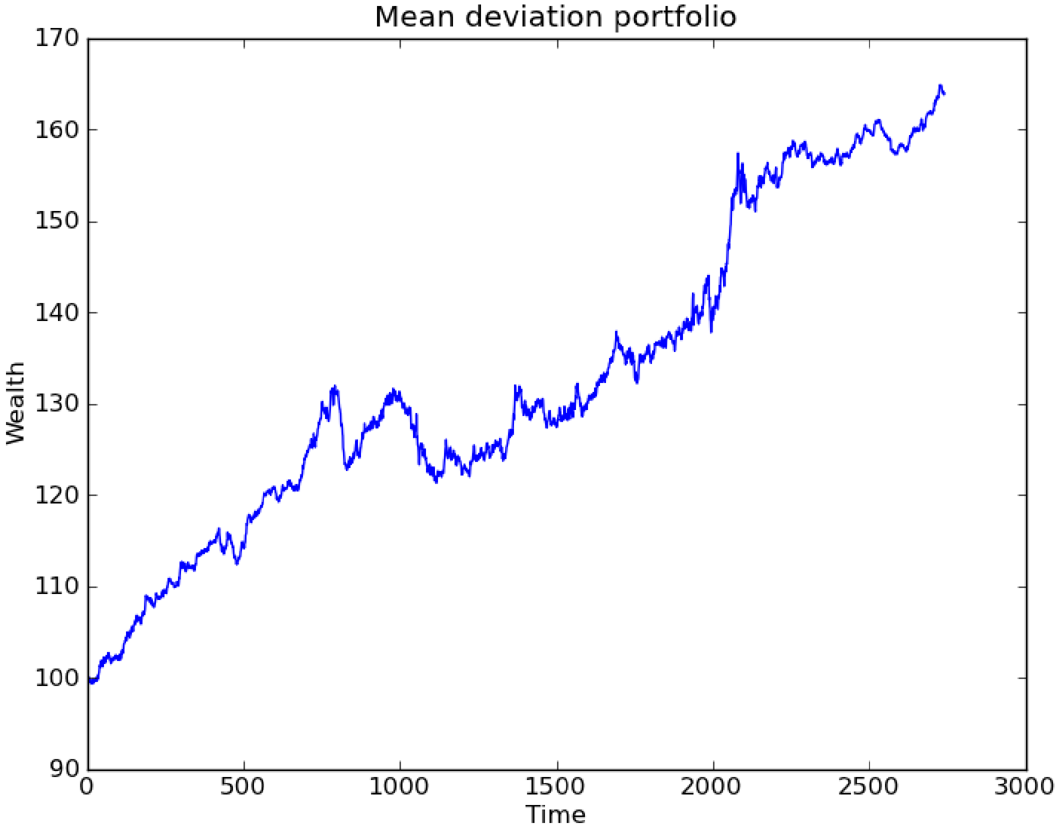


Figure 1: Equal weights

4.4 Returns over longer time periods

We will now try to predict returns over longer time periods, e.g. weekly, simply by transforming the returns. Hopefully this will lead to improved predictability, since short-term noise is averaged out and hopefully there is more momentum in returns over slightly longer time periods.

One additional difficulty when calculating returns over other time periods is the smaller number of observations available, reducing statistical accuracy in the models. Looking at data over k days reduces the available data by more than a factor $1/k$.

First we use the retutns over two days. This way, some noise will be eliminated, since on average the noise takes positive and negative values the same number of times. We also maintain a fairly large sample size. Please see Table 13 for some results for the mean deviations.

As can be seen results are fairly good, however not better than the simple one-day returns. With data snooping we obtained 0.0513 for $n = 5$, $N = 225$. However one has to keep in mind that although predictability is reduced, results in a trading implementation might be improved since the returns extend over a longer period of time.

We also look at returns over three, four, five and ten days. Please see Tables 14 to 17.

Report/Report/figptradingmeandevwt100-1.png

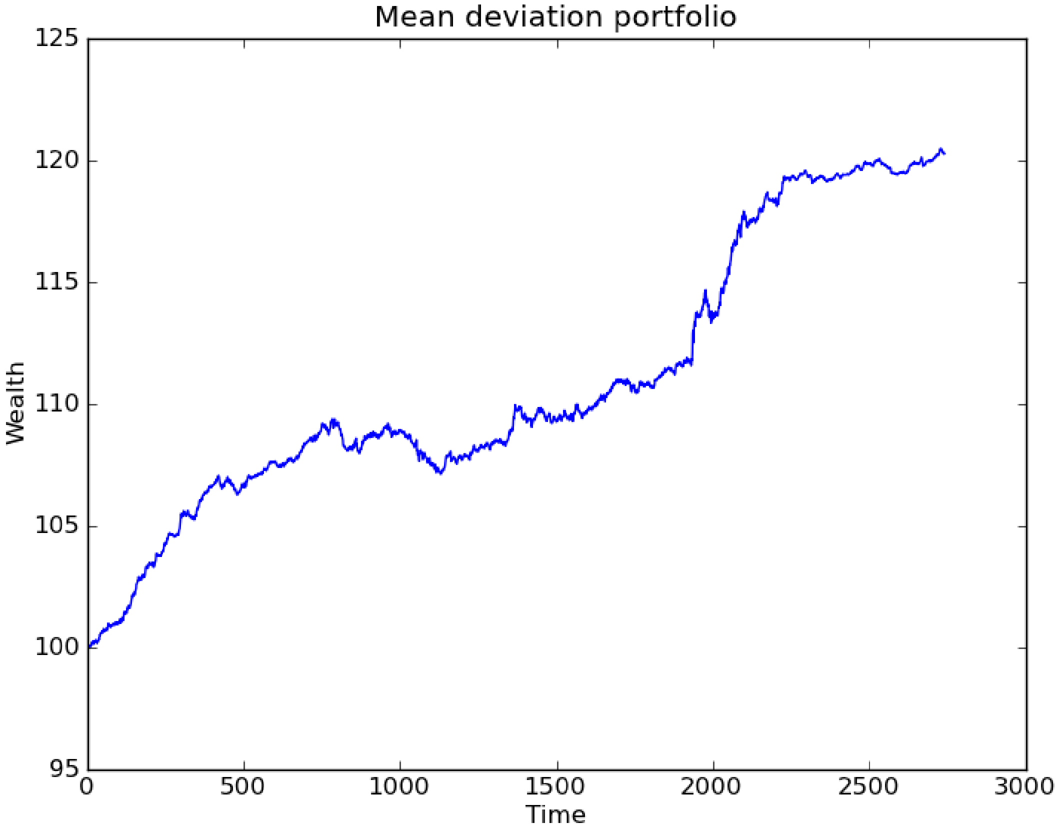


Figure 2: Weighted by prediction

Table 13: Correlation mean, r = 2

	N = 50	N = 100	N = 200	N = 500
n = 1	0.0283	0.0309	0.0280	-0.0121
n = 2	0.0227	0.0358	0.0369	-0.0006
n = 3	0.0228	0.0393	0.0406	0.0028
n = 4	0.0229	0.0412	0.0444	0.0215
n = 5	0.0278	0.0464	0.0495	0.0199
n = 6	0.0237	0.0461	0.0432	0.0056
n = 10	0.0246	0.0312	0.0423	-0.0032

Table 14: Correlation mean, r = 3

	N = 50	N = 100	N = 200	N = 500
n = 1	0.0113	0.0186	0.0156	-0.0003
n = 2	0.0288	0.0285	0.0243	0.0215
n = 3	0.0246	0.0212	0.0308	0.0259
n = 4	0.0258	0.0203	0.0305	0.0326
n = 5	0.0255	0.0211	0.0348	0.0277
n = 6	0.0418	0.0406	0.0354	0.0202
n = 7	0.0438	0.0346	0.0297	0.0171
n = 10	0.0276	0.0218	0.0273	0.0049

Table 15: Correlation mean, r = 4

	N = 50	N = 100	N = 200
n = 1	0.0039	0.0217	-0.0049
n = 2	0.0155	0.0205	0.0103
n = 3	0.0008	0.0118	0.0079
n = 4	0.0269	0.0324	0.0431
n = 5	0.0198	0.0248	0.0364
n = 6	0.0217	0.0332	0.0441
n = 7	0.0155	0.0330	0.0397
n = 10	0.0187	0.0208	0.0263

Table 16: Correlation mean, r = 5

	N = 50	N = 100	N = 200
n = 1	-0.0105	0.0200	-0.0094
n = 2	-0.0146	0.0142	-0.0181
n = 3	-0.0105	0.0221	-0.0109
n = 4	-0.0010	0.0099	-0.0109
n = 5	-0.0002	0.0032	-0.0238
n = 6	-0.0022	0.0065	-0.0270
n = 7	-0.0080	0.0121	-0.0159
n = 10	-0.0105	0.0097	-0.0271

Table 17: Correlation mean, r = 10

	N = 50	N = 100
n = 1	0.0485	-0.0347
n = 2	0.0204	-0.0378
n = 3	0.0254	-0.0507
n = 4	0.0220	-0.0532
n = 5	0.0025	-0.0576
n = 6	-0.0073	-0.0640
n = 10	-0.0068	-0.0779

5 Multivariate prediction over arbitrary time periods

Previously we have looked at the indices individually, predicting an index's value only based on the past information contained in that index. We will here use a multivariate analysis, looking at a model of the form

$$X_{t,i} = \sum_{j=1}^{p}\sum_{k=1}^{m} \beta_{j,k}^{(i)} X_{t-j,k} + \epsilon_{t,i}, \quad i = 1, ..., m,$$

where p is the number of lags and m is the number of series.

The projection model is the same as the above with $m = 1$. There are a number of ways to implement the above general model. One option is estimating a vector autoregression model (VAR). Another option is an ordinary least squares regression, or ridge regression for increased accuracy.

An extension is to regress upon time periods of different lengths, i.e.

$$X_{t-r,t}^{(j)} = \sum_{k=1}^{\kappa}\sum_{i=1}^{I} \beta_i^{(k)} X_{t_i-r_i,t_i}^{(k)} + \epsilon_{t,j}, \quad r = 0, 1, 2, ..., \quad j = 1, 2, ..., \kappa, \tag{5.1}$$

Where κ is the number of assets, $X_{\xi,\zeta}$ denotes the return from and including day ξ to day ζ, and the indices in parentheses refer to the asset. Here we have assumed identical time periods of past returns for each asset.

5.1 Multivariate-univariate mixture OLS

We will start with a mixture of a univariate and multivariate approach – a univariate approach in calculating the prediction, but making use of data from all indices in the calculation of the coefficients, using the same regression coefficients for all indices. This somewhat mitigates the problem of having less data to make use of when the previous time periods we are regressing upon increase.

The specification is as follows

$$Y_t^{(j)} = \sum_{i=1}^{I} \beta_i X_{t,i}^{(j)} + \epsilon_{t,j}, \quad j = 1, 2, ..., \kappa,$$

where Y is the one-day return we are trying to predict, and $X_{t,i}$ are the previous returns, over various time periods, that we are regressing upon. Since these returns are not intersecting, they are supposedly only weakly correlated, so we can use a normal OLS regression. This approach, then, assumes that previous returns are the same random variable regardless of the index, when estimating the beta coefficients. So for one index we can write

$$Y_t = \bar{\beta}\bar{X}_t + \epsilon_t,$$

where $\bar{X}$ is the vector of random variables we are regressing upon, and Y_t is the return random variable.

Through OLS the coefficient vector $\bar{\beta}$ is given by

$$\bar{\beta} = \mathbb{E}[\bar{\mathrm{X}}\bar{\mathrm{X}}']^{-1}\mathbb{E}[\bar{\mathrm{X}}\mathrm{Y}] = \Sigma^{-1}\mathbb{E}[\bar{\mathrm{X}}\mathrm{Y}],$$

yielding the estimate

$$\hat{\bar{\beta}} = \left(\sum_{i=1}^{N} \bar{x}_i \bar{x}_i'\right)^{-1} \left(\sum_{i=1}^{N} \bar{x}_i y_i\right).$$

We thus would need to estimate the covariances of all the variables. Another, equivalent option is to solve the system $A\bar{x} = \bar{y}$ in a least-squares sense, i.e. solving

$$\begin{aligned}
\beta_1 x_{11} + ... + \beta_n x_{n1} &= y_1 \\
\beta_1 x_{12} + ... + \beta_n x_{n2} &= y_2 \\
&\vdots \\
\beta_1 x_{1i} + ... + \beta_n x_{ni} &= y_i \\
&\vdots \\
\beta_1 x_{1N} + ... + \beta_n x_{nN} &= y_N \\
&,
\end{aligned}$$

by minimizing $\|A\bar{x} - \bar{y}\|_2$, where the x_{ij}'s and y_j's are the observations, for all indices. Recall that we act as though we only have one random variable of returns Y and $\bar{X}$, comprising the returns of all indices.

5.1.1 Performance

We begin with a regression on return periods of just one past day, which will give an interesting comparison to the projection model applied to the non-normalized returns. Since we regard the indices as one random variable, it makes more sense to use the non-normalized returns, however we might also try applying it to the mean deviations, that are still fairly correlated.

Please see below Table 18 for some results.

Table 18: Correlation mean, non-normalized

	N = 50	N = 100	N = 400	N = 500
n = 1	-0.0072	0.0016	0.0139	0.0086
n = 2	-0.0024	-0.0018	0.0146	0.0150
n = 3	0.0026	0.0029	0.0213	0.0240
n = 4	0.0021	0.0024	0.0178	0.0246
n = 5	-0.0006	0.0048	0.0123	0.0182
n = 6	0.0020	0.0069	0.0127	0.0167
n = 10	0.0102	0.0148	0.0043	0.0062

We get slightly improved performance as compared with the projection model in general, although not for all sample sizes and lags.

See below some more results for different previous return periods, with both normal log returns and mean deviations. The notation $[a, b, c]$ is to be interpreted as a regression on the returns over the last a days, and the following $b - a$ and $c - b$ days.

Table 19: Correlation mean

Periods	N = 50	N = 100	N = 400	N = 500	N = 600
			Non-normalized		
$[1, 5, 20]$	-0.0046	-0.0003	0.0146	0.0117	0.0064
$[1, 5, 20, 250]$	-	-	0.0172	0.0136	0.0190
$[1, 2, 3, 20]$	-0.0045	-0.0006	0.0156	0.0199	0.0164
$[1, 2, 3, 200]$	-	-	0.0132	0.0144	0.0198
$[1, 2, 3, 250]$	-	-	0.0133	0.0155	0.0175
$[10]$	-0.0082	0.0038	0.0094	0.0086	0.0028
$[10, 20]$	-0.0070	-0.0034	0.0126	0.0179	0.0175
$[10, 20, 30]$	-0.0047	-0.0030	0.0134	0.0195	0.0136
$[10, 20, 30, 200]$	-	-	0.0098	0.0062	0.0077
$[1, 2, 3, 10, 20, 200]$	-	-	0.0144	0.0095	0.0155
$[5, 10, 20, 30]$	-0.0016	-0.0021	0.0149	0.0153	0.0162
$[5, 10, 15, 20, 25, 30]$	-0.0060	0.0023	0.0139	0.0155	0.0152
			Normalized		
$[1]$	0.0099	0.0227	0.0102	0.0088	0.0109
$[1, 2]$	0.0118	0.0250	0.0105	0.0115	0.0138
$[1, 2, 3]$	0.0183	0.0275	0.0142	0.0188	0.0231
$[1, 2, 3, 4]$	0.0181	0.0257	0.0168	0.0239	0.0228
$[1, 2, 3, 4, 5]$	0.0135	0.0224	0.0155	0.0218	0.0217
$[1, 5, 20]$	0.0085	0.0175	0.0101	0.0131	0.0154
$[1, 5, 20, 250]$	-	-	0.0260	0.0179	0.0146
$[1, 2, 3, 20]$	0.0071	0.0258	0.0134	0.0135	0.0183
$[1, 2, 3, 200]$	-	-	0.0253	0.0215	0.0155
$[1, 2, 3, 250]$	-	-	0.0224	0.0222	0.0187
$[10]$	0.0043	0.0097	0.0077	0.0070	0.0047
$[10, 20]$	0.0061	0.0099	0.0041	0.0052	0.0086
$[10, 20, 30]$	0.0059	0.0079	0.0015	0.0061	0.0074
$[10, 20, 30, 200]$	-	-	0.0029	-0.0008	-0.0037
$[1, 2, 3, 10, 20, 200]$	-	-	0.0177	0.0109	0.0109
$[5, 10, 20, 30]$	0.0064	0.0115	0.0025	0.0035	0.0065
$[5, 10, 15, 20, 25, 30]$	0.0106	0.0193	0.0007	-0.0031	-0.0008

Note that performance was improved when adding the 250-day return period to $[1, 5, 20]$, for both normalized and non-normalized returns. However performance was not improved when adding it to the returns $[1, 2, 3]$ for the non-normalized returns.

5.1.2 Extension: predicting longer returns

Instead of predicting one day's return, the model can be applied to predicting returns over longer future time periods as well. In general, longer future return periods are preferred, yielding lower turnover and lowering transaction costs.

Performance of the model was satisfactory. As an example, with $N = 400$, $\mathit{periods} = [1, 2, 3]$ and $r = 2, 3, 4$, we got correlation means of 0.0276, 0.0312 and 0.0165 for non-normalized returns, an improvement in the first two cases as compared with predicting just one day.

5.2 Multivariate OLS regression

We will derive the OLS best estimates of the more general model above, i.e. a multivariate regression of and on returns over arbitrary time periods. We apply the model to the original non-normalized returns. In this setting, when using several indices as independent variables, there is no point in using the mean deviation returns, since the prediction is already a linear combination of the other indices, so we are implicitly predicting a deviation from a linear combination of indices.

The simplest approach, regressing upon the return of one past day, gave among the best results. However for some sample sizes performance was improved by adding additional previous return periods. See below Table 20 for a short summary of the performance, predicting the future return over one day.

Table 20: Correlation mean

Periods	N = 50	N = 100	N = 200	N = 500
$[1]$	0.0335	0.0530	0.0502	0.0535
$[1, 2]$	0.0312	0.0442	0.0522	0.0547
$[1, 2, 3]$	0.0205	0.0279	0.0366	0.0413
$[1, 2, 5]$	0.0343	0.0418	0.0541	0.0541
$[1, 2, 10]$	0.0089	0.0301	0.0423	0.0480
$[1, 20]$	0.0060	0.0343	0.0429	0.0456
$[1, 100]$	-	-	0.0142	0.0387
$[1, 250]$	-	-	-	0.0248

Performance was satisfactory, and better absolute performance than the simplest approach was obtained for both [1,2] and [1,2,5], indicating there is some merit to increasing the number of previous return periods.

6 The Hodrick-Prescott filter

The Hodrick-Prescott (HP) filter is a time series filter often applied in economics, decomposing the series into a trend component and a residual component, which may or may not contain a cyclical component.

The specification of the HP filter is the following. If $\{X_t\}$ is a time series, with available observations $\{x_t\}_{t=1,\dots,T}$, then the series is supposed to be made up of a trend component $\{\tau_t\}$ and a residual component $\{u_t\}$, such that

$$x_t = \tau_t + u_t, \tag{6.1}$$

where $\mathbb{E}[u_t] = 0$ and the trend component is the one that minimizes the following expression

$$\sum_{t=1}^{T} (x_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} ((\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1}))^2 = \tag{6.2}$$

$$= \sum_{t=1}^{T} (x_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} (\Delta^2(\tau_t))^2 \tag{6.3}$$

The second term is the squared second difference of the trend, thus penalizing a large change in growth rate of the trend. The higher the parameter λ, the smoother the trend component is forced to be. Without the second term, the trend component would simply be equal to original series $\{x_t\}$. The minimization is similar to a least squares minimization, but instead of specifying $\{\tau_t\}$ as some predetermined function, the penalization is added. As $\lambda \to \infty$, the minimization approaches ordinary least squares minimization against a linear function.

Note that the trend component can be written as

$$\tau_t = 2\tau_{t-1} - \tau_{t-2} + \epsilon_t,$$

with ϵ_t as a residual noise term. Since λ is penalizing changes in τ_t, a higher λ leads to lower variance of the residual term ϵ_t.

The HP filter is a type of Kalman filter and can be written in state-space form, with $\{x_t\}$ as the observed variable and the trend $\{\tau_t\}$ as the unobserved state variable. Recall that a time series $\{\mathbf{Y}_t\}$ is in linear state-space form if it can be written

$$\mathbf{Y}_t = \mathrm{G}_t\mathbf{X}_t + \mathbf{W}_t, \tag{6.4}$$

$$\mathbf{X}_{t+1} = \mathrm{F}_t\mathbf{X}_t + \mathbf{V}_t, \tag{6.5}$$

where $\{\mathbf{Y}_t\}$ is the observed time series, $\{\mathbf{X}_t\}$ can be interpreted as an unobserved state vector, G_t and F_t are matrices, $\mathbf{X}_1$ is a random variable, and $\mathbf{W}_t$ and $\mathbf{V}_t$ are orthogonal random "noise" vectors. The matrices are often taken to be constant. For further details see e.g. Brockwell and Davis (1991). In the state-space representation some assumptions on the initial value on the state variables are needed. In the minimization problem (6.2) no such assumptions are needed, rather they are implicit in the model.

From the defining equation for the HP filter above we can deduce the state-space representation. The observation equation (6.4) is

$$x_t = \tau_t + u_t,$$

with u_t as the noise term. The state equation (6.5) is

$$\begin{pmatrix} \tau_{t+1} \\ \tau_t \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \tau_t \\ \tau_{t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_t \\ 0 \end{pmatrix}$$

Note that ϵ_t and u_t are two different residuals, u_t is the difference between the trend and the observations, and ϵ_t is the random part in the next trend point. Furthermore, if another observation is added, in general the previous trend points will change, since the whole minimization has to be repeated.

Since the HP filter can be written in state-space form, the Kalman recursions can be used to find a prediction for the time series $\{\mathbf{Y}_t\}$ and the state equation $\{\mathbf{X}_t\}$. The prediction using the Kalman recursions works as follows. First the state equation is forecasted. Given the supplied structure of the state evolution, the prediction returned is the projection of the state variable on all observed data, i.e. the estimate that minimizes $\mathbb{E}[(\mathbf{X}_t - \hat{\mathbf{X}}_t)^2]$. The prediction of $\{\mathbf{Y}_t\}$ is then straightforward, given simply by $\hat{\mathbf{Y}}_t = \mathrm{G}_t\hat{\mathbf{X}}_t$, since $\mathbf{W}_t \perp \{\mathbf{X}_t\}$. Since $\mathrm{G}_t = [1]$ in our case, the prediction would be equal to the prediction of the trend.

However, in our case we are rather interested in the state variable itself. There would be little use to the state-space approach in predicting the series itself, given the constraints put on the state variable. Rather than estimating the matrices and coefficients, we take those as given, the noise then being a result of what is not explained by the trend. Often the main objective is rather to estimate the unknown parameters in the state-space model.

6.1 Initial analysis

We first plot the returns together with the trend calculated based on the returns, rather than prices. See Figure 3 for an example with $\lambda = 16000$.

It is hard to get any idea about the trend plotted for the returns. The trend is oscillating around zero seemingly in no predictable manner. However recall that exactly the same information is present in returns as in prices, the only information lost when transforming from prices to returns is the initial value, which is irrelevant for any trading implementation or predictability.

We next plot the trend for prices instead, see below Figure 4 for an example again with $\lambda = 16000$, zoomed in for greater visibility. In Figure 5 the smoothing parameter is $\lambda = 100000$ instead.

Now the trend is clearly visible, and indeed there seems to be some momentum in the trend. However, one has to keep in mind that each trend point is calculated using all available data, where future and past values have equal weight.

I also applied the HP filter to prices calculated from the deviations from the mean return. However in the implementation of the HP filter it is not obvious whether we should use the original or normalized returns, since there might be some trend component that is lost if we take away the mean.

We also compute a simple rolling average to compare with the HP trend. See below Figure 6 for an example with $\lambda = 16000$ and a moving average using 47 data points symmetrically. As can be seen the moving average is not as smooth as the HP filter trend.

Report/Report/figftseret16000.png

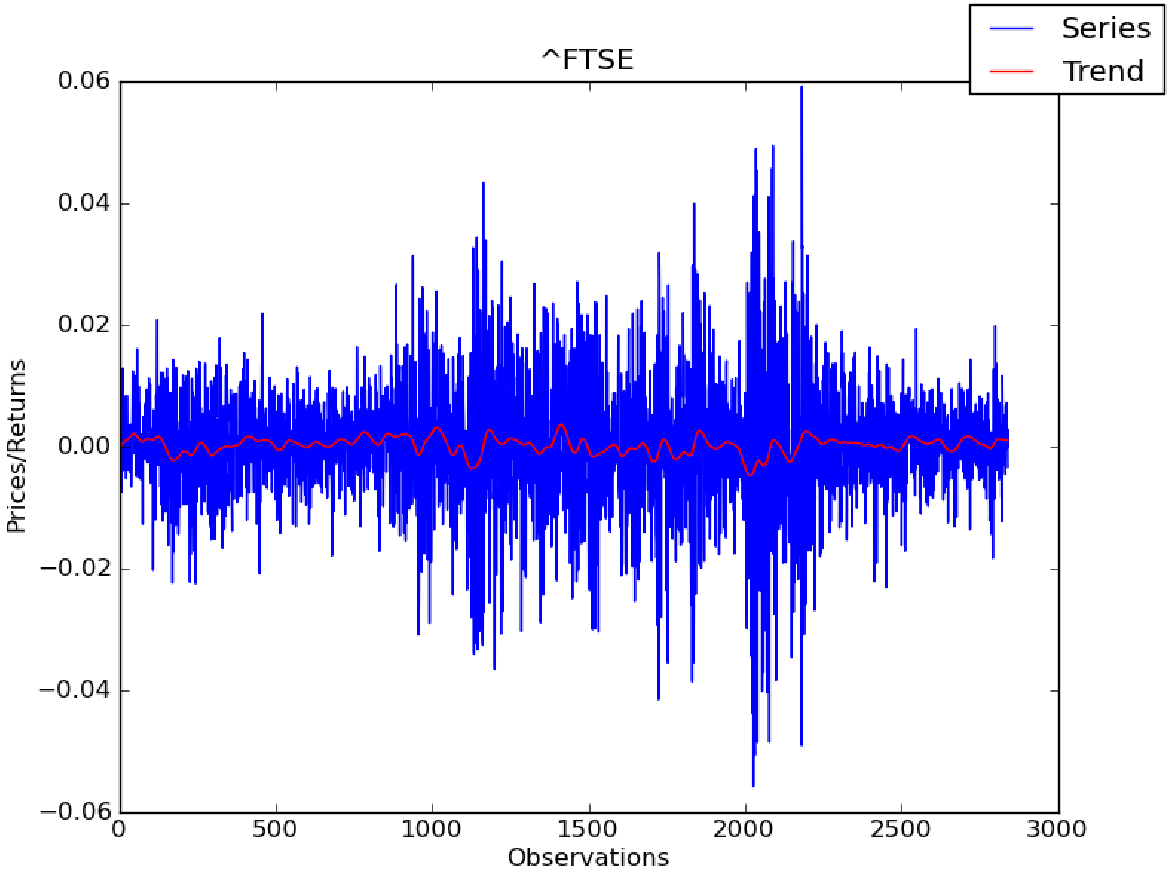


Figure 3: Returns and trend

One problem with the trend detection using the HP filter is that almost by definition, trends will appear in any time series, although being spurious. Consequently, we also apply the HP filter to a Brownian motion, where we know that any apparent trends will be purely coincidental. Please see Figure 7. Indeed, some trends seem to be present, however the trends are not as persistent and seem to be fluctuating more than with our real financial data.

6.2 Determining the smoothing parameter

The smoothing parameter λ is the only free parameter in the HP filter. Often it is determined on a rule-of-thumb basis, e.g. set to $\lambda = 1600$ for quarterly data. By changing the λ parameter one can adjust the trend component and make it reflect more short-term or long-term fluctuations. The parameter can be thought of as corresponding to the number of observations used in a moving average, thus deciding how much you want to rely on closer or more distant observations. Indeed each trend point τ_t is a linear combination of observations x_t as seen from the relation $\tau = (I + \lambda P'P)^{-1}x$ below. However there are ways to determine the parameter in a more structured manner.

6.2.1 Maximum-likelihood estimation of the smoothing parameter

A maximum-likelihood estimate of the smoothing parameter is derived in e.g. Schlicht (1994). First define the second term in the filter specification (6.2) as the disturbances

Report/Report/figfchipcs16000zm.png

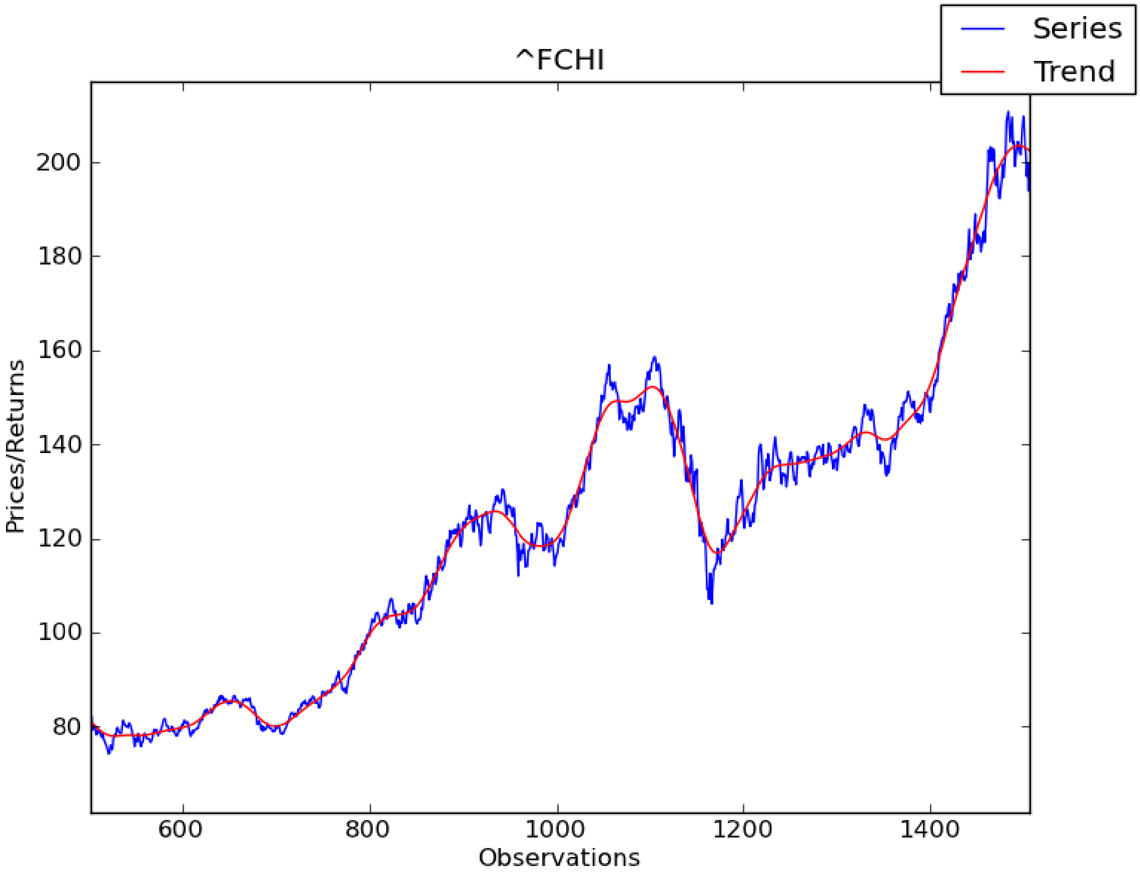


Figure 4: Prices and trend

$$v_t = ((\tau_t - \tau_{t-1}) - (\tau_{t-1} - \tau_{t-2})). \tag{6.6}$$

Writing the HP filter in matrix form, the expression to minimize is, with u_t as the residuals, v_t as the trend disturbances, τ_t as the trend and x_t as the original series

$$u'u + \lambda v'v = (x - \tau)'(x - \tau) + \lambda \tau' P' P \tau,$$

where

$$P = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ & & \vdots & & & \ddots \end{pmatrix}$$

This gives the first-order condition

$$(I_T + \lambda P' P)\tau = x,$$

which gives the unique solution

Report/Report/figfchipcs100000zm.png

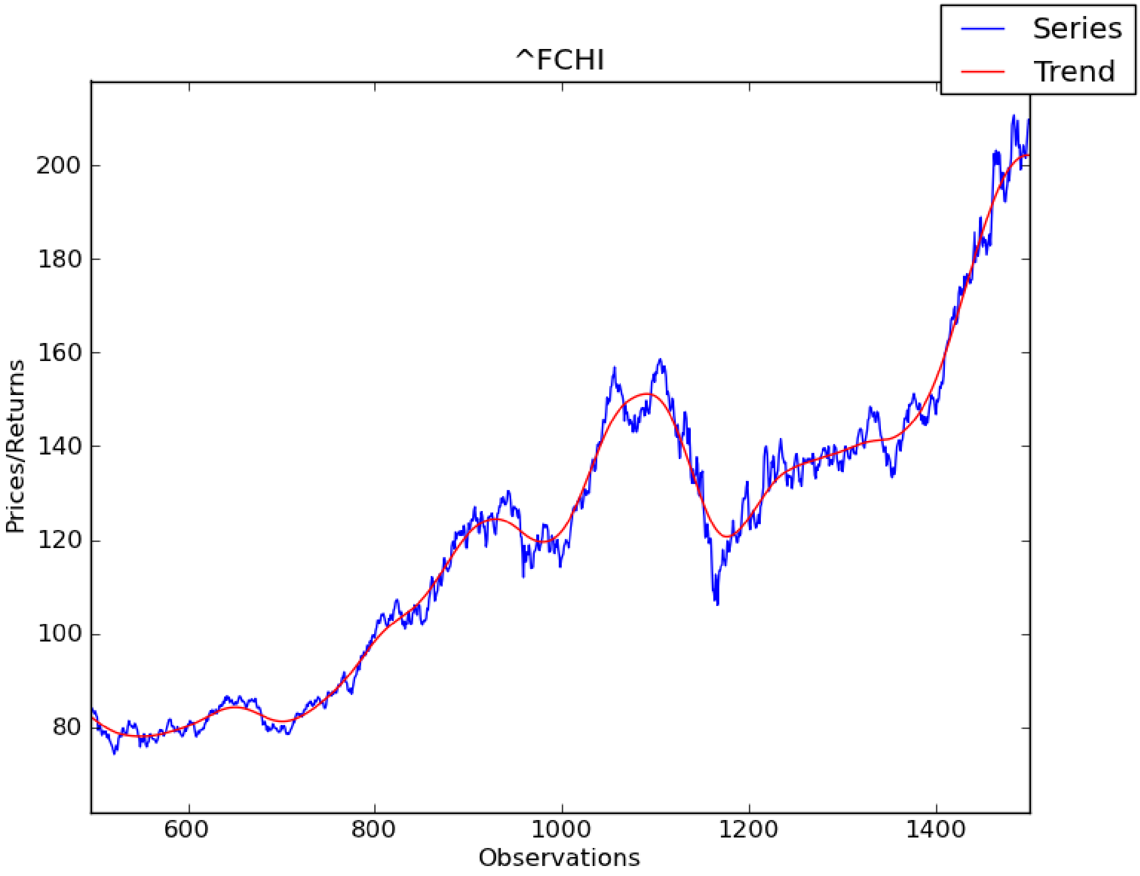


Figure 5: Prices and trend

$$\tau = (I_T + \lambda P'P)^{-1}x.$$

To determine the smoothing parameter it is assumed that $\{v_t\}$ and $\{u_t\}$ are normally distributed, iid sequences.

$$v_t \sim \mathcal{N}(0, \sigma_v^2), \quad u_t \sim \mathcal{N}(0, \sigma_u^2),$$

from where a distribution for the trend $\{\tau_t\}$ can be determined.

Given $\{v_t\}$, any solution to $v = P\tau$ can be written

$$\tau = P'(PP')^{-1}v + Z\beta,$$

where Z is a $(T \times 2)$ matrix with the two orthogonal solutions to the the equation $P\tau = 0$ as columns. Since the matrix P is of rank $T-2$ there exists two orthogonal solutions and any linear combination of these is also a solution. So given a distribution for v_t there is no unique solution for the trend.

Writing the original series as $x_t = u_t + \tau_t$ we get

$$x = u + P'(PP')^{-1}v + Z\beta.$$

Report/Report/figaexpcsmvavg16000-47zm.png

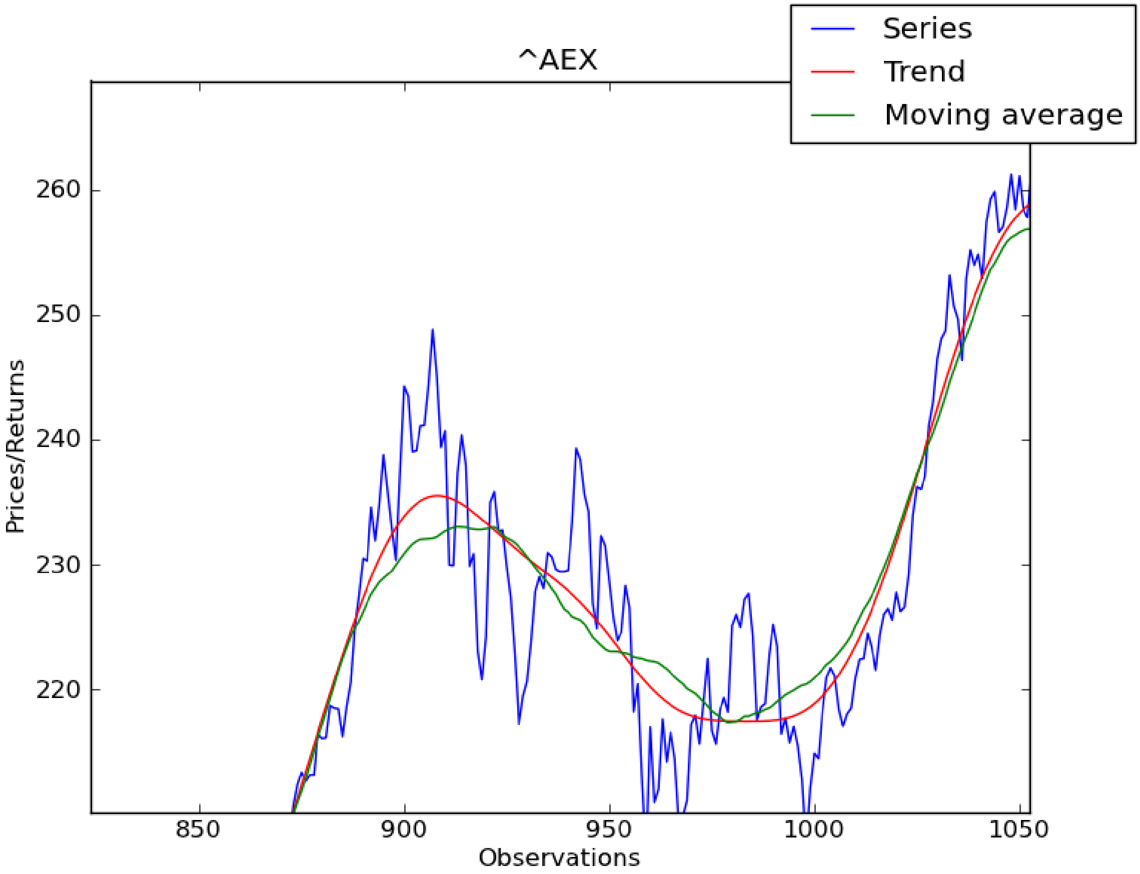


Figure 6: Prices, trend and moving average

Given the distributions for $\{v_t\}$ and $\{u_t\}$ the distribution for τ can then be determined, which depends on the parameter β. This parameter is then determined by maximizing the likelihood of the observations with respect to this parameter, yielding $\hat{\beta} = Z'x$.

Next the likelihood of x is maximized with respect to λ. The log-likelihood function becomes

$$L(x;\lambda) = -\log(\det(\lambda I_T + Q)) - T\log(\hat{u}'\hat{u} + \lambda\hat{v}'\hat{v}) + T\log(\lambda),$$

where $Q = P'(PP')^{-1}(PP')^{-1}P$. This can be simplified to

$$L(x;\lambda) = -\log(\det(I_T + \lambda P'P)) - T\log(\hat{u}'\hat{u} + \lambda\hat{v}'\hat{v}) + (T+2), \log(\lambda)$$

where $\hat{\tau} = (I_t - \lambda P'P)^{-1}x$, $\hat{u} = x - \hat{\tau}$ and $\hat{v} = P\hat{\tau}$. This likelihood function can then be maximized numerically to obtain an estimate for the smoothing parameter.

Attepmting to maximize the likelihood numerically proved difficult, the likelihood function exhibiting erratic behaviour for small sample sizes, with seemingly no global maximum, or a global maximum tending to infinity, revealed through a graphical inspection. From a sample size of about 50 a global maximum appeared, which then seemed to converge as the sample size increased. However the determinant in the likelihood function quickly approaches negative infinity, whereby very large sample sizes are not feasible. This problem is inherent in it being

Report/Report/figbrownian16000zm.png

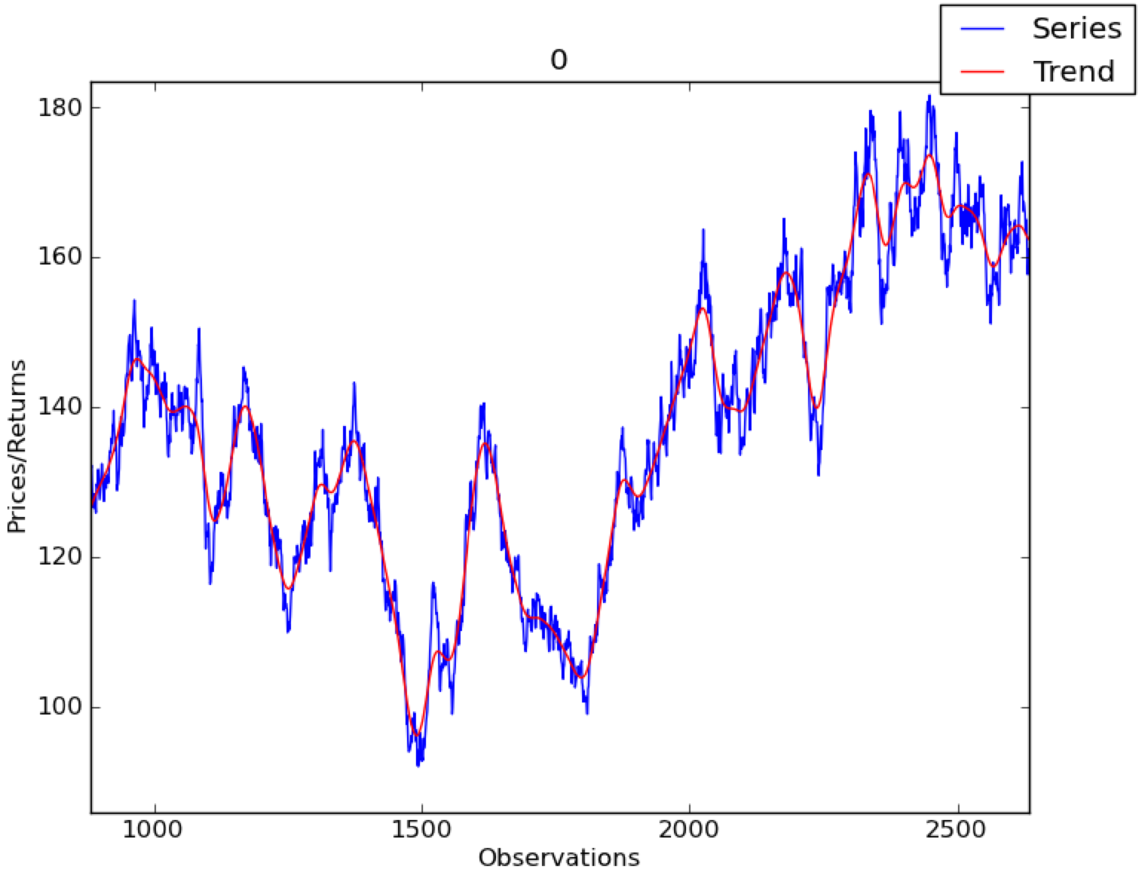


Figure 7: Brownian motion and trend

a maximum-likelihood estimation, where as the sample size of the time series increases, the probability of the actual observations quickly becomes minuscule.

Requiring the residuals $\{u_t\}$ to be a white noise sequence will supposedly lead to small values of the parameter λ, since large values would make the residuals highly correlated, since clearly if $\mathbb{P}[u_t u_{t-k} > 0] > \frac{1}{2} \Rightarrow \mathbb{E}[u_t u_{t-k}] > 0$. The last term is the correlation, since the residuals have zero mean. If they had not zero mean, the trend would not be the minimizer of the defining equation. To see this, note that

$$u = x - \tau = (I - \lambda P'P)\tau - \tau = \lambda P'P\tau,$$

with

$$P'P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ -2 & 1 & 0 & 0 & 0 & \cdots \\ 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ & & \vdots & & & \ddots \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & \cdots \\ 0 & 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 0 & 1 & -2 & \cdots \\ & & \vdots & & & \ddots \end{pmatrix} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots \\ -2 & 5 & -4 & 1 & 0 & \cdots \\ 1 & -4 & 6 & -4 & 1 & \cdots \\ 0 & 1 & -4 & 6 & -4 & \cdots \\ & & \vdots & & & \ddots \end{pmatrix}$$

Thus $\hat{\mathbb{E}}[u_t] = \bar{u}_t = \frac{1}{T}\sum_{j=1}^{T} u_j = 0$. Likewise, since $v = P\tau$, $\bar{v}_t = \frac{1}{T}(\tau_1 - \tau_2 - \tau_{T-1} + \tau_T) \to 0$ as $T \to +\infty$ or as $\lambda \to +\infty$.

6.2.2 A consistent estimator of the smoothing parameter

Another way to estimate the λ parameter is the approach derived in Dermoune, Djehiche and Rahmania (2008), which leads to a much easier implementation, henceforth called the DDR method. The smoothing parameter is determined by setting $\hat{\tau}(\lambda, x) = \mathbb{E}[\tau|x]$, following Schlicht, which leads to the smoothing parameter being a ratio of variances. Thus, given the variances σ_u^2 and σ_v^2, the optimal choice of smoothing parameter is the ratio of these variances, minimizing the mean-squared error.

In order to derive a consistent estimator of the noise-to-signal ratio λ we consider the centered series

$$Px = P\tau + Pu = v + Pu.$$

Thus $(Px)_t = v_t + u_{t+1} - 2u_t + u_{t-1}$

The transformed series is stationary, since $P\tau$ and Pu have zero mean and the variances σ_u^2 and σ_u^2 are assumed constant. Recall that in this approach it is assumed that $\mathbb{E}[u_t u_{t-k}] = 0, \forall k$, i.e. the residuals are white noise. The autocovariance function is given by

$$\gamma(h) = \begin{cases} \sigma_v^2 + 6\sigma_u^2, & \text{if } h = 0 \\ -4\sigma_u^2, & \text{if } h = 1 \\ \sigma_u^2, & \text{if } h = 2 \\ 0, & \text{otherwise} \end{cases}$$

Thus the variances can be estimated by estimating the autocovariance function of the transformed series, which will lead to an estimate of the smoothing parameter, through $\lambda = \sigma_u^2/\sigma_v^2$. This estimator is consistent by the consistency of the covariance estimator. Note that we do not need to estimate the trend. Hence the estimates for the variances become $\hat{\sigma}_u^2 = -\frac{1}{4}\gamma(1)$ and $\hat{\sigma}_v^2 = \gamma(0) + \frac{3}{2}\gamma(1)$.

Typical smoothing parameter values are around 1 when applying the method to the price series, and the resulting trend is somewhat similar to a simple moving average of 5 observations symmetrically, in terms of how much the trend is affected by each observation.

Note that if our return series are completely uncorrelated this leads to an estimated smoothing parameter of zero. This is logical, since if the prices follow a random walk there obviously cannot be any trend (or, rather, the trend coincides with the original series). The smoothing parameter tends to infinity as the first autocovariance tends to $-\frac{2}{3}$ of the variance, i.e. as $\rho(1) \to -\frac{2}{3}$, which is when $\sigma_v^2 \to 0$.

6.2.3 Determining the smoothing parameter through Generalized Cross-Validation

Cross-validation is a general technique for estimating parameters that can be applied to many different problems. It is applied specifically to the determination of the HP smoothing parameter in Weinert (2007).

In cross-validation, the data sample is first divided into different subsets. In *K-fold cross-validation*, the parameter α of a model $f(x, \alpha)$ is estimated, using some suitable estimation technique, for all but one of the partitions. In our case the function $f(\cdot)$ is the function estimating the trend, $\hat{f} : x \mapsto \tau$. Next the prediction error is calculated when predicting the left-out partition using the model fitted with the training data sets. The procedure is then repeated with each partition as a validation data set. Denoting the fitted function with the kth partition removed by $\hat{f}^{-k}(x)$, the cross-validation estimate of the prediction error is given by

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

where $\kappa : \{1, ..., N\} \mapsto \{1, ..., K\}$ is an indexing function giving the partition for each observation. Thus we are calculating the average prediction error for all points in the data sets, using the model estimated using the other partitions. $L(\cdot)$ is the prediction error. The $K = N$ case is known as *leave-one-out* cross-validation. The parameter α is finally chosen as the value that minimizes $\mathrm{CV}(\cdot)$.

Generalized Cross-Validation provides an approximation to leave-one-out cross-validation, for linear fitting methods, i.e. methods for which the estimator can be written $\hat{\mathrm{y}} = \mathrm{Sy}$, where y is the outcomes and $\hat{y}$ is the estimator. In our case, we can view the outcomes y as the original observations, and $\hat{y}$ as the trend points, then being estimations of the original series. In this sense the trend estimation is a linear fitting, since as seen before, $\hat{\tau} = (I + \lambda P'P)^{-1}x$.

For many linear fitting models the following holds,

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-i}(x_i))^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}\right)^2. \tag{6.7}$$

The GCV approximation is then

$$\mathrm{GCV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{f}(x_i)}{1 - \mathrm{trace(S)}/N}\right)^2, \tag{6.8}$$

which is useful if the trace is easier to calculate than the individual diagonal elements. In our case recall that $S = (I + \lambda P'P)^{-1}$. The smoothing parameter is then determined through minimizing the GCV score (6.8).

The GCV method for the HP filter was originally developed for smoothing splines. A smoothing spline is a function $f \in L^2$ that minimizes

$$\frac{1}{N}\sum_{j=1}^{N}(f(x_j) - x_j)^2 + \lambda\int_0^{x_N}(f''(t))^2 dt. \tag{6.9}$$

This looks like a continuous version of the HP filter, with the sum of second differences replaced by an integral of the second derivative. See e.g. Craven and Wahba (1979) for further details. The optimal λ is chosen as the one that minimizes the true mean squared error $R(\lambda)$, defined as

$$R(\lambda) = \frac{1}{N}\sum_{j=1}^{N}(g_\lambda(x_j) - g(x_j))^2, \tag{6.10}$$

where g_λ is the fitted spline and g is the true smoothing function, i.e. this is the discrepancy at the trend points. Next we define the $n \times n$ matrix $A(\lambda)$ through

$$\begin{pmatrix} g_\lambda(x_1) \\ g_\lambda(x_2) \\ \vdots \\ g_\lambda(x_N) \end{pmatrix} = A(\lambda) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

Such a matrix exists since $g_{n,\lambda}(t)$ is a linear function of $x_1, ..., x_N$ at each data point x_i. In particular, we know that such a matrix exists in the case of the HP filter, with $A(\lambda) := (I + \lambda P'P)$. The expected mean-squared error is

$$\mathbb{E}[R(\lambda)] = \mathbb{E}[\frac{1}{N}\|A(\lambda)x - g\|^2], \tag{6.11}$$

where the absence of subscripts indicate vectors. We assume the model $x_t = g_t + \varepsilon_t$, where $\{\varepsilon_t\}$ is a white noise process with variance σ^2. We thus get

$$\mathbb{E}[\frac{1}{N}\|A(\lambda)x - g\|_2^2] = \mathbb{E}[\frac{1}{N}\|A(\lambda)(g + \varepsilon) - g\|^2] = \frac{1}{N}\|(I - A(\lambda))g\| + \frac{\sigma^2}{N}\mathrm{Tr}[A^2(\lambda)],$$

where $\mathrm{Tr}[\cdot]$ is the trace of the matrix. An unbiased estimator $\hat{R}(\lambda)$ is then given by

$$\hat{R}(\lambda) = \frac{1}{N}\|(I - A(\lambda))x\| - \frac{\sigma^2}{N}\mathrm{Tr}[(I - A(\lambda))^2] + \frac{\sigma^2}{N}\mathrm{Tr}[A^2(\lambda)],$$

with g replaced by x and since $\mathbb{E}[A(\lambda)] = I$.

We will now note down the derivation of (6.7). Minimizing the GCV score minimizes $R(\lambda)$ as defined above. We start out from

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-i}(x_i))^2 = \frac{1}{N}\sum_{i=1}^{N}(g_\lambda^{-i} - x_i)^2.$$

In the setting with smoothing splines, if we replace the data point x_i with the estimate of that data point produced leaving the point out, $g_\lambda^{-i}(x^{-i}, x_i)$, and repeat the whole original minimization, then the leave-one-out estimation is reproduced, i.e.

$$g_\lambda(x^{-i}, g^{-i}(x_i)) = g_\lambda^{-i}(x). \tag{6.12}$$

The notation $g = g(\xi; \zeta)$ means the function evaluated at the point ζ estimated using the points ξ. (6.12) is seen in the following way. Suppose we have the original minimization problem (6.9) and the function solving the leave-one-out problem, g_λ^{-i}. I we then replace the point z_i with the estimate $g_\lambda^{-i}(z_i)$, then the first sum becomes the sum in the leave-one-out problem, since that point disappears, and the integral is smaller than the integral of any other function solving the original problem. Thus g_λ^{-i}, the minimizer of the leave-one-out problem, also solves the original problem with z_i replaced by $g_\lambda^{-i}(z_i)$.

Since g_λ depends linearly on the observations x, the first-order Taylor expansion of g_λ around any point x_i holds exactly. With x as the variables used in the estimation and ξ as the point of evaluation, and we are expanding about the point y_i

$$g_\lambda(x; \xi) = g_\lambda(x^{-i}, y_i; \xi) + (x_i - y_i)\frac{\partial g_\lambda}{\partial x_i}(x^{-i}, y_i; \xi) + \mathcal{O}\left(\left(\frac{\partial g_\lambda}{\partial x_j}\right)_{j \neq i}\right),$$

yielding

$$g_\lambda^{-i}(x^{-i};\xi) = g_\lambda(x^{-i}, g_\lambda^{-i}(x^{-i};\xi);\xi) = g_\lambda(x^{-i}, y_i;\xi) + (g^{-i}(x^{-i};\xi) - y_i)\frac{\partial g_\lambda}{\partial x_i}(x^{-i};\xi). \tag{6.13}$$

Recall that $g : \mathbb{R}^{N+1} \to \mathbb{R}$ is the function from the observations to the estimation of a point, $g = A(\lambda)x$. The notation above is x_i for the variables, i.e. one for each observation, depending on the value of the observations we input, and y is the point we are expanding about. This yields

$$g_\lambda^{-i} = g_\lambda(y) + (g_\lambda^{-i} - y_i)\frac{\partial g_\lambda}{\partial x_i} = g_\lambda(y) - (g_\lambda^{-i} - y_i)(1 - \frac{\partial g_\lambda}{\partial x_i}) + g_\lambda - y_i$$

$$\Rightarrow g_\lambda^{-i} - y_i = (g_\lambda(y) - y_i)/(1 - \frac{\partial g_\lambda}{\partial x_i})$$

Now, since $g_\lambda = A(\lambda)x$ and $\partial g_\lambda / \partial x_i = A_{ii}$, the ith diagonal element, we finally get

$$\frac{1}{N}\sum_{k=1}^{N}(g_\lambda^{-i} - x_k)^2 = \frac{1}{N}\sum_{k=1}^{N}\frac{(g_\lambda - x_k)^2}{(1 - A_{kk})^2},$$

which is the GCV score. Minimizing the GCV score yields an estimate for λ which minimizes the expected error derived above in (6.10). The GCV score for the HP filter is the same as above, only with $(I + \lambda P'P)$ as the matrix A. To show this, either we can repeat the above analysis with the trend τ instead of the spline estimation g_λ, or show that the spline approximation coincides with the HP filter at the points $t = 1, 2, ..., N$, for the smoothing parameters as estimated by the minimization of the GCV score.

What we would need to show to derive the HP filter GCV score is (6.12) and (6.13). However (6.13) depends only on the linearity of the matrix $A(\lambda)$, so this should hold in the case of the HP filter. We thus only need to show that the relation $g_\lambda(x^{-i}, g^{-i}(x_i)) = g_\lambda^{-i}(x^{-i})$ still holds, i.e. that the trend obtained when solving the leave-one-out problem solves the original problem with the left-out point replaced by the estimated point.

One problem though is that thet leave-one-out trend estimation τ^{-i} does not provide an unambiguous estimation of any other points left out of the sample, i.e. it is not clear what is the meaning of $\tau^{-i}(x_i)$. However, assuming it exists, we can show that the relation holds.

Consider the problem of estimating the original problem with the point x_i replaced by $\tau^{-i}(t^{-i}; x_i)$. Again, the notation $\tau(\xi;\zeta)$ indicates the function $\tau : \mathbb{R}^{N+1} \to \mathbb{R}$ obtained through the minimization using the points ξ and evaluated at the point ζ. Following the reasoning above we consider the minimization (the x_j's are the points of evaluation)

$$\frac{1}{N}[\sum_{k=1,k\neq i}^{N}[\tau^{-i}(x_k) - x_k]^2 + (\tau^{-i}(x_i) - \tau^{-i}(x_i)^2] + \lambda\sum_{k=1}^{N-2}([\tau^{-i}(x_k) - \tau^{-i}(x_{k+1})] - [\tau^{-i}(x_{k+1}) - \tau^{-i}(x_{k+2})])^2 \tag{6.14}$$

$=$

$$\frac{1}{N}\sum_{k=1,k\neq i}^{N}(\tau^{-i}(x_k)-x_k)^2+\lambda\sum_{k=1}^{N-2}((\tau^{-i}(x_k)-\tau^{-i}(x_{k+1}))-(\tau^{-i}(x_{k+1})-\tau^{-i}(x_{k+2})))^2 \quad (6.15)$$

We want to show that τ^{-i} solves the original minimization problem with x_i replaced by $\tau^{-i}(x;x_i)$. If we take the trend function to be linear between the points, then $\tau^{-i}(x_i)$ solves the N-data point minimization problem with x_i replaced by $\tau^{-i}(x_i)$, since the left sum is the same as the (N-1)-data point problem and the right sum will also have the same value. Supposedly, $\tau^{-i}(x_i)$ has to be equal to $\tau^{-i}(x_{i-1})+\frac{1}{2}(\tau^{-i}(x_{i+1})-\tau^{-i}(x_{i-1}))$ for the relation to hold, i.e. a linear interpolation. Otherwise the relation will not hold, and presumably neither the GCV relation.

Having established this, we can turn to the implementation. I used Weinert's algorithm to calculate the GCV score for the HP filter. We have used prices to calculate the GCV score.

It is important to use a constrained optimization algorithm in the implementation of the algorithm, since the estimated parameter is given by a local minimum within the allowed region of values for the smoothing parameter, and there sometimes existed a global minimum for the function outside the allowed region. Please see next section for some results.

6.2.4 Summary and evaluation

Please find below some typical values of the smoothing parameter using the three methods, applied to the most recent price series of the AEX index.

Table 21: Some estimated values of the smoothing parameter for AEX

	$\hat{\lambda}_{Schlicht}$	$\hat{\lambda}_{DDR}$	$\hat{\lambda}_{GCV}$
N = 10	-	0.939	6.754
N = 20	-	0.769	1.309
N = 30	-	1.206	1.784
N = 40	-	1.079	0.992
N = 50	2691	1.252	1.068
N = 60	29.34	1.227	0.453
N = 70	13.27	1.096	0.255
N = 80	13.95	1.018	0.212
N = 90	7.195	0.962	0.188
N = 100	9.807	0.847	0.141
N = 110	13.77	0.752	0.098
N = 120	14.05	0.788	0.101
N = 130	13.84	0.852	0.128
N = 140	19.63	0.832	0.119
N = 150	23.13	0.866	0.100
N = 200	6.136	0.787	0.099
N = 250	4.209	0.791	0.102

The maximum-likelihood estimator is quite unstable, although seemingly converging towards smaller values as the sample size grows. To get an idea of what a trend with $\lambda = 1$ and $\lambda = 10$ means in practice, we plot the series with these trends. Please see Figures 8 and 9.

To get an idea of how much the estimates of the smoothing parameter vary over different time periods, we summarize below some values for a sample size of 100 over different intervals, again

Report/Report/aexddrlambda1.png

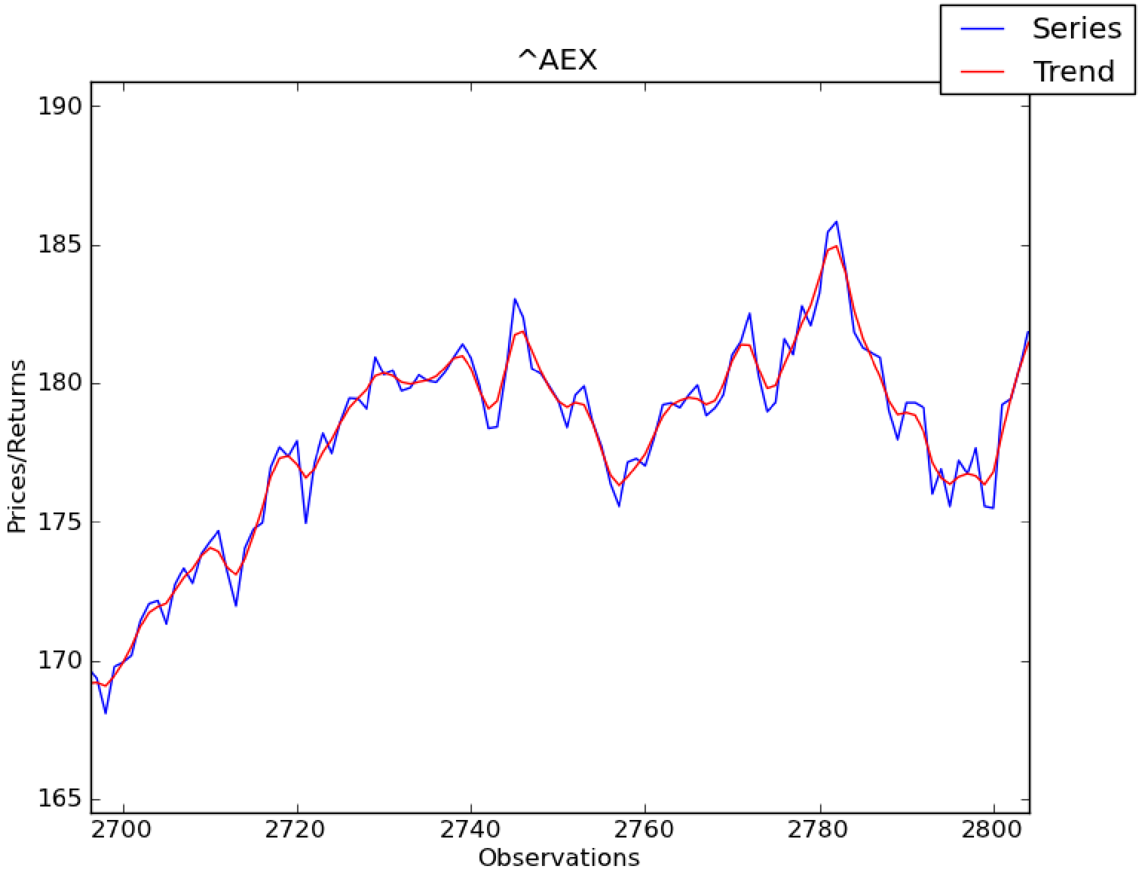


Figure 8: Prices and trend with $\lambda = 1$

for the AEX series. THe notation $[-b : -a]$ means the part of the sample from the b^{th} observation from the last date to the a^{th} observation from the last date.

Table 22: Some estimated values of the smoothing parameter for AEX, N = 100

	$\hat{\lambda}_{Schlicht}$	$\hat{\lambda}_{DDR}$	$\hat{\lambda}_{GCV}$
$[-100 :]$	9.807	0.847	0.141
$[-200 : -100]$	6.292	0.808	0.110
$[-300 : -200]$	3.353	0.776	0.068
$[-400 : -300]$	8.080	1.624	0.962
$[-500 : -400]$	4.924	1.316	0.359
$[-600 : -500]$	5.472	1.211	0.367

The DDR parameter is less varying than the Schlicht one. Recall that the DDR estimates are directly calculated from the variance and the first and second covariances. The GCV and DDR estimates are quite correlated, with a correlation coefficient of ~ 0.9.

To see how close the residuals really are to a white noise sequence, we calculate their autocovariance function for a sample size of 100, to see if this is statistically significant. We know from section 3.6 above that the autocovariance function is approximately normal with standard deviation N^{-1} under the null hypothesis of white noise. Please see Table 23 for the autocovariances. An estimate is statistically significant at the 95 % level if it is outside $\pm 1.96/\sqrt{100} = \pm 0.196$, so with $\lambda = 1$ we seem to have close to white noise, which is a typical value when using the DDR

Report/Report/aexschlichtlambda10.png

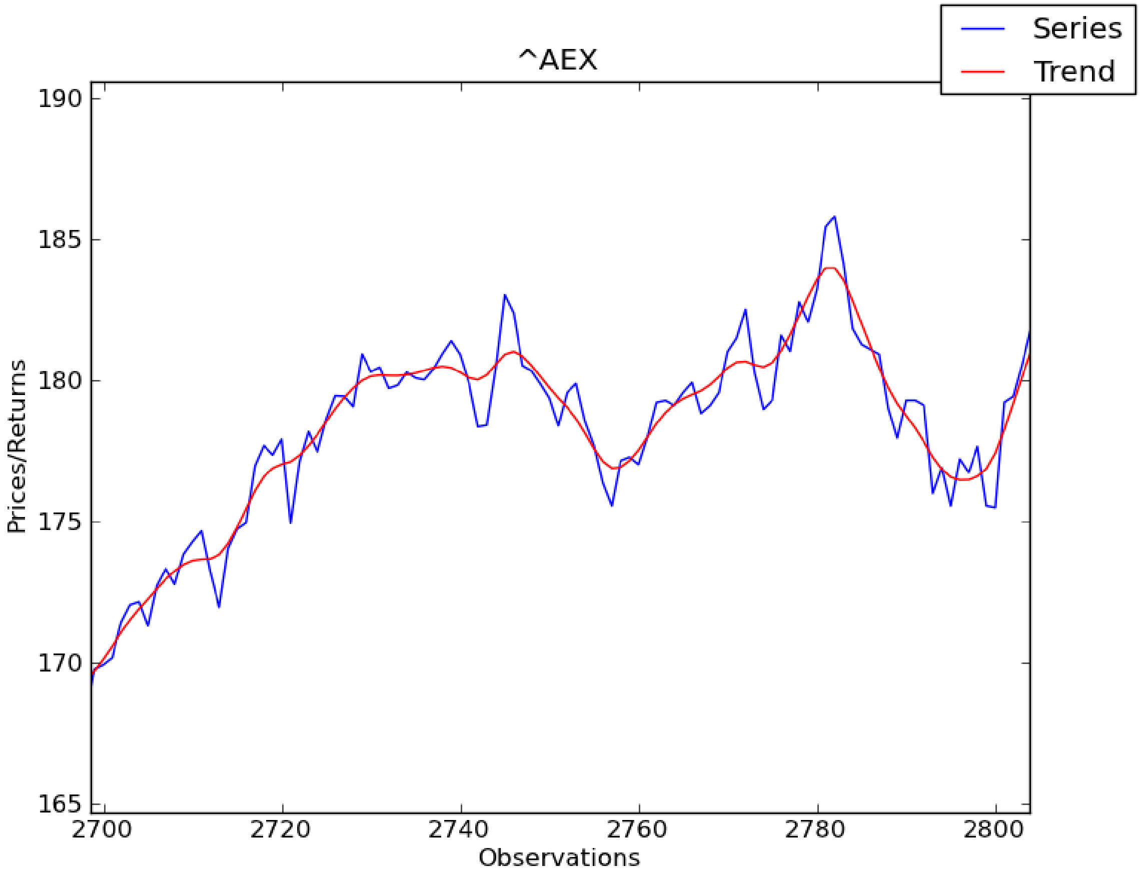


Figure 9: Prices and trend with $\lambda = 10$

method.

Table 23: Autocovariance function of the residuals, N = 100

	$\lambda = 1$	$\lambda = 10$
$\gamma(1)$	-0.0620	0.1085
$\gamma(2)$	-0.0935	-0.1885
$\gamma(3)$	-0.0475	-0.2378
$\gamma(4)$	0.0260	-0.1089
$\gamma(5)$	0.0257	-0.0261

We further investigated the characteristics of the DDR and Schlicht (maximum-likelihood) estimates by dividing our sample into subsections and estimating the parameters for all subsamples, and calculating summary statistics. See below Tables 24 to 25 for summary statistics for different partitions.

The DDR estimate has much smaller variance than the Schlicht estimate. The high variance of the Schlicht estimator for our series makes it rather unsuitable for implementation. The estimates in the Schlicht method are in general fairly low, but affected by a few very large outliers. In the AEX case, most estimates are below 10, all but one below 100, and one is around 1030.

Table 24: Summary statistics for $\hat{\lambda}_{DDR}$

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
			N = 100			
Mean	0.967	0.957	0.972	1.029	0.884	0.972
Standard deviation	0.292	0.285	0.327	0.317	0.256	0.308
			N = 200			
Mean	0.955	0.947	0.966	1.045	0.877	0.948
Standard deviation	0.233	0.220	0.289	0.255	0.223	0.212

Table 25: Summary statistics for $\hat{\lambda}_{Schlicht}$

	AEX	FCHI	FTSE	GDAXI	IBEX	SSMI
			N = 100			
Mean	47.40	13.42	15.62	14.23	46.25	6.851
Standard deviation	190.4	17.80	31.55	23.32	189.8	5.908

6.3 Regression on the HP filter trend

Having a method to determine the smoothing parameter λ, we can now proceed to making use of the trend in a prediction. First we will simply regress upon the slope of the trend, next attempt to find two trends for the same series with two different λ, and implement this in a regression.

6.3.1 Adding the trend slope as an indicator

We adapt the multivariate regression model from above, adding the slope of the HP filter trend as an additional independent variable. The trend is calculated as the difference between the last and next to last trend point. Initially, for each observation the smoothing parameter is estimated using the DDR method.

We implement the model calculating the trend slope based on prices. Performance was the best so far. See below some results in Table 27. With lookahead bias the model outperformed the original multivariate regression model model (shown in Table 26).

Table 26: Correlation mean multivariate regression

Periods	N = 100	N = 200	N = 500
$[1]$	0.0530	0.0502	0.0535
$[1, 2]$	0.0442	0.0522	0.0547
$[1, 2, 3]$	0.0279	0.0366	0.0413
$[1, 2, 5]$	0.0418	0.0541	0.0541
$[1, 2, 10]$	0.0301	0.0423	0.0480
$[1, 20]$	0.0343	0.0429	0.0456
$[1, 100]$	-	0.0142	0.0387
$[1, 250]$	-	-	0.0248

The best performance with data snooping was 0.0667, with $[1, 2, 5], N = 450$.

Table 27: Correlation mean multivariate regression with HP trend

Periods	N = 100	N = 200	N = 500
$[1]$	0.0392	0.0560	0.0606
$[1, 2]$	0.0256	0.0389	0.0526
$[1, 2, 3]$	0.0136	0.0322	0.0403
$[1, 2, 5]$	0.0314	0.0421	0.0622
$[1, 2, 10]$	0.0163	0.0370	0.0489
$[1, 20]$	0.0243	0.0431	0.0572
$[1, 100]$	-	0.0182	0.0359

6.3.2 Regression on the trend slope only

Next, we regressed the return of one index on the trend slopes of all other indices getting good results. The fact that the residuals are white noise in the specification, and that this holds true in the DDR method of estimating the parameter, was very good for predictability. Please see Table 28 below.

Table 28: Correlation mean regression on trend slope using DDR

	$lperiod = 50$	$lperiod = 60$	$lperiod = 70$	$lperiod = 80$
$N = 200$	0.0561			
$N = 300$	0.0563	0.0572	0.0583	0.0573
$N = 400$	0.0533			
$N = 500$	0.0525			

The best result was obtained for $N = 290$ and $lperiod = 70$, at 0.0584, where $lperiod$ is the window used for calculation of the HP filter. We also partitioned the sample into two subsamples and got a correlation mean of 0.0763 and 0.0556 respectively.

We also repeated the implementation using the GCV estimate. See below some results in the table below.

	$lperiod = 50$	$lperiod = 60$	$lperiod = 70$	$lperiod = 80$
$N = 150$	0.0505			
$N = 200$	0.0598	0.0577	0.0599	0.0506
$N = 300$	0.0493			
$N = 400$	0.0488			
$N = 500$	0.0453			

Table 29: Correlation mean regression on trend slope using GCV

The calculated λ parameters are generally quite small in both regressions above. Note that as λ tends to zero, the slope regression tends to the regression on the previous day's return.

6.4 Long and short trends

Another option is to add an indicator of the form

$$\beta^* X_{t-1} = \beta^* (\alpha S_{t-1} - L_{t-1}),$$

to the regression, where S_t is a short trend and L_t is a long trend, corresponding to a small and large smoothing parameter respectively. We can take the scaling constant α to be one.

I implemented this model for a number of different values for the smoothing parameter. The results were generally best for low values of the smoothing parameters. The best result was 0.0444 for $N = 200$, $\lambda_{short} = 1$, $\lambda_{long} = 2$. Some other results are shown below.

Table 30: Regression on difference between long and short trend

N	λ_{long}	λ_{short}	Correlation mean
200	2	1	0.0444
200	3	1	0.0443
200	3	2	0.0413
200	4	1	0.0433
200	4	2	0.0400
200	4	3	0.0381
200	5	1	0.0422
200	5	2	0.0391
200	6	1	0.0414
200	8	1	0.0404
500	3	2	0.0400
500	4	1	0.0413
500	5	1	0.0410
500	6	1	0.0408
500	8	1	0.0404

I also ran the same regression but rather than taking the difference, I regressed upon the constituent trends, letting the regression create a difference. However, this did not lead to improved performance.

7 Conclusion

We have in this thesis investigated different methods of time series forecasting, applied to daily returns of European stock indices. We have seen that historical prices give some predictability to future prices of different time periods, even using fairly simple statistical techniques, such as OLS regression. In order to get good predictive performance, it proved important to use information from the the other series in the prediction of an index, either through normalizing the data or through multivariate methods. We further explored the Hodrick-Prescott (HP) filter as a tool to imrpove predictability of the forecasting methods, with good results. The smoothing parameter being the only free parameter in the HP filter specification, we examined different methods for its determination, specifically a consistent estimator based on the autocovariance, a maximum-likelihood estimator and a Generalized Cross-Validation estimator. In all these estimation techniques, the residuals between the data and the trend extracted from the HP filter are modeled as white noise. Investigating the residuals obtained when calculating the trend for our data, the consistent estimator turned out to yield residuals very close to white noise. This was also the preferred method in terms of ease of implementation and performance. The maximum-likelihood estimator was unstable and exhibited high variance across different parts of our sample, so it was not deemed suitable for implementation. We ran a regression of daily index returns on the slope of the HP filter trend, and good results were obtained, especially for the consistent estimator. The GCV estiamtor yielded fairly good performance in the regression, but required a significant amount of computing power compared to the consistent estimator.

References

1. Brockwell, Peter J. and Richard A. Davis, 1991. Time Series: Theory and Methods. Springer, New York

2. Craven, Peter and Grace Wahba, 1979. Smoothing noisy data with spline functions. *Numerische Mathematik*, **31**, 377-403

3. Dermoune, Azzouz, Boualem Djehiche and Nadji Rahmania, 2008. A consistent estimator of the smoothing parameter in the Hodrick-Precott Filter. *J. Japan Statist. Soc.*, **38**, 225-241

4. Dermoune, Azzouz, Boualem Djehiche and Nadji Rahmania, 2009. Multivariate extension of the Hodrick-Prescott filter - optimality and characterization. *Studies in Nonlinear Dynamics and Control*, **13**

5. Hastie, Trevor, Robert Tibshirani and Jerome Friedman, 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York

6. Hamilton, James, 1994. Time Series Analysis. Princeton University Press, Princeton

7. Lo, Andrew W., Harry Mamaysky and Jiang Wang, 2000. The foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, **55**, 1705-1765

8. Lo, Andrew W. and A. Craig MacKinlay, 1999. A non-random walk down Wall Street. Princeton University Press, Princeton

9. Schlicht, Ekkehart, 2004. Estimating the smoothing parameter in the so-called Hodrick-Prescott filter. *IZA Discussion paper series No. 1054*

10. Weinert, Howard, 2007. Efficient computation for Whittaker-Henderson smoothing. *Computational Statistics and Data Analysis*, **52**, 959-974

11. Weinert, Howard, 2009. A fast compact algorithm for cubic spline smoothing. *Computational Statistics and Data Analysis*, **53**, 932-940