LUISS Università Guido Carli

Master's Degree Thesis in
Equity Markets and Alternative Investments

Analysis of Performance in European Mutual Funds

Supervisors
Paolo Vitale
Marco Morelli

Candidate
Vittorio Pantano
id: 713111

LUISS

Department of Business and Management
Corporate Finance
A.Y. 2019/2020

Abstract

This thesis aims to extend the current literaure regarding Mutual Funds performance persistence, which has been subject to much debate. The relevance of this purpose lies in the interest by stock market in the possibility of achieving positive returns by investing in funds that are already performing well. This research tries to assess if this is feasible analyzing an European dataset of actively managed mutual funds. After a brief presentation of the current literature about the subject, it is described how to gather a complete dataset of mutual funds monthly returns free of survivorship bias, using Python to access Refinitiv Eikon API. Then, data are analyzed differentiating for four geographical focus by performing one-factor and four-factor regression in order to verify the presence of positive performance persistence. The concluding results show no significant involvement by managers in achieving better returns.

Dedication

Mi è doveroso dedicare questo spazio del mio elaborato alle persone che mi hanno permesso di raggiungere il risultato fin qui più importante della mia vita.
In primis, un ringraziamento speciale va al mio relatore Paolo Vitale, per la sua immensa disponibilità e i suoi preziosi consigli che mi hanno consentito di concludere il mio lavoro con fiducia.
Ringrazio poi infinitamente mia madre e mio padre, che nonostante i problemi non hanno mai mancato di sostenermi e darmi tutto senza mai chiedere, appoggiando ogni mia scelta. Ringrazio poi mia sorella Viola per avermi sopportato tutta la vita, e di cui sento la mancanza ora che non viviamo più insieme.
Non posso mancare poi di ringraziare Francesca, che mi ha accompagnato e supportato per tutto questo percorso di studi fin dall'inizio, e che rappresenta una costante e imprescindibile fonte di forza e sicurezza.
Una menzione importante va infine a Lorenzo, che mi ha aiutato a imparare come scrivere in Python, e perciò ad accrescere le mie competenze.
Per concludere, voglio lasciare la parte più sentita dei miei ringraziamenti a mio nonno Riccardo, che mi è stato accanto come l'amico più prezioso fin da quando sono nato, e a cui dedico questo risultato che ho raggiunto anche grazie al suo sostegno e la sua saggezza.

Contents

List of Tables

Chapter 1

Introduction

The main goal of this thesis is to extend the current literaure regarding Mutual Funds performance persistence, which has been subject to much debate. The relevance of this purpose lies in the interest by stock market in the possibility of achieving positive returns by investing in funds that are already performing well. In particular, there are not many researches about European Mutual Funds that are actively managed, therefore I considered a dataset of just Luxembourg domicilied funds to investigate if the most widespread conclusions are reproducible. My main focus will be verifying the returns obtained by investing in past winners funds, as well as checking if these returns can be ascribable to the managers skills through a one-factor and a four-factor models.

First of all, I will describe the current literature and the most important findings; in particular, I will firstly present the evidences by Hendricks, Patel, and Zeckhauser (1990) for believing in a significant performance persistence; secondly, I will show Carhart (1997) research, one of the most important of the field, that demonstrate how persistence is mostly achieved by the chance of holding momentum-driven stocks in the portfolio.

Afterwards, I will explain how I constructed my dataset using my own code on Python and the Refinitive Eikon database by Thomson Reuters. The entire code I wrote is presented in the appendix, but in this section I will rapidly guide you through the main steps of the process and I will describe my dataset summary statistics. Finally, in chapter 4 I will show the methodology utilized for the analysis and represent the results; this part is divided into the analysis of the whole timeframe at my disposal, and subsequently its division into three different subperiods. My conclusion will confirm a not significant positive difference between past winners and past losers returns, but also no evidences of managers' direct involvement in achieving it; in fact, it will be mostly explained by exposure to momentum factor and slightly by size and value factors.

Chapter 2

Literature

Persistence in mutual fund performance has been subject to much debate in the past literature. My analysis takes as a starting point the important findings by Carhart (1997), who considers that the persistence of mutual fund performance is explainable by the chance of holding larger positions in last year's winning stocks, and thus it is not attributable to some managerial picking skills or strategies. This conclusion extend and clarify the precedent findings by other authors like Hendricks, Patel, and Zeckhauser (1990, and 1993), Goetzmann and Ibbotson (1994), Brown and Goetzmann (1995) and Wermer (1997), who claimed that short-term performance persistence is explainable by investment strategies that aims at buying past best performing stocks called 'Hot-Hands' (Hendricks, Patel, and Zeckhauser, 1990) as an expression from the sports world. Their discussion finds that mutual funds that performs well in the previous year continue to be superior in the immediate short term (four months up to two years), thus demonstrating an exploitable weakness of the efficient-markets hypothesis. Moreover, other authors like Grinblatt and Titman (1992), Elton et al. (1993) and Elton, Gruber, and Blake (1996) attribute the phenomenon to the ability of fund managers to earn abnormal returns that persist over periods of five to ten years. Carhart goes beyond these first results and identifies all the factors that plausibly allow the performance persistence, concluding that skills and strategies are not one of them.

2.1 Evidence of short-term persistence

Hendricks, Patel, and Zeckhauser (1990) analyze open-end, no-load funds performance from 1974 to 1987 applying conventional Jensen and Sharpe measures on strategies that exploit the identification of funds with 'hot hands', and find strong evidence towards their superior performance.

They assess fund performance as the α of the Capital Asset Pricing Model applied to excess returns, i.e.

$$(\mathrm{R_{it}} - \mathrm{R_{ft}}) = \alpha_{\mathrm{it}} + \beta_{\mathrm{i}}\,(\mathrm{R_{mt}} - \mathrm{R_{ft}}) + \epsilon_{\mathrm{it}}, \quad with \; \mathrm{i} = 1, \ldots, \mathrm{N}, \mathrm{t} = 1, \ldots, \mathrm{T} \tag{2.1}$$

where $(\mathrm{R_{it}} - \mathrm{R_{ft}})$ is the asset excess return over the risk-free asset return, β_i is the coefficient measuring the relation between the asset and the market portfolio excess return $(\mathrm{R_{mt}} - \mathrm{R_{ft}})$ and α_{it} represents the active return of the asset (in this case the fund performance), while ϵ_{it} are the residuals.
They formulate three hypothesis; the traditional null hypothesis states that performance is unpredictable:

$$H1: \quad \alpha_{it|t-1} = 0, \text{ for all i.} \tag{2.2}$$

where $| \; t-1$ means that information is available ex-ante. The second hypothesis is that funds can have a constant non-zero excess performance:

$$H2: \quad \alpha_{it|t-1} = \mu_i, \quad with \; \mu_i \neq 0 \text{ for some i.} \tag{2.3}$$

Finally, the third hypothesis, which is the most significant for their results, is that the excess performance is non-zero and time varying:

$$H3: \quad \alpha_{it|t-1} = \mu_i + f_i(\alpha_{t-j}; j > 0), \quad with \; f_i \neq 0 \text{ for some i and some t.} \tag{2.4}$$

It is worth noting that in the latter case we would reject H1 even if the unconditional mean μ_i is equal to 0, as long as the unconditional prediction f_i is nonzero for some t, thus accepting some funds to have hot hands during some specific periods.

Subsequently, they test the hypothesis for the excess returns of their funds sample, composed by 96 funds. While the majority of fund α's are not significantly different from 0, in testing H1 vs H2 they easily reject the joint hypothesis of zero α's in all the tests, mostly because they concentrate the analysis on growth-funds that constantly achieve positive excess return; nevertheless, they don't find a feasible investment strategy[1] that could generate significant excess returns exploiting this finding, thus making it devoid of practical consequences.

The results of testing H2 vs H3 are instead of much more interest. In fact under H3 the residuals ϵ_{it} (which would represent white noise in the CAPM formula under H2) would be serially auto-correlated, and the approximate autoregressive order is of interest in order to find a practical exploitation of this persistence. To assess this auto-correlation, they use the modified Q-statistics (see Harvey, 1990, p.211):

$$\mathrm{Q} = \mathrm{T}(\mathrm{T}+2)\sum_{\mathrm{j}=1}^{\mathrm{L}}\left[\dot{\rho}_{\mathrm{j}}^{2}/(\mathrm{T}-\mathrm{j})\right] \tag{2.5}$$

where T is the number of observations and $\dot{\rho}_{\mathrm{j}}$ is the estimated residual auto-correlation at lag j; the Q-statistics tests if all the auto-correlation of a series up to a lag L are zero. They allow a lag period of 12 in order to comprehend three years of returns, and find out that approximately one third of their dataset has a significant Q-statistics at 10% level, thus rejecting H2 in favor of H3. They also find the optimal time period for predicting future performance to be four quarters (i.e. 1 year) examining the sums of squared partial correlations:

$$q_k = T\sum_{i=1}^{N}\dot{\rho}_{(i)kk'}^{2} \tag{2.6}$$

where $\dot{\rho}_{(i)kk'}$ is the estimate of the k^{th} partial autocorrelation in the residuals of fund i; indeed they find the pattern of q_k to have lower p-values up to k = 4, indicating

[1] The possible investment strategies considered are not reported in their research

that an AR(4) process adequately approximates the time-dependence in the market model residuals.

Finally, they test different investment strategies by analyzing α persistence for different sub-period lengths and choosing a weighted mix of top performers mutual funds updated at the end of every holding period. The results for the best mix of estimation vs holding period is reported in Table 2.1, where the notation "4E 4H" indicates a 4 quarters estimation period and a 4 quarters holding period (i.e. one year each). The Table shows a quarterly average return spread of more than 5% between top and worst performer, as well as a difference of 0.4 in Sharpe's Measure[2]; moreover, the α's estimate appears to be statistically significant and indicates an excess annualized return greater than 10%, which is an exceptional finding considering that the Efficient Market Theory postulates a 0 α.

Table 2.1: **Hendricks, Patel, and Zeckhauser (1990) Performance results**

The Table shows the difference between the best and worst fund measures calculated following a 4 quarters estimation and 4 quarters holding periods. The measures described are the average quarterly return, the Sharpe's Measure[2] and the α's estimated respectively on a value weighted market index (S&P500) and a equally weighted market index benchmarks, together with their t-statistics

			Jensen's Alpha (%) Benchmarks			
			S&P 500		EWMF	
4E 4H Strategy	Mean Return	Sharpe's Measure	Value	t-statistics	Value	t-statistics
Worst Fund	-0.55	-0.05	-2.26	-2.15	-2.51	-2.76
Best Fund	4.78	0.40	3.10	2.67	2.77	2.88
Best - Worst	5.33	0.52	5.36	3.55	5.28	3.49

[2] Sharpe's Measure is the mean of the quarterly returns divided by the standard deviation of the quarterly returns

2.2 Carhart Research on Performance Persistence

The Analysis by Carhart is one of the most complete and exhaustive analysis on the subject. He starts from the Hendricks, Patel, and Zeckhauser (1993) findings and examines how the momentum factor described by Jegadeesh and Titman (1993), and not considered by the former, can explain the great part of Mutual Funds excess return. Its dataset is composed of monthly returns of 1892 different US domiciled funds comprehending inactive and merged funds and thus free of survivorship bias.

The model used for performance measurement are the standard Capital Asset Pricing Model (CAPM; Sharpe, 1964) and its own 4-factor model, built adding an extra factor accounting for one-year momentum in stock returns to the 3-factor model by Fama and French (1993); its momentum factor (i.e. $\mathrm{PR1YR}_t$) is constructed as the equal-weight average of firms with the highest 30 percent eleven-month returns lagged one month minus the equal-weight average of firms with the lowest 30 percent eleven-month returns lagged one month. The portfolios include all NYSE, Amex, and Nasdaq stocks and are re-formed monthly:

$$r_{it} = \alpha_{iT} + b_{iT}\mathrm{RMRF}_t + s_{iT}\mathrm{SMB}_t + h_{iT}\mathrm{HML}_t + p_{iT}\mathrm{PR1YR}_t + e_{it} \qquad t = 1, 2, \cdots, T \tag{2.7}$$

All the used models including the 4-factor model showed above are described in details in Chapter 4.

The core of its analysis consists of the construction of ten portfolios in which funds are categorized each January based on the decile distribution of their previous year annualized return; the portfolios are held for the subsequent year and then rebalanced. In this way he finds evidence of a clear performance persistence in the past winners portfolio with an approximate 67 basis point spread per month between top and bottom performers returns.

Afterwards, he performs the CAPM and the 4-factor model regression on these portfolios returns and analyzes its findings; the CAPM appears to not being able to explain the relative returns, since the market beta is almost identical for all portfolios. Its 4-factor model instead produces a much more valuable output because the momentum and size coefficients can better describe the differences between the portfolios: in particular the momentum factor accounts for almost half of the spread, suggesting that the winner portfolio can benefit from the performance of the top performer stocks in its assets. This being said, the alphas appear to not be significantly different from 0, highlighting the extraneousness of managerial skills from this explanation.

In addition, Carhart's analysis goes further in trying to assess if the remaining part of unexplained persistence can be associated with specific funds characteristics. He finds that expense ratios and portfolio turnover alone cannot explain much of the remaining spread between the performance of low and high decile portfolios pointed out from the regression, but at least he demonstrates that the losers portfolios have above average expenses, suggesting that this could negatively contribute to its low performance. The next analytical excercise he performs consists of estimating the effect of these fund characteristics on the individual fund performance. In order to do this, he calculates the single fund alphas from the 4-factor model, and estimate a supplementary regression using expense ratio, and turnover in turns as explanatory variables. The outputs of this complex process suggests a strong relation between performance and these fund properties. In particular, he can confirm that on average funds do not recoup their transaction costs through higher returns, another evidence against a positive contribution of management decisions.

The final part of Carhart research investigates consistency in fund ranking through a contingency table of the evolution of fund decile membership through the years. This lead to the trivial conclusion that winners are more likely to remain winners and losers are more likely to remain losers or perish; however, the interesting finding is that fund composition of the top decile differs substantially each year with a turnover of about 80%. In addition, previous year winners frequently become next year losers. Finally, Carhart extends the estimation period for the decile portfolios construction to two, three, four and five years in order to find if the degree of explanation for the spread in mean returns between portfolios ranks is affected. He concludes that a larger estimation period clearly reduces this spread, but its absolute unexplained quantity remains unchanged; this means that the 4-factor model is practically unable to explain the differences in returns after an extension to 3 years estimation period, and the remaining spread is attributable only to expense ratios (for about 1/3) and mostly to unidentifiable variables

Chapter 3

Data

Data I analyze in this study comprehends monthly returns of Luxembourg domiciled mutual funds, since Luxembourg is home for the majority of funds operating in the European Area; these are divided in 4 subsets by geographical focus: Global, Europe, USA and Emerging Markets. The historical period researched goes from 1962 to 2020, but available data starts from 1970 and varies among subsets, thus is better explained in Table 3.1.

I constructed my dataset with Lipper provided data accessed from the Refinitive Eikon web platform by Thomson Reuters, using the advanced fund screener tool, which helped me filtering for Primary-Funds (thus including just aggregated funds by share classes) and for different geographical focus mentioned above. Furthermore, I included only common equity mutual funds and excluded sectoral, hedged and short funds, as well as pure index funds, ETFs and ETNs since the research aims to find the effects of actively managed strategies.Survivorship-bias is prevented by including in the dataset also inactive (or "dead") Funds.

Another important aspect is that I computed fund returns from reported monthly Net Asset Value: this has been necessary not only because returns data were heterogeneous and unavailable for many funds, especially for inactive funds, but also because thereby my dataset is free of dividends, interest and capital gains distributions paid out to investors unless reinvested.

Table 3.1: **Summary Statistics of Mutual funds**

The summury of funds is presented dividing them into each dataset by geographical focus. A further differentiation concerns funds that are still operating and funds that ceased operations or have been merged. The proportion of "dead" funds is about 1/3 for each subset. Furthermore, the starting observation is different due to data unavailability: 1970 for Global, 1983 for Europe, 1972 for US and 1986 for Emerging Markets; then, the datasets have been cleaned and standardized so that all have at least 10 active funds at first date, thus the final timeframe obtained goes from January 1990 to November 2020

	Total Number			Average Number			Average Age (years)		
Group	Live	Dead	**Total**	Live	Dead	**Total**	Live	Dead	**Total**
All funds	2381	1245	**3626**	536.9	134.9	**808**	41.6	19.5	**40.6**
By Geographical focus									
Global	1305	651	**1956**	230.5	59.7	**350.2**	9.0	4.7	**9.1**
Europe	536	310	**846**	154.8	41.4	**237.8**	10.9	5.0	**10.6**
US	284	191	**475**	69.6	18.2	**109.6**	11.5	4.4	**10.8**
Emerging Markets	256	93	**349**	82	15.6	**110.4**	10.2	5.4	**10.1**

3.1 Retrieving Funds RIC Codes

The data gathering and analysis are performed through Python 3, using the Jupyter Notebook in the Anaconda environment, since this is the most efficient way of taking advantage of the Refinitive Eikon API. I will omit the installation procedure of Anaconda and all the packages necessary to carry out the analysis since it is not in the purpose of this research; however, I suggest to browse into the official documentation that is simple and straightforward[1].

As mentioned above, the only step carried out of the python environment is the fund selection, a procedure that can be easily performed in the fund screener section of the Eikon DataStream desktop or web platform The first issue encountered is the irregular identity of inactive funds inside the database: all funds are indeed recognized by RIC codes which are needed for accessing data through the program API in the Python environment; however, 'dead' funds have this code modified by a suffix and this change is not explicit in the screener, where funds are still identified by their old RIC code even if inactive, thus downloading directly from there is not

[1] https://docs.anaconda.com/anaconda/user-guide/getting-started/

suitable. Fortunately, I found out that starting from the funds’ LipperIDs (the identity code by Lipper, a leading provider of independent fund content, analytics, and insight) it is possible to retrieve the correct RIC code using a Python function of the Eikon module. Therefore I trivially downloaded the LipperIDs as csv from the search panel and used this code in order to get the actual RIC.

```python
def get_ric(lipper_list, geo = 'europe'):
'''
Retrives ric codes from a list of lipperIDs
'''
ric_list = pd.read_csv('data/primary_funds/ric_'+geo+'.csv')
ric_list = ric_list.T.iloc[1].to_list()
error_list = []
for i in tqdm(lipper_list[len(ric_list):]):
  try:
    ric, none = ek.get_data([str(i)], 'TR.RIC')
    ric_list.append(ric.iloc[0,1])
  except:
    error_list.append(i)
    break
pd.Series(ric_list).to_csv('data/primary_funds/ric_'+geo+'.csv')
return ric_list, error_list
```

The code above simply takes a list of LipperIDs and request the respective RIC from the database, outputting the results and a list of errors and finally saving them as a csv document. Note that I designed the argument ”geo” to be able to change the list directly from the function.

3.2 Gathering of Mutual Funds Returns

The second step represents the real data gathering process to retrieve the mutual funds returns. Simply said, the Eikon API uses the same get.data command to access numerous different characteristics of a fund indicated by its RIC code, such as the Net Asset Value (NAV), Price change and Total Expense Ratio (TER). Despite the relatively straightforward concept, in practice I found several issues:

- First of all, the availability of data is not the same for every fund. Therefore, I needed to attempt several alternative routes in order to find the right category to inquire for the completeness of the research. Eventually, I found the NAV to be the only available for both live and dead funds.

- Secondly, whenever the process runs into an error, the whole code stops and all data are lost. This happens not only when data is unavailable for a certian RIC, but also when the latency from Eikon servers lead to a request timeout. I solved this issue by saving the data retrived in a csv at the end of the function and setting the number of lines already inquired as the new starting point when the function is run again.

- Finally, the output of get.data on the monthly NAV of a fund is a single line of text divided by semicommas, where the first section includes all the dates, the second includes the actual NAV, and the last the single RIC code repeated for each single entries. Thus, I had to create an ”unnesting” function in order to clean the data into a dataframe where each line shows the value for the relative month

```python
def get_eikon_data(geo = 'europe'):
  ric_list = pd.read_csv('data/primary_funds/ric_'+geo+'.csv')
  ric_list = ric_list.dropna().T.iloc[1].to_list()
  diz_df2 = {}
  error_list = []
  data = pd.read_csv('data/primary_funds/data_'+geo+'.csv')
  count = 0
  for i in tqdm(ric_list[len(np.unique(data.Ric)):]):
    try:
      count += 1
      diz_df = {}
      name, err1 = ek.get_data([i],['TR.FundName'])
      values, err2 = ek.get_data([i],['TR.NETASSETVAL.date', 'TR.NETASSETVAL','TR.FundTER'],{"SDate":"19620101","EDate":"20201101", 'Frq':'CM'})
      diz_df['Ric'] = i
      diz_df['Name'] = name['Fund Name']
      diz_df['Date'] = np.array(values['Date'])
      diz_df['Value'] = np.array(values['Net Asset Value'])
      diz_df['Total Exp. Ratio'] = np.array(values['Total Expense Ratio'])
      diz_df2["{}".format(count)] = diz_df
    except:
      error_list.append(i)
      break
  data_nested = pd.DataFrame(diz_df2).T
  data_unnested = unnesting(data_nested,['Date','Value','Total Exp. Ratio']).reset_index().drop('index', axis = 1)
  data_final = data.append(data_unnested)
  data_final = data_final.drop(columns=['Unnamed: 0'])
  data_final.to_csv('data/primary_funds/data_'+geo+'.csv')
  return data_final, error_list

def unnesting(df, explode):
  idx = df.index.repeat(df[explode[0]].str.len())
  df1 = pd.concat([
  pd.DataFrame({x: np.concatenate(df[x].values)}) for x in explode], axis=1)
  df1.index = idx
  return df1.join(df.drop(explode, 1), how='left')
```

The code above shows the function I created (get_eikon_data) for importing the csv of RIC codes and a previously generated csv of returns (that is originally blank), and starts to query data from the API starting from the RIC corresponding to the number of funds already found, while it add the new data in the same returns csv file at the end. This permits to restart the code multiple times without losing data and without having double entries. More specifically, the function gather NAV, TER and the corresponding dates and stores them into separates Python dictionaries[2]; then the dictionaries are formatted in a single comprehensive dictionary that is finally unnested in order to clean the data and present it in rows in which each value is lined up with its fund name and date (expressed in month/year). The output of the function is in fact a dataframe containing the timeseries of all funds, together with an error list of all the funds not available. From this process I was able to obtain data for respectively 2906 Global funds, 1265 Europe funds, 723 USA funds and 467 Emerging Markets funds.

To conclude, the last step was preparing data for the analysis. In order to do that I just converted the previously obtained dataframes into pivot tables with funds as columns and dates as rows with the following lines of code:

```python
# import data precedently saved
data_europe = pd.read_csv('data/primary_funds/data_europe.csv', index_col='Date')
data_europe = data_europe.drop(columns=['Total Exp. Ratio','Unnamed: 0']).dropna()
data_europe.index = pd.to_datetime(data_europe.index, format='%Y-%m-%d').to_period('M')
#convert data into pivot table
```

[2] Python dictionaries are ordered lists presented in key:value pairs and thus indexed by keys like strings and numbers

```
data_europe_pivot = pd.pivot_table(data_europe,index='Date',
   columns='Name', values='Value', aggfunc='first')
#data_europe_pivot = data_europe_pivot.pct_change().round(6)
data_europe_pivot.to_csv('data/pivot/data_europe_pivot.csv')
data_europe_pivot
```

The above code loads the dataframe, drops all the columns not needed for the pivot and eliminates missing values[3]; finally it converts full dates to monthly periods and set them as row indexes, generating the final pivot table while also saving it as a csv.

[3] The deletion is performed by the command ”dropna()”: this doesn’t imply any loss of data but only the cleaning of blank cells necessary since in most cases dates for TER and NAV differs, and thus when TER is drop some rows remain empty

Chapter 4

Analysis

4.1 Models of Performance Measurement

I used two among the most popular models for measuring the performance in the dataset: the classic Capital Asset Pricing Model (CAPM) by Sharpe (1964), and the 4-factor model used by Carhart (1997). The Carhart 4-factor model is constructed by adding a factor which captures one-year momentum anomaly described by Jegadeesh and Titman (1993) to the well known Fama and French (1993) 3-factor model. All the factors including the momentum factor are updated to January 2021 and downloaded directly from the Kenneth R. French official Data Library (see French, n.d.), and are chosen depending on geography: developed countries factors for global dataset[1], european for europe dataset, emerging markets for emerging dataset and US for US dataset (the same used by Carhart and Fama and French themselves). Here a short explanation of the models:

$$CAPM: \quad E\left(R_{it}\right) - RF_t = \alpha_{iT} + \beta_{iT}(\mathrm{RM_t} - \mathrm{RF_t}) + e_{it} \quad t = 1, 2, \cdots, T \tag{4.1}$$

[1] Since the correspondent global factor-mimicking portfolios datatset were not available, I chose the developed countries factors which comprehend data from 23 of the most important economies (see http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_3developed.html)

$$C4FM: \quad E\left(R_{it}\right) - RF_t = \alpha_{iT} + \beta_{iT}(\mathrm{RM_t} - \mathrm{RF_t}) \\ + s_{iT}\mathrm{SMB}_t + h_{iT}\mathrm{HML}_t + p_{iT}\mathrm{WML}_t + e_{it} \quad t = 1, 2, \cdots, T \tag{4.2}$$

where $\mathrm{RF_t}$ is the US one-month T-bill rate, $\mathrm{RM_t}$ is the return on the specific region's value-weight market portfolio, while $\mathrm{SMB_t}$, $\mathrm{HML_t}$ [2] and $\mathrm{WML_t}$[3] are factor-mimicking portfolios respectively for size, book-to-market equity ratio and previous year return.

Cross-correlations between factors are extremely low, demonstrating their potential for explaining fund returns variations; I reported the summary statistics for the European factor-mimicking portfolios in Table 4.1 as an example, but the results are similar for all the datasets.

Table 4.1: **Summary Statistics for European 4-model Factors**

Factor Portfolio	Monthly Return	Std Dev	t-stat for Mean = 0	Cross-Correlations			
				RM-RF	SMB	HML	WML
RM-RF	0.55	4.90	2.13	1.00			
SMB	0.03	2.19	0.26	-0.11	1.00		
HML	0.22	2.53	1.62	0.24	-0.07	1.00	
WML	0.90	3.98	4.30	-0.35	0.08	-0.35	1.00

The four risk factors are used to explain and decompose the expected return on an asset into four different investment strategies: $\mathrm{SMB_t}$ accounts for strategies that invest on small-size firms and short big-size firms, while $\mathrm{HML_t}$ portfolios are long on high book-to-market firms and short on low B/M firms; finally $\mathrm{WML_t}$ (literally

[2] To construct the SMB and HML factors, stocks in the region of interest are sorted into two market cap and three book-to-market equity (B/M) groups at the end of each June. Big stocks are those in the top 90% of June market cap for the region, and small stocks are those in the bottom 10%. The B/M breakpoints for a region are the 30th and 70th percentiles of B/M for the big stocks of the region; then, SMB is the equal-weight average of the returns on the three small stock portfolios for the region minus the average of the returns on the three big stock portfolios, while HML is the equal-weight average of the returns for the two high B/M portfolios minus the average of the returns for the two low B/M portfolios.

[3] The 2x3 sorts on size and lagged momentum to construct WML are formed monthly. For portfolios formed at the end of month t–1, the lagged momentum return is a stock's cumulative return for month t–12 to month t–2. The momentum breakpoints for a region are the 30th and 70th percentiles of the lagged momentum returns of the big stocks of the region; then, WML is calculated as the equal-weight average of the returns for the two winner portfolios for a region minus the average of the returns for the two loser portfolios

Winners minus Losers) portfolios invest on past-year best performing stocks against past-year worst performing stocks. Regressing the returns on these time-series makes possible to estimate the magnitude of the contribution of each strategy to the actual excess performance.

4.2 Persistence in One-Year Decile Portfolios

Following Hendricks, Patel, and Zeckhauser (1990) and Carhart (1997) analysis, I divided Mutual Funds returns of each dataset into decile portfolios based on the previous year return. On 1st January of each year I sort the funds based on the previous year annualized return, and split them into ten portfolios by equally weighting the funds depending on the decile they falls into. The portfolios thus constructed are held until the following January, when the process is repeated. During the holding period, the weights are calculated for each month so that the portfolios are rebalanced as soon as a fund disappears. Summary statistics on the excess returns on these portfolios are shown in Table 4.2, where the portfolios means and standard deviations are reported for each subset.

As you can see, the winner and loser portfolios present the top and bottom average monthly excess return only for Global focus, while the higher return can be found in the second portfolio for the others; the worst return is found in the bottom portfolio except for Emerging markets that produce a much more casual pattern. The average monthly return decrease between decile portfolios is about 0.03% (0.04% for Global, 0.02% for Emerging) and the average spread between the first and last rank is 0.26% while the spread between the 9^{th} and 10^{th} deciles accounts for about 0.1% of it; however, none of this differences appears to be statistically significant (the lowest p value is 0.2 for the Global dataset). Nevertheless, the results obtained on the US funds are in line with those of Choi and Zhao (2020, p. 5); in their Working Paper, Choi and Zhao achieve the exact same results as Carhart using its own dataset (which I had no access to), and then use the procedure on a more recent timeframe (1994 - 2018) of US Mutual Funds returns that I took as a proxy for the

Table 4.2: **Decile Portfolios Monthly Excess Returns**

The table shows the average monthly excess returns for each portfolio of funds divided in deciles based on previous year annualized return. The operation is repeated for each subset of different geographical focus. The results show a clear decrease from top to bottom, with the exception of Emerging Markets whose returns are more randomly distributed while still maintaining the difference between top and bottom.

	Global		Europe		US		Emerging	
Portfolio	**Monthly Excess Return**	Std Dev	**Monthly Excess Return**	Std Dev	**Monthly Excess Return**	Std Dev	**Monthly Excess Return**	Std Dev
1 (Winners)	**0.50%**	4.29%	**0.41%**	4.06%	**0.67%**	4.38%	**0.45%**	4.06%
2	**0.43%**	3.74%	**0.50%**	4.03%	**0.74%**	4.41%	**0.57%**	4.03%
3	**0.34%**	3.67%	**0.43%**	4.06%	**0.63%**	4.19%	**0.38%**	4.06%
4	**0.31%**	3.41%	**0.36%**	4.04%	**0.55%**	3.96%	**0.51%**	4.04%
5	**0.40%**	3.61%	**0.31%**	4.06%	**0.48%**	4.02%	**0.38%**	4.06%
6	**0.26%**	3.60%	**0.33%**	4.12%	**0.41%**	3.96%	**0.33%**	4.12%
7	**0.27%**	3.61%	**0.35%**	4.07%	**0.51%**	3.90%	**0.25%**	4.07%
8	**0.31%**	3.63%	**0.26%**	3.99%	**0.45%**	4.16%	**0.46%**	3.99%
9	**0.33%**	3.58%	**0.28%**	3.89%	**0.43%**	4.07%	**0.36%**	3.89%
10 (Losers)	**0.15%**	4.22%	**0.17%**	4.03%	**0.39%**	4.24%	**0.29%**	4.03%
1 - 10	**0.35%**	0.07%	**0.24%**	0.02%	**0.28%**	0.14%	**0.16%**	0.02%

correctness of my methodology, which makes use of a similar dataset.

The results of the CAPM regression (Table 4.3) show how betas are similar between deciles, and thus alphas cannot offer specific explanations for the difference in returns among deciles: the winner portfolios have always positive alphas, while loser portfolios have always negative alphas, meaning that managers would be using their skills to deliberately achieve negative performance. However, the high p-values for all alphas (not reported) strongly indicates that the null hypothesis ($\alpha = 0$) cannot be rejected in general.

Conversely, the results from the 4-factor model, reported for each dataset at the end of the section (Tables 4.4, 4.5, 4.5a, 4.6), appear to explain much more of the returns pattern. First of all, for all the dataset the momentum factor is significatively

Table 4.3: **CAPM Regression Coefficients of Decile Portfolios divided by Geographical Focus**

The table shows the alphas and market betas obtained by the CAPM regression on each subset of data; t-statistic for each parameter is presented in parenthesis. The table also indicates the spread between top and bottom values.
**p value is below 0.01

	Global		Europe		US		Emerging	
Portfolio	**Alpha**	RMRF	**Alpha**	RMRF	**Alpha**	RMRF	**Alpha**	RMRF
1 (Winners)	**0.04%** (0.29)	0.79 (25.24)**	**0.10%** (0.70)	0.62 (21.05)**	**0.02%** (0.21)	0.89 (33.80)**	**0.05%** (0.31)	0.72 (27.71)**
2	**0.00%** (0.04)	0.72 (32.28)**	**0.08%** (0.58)	0.62 (22.59)**	**0.00%** (0.02)	0.90 (44.84)**	**0.08%** 0.52)	0.78 (31.63)**
3	**-0.04%** (-0.40)	0.69 (31.09)**	**0.06%** (0.47)	0.64 (22.83)**	**0.03%** (0.31)	0.79 (33.18)**	**0.11%** (0.88)	0.79 (38.24)**
4	**-0.06%** (-0.65)	0.70 (35.06)**	**0.08%** (0.57)	0.62 (22.93)**	**-0.03%** (-0.42)	0.80 (44.62)**	**0.22%** (1.61)	0.77 (35.27)**
5	**-0.00%** (-0.02)	0.70 (35.15)**	**-0.04%** (-0.30)	0.63 (22.83)**	**-0.08%** (-0.99)	0.83 (42.79**)	**-0.02%** (-0.15)	0.85 (47.66)**
6	**-0.08%** (-0.81)	0.70 (31.35)**	**0.05%** (0.40)	0.61 (23.48)**	**-0.10%** (-1.46)*	0.83 (52.12)**	**0.09%** (0.59)	0.78 (33.03)**
7	**-0.10%** (-1.05)	0.69 (29.58)**	**0.02%** (0.14)	0.58 (20.53)**	**-0.10%** (-1.29)	0.82 (44.84)**	**-0.14%** (-1.33)	0.82 (47.68)**
8	**-0.04%** (-0.43)	0.64 (27.25)**	**-0.02%** (-0.18)	0.58 (21.95)**	**-0.10%** (-1.21)	0.83 (42.85)**	**0.19%** (1.38)	0.73 (32.38)**
9	**0.06%** (0.53)	0.67 (27.69)**	**0.03%** (0.19)	0.60 (22.44)**	**-0.08%** (-0.83)	0.75 (36.11)**	**0.12%** (0.68)	0.73 (24.77**)
10 (Losers)	**-0.22%** (-1.48)	0.69 (19.74)**	**-0.05%** (-0.34)	0.60 (20.06)**	**-0.07%** (-0.63)	0.74 (27.42)**	**-0.11%** (-0.78)	0.71 (30.55)**
1 - 10 Spread	**0.26%** (1.77)	0.10 (5.5)	**0.15%** (1.04)	0.02 (0.99)	**0.09%** (0.84)	0.15 (6.38)	**0.16%** (1.09)	0.01% (-2.84)

correlated with the winners' excess returns, and negatively correlated with the losers, demonstrating a huge spread of an average 36 basis points between top and bottom. Secondly, also the Size factor explains a good part of the returns, confirming that small firms accounts for a good part of the top decile returns, especially in the US where the spread is greater, while in the global dataset and in Europe the difference is less pronounced. In this regard it is worth noting that the Europe dataset is the least explained by the model in terms of adjusted R-squared; this could be explained with the longer period of recession that europe has experienced since the subprime mortgage crisis and the subsequent sovereign debt crisis, which could have influenced the results of the model as exogenous variables. Another possible explanation could be the fact that the US is the only single-country dataset, thus immune to all the exogenous alteration due to differences in market regulation and economic/monetary policies that affect multi-countries datasets.

Nevertheless, in this case we can see a pattern reversal of the performance coefficients for which the top decile's alphas are always negative and the bottom decile's are always positive and this might hint at the fact that loss-making funds are actually trying to improve their situation, while top performers could earn more by making better decisions; although, p values from all datasets (not reported) show how we cannot consider the alphas statistically different from zero, as well as the differences between winners' and losers' portfolios. Conversely, the 1-10 spread is highly significant for Market and Momentum factor (p value below 1% in every dataset) and Size factor (p value below 5% for Europe and US)

Table 4.4: **Carhart 4-factor Regression on Global-focused Mutual Funds**

*p value is below 5%; **p value is below 1%

	Global					
Portfolio	**Alpha**	RMRF	Value	Size	Momentum	Adj R-sq
1 (Winners)	**-0.12%**	0.83	-0.20	0.27	0.27	0.749
	(-1.04)	(30.18)**	(-4)**	(4.75)**	(8.38)**	
2	**-0.06%**	0.74	-0.08	-0.02	0.11	0.758
	(-0.62)	(32.42)**	(-1.91)	(-0.42)	(4.06)**	
3	**-0.06%**	0.69	-0.10	-0.01	0.05	0.735
	(-0.59)	(30.14)**	(-2.44)**	(-0.14)	(1.98)**	
4	**-0.06%**	0.70	-0.13	0.13	0.03	0.791
	(-0.66)	(34.55)**	(-3.58)**	(3.15)	(1.46)	
5	**0.02%**	0.69	-0.06	0.03	-0.01	0.773
	(0.21)	(33.05)**	(-1.66)**	(0.66)*	(-0.42)	
6	**-0.05%**	0.69	-0.11	0.02	-0.01	0.734
	(-0.48)	(29.49)**	(-2.71)*	(0.36)	(-0.43)	
7	**-0.04%**	0.67	-0.16	0.09	-0.04	0.718
	(-0.43)	(27.73)**	(-3.61)	(1.71)	(-1.53)*	
8	**0.05%**	0.61	-0.12	0.06	-0.10	0.683
	(0.51)	(25.1)**	(-2.72)	(1.22)	(-3.5)**	
9	**0.16%**	0.64	-0.12	0.07	-0.11	0.692
	(1.56)	(25.51)**	(-2.78)	(1.3)**	(-3.91)**	
10 (Losers)	**0.03%**	0.61	-0.14	0.24	-0.31	0.590
	(0.21)	(18.05)**	(-2.24)	(3.36)**	(-7.89)**	

Table 4.5: **Carhart 4-factor Regression on Europe and US-focused Mutual Funds**

*p value is below 5%; **p value is below 1%

Europe						
Portfolio	**Alpha**	RMRF	Value	Size	Momentum	Adj R-sq
1 (Winners)	**-0.03%** (-0.18)	0.67 (21.51)**	-0.08 (-1.37)*	0.06 (0.98)	0.13 (3.15)**	0.576
2	**0.01%** (0.06)	0.66 (22.51)**	-0.10 (-1.8)*	0.05 (0.76)	0.08 (2.05)*	0.614
3	**0.05%** (0.37)	0.66 (22.28)**	-0.15 (-2.62)*	-0.08 (-1.3)	0.03 (0.88)	0.599
4	**0.10%** (0.72)	0.63 (21.95)**	-0.13 (-2.42)*	-0.10 (-1.71)	0.00 (0)	0.561
5	**0.01%** (0.07)	0.63 (21.43)**	-0.11 (-1.89)	-0.12 (-1.97)	-0.03 (-0.72)	0.622
6	**0.11%** (0.87)	0.62 (22.67)**	-0.19 (-3.66)*	-0.12 (-2.16)*	-0.03 (-0.81)	0.598
7	**0.13%** (0.92)	0.58 (19.44)**	-0.19 (-3.37)*	-0.16 (-2.63)	-0.07 (-1.86)	0.602
8	**0.09%** (0.71)	0.58 (21.24)**	-0.24 (-4.56)	-0.12 (-2.08)	-0.07 (-2.02)**	0.602
9	**0.22%** (1.68)	0.58 (21.16)**	-0.23 (-4.29)**	-0.06 (-1.07)	-0.15 (-4.28)**	0.597
10 (Losers)	**0.23%** (1.57)	0.57 (18.76)**	-0.26 (-4.41)**	-0.04 (-0.55)	-0.22 (-5.74)**	0.570

(a) **Carhart 4-factor Regression on US-focused Mutual Funds**

*p value is below 5%; **p value is below 1%

United States						
Portfolio	**Alpha**	RMRF	Value	Size	Momentum	Adj R-sq
1 (Winners)	**-0.09%** (-0.88)	0.90 (36.38)**	-0.02 (-0.7)	0.24 (7.26)**	0.17 (7.42)**	0.816
2	**-0.03%** (-0.33)	0.89 (42.39)**	-0.02 (-0.74)	0.10 (3.6)**	0.05 (2.52)**	0.852
3	**0.01%** (0.06)	0.79 (30.76)**	0.02 (0.56)	0.09 (2.6)	0.03 (1.46)	0.751
4	**-0.03%** (-0.38)	0.80 (40.97)**	0.00 (0.11)	0.01 (0.24)	0.00 (-0.18)	0.841
5	**-0.07%** (-0.79)	0.82 (39.16)**	-0.04 (-1.36)	-0.01 (-0.2)	-0.02 (-0.84)	0.83
6	**-0.08%** (-1.06)	0.81 (47.47)**	-0.01 (-0.58)	0.06 (2.71)	-0.03 (-2.18)**	0.882
7	**-0.09%** (-1.09)	0.81 (40.83)**	-0.02 (-0.8)	0.03 (1.05)**	-0.02 (-0.9)	0.843
8	**-0.07%** (-0.81)	0.83 (39.88)**	-0.01 (-0.2)	-0.06 (-2.11)	-0.05 (-2.48)*	0.835
9	**-0.01%** (-0.1)	0.72 (32.85)**	-0.07 (-2.23)	-0.02 (-0.52)	-0.08 (-4.14)**	0.787
10 (Losers)	**0.05%** (0.42)	0.70 (25.42)**	-0.06 (-1.51)	-0.08 (-2.07)*	-0.16 (-6.44)**	0.703

Table 4.6: **Carhart 4-factor Regression on Emerging-focused Mutual Funds**

*p value is below 5%; **p value is below 1%

			Emerging Markets			
Portfolio	**Alpha**	RMRF	Value	Size	Momentum	Adj R-sq
1 (Winners)	**-0.10%**	0.73	0.07	0.05	0.14	0.706
	(-0.59)	(26.82)**	(0.92)	(0.57)	(2.34)*	
2	**0.09%**	0.76	0.01	-0.17	-0.01	0.757
	(0.57)	(29.17)**	(0.13)	(-2.29)*	(-0.25)	
3	**0.12%**	0.78	0.03	-0.10	-0.03	0.819
	(0.88)	(35.37)**	(0.52)	(-1.55)	(-0.55)	
4	**0.17%**	0.76	0.08	-0.16	0.01	0.796
	(1.19)	(32.72)**	(1.35)	(-2.39)*	(0.23)	
5	**-0.11%**	0.85	0.07	-0.09	0.07	0.877
	(-0.97)	(45.02)**	(1.48)	(-1.58)**	(1.88)	
6	**0.05%**	0.76	0.16	-0.15	-0.04	0.777
	(0.29)	(30.55)**	(2.32)**	(-2.08)*	(-0.76)	
7	**-0.20%**	0.81	0.02	-0.20	0.06	0.881
	(-1.72)	(45.13)**	(0.5)	(-3.94)	(1.54)	
8	**0.22%**	0.71	0.03	-0.21	-0.03	0.769
	(1.44)	(29.8)**	(0.4)	(-3.09)*	(-0.65)	
9	**0.18%**	0.73	-0.13	-0.09	0.01	0.654
	(0.93)	(23.17)**	(-1.48)	(-1.01)**	(0.14)	
10 (Losers)	**0.05%**	0.69	-0.10	-0.12	-0.13	0.748
	(0.34)	(28.12)**	(-1.47)	(-1.76)	(-2.5)	

4.2.1 Performance Analysis on different subperiods

After the previous results, I decided to extend this research dividing the time frame into different subperiods. This choice derives from the will to isolate the possible influence of the dot-com bubble burst (which affects the sharp drop in prices between 2000 - 2003) and the subprime mortgage crisis (2008); thus, the subperiods considered are 1990 - 1999, 2000 - 2007 and 2008 - 2020 (all included). I selected the US focus Mutual Funds dataset since it showed the highest adjusted R-sq, and also in order to compare my results with Choi and Zhao (2020).

Interestingly, I found that the first subperiod accounts for the majority of inexplicability in the 4 factor model, having the lowest adjusted R-sq (between 0.49 and 0.72), while both the subsequent periods shows high levels between 0.85 and 0.98. Moreover, I showed in Table 4.7 how alphas have highly significant negative values for center ranks of the decile distribution for the period after the subprime crisis, even with high monthly excess returns, an evidence that these funds cannot achieve better results thanks to their management, but owe their growth principally to Market, Size and Value factors (that have very low p values, not reported); this result is also shared by some of the top-half of the decile distribution in the first subperiod. Anyhow, the difference between winners and losers portfolios is never significantly different from 0, if not for the CAPM alpha during the 2000 - 2002 period.

From this analysis I am tempted to conclude that either in a period of high market growth (the after-2008 crisis stock prices recover) we have no sufficient evidence of active managers' contribution to positive excess returns.

After repeating this same division for the other geographical focus, I confirmed that adjusted R-sq is lower in the first subperiod, probably mainly because it contains the least data across all my datasets, but at the same time I have found no evidence of statistically significant alphas and indeed p values very similar to those found for the complete timeframe analyzed in the previous section also for the other factors; for this reasons, I decided not to report them.

Table 4.7: **Monthly Excess Returns and Alphas for three subperiods of the US-focused Mutual Funds decile portfolios**

*p value is below 5%; **p value is below 1%

Portfolio	1990 - 1999			2000 - 2007			2008 - 2020		
	Monthly Excess Return	1-factor Alpha	4-factor Alpha	Monthly Excess Return	1-factor Alpha	4-factor Alpha	Monthly Excess Return	1-factor Alpha	4-factor Alpha
1	1.35%	0.34%	0.23%	0.13%	-0.04%	-0.24%	1.10%	0.05%	-0.06%
		(1.28)	(0.22)		(-0.27)	(-1.75)		(0.39)	(-0.55)
2	1.22%	**0.47%**	**0.45%**	0.29%	0.13%	0.04%	1.01%	0.00%	-0.07%
		(2.20)*	**(0.96)***		(0.98)	(-0.34)		(0.00)	(-0.64)
3	0.90%	0.02%	-0.11%	0.09%	-0.08%	-0.18%	0.90%	-0.08%	**-0.15%**
		(0.08)	(1.44)		(-0.75)	(-1.88)		(-1.15)	**(-2.0)***
4	1.03%	**0.45%**	0.33%	-0.12%	-0.30%	-0.31%	0.85%	**-0.19%**	**-0.21%**
		(2.09)*	(0.14)		(-2.87)	(-2.99)		**(-2.99)****	**(-3.15)****
5	1.26%	**0.62%**	**0.73%**	0.06%	-0.14%	-0.14%	0.89%	**-0.16%**	**-0.19%**
		(2.74)**	**(0.45)****		(-1.7)	(-0.98)		**(-3.06)****	**(-3.56)****
6	0.67%	-0.08%	0.10%	0.03%	-0.14%	-0.08%	0.92%	**-0.14%**	**-0.18%**
		(-0.37)	(3.11)		(-1.24)	(-1.25)		**(-2.24)***	**(-2.96)****
7	0.61%	-0.09%	0.03%	-0.07%	-0.25%	-0.23%	0.88%	**-0.15%**	**-0.15%**
		(-0.46)	(1.42)		(-3.08)	(-2.77)		**(-1.97)***	**(-2.04)***
8	1.24%	0.33%	0.39%	0.00%	-0.17%	-0.12%	0.90%	-0.13%	**-0.15%**
		(1.29)	(-0.41)		(-1.29)	(-0.93)		(-1.66)	**(-1.89)***
9	0.77%	0.18%	0.24%	-0.18%	-0.36%	-0.26%	0.91%	-0.11%	-0.06%
		(0.75)	(2.01)		(-2.37)	(-1.74)		(-1.28)	(-0.67)
10 (Losers)	0.80%	-0.04%	0.05%	-0.45%	-0.70%	-0.54%	0.97%	0.06%	0.10%
		(-0.20)	(0.89)		(-3.12)	(-2.47)		(0.51)	(0.75)
1 - 10 Spread	0.55%	0.38%	0.18%	0.58%	**0.74%**	0.78%	0.13%	-0.01%	-0.16%
	(-0.98)	(1.07)	(0.5)	(-0.84)	**(2.24)***	(1.15)	(-0.28)	(-0.05)	(-0.95)

Chapter 5

Conclusion

This research was intended to extend Carhart's analysis regarding a possible short-term persistence in Mutual Fund performance to the population of European funds and to a more recent time frame of returns. The main effort in doing so has been the data-set gathering and construction which took me the most time, not having available a complete and accessible source. Therefore, I think the python code I wrote for the Thomas Reuters Eikon DataStream API represents an important contribution of this thesis, that is completely replicable for every research that needs to access the API for any kind of fund characteristic in a straightforward manner. Another important prerogative of my code is the fact that it takes into account the survivorship bias, being able to download returns even for inactive or merged funds. The only step requested outside the python environment is the fund screening process, which is pretty easy and detailed on the Eikon DataStream platform using the filtering system it provides; once the LipperIDs are downloaded, everything is performed using a few Python functions that download the data, clean it and transform in a neat table.

For the same reason, also the code I wrote for the data analysis manages to achieve the same purpose of simplicity and cleanliness. I created my functions from scratch and despite I ran into some missteps due to my previous inexperience with the language, I managed to build a real module for this particular kind of quantile portfolios construction as well as CAPM, Fama and French 3 Factor and Carhart's

4 factor regression, in addition to the presentation of the results. This module is certainly improvable, but still it is ready for repeating this process with any kind of time-series and regressors.

As for the results of my analysis, I can summarize a few conclusions: first of all I demonstrated the correctness of the process by being able to achieve virtually the same outputs with a similar time frame of Choi and Zhao (2020), who themselves replicated the Carhart (1997) results with the same data-set which I had no access to. Secondly, I found evidences of short-term performance persistence in different geographical focus subsets of Mutual Funds; I also showed that this persistence is more erratic if not negligible in Emerging Markets, and it is instead more pronounced in both Europe and Global equity investments as well as in the US, even including the subprime crisis period in the time frame. Finally, I demonstrated through a 4-factor model which incorporates a Momentum exploiting factor, that there are no evidence for presuming that active management is able to achieve better results than a market benchmark index. Indeed, my results shows greater alphas for losers portfolios than for winners, thus I am tempted to conclude that if there were ETFs based on momentum and the 3-factor models that would not do worse than the actively managed funds, this would imply that best performing managers would be using their differential information, if any, to achieve worse returns.

Bibliography

Brown, Stephen J. and William N. Goetzmann (1995). “Performance persistence”. In: *The Journal of Finance* 50, pp. 679–698. DOI: https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1540-6261.1995.tb04800.x.

Carhart, Mark M. (1997). “On Persistence in Mutual Fund Performance.” In: *The Journal of Finance* 52.1, pp. 57–82. URL: https://www.jstor.org/stable/2329556?seq=1.

Choi, James J and Kevin Zhao (2020). *Did Mutual Fund Return Persistence Persist?* Working Paper 26707. National Bureau of Economic Research. DOI: 10.3386/w26707. URL: http://www.nber.org/papers/w26707.

Elton, Edwin J., Martin J. Gruber, and Christopher R. Blake (1996). “The Persistence of Risk-Adjusted Mutual Fund Performance”. In: *The Journal of Business* 69.2, pp. 133–157. URL: http://www.jstor.org/stable/2353461.

Elton, Edwin J. et al. (1993). “Efficiency with Costly Information: A Reinterpretation of Evidence from Managed Portfolios”. In: *The Review of financial studies* 6.1, pp. 1–22. URL: https://www.jstor.org/stable/2961987.

Fama, Eugene F. and Kenneth R. French (1993). “Common risk factors in the returns on stocks and bonds”. In: *Journal of Financial Economics* 33.1, pp. 3–56. ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(93)90023-5.

French, Kenneth R. (n.d.). *Kenneth R. French - Data Library.* URL: http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html (visited on 2021).

Goetzmann, William N. and Roger G. Ibbotson (1994). “Do winners repeat? Patterns in mutual fund performance”. In: *The Journal of Portfolio Management* 20.2, pp. 9–18. DOI: https://doi.org/10.3905/jpm.1994.9.

Grinblatt, Mark and Sheridan Titman (1992). “The Persistence of Mutual Fund Performance”. In: *The Journal of finance* 47.5, pp. 1977–1984. DOI: https://doi.org/10.2307/2329005.

Harvey, Andrew C. (1990). *The Econometric Analysis of Time Series.* The London School of Economics, p. 211. URL: https://books.google.it/books?id=yKQN62Itmg0C&printsec=frontcover&hl=it&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

Hendricks, Darryll, Jayendu Patel, and Richard Zeckhauser (1990). *Hot Hands in Mutual Funds: The Peristence of Performance, 1974-87.* 3389. 1050 Massachusetts Avenue, Cambridge, MA: National Bureau of Economic Research. DOI: https://doi.org/10.3386/w3389.

— (1993). “Hot hands in mutual funds: Short-run persistence of performance”. In: *The Journal of Finance* 48.1, pp. 93–130. URL: https://www.jstor.org/stable/2328883?pq-origsite=summon&seq=1.

Jegadeesh, Narasimham and Sheridan Titman (1993). "Returns to buying winners and selling losers: Implications for stock market efficiency". In: *The Journal of Finance* 48.1, pp. 65–91. DOI: `https://doi.org/10.2307/2328882`.

Sharpe, William F. (1964). "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk". In: *The Journal of Finance* 19.3, pp. 425–442.

Wermer, Russ (1997). *Momentum investment strategies of mutual funds, performance persistence, and survivorship bias.* Boulder, CO.: Graduate School of Business and Administration, University of Colorado. URL: `https://www.researchgate.net/profile/Russ_Wermers/publication/228392513_Momentum_Investment_Strategies_of_Mutual_Funds_Performance_Persistence_and_Survivorship_Bias/links/00463514770418086700000.pdf`.

Appendix A

Code

A.1 Data Gathering code

```python
%load_ext autoreload
%autoreload 2
%matplotlib inline
import pandas as pd
import pandas_datareader as web
import risk_kit as rk
import statsmodels.api as sm
import numpy as np
import eikon as ek
from tqdm import tqdm
from varname import nameof

#set key for accessing API
ek.set_app_key('b0************************************30')

## import selected fund list from csv
global_funds = pd.DataFrame(pd.read_excel('data/primary_funds/global_primary.xlsx'))
europe_funds = pd.DataFrame(pd.read_excel('data/primary_funds/europe_primary.xlsx'))
emerging_funds = pd.DataFrame(pd.read_excel('data/primary_funds/emerging_primary.xlsx'))
usa_funds = pd.DataFrame(pd.read_excel('data/primary_funds/usa_primary.xlsx'))

lipper_global = global_funds.T.iloc[0].to_list()
lipper_europe = europe_funds.T.iloc[0].to_list()
lipper_emerging = emerging_funds.T.iloc[0].to_list()
lipper_usa = usa_funds.T.iloc[0].to_list()

#Retrive RIC codes from LipperIDs
ric_europe, error_ric_europe = get_ric(lipper_europe, geo ='europe')
ric_emerging, error_ric_emerging = get_ric(lipper_emerging, geo='emerging')
ric_usa, error_ric_usa = get_ric(lipper_usa, geo='usa')
ric_global, error_ric_global = get_ric(lipper_global, geo='global')

# Data Gathering
```

```
data_europe, error_data_europe = get_eikon_data(geo='europe')
data_global, error_data_global = get_eikon_data(geo='global')
data_emerging, error_data_emerging = get_eikon_data(geo='emerging')
data_usa, error_data_usa = get_eikon_data(geo='usa')

#Converting DataFrames into Pivot Tables
# import data precedently saved
data_europe = pd.read_csv('data/primary_funds/data_europe.csv',
    index_col='Date')
data_europe = data_europe.drop(columns=['Total Exp. Ratio','Unnamed
    : 0']).dropna()
data_europe.index = pd.to_datetime(data_europe.index, format='%Y-%m
    -%d').to_period('M')
#convert data into pivot table
data_europe_pivot = pd.pivot_table(data_europe,index='Date',
    columns='Name', values='Value', aggfunc='first')
#data_europe_pivot = data_europe_pivot.pct_change().round(6)
data_europe_pivot.to_csv('data/pivot/data_europe_pivot.csv')
```

A.2 Analysis code

```

  %load_ext autoreload
  %autoreload 2
  %matplotlib inline
  import pandas as pd
  pd.reset_option('max_rows')
  import pandas_datareader as web
  import thesis_kit as tk
  import statsmodels.api as sm
  import numpy as np
  import eikon as ek
  from tqdm import tqdm
  import risk_kit as rk
  import scipy.stats
  from sklearn.metrics import r2_score

  data_global_pivot = tk.get_pivot(geo='global')
  data_global_pivot = tk.replace_outliers(data_global_pivot)

  data_europe_pivot = tk.get_pivot(geo='europe')
  data_europe_pivot = tk.replace_outliers(data_europe_pivot)

  data_usa_pivot = tk.get_pivot(geo='usa')
  data_usa_pivot = tk.replace_outliers(data_usa_pivot)

  data_emerging_pivot = tk.get_pivot(geo='emerging')
  data_emerging_pivot = tk.replace_outliers(data_emerging_pivot)

  #showing funds summary stats
  fund_summary_stats(data_global_dead).round(1)

  #portfolios construction
  n = 10
  p_global = tk.get_quantiles_portfolios(data_global_pivot,
   n_portfolios=n)
```

```
p_europe = tk.get_quantiles_portfolios(data_europe_pivot, n_portfolios=n)
p_usa = tk.get_quantiles_portfolios(data_usa_pivot,n_portfolios=n)
p_emerging = tk.get_quantiles_portfolios(data_emerging_pivot, n_portfolios=n)

#showing results summary
mean_global = tk.excess_ret_mean(p_global,fff_global)
std_global = p_global.std()*100
mean_europe = tk.excess_ret_mean(p_europe,fff_europe)
std_europe = p_europe.std()*100
mean_usa = tk.excess_ret_mean(p_usa,fff_usa)
std_usa = p_usa.std()*100
mean_emerging = tk.excess_ret_mean(p_emerging,fff_emerging)
std_emerging = p_emerging.std()*100
mean_results = pd.concat([mean_global,std_global,mean_europe, std_europe,mean_usa,std_usa,mean_emerging,std_europe], axis=1)
mean_results.columns = [’global’,’std’,’europe’,’std’,’usa’,’std’,’emerging’,’std’]
mean_results.round(2)

#uploading fama and french factors + momentum factor
fff_global , mom_global = tk.get_ff_file(geo=’developed’)
fff_europe , mom_europe = tk.get_ff_file(geo=’europe’)
fff_usa , mom_usa = tk.get_ff_file(geo=’usa’)
fff_emerging , mom_emerging = tk.get_ff_file(geo=’emerging’)

#calculating and displaying regression coefficients
#output = ’coeff’ for parameters, ’pvals’ for p-values, ’t-stat’ for t-statistic, ’adj_r2’ for adjusted R-squared
global_coeff = tk.get_regression_coeff(fff_global, p_global,ff=True,output=’coeff’)
pd.concat([global_coeff[’Alpha’]*100, global_coeff[[’Mkt-RF’,’Value’,’Size’,’Momentum’]]], axis = 1).round(3)
europe_coeff = tk.get_regression_coeff(fff_europe, p_europe,ff=True,output=’pvals’)
pd.concat([europe_coeff[’Alpha’], europe_coeff[[’Mkt-RF’,’Value’,’Size’,’Momentum’]]], axis = 1).round(3)
usa_coeff = tk.get_regression_coeff(fff_usa, p_usa,ff=True,output=’t-stat’)
pd.concat([usa_coeff[’Alpha’], usa_coeff[[’Mkt-RF’,’Value’,’Size’,’Momentum’]]], axis = 1).round(3)
emerging_coeff = tk.get_regression_coeff(fff_emerging, p_emerging,ff=True,output=’adj_r2’)
pd.concat([emerging_coeff[’Alpha’], emerging_coeff[[’Mkt-RF’,’Value’,’Size’,’Momentum’]]], axis = 1).round(3)
```

A.3 Module of functions

```

# DATA GATHERING

def get_ric(lipper_list, geo = ’europe’):
’’’
Retrives ric codes from a list of lipperIDs
```

```
'''
ric_list = pd.read_csv('data/primary_funds/ric_'+geo+'.csv')
ric_list = ric_list.T.iloc[1].to_list()
error_list = []
for i in tqdm(lipper_list[len(ric_list):]):
try:
ric, none = ek.get_data([str(i)], 'TR.RIC')
ric_list.append(ric.iloc[0,1])
except:
error_list.append(i)
break
pd.Series(ric_list).to_csv('data/primary_funds/ric_'+geo+'.csv')
return ric_list, error_list

def get_eikon_data(geo = 'europe'):
'''
from a list of Ric codes, gather access to the Eikon API
requesting monthly Net Asset Value and Total Expense Ratio for each
'''
ric_list = pd.read_csv('data/primary_funds/ric_'+geo+'.csv')
ric_list = ric_list.dropna().T.iloc[1].to_list()
diz_df2 = {}
error_list = []
data = pd.read_csv('data/primary_funds/data_'+geo+'.csv')
count = 0
for i in tqdm(ric_list[len(np.unique(data.Ric)):]):
try:
count += 1
diz_df = {}
name, err1 = ek.get_data([i],['TR.FundName'])
values, err2 = ek.get_data([i],['TR.NETASSETVAL.date', 'TR.NETASSETVAL','TR.FundTER'],{"SDate":"19620101","EDate":"20201101", 'Frq':'CM'})
diz_df['Ric'] = i
diz_df['Name'] = name['Fund Name']
diz_df['Date'] = np.array(values['Date'])
diz_df['Value'] = np.array(values['Net Asset Value'])
diz_df['Total Exp. Ratio'] = np.array(values['Total Expense Ratio'])
diz_df2["{}".format(count)] = diz_df
except:
error_list.append(i)
break
data_nested = pd.DataFrame(diz_df2).T
data_unnested = unnesting(data_nested,['Date','Value','Total Exp. Ratio']).reset_index().drop('index', axis = 1)
data_final = data.append(data_unnested)
data_final = data_final.drop(columns=['Unnamed: 0'])
data_final.to_csv('data/primary_funds/data_'+geo+'.csv')
return data_final, error_list

def unnesting(df, explode):
idx = df.index.repeat(df[explode[0]].str.len())
df1 = pd.concat([
pd.DataFrame({x: np.concatenate(df[x].values)}) for x in explode], axis=1)
df1.index = idx
```

```
return df1.join(df.drop(explode, 1), how='left')

#ANALYSIS

def get_pivot(geo='europe'):
'''
import data precedently saved
'''
df = pd.read_csv('data/primary_funds/data_'+geo+'.csv',index_col='Date')
df = df.drop(columns=['Total Exp. Ratio','Unnamed: 0']).dropna()
df.index = pd.to_datetime(df.index, format='%Y-%m-%d').to_period('M')
#convert data into pivot table
pivot = pd.pivot_table(df,index='Date', columns='Name', values='Value', aggfunc='first')
pivot = pivot.pct_change().round(6)
pivot.to_csv('data/pivot/data_'+geo+'_pivot.csv')
return pivot

def replace_outliers(df):
'''
Clean a dataframe from the values otside the 1st and 99th percentile
'''
lb = df.quantile(0.01)
ub = df.quantile(0.99)
df_new = df[(df < ub) & (df > lb)]
return df_new

def fund_summary_stats(df):
'''
Returns a DataFrame of statistics for a group of fund returns
'''
living = delete_funds(df, dead=True)
dead = delete_funds(df, dead=False)

tot_n = len(df.columns) , len((df.iloc[-1]).dropna()), len(df.columns) - len((df.iloc[-1]).dropna())
avg_n = df.apply(lambda x: x.count(), axis=0).mean()
avg_age = df.apply(lambda x: x.count(), axis=1).mean()/12

return pd.DataFrame({
  'Total Number': tot_n,
  'Avg Number' : avg_n,
  'Avg Age' : avg_age
}, index = ['All', 'Live Funds', 'Dead Funds']).round(2)

def annualize_returns(r, periods_per_year=12):
'''
Annualizes a set of returns
You should infer the number of periods per year
'''
compound_growth = (1+r).prod()
n_periods = r.shape[0]
return compound_growth**(periods_per_year/n_periods)-1

def annualize_pivot(df):
```

```
'''
Runs the annualization of monthly returns on a DataFrame
'''
df = df.reset_index()
df = df.groupby(df.Date.dt.year).aggregate(annualize_returns,12).drop('Date',axis=1)
return df

def get_ff_file(geo='usa'):
'''
Load the Fama and French 3 Factors + momentum factor for the specified geo focus
Possible Geo:   .global
.europe
.usa (default)
.emerging
'''
fff = pd.read_csv('data/factors/'+geo+'_fff.csv',header=0, index_col=0, parse_dates=True,na_values=-99.99)
fff.index = pd.to_datetime(fff.index, format='%Y%m').to_period('M')

mom = pd.read_csv('data/factors/'+geo+'_mom.csv', header=0, index_col=0, parse_dates=True,na_values=-99.99)
mom.index = pd.to_datetime(mom.index, format='%Y%m').to_period('M')

fff = pd.merge(fff, mom, how='left', left_index=True, right_index=True).dropna()
fff = fff[['Mkt-RF','SMB','HML','WML','RF']]
fff = fff/100
return fff , mom

def ff_regression(fff, df, fund_column=0, ff=True):
'''
Returns the summary of a Fama and French regression
Requires a DataFrame where columns are series of fund returns, the number of the column of the fund interested,
and the DataFrame with fama and french explanatory variables
'''
fund = pd.DataFrame(df[df.columns[fund_column]].dropna().mask(df[df.columns[fund_column]]==0, df[df.columns[fund_column]].mean(skipna=True))) #extract column from the table
start = max(fund.index[0], fff.index[0]) # set start date as first observable return of the fund
end = min(fund.index[-1], fff.index[-1]) # set end date as last observable return of the fff
fund_excess = fund[start:end] - fff.loc[start:end,['RF']].values
mkt_excess = fff.loc[start:end,['Mkt-RF']]
exp_var = mkt_excess.copy() #variable for the market exposure
exp_var['Alpha'] = 1 #variable for the alpha
if ff:
exp_var['Value'] = fff.loc[start:end,['HML']] # variable for value exposure
exp_var['Size'] = fff.loc[start:end,['SMB']] # variable for size exposure
exp_var['Momentum'] = fff.loc[start:end,['WML']] # variable for value exposure
```

```
lm = sm.OLS(fund_excess.astype(float), exp_var.astype(float)).fit
 ()
else:
lm = sm.OLS(fund_excess.astype(float), exp_var[['Alpha', 'Mkt-RF'
 ]]).fit()
return lm

def results_summary_to_dataframe(results, ff=True):
'''
take the result of an statsmodel results table and transforms it
 into a dataframe
'''
if ff:
hypothesis = 'Mkt-RF = 0, Alpha = 0,Value = 0, Size = 0, Momentum
 = 0'
else:
hypothesis = 'Alpha = 0, Mkt-RF = 0'
pvals = results.pvalues
coeff = results.params
tstats = results.t_test(hypothesis).tvalue
conf_lower = results.conf_int()[0]
conf_higher = results.conf_int()[1]
t_stat = results.t_test(hypothesis)
r2 = results.rsquared
adj_r2 = 1-(1-r2)*(370-1)/(370-4-1)

results_df = pd.DataFrame({"pvals":pvals,
  "coeff":coeff,
  "t-stat":tstats,
  "conf_lower":conf_lower,
  "conf_higher":conf_higher,
  "adj_r2":adj_r2
})
#Reordering...
results_df = results_df[["coeff",'t-stat',"pvals","conf_lower","
 conf_higher","adj_r2"]]
return results_df

def get_regression_coeff(fff,df,ff=True, output='coeff'):
'''
Runs the ff regression on a complete DataFrame of returns
when ff = False, it runs the CAPM regression
'''
coeff_list=[]
error_list=[]
for col in range(df.shape[1]):
try:
coeff = results_summary_to_dataframe(ff_regression(fff,df,
 fund_column=col, ff=ff),ff=ff)[output]
coeff.name = df.columns[col]
coeff_list.append(coeff)
except:
error_list.append(col)
pass
results = pd.DataFrame(coeff_list)
return results

def eq_w(r):
```

```
'''
Find the equal weights for a series of returns
'''
n = len(r)
if n == 0:
ew = 0
else:
ew = 1/n
return ew

def get_quantiles_portfolios(df, n_portfolios=10):
'''
Starting from a DataFrame of monthly funds returns, with funds as columns and dates as rows
each Jan calculates previous year annual return and divide funds into quantile portfolios, then held for the following year
Returns n-portfolios monthly returns
'''
df_month = df
df_year = annualize_pivot(df_month)
p_df = pd.DataFrame(index=df.index, columns=[np.arange(1, n_portfolios+1)])
for row in range(0,len(df_year)-1):
r = pd.DataFrame(df_year.iloc[row].replace(0,np.nan).dropna()).T
r.index = ['ret']
r = r.T.sort_values(by='ret', ascending=False)
if (r.values.astype(bool).sum(axis=0)) >= n_portfolios:
r['Decile Rank'] = pd.qcut(r['ret'],n_portfolios, labels=False)
for quantile in range(0,n_portfolios):
funds = r.loc[r['Decile Rank']==quantile].index
year = df_month.iloc[row*12+12:row*12+12+12][funds].dropna(axis=1,how='all')
for month in range(0,len(year)):
rets = year.iloc[month].dropna()
w = eq_w(rets)
p_ret = rets.values*w
p_ret = p_ret.sum()
p_df.iloc[row*12+12+month,quantile] = p_ret
else:
for quantile in range(0,n_portfolios):
for month in range(0,12):
p_df.iloc[row*12+12+month,quantile] = np.nan
return p_df

def delete_funds(df, dead=True):
'''
Delete columns in which the last observation:
is NaN -- dead = True
is not Nan -- dead = False
'''
df = df.replace(0,np.nan)
df = df.T
if dead:
df = df[df.iloc[:,-1].notna()]
else:
df = df[df.iloc[:,-1].isna()]
df = df.T
return df
```