

# ASSET PRICING - EMPIRICAL APPLICATION 1

## FACTORIAL MODEL AND RISK PREMIUM DECOMPOSITION - APT

Luis Miguel Fonseca  
Stéphane Eloundou Mvondo  
Natalia Cárdenas Frías

November 29, 2023

### Introduction

Focusing on recent data from the French equity market, we want to better comprehend how the market prices systemic, non-diversifiable risk embedded in the risk premium of stocks, i.e. any expected compensation beyond the risk-free return. We base our analysis on a linear decomposition of said premium on different *factors* of risk in the spirit of the Arbitrage Pricing Theory (APT) pioneered by Ross [1976]. Unlike the CAPM model which considers a unique risk premium in the market, the Ross model gives a more detailed description of the pricing of aggregate risk by decomposing the contributions of different sources of risk. Here, a risky portfolio of  $j$  stocks<sup>1</sup> is compensated with  $k$  risk premia associated with the  $k$  *common factors* that the portfolio is exposed to.

## 1 Data

### 1.1 French stock market data

We decided to build a case study of the French market because it is a liquid and matured market, central in Europe. In the case of this analysis, we had trouble getting the data needed to perform it for other countries<sup>2</sup> and the fact that France has more publicly available data helped us choose it as our market of study.

We built a portfolio with 30 French stocks that we got from Yahoo Finance. For simplicity, the synthetic portfolio is composed of one stock of each company and its composition does not change during the period studied. Table 1 shows the companies that we used to create this portfolio, they are all publicly traded companies in France since the early 2000's in Euronext Paris. Importantly, we tried to have a certain diversity in the sectors

---

<sup>1</sup>Let  $j \in \{1, \dots, J\}$  with  $J$  sufficiently large so that all idiosyncratic risk can be fully diversified. We better explain the difference between idiosyncratic and aggregate in the context of the Ross model in Section 1.1

<sup>2</sup>Initially we thought about using data from the German market but we couldn't for instance find data for their inflation-linked bond yield that we use as an endogenous factor in Section 2.1.2

represented to be able to capture some diversification to risk even if the portfolio is too small and we do not reweight it. However, because we try to implement a version of the [Fama and French \[1992\]](#) factor analysis, we restrain ourselves from choosing financial companies as the authors do due to their high leverage. Other than these two conditions, the choice of the companies was mainly restricted to data availability on public 'long' series on firm-level data, notably on market capitalization and book-to-market ratio to be able to incorporate the [Fama and French \[1992\]](#) factors to our analysis.

Because we are interested in the underlying determinants of the risk premia, and due to data availability issues, we decided to have a broad analysis with monthly data for our selection of stocks for the period 2005-2022. Monthly data is for instance used by [Chen et al. \[1986\]](#). While this is not a very long period, it encompasses important moments in the financial markets in particular the Great Financial Crisis, the subsequent European Debt Crisis, and the Covid years. Also importantly, during most of this period (following the GFC), monetary policy fixed interest rates were extremely low driving down the return of sovereign debt for countries like France and Germany<sup>3</sup> that could have been assimilated to the risk-free rate. This means that for an investor to get any returns, it had to hold risky assets. Moreover, the period following the GFC and up to 2021 was also characterized by extremely low inflation in the Euro Zone. This is interesting because the APT usually incorporates inflation risk as market risk, yet inflation was nowhere to be found for more than a decade. Seeing how the market incorporated this monetary reality is by itself an intriguing question.

## 1.2 Data description and sources

The other series that we use are the following and its sources, how they are used in the context of the analysis is described in subsequent sections.

- As a proxy for the free rate of the market we consider two measures:
  - The yield of short-term OAT, i.e. French treasuries taken from [Banque de France's website](#). As for most developed, stable countries, short-term sovereign bonds are taken as the risk-free asset as Governments are supposed to be more solvent than other agents in the economy, after all, they decide their income and could seize resources via taxes to meet their obligations.
  - The spot yield curve spot rate, for 3-month maturity of all government bonds rated triple A in the Euro Area, retrieved from the [ECB webpage](#). On top of the fact that this is a measure for short-term sovereign bonds, we consider this to be a relevant proxy for the French market due to the strong integration within the European capital market. If an investor decides that the French market becomes risky, she can easily move her investments to another European capital market that looks safer.
- To get the market rate, we use the return of the main index of the country, the CAC40 also taken from Yahoo Finance as for the components of our synthetic portfolio. In hindsight, we are not sure of the pertinence of comparing our portfolio to this index. While the composition of our portfolio is not the

---

<sup>3</sup>They were negative for certain maturities in real terms for a part of the time frame analyzed, pretty much since the European Debt Crisis until the inflation surge after Covid.

Company Name	Ticker	Industry
Accor	AC.PA	Hospitality
Air Liquide	AI.PA	Industrial Gases
Air France-KLM	AF.PA	Airlines
Airbus	AIR.PA	Aerospace
Biomerieux	BIM.PA	Biotechnology
BIC	BB.PA	Consumer Goods
Bouygues	EN.PA	Construction
Capgemini	CAP.PA	Information Technology
Carrefour	CA.PA	Retail
Casino	CO.PA	Retail
Dassault Aviation	AM.PA	Aerospace
Danone	BN.PA	Food and Beverage
Hermes International	RMS.PA	Fashion and Luxury
JCDecaux	DEC.PA	Advertising
Kering	KER.PA	Fashion and Luxury
L'Oreal	OR.PA	Cosmetics
LVMH	MC.PA	Fashion and Luxury
Michelin	ML.PA	Automotive
Nexans	NEX.PA	Electrical Equipment
Orange	ORA.PA	Telecommunications
Renault	RNO.PA	Automotive
Saint-Gobain	SGO.PA	Manufacturing
Sanofi	SAN.PA	Pharmaceuticals
Sodexo	SW.PA	Food Services
TF1	TFL.PA	Broadcasting
Thales	HO.PA	Aerospace and Defense
TotalEnergies	TTE.PA	Energy
Ubisoft	UBI.PA	Video Games
Vinci	DG.PA	Construction
Vivendi	VIV.PA	Entertainment

Table 1: Synthetic portfolio: Companies, Tickers, and Industries

same as the CAC40<sup>4</sup>, due to the data availability issues we've been mentioning, we see that our choices are heavily biased towards 'big name' companies that are those belonging to the index.

- We got the series of the exchange rate between the Euro and the US dollar from Yahoo Finance. It is read as the amount of USD needed to get one euro.
- The GDP series is taken from the [ECB webpage](#). It is available at a quarterly frequency and is available at market prices.
- The harmonized headline inflation rate is taken from the [INSEE webpage](#).
- For the market inflation expectation in a 10-year horizon, we use the break-even inflation rate published by [Agence France Trésor](#) online. Sadly, data is only available from 2013.
- For the implementation of the two additional [Fama and French \[1992\]](#) factors, we took different routes
  - We found the estimation of the factors published by K. French in his [online Data Library](#) that are

<sup>4</sup>Not all the stocks we chose are necessarily part of the CAC40 at every period studied, and the CAC40 is a weighted index that evolves over time.

constantly updated. Their estimations start in the 1990s and are made for different markets using the comprehensive CRSP dataset that is not freely available. He has an estimation for the European market that we downloaded to use but it is not clear which stocks are used to replicate their portfolio.

- To try to build these estimations ourselves for our portfolio meaning that a minima we need data on the market capitalization of each company during the time frame studied and its *book*. This information is hardly available without having access to platforms like Bloomberg or CRSP. The best information that we could find comes from [this](#) website that publishes the market capitalization and the price-to-book (the inverse of the book-to-market ratio) annual series for several stocks. The data is however cannot be directly downloaded from the site so we scrap it to get the series (see Code Appendix B). Our biggest fear with this source is that it is not clear at all where the information comes from even if they mention several quality [data providers](#) as their partners. Since it is the only source that resembles what is needed for this part we used it but we are not confident about it.

## 2 Empirical strategy and implementation

We implement a minimal approach to [Ross \[1976\]](#), namely using fewer factors than in [Chen et al. \[1986\]](#). We include both exogenous and endogenous macroeconomic risk factors as well as an approximation to implement [Fama and French \[1993\]](#) three-factor model which used stock-specific data.

Let the return  $R_j$  of the  $j$ -th component of her portfolio can be described by the following expression  $\forall j \in \{1, \dots, N\}$ :

$$R_j = \mathbb{E}[R_j] + \underbrace{\sum_{k=1}^K \beta_{j,k} f_k}_{\text{Systemic risk}} + \overbrace{u_j}^{\text{Idiosyncratic risk}} \quad (1)$$

Where  $\mathbb{E}[R_j]$  is the expected return of asset  $j$  and  $R_j$  its return without dividend i.e. the first difference of the stock price. This is the so-called *factorial model* that includes two sources of risk:

- The investor faces centered idiosyncratic risks  $u_j$ ,  $\mathbb{E}[u_j] = 0$  that are assumed to be completely diversifiable with a portfolio "large enough" ( $N$  big) because they are independent of each other  $u_j \perp u_{j'} \forall j \neq j'$ .
- She also faces  $k$  different sources of aggregate risk, modeled by the linear combination of  $f_k$  centered *shocks* that influence all  $R_j$  with a sensitivity  $\beta_{j,k}$ . By definition, these risks cannot be diversified because they affect the returns of all asset and thus has to be compensated which is the focal point of our study.

In order to implement a regression analysis and estimation as the one that follows in this section, we shall also assume that the idiosyncratic risks are uncorrelated with aggregate risk  $\text{corr}(u_j, f_k) = 0, \forall j, k$ .

This section is organized as follows. Section 2.1 identifies the aggregate market risks  $(f_k)_k$  that we are going to consider as the ones priced by the market. Section 2.2 runs a first regression analysis to identify the sensitivity of each return to each factor of risk, i.e. to identify the  $(\beta_{j,k})_{j,k}$  from the factorial model (Eq. 1). It also implements an approximation to a parallel model that decomposes the return of stocks: the [Fama and French \[1992\]](#) that

also will lead us to estimate the sensitivity of the return of each return to a series of factors. Finally, section 2.3 uses the series of  $(\hat{\beta}_{j,k})_{j,k}$  estimated before to implement an estimation of the *multibeta relationship* which is the regression that will allow us to get how much the market is remunerating the exposition to a market factor risk.

## 2.1 Identify the risk factors

We decided to explore the role of the following sources of risk for a first model inspired by Ross [1976] and Chen et al. [1986].

- The activity risk, measured by changes in GDP
- Inflation risk, measured both by the HICP of France for the short term and by the market inflation expectation in a 10-y horizon.
- Devaluation risk measured by the exchange rate between the Euro and the US dollar.

We then consider the factors of a complementary factor model, Fama and French [1992] which are not directly linked to risks.

### 2.1.1 Exogeneous factors

These are risk factors that do not depend directly on the financial markets or more precisely that are not deduced from a linear combination of the returns of financial assets. In our case, it is mainly the activity risk, the short-term interest rate (measured by the HICP), and the devaluation risk. Importantly, the measure of these risks is not directly measured by changes or by the level of the underlying variables as markets are informational efficient [Fama, 1970], and they have already priced in all relevant information conveyed by prices. In particular, all *predictable* movements, say in inflation, have already been incorporated by the market. This means that the factors of risk are actually the *surprises* in the movements of these variables.

To extract this unpredictable part of these variables we need to use the residuals of some time series model of the (stationary) variables<sup>5</sup>, in our case ARIMA( $p, d, q$ ) models. We used the function *auto.arima* from the R *forecasts* package that set de parameters optimally by minimizing the BIC between different specifications. We ended up with:

- ARIMA(0,1,0) (i.e. a random walk<sup>6</sup>) for the series of log GDP.
- ARIMA(0,1,1) for the exchange rate series.
- ARIMA(2,1,2) for the HICP series.

We then stored the residuals of each one of these models and merge it in the data frame with the stock return. We shall note that by construction the factor (i.e. the residuals) are by construction centered around zero. They are actually the innovations of the initial series for the macroeconomic variables.

---

<sup>5</sup>Evidently, we need that the series used are stationary before conducting any TS analysis. We tested for stationarity of our series using the ADP test and then differentiate the series that were non-stationary (see Appendix A). We didn't apply the ADF test to the series in logs afterward as the R function *auto.arima* automatically differentiates the series  $d$  times until they are stationary.

<sup>6</sup>We do find this result 'weird' but didn't find any mistake in the code and it does square with the factors being centered.

### 2.1.2 Endogeneous factors

These factors are linear combinations of the returns of financial assets. We used the *Breakeven Inflation* at a 10-year horizon which is simply the yield difference between 10-y OATi 0.10% (*Obligations Assimilables du Trésor*) indexed by the HICP and non-indexed 10-y OAT. This gap is assimilated to how markets price the possibility of having future inflation<sup>7</sup>. Once again we need to extract the unforecastable movements in this indicator to have an adequate measure of risk. We follow the same methodology as with the exogenous factors and retrieve the residuals of an ARIMA(2,1,2) model.

### 2.1.3 French-Fama factors

Fama and French [1993] can also be seen as an extension of the CAPM model but their factors, while significant specially in the US market, are hardly interpretable as risk factors. The authors show that the variation of the returns of an asset can be explained not only by the exposure to market risk as in the CAPM represented by the difference of the market return and the risk-free rate  $[R_M - R_f]$ , but also by a size and value premium in the following model.

$$R_j = \alpha_j + R_f + \beta_{m,j}[R_M - R_f] + \beta_S SMB + \beta_V HML + \varepsilon_j \quad (2)$$

The size premium refers to the observation that stocks with small market capitalizations tend to outperform stocks with larger ones and it is captured by the factor SMB, *small minus big*. It is computed as the difference in average returns of the 30% stocks with the smallest market capitalization and the average returns of the 30% stocks associated with the firms with the largest market capitalization. The value premium refers to the outperformance of "value stocks" i.e. those that have high book-to-market (B/M) and it is represented by the difference in an average return of the 50% of stocks with the highest  $B/M$  ratio (value stocks) and the 50% with lowest  $B/M$  ratio (growth stocks). We use this methodology to create our own SMB and HML factors at a yearly frequency.

As we previously mentioned, K. French made public his estimation of the factors for the European model using a more complex approach with "6 value-weight portfolios formed on size and book-to-market". SMB (Small Minus Big) is the average return on the three small portfolios minus the average return on the three big portfolios, and HML (High Minus Low) is the average return on the two value portfolios minus the average return on the two growth portfolios.

## 2.2 Estimate the sensitivities of the factorial model

### 2.2.1 Ross model

Given that we assumed that the idiosyncratic risks  $u_j$  are uncorrelated with the aggregate factor risks  $f_k$  we can estimate the factorial model in Eq. 1 with linear regression models<sup>8</sup>. We will also pose that  $f_k \perp f_{k'} \forall k, k'$  in

<sup>7</sup>Note that the Breakeven inflation is the inflation rate that would equalize the return of these two sovereign bonds (due to AOA).

<sup>8</sup>The residuals will be uncorrelated with the explanatory variables

order to have a good identification of the parameters with linear regression estimators  $\forall k_0, \beta_{j,k_0} = \frac{cov(R_j, f_{k_0})}{Var(f_{k_0})} \equiv \hat{\beta}$ .

For this first estimation, we need to include a time dimension on the baseline factorial model  $\forall j, \forall t$ :

$$R_{j,t+1} = \mathbb{E}_t[R_{j,t+1}] + \sum_{k=1}^K \beta_{j,k} f_{k,t+1} + u_{j,t+1} \quad (3)$$

We will therefore estimate the following:

$$R_{j,t} \sim \alpha + \sum_{k=1}^K \beta_{j,k} \hat{f}_{k,t}$$

Note that for the macroeconomic risks  $\hat{f}_{k,t}$  denotes the estimated errors we found in the previous section for each series with standard TS analysis. To do so, we split the dataset by stock and performed linear regressions for each subgroup.

**Exogeneous factors only** We performed a first version of this estimation using only the exogenous factors which allows us to have an analysis of the 2005-2022 period. Results are shown in the following table.

Table 2: Estimate the beta coefficients for each exogeneous factor

	accor (1)	air-france-klm (2)	air-liquide (3)	airbus (4)	bic (5)	biomerieux (6)	bouygues (7)	capgemini (8)	carrefour (9)	casino (10)	saint-gobain (11)	danone (12)	dass (13)
res_pib	4.417 (16.882)	87.731 (171.635)	6.695 (8.268)	13.047 (27.583)	-11.895 (10.889)	8.568 (15.496)	-49.367*** (13.562)	25.025 (22.659)	5.304 (22.047)	-19.160 (12.504)	-22.502 (66.982)	-52.184 (35.688)	0.215 (1.600)
res_xr	12.577 (9.864)	9.750 (100.281)	6.565 (4.830)	18.735 (16.116)	-13.262** (6.362)	0.480 (9.054)	1.404 (7.924)	-8.086 (13.239)	10.413 (12.881)	-12.388* (7.306)	25.959 (39.135)	-5.407 (20.852)	-0.909 (0.900)
res_infl	1.383 (0.880)	-0.111 (8.947)	0.556 (0.431)	-0.840 (1.438)	1.128** (0.568)	0.239 (0.808)	0.925 (0.707)	2.895** (1.181)	3.652*** (1.149)	0.252 (0.652)	2.233 (3.492)	1.565 (1.860)	0.000 (0.000)
Constant	-0.031 (0.263)	6.689** (2.674)	0.059 (0.129)	-0.097 (0.430)	0.251 (0.170)	0.081 (0.241)	0.563*** (0.211)	0.366 (0.353)	0.556 (0.343)	0.196 (0.195)	3.082*** (1.044)	1.418** (0.556)	0.000 (0.000)
Observations	215	215	215	215	215	215	215	215	215	215	215	215	215
R <sup>2</sup>	0.021	0.001	0.020	0.009	0.042	0.002	0.068	0.033	0.050	0.025	0.005	0.014	0.000
Adjusted R <sup>2</sup>	0.007	-0.013	0.006	-0.005	0.029	-0.012	0.055	0.019	0.037	0.011	-0.009	0.0001	-0.000
Residual Std. Error	3.844	39.080	1.882	6.280	2.479	3.528	3.088	5.159	5.020	2.847	15.251	8.126	0.300
F Statistic	1.474	0.091	1.454	0.629	3.100**	0.129	5.162***	2.390*	3.715**	1.825	0.344	1.009	0.400

Note:



**Exogeneous and endogeneous factors** We add another specification to include the endogenous market expectation of long-term inflation. Sadly due to the short series on the Breakeven inflation, this analysis is conducted only on the period 2013-2022, still at a monthly frequency.

Table 3: Estimate the beta coefficients for exogeneous and endogeneous factors (2013-2022)

	accor (1)	air-france-klm (2)	air-liquide (3)	airbus (4)	bic (5)	biomerieux (6)	b (7)
res_pib	47.225 (99.831)	-114.811 (239.823)	-0.127 (42.066)	252.848 (208.506)	22.185 (45.090)	36.520 (45.088)	
res_xr	14.768 (11.125)	23.779 (26.725)	10.730** (4.688)	20.465 (23.235)	-4.871 (5.025)	2.910 (5.024)	
res_infl	1.153 (1.330)	-2.827 (3.196)	-0.124 (0.561)	-2.077 (2.779)	-0.075 (0.601)	-0.113 (0.601)	
res_bkeven	65.720 (341.840)	-571.630 (821.196)	-213.246 (144.043)	-1,300.032* (713.961)	-53.868 (154.397)	2.813 (154.389)	
Constant	0.084 (0.357)	1.753** (0.859)	0.212 (0.151)	-0.061 (0.747)	0.188 (0.161)	0.007 (0.161)	
Observations	109	109	109	109	109	109	
R <sup>2</sup>	0.036	0.018	0.064	0.061	0.013	0.011	
Adjusted R <sup>2</sup>	-0.001	-0.020	0.028	0.025	-0.025	-0.027	
Residual Std. Error (df = 104)	3.619	8.695	1.525	7.559	1.635	1.635	
F Statistic (df = 4; 104)	0.965	0.482	1.768	1.700	0.349	0.300	

*Note:*

**Exogeneous and French and Fama factors** The last specification includes the French and Fama factors given by K. French data library. As in the baseline it refers to monthly data on the 2005-2022 period.

### 2.2.2 French and Fama model

We also wanted to try to estimate the three French and Fama with the data we collected. We used the approach mentioned in Section 2.1.3 and compared the mean return of value stocks and non-value stocks as well as of growth vs non-growth stocks. Once again, due to data availability issues, we had to restrain our sample to study only the 2010-2022 period at a yearly frequency.

The results are not very significant but we think it is likely due to our data quality. We show the results for this but we will ignore this representation of the French and Fama model moving forward.

Table 4: Estimate the beta coefficients for each exogeneous factor and French and Fama factors

	accor (1)	air-france-klm (2)	air-liquide (3)	airbus (4)	bic (5)	biomerieux (6)
res_pib	9.008 (14.726)	197.078 (168.809)	8.992 (7.026)	22.914 (24.369)	−9.534 (10.381)	14.176 (15.545)
res_xr	−21.529** (9.354)	−85.560 (107.234)	−10.601** (4.463)	−32.113** (15.480)	−27.630*** (6.594)	−8.154 (9.875)
res_infl	0.344 (0.767)	−3.228 (8.795)	−0.002 (0.366)	−2.571** (1.270)	0.792 (0.541)	−0.103 (0.810)
HML	0.331*** (0.093)	−2.902*** (1.066)	0.198*** (0.044)	0.524*** (0.154)	0.034 (0.066)	−0.033 (0.098)
SMB	0.362*** (0.127)	1.365 (1.459)	0.261*** (0.061)	0.942*** (0.211)	−0.075 (0.090)	0.262* (0.134)
Mkt.RF	0.296*** (0.049)	1.888*** (0.558)	0.134*** (0.023)	0.401*** (0.080)	0.172*** (0.034)	0.095* (0.051)
Constant	−0.211 (0.228)	5.396** (2.614)	−0.029 (0.109)	−0.376 (0.377)	0.165 (0.161)	0.005 (0.241)
Observations	215	215	215	215	215	215
R <sup>2</sup>	0.286	0.075	0.322	0.259	0.166	0.038
Adjusted R <sup>2</sup>	0.266	0.048	0.303	0.238	0.142	0.010
Residual Std. Error (df = 208)	3.305	37.887	1.577	5.469	2.330	3.489
F Statistic (df = 6; 208)	13.904***	2.799**	16.492***	12.121***	6.910***	1.369

Note:

Table 5: Estimate the beta coefficients for computed French and Fama factors (2010-2022)

	accor (1)	air-france-klm (2)	air-liquide (3)	airbus (4)	bic (5)	biomerieux (6)	bou (7)
mkt.free	0.012*** (0.001)	0.180*** (0.010)	0.007*** (0.0004)	0.010*** (0.001)	0.006*** (0.001)	−0.0004 (0.001)	0.0 (0.0)
HMLest	−0.006 (0.005)	−0.017 (0.057)	−0.001 (0.002)	0.004 (0.006)	0.007* (0.004)	0.005 (0.007)	0.0 (0.0)
SMBest	−0.032** (0.013)	0.127 (0.148)	−0.006 (0.006)	−0.012 (0.016)	0.014 (0.010)	0.004 (0.019)	0.0 (0.0)
Constant	0.133 (0.121)	1.124 (1.374)	0.008 (0.053)	−0.145 (0.152)	−0.011 (0.094)	0.010 (0.176)	−0.0 (0.0)
Observations	390	390	390	390	390	390	390
R <sup>2</sup>	0.349	0.469	0.467	0.192	0.156	0.002	0.0
Adjusted R <sup>2</sup>	0.344	0.465	0.463	0.186	0.150	−0.006	0.0
Residual Std. Error (df = 386)	1.673	18.957	0.731	2.102	1.301	2.429	1.6
F Statistic (df = 3; 386)	68.960***	113.475***	112.609***	30.571***	23.869***	0.212	57.0

Note:

### 2.3 Estimate the remuneration of risk from the multibeta relationship

Under AOA, assuming that the factorial model (Eq. 1) is an accurate depiction of how equity is price, implies that the expected returns are constrained by a multibeta relationship of the following form  $\exists \rho, \exists \lambda_1, \dots, \lambda_k$ :

$$\mathbb{E}[R_j] = \rho + \sum_{k=1}^K \lambda_k \beta_{j,k} \quad (4)$$

where each  $\lambda_k$  is the parameter representing the market price of risk (the risk premium) that the market re-tributes for being exposed to a given risk factor  $f_k$  with a sensitivity  $\beta_{j,k}$

To estimate the lambda coefficients we need to have a new dependent variable: the historical k=mean return of every stock in order to get a cross-sectional analysis. To do this last part of the analysis we kept the

### 2.4 Test the validity of the multibeta relationship

## Conclusion

## Appendix A Additional tables and figures

### Macroeconomic factors

Table 6: Results ADF with trend and drift

	EuroUSD	CAC40	Inflation	PIB	RF AAA	OAT	1pct	5pct	10pct
tau3	-2.945	-2.028	-0.565	-4.716	-1.011	-0.720	-3.990	-3.430	-3.130
phi2	3.012	1.652	1.243	7.665	0.800	0.555	6.220	4.750	4.070
phi3	4.421	2.220	1.522	11.121	1.198	0.683	8.430	6.490	5.470

Table 7: Results ADF with drift

	EuroUSD	CAC40	Inflation	PIB	RF AAA	OAT	1pct	5pct	10pct
tau2	-1.841	-1.428	-0.444	-2.117	-1.545	-1.142	-3.460	-2.880	-2.570
phi1	1.791	1.277	0.438	2.589	1.195	0.803	6.520	4.630	3.810

Table 8: Results ADF with no trend nor drift

	EuroUSD	CAC40	Inflation	PIB	RF AAA	OAT	1pct	5pct	10pct
tau1	-0.618	0.414	0.353	0.721	-1.437	-0.995	-2.580	-1.950	-1.620

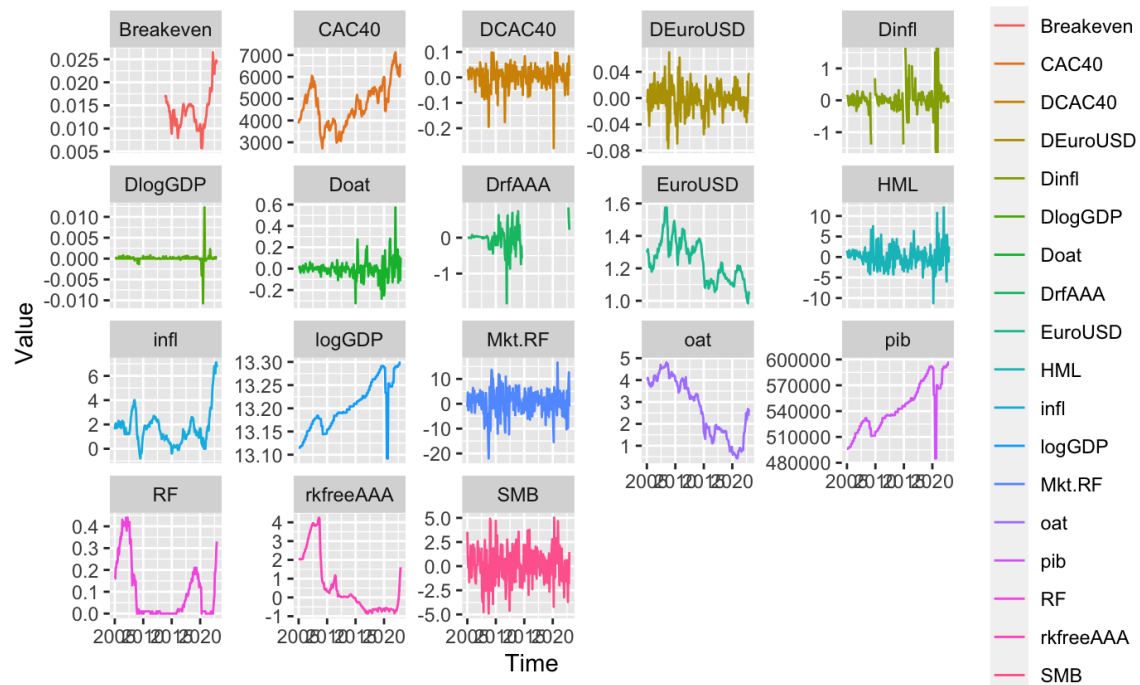


Figure 1: Factors: Series in level and in deltas

## Appendix B Code - Data gathering and cleaning in Python

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 APi - Data gathering and Data Cleaning
5
6 Scrapping - Firm level data for French and Fama factors
7 Use Yahoo Finance API to get the financial data for all the stocks
8 Use Eurostat API to get the macro data
9 Two data sets (French-Fama factors and long term inflation expectation) are found online and
   have been downloaded in CSV file beforehang
10
11 Merge and clean the dataset
12
13 @author: nataliacardenasf
14 """
15
16 import pandas as pd
17 import numpy as np
18 import os
19
20 import requests
21 from bs4 import BeautifulSoup
22
23 #import pandas_datareader.data as web

```

```

24 import yfinance as yf
25 #from eurostatapidclient import EurostatAPIClient
26 import datetime
27
28
29 os.chdir('/Users/nataliacardenasf/Documents/GitHub/PROJECTS_AP_FE/AP 1')
30
31 company_names_lower = [
32     'air-liquide', 'airbus', 'bouygues', 'capgemini', 'carrefour', 'casino-guichard-perrachon',
33     , 'vivendi',
34     'kering', 'l-oreal', 'lvmh', 'michelin', 'orange', 'renault', 'sanofi', 'thales',
35     'totalenergies', 'vinci', 'compagnie-de-saint-gobain', 'ubisoft', 'tf1', 'danone',
36     'dassault-aviation', 'air-france-klm', 'accor', 'bic', 'hermes-international',
37     'jcdecaux', 'nexans', 'sodexo', 'biomerieux', "CAC40", "EuroUSD"]
38
39 tickers = [
40     'AI.PA', 'AIR.PA', 'EN.PA', 'CAP.PA', 'CA.PA', 'CO.PA', 'VIV.PA', 'KER.PA', 'OR.PA', 'MC.
41     PA',
42     'ML.PA', 'ORA.PA', 'RNO.PA', 'SAN.PA', 'HO.PA', 'TTE.PA', 'DG.PA', 'SGO.PA', 'UBI.PA', '
43     TFI.PA',
44     'BN.PA', 'AM.PA', 'AF.PA', 'AC.PA', 'BB.PA', 'RMS.PA', 'DEC.PA', 'NEX.PA', 'SW.PA', 'BIM.
45     PA', "^FCHI", 'EURUSD=X']
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
262
```

```

66         data.append({'Year': year, 'MarketCap': market_cap, 'Company':
        company_name})
67         return pd.DataFrame(data)
68     return None
69
70 def scrape_price_book(url, company_name):
71     response = requests.get(url)
72     if response.status_code == 200:
73         soup = BeautifulSoup(response.content, 'html.parser')
74         table_body = soup.find('table', class_='table').find('tbody')
75         if table_body:
76             data = []
77             rows = table_body.find_all('tr')
78             for row in rows:
79                 cols = row.find_all('td')
80                 if len(cols) >= 2:
81                     year = cols[0].text.strip()
82                     pricebook = cols[1].text.strip()
83                     #variation = cols[2].text.strip()
84                     data.append({'Year': year, 'PriceBook': pricebook, 'Company': company_name
85                                 })
86             return pd.DataFrame(data)
87         return None
88
89 # Scraping market cap data
90 dfmktcap = pd.DataFrame()
91 for url, company in zip(mktcap_urls, company_names_lower[:-2]):
92     data = scrape_market_cap(url, company)
93     if data is not None:
94         dfmktcap = pd.concat([dfmktcap, data])
95
96 #Scap price book
97 dfpricebook = pd.DataFrame()
98 for url, company in zip(pricebook_urls, company_names_lower[:-2]):
99     data = scrape_price_book(url, company)
100    if data is not None:
101        dfpricebook = pd.concat([dfpricebook, data])
102
103 #indexes
104 dfmktcap['Year'] = pd.to_datetime(dfmktcap['Year'])
105 dfmktcap['Year'] = pd.DatetimeIndex(dfmktcap['Year']).year
106
107 dfpricebook['Year'] = pd.to_datetime(dfpricebook['Year'])
108 dfpricebook['Year'] = pd.DatetimeIndex(dfpricebook['Year']).year
109
110
111 ##Merge datasets

```

```

112 final_firm = dfmktcap.copy()
113 final_firm = final_firm.merge(dfpricebook, how='outer', on=['Year', 'Company'])
114
115
116 del dfmktcap, dfpricebook, mktcap_urls, pricebook_urls, url, data, company
117
118
119 missing = final_firm[final_firm.isna().any(axis=1)]
120 missing = missing.sort_values(by=['Year'])
121 missing = missing.reset_index()
122 #have both data points for all firms for 2010-2022
123 # in 09 only missing data is from BIC, Carrefour, Ubisolft, AirFrance
124
125 #remove 2023
126 final_firm = final_firm[final_firm.Year != 2023]
127
128 #Get book to market ratio = inverse of price-book ratio
129 final_firm['PriceBook'] = pd.to_numeric(final_firm['PriceBook'], errors='coerce')
130 final_firm['PriceBook'].replace('nan', np.nan, inplace=True)
131 final_firm['BookMarket'] = final_firm['PriceBook'].apply(lambda x: x ** -1 if not pd.isnull(x)
    ) else np.nan)
132
133 #Clean MarketCap
134 final_firm['MarketCap'] = (final_firm['MarketCap'].replace({'\$: ': '', ' B': ''}, regex=True).
    astype(float) * 1_000) # Clear the letters, convert to float and scale to millions
135
136 #Date format
137 final_firm['Year'] = pd.to_datetime(final_firm['Year'], format='%Y')
138
139 #====Get monthly data
140 monthly_data = pd.DataFrame()
141 # Repeat the yearly data for each month and each firm
142 for index, row in final_firm.iterrows():
143     firm_data = pd.DataFrame()
144     monthly_year = pd.date_range(start=row['Year'], periods=12, freq='MS')
145     firm_data['Date'] = monthly_year
146     firm_data['Company'] = row['Company']
147     firm_data['MarketCap'] = row['MarketCap']
148     firm_data['BookMarket'] = row['BookMarket']
149     firm_data['PriceBook'] = row['PriceBook']
150     monthly_data = pd.concat([monthly_data, firm_data])
151
152 del index, monthly_year, row, firm_data
153
154 firms_year = final_firm.copy()
155 firms_month = monthly_data.copy()
156
157 del final_firm, monthly_data, missing

```



```

158
159 %%Get return data with Yahoo finance
160
161 start = datetime.datetime(2002, 1, 1)
162 end = datetime.datetime(2022, 12, 31)
163
164 #Get all data
165 data = yf.download(tickers, start=start,
166                     end=end)
167
168 #Focus on adjusted closed values only
169 adjclose=data['Adj Close']
170 adjclose = adjclose.set_axis(company_names_lower, axis=1)
171
172 #Use monthly data: mean of the months value
173 adjclose = adjclose.resample('1M').mean(numeric_only=True)
174 adjclose_y = adjclose.resample('1Y').mean(numeric_only=True)
175
176 ## extract CAC40 and exchange rate
177 cac_xrate_month = adjclose.loc[:, ["CAC40", "EuroUSD"]]
178 cac_xrate_year = adjclose_y.loc[:, ["CAC40", "EuroUSD"]]
179 adjclose = adjclose.drop(columns=["CAC40", "EuroUSD"])
180 adjclose_y = adjclose_y.drop(columns=["CAC40", "EuroUSD"])
181
182 #Reshape
183 prices_monthly = pd.melt(adjclose, value_vars=company_names_lower[0:-2], ignore_index=False)
184 prices_yearly = pd.melt(adjclose_y, value_vars=company_names_lower[0:-2], ignore_index=False)
185
186
187 del data, adjclose, adjclose_y
188
189 #I'm not getting right values for xrate when downloading in bulk
190 data = yf.download(['EURUSD=X', '^FCHI'], start=start, end=end)
191 adjclose=data['Adj Close']
192 adjclose = adjclose.set_axis(['EuroUSD', 'CAC40'], axis=1)
193 #adjclose = pd.DataFrame(adjclose, columns=['Date', 'EuroUSD'])
194
195 cac_xrate_month = adjclose.resample('1M').mean(numeric_only=True)
196 cac_xrate_year = adjclose.resample('1Y').mean(numeric_only=True)
197
198 del data, adjclose, start, end
199
200
201 %% Endogeneous factor: long term inflation expectation from external file
202
203 #upload the Agence France Tresor data
204 pi_endo= pd.read_excel('2023_11_01_rend_tit_ref_oatei.xls', skiprows=[0,1,2,3,4], usecols
                    =[0,3])

```

```

205 pi_endo.columns = ['Date', "Breakeven"]
206
207 pi_endo["Date"] = pd.to_datetime(pi_endo["Date"])
208 pi_endo= pi_endo.set_index(pi_endo["Date"])
209 pi_endo.drop(columns=['Date'])
210
211 #get monthly data
212 piendo_month = pi_endo.resample('1M').mean(numeric_only=True)
213
214 #get yearly data
215 piendo_year = pi_endo.resample('1Y').mean(numeric_only=True)
216
217 del pi_endo
218
219 ### French and Fama - their data
220
221 df = pd.read_csv('Europe_3_Factors.csv',skiprows=[0,1,2])
222
223 #montly data, need to fix dates
224 frenchfama_month = df.iloc[:399,:]
225
226 frenchfama_month['Unnamed: 0'] = frenchfama_month['Unnamed: 0'].astype(str) # Convert to
    string for manipulation
227 frenchfama_month['Year'] = frenchfama_month['Unnamed: 0'].str[:4] # Extract year from the
    encoded date
228 frenchfama_month['Month'] = frenchfama_month['Unnamed: 0'].str[4:] # Extract month from the
    encoded date
229 frenchfama_month['Date'] = pd.to_datetime(dict(year=frenchfama_month['Year'], month=
    frenchfama_month['Month'], day=1))
230 frenchfama_month.drop(['Year', 'Month', 'Unnamed: 0'], axis=1, inplace=True)
231 frenchfama_month = frenchfama_month.set_index(frenchfama_month['Date'])
232 frenchfama_month = frenchfama_month.drop(columns=["Date"])
233 frenchfama_month = frenchfama_month.loc['2002-01-01':]
234 frenchfama_month = frenchfama_month.astype(float)
235
236
237 #yearly data
238 frenchfama_year = df.iloc[402:,:]
239 frenchfama_year["Unnamed: 0"] = pd.to_datetime(frenchfama_year['Unnamed: 0'])
240 frenchfama_year.rename(columns={"Unnamed: 0":'Date'}, inplace=True)
241 frenchfama_year = frenchfama_year.set_index(frenchfama_year['Date'])
242 frenchfama_year = frenchfama_year.drop(columns=["Date"])
243 frenchfama_year = frenchfama_year.loc['2002-01-01':]
244 frenchfama_year = frenchfama_year.astype(float)
245
246 del df
247
248

```

```

249 ### Macro data
250 #APIs didn't work as planned
251
252 ## PIB Q
253 pib = pd.read_csv("ECB_PIB.csv")
254 pib.columns = ['Date', 'Q', 'pib']
255 pib['Date'] = pd.to_datetime(pib['Date'])
256 pib = pib.drop(columns=['Q'])
257 pib = pib.set_index(pib['Date'])
258 pib = pib.loc['2002-01-01':]
259 pib = pib.drop(columns=['Date'])
260
261 #monthly
262 pib_monthly = pib.resample('MS').ffill()
263 #yearly
264 pib_year = pib.resample('1Y').last()
265
266 del pib
267
268 ##risk free AAA
269 rkfreeAAA = pd.read_csv('ECB_yield.csv')
270 rkfreeAAA.columns=['Date', "time", "rkfreeAAA"]
271 rkfreeAAA['Date'] = pd.to_datetime(rkfreeAAA['Date'])
272 rkfreeAAA = rkfreeAAA.set_index(rkfreeAAA['Date'])
273 rkfreeAAA = rkfreeAAA.drop(columns=['time', "Date"])
274
275 rkfreeAAA_monthly = rkfreeAAA.resample("1M").mean(numeric_only=True)
276 rkfreeAAA_year = rkfreeAAA.resample("1Y").mean(numeric_only=True)
277
278 del rkfreeAAA
279
280 ##HICP
281 pi_month = pd.read_csv("HICP.csv", sep=';', encoding = 'latin1', skiprows=[0,1,2,3], usecols
    = [0,1], header=None)
282 pi_month.columns= ['Date', "infl"]
283 pi_month["Date"] = pd.to_datetime(pi_month['Date'], format = "%Y-%m")
284 pi_month=pi_month.set_index(pi_month["Date"])
285 pi_month = pi_month.drop(columns=['Date'])
286
287 #yearly
288 pi_year = pi_month.resample("1Y").mean(numeric_only=True)
289
290
291 ##OAT
292 oat = pd.read_csv('OAT.csv', sep=';', skiprows=[0,1,2,3,4,5], usecols=[0,5], header=None)
293 oat.columns = ['Date', 'oat']
294 oat['Date'] = pd.to_datetime(oat["Date"])
295 oat = oat.set_index(oat["Date"])

```

```

296 oat = oat.drop(columns=["Date"])
297 oat = oat.loc["2002-01-01":]
298
299 oat['oat'] = oat['oat'].replace("-", np.nan)
300 oat['oat'] = oat['oat'].str.replace(',', '.').astype(float)
301
302 oat_month = oat.resample('M').mean(numeric_only=True)
303 oat_year = oat.resample('1Y').mean(numeric_only=True)
304
305 del oat
306
307
308 ### Merge the dataframes
309
310 === MONTHLY
311 #macro stuff, only date index matter
312 monthly = pd.concat([cac_xrate_month, frenchfama_month, oat_month, pi_month, pib_monthly,
313                     piendo_month, rkfreeAAA_monthly], axis=1)
314 monthly = monthly.resample('M').last() #some tables encoded end of month, others on the 1st
315 monthly.to_csv('Monthly_series.csv')
316
317
318 #firm specific data
319 firms_month['Date'] = firms_month['Date'] + pd.offsets.MonthEnd(0) #all other df have eomonth
320                                     date
321 firms_month = firms_month.set_index(['Date'])
322 firms_month = firms_month.set_index('Company', append=True)
323
324 prices_monthly.columns = ['Company', 'value']
325 prices_monthly.set_index('Company', append=True)
326
327 monthly_stock = pd.merge(prices_monthly.reset_index(), firms_month.reset_index(), on=["Date",
328                                     "Company"], how='outer').set_index(["Date", "Company"])
329 monthly_stock.to_csv("Firm_monthly.csv")
330
331 #merge the two
332 merged_monthly = pd.merge(monthly, monthly_stock, left_index=True, right_index=True, how='
333                                     right')
334
335 #Export df in csv
336 merged_monthly.to_csv("DATA_month.csv")
337
338 === YEARLY
339 yearly = pd.concat([cac_xrate_year, frenchfama_year, oat_year, pi_year, pib_year, piendo_year,
340                     rkfreeAAA_year], axis=1)

```

```

339 yearly = yearly.resample('Y').last()
340
341 yearly.to_csv('Yearly_series.csv')
342
343 # =====
344 # #firm specific
345 # firms_year.rename(columns={"Year":'Date'}, inplace=True)
346 # firms_year['Date'] = firms_year['Date'] + pd.offsets.MonthEnd(0)
347 # firms_year = firms_year.set_index(['Date'])
348 # firms_year = firms_year.set_index('Company', append=True)
349 #
350 # prices_yearly.columns = ['Company', 'value']
351 # prices_yearly = prices_yearly.set_index('Company', append=True)
352 #
353 # yearly_stock = pd.merge(prices_yearly.reset_index(), firms_year.reset_index(), on=["Date",
354 #     "Company"], how='outer').set_index(["Date", "Company"])
355 # =====
356
357 #yearly_stock = monthly_stock.groupby('Company').resample('Y').mean()
358
359 monthly_data_reset = monthly_stock.reset_index()
360 yearly_stock = monthly_data_reset.groupby('Company').resample('Y', on='Date').mean()
361 #yearly_stock = yearly_stock.set_index(["Date", "Company"])
362
363
364
365 #merge the two
366 merged_year = pd.merge(yearly, yearly_stock, left_index=True, right_index=True, how='right')
367
368 merged_year.to_csv("DATA_yearly.csv")

```

## Appendix C Code - Analysis in R

```
1 #%% AP 1
2 #%% ncardenasfrias
3
4 pacman::p_load(data.table, urca, tidyverse, ggplots, xts, stargazer, forecast, plm, ggplot2,
5   tidyrr)
6 library(dplyr)
7 setwd('/Users/nataliacardenasf/Documents/GitHub/PROJECTS_AP_FE/AP 1')
8
9 df <- fread("DATA_month.csv")
10 as.data.table(df)
11 #####
12 ## Identify the risk factors
13 #####
14
15 #Need to remove the predictable part to the endogeneous and exogeneous series
16
17 #upload the monthly data with the yearly factors and transform into a list of TS
18 data = read.csv('Monthly_series.csv')
19 data$Date <- as.Date(data$Date)
20
21 filtered_data <- data %>% #Filter rows between 2005 and 2022
22   filter(Date >= as.Date("2005-01-01") & Date <= as.Date("2022-12-31"))
23 time_series_cols <- filtered_data %>%
24   select(-Date)
25 time_series_list <- lapply(time_series_cols, function(col) {
26   ts_values <- ts(col, start = c(year(min(filtered_data$Date)), month(min(filtered_data$Date))
27     ), frequency = 12)
28   return(ts_values)
29 })
30 names(time_series_list) <- names(time_series_cols)
31
32 #plot series
33 time_series_df <- as.data.frame(time_series_list)
34 time_series_df$Date <- time(time_series_list[[1]])
35 time_series_long <- pivot_longer(time_series_df, cols = -Date, names_to = "Series", values_to
36   = "Value")
37 ggplot(time_series_long, aes(x = Date, y = Value, color = Series)) +
38   geom_line() +
39   facet_wrap(~ Series, scales = "free_y") +
40   labs(x = "Time", y = "Value")
41
42 ##check stationarity
43
44 lv = filtered_data[,c("EuroUSD", "CAC40", "infl", "pib", "Breakeven", "rkfreeAAA", 'oat')]
```

```

44
45 lv.adf.ln.trend = list(
46   XR = ur.df(lv$EuroUSD, type='trend', selectlags = c('BIC')),
47   cac = ur.df(lv$CAC40, type='trend', selectlags = c('BIC')),
48   infl = ur.df(lv$infl, type='trend', selectlags = c('BIC')),
49   pib = ur.df(lv$pib, type='trend', selectlags = c('BIC')),
50   #breakeven = ur.df(lv$Breakeven_no_na, type='trend', selectlags = c('BIC')),
51   rfAAA = ur.df(lv$rkfreeAAA, type='trend', selectlags = c('BIC')),
52   oat = ur.df(lv$oat, type='trend', selectlags = c('BIC'))
53 )
54
55 summary(lv.adf.ln.drift$XR)
56 print("levelVariable with drift and trend")
57 test = cbind(t(lv.adf.ln.trend$XR@teststat), t(lv.adf.ln.trend$cac@teststat),
58             t(lv.adf.ln.trend$infl@teststat), t(lv.adf.ln.trend$pib@teststat),
59             t(lv.adf.ln.trend$rfAAA@teststat), t(lv.adf.ln.trend$oat@teststat),
60             lv.adf.ln.trend$XR@cval)
61 #stargazer(test, out = 'Tables/trend_macro.tex')
62
63 lv.adf.ln.drift = list(
64   XR = ur.df(lv$EuroUSD, type='drift', selectlags = c('BIC')),
65   cac = ur.df(lv$CAC40, type='drift', selectlags = c('BIC')),
66   infl = ur.df(lv$infl, type='drift', selectlags = c('BIC')),
67   pib = ur.df(lv$pib, type='drift', selectlags = c('BIC')),
68   #breakeven = ur.df(lv$Breakeven_no_na, type='drift', selectlags = c('BIC')),
69   rfAAA = ur.df(lv$rkfreeAAA, type='drift', selectlags = c('BIC')),
70   oat = ur.df(lv$oat, type='drift', selectlags = c('BIC'))
71 )
72 print("levelVariable with drift ")
73 stat_macro_drift = cbind(t(lv.adf.ln.drift$XR@teststat), t(lv.adf.ln.drift$cac@teststat),
74                          t(lv.adf.ln.drift$infl@teststat), t(lv.adf.ln.drift$pib@teststat),
75                          t(lv.adf.ln.drift$rfAAA@teststat), t(lv.adf.ln.drift$oat@teststat),
76                          lv.adf.ln.drift$XR@cval)
77 #stargazer(stat_macro_drift, out='Tables/drift_macro.tex')
78
79 lv.adf.ln.none = list(
80   XR = ur.df(lv$EuroUSD, type='none', selectlags = c('BIC')),
81   cac = ur.df(lv$CAC40, type='none', selectlags = c('BIC')),
82   infl = ur.df(lv$infl, type='none', selectlags = c('BIC')),
83   pib = ur.df(lv$pib, type='none', selectlags = c('BIC')),
84   #breakeven = ur.df(lv$Breakeven_no_na, type='none', selectlags = c('BIC')),
85   rfAAA = ur.df(lv$rkfreeAAA, type='none', selectlags = c('BIC')),
86   oat = ur.df(lv$oat, type='none', selectlags = c('BIC'))
87 )
88 print("levelVariable with none ")
89 stat_macro_none = cbind(t(lv.adf.ln.none$XR@teststat), t(lv.adf.ln.none$cac@teststat),
90                        t(lv.adf.ln.none$infl@teststat), t(lv.adf.ln.none$pib@teststat),
91                        t(lv.adf.ln.none$rfAAA@teststat), t(lv.adf.ln.none$oat@teststat),

```

```

92         lv.adf.ln.none$XR@cval)
93 #stargazer(stat_macro_none, out='Tables/drift_none.tex')
94
95
96
97
98 #differentiate
99 filtered_data$logGDP = log(filtered_data$pib)
100
101 filtered_data$logGDP <- c(NA, diff(filtered_data$logGDP))
102 filtered_data$DCAC40 <- c(NA, diff(filtered_data$CAC40))
103 filtered_data$DEuroUSD <- c(NA, diff(filtered_data$EuroUSD))
104 filtered_data$Dinfl <- c(NA, diff(filtered_data$infl))
105 filtered_data$Doat <- c(NA, diff(filtered_data$oat))
106 filtered_data$DrfAAA <- c(NA, diff(filtered_data$rkfreeAAA))
107
108 time_series_cols <- filtered_data %>%
109   select(-Date)
110 time_series_list <- lapply(time_series_cols, function(col) {
111   ts_values <- ts(col, start = c(year(min(filtered_data$Date)), month(min(filtered_data$Date))
112     ), frequency = 12)
113   return(ts_values)
114 })
115 names(time_series_list) <- names(time_series_cols)
116
117 #plot series
118 time_series_df <- as.data.frame(time_series_list)
119 time_series_df$Date <- time(time_series_list[[1]])
120 time_series_long <- pivot_longer(time_series_df, cols = -Date, names_to = "Series", values_to
  = "Value")
121 ggplot(time_series_long, aes(x = Date, y = Value, color = Series)) +
122   geom_line() +
123   facet_wrap(~ Series, scales = "free_y") +
124   labs(x = "Time", y = "Value")
125
126 # Auto arima differentiates as much as needed to get stationary variables
127
128 logGDP_arima = auto.arima(filtered_data$logGDP) #ARIMA 0,1,0
129 logGDP_arima
130 EuroUSD_arima = auto.arima(filtered_data$EuroUSD) #ARIMA 0,1,1
131 EuroUSD_arima
132 infl_arima = auto.arima(filtered_data$infl) #ARIMA 1,1,1
133 infl_arima
134 breakeven_arima = auto.arima(filtered_data$Breakeven) #ARIMA 2,1,2
135 breakeven_arima
136
137 timestamps <- filtered_data$Date

```



```

138 res_pib = resid(logGDP_arima)
139 res_xr = resid(EuroUSD_arima)
140 res_infl = resid(infl_arima)
141 res_bkeven = resid(breakeven_arima)
142 min_length <- min(length(res_pib), length(res_xr), length(res_infl))
143 # Create a dataframe with aligned timestamps and residuals padded with NA for breakeven
144 residuals_df <- data.frame(
145   Date = timestamps[1:min_length],
146   res_pib = c(res_pib[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length)))
147   ,
148   res_xr = c(res_xr[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length))),
149   res_infl = c(res_infl[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length)
150   )),
151   res_bkeven = c(res_bkeven[1:min_length], rep(NA, times = max(0, length(timestamps) - min_
152   length)))
153 )
154 #####
155 ## Estimate the beta coefficients
156 #####
157
158 data_firm = read.csv('Firm_monthly.csv')
159 data_firm$Date <- as.Date(data_firm$Date)
160
161 filtered_data_firm <- data_firm %>% #Filter rows between 2005 and 2022
162   filter(Date >= as.Date("2005-01-01") & Date <= as.Date("2022-12-31"))
163 time_series_cols_firm <- filtered_data_firm %>%
164   select(-Date)
165 time_series_list_firm <- lapply(time_series_cols_firm, function(col) {
166   ts_values_firm <- ts(col, start = c(year(min(filtered_data_firm$Date)), month(min(filtered_
167   data_firm$Date))), frequency = 12)
168   return(ts_values_firm)
169 })
170 names(time_series_list_firm) <- names(time_series_cols_firm)
171
172 ## to get return, differenciate value of the stock (within each company)
173 filtered_data_firm <- filtered_data_firm[order(filtered_data_firm$Company, filtered_data_firm$
174   Date), ]
175 filtered_data_firm$Return <- with(filtered_data_firm, ave(value, Company, FUN = function(x) c(
176   NA, diff(x))))
177
178 finalmonthly =merge(filtered_data_firm, filtered_data, by = "Date")
179 finalmonthly =merge(finalmonthly, residuals_df, by = "Date")

```

```

180 dta_bystock = split(finalmonthly, finalmonthly$Company)
181
182 fit_lm <- function(data) {
183   lm(Return ~ res_pib + res_xr + res_infl, data = data)
184 }
185 regs_beta <- lapply(dta_bystock, function(subset) fit_lm(subset))
186
187 summary(regs_beta)
188 stargazer(regs_beta,
189           title = "Estimate the beta coefficients for each exogeneous factor",
190           column.labels = names(regs_beta),
191           out="Tables/betas_exo.tex")
192
193
194 #Include endofactor -> start in 2013
195 fit_lm_endo <- function(data) {
196   lm(Return ~ res_pib + res_xr + res_infl+ res_bkeven, data = data)
197 }
198 regs_beta_endo <- lapply(dta_bystock, function(subset) fit_lm_endo(subset))
199 stargazer(regs_beta_endo,
200           title = "Estimate the beta coefficients for exogeneous and endogeneous factors
201             (2013-2022)",
202           column.labels = names(regs_beta),
203           out="Tables/betas_exo_endo.tex")
204
205
206 #Include FF factors
207 fit_lm_exoff <- function(data) {
208   lm(Return ~ res_pib + res_xr + res_infl+ HML+SMB+Mkt.RF, data = data)
209 }
210 regs_beta_exoff <- lapply(dta_bystock, function(subset) fit_lm_exoff(subset))
211
212 summary(regs_beta_exoff)
213 stargazer(regs_beta_exoff,
214           title = "Estimate the beta coefficients for each exogeneous factor and French and
215             Fama factors",
216           column.labels = names(regs_beta),
217           out="Tables/betas_exo_ff.tex")
218
219
220 ##### FF model with annual data and our estimation for this sample
221 Firm_yearly <- read_csv("Firm_yearly.csv")
222 Firm_yearly = merge(Firm_yearly,
223                     Firm_yearly%>%
224                       filter(!is.na(MarketCap))%>%
225                       group_by(Date)%>%
226                       dplyr::summarise(MedianMCt=median(MarketCap)),
227                     by="Date")
228 DiffMC = merge(

```

```

226 Firm_yearly%>%
227   filter(!is.na(MarketCap) & MarketCap>MedianMCt & !is.na(value))%>%
228   group_by(Date)%>%
229   dplyr::summarise(AvgTop = mean(value)),
230 Firm_yearly%>%
231   filter(!is.na(MarketCap) & MarketCap<MedianMCt & !is.na(value))%>%
232   group_by(Date)%>%
233   dplyr::summarise(AvgBot = mean(value)),
234   by = "Date")
235 DiffMC$SMBest = DiffMC$AvgTop-DiffMC$AvgBot
236
237 Firm_yearly = merge(Firm_yearly,
238                     Firm_yearly%>%
239                       filter(!is.na(BookMarket))%>%
240                       group_by(Date)%>%
241                       dplyr::summarise(MedianBmt=median(BookMarket)),
242                     by="Date")
243 DiffBM = merge(
244   Firm_yearly%>%
245     filter(!is.na(BookMarket) & BookMarket>MedianBmt & !is.na(value))%>%
246     group_by(Date)%>%
247     dplyr::summarise(AvgTop = mean(value)),
248   Firm_yearly%>%
249     filter(!is.na(BookMarket) & BookMarket<MedianBmt & !is.na(value))%>%
250     group_by(Date)%>%
251     dplyr::summarise(AvgBot = mean(value)),
252   by = "Date")
253 DiffBM$HMLest = DiffBM$AvgTop-DiffBM$AvgBot
254
255 Firm_yearly =merge(Firm_yearly, DiffBM, by = "Date")
256 Firm_yearly =merge(Firm_yearly, DiffMC, by = "Date")
257
258 Data_year = read_csv("DATA_yearly.csv")
259 Firm_yearly =merge(Firm_yearly, Data_year, by = "Date")
260 Firm_yearly$Date = as.Date(Firm_yearly$Date)
261
262 Firm_yearly <- Firm_yearly[order(Firm_yearly$Company.x, Firm_yearly$Date), ]
263 Firm_yearly$return <- with(Firm_yearly, ave(value.x, Company.x, FUN = function(x) c(NA, diff(x))))
264
265 Firm_yearly$DCAC40 = c(NA, diff(Firm_yearly$CAC40))
266 Firm_yearly$Doat = c(NA, diff(Firm_yearly$Doat))
267 Firm_yearly$mkt_free = Firm_yearly$DCAC40 - Firm_yearly$Doat
268
269 filtered_Firm_yearly <- Firm_yearly %>% #Filter rows between 2010 and 2022 (no missing values)
270   filter(Date >= as.Date("2010-01-01") & Date <= as.Date("2022-12-31"))
271
272 fit_lm_myFFonly <- function(data) {

```

```

273   lm(Return ~ mkt_free+HMLest+SMBest, data = data)
274 }
275 year_by_stock_FF = split(filtered_Firm_yearly, filtered_Firm_yearly$Company.x)
276 regs_beta_myFF <- lapply(year_by_stock_FF, function(subset) fit_lm_myFFonly(subset))
277
278 stargazer(regs_beta_myFF,
279           title = "Estimate the beta coefficients for computed French and Fama factors
280                 (2010-2022)",
281           column.labels = names(regs_beta_myFF),
282           out="Tables/betas_myff.tex")
283
284 #####
285 ## Estimate the lambdas
286 #####
287
288
289
290
291
292
293
294
295
296
297
298
299
300 #####
301 #Test the validity of the multi-beta relationship
302 #####

```

## References

- Nai-Fu Chen, Richard Roll, and Stephen A. Ross. Economic Forces and the Stock Market. *The Journal of Business*, 59(3):383–403, 1986. ISSN 0021-9398.
- Eugene F. Fama and Kenneth R. French. The Cross-Section of Expected Stock Returns. *The Journal of Finance*, 47(2):427–465, 1992. ISSN 0022-1082. doi: 10.2307/2329112.
- Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, February 1993. ISSN 0304-405X. doi: 10.1016/0304-405X(93)90023-5.
- Stephen A Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360, December 1976. ISSN 0022-0531. doi: 10.1016/0022-0531(76)90046-6.