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[1] 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[2] 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

[1] Let $j \in \{1, ..., J\}$ 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

[2] 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[3] 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

[3]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	TFI.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[4], 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

[4] 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 $\forall j \in \{1, ..., N\}$:

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

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:

i. The investor faces centered idiosyncratic risks u_j, $\mathbb{E}[u_j] = 0$ that are assumed to be completely diversiable with a portfolio "large enough" (N big) because they are independent of each other $u_j \perp\!\!\!\perp u_{j'} \forall j \neq j'$.

ii. 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 β_{jk}. 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 $corr(u_j, f_k) = 0, \forall j, k$.

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[5], 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[6]) 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.

[5]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.

[6]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% (*Obligatiosn 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[7]. 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.

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

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 sentitivities 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[8]. We will also pose that $f_k \perp\!\!\!\perp f_{k'}$ $\forall k, k'$ in

[7] Note that the Breakeven inflation is the inflation rate that would equalize the return of these two sovereign bonds (due to AOA).
[8] The residuals will be uncorrelated with the explanatory variables

order to have a good identification of the parameters with linear regression estimators $\forall k_0,\ \beta_{j,k0} = \frac{cov(R_j, f_{k0})}{Var(f_{k0})} \equiv \hat{\beta}$.

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

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

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.

9

Table 2: Estimate the beta coefficients for each exogeneous factor

	accor	air-france-klm	air-liquide	airbus	bic	biomerieux	bouygues	capgemini	carrefour	casino	saint-gobain	danone	dass
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(1
res_pib	4.417	87.731	6.695	13.047	−11.895	8.568	−49.367***	25.025	5.304	−19.160	−22.502	−52.184	0.2
	(16.882)	(171.635)	(8.268)	(27.583)	(10.889)	(15.496)	(13.562)	(22.659)	(22.047)	(12.504)	(66.982)	(35.688)	(1.6
res_xr	12.577	9.750	6.565	18.735	−13.262**	0.480	1.404	−8.086	10.413	−12.388*	25.959	−5.407	−0.4
	(9.864)	(100.281)	(4.830)	(16.116)	(6.362)	(9.054)	(7.924)	(13.239)	(12.881)	(7.306)	(39.135)	(20.852)	(0.9
res_infl	1.383	−0.111	0.556	−0.840	1.128**	0.239	0.925	2.895**	3.652***	0.252	2.233	1.565	0.0
	(0.880)	(8.947)	(0.431)	(1.438)	(0.568)	(0.808)	(0.707)	(1.181)	(1.149)	(0.652)	(3.492)	(1.860)	(0.0
Constant	−0.031	6.689**	0.059	−0.097	0.251	0.081	0.563***	0.366	0.556	0.196	3.082***	1.418**	0.0
	(0.263)	(2.674)	(0.129)	(0.430)	(0.170)	(0.241)	(0.211)	(0.353)	(0.343)	(0.195)	(1.044)	(0.556)	(0.0
Observations	215	215	215	215	215	215	215	215	215	215	215	215	21
R^2	0.021	0.001	0.020	0.009	0.042	0.002	0.068	0.033	0.050	0.025	0.005	0.014	0.0
Adjusted R^2	0.007	−0.013	0.006	−0.005	0.029	−0.012	0.055	0.019	0.037	0.011	−0.009	0.0001	−0.0
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	0.3
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	0.4

Note:

9

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	air-france-klm	air-liquide	airbus	bic	biomerieux
	(1)	(2)	(3)	(4)	(5)	(6)
res_pib	47.225	−114.811	−0.127	252.848	22.185	36.520
	(99.831)	(239.823)	(42.066)	(208.506)	(45.090)	(45.088)
res_xr	14.768	23.779	10.730**	20.465	−4.871	2.910
	(11.125)	(26.725)	(4.688)	(23.235)	(5.025)	(5.024)
res_infl	1.153	−2.827	−0.124	−2.077	−0.075	−0.113
	(1.330)	(3.196)	(0.561)	(2.779)	(0.601)	(0.601)
res_bkeven	65.720	−571.630	−213.246	−1,300.032*	−53.868	2.813
	(341.840)	(821.196)	(144.043)	(713.961)	(154.397)	(154.389)
Constant	0.084	1.753**	0.212	−0.061	0.188	0.007
	(0.357)	(0.859)	(0.151)	(0.747)	(0.161)	(0.161)
Observations	109	109	109	109	109	109
R^2	0.036	0.018	0.064	0.061	0.013	0.011
Adjusted R^2	−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	air-france-klm	air-liquide	airbus	bic	biomerieux
	(1)	(2)	(3)	(4)	(5)	(6)
res_pib	9.008	197.078	8.992	22.914	-9.534	14.176
	(14.726)	(168.809)	(7.026)	(24.369)	(10.381)	(15.545)
res_xr	-21.529^{**}	-85.560	-10.601^{**}	-32.113^{**}	-27.630^{***}	-8.154
	(9.354)	(107.234)	(4.463)	(15.480)	(6.594)	(9.875)
res_infl	0.344	-3.228	-0.002	-2.571^{**}	0.792	-0.103
	(0.767)	(8.795)	(0.366)	(1.270)	(0.541)	(0.810)
HML	0.331^{***}	-2.902^{***}	0.198^{***}	0.524^{***}	0.034	-0.033
	(0.093)	(1.066)	(0.044)	(0.154)	(0.066)	(0.098)
SMB	0.362^{***}	1.365	0.261^{***}	0.942^{***}	-0.075	0.262^{*}
	(0.127)	(1.459)	(0.061)	(0.211)	(0.090)	(0.134)
Mkt.RF	0.296^{***}	1.888^{***}	0.134^{***}	0.401^{***}	0.172^{***}	0.095^{*}
	(0.049)	(0.558)	(0.023)	(0.080)	(0.034)	(0.051)
Constant	-0.211	5.396^{**}	-0.029	-0.376	0.165	0.005
	(0.228)	(2.614)	(0.109)	(0.377)	(0.161)	(0.241)
Observations	215	215	215	215	215	215
R^2	0.286	0.075	0.322	0.259	0.166	0.038
Adjusted R^2	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	air-france-klm	air-liquide	airbus	bic	biomerieux	bou
	(1)	(2)	(3)	(4)	(5)	(6)	(
mkt_free	0.012***	0.180***	0.007***	0.010***	0.006***	−0.0004	0.0
	(0.001)	(0.010)	(0.0004)	(0.001)	(0.001)	(0.001)	(0.
HMLest	−0.006	−0.017	−0.001	0.004	0.007*	0.005	0.
	(0.005)	(0.057)	(0.002)	(0.006)	(0.004)	(0.007)	(0.
SMBest	−0.032**	0.127	−0.006	−0.012	0.014	0.004	0.0
	(0.013)	(0.148)	(0.006)	(0.016)	(0.010)	(0.019)	(0.
Constant	0.133	1.124	0.008	−0.145	−0.011	0.010	−0
	(0.121)	(1.374)	(0.053)	(0.152)	(0.094)	(0.176)	(0.
Observations	390	390	390	390	390	390	3
R^2	0.349	0.469	0.467	0.192	0.156	0.002	0.
Adjusted R^2	0.344	0.465	0.463	0.186	0.150	−0.006	0.
Residual Std. Error (df = 386)	1.673	18.957	0.731	2.102	1.301	2.429	1.
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 $\exists \rho, \exists \lambda_1, ..., \lambda_k$:

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

where each λ_k is the parameter representing the market price of risk (the risk premium) that the market retributes 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

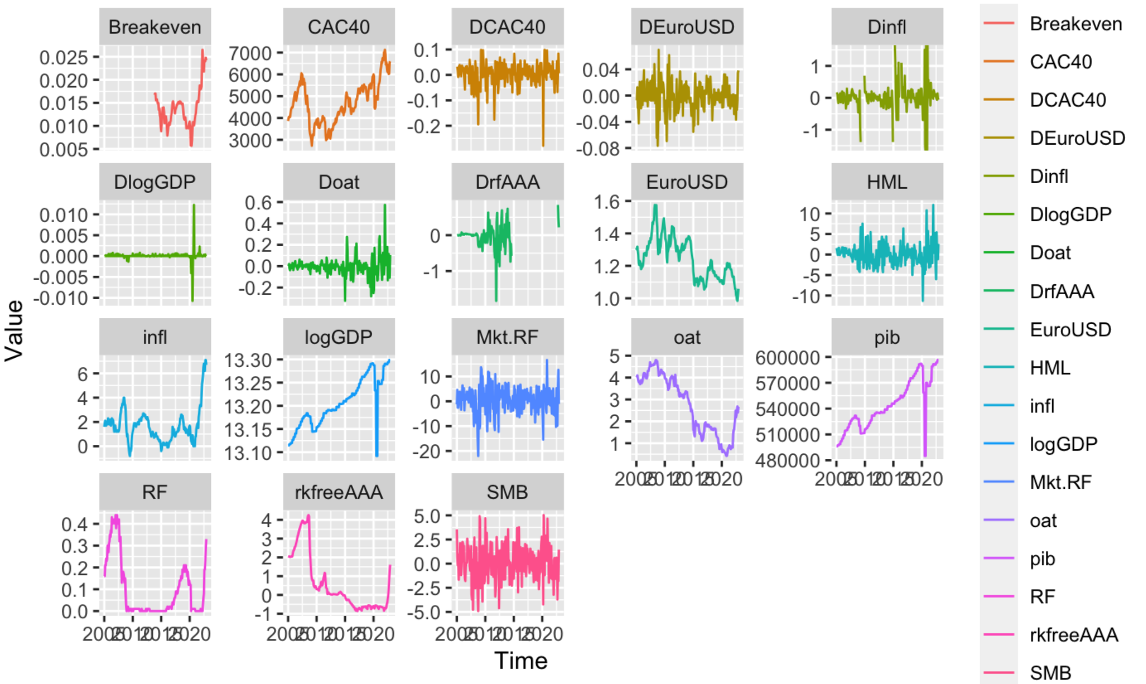

Figure 1: Factors: Series in level and in deltas

Appendix B Code - Data gathering and cleaning in Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
AP1 - Data gathering and Data Cleaning

Scrapping - Firm level data for French and Fama factors
Use Yahoo Finance API to get the financial data for all the stocks
Use Eurostat API to get the macro data
Two data sets (French-Fama factors and long term inflation expectation) are found online and
    have been downloaded in CSV file beforehang

Merge and clean the dataset

@author: nataliacardenasf
"""

import pandas as pd
import numpy as np
import os

import requests
from bs4 import BeautifulSoup

#import pandas_datareader.data as web
```

```python
import yfinance as yf
#from eurostatapiclient import EurostatAPIClient
import datetime


os.chdir('/Users/nataliacardenasf/Documents/GitHub/PROJECTS_AP_FE/AP 1')

company_names_lower = [
    'air-liquide', 'airbus', 'bouygues', 'capgemini', 'carrefour', 'casino-guichard-perrachon', 'vivendi',
    'kering', 'l-oreal', 'lvmh', 'michelin', 'orange', 'renault', 'sanofi', 'thales',
    'totalenergies', 'vinci', 'compagnie-de-saint-gobain', 'ubisoft', 'tf1', 'danone',
    'dassault-aviation', 'air-france-klm', 'accor', 'bic', 'hermes-international',
    'jcdecaux', 'nexans', 'sodexo', 'biomerieux', "CAC40", "EuroUSD"]

tickers = [
    'AI.PA', 'AIR.PA', 'EN.PA', 'CAP.PA', 'CA.PA', 'CO.PA', 'VIV.PA', 'KER.PA', 'OR.PA', 'MC.PA',
    'ML.PA', 'ORA.PA', 'RNO.PA', 'SAN.PA', 'HO.PA', 'TTE.PA', 'DG.PA', 'SGO.PA', 'UBI.PA', 'TFI.PA',
    'BN.PA', 'AM.PA', 'AF.PA', 'AC.PA', 'BB.PA', 'RMS.PA', 'DEC.PA', 'NEX.PA', 'SW.PA', 'BIM.PA', "^FCHI", 'EURUSD=X']



#%% Scrap firm level data for French and Fama factors
# List of companies' URLs
mktcap_urls = ['https://companiesmarketcap.com/'+ x+'/marketcap/' for x in company_names_lower[:-2]]
pricebook_urls = ['https://companiesmarketcap.com/'+ x+'/pb-ratio/' for x in company_names_lower[:-2]]


# Scrapping functions
def scrape_market_cap(url, company_name):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        table_body = soup.find('table', class_='table').find('tbody')
        if table_body:
            data = []
            rows = table_body.find_all('tr')
            for row in rows:
                cols = row.find_all('td')
                if len(cols) >= 2:
                    year = cols[0].text.strip()
                    market_cap = cols[1].text.strip()
                    #variation = cols[2].text.strip()
```

```python
                    data.append({'Year': year, 'MarketCap': market_cap, 'Company': company_name})
            return pd.DataFrame(data)
    return None

def scrape_price_book(url, company_name):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        table_body = soup.find('table', class_='table').find('tbody')
        if table_body:
            data = []
            rows = table_body.find_all('tr')
            for row in rows:
                cols = row.find_all('td')
                if len(cols) >= 2:
                    year = cols[0].text.strip()
                    pricebook = cols[1].text.strip()
                    #variation = cols[2].text.strip()
                    data.append({'Year': year, 'PriceBook': pricebook, 'Company': company_name})
            return pd.DataFrame(data)
    return None


# Scraping market cap data
dfmktcap = pd.DataFrame()
for url, company in zip(mktcap_urls, company_names_lower[:-2]):
    data = scrape_market_cap(url, company)
    if data is not None:
        dfmktcap = pd.concat([dfmktcap, data])

#Scap price book
dfpricebook = pd.DataFrame()
for url, company in zip(pricebook_urls, company_names_lower[:-2]):
    data = scrape_price_book(url, company)
    if data is not None:
        dfpricebook = pd.concat([dfpricebook, data])

#indexes
dfmktcap['Year'] = pd.to_datetime(dfmktcap['Year'])
dfmktcap['Year'] = pd.DatetimeIndex(dfmktcap['Year']).year

dfpricebook['Year'] = pd.to_datetime(dfpricebook['Year'])
dfpricebook['Year'] = pd.DatetimeIndex(dfpricebook['Year']).year


##Merge datasets
```

```python
final_firm = dfmktcap.copy()
final_firm = final_firm.merge(dfpricebook, how='outer', on=['Year', 'Company'])


del dfmktcap, dfpricebook, mktcap_urls, pricebook_urls, url, data, company


missing = final_firm[final_firm.isna().any(axis=1)]
missing = missing.sort_values(by=['Year'])
missing = missing.reset_index()
#have both data points for all firms for 2010-2022
# in 09 only missing data is from BIC, Carrefour, Ubisolft, AirFrance

#remove 2023
final_firm = final_firm[final_firm.Year != 2023]

#Get book to market ratio = inverse of price-book ratio
final_firm['PriceBook'] = pd.to_numeric(final_firm['PriceBook'], errors='coerce')
final_firm['PriceBook'].replace('nan', np.nan, inplace=True)
final_firm['BookMarket']  = final_firm['PriceBook'].apply(lambda x: x ** -1 if not pd.isnull(x
    ) else np.nan)

#Clean MarketCap
final_firm['MarketCap'] = (final_firm['MarketCap'].replace({'\$': '', ' B': ''}, regex=True).
    astype(float) * 1_000)  # Clear the letters, convert to float and scale to millions

#Date format
final_firm['Year'] = pd.to_datetime(final_firm['Year'], format='%Y')

#====Get monthly data
monthly_data = pd.DataFrame()
# Repeat the yearly data for each month and each firm
for index, row in final_firm.iterrows():
    firm_data = pd.DataFrame()
    monthly_year = pd.date_range(start=row['Year'], periods=12, freq='MS')
    firm_data['Date'] = monthly_year
    firm_data['Company'] = row['Company']
    firm_data['MarketCap'] = row['MarketCap']
    firm_data['BookMarket'] = row['BookMarket']
    firm_data['PriceBook'] = row['PriceBook']
    monthly_data = pd.concat([monthly_data, firm_data])

del index, monthly_year, row, firm_data

firms_year = final_firm.copy()
firms_month = monthly_data.copy()

del final_firm, monthly_data, missing
```

```python

#%%Get return data with Yahoo finance

start = datetime.datetime(2002, 1, 1)
end = datetime.datetime(2022, 12, 31)

#Get all data
data = yf.download(tickers, start=start,
                end=end)

#Focus on adjusted closed values only
adjclose=data['Adj Close']
adjclose = adjclose.set_axis(company_names_lower, axis=1)

#Use monthly data: mean of the months value
adjclose = adjclose.resample('1M').mean(numeric_only=True)
adjclose_y = adjclose.resample('1Y').mean(numeric_only=True)

## extract CAC40 and exchange rate
cac_xrate_month = adjclose.loc[:, ["CAC40", "EuroUSD"]]
cac_xrate_year = adjclose_y.loc[:, ["CAC40", "EuroUSD"]]
adjclose = adjclose.drop(columns=["CAC40", "EuroUSD"])
adjclose_y = adjclose_y.drop(columns=["CAC40", "EuroUSD"])

#Reshape
prices_monthly = pd.melt(adjclose, value_vars=company_names_lower[0:-2], ignore_index=False)
prices_yearly = pd.melt(adjclose_y, value_vars=company_names_lower[0:-2], ignore_index=False)


del data, adjclose, adjclose_y

#I'm not getting right values for xrate when dowloading in bulk
data = yf.download(['EURUSD=X', '^FCHI'], start=start, end=end)
adjclose=data['Adj Close']
adjclose = adjclose.set_axis(['EuroUSD','CAC40'], axis=1)
#adjclose = pd.DataFrame(adjclose, columns=['Date', 'EuroUSD'])

cac_xrate_month = adjclose.resample('1M').mean(numeric_only=True)
cac_xrate_year = adjclose.resample('1Y').mean(numeric_only=True)

del data, adjclose, start, end


#%% Endogeneous factor: long term inflation expectation from external file

#upload the Agence France Tresor data
pi_endo= pd.read_excel('2023_11_01_rend_tit_ref_oatei.xls', skiprows=[0,1,2,3,4], usecols
    =[0,3])
```

```
pi_endo.columns = ['Date', "Breakeven"]

pi_endo["Date"] = pd.to_datetime(pi_endo["Date"])
pi_endo= pi_endo.set_index(pi_endo["Date"])
pi_endo.drop(columns=['Date'])

#get monthly data
piendo_month = pi_endo.resample('1M').mean(numeric_only=True)

#get yearly data
piendo_year = pi_endo.resample('1Y').mean(numeric_only=True)

del pi_endo

#%% French and Fama - their data

df = pd.read_csv('Europe_3_Factors.csv',skiprows=[0,1,2])

#montly data, need to fix dates
frenchfama_month = df.iloc[:399,:]

frenchfama_month['Unnamed: 0'] = frenchfama_month['Unnamed: 0'].astype(str)  # Convert to
    string for manipulation
frenchfama_month['Year'] = frenchfama_month['Unnamed: 0'].str[:4]  # Extract year from the
    encoded date
frenchfama_month['Month'] = frenchfama_month['Unnamed: 0'].str[4:]  # Extract month from the
    encoded date
frenchfama_month['Date'] = pd.to_datetime(dict(year=frenchfama_month['Year'], month=
    frenchfama_month['Month'], day=1))
frenchfama_month.drop(['Year', 'Month', 'Unnamed: 0'], axis=1, inplace=True)
frenchfama_month = frenchfama_month.set_index(frenchfama_month['Date'])
frenchfama_month = frenchfama_month.drop(columns=["Date"])
frenchfama_month = frenchfama_month.loc['2002-01-01':]
frenchfama_month = frenchfama_month.astype(float)


#yearly data
frenchfama_year = df.iloc[402:,:]
frenchfama_year["Unnamed: 0"] = pd.to_datetime(frenchfama_year['Unnamed: 0'])
frenchfama_year.rename(columns={"Unnamed: 0":'Date'}, inplace=True)
frenchfama_year = frenchfama_year.set_index(frenchfama_year['Date'])
frenchfama_year = frenchfama_year.drop(columns=["Date"])
frenchfama_year = frenchfama_year.loc['2002-01-01':]
frenchfama_year = frenchfama_year.astype(float)

del df
```

```
#%% Macro data
#APIs didn't work as planned

## PIB Q
pib = pd.read_csv("ECB_PIB.csv")
pib.columns = ['Date', 'Q', 'pib']
pib['Date'] = pd.to_datetime(pib['Date'])
pib = pib.drop(columns=['Q'])
pib = pib.set_index(pib['Date'])
pib = pib.loc['2002-01-01':]
pib = pib.drop(columns=['Date'])

#monthly
pib_monthly = pib.resample('MS').ffill()
#yearly
pib_year = pib.resample('1Y').last()

del pib

##risk free AAA
rkfreeAAA = pd.read_csv('ECB_yield.csv')
rkfreeAAA.columns=['Date', "time" ,"rkfreeAAA"]
rkfreeAAA['Date'] = pd.to_datetime(rkfreeAAA['Date'])
rkfreeAAA = rkfreeAAA.set_index(rkfreeAAA['Date'])
rkfreeAAA = rkfreeAAA.drop(columns=['time', "Date"])

rkfreeAAA_monthly = rkfreeAAA.resample("1M").mean(numeric_only=True)
rkfreeAAA_year = rkfreeAAA.resample("1Y").mean(numeric_only=True)

del rkfreeAAA

##HICP
pi_month = pd.read_csv("HICP.csv", sep=';', encoding = 'latin1',skiprows=[0,1,2,3], usecols
    =[0,1], header=None)
pi_month.columns= ['Date', "infl"]
pi_month["Date"] = pd.to_datetime(pi_month['Date'], format = "%Y-%m")
pi_month=pi_month.set_index(pi_month["Date"])
pi_month = pi_month.drop(columns=['Date'])

#yearly
pi_year = pi_month.resample("1Y").mean(numeric_only=True)


##OAT
oat = pd.read_csv('OAT.csv', sep=';', skiprows=[0,1,2,3,4,5], usecols=[0,5], header=None)
oat.columns = ['Date', 'oat']
oat['Date'] = pd.to_datetime(oat["Date"])
oat = oat.set_index(oat["Date"])
```

```
oat = oat.drop(columns=["Date"])
oat = oat.loc["2002-01-01":]

oat['oat'] = oat['oat'].replace("-", np.nan)
oat['oat'] = oat['oat'].str.replace(',', '.').astype(float)

oat_month = oat.resample('M').mean(numeric_only=True)
oat_year = oat.resample('1Y').mean(numeric_only=True)

del oat


#%% Merge the dataframes

#===MONTHLY
#macro stuff, only date index matter
monthly = pd.concat([cac_xrate_month, frenchfama_month, oat_month, pi_month, pib_monthly,
    piendo_month, rkfreeAAA_monthly], axis=1)
monthly= monthly.resample('M').last() #some tables encoded end of month, others on the 1st

monthly.to_csv('Monthly_series.csv')


#firm specific data
firms_month['Date'] = firms_month['Date'] + pd.offsets.MonthEnd(0) #all other df have eomonth
    date
firms_month = firms_month.set_index(['Date'])
firms_month = firms_month.set_index('Company', append=True)

prices_monthly.columns = ['Company', 'value']
prices_monthly.set_index('Company', append=True)

monthly_stock = pd.merge(prices_monthly.reset_index(), firms_month.reset_index(), on =["Date",
     "Company"], how='outer').set_index(["Date", "Company"])
monthly_stock.to_csv("Firm_monthly.csv")

#merge the two
merged_monthly = pd.merge(monthly, monthly_stock, left_index=True, right_index=True, how='
    right')

#Export df in csv
merged_monthly.to_csv("DATA_month.csv")



#=== YEARLY
yearly = pd.concat([cac_xrate_year, frenchfama_year, oat_year, pi_year, pib_year, piendo_year,
     rkfreeAAA_year], axis=1)
```

```
yearly = yearly.resample('Y').last()

yearly.to_csv('Yearly_series.csv')

# =============================================================================
# #firm specific
# firms_year.rename(columns={"Year":'Date'}, inplace=True)
# firms_year['Date'] = firms_year['Date'] + pd.offsets.MonthEnd(0)
# firms_year = firms_year.set_index(['Date'])
# firms_year = firms_year.set_index('Company', append=True)
#
# prices_yearly.columns = ['Company', 'value']
# prices_yearly = prices_yearly.set_index('Company', append=True)
#
# yearly_stock = pd.merge(prices_yearly.reset_index(), firms_year.reset_index(), on =["Date",
    "Company"], how='outer').set_index(["Date", "Company"])
# =============================================================================

#yearly_stock = monthly_stock.groupby('Company').resample('Y').mean()

monthly_data_reset = monthly_stock.reset_index()
yearly_stock = monthly_data_reset.groupby('Company').resample('Y', on='Date').mean()
#yearly_stock = yearly_stock.set_index(["Date", "Company"])

yearly_stock.to_csv("Firm_yearly.csv")


#merge the two
merged_year = pd.merge(yearly, yearly_stock, left_index=True, right_index=True, how='right')

merged_year.to_csv("DATA_yearly.csv")
```

Appendix C Code - Analysis in R

```r
#%% AP 1
#%% ncardenasfrias

pacman::p_load(data.table, urca, tidyverse, gplots, xts, stargazer, forecast, plm, ggplot2,
    tidyr)
library(dplyr)
setwd('/Users/nataliacardenasf/Documents/GitHub/PROJECTS_AP_FE/AP 1')

df <- fread("DATA_month.csv")
as.data.table(df)

############################
## Identify the risk factors
############################

#Need to remove the predicatable part to the endogeneous and exogeneous series

  #upload the monthly data with the yearly factorsand transform into a list of TS
data = read.csv('Monthly_series.csv')
data$Date <- as.Date(data$Date)

filtered_data <- data %>% #Filter rows between 2005 and 2022
  filter(Date >= as.Date("2005-01-01") & Date <= as.Date("2022-12-31"))
time_series_cols <- filtered_data %>%
  select(-Date)
time_series_list <- lapply(time_series_cols, function(col) {
  ts_values <- ts(col, start = c(year(min(filtered_data$Date)), month(min(filtered_data$Date))
    ), frequency = 12)
  return(ts_values)
})

names(time_series_list) <- names(time_series_cols)

  #plot series
time_series_df <- as.data.frame(time_series_list)
time_series_df$Date <- time(time_series_list[[1]])
time_series_long <- pivot_longer(time_series_df, cols = -Date, names_to = "Series", values_to
    = "Value")
ggplot(time_series_long, aes(x = Date, y = Value, color = Series)) +
  geom_line() +
  facet_wrap(~ Series, scales = "free_y") +
  labs(x = "Time", y = "Value")

  ##check stationarity

lv = filtered_data[,c("EuroUSD","CAC40", "infl","pib","Breakeven","rkfreeAAA", 'oat')]
```

```
lv.adf.ln.trend = list(
  XR = ur.df(lv$EuroUSD, type='trend', selectlags = c('BIC')),
  cac = ur.df(lv$CAC40, type='trend', selectlags = c('BIC')),
  infl = ur.df(lv$infl, type='trend', selectlags = c('BIC')),
  pib = ur.df(lv$pib, type='trend', selectlags = c('BIC')),
  #breakeven = ur.df(lv$Breakeven_no_na, type='trend', selectlags = c('BIC')),
  rfAAA = ur.df(lv$rkfreeAAA, type='trend', selectlags = c('BIC')),
  oat = ur.df(lv$oat, type='trend', selectlags = c('BIC'))
)

summary(lv.adf.ln.drift$XR)
print("levelVariable with drift and trend")
test =cbind(t(lv.adf.ln.trend$XR@teststat),t(lv.adf.ln.trend$cac@teststat),
            t(lv.adf.ln.trend$infl@teststat),t(lv.adf.ln.trend$pib@teststat),
            t(lv.adf.ln.trend$rfAAA@teststat), t(lv.adf.ln.trend$oat@teststat),
            lv.adf.ln.trend$XR@cval)
#stargazer(test, out ='Tables/trend_macro.tex')

lv.adf.ln.drift = list(
  XR = ur.df(lv$EuroUSD, type='drift', selectlags = c('BIC')),
  cac = ur.df(lv$CAC40, type='drift', selectlags = c('BIC')),
  infl = ur.df(lv$infl, type='drift', selectlags = c('BIC')),
  pib = ur.df(lv$pib, type='drift', selectlags = c('BIC')),
  #breakeven = ur.df(lv$Breakeven_no_na, type='drift', selectlags = c('BIC')),
  rfAAA = ur.df(lv$rkfreeAAA, type='drift', selectlags = c('BIC')),
  oat = ur.df(lv$oat, type='drift', selectlags = c('BIC'))
)
print("levelVariable with drift ")
stat_macro_drift =cbind(t(lv.adf.ln.drift$XR@teststat),t(lv.adf.ln.drift$cac@teststat),
            t(lv.adf.ln.drift$infl@teststat),t(lv.adf.ln.drift$pib@teststat),
            t(lv.adf.ln.drift$rfAAA@teststat), t(lv.adf.ln.drift$oat@teststat),
            lv.adf.ln.drift$XR@cval)
#stargazer(stat_macro_drift, out='Tables/drift_macro.tex')

lv.adf.ln.none = list(
  XR = ur.df(lv$EuroUSD, type='none', selectlags = c('BIC')),
  cac = ur.df(lv$CAC40, type='none', selectlags = c('BIC')),
  infl = ur.df(lv$infl, type='none', selectlags = c('BIC')),
  pib = ur.df(lv$pib, type='none', selectlags = c('BIC')),
  #breakeven = ur.df(lv$Breakeven_no_na, type='none', selectlags = c('BIC')),
  rfAAA = ur.df(lv$rkfreeAAA, type='none', selectlags = c('BIC')),
  oat = ur.df(lv$oat, type='none', selectlags = c('BIC'))
)
print("levelVariable with none ")
stat_macro_none =cbind(t(lv.adf.ln.none$XR@teststat),t(lv.adf.ln.none$cac@teststat),
                        t(lv.adf.ln.none$infl@teststat),t(lv.adf.ln.none$pib@teststat),
                        t(lv.adf.ln.none$rfAAA@teststat), t(lv.adf.ln.none$oat@teststat),
```

```
                        lv.adf.ln.none$XR@cval)
#stargazer(stat_macro_none, out='Tables/drift_none.tex')




#differenciate
filtered_data$logGDP = log(filtered_data$pib)

filtered_data$DlogGDP <- c(NA, diff(filtered_data$logGDP))
filtered_data$DCAC40 <- c(NA, diff(filtered_data$CAC40))
filtered_data$DEuroUSD <- c(NA, diff(filtered_data$EuroUSD))
filtered_data$Dinfl <- c(NA, diff(filtered_data$infl))
filtered_data$Doat <- c(NA, diff(filtered_data$oat))
filtered_data$DrfAAA <- c(NA, diff(filtered_data$rkfreeAAA))

time_series_cols <- filtered_data %>%
  select(-Date)
time_series_list <- lapply(time_series_cols, function(col) {
  ts_values <- ts(col, start = c(year(min(filtered_data$Date)), month(min(filtered_data$Date))
    ), frequency = 12)
  return(ts_values)
})

names(time_series_list) <- names(time_series_cols)

#plot series
time_series_df <- as.data.frame(time_series_list)
time_series_df$Date <- time(time_series_list[[1]])
time_series_long <- pivot_longer(time_series_df, cols = -Date, names_to = "Series", values_to
    = "Value")
ggplot(time_series_long, aes(x = Date, y = Value, color = Series)) +
  geom_line() +
  facet_wrap(~ Series, scales = "free_y") +
  labs(x = "Time", y = "Value")

# Auto arima differenciates as much as needed to get stationary variables

logGDP_arima = auto.arima(filtered_data$logGDP) #ARIMA 0,1,0
logGDP_arima
EuroUSD_arima = auto.arima(filtered_data$EuroUSD) #ARIMA 0,1,1
EuroUSD_arima
infl_arima = auto.arima(filtered_data$infl) #ARIMA 1,1,1
infl_arima
breakeven_arima = auto.arima(filtered_data$Breakeven) #ARIMA 2,1,2
breakeven_arima

timestamps <- filtered_data$Date
```

```
res_pib = resid(logGDP_arima)
res_xr = resid(EuroUSD_arima)
res_infl = resid(infl_arima)
res_bkeven = resid(breakeven_arima)
min_length <- min(length(res_pib), length(res_xr), length(res_infl))
# Create a dataframe with aligned timestamps and residuals padded with NA for breakeven
residuals_df <- data.frame(
  Date = timestamps[1:min_length],
  res_pib = c(res_pib[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length))),
  res_xr = c(res_xr[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length))),
  res_infl = c(res_infl[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length))),
  res_bkeven = c(res_bkeven[1:min_length], rep(NA, times = max(0, length(timestamps) - min_length)))
)



##################################
## Estimate the beta coefficients
##################################

data_firm = read.csv('Firm_monthly.csv')
data_firm$Date <- as.Date(data_firm$Date)

filtered_data_firm <- data_firm %>% #Filter rows between 2005 and 2022
  filter(Date >= as.Date("2005-01-01") & Date <= as.Date("2022-12-31"))
time_series_cols_firm <- filtered_data_firm %>%
  select(-Date)
time_series_list_firm <- lapply(time_series_cols_firm, function(col) {
  ts_values_firm <- ts(col, start = c(year(min(filtered_data_firm$Date)), month(min(filtered_data_firm$Date))), frequency = 12)
  return(ts_values_firm)
})

names(time_series_list_firm) <- names(time_series_cols_firm)

## to get return, differenciate value of the stock (within each company)
filtered_data_firm <- filtered_data_firm[order(filtered_data_firm$Company, filtered_data_firm$Date), ]
filtered_data_firm$Return <- with(filtered_data_firm, ave(value, Company, FUN = function(x) c(NA, diff(x))))

finalmonthly =merge(filtered_data_firm, filtered_data, by = "Date")
finalmonthly =merge(finalmonthly, residuals_df, by = "Date")


```

```r
dta_bystock = split(finalmonthly, finalmonthly$Company)

fit_lm <- function(data) {
  lm(Return ~ res_pib + res_xr + res_infl, data = data)
}
regs_beta <- lapply(dta_bystock, function(subset) fit_lm(subset))

summary(regs_beta)
stargazer(regs_beta,
          title = "Estimate the beta coefficients for each exogeneous factor",
          column.labels = names(regs_beta),
          out="Tables/betas_exo.tex")


#Include endofactor -> start in 2013
fit_lm_endo <- function(data) {
  lm(Return ~ res_pib + res_xr + res_infl+ res_bkeven, data = data)
}
regs_beta_endo <- lapply(dta_bystock, function(subset) fit_lm_endo(subset))
stargazer(regs_beta_endo,
          title = "Estimate the beta coefficients for exogeneous and endogeneous factors
    (2013-2022)",
          column.labels = names(regs_beta),
          out="Tables/betas_exo_endo.tex")

#Include FF factors
fit_lm_exoff <- function(data) {
  lm(Return ~ res_pib + res_xr + res_infl+ HML+SMB+Mkt.RF, data = data)
}
regs_beta_exoff <- lapply(dta_bystock, function(subset) fit_lm_exoff(subset))

summary(regs_beta_exoff)
stargazer(regs_beta_exoff,
          title = "Estimate the beta coefficients for each exogeneous factor and French and
    Fama factors",
          column.labels = names(regs_beta),
          out="Tables/betas_exo_ff.tex")


#### FF model with annual data and our estimation for this sample
Firm_yearly <- read_csv("Firm_yearly.csv")
Firm_yearly = merge(Firm_yearly,
                    Firm_yearly%>%
                      filter(!is.na(MarketCap))%>%
                      group_by(Date)%>%
                      dplyr::summarise(MedianMCt=median(MarketCap)),
                    by="Date")
DiffMC = merge(
```

```
  Firm_yearly%>%
    filter(!is.na(MarketCap) & MarketCap>MedianMCt & !is.na(value))%>%
    group_by(Date)%>%
    dplyr::summarise(AvgTop = mean(value)),
  Firm_yearly%>%
    filter(!is.na(MarketCap) & MarketCap<MedianMCt & !is.na(value))%>%
    group_by(Date)%>%
    dplyr::summarise(AvgBot = mean(value)),
  by = "Date")
DiffMC$SMBest = DiffMC$AvgTop-DiffMC$AvgBot

Firm_yearly = merge(Firm_yearly,
                    Firm_yearly%>%
                      filter(!is.na(BookMarket))%>%
                      group_by(Date)%>%
                      dplyr::summarise(MedianBMt=median(BookMarket)),
                    by="Date")
DiffBM = merge(
  Firm_yearly%>%
    filter(!is.na(BookMarket) & BookMarket>MedianBMt & !is.na(value))%>%
    group_by(Date)%>%
    dplyr::summarise(AvgTop = mean(value)),
  Firm_yearly%>%
    filter(!is.na(BookMarket) & BookMarket<MedianBMt & !is.na(value))%>%
    group_by(Date)%>%
    dplyr::summarise(AvgBot = mean(value)),
  by = "Date")
DiffBM$HMLest = DiffBM$AvgTop-DiffBM$AvgBot

Firm_yearly =merge(Firm_yearly, DiffBM, by = "Date")
Firm_yearly =merge(Firm_yearly, DiffMC, by = "Date")

Data_year = read_csv("DATA_yearly.csv")
Firm_yearly =merge(Firm_yearly, Data_year, by = "Date")
Firm_yearly$Date = as.Date(Firm_yearly$Date)

Firm_yearly <- Firm_yearly[order(Firm_yearly$Company.x, Firm_yearly$Date), ]
Firm_yearly$Return <- with(Firm_yearly, ave(value.x, Company.x, FUN = function(x) c(NA, diff(x
    ))))

Firm_yearly$DCAC40 = c(NA, diff(Firm_yearly$CAC40))
Firm_yearly$Doat = c(NA, diff(Firm_yearly$oat))
Firm_yearly$mkt_free = Firm_yearly$DCAC40 - Firm_yearly$Doat

filtered_Firm_yearly <- Firm_yearly %>% #Filter rows between 2010 and 2022 (no missing values)
  filter(Date >= as.Date("2010-01-01") & Date <= as.Date("2022-12-31"))

fit_lm_myFFonly <- function(data) {
```

```
  lm(Return ~ mkt_free+HMLest+SMBest, data = data)
}
year_by_stock_FF = split(filtered_Firm_yearly, filtered_Firm_yearly$Company.x)
regs_beta_myFF <- lapply(year_by_stock_FF, function(subset) fit_lm_myFFonly(subset))

stargazer(regs_beta_myFF,
          title = "Estimate the beta coefficients for computed French and Fama factors
    (2010-2022)",
          column.labels = names(regs_beta_myFF),
          out="Tables/betas_myff.tex")


#######################
## Estimate the lambdas
#######################


#Extarct the beta coefficients of reg with exo and FF factors
# Initialize an empty dataframe to store coefficients and model names
coefficients_df <- data.frame(Model = character(), Intercept = numeric(),
                              res_pib = numeric(), res_xr = numeric(),
                              res_infl = numeric(), HML = numeric(),
                              SMB = numeric(), Mkt.RF = numeric(),
                              stringsAsFactors = FALSE)

# Iterate through subsets of data and fit models
for (i in seq_along(dta_bystock)) {
  subset <- dta_bystock[[i]]
  model <- fit_lm_exoff(subset)
  coefficients <- c(model$coefficients[1], model$coefficients[-1])
  row <- data.frame(Model = paste0("Model_", i),
                    Intercept = coefficients[1],
                    res_pib = coefficients[2],
                    res_xr = coefficients[3],
                    res_infl = coefficients[4],
                    HML = coefficients[5],
                    SMB = coefficients[6],
                    Mkt.RF = coefficients[7])
  coefficients_df <- rbind(coefficients_df, row)
}

coefficients_df
coefficients_df$Company = names(regs_beta_exoff) #added column with company name



#Get the historical mean return of every stock

```

```r
historical_mean_returns <- finalmonthly %>%
  group_by(Company) %>%
  summarise(mean_return = mean(Return, na.rm = TRUE))

# View the resulting dataframe with mean returns by company
print(historical_mean_returns)


#merge two data sets
multibeta = merge(coefficients_df, historical_mean_returns, by="Company")








#################################################
#Test the validity of the multi-beta relationship
#################################################
```

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.