

ASSET PRICING - EMPIRICAL APPLICATION 1

FACTORIAL MODEL AND RISK PREMIUM DECOMPOSITION - APT

Luis Miguel Fonseca
Stéphane Eloundou Mvondo
Natalia Cárdenas Frías

November 27, 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 stocks¹ is compensated with k risk premia associated with the k *common factors* that the portfolio is exposed to.

1 Data and Framework

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 countries² 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

¹Let $j \in \{1, \dots, J\}$ 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

²Initially 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.

describe frequency of data

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 at the

| 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

1.1.1 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 same as the CAC40³, 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 an 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 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

³Not 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.

Appendix A). 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 on it.

1.2 Factorial model and multibeta relationship

For the sake of completeness, this section summarizes the lecture on the [Ross \[1976\]](#) model and the multibeta relationship as they are the theoretical foundation of the empirical application.

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 . The sources of risk are two-fold. 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 $u_j \perp u_{j'} \forall j \neq j'$, and uncorrelated with aggregate risk $\text{corr}(u_j, f_k) = 0, \forall j, k$ which is a required assumption to perform the estimations that will follow. 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.

2 Empirical strategy and implementation

2.1 Identify the risk factors

2.1.1 Exogeneous factors

2.1.2 Endogeneous factors

2.1.3 French-Fama factors

2.2 Estimate the β_k coefficients from the factorial model

2.3 Estimate the λ_k parameters from the multibeta relationship

2.4 Test the validity of the multibeta relationship

Conclusion

Appendix A Code - Data gathering and cleaning

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 AP1 - 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 eurostatapiclient 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', 'tfl', 'danone',
36     'dassault-aviation', 'air-france-klm', 'accor', 'bic', 'hermes-international',
37     'jcdcaux', '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']
```

```

42
43
44
45 %% Scrap firm level data for French and Fama factors
46 # List of companies' URLs
47 mktcap_urls = ['https://companiesmarketcap.com/' + x + '/marketcap/' for x in company_names_lower
   [:-2]]
48 pricebook_urls = ['https://companiesmarketcap.com/' + x + '/pb-ratio/' for x in
    company_names_lower[:-2]]
49
50
51 # Scrapping functions
52 def scrape_market_cap(url, company_name):
53     response = requests.get(url)
54     if response.status_code == 200:
55         soup = BeautifulSoup(response.content, 'html.parser')
56         table_body = soup.find('table', class_='table').find('tbody')
57         if table_body:
58             data = []
59             rows = table_body.find_all('tr')
60             for row in rows:
61                 cols = row.find_all('td')
62                 if len(cols) >= 2:
63                     year = cols[0].text.strip()
64                     market_cap = cols[1].text.strip()
65                     #variation = cols[2].text.strip()
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     return pd.DataFrame(data)

```

```

86     return None
87
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, Ubisolt, 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)
132                  ) else np.nan)

```

```

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, start, end
188
189 ### Endogeneous factor: long term inflation expectation from external file
190
191 #upload the Agence France Tresor data
192 pi_endo= pd.read_excel('2023_11_01_rend_tit_ref_oatei.xls', skiprows=[0,1,2,3,4], usecols
    =[0,3])
193 pi_endo.columns = ['Date', "Breakeven"]
194
195 pi_endo["Date"] = pd.to_datetime(pi_endo["Date"])
196 pi_endo= pi_endo.set_index(pi_endo["Date"])
197 pi_endo.drop(columns=['Date'])
198
199 #get monthly data
200 piendo_month = pi_endo.resample('1M').mean(numeric_only=True)
201
202 #get yearly data
203 piendo_year = pi_endo.resample('1Y').mean(numeric_only=True)
204
205 del pi_endo
206
207 ### French and Fama - their data
208
209 df = pd.read_csv('Europe_3_Factors.csv',skiprows=[0,1,2])
210
211 #montly data, need to fix dates
212 frenchfama_month = df.iloc[:399,:]
213
214 frenchfama_month['Unnamed: 0'] = frenchfama_month['Unnamed: 0'].astype(str) # Convert to
    string for manipulation
215 frenchfama_month['Year'] = frenchfama_month['Unnamed: 0'].str[:4] # Extract year from the
    encoded date
216 frenchfama_month['Month'] = frenchfama_month['Unnamed: 0'].str[4:] # Extract month from the
    encoded date
217 frenchfama_month['Date'] = pd.to_datetime(dict(year=frenchfama_month['Year'], month=
    frenchfama_month['Month'], day=1))
218 frenchfama_month.drop(['Year', 'Month', 'Unnamed: 0'], axis=1, inplace=True)
219 frenchfama_month = frenchfama_month.set_index(frenchfama_month['Date'])
220 frenchfama_month = frenchfama_month.drop(columns=["Date"])
221 frenchfama_month = frenchfama_month.loc['2002-01-01':]
222 frenchfama_month = frenchfama_month.astype(float)

```

```

223
224
225 #yearly data
226 frenchfama_year = df.iloc[402:,:]
227 frenchfama_year["Unnamed: 0"] = pd.to_datetime(frenchfama_year['Unnamed: 0'])
228 frenchfama_year.rename(columns={"Unnamed: 0": 'Date'}, inplace=True)
229 frenchfama_year = frenchfama_year.set_index(frenchfama_year['Date'])
230 frenchfama_year = frenchfama_year.drop(columns=["Date"])
231 frenchfama_year = frenchfama_year.loc['2002-01-01':]
232 frenchfama_year = frenchfama_year.astype(float)
233
234 del df
235
236
237 #%% Macro data
238 #APIs didn't work as planned
239
240 ## PIB Q
241 pib = pd.read_csv("ECB_PIB.csv")
242 pib.columns = ['Date', 'Q', 'pib']
243 pib['Date'] = pd.to_datetime(pib['Date'])
244 pib = pib.drop(columns=['Q'])
245 pib = pib.set_index(pib['Date'])
246 pib = pib.loc['2002-01-01':]
247 pib = pib.drop(columns=['Date'])
248
249 #monthly
250 pib_monthly = pib.resample('MS').ffill()
251 #yearly
252 pib_year = pib.resample('1Y').last()
253
254 del pib
255
256 ##risk free AAA
257 rkfreeAAA = pd.read_csv('ECB_yield.csv')
258 rkfreeAAA.columns=['Date', "time", "rkfreeAAA"]
259 rkfreeAAA['Date'] = pd.to_datetime(rkfreeAAA['Date'])
260 rkfreeAAA = rkfreeAAA.set_index(rkfreeAAA['Date'])
261 rkfreeAAA = rkfreeAAA.drop(columns=['time', "Date"])
262
263 rkfreeAAA_monthly = rkfreeAAA.resample("1M").mean(numeric_only=True)
264 rkfreeAAA_year = rkfreeAAA.resample("1Y").mean(numeric_only=True)
265
266 del rkfreeAAA
267
268 ##HICP
269 pi_month = pd.read_csv("HICP.csv", sep=';', encoding = 'latin1', skiprows=[0,1,2,3], usecols
    = [0,1], header=None)

```

```

270 pi_month.columns= ['Date', "infl"]
271 pi_month["Date"] = pd.to_datetime(pi_month['Date'], format = "%Y-%m")
272 pi_month=pi_month.set_index(pi_month["Date"])
273 pi_month = pi_month.drop(columns=['Date'])
274
275 #yearly
276 pi_year = pi_month.resample("1Y").mean(numeric_only=True)
277
278
279 ##OAT
280 oat = pd.read_csv('OAT.csv', sep=';', skiprows=[0,1,2,3,4,5], usecols=[0,5], header=None)
281 oat.columns = ['Date', 'oat']
282 oat['Date'] = pd.to_datetime(oat["Date"])
283 oat = oat.set_index(oat["Date"])
284 oat = oat.drop(columns=["Date"])
285 oat = oat.loc["2002-01-01":]
286
287 oat['oat'] = oat['oat'].replace("-", np.nan)
288 oat['oat'] = oat['oat'].str.replace(',', '.').astype(float)
289
290 oat_month = oat.resample('M').mean(numeric_only=True)
291 oat_year = oat.resample('1Y').mean(numeric_only=True)
292
293 del oat
294
295
296 ### Merge the dataframes
297
298 #===MONTHLY
299 #macro stuff, only date index matter
300 monthly = pd.concat([cac_xrate_month, frenchfama_month, oat_month, pi_month, pib_monthly,
301                     piendo_month, rkffreeAAA_monthly], axis=1)
302 monthly= monthly.resample('M').last() #some tables encoded end of month, others on the 1st
303
304 monthly.to_csv('Monthly_series.csv')
305
306 #firm specific data
307 firms_month['Date'] = firms_month['Date'] + pd.offsets.MonthEnd(0) #all other df have eomonth
308                                     date
309 firms_month = firms_month.set_index(['Date'])
310 firms_month = firms_month.set_index('Company', append=True)
311
312 prices_monthly.columns = ['Company', 'value']
313 prices_monthly.set_index('Company', append=True)
314
315 monthly_stock = pd.merge(prices_monthly.reset_index(), firms_month.reset_index(), on =["Date",
316                                     "Company"], how='outer').set_index(["Date", "Company"])

```

```

315 monthly_stock.to_csv("Firm_monthly.csv")
316
317 #merge the two
318 merged_monthly = pd.merge(monthly, monthly_stock, left_index=True, right_index=True, how='
    right')
319
320 #Export df in csv
321 merged_monthly.to_csv("DATA_month.csv")
322
323
324
325 #=== YEARLY
326 yearly = pd.concat([cac_xrate_year, frenchfama_year, oat_year, pi_year, pib_year, piendo_year,
    rkfreeAAA_year], axis=1)
327 yearly = yearly.resample('Y').last()
328
329 yearly.to_csv('Yearly_series.csv')
330
331 # =====
332 # #firm specific
333 # firms_year.rename(columns={"Year":'Date'}, inplace=True)
334 # firms_year['Date'] = firms_year['Date'] + pd.offsets.MonthEnd(0)
335 # firms_year = firms_year.set_index(['Date'])
336 # firms_year = firms_year.set_index('Company', append=True)
337 #
338 # prices_yearly.columns = ['Company', 'value']
339 # prices_yearly = prices_yearly.set_index('Company', append=True)
340 #
341 # yearly_stock = pd.merge(prices_yearly.reset_index(), firms_year.reset_index(), on=["Date",
    "Company"], how='outer').set_index(["Date", "Company"])
342 # =====
343
344 #yearly_stock = monthly_stock.groupby('Company').resample('Y').mean()
345
346 monthly_data_reset = monthly_stock.reset_index()
347 yearly_stock = monthly_data_reset.groupby('Company').resample('Y', on='Date').mean()
348 yearly_stock = yearly_stock.set_index(["Date", "Company"])
349
350 yearly_stock.to_csv("Firm_yearly.csv")
351
352
353 #merge the two
354 merged_year = pd.merge(yearly, yearly_stock, left_index=True, right_index=True, how='right')
355
356 merged_year.to_csv("DATA_yearly.csv")

```

Appendix B Code - Analysis

```

1  ### AP 1
2  ### ncardenasfrias
3
4  pacman::p_load(data.table, tidyverse, gplots, xts, stargazer)
5
6  setwd('/Users/nataliacardenasf/Documents/GitHub/PROJECTS_AP_FE/AP 1')
7
8  df <- fread("DATA_month.csv")
9  as.data.table(df)
10
11 typeof(df)
12 colnames(df)
13
14 ## Identify the risk factors
15 #Need to remove the predicable part to the endogeneous and exogeneous series
16
17
18 ## Estimate the beta coefficients
19
20
21 ## Estimate the lamdas
22
23
24 #Test the validity of the multi-beta relationship

```

3 Data and Framework

3.1 German Stock Market

We decided to consider the German stock market for this analysis because it is a major liquid stock market in Europe. It is also the biggest economy in the Europe, with mayor

3.2 Estimation of the Factors

We need to choose what factors we are going to consider to generate risk premia that affect the return for the investor. In this section we examine the role of different types of factors (i) exogenous, (ii) endogeneous and we examine more closely the three-factor model proposed by [Fama and French \[1993\]](#) in (iii).

3.3 Exogeneous Factors

These are risk factors that are supposed to be orthogonal to the portfolio itself. In particular, it is interesting to consider the role of

3.4 Endogeneous Factors

3.5 French-Fama Factors

[Fama and French \[1993\]](#) can be seen as an extension of the CAPM model. 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).

4 Estimation of the exposure

5 Estimation of the market price of risk(s)

Consider a series of returns for different stock prices of at least 30 over a given period of time and frequency. The goal is to estimate risk premium by choosing a relevant so-called risk-free asset obtained as the return of treasury bond with relevant maturity

Develop econometric analysis which provides the multi-beta relationship 1. Identify the series for the risk factors (endo and exo) and justify choices + including 2 factors proposed by French and Fama 2. Estimate beta coefficients for different stocks with relevant linear regression 3. Estimate market price of different sources of risk retained in analysis with appropriate linear regression

Comment the results from a financial point of view: are the estimated exposures of the different stocks to the different factors in line with expectations

References

- 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.